



Cisco Nexus 1000V Series Switches Deployment Guide

Contents

Overview	3
Audience	3
Introduction	3
Cisco Nexus 1000V Series Components.....	3
Network Policy.....	4
Cisco Nexus 1000V Series Theory of Operation	4
VMware Networking Overview	4
System Overview	6
Virtual Chassis.....	6
Network Policy Management.....	6
Policy Mobility.....	7
Installation.....	7
Virtual Supervisor Module.....	7
Cisco NX-OS Software	8
VSM Networking.....	8
Control Interface.....	9
Management Interface	9
Packet Interface	10
Communication Between VSM and VMware vCenter	10
Cisco Nexus 1000V Series VMware vCenter Server Extension	10
Opaque Data	11
Virtual Ethernet Module	11
Switch Port Interfaces	12
Switch Forwarding.....	12
MAC Address Learning.....	13
Loop Prevention	13
VEM-VSM Communication	13
Domain ID.....	14
Packet Interface Communication	14
Port Profiles	14
Live Policy Changes.....	15
Uplink Profiles.....	15
System VLANs	15
Cisco Nexus 1000V Series Network Design	16
Network Design Considerations	16
Design Goals	16
Traffic Classification	16
VLAN Consistency	17
Traffic Separation	17
Upstream Switch Connectivity.....	17
Individual Uplinks	17
PortChannels.....	18
Virtual Port Channel Host Mode.....	18
Load Balancing.....	18
Source-Based Hashing.....	18
Flow-Based Hashing	19
Control Interface Prioritization	19
Spanning Tree Protocol	19
VSM Design.....	19
Virtual Machine Design	19
Adjacency	20
Latency	20
Traditional Cisco Network.....	20
Two-NIC Design Examples.....	20
Four-NIC Design Examples.....	21
Single-PortChannel Alternative	22
Six-NIC Design Examples	23
For More Information	24

Overview

This document provides design and configuration guidance for deploying the Cisco Nexus™ 1000V Series Switches with VMware vSphere 4.0. For detailed configuration documentation, please refer to the respective Cisco® and VMware product configuration guides. Links to the product configuration guides can be found in the “For More Information” section of this document.

Audience

This document is intended for network architects, network engineers, virtualization administrators, and server administrators interested in understanding and deploying VMware vSphere 4.0 hosts in a Cisco data center environment.

Introduction

Cisco Nexus 1000V Series Switches are virtual machine access switches that are an intelligent software switch implementation for VMware vSphere environments running the Cisco NX-OS Software operating system. Operating inside the VMware ESX hypervisor, the Cisco Nexus 1000V Series supports Cisco VN-Link server virtualization technology to provide:

- Policy-based virtual machine connectivity
- Mobile virtual machine security and network policy
- Nondisruptive operational model for your server virtualization, and networking teams

When server virtualization is deployed in the data center, virtual servers typically are not managed the same way as physical servers. Server virtualization is treated as a special deployment, leading to longer deployment time, with a greater degree of coordination among server, network, storage, and security administrators. With the Cisco Nexus 1000V Series, you can have a consistent networking feature set and provisioning process all the way from the virtual machine access layer to the core of the data center network infrastructure. Virtual servers can now use the same network configuration, security policy, diagnostic tools, and operational models as their physical server counterparts attached to dedicated physical network ports. Virtualization administrators can access predefined network policy that follows mobile virtual machines to help ensure proper connectivity, saving valuable time to focus on virtual machine administration. This comprehensive set of capabilities helps you deploy server virtualization faster and gain its benefits sooner.

Developed in close collaboration with VMware, the Cisco Nexus 1000V Series is certified by VMware to be compatible with VMware vSphere, vCenter, ESX, and ESXi, and with many other VMware vSphere features. You can use the Cisco Nexus 1000V Series to manage your virtual machine connectivity with confidence in the integrity of the server virtualization infrastructure.

Cisco Nexus 1000V Series Components

The Cisco Nexus 1000V Series provides Layer 2 switching advanced networking functions and a common network management model in a virtualized server environment by replacing the virtual switch within VMware vSphere. The Cisco Nexus 1000V Series manages a data center as defined in VMware vCenter Server. Each server in the data center is represented as a line card in the Cisco Nexus 1000V Series and can be managed as if it were a line card in a physical Cisco switch.

The Cisco Nexus 1000V Series implementation has two main components:

- Virtual supervisor module (VSM)
- Virtual Ethernet module (VEM)

These two components together make up the Cisco Nexus 1000V Series, with the VSM providing the management plane and the VEM providing the data plane.

Network Policy

A unique aspect of the Cisco Nexus 1000V Series is the way network policy is defined and deployed. Today, a network administrator would typically configure each interface on a switch one at a time. For Cisco switches, this typically means entering configuration mode and applying a series of switch commands that define the interface configuration.

Configuration may be manually applied to multiple interfaces on the same switch or different switches, connected to similar types of servers. This management model requires server administrators to depend on network administrators to reconfigure the network each time a server is brought online. This process can create unwanted delays in deploying new servers.

In a VMware environment, server administrators are required to configure network policy, using the VMware virtual switch (vSwitch) and port group features, to match the policy configured on the upstream physical switches. This requirement removes a dependency on the network administrator for virtual access layer switch configuration (the first network hop in the data center) and makes addition of a new virtual machine as simple as selecting the appropriate predefined port group. This approach creates operational and security challenges such as policy enforcement and troubleshooting, but it addresses many delays in deploying new virtual machines (no physical infrastructure to configure).

The Cisco Nexus 1000V Series provides an ideal model in which network administrators define network policy that virtualization or server administrators can use as new similar virtual machines are added to the infrastructure. Policies defined on the Cisco Nexus 1000V Series are exported to VMware vCenter Server to be used and reused by server administrators as new virtual machines require access to a specific network policy. This concept is implemented on the Cisco Nexus 1000V Series using a feature called port profiles. The Cisco Nexus 1000V Series with the port profile feature eliminates the requirement for the virtualization administrator to create or maintain vSwitch and port group configurations on any of their VMware ESX hosts.

Port profiles create a unique collaborative model, giving server administrators the autonomy to provision new virtual machines without waiting for network reconfigurations to be implemented in the physical network infrastructure. For network administrators, the combination of the Cisco Nexus 1000V Series feature set and the capability to define a port profile using the same syntax as for existing physical Cisco switches helps ensure that consistent policy is enforced without the burden of managing individual switch ports. The Cisco Nexus 1000V Series solution also provides a consistent network management, diagnostic, and troubleshooting interface to the network operations team, allowing the virtual network infrastructure to be managed like the physical infrastructure.

Cisco Nexus 1000V Series Theory of Operation

This section describes the main concepts and components of the Cisco Nexus 1000V Series and how the components interact.

VMware Networking Overview

To understand the Cisco Nexus 1000V Series, you must first understand the basics of the VMware networking model. VMware networking consists of virtual network interface cards (vNICs) of various types, the physical NICs on the hosts, and virtual switches to interconnect them.

Each virtual machine has one or more vNICs. These vNICs are connected to a virtual switch (such as the Cisco Nexus 1000V Series) to provide network connectivity to the virtual machine. The guest OS sees the vNICs as physical NICs. VMware can emulate several popular NIC types (vLance and Intel e1000), so the guest OS can use

standard device drivers for these vNICs. Alternatively, the VMware vmxnet interface type can be used; this interface type requires VMware drivers on the guest OS.

Hosts running VMware ESX have a virtual management port called vswif, sometimes referred to as the service console interface. This interface is used for communication with VMware vCenter Server, to manage the box directly with the VMware vSphere Client, or to use Secure Shell (SSH) to log in to the host's command-line interface (CLI). VMware ESXi hosts do not use vswif interfaces due to their lack of a service console OS.

