Data Center Service Patterns

March 20, 2009

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Introduction

Audience

This document is intended for engineers who are interested in implementing or want to understand the implementation of Cisco network and server technologies in the data center.

Document Objectives

This document provides guidance to engineers interested in deploying Cisco data center network, security, and application services. The design options and implementation details provided are intended to be a reference for an enterprise data center.

Overview

This document describes the use of Cisco technology to provide a robust data center environment for enterprise serverfarms. The document reviews and references established best practices while offering new methods for improving the availability, scalability, and security of the server and its applications. The design referenced in this document heavily uses both server and network virtualization technologies to achieve the aforementioned goals. In addition, the data center benefits from the well-known by-products of virtualization namely—improved infrastructure utilization and overall operational efficiency gains.

Solution Topology

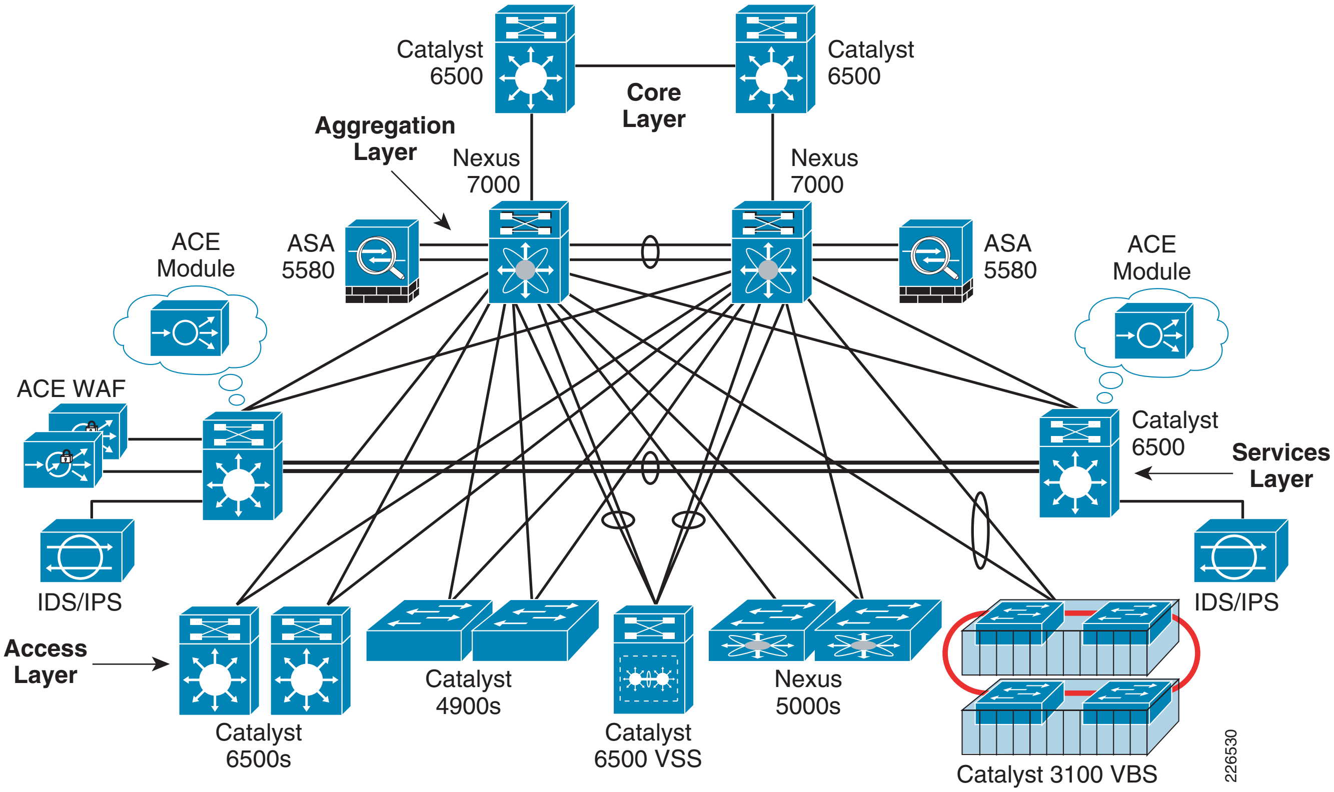
The enterprise data center infrastructure encompasses a wide range of products and technologies addressing environmental conditions, networking and storage needs, as well as application requirements. The solution documented in this document concentrates on the IP networking infrastructure and the functionality it affords to the various application environments within the data center, including the following:

- High Availability
- Scalability
- Session Persistence
- Security

Figure 1 depicts the tested network solution. At a high-level, the following four functional areas are addressed in this design by the physical devices:

- Core layer
- Aggregation layer
- Services layer
- Access layer

Figure 1 Implemented Solution Topology



Note All of the connections shown in Figure 1 between platforms are 10-Gigabit Ethernet.

- **Core Layer**—The core of the data center is a high-speed Layer 3 fabric. In Figure 1, the core of the data center consists of the Cisco Catalyst 6500s devices. From a physical standpoint, the core shown appears to be highly traditional in its deployment model; however, the use of Virtual Device Contexts (VDC) allows engineers to logically partition or virtualize the Nexus 7000 connection into the traditional Catalyst 6500 core devices. For more information, refer to the routing design section of the *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services* at the following URL:
http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html.
- **Aggregation Layer**—The devices in this layer of the solution are Nexus 7000 Series switches. From a physical perspective, the Nexus 7000 provides more than enough slot and port density to support the surrounding core, services, and the access layer devices within the topology. In addition, the Nexus devices offer rich set of Layer 2, Layer 3, and virtualization features permitting a new level of segmentation and control within a single aggregation device in the data center. These features are discussed throughout this document and build upon previous design recommendations (see the references below) documented for the Nexus 7000 platform.
- **Services Layer**—This layer consists of Catalyst 6500 series switches using service modules and dedicated appliance platforms. As shown in Figure 1, the appliance services may attach directly to the Nexus 7000 aggregation layer or use the port density available on the services chassis themselves. The services layer design used for this solution is based upon previous efforts surrounding services chassis design but expands upon this foundation of load balancing and

firewalling to include intrusion detection, intrusion prevention, and web-application firewall services. In addition, techniques to monitor data center traffic, including virtual machines, are introduced.

For more information on the integration of services with a Nexus 7000, refer to the *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services* at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html

For more information on the integration of dedicated services switches, refer to the *Data Center Service Integration: Service Chassis Design Guide* at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/dc_servchas/service-chassis_design.html

- **Access Layer**—This layer is the point of connectivity for data center endpoints that provides entry to the greater network. Traditionally, this is a Layer-2 domain requiring network administrators to concern themselves with loop mitigation, oversubscription requirements, and physical port availability for server connectivity. This description highlights the traditional data center access layer switching environment. However, as Figure 1 illustrates, there are many solutions available to the network administrator to address and eliminate these issues including Virtual Switching System (VSS) 1440 and Virtual Blade Switch (VBS) technologies. Each of these technologies simplifies the network by creating a single virtual instance of a switch, simultaneously removing the complexities of spanning tree and allowing server endpoints to achieve greater performance levels through NIC teaming.

The solution shown in Figure 1 also introduces Nexus technology at the access layer with the Nexus 5000 platform. Note that this platform provides dense 10-Gigabit and 1-Gigabit Ethernet connectivity through the Nexus 2000. The Nexus 5000 also supports Fibre Channel over Ethernet (FCoE), consolidating Fibre Channel and Ethernet traffic over the same server adapter. The Nexus 5000 allows for a new era of access layer consolidation while providing ample uplink capacity.

Note The presence of 10-Gigabit Ethernet attached servers in the enterprise data center should compel network administrators to understand the current and future traffic flows within their data center. To ensure the quality-of-service (QoS) that each application environment requires within these consolidated serverfarms, network administrators will be looking to technologies such as QoS, NetFlow, and SPAN to assist.

The Nexus 1000v platform, not shown in Figure 1, is also addressed in this document. The Nexus 1000v is a Virtual Distributed Switch (VDS) that may be incorporated into VMware's vSphere 4.0 infrastructure on the ESX/ESXi 4.0 hypervisor. The Nexus 1000v spans multiple ESX/ESXi hosts simplifying network administration and provisioning. Simply put, the physical lines between the network and the server have blurred; the network has expanded onto the server platform. The Nexus 1000v's NX-OS allows network administrators to centrally manage the networking interface of VMware's data center virtualization platform using a set of features and functionality available only on the Nexus 1000v.

Benefits of the Solution

The Virtual Data Center with Virtual Transparent Services Mode solution focuses on the integration of services within a virtualized data center environment. The primary goal is to reliably and transparently apply network services in a data center environment to create a more flexible, functional, and secure serverfarm.

Required Hardware/Software

Table 1 lists the hardware, software, and other key features required to implement the solution.

Table 1 Solution Hardware and Solution Software Components

	Platforms, Line Cards, End Points within Role	Releases
Core Router/Switch	Catalyst 6500 Series	12.2(33)SXH2
	WS-X6724-SFP	
	WS-6704-10GE	
	VS-S720-10G	
Aggregation Router /Switch	Nexus 7000 Series	4.1(2)
	N7K-M132XP-12	
	N&K-M148GT-11	
	N7K-SUP1	
Services Layer Switch	Catalyst 6500 Series	12.2(33)SXI
	VS-S720-10G	
	WS-6708-10GE	
	WS-6748-GE-TX	
	ACE20-MOD-K9	A2(1.3)
	WS-SVC-NAM-2	4.0(1)
	ASA5580-40	8.1(2)
	IPS4270-20-k9	6.2(1)E3
	ACE-XML-K9	6.0.2-2008-09-15T22
Access Layer Switches	Catalyst 6500 Series	12.2(33)SXI
	WS-6708-10GE	
	WS-6704-10GE	4.0(1)
	WS-SVC-NAM-2	
	VS-S720-10G	
	WS-X6748-GE-TX	
	WS-C4900M	12.2(46)SG
	WS-X4904-10GE	
	WS-X4920-GB-RJ45	
	N5k-C5020P-BF	4.0(1a)N1(1)
	N5K-M1404	
	WS-CBS3120G-S	12.2(40)EX1
	WS-CBS3120X-S	12.2(40)EX1
	Nexus 1000V	4.0(1a)S1(0.14

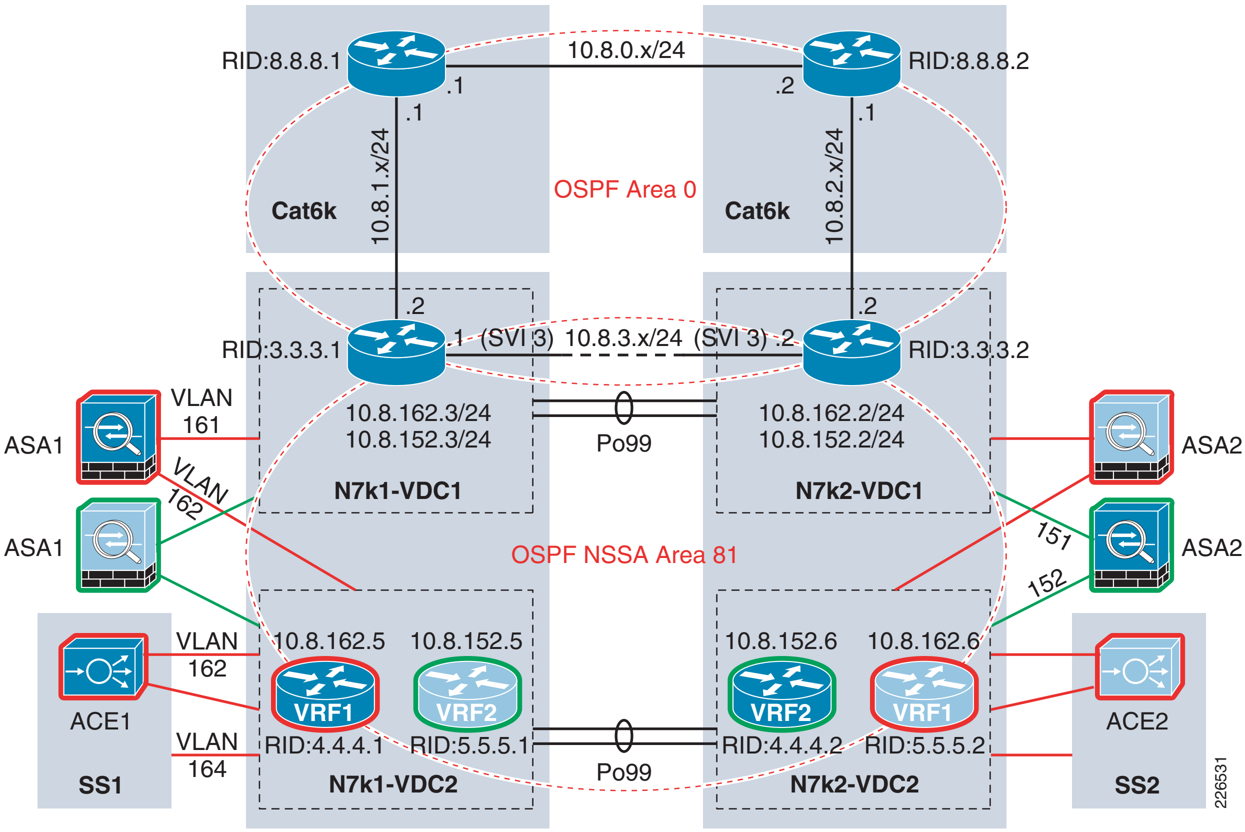
Table 1 Solution Hardware and Solution Software Components (continued)

Cisco NetFlow Collector	RHEL 4u2	6.0.0 (Build 31)
Server Environments	HP DL380 G5 HP DL380 G4 Windows Server 2003	VMware 3.5u2 and 4.0(RC) Windows Server 2003 Oracle Linux 5.2 (Carthage)

Design Details

Figure 2 depicts the fundamental logical topology of the validated solution. The solution may be best described as an active-active model with transparent services. The transparent services are applied between Layer 3 devices defined on the Nexus 7000 aggregation layer platforms. This contained Layer-2 service domain is both a flexible and a scalable solution. The red and green highlighted devices indicate the existence of two active data paths (i.e., service patterns in this aggregation block of the data center). These independent service patterns are achieved through virtualization of network services from Layer 2 and above.

Figure 2 Logical Solution Topology



The following is a brief description of the primary components shown in Figure 2:

- OSPF Area 0 is defined between the core Catalyst 6500s and VDC1 on the aggregation layer Nexus 7000s. The Layer-2 port channel 99 (po99) supports all VLANs that exist between the VDC1 instances.
- Not-So-Stubby-Area 81 (NSSA Area 81) exists between VDC1 and virtual routing/forwarding (VRF) instances on VDC2. Note the use of alternating primary and standby VRFs to create an active-active Layer 3 topology. The NSSA Area 81 also introduces a service layer consisting of active-active ASA and ACE virtual contexts attached directly to the Nexus aggregation switches or through dedicated services chassis. The NSSA Area 81 will be expanded to include intrusion detection and prevention, network analysis , and web-application firewall services that are transparently positioned between the Layer 3 devices of the NSSA Area 81. These services are described "Services Layer" section on page 37.
- Active-active ASA 5580 virtual contexts in transparent mode allows neighboring between VDC1 instances and VDC2 VRF devices to secure and optimize utilization of active-active data center resources. The use of the ASA virtual contexts between Nexus 7000 VDCs essentially creates a DMZ within the aggregation layer of the data center.
- Active-Active ACE Service Module virtual contexts in transparent mode allows for neighboring between VDC1 and VDC2 VRF devices to optimize utilization of data center resources. In addition, the ACE virtual contexts provide application layer services such as load balancing, SSL offload, and session persistence. Note that these services are not shown applied to the active green data center path, implying that transparent virtual services may be selectively applied only to those serverfarm resources, (i.e., applications that require them).
- The **vrf1** and **vrf2** positioned in VDC2 of the Nexus 7000 aggregation block are the default gateways for their respective serverfarms. The default gateway is a HSRP group instance that is prioritized and aligned with spanning tree root definition of the associated VLAN. Figure 2 shows this logical alignment of network services including application and security, default gateway, and spanning tree root through the red and green paths. In this example, the red data center pattern is active on the left side of the infrastructure while the green uses the right side of the aggregation block.

Note The use of multiple VRFs on the "southern" VDC, or in this example VDC2, may be advantageous to create an isolated Layer 3 topology for server-to-server traffic patterns or network monitoring traffic. An example of this implementation can be found in the "ERSPAN" section on page 24.

Additional Network Services

The solution topology in Figure 2 is expandable, allowing for new network services to be easily introduced. The following network services were also validated in this solution:

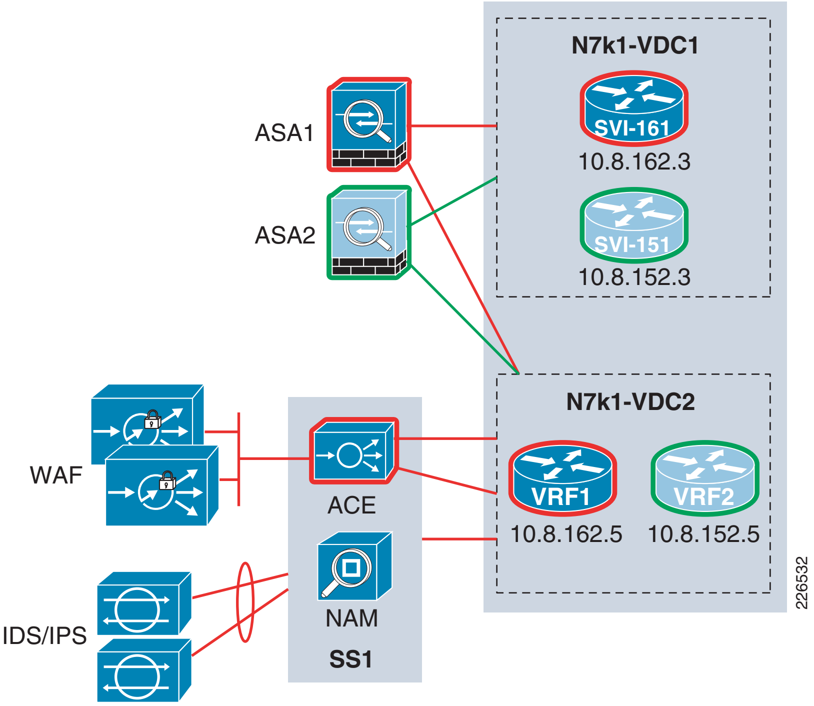
- Web Application Firewall Services
- Intrusion Prevention/Detection Services
- Network Analysis

Web Application Firewall Services

The Cisco Web Application Firewall (WAF) is a proxy security service providing services for web and XML-based traffic. WAF appliances are managed as a cluster, allowing for centralized configuration and deployment of security policies. Figure 3 illustrates the flexibility of the validated solution topology as web-application services are readily introduced to the environment. Note that Figure 3 only shows half of the deployment within the data center aggregation block; this is in reality a symmetric solution with predictable traffic patterns.

As shown in Figure 3, the WAF cluster deployment uses the ports available on the Catalyst 6500 service switch and the load-balancing services of the ACE module in the same services chassis. This one-arm deployment model allows for physical redundancy, scalability, and session persistence as the transparent ACE virtual context treats the WAF cluster as a *service* serverfarm. In addition, the ACE module may provide SSL-termination allowing the WAF cluster to process non-encrypted data. This is a significant service, because obfuscation is a well-known method used by hackers to circumvent data center security.

Figure 3 Extended Services Topology



For more information about the Cisco WAF, refer to the following URL:

http://www.cisco.com/en/US/products/ps9586/index.html

Intrusion Prevention/Detection Services

Intrusion prevention and detection is a necessity within the data center. This active-active solution introduces each security service on the same IPS 4270 platforms through virtualization. As shown in Figure 3, the services chassis provides the port capacity to attach a number of physical sensor devices. The Cisco IPS/IDS solution allows one to partition each platform and create independent IPS/IDS virtual sensors on a single physical platform.

In the validated solution, IPS services were employed inline using an EtherChannel between the ACE transparent virtual context and its associated VRF on the Nexus 7000 VDC2 instance. The EtherChannel allowed for IPS redundancy and load sharing. The extended services layer between the two OSPF neighbors on the Nexus 7000 VDC routing instances included a transparent ASA virtual context, a transparent ACE virtual context, and an inline virtual sensor. As noted earlier, the ACE can terminate SSL connections allowing the IPS device to inspect non-encrypted traffic.

The IDS virtual sensors are mirrored traffic destinations. In the validated design, the Nexus 1000v platform is the primary source of ERSPAN traffic. This allowed for inspection of all traffic ingress and egress from each virtual machine in the test bed. For more details, refer to the Network Analysis section below.

Note The IDS/IPS devices are use both Gigabit and 10-Gigabit Ethernet connections.

For more data center specific security recommendations, refer to the *Security and Virtualization in the Data Center* at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_sec_design.html

Network Analysis

The use of network analysis devices is another service readily available in this design. The following methods of data collection were used in the data center network:

- NetFlow
- Encapsulated Remote Switched Port Analyzer (ERSPAN)

NetFlow

NetFlow was developed by Cisco to provide better insight into the IP traffic on the network. NetFlow defines flows as records and exports these records to collection devices. NetFlow provides information about the applications in and utilization of the data center network. The NetFlow collector aggregates and assists network administrators and application owners to interpret the performance of the data center environment.

Encapsulated Remote Switched Port Analyzer (ERSPAN)

ERSPAN allows network administrators to remotely monitor devices that are attached to the network. ERSPAN uses GRE tunnels to route remote SPAN traffic to a network-analysis device. In the validation of this solution, the final destination for ERSPAN traffic were the IDS and Network Analysis Module (NAM) resident in or attached to the services switches as shown in Figure 3.