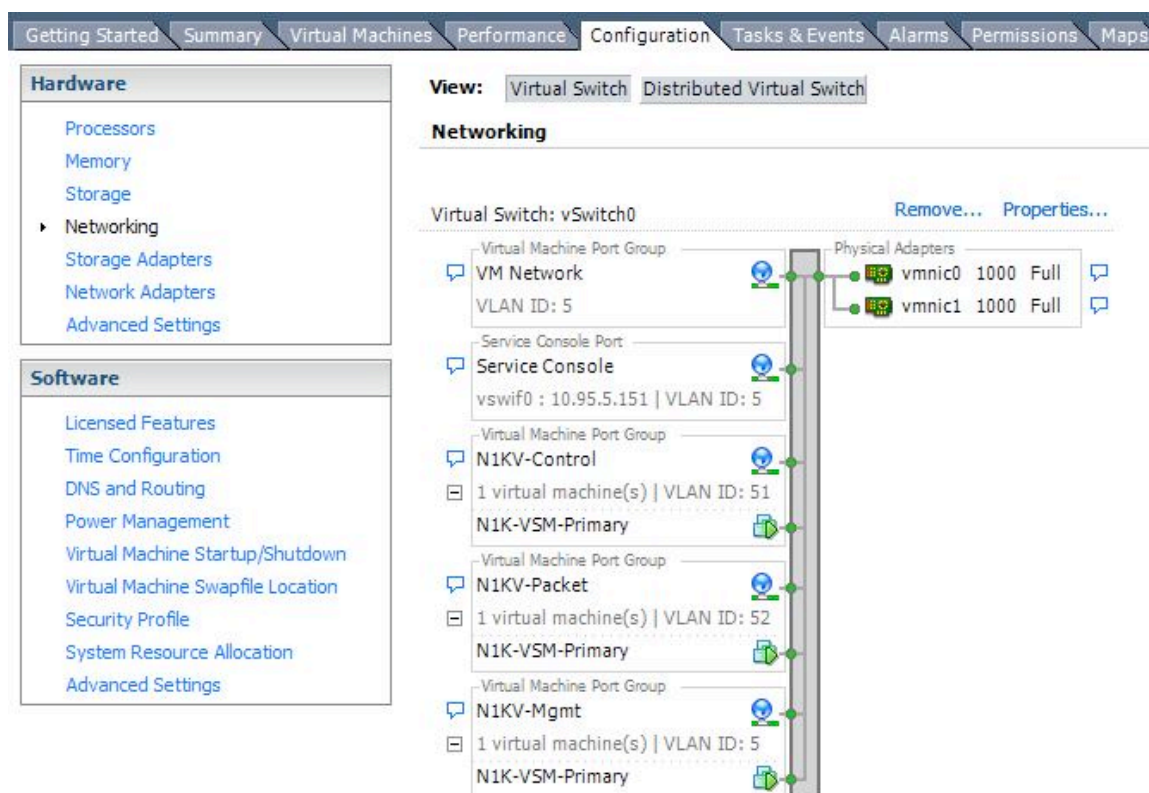
Each host also has one or more virtual ports called virtual machine kernel NICs (vmknics). These are used by VMware ESX for Small Computer Systems Interface over IP (iSCSI) and Network File System (NFS) access, as well as by VMware VMotion. On a VMware ESXi system, a vmknic is also used for communication with VMware vCenter Server.

The physical NICs on an VMware ESX host, called virtual machine NICs (VMNICs), are used as uplinks to the physical network infrastructure.

The virtual and physical NICs are all tied together by virtual switches. VMware provides two types of virtual switches. The standard vSwitch is individually created for each host. VMware vNetwork Distributed Switch (vDSs) provides a consistent virtual switch across a set of physical hosts. The Cisco Nexus 1000V Series is implemented as a type of vDS.

Each vNIC is connected to a standard vSwitch or vDS through a port group. Each port group belongs to a specific vSwitch or vDS and specifies a VLAN or set of VLANs that a VMNIC, vswif, or vmknic will use. The port group specifies other network attributes such as rate limiting and port security. Virtual machines are assigned to port groups during the virtual machine creation process or by editing the virtual machine properties later (Figure 1).

Figure 1. VMware vSwitch Network Configuration



System Overview

The Cisco Nexus 1000V Series is a software-based switch that extends across multiple hosts running VMware ESX or ESXi 4.0. It consists of two components: the virtual supervisor module, or VSM, and the virtual Ethernet module, or VEM. The VSMS are deployed in pairs that act as the switch's supervisors. One or more VEMs are deployed; these act like line cards within the switch.

The VSM is a virtual appliance that can be installed independent of the VEM: that is, the VSM can run on an VMware ESX server that does not have the VEM installed. The VEM is installed on each VMware ESX server to provide the packet-forwarding capability. The VSM pair and VEMs make up a single Cisco Nexus 1000V Series Switch, which appears as a single modular switch to the network administrator.

Each instance of the Cisco Nexus 1000V Series Switch is represented in VMware vCenter Server as a vNetwork Distributed Switch, or vDS. A vDS is a VMware concept that enables a single virtual switch to span multiple VMware ESX hosts. The Cisco Nexus 1000V Series is created in VMware vCenter Server by establishing a link between the VSM and VMware vCenter Server using the VMware VIM API.

VMware's management hierarchy is divided into two main elements: a data center and a cluster. A data center contains all components of a VMware deployment, including hosts, virtual machines, and network switches, including the Cisco Nexus 1000V Series.

Note: A VMware ESX host can have only a single VEM installed.

Within a VMware data center, the user can create one or more clusters. A cluster is a group of hosts and virtual machines that form a pool of CPU and memory resources. A virtual machine within a cluster can be run on or migrated to any host in the cluster. Hosts and virtual machines do not need to be part of a cluster; they can exist on their own within the data center as well.

Virtual Chassis

The Cisco Nexus 1000V Series uses a virtual chassis model to represent a pair of VSMS and their associated VEMs. Like any Cisco chassis base platform, the Cisco Nexus 1000V Series virtual chassis has slots and modules, or line cards, associated with it. The VSMS are always associated with slot numbers 1 and 2 in the virtual chassis. The VEMs are sequentially assigned to slots 3 through 66 based on the order in which their respective hosts were added to the Cisco Nexus 1000V Series Switch.

Network Policy Management

Software-based virtual switching presents new data center management challenges. The traditional management model calls for server administrators to manage the OS and applications while the network administrator manages the switches and their associated policies. The link between the server and switch, usually a Category 5 cable, is a clear boundary between administrative roles. The Cisco Nexus 1000V Series management model calls for collaboration between server and network administrators who are maintaining the configuration of the same piece of hardware: a VMware ESX host.

Server and network administrators are separate entities with separate responsibilities. The Cisco Nexus 1000V Series maintains this separation, with distinct roles for each administrator. Collaboration between the administrators is required, but the Cisco Nexus 1000V Series is designed to provide server and network administrators with a high level of autonomy.

The Cisco Nexus 1000V Series provides a feature called port profiles to simplify network provisioning with VMware. Port profiles create a virtual boundary between server and network administrators. Port profiles are network policies that are defined by the network administrator and exported to VMware vCenter Server. Within VMware vCenter Server, port profiles appear as VMware port groups in the same locations as a traditional VMware port group would. The server administrator is free to use the port profile in the same manner as a port group defined by VMware.

```
Switch# show port-profile name Basic-VM
port-profile Basic-VM
...
  config attributes:
    switchport mode access
    switchport access vlan 53
    no shutdown
...
```

When a new virtual machine is provisioned, the server administrator selects the appropriate port profile. The Cisco Nexus 1000V Series creates a new switch port based on the policies defined by the port profile. The server administrator can reuse the port profile to provision similar virtual machines as needed.

Port profiles are also used to configure the physical NICs in a server. These port profiles, known as uplink port profiles, are assigned to the physical NICs as part of the installation of the VEM on a VMware ESX host.

Policy Mobility

Network policies enforced by a port profile follow the virtual machine throughout its lifecycle, whether the virtual machine is being migrated from one server to another, suspended, hibernated, or restarted. In addition to migrating the policy, the Cisco Nexus 1000V Series moves the virtual machine's network state, such as the port counters and flow statistics. Virtual machines participating in traffic monitoring activities, such as Cisco NetFlow or Encapsulated Remote Switched Port Analyzer (ERSPAN), can continue these activities uninterrupted by VMotion operations.

Installation

Installation of the Cisco Nexus 1000V Series is outside the scope of this document. This section describes the installation at a very high level for conceptual completeness. For guidance and detailed instructions about installation, please refer to the Cisco Nexus 1000V Series Switches installation guide.

1. The network administrator installs the VSM and defines one or more uplink port profiles.
2. The server administrator uses a standard web browser to download the Cisco Nexus 1000V Series plug-in from the VSM and installs it in VMware vCenter Server.
3. The network administrator creates a link between the VSM and VMware vCenter Server. This process creates an instance of the Cisco Nexus 1000V Series within VMware vCenter Server.
4. The server administrator adds VMware ESX hosts to the Cisco Nexus 1000V Series using the VMware vSphere Client, assigning the uplink port profiles to the appropriate physical NICs.
5. The network administrator defines one or more port profiles to be used by virtual machines and other virtual interfaces.

At this point, the Cisco Nexus 1000V Series installation is complete. The server administrator can begin assigning port profiles to virtual machines, providing network connectivity to the guest OS.

Virtual Supervisor Module

The VSM provides the management plane functions of the Cisco Nexus 1000V Series. Much like a supervisor module of a Cisco Nexus 7000 Series Switch, the VSM is the single point of management for the network administrator, providing coordination of configuration and functions across VEMs.

Unlike a traditional Cisco switch, in which the management plane is integrated into the hardware, the VSM is deployed as a virtual machine. Running Cisco NX-OS Software, the Cisco Nexus operating system, the Cisco Nexus 1000V Series VSM is installed in a way similar to other virtual machines (such as those running Linux or Microsoft Windows), using either an ISO file or an Open Virtualization Format (OVF) template.

The VSM has virtual machine requirements much like other more traditional guest operating systems. At a high level, the VSM requires a single virtual CPU, 2 GB of dedicated RAM, and three virtual network adapters (more information about these virtual network adapters is provided later in this document).