Additional Access Layer Details

The Nexus 7000 aggregation block supports numerous access layer devices. This flexibility and inherit scale of the Nexus 7000 platform, makes it an ideal fit for aggregation layer service. The tested solution employed a Layer-2 access design that used different platforms. This section discusses the connectivity of these varied access layer devices, including the following:

- Traditional Model
- Virtual Blade Switching (VBS)
- Virtual Switching System (VSS)
- Nexus 1000v

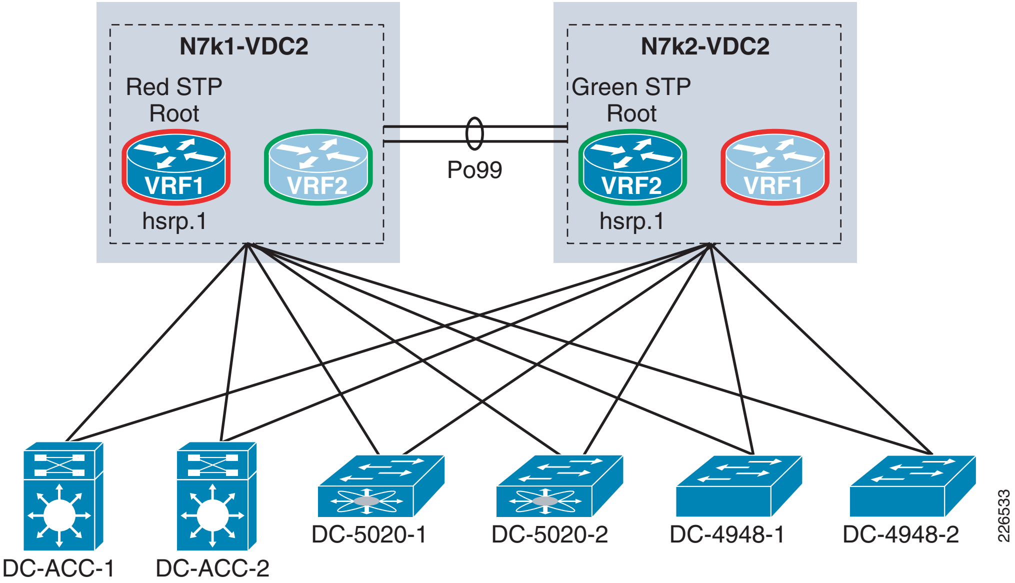
Traditional Model (including Nexus 5000, Catalyst 4900M, Catalyst 6500)

The Nexus 7000 aggregation layer supports numerous Layer 2 access devices. The traditional "triangle" deployment model was used for testing across the following platforms:

- Catalyst 6500 using native IOS
- Nexus 5000
- Catalyst 4900M

Figure 4 illustrates the connections of these devices to the each of the Nexus 7000 aggregation switches. The access layer switches in this example are connected to physical ports allocated to the second VDC on each Nexus 7000 platform. The Layer 2 domains of the solution uses Rapid PVST+ as the redundancy protocol. Under normal operating conditions, the configuration dictated that the uplinks from each of these access layer switches would alternate between blocking or forwarding state for the red and green VLANs as determined by the spanning tree root definition on the VDC.

Figure 4 Traditional Access Layer Topology



To assist the spanning tree implementation, bridge assurance was enabled between the Nexus 7000 aggregation switches and the Catalyst 6500 and Nexus 5000 switches. Bridge assurance enables BPDU transmission on all ports defined as *network* and verifies that the port is truly operational, meaning unidirectional links and switch software failures will be detected. Enabling bridge assurance requires that the connection be point-to-point and that each device has bridge assurance enabled on the interesting ports.

The default gateway for all servers using this access layer is the HSRP address associated with the red and green VRFs. Note that the HSRP priority is aligned with the spanning tree root configuration to create predictable traffic patterns within the data center. Implementation details are provided in the "Access Layer" section on page 52.

Note This solution effort did not focus on some of the advanced features of the Nexus 5000 and 7000 platforms. For more information about the Layer 2 implementation of Nexus 5000 and 7000 Series switches in the data center , refer to *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services* at the following URL:
http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html

During this testing effort, the Nexus 5000 switches supported a VMware vSphere infrastructure using Cisco's Nexus 1000v technology. Each of the ESX servers were dual-homed to the Nexus 5000 switches through a 10-Gigabit dual-port converged network adapter (CNA). This combination of Nexus 1000v, Nexus 5000, and the vSphere infrastructure is discussed in more detail in the "Nexus 1000v" section on page 16.

Virtual Blade Switching (VBS)

The Cisco Catalyst 3100 Blade Switches (CBS) are currently available for Dell, IBM, and HP-blade chassis models. The integrated blade switches provide Ethernet connectivity for the blade server NICs. Traditionally, this has meant a blade chassis would support 2 or 4 independent Ethernet switching modules. This model is effective; however, as blade chassis environments grow, additional strain is placed on the scalability of the Layer 2 environment and the management load of network operation teams. The Cisco 3100 Series blade switches addresses these issues through virtualization.

Figure 5 illustrates the logical topology of the CBS VBS access layer used during testing. The diagram shows the two Nexus 7000 aggregation switches connected to the VBS-enabled blade chassis access layer. The VBS uses two EtherChannels as uplinks to the aggregation layer.

Figure 5 Cisco Network Assistant Logical Topology View

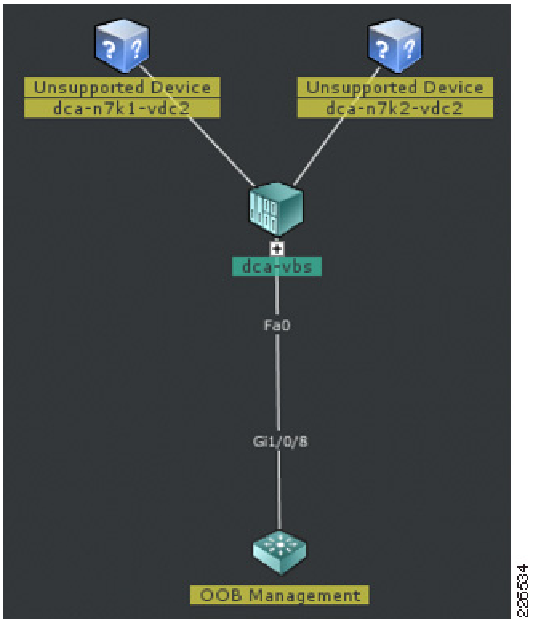
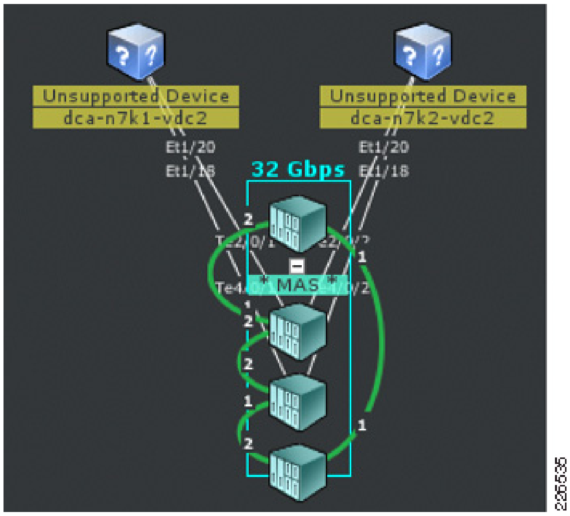


Figure 6 is the physical view of the VBS solution. Note that VBS is actually comprised of four independent Catalyst 3100 switches that are physically connected, by four stack cables. The stack cables support up to 48 Gbps. The four physical switches are deployed within two HP C-class blade chassis. The four 10-Gigabit uplinks reside on two of the four switches but are shared by all servers using the cluster. Redundancy at the uplink and switching platform is achieved using VBS.

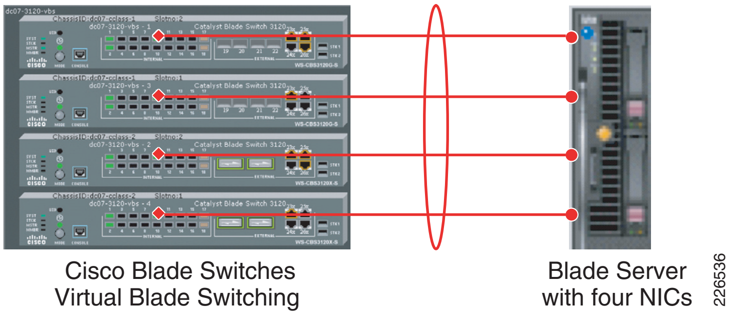
Figure 6 Cisco Network Assistant Topology Physical View



Note The Cisco Network Assistant (CNA) Version 5.4 does not manage the Nexus line of products and hence the **Unsupported Device** annotation in Figure 6.

A primary advantage to VBS beyond simplifying management and Layer 2 redundancy is the ability to fully use the server NICs present on the blades. VBS presents a single logical switching instance to the blade servers it supports. This allows server administrators to realize new levels of blade server productivity by enabling an active-active NIC teaming configuration. For example, Figure 7 shows how a single blade server can create a 4-Gigabit Ethernet EtherChannel using four NICs across four Catalyst 3100 blade switches using VBS in a single HP C-class chassis.

Figure 7 Example of Blade Server using EtherChannel NIC Teaming across a Virtual Blade Switch



Note The introduction of virtual port channeling (vPC) on the Nexus 7000 platform combined with VBS provides another opportunity to simplify the data center access layer.

For more information about Cisco Virtual Blade Switches, refer to the following URL:

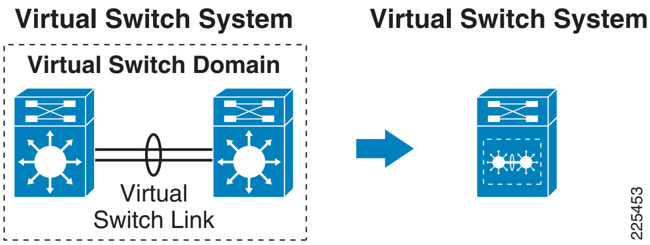
http://www.cisco.com/en/US/products/ps6748/prod_white_papers_list.html

Virtual Switching System (VSS)

The Virtual Switching System (VSS) technology allows the grouping of two Cisco Catalyst 6500 switches into a single virtual switch. A VSS provides physical infrastructure redundancy while simultaneously simplifying the logical topology of the data center.

Figure 8 illustrates the concept of VSS. The left side of Figure 8 represents the physical layout of the VSS; two Catalyst 6500s are physically connected through a Virtual Switch Link (VSL). The two switches are members of a Virtual Switch Domain (VSD) and as the right side of the diagram shows this design forms a single logical switch with a single control plane, a VSS.