The Cisco Nexus 1000V Series requires a VSM high-availability deployment model much like dual supervisors in a physical chassis. These two VSMs are deployed in an active-standby configuration, with the first VSM functioning in the primary role and the other VSM functioning in a secondary role. If the primary VSM fails, the secondary VSM will take over.

Note that unlike cross-bar based modular switching platforms, the VSM is not in the data path. General data packets are not forwarded to the VSM to be processed, but rather switched by the VEM directly. In two specific cases, described later in this document, control traffic is processed by the VSM to be coordinated across all VEMs.

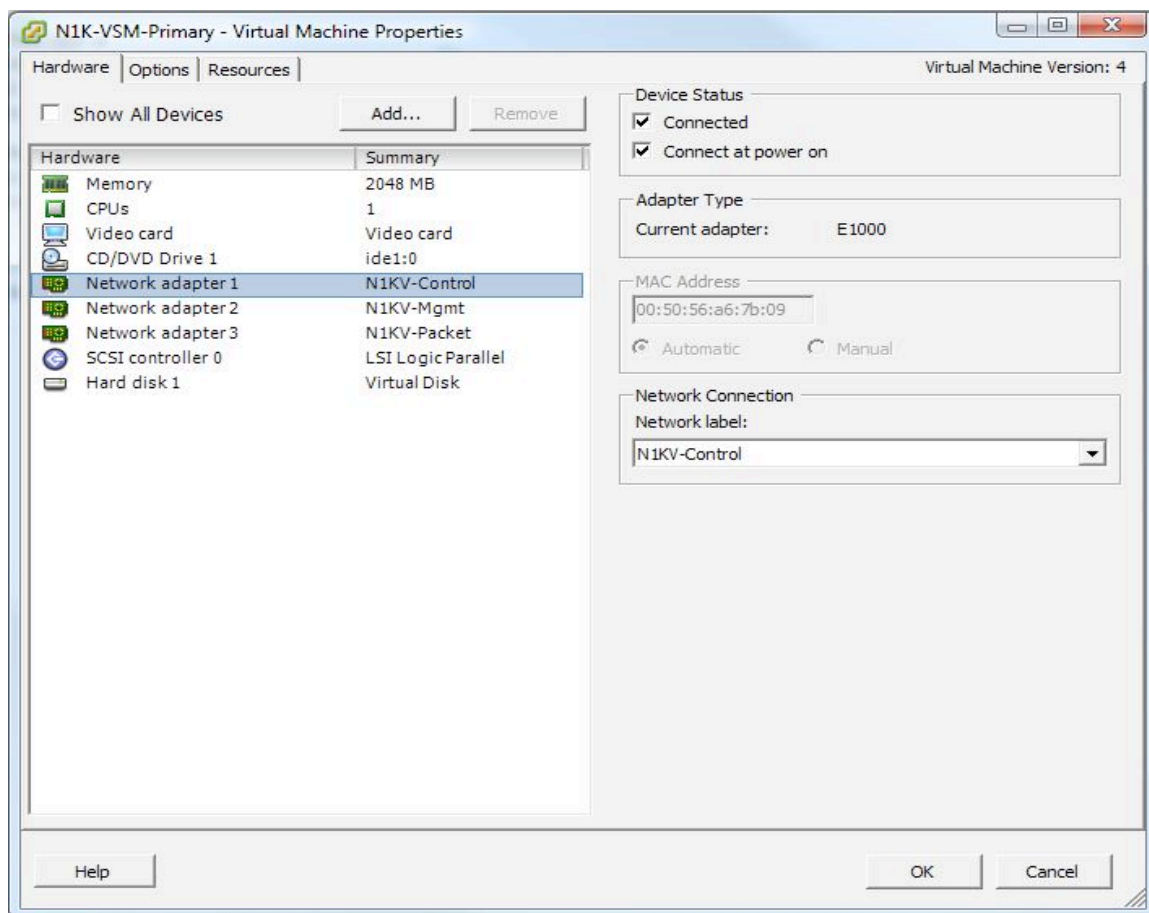
Cisco NX-OS Software

Cisco NX-OS Software is a data center–class operating system built with modularity, resiliency, and serviceability at its foundation. Based on the industry-proven Cisco MDS 9000 SAN-OS Software, Cisco NX-OS helps ensure continuous availability and sets the standard for mission-critical data center environments. The self-healing and highly modular design of Cisco NX-OS makes zero-impact operations a reality and enables exceptional operational flexibility. Focused on the requirements of the data center, Cisco NX-OS provides a robust and rich feature set that fulfills the Ethernet and storage networking requirements of present and future data centers. With a CLI like that of Cisco IOS® Software, Cisco NX-OS provides state-of-the-art implementations of relevant networking standards as well as a variety of true data center–class Cisco innovations.

VSM Networking

The VSM is a virtual machine that requires three vNICs. Each vNIC has a specific function, and all are fundamental to the operation of the Cisco Nexus 1000V Series. For definition of the VSM virtual machine properties, the vNICs require the Intel e1000 network driver (Figure 2).

The e1000 network driver may not be the default driver when the virtual machine definition is built. Also note that the e1000 driver may not be an available option based on the operating system selected when the virtual machine is defined. You can manually change the driver in the virtual machine configuration file stored with the virtual machine. Selecting “Other Linux 64-bit” as the operating system enables the selection of the e1000 driver and sets it as the default driver.

Figure 2. Proper Virtual Supervisor Module Networking Configuration

Note: Please refer to the Cisco Nexus 1000V Series Switches installation guide for detailed VSM installation instructions.

Control Interface

The control interface is a Layer 2 interface used to communicate with the VEMs. This interface handles low-level control packets such as heartbeats as well as any configuration data that needs to be exchanged between the VSM and VEM. Because of the nature of the traffic carried over the control interface, it is the most important interface in the Cisco Nexus 1000V Series solution.

The control interface is always the first interface on the VSM and is usually labeled “Network Adapter 1” in the virtual machine network properties.

Management Interface

The management interface is the interface that appears as the mgmt0 port on a Cisco switch. As with the management interfaces of other Cisco switches, an IP address is assigned to mgmt0. 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.

The management interface is always the second interface on the VSM and is usually labeled “Network Adapter 2” in the virtual machine network properties.

Packet Interface

The packet interface is a Layer 2 interface that is used to carry network packets that need to be coordinated across the entire Cisco Nexus 1000V Series Switch. This interface is used for only two type of control traffic: Cisco Discovery Protocol and Internet Group Management Protocol (IGMP) control packets.

The packet interface is always the third interface on the VSM and is usually labeled “Network Adapter 3” in the virtual machine network properties.

Communication Between VSM and VMware vCenter

The VSM maintains a link to VMware vCenter Server that is used to maintain the definition of the Cisco Nexus 1000V Series within VMware vCenter Server as well as propagate port profiles.

The server and network administrators both have roles in establishing the link between the Cisco Nexus 1000V Series and VMware vCenter Server. First, the server administrator installs the Cisco Nexus 1000V Series VMware vCenter Server plug-in (described later in this document). After the plug-in is installed, the network administrator can define the svcs connection. The svcs connection defines the link between the VSM and VMware vCenter Server. The connection contains the following parameters:

- VMware vCenter Server IP address
- Communication protocol (always VMware VIM over HTTPS)
- Name of the VMware data center in which the VMware ESX hosts reside

After the connection is defined, the network administrator enables the connection, which establishes the link and creates the instance of the Cisco Nexus 1000V Series within VMware vCenter Server. Each VSM contains a unique extension key used to bind that specific VSM to VMware vCenter Server.

During the process in creating the Cisco Nexus 1000V Series within VMware vCenter Server, the VSM pushes down any port profiles that are already defined as well as important information required for VEM installation, called opaque data. The opaque data provides limited configuration details to the VEM so that it can communicate with the VSM after installation