Figure 8 Virtual Switching System Physical and Logical View



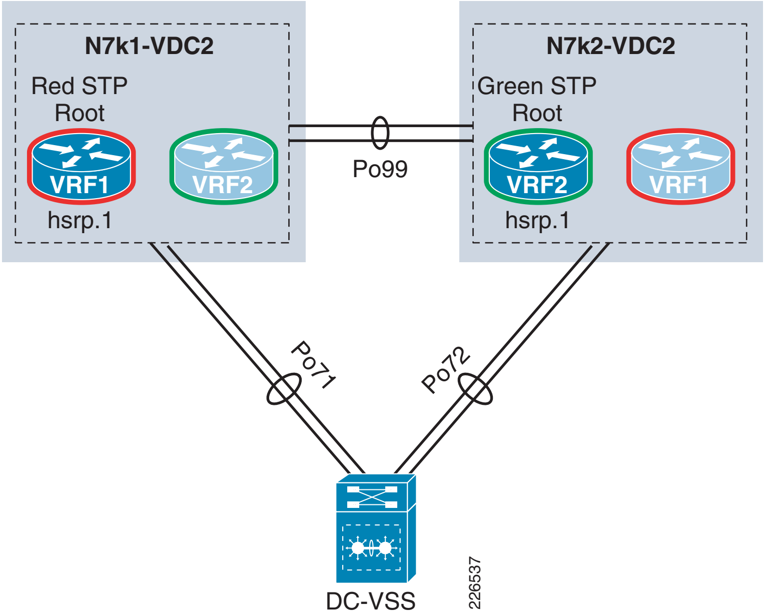
The primary benefits of this logical grouping include:

- Increased operational efficiency by simplifying the network via virtualization
- Increased availability and forwarding performance via Inter-chassis Stateful Switchover (SSO) and nonstop forwarding (NSF)
- Increased availability and forwarding performance through Multichassis EtherChannel (MEC). This performance benefit extends to the adjacent server platforms when VSS operates as an access layer switch.

Figure 9 represents the VSS deployment used in this active-active solution. The VSS is dual-homed through two MECs to the VDCs defined on the Nexus 7000 aggregation switches. It is important to remember that multichassis EtherChannel implies that each physical switch has a member port of each EtherChannel connected to the Nexus 7000 chassis. This configuration allows the access layer to benefit from physical switch and EtherChannel-based redundancy; however, the greatest advantage to using VSS in the access layer may be to the server itself as advantages inherent in today's NIC teaming capabilities may easily be used when connecting to a single logical switching entity.

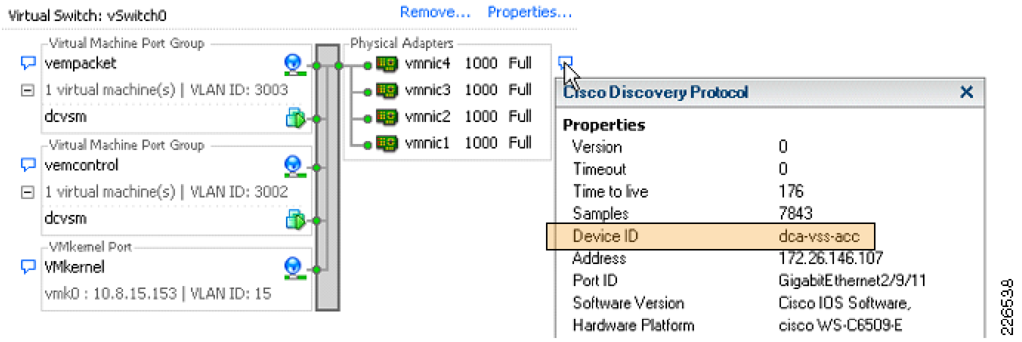
Note The introduction of vPC on the Nexus 7000 platform combined with VSS provides another opportunity to simply the data center Layer 2 design. For more details regarding the use of vPC in the data center, refer to the *Data Center Design—IP Network Infrastructure* at the following URL*:* http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/DC-3_0_IPInfra.html

Figure 9 Logical Topology of VSS and VBS Solutions



VSS allows the server to fully use its NIC teaming features. VSS supports active-active uplinks from the server providing redundancy and high availability based upon EtherChanneling. For example in this solution environment, RHEL 5.2 bonding and ESX 3.5 static EtherChannels used the VSS access layer. Figure 10 shows the virtual switch configuration on an ESX server indicating that all the physical adapters are homed to a single VSS access layer switch; in this case, VSS access layer switch **dca-vss-acc**. This single ESX host is able to use four vMNIC Gigabit connections by creating a static EtherChannel to the VSS.

Figure 10 Virtual Center view of ESX 3.5 Static EtherChannel to VSS



Virtual Machine Port Group
vempacket
1 virtual machine(s) | VLAN ID: 3003
dcvsm

Virtual Machine Port Group
vemcontrol
1 virtual machine(s) | VLAN ID: 3002
dcvsm

VMkernel Port
VMkernel
vmk0 : 10.8.15.153 | VLAN ID: 15

Physical Adapters		
vmnic4	1000	Full
vmnic3	1000	Full
vmnic2	1000	Full
vmnic1	1000	Full

Properties	
Version	0
Timeout	0
Time to live	176
Samples	7843
Device ID	dca-vss-acc
Address	172.26.146.107
Port ID	GigabitEthernet2/9/11
Software Version	Cisco IOS Software,
Hardware Platform	cisco WS-C6509-E

For further details regarding VSS deployment in the aggregation and access layer of the data center , refer to the *Integrating the Virtual Switching System in Cisco Data Center Infrastructure* document at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/vssdc_integrate.html

Nexus 1000v

The Cisco Nexus 1000v provides Layer-2 switching functionality in a virtualized server environment. The Nexus 1000v will be available (Summer /09) as an additional feature of VMware’s vSphere infrastructure. Nexus 1000v allows network administrators to use Cisco’s NX-OS switching and monitoring features within an ESX host. The Nexus 1000v is a software-based solution to virtual machine switching requirements.

For more product-specific information on the Nexus 1000v, refer to the following URL:

http://www.cisco.com/en/US/products/ps9902/index.html

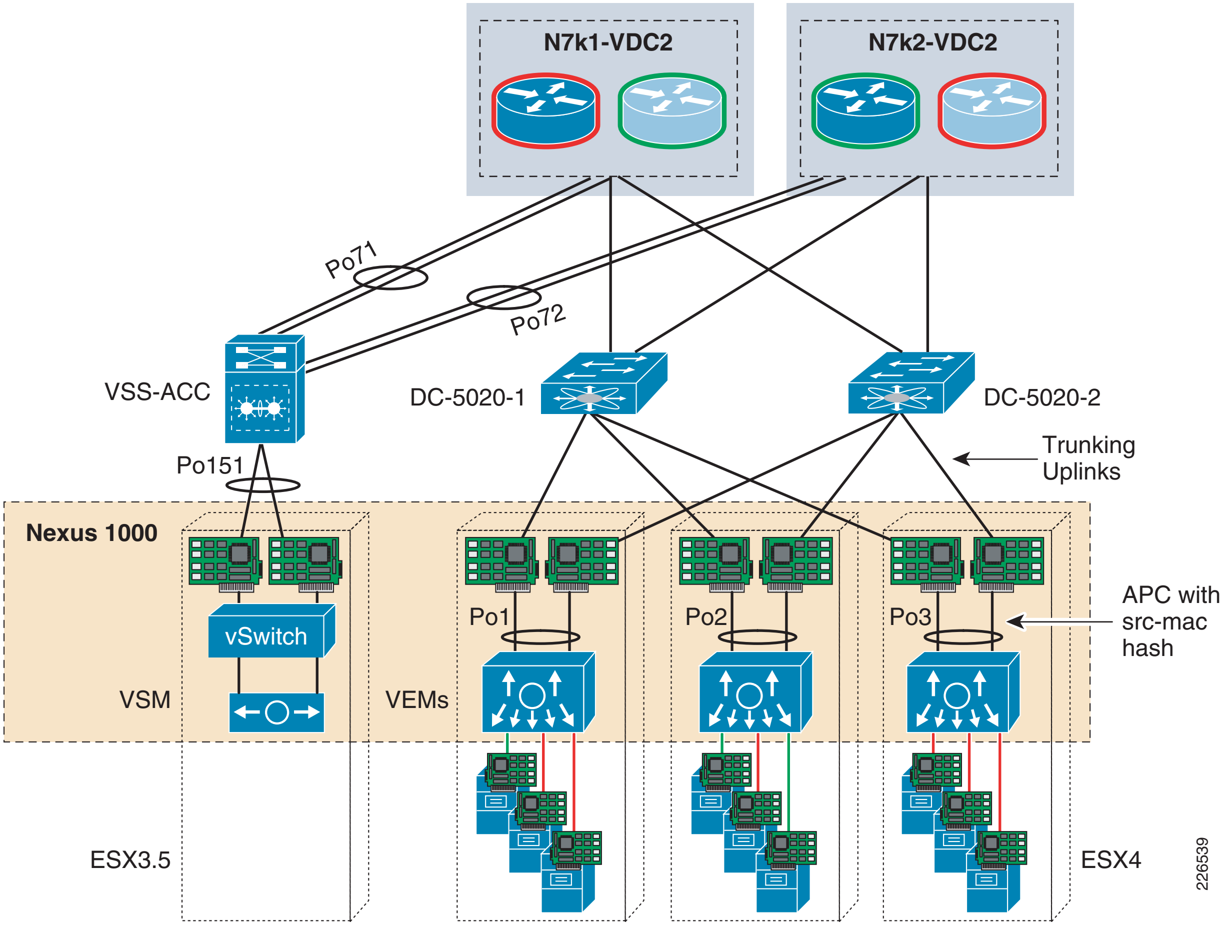
The Nexus 1000v is designed using two primary software objects: Virtual Supervisor Module (VSM) and Virtual Ethernet Module (VEM). VSM is the management interface or controller of a Nexus 1000v deployment. VSM is actually a dedicated virtual machine or physical server running an image based on Cisco’s NX-OS. VSM manages the VEMs and communicates directly to the vSphere Virtual Center.

VEMs perform the local Nexus 1000v-switching role on each ESX host defined as part of the Nexus 1000v Distributed Virtual Switch (DVS). VSM communicates with the VEM instances through dedicated control and packet VLANs. This is a one-to-many model where one VSM manages one or more VEMs.

Figure 11 represents the Nexus 1000v deployment used in this solution. Note the following:

- VSM resides on a separate ESX platform. This platform is part of the same vSphere data center. In this example, an ESX 3.5 environment houses the VSM and connects to a traditional vSwitch defined on the local ESX host. This vSwitch is using a static EtherChannel (Po151) to a VSS access layer switch.
- The ESX servers are dual-homed to a Nexus 5000 access layer through 10-Gigabit dual-port CNAs.
- The ESX uplinks are trunks carrying Nexus 1000v control traffic, VMkernel traffic (including iSCSI and VMotion), monitored, and data traffic. Integrated Gigabit interfaces of the server platform provide service console support. The service console connection is not shown.
- The VSM allows for centrally defined port profiles for both uplinks and VM access ports across all VEMs in the VDS.
- Asymmetric Port Channeling (APC) is used by each VEM to provide functionality similar to EtherChannel to the Nexus 5000 access layer. (For more information on APC in the Nexus 1000v, refer to Nexus 1000v, page 55.) It is important to note that the port channel exists within the ESX host; the Nexus 5000 layer is not configured as a channel, but simply as trunk ports to the ESX server.
- All VMs use the VMXnet3 driver allowing for 10-Gigabit connectivity.
- The default gateways for the VMs reside in the Nexus 7000 VRF aggregation instances.

Figure 11 Tested Solution Nexus 1000v Deployment Model



The Nexus 1000v deployment allowed for centralized management of the virtual switching environment within the data center. In addition, the Nexus 1000v optimized the utilization of the ESX uplinks to the access layer through APC while the NetFlow and ERSPAN features provided unprecedented network monitoring to the VMs. The convergence of all these traffic types on the CNAs is a major benefit and trend that will most likely build momentum.