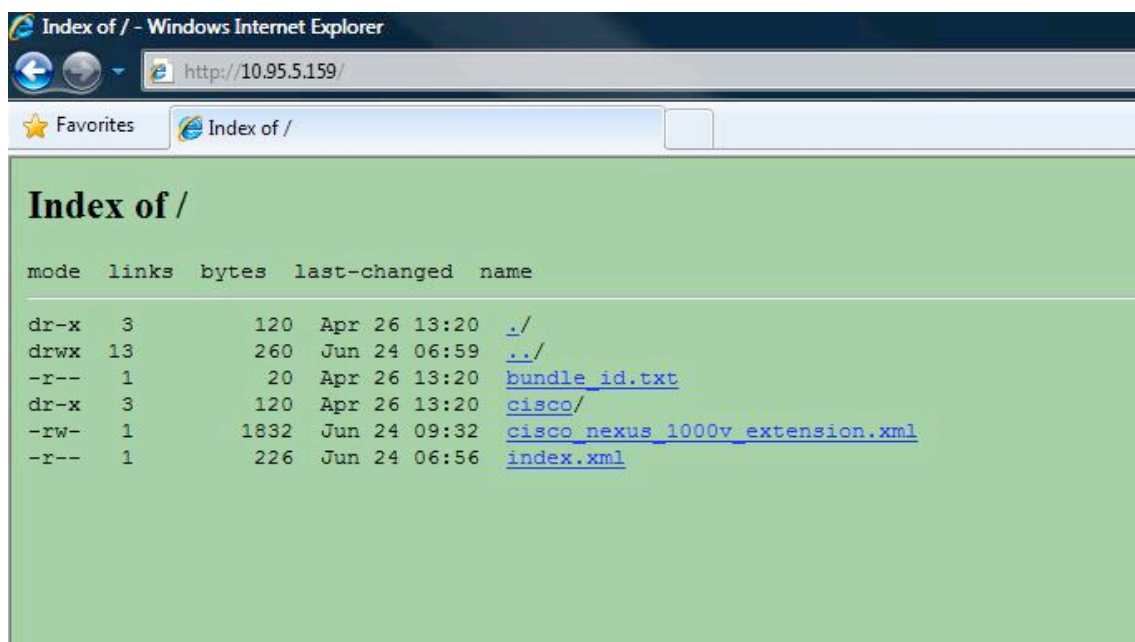
The VSM is considered the authoritative container for all configuration information. If the connection between the VSM and VMware vCenter Server is disrupted, the VSM helps ensure that any configuration changes that have been made during this period of disrupted communication are propagated to VMware vCenter Server when the link is restored.

After the connection between the VSM and VMware vCenter Server is established, the link is primarily used to propagate new port profiles and any changes to existing ones.

Cisco Nexus 1000V Series VMware vCenter Server Extension

VMware vCenter Server is an extensible application that allows third-party management plug-ins, thus enabling external applications to extend the capabilities of VMware vCenter Server and its companion GUI, VMware vSphere Client. The Cisco Nexus 1000V Series uses a VMware vCenter Server extension to properly display a representation of the Cisco Nexus 1000V Series and its main components within VMware vSphere Client.

The Cisco Nexus 1000V Series extension is a small XML file (`cisco_nexus_1000v_extension.xml`) that is downloaded from the VSM’s management IP address using a web browser. This plug-in must be installed before the VSM can establish a link to VMware vCenter Server (Figure 3).

Figure 3. Downloading the XML Extension File

Opaque Data

Opaque data is a collection of Cisco Nexus 1000V Series configuration parameters that is maintained by the VSM and propagated to VMware vCenter Server when the link between the two is established. The opaque data contains configuration details that each VEM needs to establish connectivity to the VSM during VEM installation.

Among other content, the opaque data contains:

- Switch domain ID
- Switch name
- Control and packet VLAN IDs
- System port profiles

When a new VEM comes online, either after initial installation or upon reboot of an VMware ESX host, it is essentially an unprogrammed line card. To be properly configured, the VEM needs to communicate with the VSM. VMware vCenter Server automatically pushes the opaque data to the VEM, which the VEM uses to establish communication with the VSM and download the appropriate configuration data.

Virtual Ethernet Module

The VEM provides the Cisco Nexus 1000V Series with network connectivity and forwarding capabilities much like a line card within a modular switching platform. Unlike multiple line cards within a single chassis, each VEM acts as an independent switch from a forwarding perspective.

The VEM is tightly integrated with VMware ESX. The VEM is installed on each VMware ESX host as a kernel component, in contrast to most third-party networking services for VMware, which are usually installed as virtual machines.

Unlike with the VSM, the VEM's resources are unmanaged and dynamic. Although the storage footprint of the VEM is fixed (approximately 6.4 MB of disk space), RAM utilization on the VMware ESX host is variable, based on the configuration and scale of the Cisco Nexus 1000V Series deployment. In a typical configuration, each VEM can be

expected to require 10 to 50 MB of RAM, with an upper hard limit of 150 MB for a fully scaled solution with all features turned on and utilized to their design limits.

Each instance of the Cisco Nexus 1000V Series is composed of two VSMs and one or more VEMs. The maximum number of VEMs supported by a pair of VSMs is 64.

Switch Port Interfaces

The Cisco Nexus 1000V Series supports multiple switch port types used for internal and external connectivity: virtual Ethernet (vEth), Ethernet (Eth), and PortChannel (Po). The most common port type within a Cisco Nexus 1000V Series environment is a new concept called a virtual Ethernet interface. This interface type represents the switch port connected to a virtual machine's vNIC or connectivity to specialized interface types such as the vswif or vmknic interface.

A vEth interface has several characteristics that differentiate it from other interface types. Besides the obvious fact that vEth interfaces are virtual and therefore have no associated physical component, the interface naming convention is unique. Unlike a traditional Cisco interface, a vEth interface's name does not indicate the module with which the port is associated. Whereas a traditional physical switch port may be notated as GigX/Y, where X is the module number and Y is the port number on the module, a vEth interface is notated like this: vEthY. This unique notation is designed to work transparently with VMotion by keeping the interface name the same regardless of the location of the associate virtual machine.

The second characteristic that makes a vEth interface unique is its transient nature. A given vEth interface may appear or disappear based on the status of the virtual machine connected to it. The mapping of a virtual machine's vNIC to a vEth interface is static. When a new virtual machine is created, a vEth interface is also created for each of the virtual machine's vNICs. The vEth interfaces will persist as long as the virtual machine exists. If the virtual machine is temporarily down (the guest OS is shut down), the vEth interfaces will remain inactive but still bound to that specific virtual machine. If the virtual machine is deleted, the vEth interfaces will become available for connection to newly provisioned virtual machines.

The Cisco Nexus 1000V Series contains two interface types related to the VMNICs (physical NICs) within a VMware ESX host. An Ethernet, or Eth, interface is the Cisco Nexus 1000V Series' representation of a VMNIC. An Eth interface is represented in standard Cisco interface notation (EthX/Y) using the Cisco NX-OS naming convention "Eth" rather than a speed such as "Gig" or "Fast," as is the custom with Cisco IOS Software. These Eth interfaces are module specific and are designed to be fairly static within the environment.

PortChannels are the third interface type supported by the Cisco Nexus 1000V Series. A PortChannel is an aggregation of multiple Eth interfaces on the same VEM.

Note: PortChannels are not created by default and must be explicitly defined.

Switch Forwarding

In many ways the Cisco Nexus 1000V Series Switches are similar to physical Ethernet switches. For packet forwarding, the Cisco Nexus 1000V Series uses the same techniques that other Ethernet switches apply, keeping a MAC address-to-port mapping table used to determine where packets should be forwarded.

The Cisco Nexus 1000V Series maintains forwarding tables in a slightly different manner than other modular switches. Unlike physical switches with a centralized forwarding engine, each VEM maintains a separate forwarding table. There is no synchronization between forwarding tables on different VEMs. In addition, there is no concept of forwarding from a port on one VEM to a port on another VEM. Packets destined for a device not local to a VEM are forwarded to the external network, which in turn may forward the packets to a different VEM.

MAC Address Learning

This distributed forwarding model within a centrally managed switch is demonstrated by the way the Cisco Nexus 1000V Series handles MAC address learning. A MAC address can be learned multiple times within a single Cisco Nexus 1000V Series Switch in either of two ways: statically or dynamically. Static entries are automatically generated for virtual machines running on the VEM; these entries do not time out. For devices not running on the VEM, the VEM can learn a MAC address dynamically, through the physical NICs in the server.

Each VEM maintains a separate MAC address table. Thus, a single Cisco Nexus 1000V Series Switch may learn a given MAC address multiple times: as often as once per VEM. For example, one VEM may be hosting a virtual machine, and the virtual machine's MAC address will be statically learned on the VEM. A second VEM, in the same Cisco Nexus 1000V Series Switch, may learn the virtual machine's MAC address dynamically. Thus, within the Cisco NX-OS CLI, you may see the virtual machine's MAC address twice: a dynamic entry and a static entry.

Loop Prevention

Another differentiating characteristic of the Cisco Nexus 1000V Series is that it does not run Spanning Tree Protocol. Although this might seem to be a significant departure from other Ethernet switches, potentially causing catastrophic network loops, in reality the Cisco Nexus 1000V Series implements a simple and effective loop-prevention strategy that does not require Spanning Tree Protocol.

Because the Cisco Nexus 1000V Series does not participate in Spanning Tree Protocol, it does not respond to Bridge Protocol Data Unit (BPDU) packets, nor does it generate them. BPDU packets that are received by Cisco Nexus 1000V Series Switches are dropped.

The Cisco Nexus 1000V Series uses a simple technique to prevent loops. Like a physical Ethernet switch, the Cisco Nexus 1000V Series performs source and destination MAC address lookups to make forwarding decisions. The VEM applies loop-prevention logic to every incoming packet on Ethernet interfaces. This logic is used to identify potential loops. Every ingress packet on a physical Ethernet interface is inspected to help ensure that the destination MAC address is internal to the VEM. If the destination MAC address is external, the Cisco Nexus 1000V Series will drop the packet preventing a loop back to the physical network.

Note: The Cisco Nexus 1000V Series prevents loops between the VEMs and the first-hop access switches without the use of Spanning Tree Protocol. However, this feature does not mean that Spanning Tree Protocol should be disabled on any access switches. Spanning Tree Protocol is still required by the access switches to prevent loops elsewhere in the physical topology.

VEM-VSM Communication

Like the VSM, each VEM has a control and packet interface. These interfaces are unmanaged and not directly configurable by the end user. The VEM uses the opaque data provided by VMware vCenter Server to configure the control and packet interfaces with the correct VLANs. The VEM then applies the correct uplink port profile to the control and packet interfaces to establish communication with the VSM.

After the VSM recognizes the VEM, a new module will be virtually inserted into the Cisco Nexus 1000V Series Switch's virtual chassis. The VSM CLI will notify the network administrator that a new module has powered on, much as with a physical chassis.

The module assignment is sequential, meaning that the VEM will be assigned the lowest available module number between 3 and 66. When a VEM comes online for the first time, the VSM assigns the module number and tracks that module using the unique user ID (UUID) of the VMware ESX server, helping ensure that if the VMware ESX host loses connectivity or is powered down for any reason, the VEM will retain its module number when the host comes back online.

The VSM maintains a heartbeat with its associated VEMs. This heartbeat is transmitted at 2-second intervals. If the VSM does not receive a response within 8 seconds, the VSM considers the VEM removed from the virtual chassis. If the VEM is not responding because of a connectivity problem, the VEM will continue to switch packets in its last known good state. When communication is restored between a running VEM and the VSM, the VEM is reprogrammed, causing a slight (1 to 15 second) pause in network traffic.

All communication between the VSM and VEM is encrypted using a 128-bit algorithm.

Domain ID

A physical Ethernet switch typically passes control information between the data plane and the control plane using an internal network (Cisco switches use an internal network called the Ethernet out-of-band channel [EoBC]) that is not exposed to the network administrator. This internal network is isolated by design. In the case of the Cisco Nexus 1000V Series, control packets between the VSM and VEM traverse the physical network. A potential although highly unlikely scenario is the case in which a VEM receives control packets from a VSM that is managing a completely different Cisco Nexus 1000V Series. If the VEM were to respond to such packets (for example, a request to reconfigure an interface), the VEM would not forward packets as expected. To prevent this scenario, the Cisco Nexus 1000V Series implements a solution called domain IDs.

A domain ID is a parameter of the Cisco Nexus 1000V Series that is used to identify a VSM and VEM as relating to one another. The domain ID of the Cisco Nexus 1000V Series is defined when the VSM is first installed and becomes part of the opaque data that is transmitted to VMware vCenter Server.

Each command sent by the VSM to any associated VEMs is tagged with this domain ID. When a VSM and VEM share the same domain ID, the VEM will accept and respond to requests and commands from the VSM. If the VEM receives a command or configuration request that is not tagged with the proper domain ID, that request is ignored. Similarly, if the VSM receives a packet from a VEM that is tagged with the wrong domain ID, it will be ignored.

Packet Interface Communication

The packet interface is used to send selected packets between the VEM and the VSM. This interface is used for only two types of packets: Cisco Discovery Protocol and IGMP control packets.

The VSM presents a unified Cisco Discovery Protocol view to the network administrator through the Cisco NX-OS CLI. When a VEM receives a Cisco Discovery Protocol packet, the VEM retransmits that packet to the VSM so that the VSM can parse the packet and populate the Cisco Discovery Protocol entries in the CLI.

The packet interface is also used to coordinate IGMP across multiple VEMs. For example, when a VEM receives an IGMP join request, that request is sent to the VSM, which coordinates the request across all VEMs in the switch.

Port Profiles

Port profiles are the primary mechanism by which network policy is defined and applied to switch interfaces. A port profile is a collection of interface-level configuration commands that are combined to create a complete network policy.

Port profiles are created on the VSM and propagated to VMware vCenter Server as VMware port groups using the VMware VIM API. After propagation, a port profile appears within VMware vSphere Client and is available to apply to a virtual machines vNICs.

When the server administrator provisions a new virtual machine, a dialog box pertaining to network configuration appears. This dialog box is consistent regardless of the presence of the Cisco Nexus 1000V Series: that is, the workflow for provisioning a new virtual machine does not change when the Cisco Nexus 1000V Series is used. The server administrator selects the port profiles to apply to each of the virtual machine's vNICs.

When the newly provisioned virtual machine is powered on, a vEth interface is created on the Cisco Nexus 1000V Series for each vNIC the virtual machine contains. The vEth inherits the definitions in the selected port profile.

The port profile concept is new, but the configurations in port profiles use the same Cisco syntax used to manage switch ports on traditional switches. The network administrator defines a new port profile in switch configuration mode. Then the network administrator applies the desired interface configuration commands. Then the port profile is marked as enabled and as a VMware port group. This process of enabling the port profile and defining it as a VMware port group pushes the port profile to VMware vCenter Server and it becomes available for use by the server administrator within a few seconds.

Live Policy Changes

Port profiles are not static entities; they are dynamic policies that can change as network needs change. Changes to active port profiles are applied to each switch port that is using the profile. This feature of port profiles is extremely useful when applying new network policies or changing existing policy.

Uplink Profiles

Port profiles are not only used to manage vEth configuration; they are also used to manage the physical NICs within a VMware ESX host. When a port profile is defined, the network administrator determines whether the profile will be used to manage vEth interfaces or physical NICs. By default, the port profile is assumed to be used for vEth management.

To define a port profile for use on physical NICs, the network administrator must apply the capability uplink option to the profile. When this option is used, the port profile will be available only to the server administrator to apply to physical NICs within an VMware ESX server.

Uplink port profiles are applied to a physical NIC when a VMware ESX host is first added to the Cisco Nexus 1000V Series. The server administrator is presented with a dialog box in which the administrator can select the physical NICs to be associated with the VEM and the particular uplink port profiles to be associated with the physical NICs. In addition, the server administrator can apply uplink port profiles to interfaces that are added to the VEM after the host has been added to the switch.

System VLANs

System VLANs are defined by an optional parameter that can be added in a port profile. When used, this parameter causes the port profile to become a special system port profile that is included in the Cisco Nexus 1000V Series opaque data. Interfaces that use the system port profile and that are members of one of the system VLANs defined are automatically enabled, even if the VEM does not have communication with the VSM, upon VMware ESX bootup. This behavior enables the use of critical host functions if the VMware ESX host boots and cannot communicate with the VSM.

The control and packet VLANs must be defined as system VLANs. Definition of other VLANs as system VLANs, such as those used for vswif and vmknic interfaces, may also be useful. VLANs used for general virtual machine data should not be defined as system VLANs.

Cisco Nexus 1000V Series Network Design

This section discusses design considerations related to Cisco Nexus 1000V Series connectivity to the physical access layer.

Network Design Considerations

Multiple design considerations must be addressed when deploying the Cisco Nexus 1000V Series. At a basic level, the design principles used when connecting the Cisco Nexus 1000V Series to a physical access layer are similar to those used when connecting two physical switches together. Some design considerations are specific to the Cisco Nexus 1000V Series. More than likely, each VEM will be connected to two access layer switches. Dual access switch designs are the focus of this section.

Design Goals

A Cisco Nexus 1000V Series network design has two primary goals. First, the network should be designed for high availability. In practical terms, such a design limits single points of failure where possible. VMware ESX provides robust features to the server administrator to help ensure virtual machine availability. The Cisco Nexus 1000V Series is responsible for network availability. Availability is usually facilitated by redundant links between each VEM and the physical network, enabling the network to recover from link or physical switch failure.

The second design goal is to account for the uniqueness of VMware traffic patterns and the performance requirements of the virtual machines. Each VMware ESX host generates and receives several classes of traffic, each with unique characteristics. Network administrators must prioritize traffic to help ensure switch uptime, application performance, and proper VMware capabilities.

Traffic Classification

Classification of traffic types in any network is not easy to achieve. In a VMware environment, traffic varies based on the types of applications being virtualized. However, some traffic types can be identified and general prioritization applied. The general classifications of traffic for a typical VMware deployment are as follows:

- **Control traffic:** Control traffic is generated by the Cisco Nexus 1000V Series and exchanged between the primary and secondary VSMs as well as the VSMs and VEMs. It requires very little bandwidth (less than 10 KBps) but demands absolute priority. Control traffic is crucial to the Cisco Nexus 1000V Series' ability to function properly, and its importance cannot be overstated. Control traffic should be considered the most important traffic in a Cisco Nexus 1000V Series network.
- **Virtual machine data traffic:** Data traffic is a generalization of all traffic transmitted or received by virtual machines. In an VMware ESX host, this is the primary traffic type. For obvious reasons, data traffic requires high priority. The VSM management interface falls within this category.
- **VMware ESX management traffic:** VMware vCenter Server requires access to the VMware ESX management interface to monitor and configure the VMware ESX host. Management traffic usually has low bandwidth requirements, but it should be treated as high-priority traffic.
- **VMware VMotion traffic:** VMware VMotion traffic does not occur on a constant basis, meaning that most of the time VMware VMotion does not use any bandwidth. When VMware VMotion is initiated, it usually generates a burst of data over a period of 10 to 60 seconds. VMware VMotion is not bandwidth sensitive. When this type of traffic is faced with bandwidth that is lower than line rate, the duration of the virtual machine move event is extended based on the amount of bandwidth available. Despite its popularity as a feature, VMware VMotion traffic can usually be considered of medium priority relative to other traffic types.
- **Packet traffic:** Packet traffic is used to transport selected packets to the VSM for processing. The bandwidth required for packet interface is extremely low, and its use is very intermittent. If Cisco Discovery Protocol and IGMP features are turned off, there is no packet traffic at all. The importance of this interface is directly related

to the use of IGMP. If IGMP is not deployed, then this interface is used only for Cisco Discovery Protocol, which is not considered a critical switch function.