Service Traffic Patterns

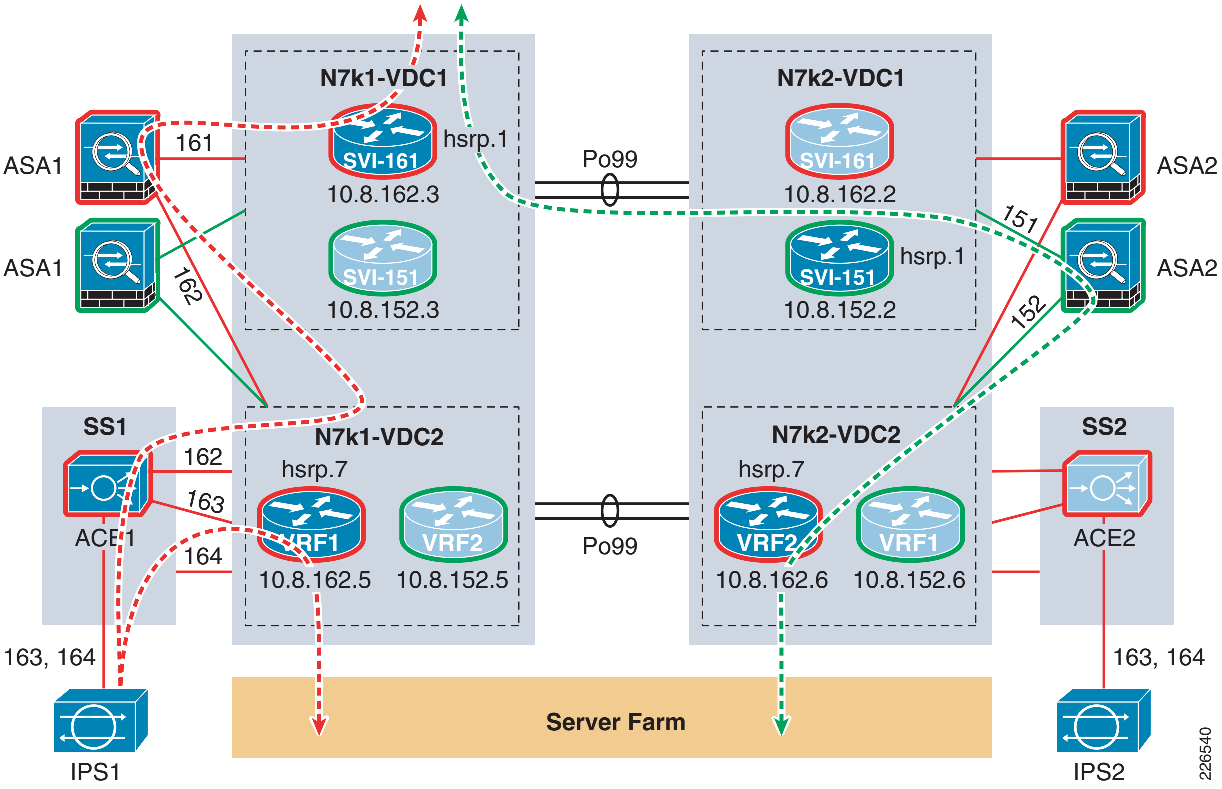
This section of the document discusses several traffic patterns within this active-active solution design including the following:

- Client-to-Server
- Server-to-Server
- ERSPAN
- NetFlow

This solution essentially provides two primary paths in the data center for ingress and egress traffic. In Figure 12, there are two service paths available to data center traffic. The red service path uses virtual devices including VDCs, VRFs, ASA, ACE, and IPS contexts primary on the left side of the infrastructure, while the green service path implements only virtual firewalling between VDCs and

VRFs. Figure 12 highlights the active-active functionality this solution offers and the ability to include/exclude services where necessary by segmenting traffic across virtual devices. The remainder of this document focuses on the red service path and its implementation.

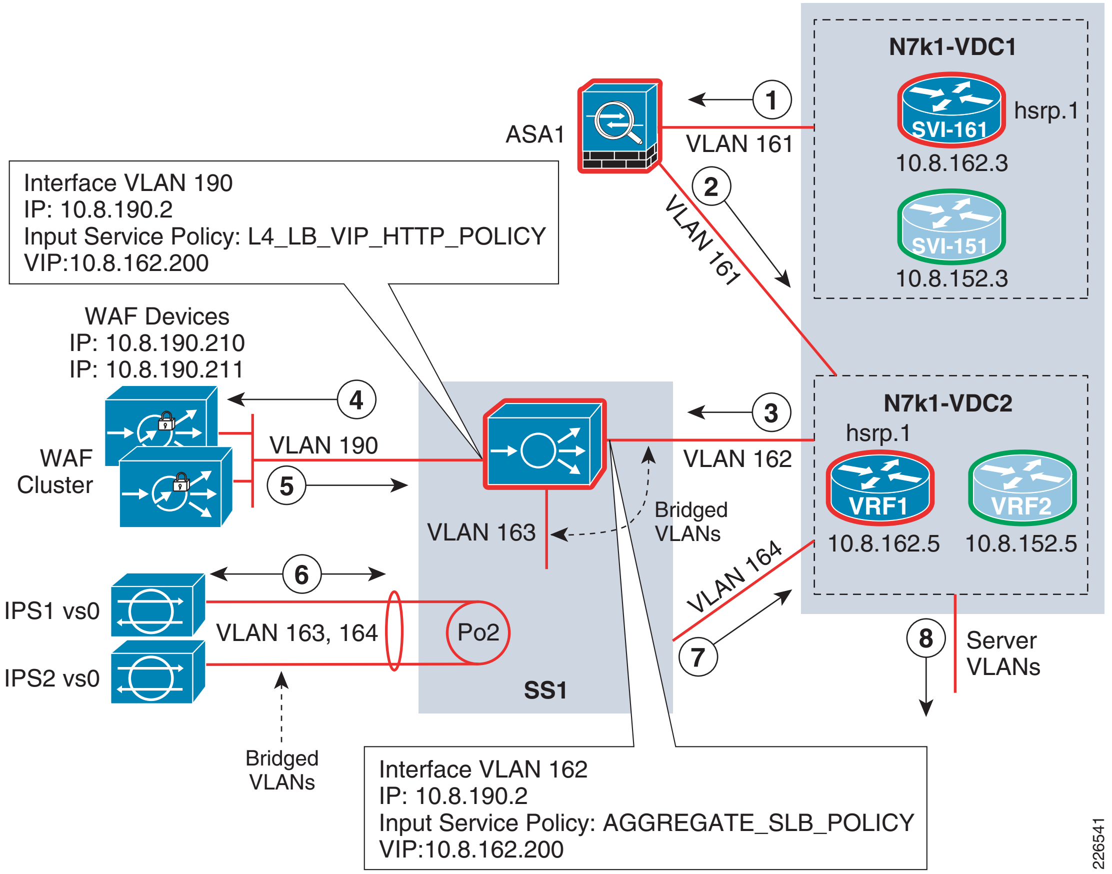
Figure 12 Example Active-Active Path Solution



Client-to-Server

Figure 13 illustrates the service flow for client-to-server traffic in the red traffic path. In this example, the client is making a web request to a virtual IP address (vIP) defined on the ACE virtual context.

Figure 13 Client-to-server Traffic Flow



The following steps describe the associated stages in Figure 13.

Step 1 Client is directed through OSPF route found on Nexus 7000-1 **VDC1** to the active ASA virtual context transparently bridging traffic between **VDC1** and **VDC2** on the Nexus 7000.

Step 2 The transparent ASA virtual context forwards traffic from VLAN 161 to VLAN 162 towards Nexus 7000-1 **VDC2**.

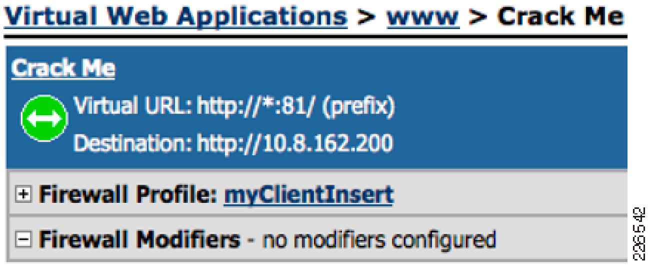
Step 3 **VDC2** shows spanning tree root for VLAN 162 through connection to services switch **SS1**. **SS1** shows spanning tree root for VLAN 162 through the ACE transparent virtual context.

Step 4 The ACE transparent virtual context applies an input service policy on VLAN 162, this service policy named **AGGREGATE_SLB** has the VIP definition. The VIP rules associated with this policy enforce SSL termination services and load balancing services to a web-application firewall serverfarm. The state of the WAF serverfarm is determined through HTTP based probes. The request is forwarded to a specific WAF appliance defined in the ACE serverfarm. The client IP address is inserted as an HTTP header by the ACE to maintain the integrity of server-based logging within the farm. Figure 15 shows this as the **ACEForwarded** HTTP header value. The source IP address of the request forwarded to the WAF is that of the originating client in this example 10.7.54.34.

Step 5 In this example, the WAF has a virtual web application defined name **Crack Me**. As shown in Figure 14, the virtual web application is listening on port 81 for all HTTP requests. The WAF appliance receives the HTTP request on port 81 that was forwarded from the ACE. The WAF applies all the relevant security policies for this traffic and proxies the request back to a VIP (10.8.162.200) located on the same virtual ACE context on VLAN interface 190.

Note The WAF appliance may also insert the source IP address of the client through an HTTP header policy. In this example, both the ACE and WAF insert the source IP address of the client. This is of course an overkill, but it does show the functionality is available in either platform. In Figure 15, the WAF inserted HTTP header is named **X-Forwarded-For**.

Figure 14 Example Virtual Web Application defined on the WAF Cluster



Virtual Web Applications > www > Crack Me

Crack Me

Virtual URL: http://*:81/ (prefix)

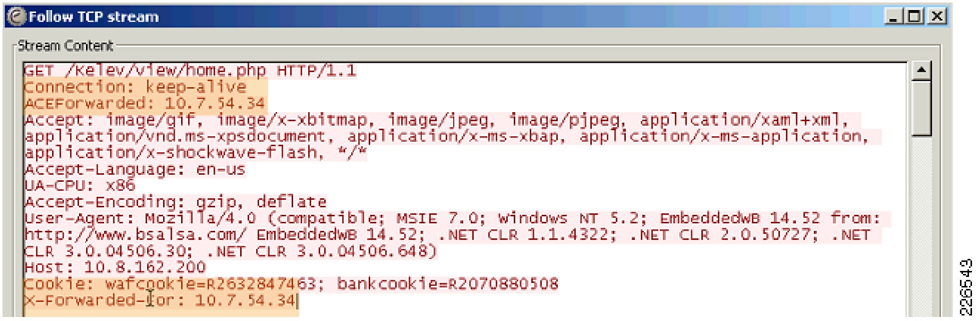
Destination: http://10.8.162.200

⊞ Firewall Profile: myClientInsert

⊟ Firewall Modifiers - no modifiers configured

Note The web site used in this example was developed by Cenzic corporation and part of their Hailstorm product line that provides security assessment tools.