VLAN Consistency

Proper VLAN configuration on the physical infrastructure is important to helping ensure that the Cisco Nexus 1000V Series functions correctly. A VLAN defined on a physical switch has universal meaning; that is, every port on the switch configured in VLAN 10 is in the same VLAN; there is no concept of two discreet VLANs with ID 10 on the same switch. The same is true for the Cisco Nexus 1000V Series, but the switch architecture relies on proper physical switch configuration to help ensure this consistency.

Multiple VEMs require a physical Ethernet switch for inter-VEM connectivity. Each VEM needs consistent connectivity to all VLANs that are defined on the Cisco Nexus 1000V Series. Thus, any VLAN that is defined on the Cisco Nexus 1000V Series must also be defined on all upstream switches connected to each VEM.

Each VLAN should be trunked to each VEM using IEEE 802.1q trunking. Although not required, the uplink port profiles should be consistently applied to each VEM.

Traffic Separation

Traditional VMware network design calls for a minimum of three VLANs trunked to the VMware ESX host. These VLANs are used for virtual machine data, the VMware ESX service console, and VMkernel (VMware VMotion), with optional VLANs used for IP-based storage or additional virtual machine connectivity. The Cisco Nexus 1000V Series requires additional VLANs to support the control and packet interfaces.

The control and packet interfaces should be deployed on discreet VLANs, even though no configuration requirement or CLI enforcement prevents the two interfaces from sharing the same VLAN. Deployment on discreet VLANs is recommended for several reasons.

The control interface is much chattier and broadcast based than the packet interface. If the two interfaces were in the same VLAN, the packet interface would receive unnecessary broadcast traffic. Also, although the packet interface is important, the control interface is much more crucial to overall system function. Keeping the control interface isolated from any external influence helps ensure reliable system operation.

Multiple Cisco Nexus 1000V Series Switches can share common VLANs for their respective control and packet interfaces, but ideally each Cisco Nexus 1000V Series has separate VLANs. When the same control and packet VLANs are shared across multiple Cisco Nexus 1000V Series Switches, take care to help ensure the uniqueness of domain IDs.

Upstream Switch Connectivity

Much of the design work of a Cisco Nexus 1000V Series solution focuses on proper upstream switch connectivity. Connecting a Cisco Nexus 1000V Series Switch to a physical switch is not unlike connecting two physical switches to each other.

You can connect a Cisco Nexus 1000V Series Switch to a physical infrastructure in two ways: using standard uplinks and using PortChannels.

Individual Uplinks

A standard uplink is an uplink that is not a member of a PortChannel from the VEM to a physical switch. It provides no capability to load balance across multiple standard uplink links and no high availability characteristics. When a standard uplink fails, no secondary link exists to take over. Defining two standard uplinks to carry the same VLAN involves the risk of creating loops within the environment and is an unsupported configuration. Cisco NX-OS will issue warnings when such a condition occurs.

In some scenarios, single standard uplinks may be used, such as for a VMware ESX host that contains only a single NIC. Another possible scenario is a host connected to a secondary network that does not require high availability, such as a single NIC connected to a dedicated backup network. Given the requirements of most data center networks, however, standard uplinks should rarely, if ever, be used in a Cisco Nexus 1000V Series design.

PortChannels

The Cisco Nexus 1000V Series implements a PortChannel mechanism that supports two modes; standard PortChannels and virtual PortChannel Host Mode (vPC-HM). Regardless of the mode, PortChannels are managed using the standard PortChannel CLI construct, but each mode behaves differently.

A standard PortChannel on the Cisco Nexus 1000V Series behaves like an EtherChannel on other Cisco switches and supports the Link Aggregation Control Protocol (LACP). Standard PortChannels require that all uplinks in the PortChannel be in the same EtherChannel on the upstream switch.

Standard PortChannels can be spread across more than one physical switch if the physical switches are clustered. Examples of clustered switching technology include the Catalyst® 6500 Virtual Switching System 1440, virtual PortChannels on the Cisco Nexus 7000 Series Switches, and the Cisco Catalyst Blade Switch 3120 for HP. Clustered switches act as a single switch and therefore allow the creation of EtherChannels across them. This clustering is transparent to the Cisco Nexus 1000V Series.

Virtual Port Channel Host Mode

Most access layer switches do not support clustering technology, yet most Cisco Nexus 1000V Series designs require PortChannels to span multiple switches. To enable this spanning of switches, you can use a PortChannel in vPC-HM. vPC-HM divides the PortChannel into subgroups, with each subgroup representing one or more uplinks to one upstream physical switch.

Note: When vPC-HM is used, the Cisco Nexus 1000V Series supports only two subgroups: 0 and 1.

Links within the PortChannel that are connected to the same physical switch are bundled in the same subgroup automatically by using the Cisco Discovery Protocol packets received from the upstream switch. Alternatively, interfaces can be manually assigned a specific subgroup using interface-level configuration.

When vPC-HM is used, each vEth interface on the VEM is mapped to one of the two subgroups using a round-robin mechanism. All traffic from the vEth interface uses the assigned subgroup unless the assigned subgroup is unavailable, in which case the vEth interface will fail over to the remaining subgroup. When the originally assigned subgroup becomes available again, traffic will shift back to its original location. Traffic from each vEth interface is then hashed within its assigned subgroup based on the configured hashing algorithm.

vPC-HM should be considered the preferred method for most Cisco Nexus 1000V Series deployments.

Load Balancing

The Cisco Nexus 1000V Series provides 17 hashing algorithms to load-balance traffic across physical interfaces in a PortChannel. These algorithms can be divided into two distinct categories: source-based hashing and flow-based hashing.

Select the hashing algorithm used carefully as it affects the available configuration options and may require configuration changes on the access layer switches. The default hashing algorithm used by the Cisco Nexus 1000V Series is source MAC address hashing (a source-based hash).

Source-Based Hashing

Source-based hashing algorithms help ensure that a MAC address is transmitted down only a single link in the PortChannel, regardless of the number of links in a PortChannel. When you use source-based hashing, upstream switches are still required to use EtherChannel when connecting to a VEM.

With source-based hashing, a MAC address can move between interfaces in a PortChannel under the following conditions:

- The virtual machine moves to a new VMware ESX or ESXi host (VMware VMotion, VMware High Availability, etc.)
- A PortChannel link fails, causing recalculation of the hashing
- A link is added to the PortChannel, causing rehashing

The following Cisco Nexus 1000V Series algorithms can be classified as source-based hashes:

- Virtual port ID
- Source MAC address
- VLAN
- Source IP address

Flow-Based Hashing

Flow-based hashing enables traffic from a single MAC address to be distributed down multiple links in a PortChannel simultaneously. Selecting a flow-based hash can potentially increase the bandwidth available to a virtual machine and increase the utilization of the uplinks in a PortChannel by providing more granular load balancing.

Flow-based hashing algorithms are any algorithms that use the following to hash:

- Packet destination
- Layer 4 port
- Combinations of source address, destination address, and Layer 4 port

Control Interface Prioritization

The control interface is a critical link between the VSMs and VEMs. The control interfaces are the most important interfaces in any Cisco Nexus 1000V Series implementation. Special care needs to be taken to provide the control interfaces on both VSMs and each VEM with a minimum level of bandwidth. Expect each control interface to require a minimum of 10 KB of per-second bandwidth. Extended bursts of network traffic, such as VMware VMotion events, could starve the control interface.

Spanning Tree Protocol

Spanning Tree Protocol goes through a series of states on each interface as it tries to build the network tree. This process causes downtime on each interface when Spanning Tree Protocol needs to converge. This process is unnecessary for ports connected to the Cisco Nexus 1000V Series. By using the PortFast feature on a switch port, a Cisco switch can suppress the progression of Spanning Tree Protocol states and move straight to a forwarding state. PortFast is configured per interface and should be enabled on interfaces connected to a VEM

VSM Design

The VSM is a standard virtual machine that is deployed in the same way as other virtual machines. It does have specific design considerations that must be followed.