Figure 15 ACE and WAF HTTP Header Insertion of Source IP Address Captured from Server



```
GET /Kelev/view/home.php HTTP/1.1
Connection: keep-alive
ACEForwarded: 10.7.54.34
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/xaml+xml,
application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-ms-application,
application/x-shockwave-flash, */*
Accept-Language: en-us
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; EmbeddedWB 14.52 from:
http://www.bsalsa.com/ EmbeddedWB 14.52; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET
CLR 3.0.04506.30; .NET CLR 3.0.04506.648)
Host: 10.8.162.200
Cookie: wafcookie=R2632847463; bankcookie=R2070880508
X-Forwarded-For: 10.7.54.34
```

Step 6 The ACE transparent virtual context receives the ingress WAF proxied request and forwards it to a real server. The path to the server is through VLAN interface 163. It is important to remember at this point the IP header reflects the source address of the relevant WAF appliance and a destination IP address of a real server.

The ACE virtual context has a static route to the real servers in the serverfarm through the HSRP address of the red VRF of 10.8.162.7. The **show arp** command verifies that gateway, if available through VLAN 163.

```
Static route configuration:
ip route 10.8.180.0 255.255.255.0 10.8.162.7
ace-vc#show arp
10.8.162.7       00.00.0c.07.ac.02   vlan163    GATEWAY    67      106 sec      up
```

The SS1 service switch confirms the path to the gateway is through port-channel 2, the IPS cluster.

```
SS1#show mac-address-table address 0000.0c07.ac02 vlan 163
vlan    mac address      type     learn     age              ports
------+----------------+--------+-----+----------+-------------------------
Module 1:
*  163   0000.0c07.ac02    dynamic  Yes           5    Po2
```

Note The spanning tree root for the bridged VLANs 161, 162, and 164 is VLAN 163 and is defined as such on the **SS1** service switch.

The use of a static EtherChannel to the IPS devices supports the high availability and scalability requirements of the data center. The IPS 4270 supports 10-Gigabit Ethernet connections in combination with an 8-port EtherChannel; thus, the possibility that the IPS cluster will be a bottleneck or single point-of-failure is greatly reduced.

The IPS supports sensor virtualization. This capability allows network administrators to create multiple inline VLAN pairs that are associated with independent virtual sensors. Network administrators may then fine tune each sensor to the specific requirements of each application and eliminate false positives.

For more details regarding data center security, refer to S*ecurity and Virtualization in the Data Center* at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_sec_design.html

Step 7 The IPS sensor applies the security policies to the traffic and forwards it to the VRF gateway.

Step 8 The VRF routes the traffic to the appropriate server in the access layer.

Server-to-Server

Typically, enterprise class applications require more availability, scalability, and/or processing power than a single server instance can provide. This has traditionally led to siloed application deployment models where dedicated physical appliances and servers form the backbone of one application. Virtualization has broken this model, offering flexibility to the logical design of the data center network and its applications.

This section discusses server-to-server traffic patterns, including the following:

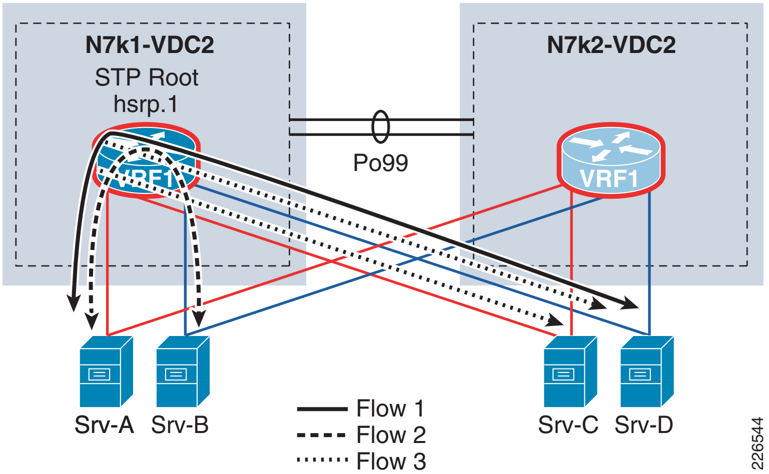
- Intra-VRF
- Intra-VRF with Services
- Inter-VRF

The access layers defined earlier in this document are not explored in this section. This section only focus on the flexibility afforded to the serverfarm when virtual network services are used.

Intra-VRF

Figure 16 depicts the traffic pattern for servers utilizing the same VRF instance. The SVI for each server VLAN is a member of the VRF located on **VDC2** of the Nexus 7000 aggregation switches. The HSRP alias address of each VLAN is the default gateway for the servers in that VLAN. For example, in Figure 16, **Srv-A**'s default gateway is the HSRP .1 address of the orange VLAN. The orange VLAN is a member of VRF **vrf1**. The VRF instance also provides local routing between servers on different VLANs. Each of the traffic flows depicted in Figure 16 illustrate this point. Server traffic within the same VRF will not exit the VRF or the Nexus 7000 VDC. The VRF contains Layer 2 and creates a local Layer-3 forwarding path between processes located in different VLANs.

Figure 16 Intra-VRF Server-to-Server Traffic Pattern

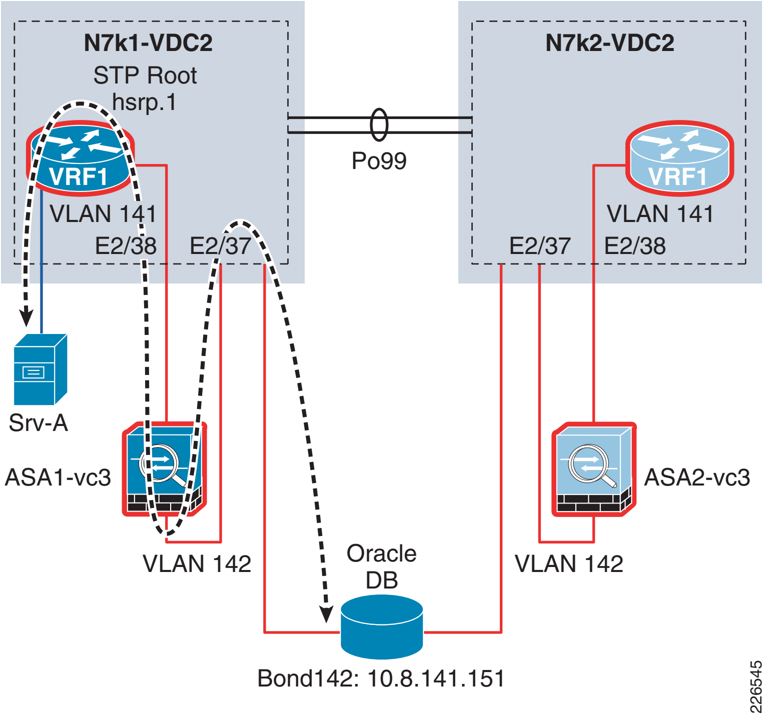


The use of VRFs to contain the Layers 2 and 3 aspects of the network can also be extended to the applications environment. For example, independent application processes the required connectivity, but do not require other services such as security or load balancing between them could be logically grouped within the same VRF instance. This concept of a VRF application zone can be extended throughout the enterprise data center providing segmentation to numerous application environments.

Intra-VRF with Services

The use of VRF application zones does not exclude the use of network services; in fact, it allows more granular control of those services. For example, Figure 17 depicts a VRF application zone where a transparent virtual ASA context is protecting an Oracle database instance. The default gateway for the Oracle database server is the **.1** HSRP address associated with VLAN 141. The 141 SVI is a member of **vrf1**. **Srv-A** is an Oracle node and uses another VLAN that is a member of the same VRF, **vrf1**. The firewall permits the **Srv-A** Oracle node to connect to the database. The virtual firewall context is not tasked with protecting the whole data center, but a specific server, a specific application. The VRF application zone allows for more specific service rules to be applied, be they security, visibility, scale, or performance-based.

Figure 17 Intra-VRF Traffic Pattern with Services



Inter-VRF

The use of VRF application zones provides segmentation and flexible service options within the aggregation layer of the data center. These characteristics are certainly beneficial, but communication between these zones may also be necessary as applications evolve and their requirements change. Figure 18 illustrates the inter-VRF communication between two server endpoints.

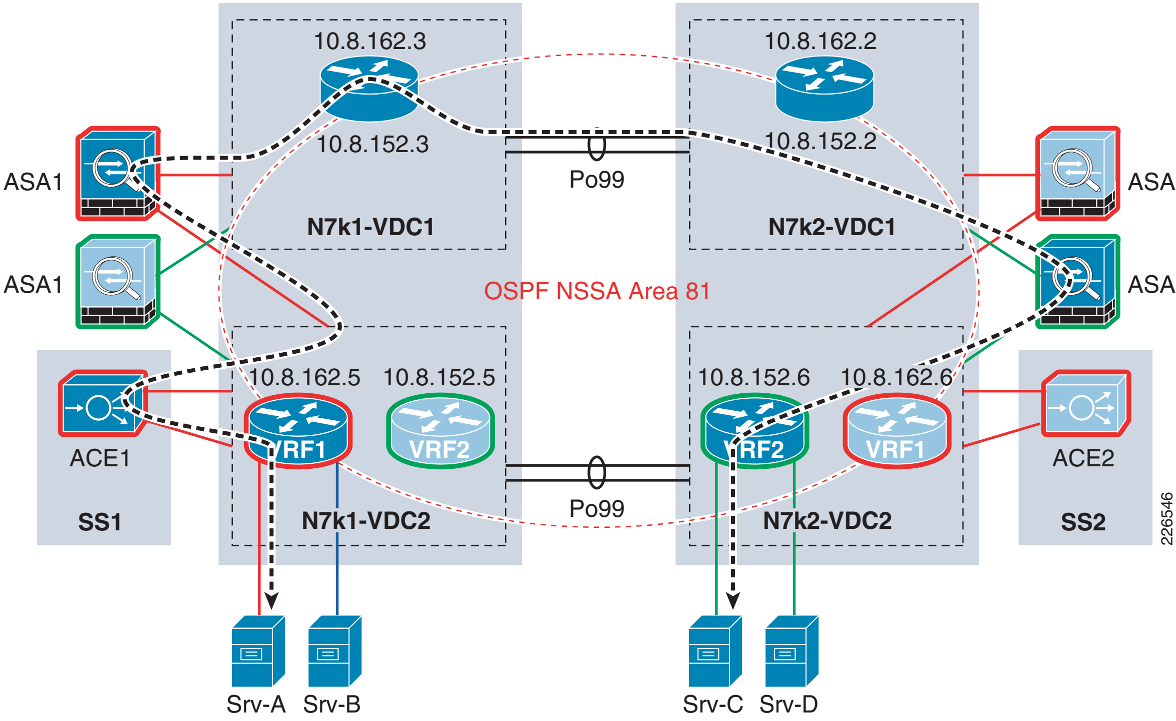
In this example, server **Srv-A** in VRF **vrf1** is communicating with **Srv-C** in VRF **vrf2**. VRFs create their own Layer 3 domain and are unaware of the other VRF located on the same VDC **VDC2**. For example, to reach 10.8.152.6 from **vrf1**, the routing table shows the following routes:

```
n7k1-vdc2# show ip route 10.8.152.6 vrf vrf1
IP Route Table for VRF "vrf1"
10.8.152.0/24, 2 ucast next-hops, 0 mcast next-hops
   *via 10.8.162.2, Vlan164, [110/50], 5d02h, ospf-8, intra
   *via 10.8.162.3, Vlan164, [110/50], 5d02h, ospf-8, intra
```

As Figure 18 shows, the route directs traffic through the transparent services layer and across the virtual ASA firewall context to **VDC1**. The **VDC1** routing table specifies a route through **ASA2**, which is a transparent virtual firewall context bridging VLAN 151 to 152 that exists on **VDC2**.

```
n7k1-vdc1# show ip route 10.8.152.6
IP Route Table for VRF "default"
10.8.152.6/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.8.152.6, Vlan151, [2/0], 5d02h, am
```

Figure 18 Inter-VRF Traffic Pattern



The inter-VRF traffic patterns maintain the integrity of the service polices established for each VRF application zone. This forces network, security, and server administrators to create policies within and workflows between VRF application zones that address the enterprise application as a whole.