Virtual Machine Design

The Cisco Nexus 1000V Series VSM virtual machine has several basic considerations for deployment. Each VSM in an active-standby pair is required to run on a separate VMware ESX host. This requirement helps ensure high availability if one of the VMware ESX servers fails. You can also use the anti-affinity option in VMware ESX to help keep the VSMs on different servers. This option does not prevent the VSMs from ending up on the same server; anti-

affinity prevents VMware Distributed Resource Scheduler (DRS) from moving the virtual machines to new machines. If the VSMs end up on the same host due to VMware High Availability, VMware DRS will issue a five-star recommendation to move one of the VSMs.

CPU and memory for the VSM virtual machine need to be guaranteed: that is, the 2 GB of memory required by each virtual machine should not be shared with other virtual machines. In addition, a minimum 1-GHz CPU capability should be assured for each VSM virtual machine.

The mgmt0 interface on the VSM does not necessarily require its own VLAN. In fact, you could simply use the same VLAN to which VMware vCenter Server belongs. The VSM management VLAN is really no different than any other virtual machine data VLAN. Alternatively, network administrators can have a special VLAN designated for network device management.

Adjacency

The VSM and VEM control and packet interfaces are Layer 2 interfaces. They do not support IP-based communication. Layer 2 adjacency from the VSMs to each VEM is required. Also, Layer 2 adjacency must be maintained between VSMs that are configured as a high-availability pair.

Latency

The control protocol used by the VSM to communicate with the VEMs is similar to those used in Cisco module chassis such as the Cisco MDS 9000 Family and the Cisco Nexus 7000 Series chassis. This protocol was designed to operate in a tightly controlled, lossless, low-latency Layer 2 network with no possibility of network contention (for example, the EoBC in a Cisco chassis). In a Cisco Nexus 1000V Series implementation, this protocol runs over the data center network.

The control protocol has been modified for the Cisco Nexus 1000V Series to take into account various performance characteristics of a data center network, but there are design limitations. The Cisco Nexus 1000V Series was designed to run in a single data center. The Cisco Nexus 1000V Series does not support long inter-data center distances between the VSM and a VEM.

As a general guideline, round-trip latency between the VSM and the VEM should be less than 50 milliseconds.

Traditional Cisco Network

This section describes various scenarios regarding the deployment of the Cisco Nexus 1000V Series in a traditional access layer design. "Traditional" in this context refers to a VMware ESX host with multiple NICs connected to two independent access layer switches.

Two-NIC Design Examples

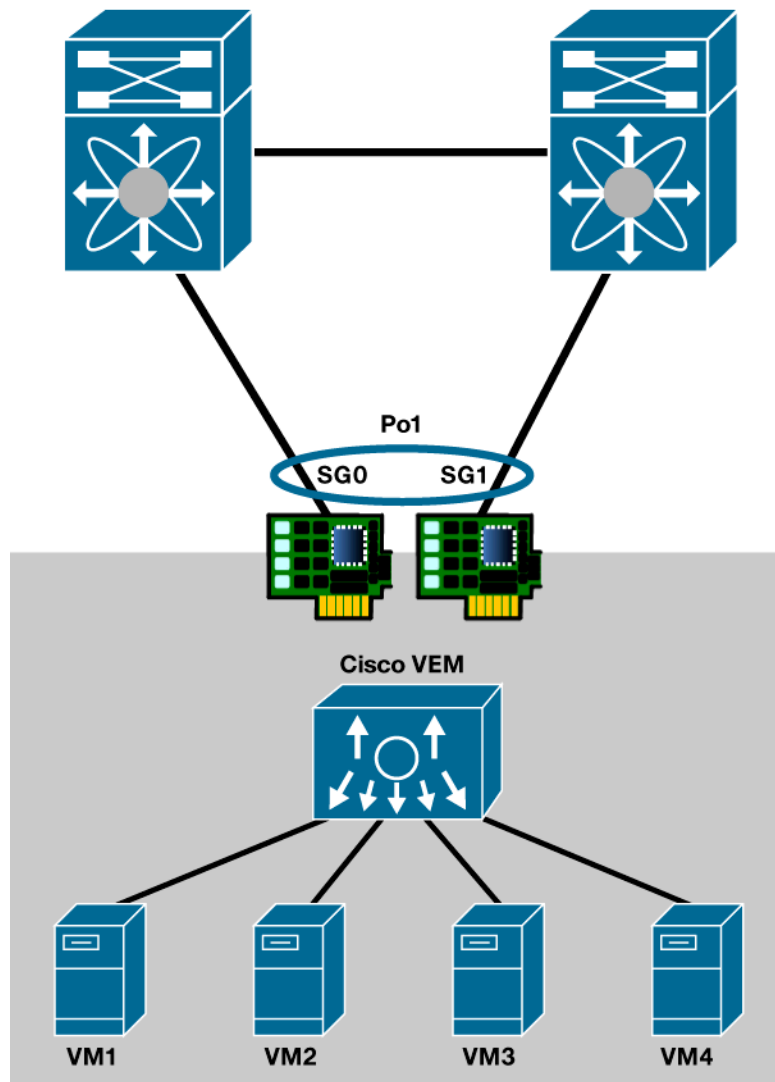
Hosts with two NICs are fairly common when deploying VMware ESX, particularly for blades and 10-Gbps connected hosts. This design is also the simplest from the perspective of the Cisco Nexus 1000V Series mainly because there is little possibility for configuration variation.

With this design, both NICs are part of a single PortChannel configured in vPC-HM connected to two access switches (Figure 4). A single uplink profile is applied to both NICs. Load balancing varies, but it requires a source-based hashing algorithm. VLAN hashing and source MAC address hashing are described, with source MAC address hashing being the preferred method.

VLAN hashing in this design may lead to undesirable load balancing on 1-Gbps NICs. Using two 1-Gbps NICs creates the potential for the data VLAN and the VMware VMotion VLAN to be hashed down the same uplink port. If VMware VMotion is initiated, all virtual machines will be contending for the same bandwidth as the VMware VMotion session. With 10-Gbps NICs, the impact may be negligible because of the substantial bandwidth available.

A more desirable configuration, particularly for 1-Gbps hosts, spreads the load of the virtual machines across both uplinks by using source MAC address hashing. When VMware VMotion is initiated, it will be contending for bandwidth among a fewer number of virtual machines.

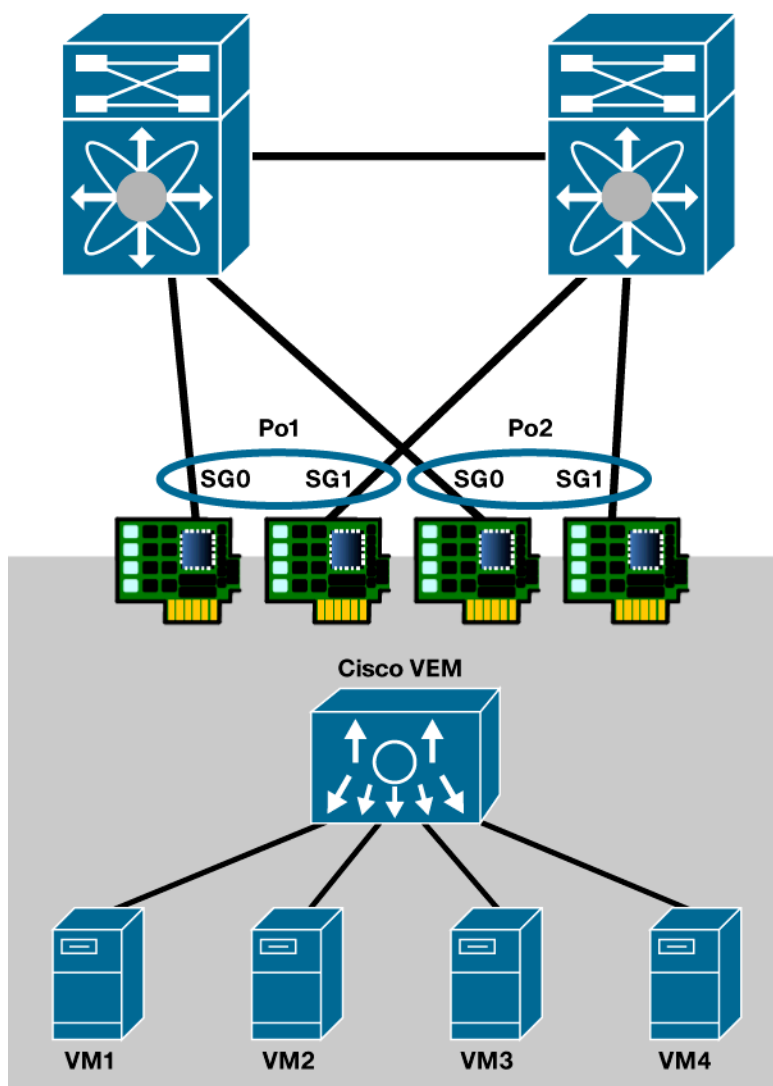
Figure 4. Two NICs with a Single PortChannel Using vPC-HM



Four-NIC Design Examples

A four-NIC design offers additional flexibility by allowing the segregation of data traffic from all other traffic types. This design uses two PortChannels in vPC-HM. Each PortChannel consists of one link to each access switch. Source-based hashing is still required. Figure 5 shows an example of this configuration: two PortChannels using vPC-HM, each with a link to both access switches. Each vPC-HM subgroup consists of a single link. There is no need to run EtherChannel on the access switches.