Note The use of a third VRF to allow inter-VRF communication without exiting the VDC will be examined in future testing efforts.
The services layer shown in Figure 18 is not comprehensive; other services including IDS, IPS, and WAF services were used for testing of this solution, but are not included in this document in order to focus on the required features and the removed features are optional.

For more information on the use of VRFs on the Nexus 7000, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_1/nx-os/unicast/configuration/guide/l3_virtual.html

ERSPAN

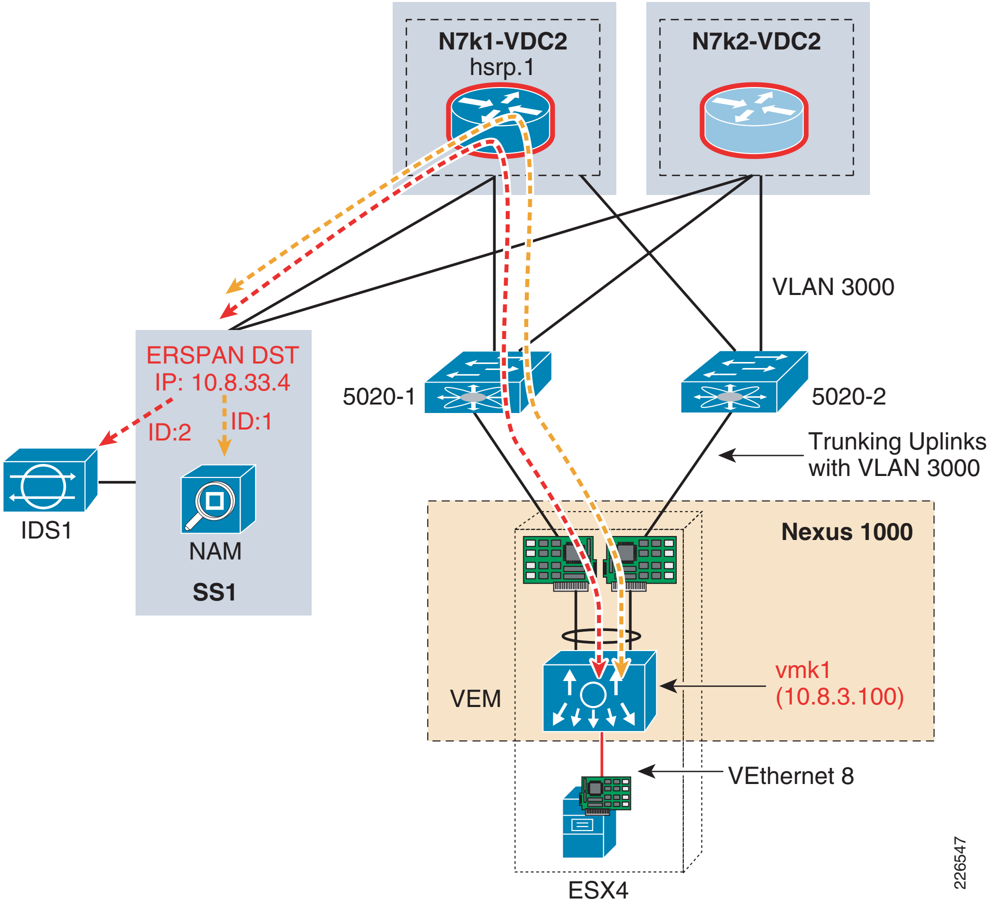
ERSPAN allows for remote monitoring of network resources. ERSPAN uses GRE tunnels to route traffic to the appropriate destination. The Nexus 1000v supports ERSPAN, allowing network administrators to observe the traffic associated with the following:

- The individual vNIC of a virtual machine connected to a VEM
- The physical ports associated with the ESX host
- Any port channels defined on the VEM

This flexibility allows the ERSPAN session to not only monitor data associated with virtual machines, but to monitor all traffic associated with the ESX host including VMkernel, VMotion, and service console data. Converging all of these traffic types onto two or a maximum of four CNAs per-ESX host simplifies not only the physical design of the data center but the configuration of the capture points as well.

Figure 19 depicts the traffic pattern used for ERSPAN monitoring in the tested solution. The ESX host is a VEM of the Nexus 1000v DVS. The ESX host has a **vmknic** defined, in this example 10.8.3.100 in VLAN 3000. The uplink port profiles for this VEM support this VLAN through APC. The default gateway for the ERSPAN VMkernel interface is an HSRP address defined on the Nexus 7000 VDC2 red VRF. The destination for all ERSPAN traffic in this example is IP address 10.8.33.4, which resides on Service Chassis **SS1**. The Nexus 7000 VRF has routes the GRE encapsulated ERSPAN packets to the final destination IP address (10.8.33.4).

Figure 19 ERSPAN Traffic Pattern



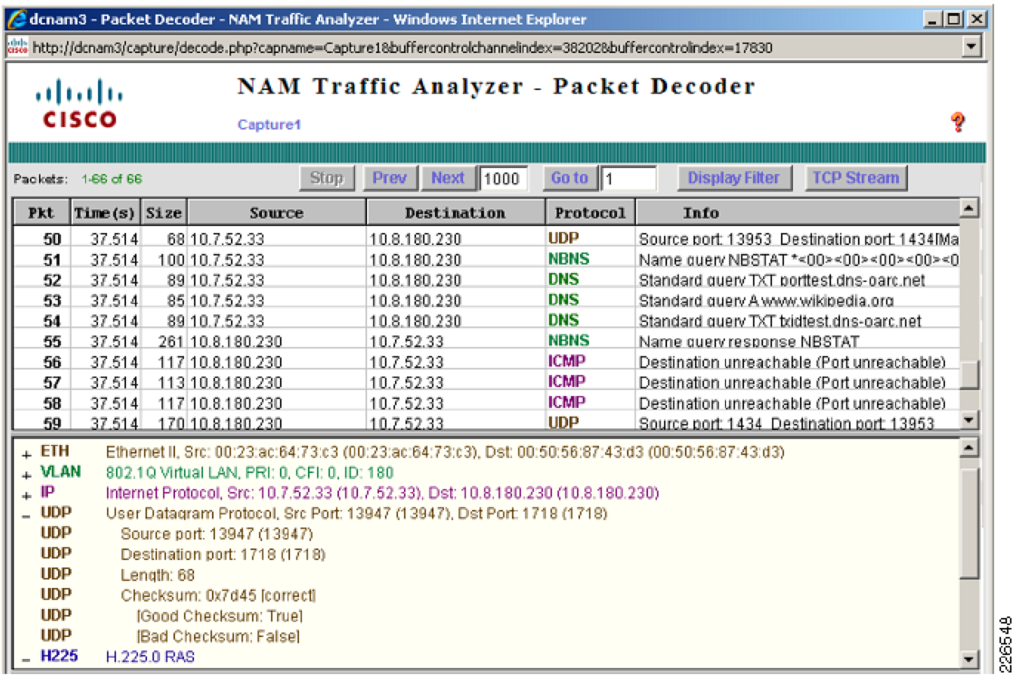
The **SS1** Catalyst 6500 employs two ERSPAN monitor sessions that mirror ERSPAN traffic simultaneously to a NAM service module or a virtual sensor IDS device. The ERSPAN capabilities of the Nexus 1000v allow for additional network visibility, down to the vNIC and security.

Note The IDS virtual sensor resides on the same physical IPS4270 devices that support the inline IPS virtual sensors.

Caution The use of advanced features such as ERSPAN will consume additional resources (i.e., memory and CPU of the ESX host). It is important to understand these resource dynamics before enabling any advanced features.

Figure 20 and Figure 21 show the impact of network monitoring at the VM level. In these examples, a port scan was executed from a remote client (10.7.52.33) on a VM (10.8.180.230) using the Nexus 1000v as an access layer. Simultaneously, the NAM and the IDS were able to capture the intrusion.

Figure 20 View of NAM Captured Data from VM NIC


dcnam3 - Packet Decoder - NAM Traffic Analyzer - Windows Internet Explorer
CISCO
Stop
Prev
Next
1000
Go to
1
Display Filter
TCP Stream
226548

http://dcnam3/capture/decode.php?capname=Capture1&buffercontrolchannelindex=38202&buffercontrolindex=17830