Figure 5. Four NICs with Two PortChannels Using vPC-HM



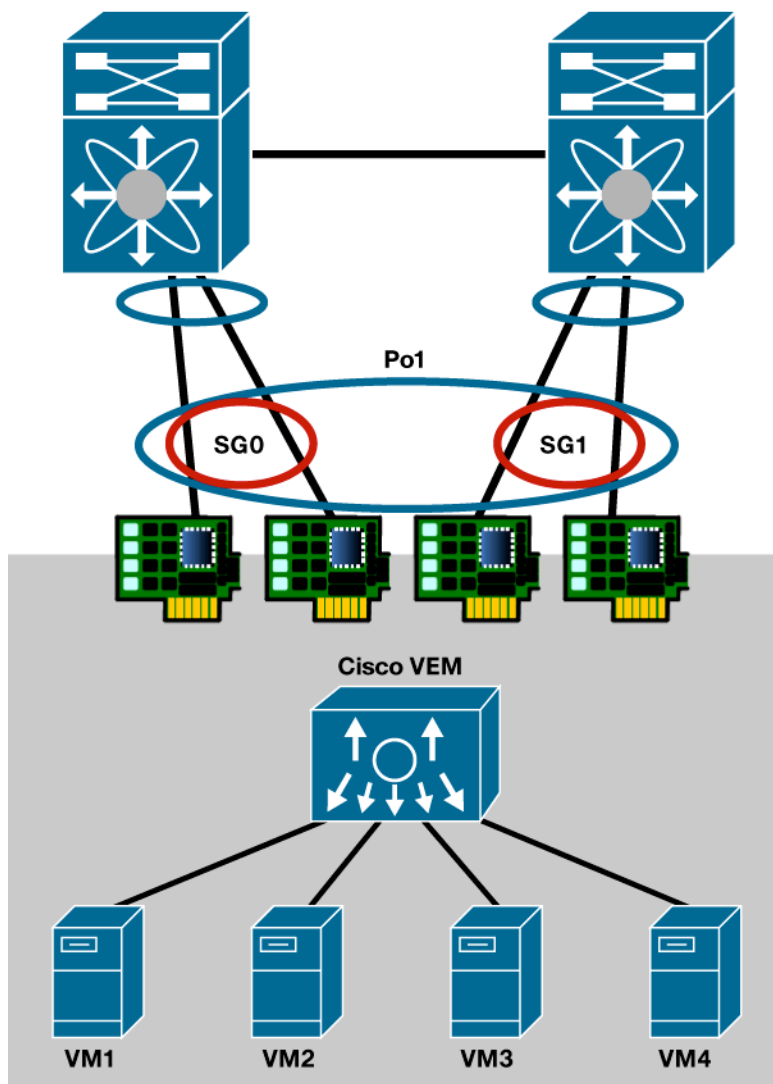
The main benefit of this design is that VMware VMotion does not affect the performance of virtual machine data traffic. PortChannel 1 carries all virtual machine data, and PortChannel 2 carries VMware VMotion, the service console, and control and packet VLANs.

Single-PortChannel Alternative

Alternatively, you can design a four-NIC solution without the use of multiple PortChannels. A single vPC-HM-based PortChannel could be used to spread the virtual machine's load across as many NICs as possible.

The advantage of this design is the availability of all four NICs for virtual machine data traffic and the capability to support flow-based hashing. The disadvantage, of course, is that the VMware VMotion traffic is again mingled with data traffic and could contend for bandwidth. Figure 6 shows a design with four NICs in a single PortChannel using vPC-HM. If flow-based hashing is used, each upstream switch will need be configured with an EtherChannel (as shown). If source-based hashing is used, the upstream switches are not required to run EtherChannel.

Figure 6. Four NICs with One PortChannel Using vPC-HM

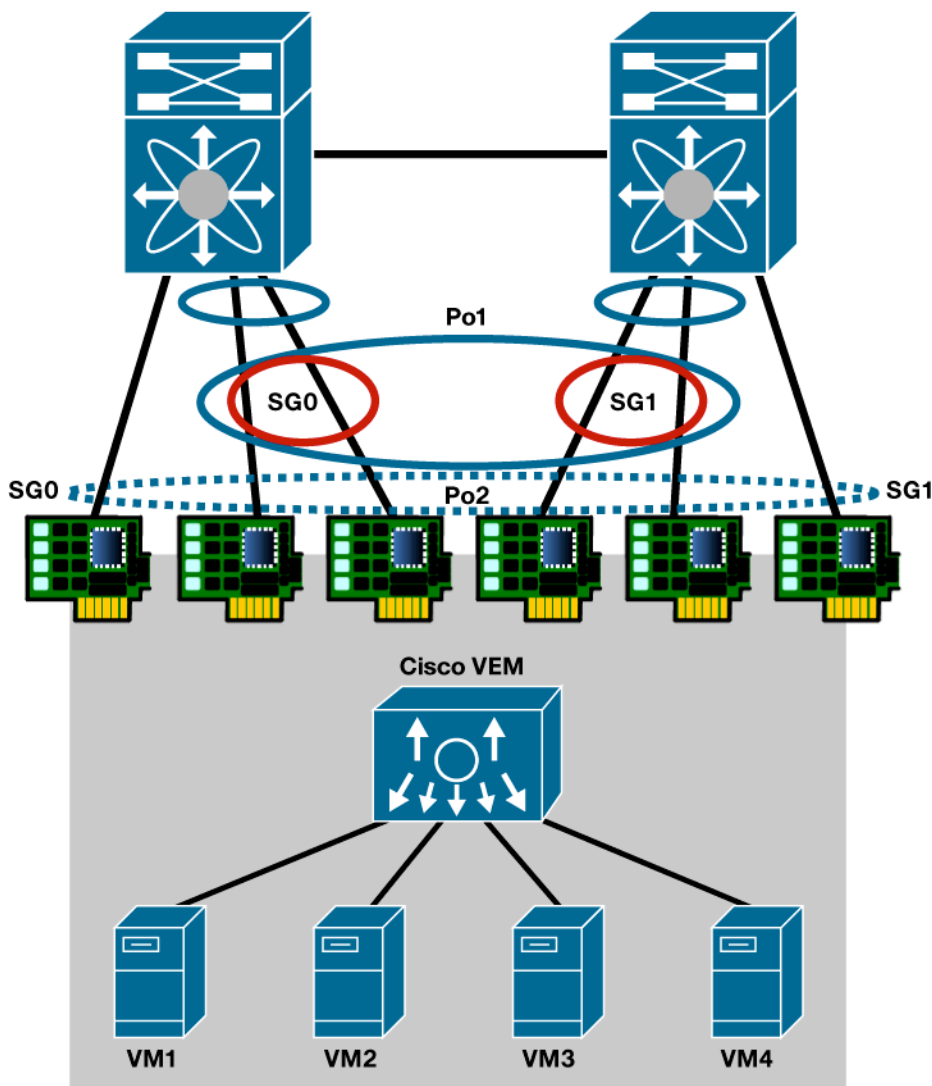


Six-NIC Design Examples

Six or more NICs provide the flexibility to use flow-based hashing and still maintain isolation between data traffic and other traffic types. This design uses two independent PortChannels in vPC-HM, much like the four-NIC design. The difference here is that the first PortChannel consists of four NICs that are dedicated to virtual machine data traffic. The second PortChannel consists of the remaining two NICs, which are used for VMware VMotion traffic, the service console, and control and packet VLANs.

This configuration could use either source- or flow-based hashing. Figure 7 shows a six-NIC example with flow-based hashing. PortChannel 1 is in vPC-HM and consists of two NICs to each access switch. The access switches are configured with EtherChannel. Each subgroup in PortChannel 1 consists of two NICs. PortChannel 2 is also in vPC-HM and consists of the two NICs dedicated for VMware VMotion as well as the service console and control and packet VLANs.

Figure 7. Six NICs with Two PortChannels Using vPC-HM



As in the four-NIC design, you could use a single PortChannel in vPC-HM with all six NICs participating. Implementing this design with six NICs may be less than ideal. If flow-based hashing is selected, each subgroup and associated EtherChannel on its respective access switch would contain an uneven number of links to hash between, which can lead to suboptimal load distribution.

For More Information

For more information about the Cisco Nexus 1000V Series, please refer to the following URLs:

- Cisco Nexus 1000V Product Information: <http://www.cisco.com/go/1000v>
- Cisco Nexus 1000V Technical Documentation: <http://www.cisco.com/go/1000vdocs>
- Cisco Nexus 1000V Community: <http://www.cisco.com/go/1000vcommunity>



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flip Video, Flip Video (Design), Flipshare (Design), Flip Ultra, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Store, and Flip Gift Card are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0907R)