NAM Traffic Analyzer - Packet Decoder

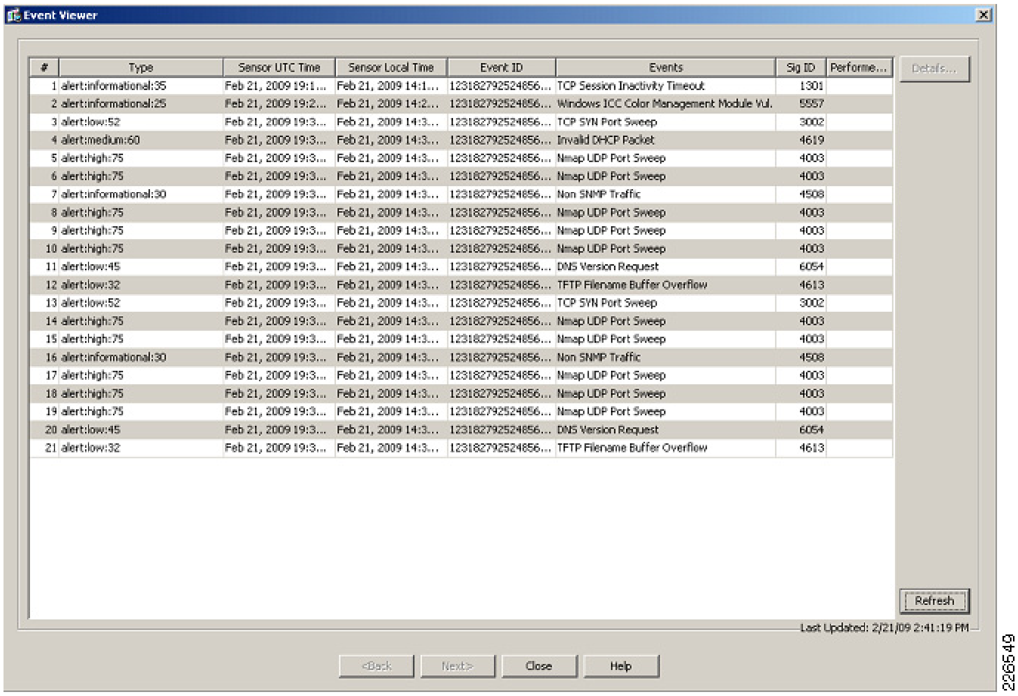
Capture1

Packets: 1-66 of 66

Pkt	Time(s)	Size	Source	Destination	Protocol	Info
50	37.514	68	10.7.52.33	10.8.180.230	UDP	Source port: 13953 Destination port: 1434[Ma
51	37.514	100	10.7.52.33	10.8.180.230	NBNS	Name query NBSTAT *<00><00><00><00><0
52	37.514	89	10.7.52.33	10.8.180.230	DNS	Standard query TXT porttest.dns-oarc.net
53	37.514	85	10.7.52.33	10.8.180.230	DNS	Standard query A www.wikipedia.org
54	37.514	89	10.7.52.33	10.8.180.230	DNS	Standard query TXT txidtest.dns-oarc.net
55	37.514	261	10.8.180.230	10.7.52.33	NBNS	Name query response NBSTAT
56	37.514	117	10.8.180.230	10.7.52.33	ICMP	Destination unreachable (Port unreachable)
57	37.514	113	10.8.180.230	10.7.52.33	ICMP	Destination unreachable (Port unreachable)
58	37.514	117	10.8.180.230	10.7.52.33	ICMP	Destination unreachable (Port unreachable)
59	37.514	170	10.8.180.230	10.7.52.33	UDP	Source port: 1434 Destination port: 13953

```
+ ETH    Ethernet II, Src: 00:23:ac:64:73:c3 (00:23:ac:64:73:c3), Dst: 00:50:56:87:43:d3 (00:50:56:87:43:d3)
+ VLAN   802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 180
+ IP     Internet Protocol, Src: 10.7.52.33 (10.7.52.33), Dst: 10.8.180.230 (10.8.180.230)
- UDP    User Datagram Protocol, Src Port: 13947 (13947), Dst Port: 1718 (1718)
  UDP      Source port: 13947 (13947)
  UDP      Destination port: 1718 (1718)
  UDP      Length: 68
  UDP      Checksum: 0x7d45 [correct]
  UDP        [Good Checksum: True]
  UDP        [Bad Checksum: False]
- H225   H.225.0 RAS
```

Figure 21 *Example IDS Alerts from ERSPAN Traffic*



#	Type	Sensor UTC Time	Sensor Local Time	Event ID	Events	Sig ID	Performe...
1	alert:informational:35	Feb 21, 2009 19:1...	Feb 21, 2009 14:1...	123182792524856...	TCP Session Inactivity Timeout	1301	
2	alert:informational:25	Feb 21, 2009 19:2...	Feb 21, 2009 14:2...	123182792524856...	Windows ICC Color Management Module Vul.	5557	
3	alert:low:52	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	TCP SYN Port Sweep	3002	
4	alert:medium:60	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Invalid DHCP Packet	4619	
5	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
6	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
7	alert:informational:30	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Non SNMP Traffic	4508	
8	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
9	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
10	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
11	alert:low:45	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	DNS Version Request	6054	
12	alert:low:32	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	TFTP Filename Buffer Overflow	4613	
13	alert:low:52	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	TCP SYN Port Sweep	3002	
14	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
15	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
16	alert:informational:30	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Non SNMP Traffic	4508	
17	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
18	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
19	alert:high:75	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	Nmap UDP Port Sweep	4003	
20	alert:low:45	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	DNS Version Request	6054	
21	alert:low:32	Feb 21, 2009 19:3...	Feb 21, 2009 14:3...	123182792524856...	TFTP Filename Buffer Overflow	4613	

Last Updated: 2/21/09 2:41:19 PM

NetFlow

The use of NetFlow is well documented in a traditional network environment, but the Nexus 1000v provides this capability within the virtual network environment. Nexus 1000v supports NetFlow v9 and by default will use the management **0** interface as an export source. However, with the presence of 10-Gigabit CNAs on the server, it is also possible to provide further consolidation within the data center by moving NetFlow traffic over these robust interfaces. Figure 22 illustrates this methodology. The NetFlow path in Figure 22 shows the CNAs using the Nexus 5000 access layer and routing over to a Cisco NetFlow Collector using a VSS access layer switch.

The NetFlow source interface defined on the DVS may also be on a dedicated out-of-band management network. This deployment model removes traffic from the CNA uplinks and takes more of a traditional approach to NetFlow captures.

Note To reduce the impact on local ESX memory resources adjust the flow cache table to limit the impact of NetFlow.

Caution The use of advanced features such as NetFlow will consume additional resources (i.e., memory and CPU, of your ESX host). It is important to understand these resource dynamics before enabling any advanced features.

Figure 22 In-band NetFlow Example

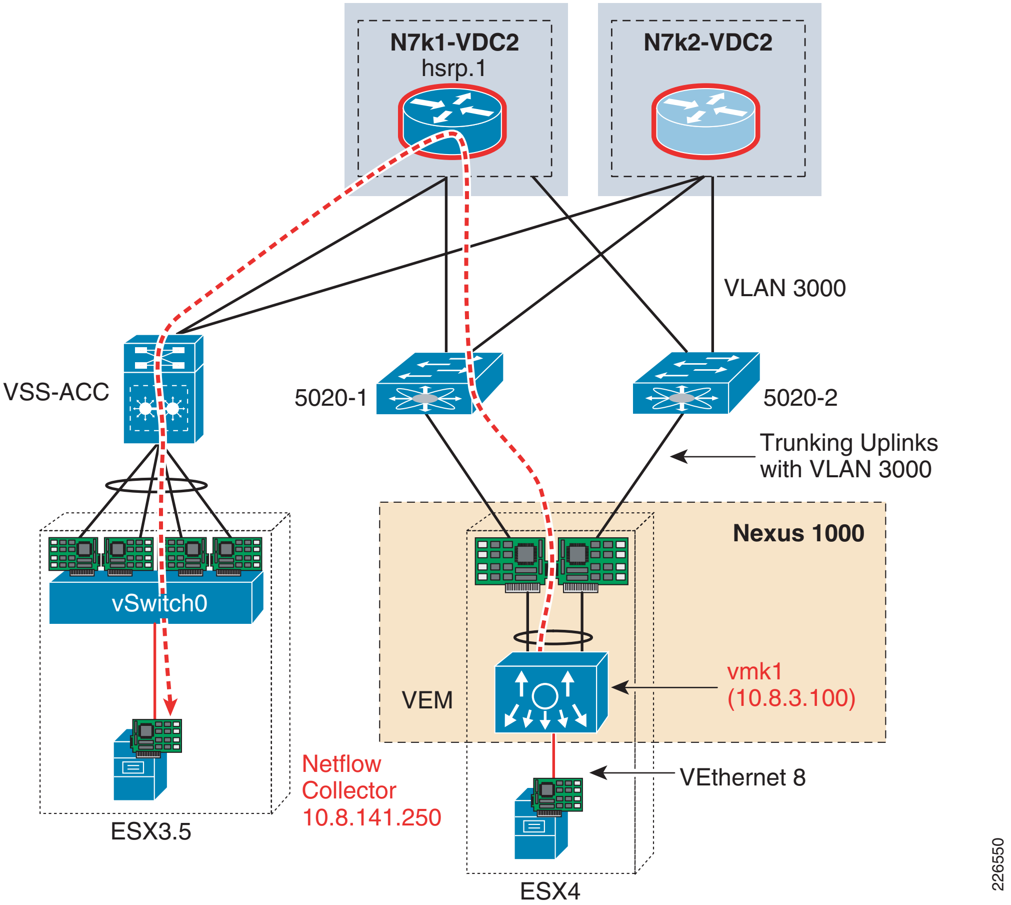
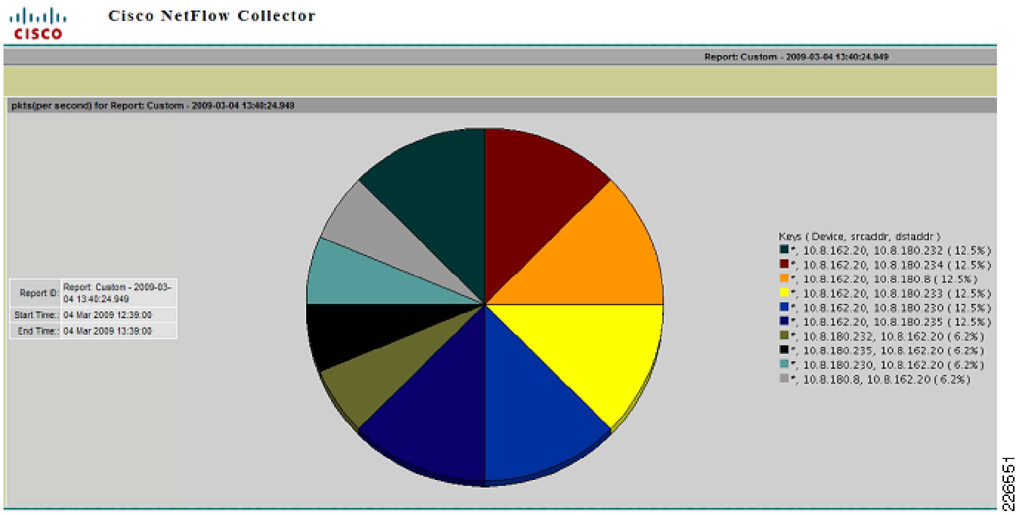


Figure 23 is an output example that shows the Cisco NetFlow Collector reporting packet/second utilization on the virtual Ethernet interfaces that reside on the Nexus 1000v. The Nexus 1000v may also monitor flows from the physical interfaces associated with the platform and VMkernel interfaces including VMotion traffic.

Figure 23 Example Cisco NetFlow Collector Nexus 1000v Results



Cisco NetFlow Collector

Report: Custom - 2009-03-04 13:40:24.949

pkts(per second) for Report: Custom - 2009-03-04 13:40:24.949

Report ID:	Report: Custom - 2009-03-04 13:40:24.949
Start Time::	04 Mar 2009 12:39:00
End Time::	04 Mar 2009 13:39:00

Keys (Device, srcaddr, dstaddr)
*, 10.8.162.20, 10.8.180.232 (12.5%)
*, 10.8.162.20, 10.8.180.234 (12.5%)
*, 10.8.162.20, 10.8.180.8 (12.5%)
*, 10.8.162.20, 10.8.180.233 (12.5%)
*, 10.8.162.20, 10.8.180.230 (12.5%)
*, 10.8.162.20, 10.8.180.235 (12.5%)
*, 10.8.180.232, 10.8.162.20 (6.2%)
*, 10.8.180.235, 10.8.162.20 (6.2%)
*, 10.8.180.230, 10.8.162.20 (6.2%)
*, 10.8.180.8, 10.8.162.20 (6.2%)

Implementation Details

This section provides more detailed configurations including the following:

- Aggregation Layer
- Services Layer
- Access Layer

Figure 24 and Figure 25 illustrate the Layer 3 and Layer 2 solution topology. These diagrams should be referenced when reviewing the implementation details. Note that Figure 25 focuses on the Layer 2 domain that exists between **vdc1** and **vdc2**. This domain is *active* on one side of the topology and uses many transparent network services.

Figure 24 Solution Layer 3 Topology

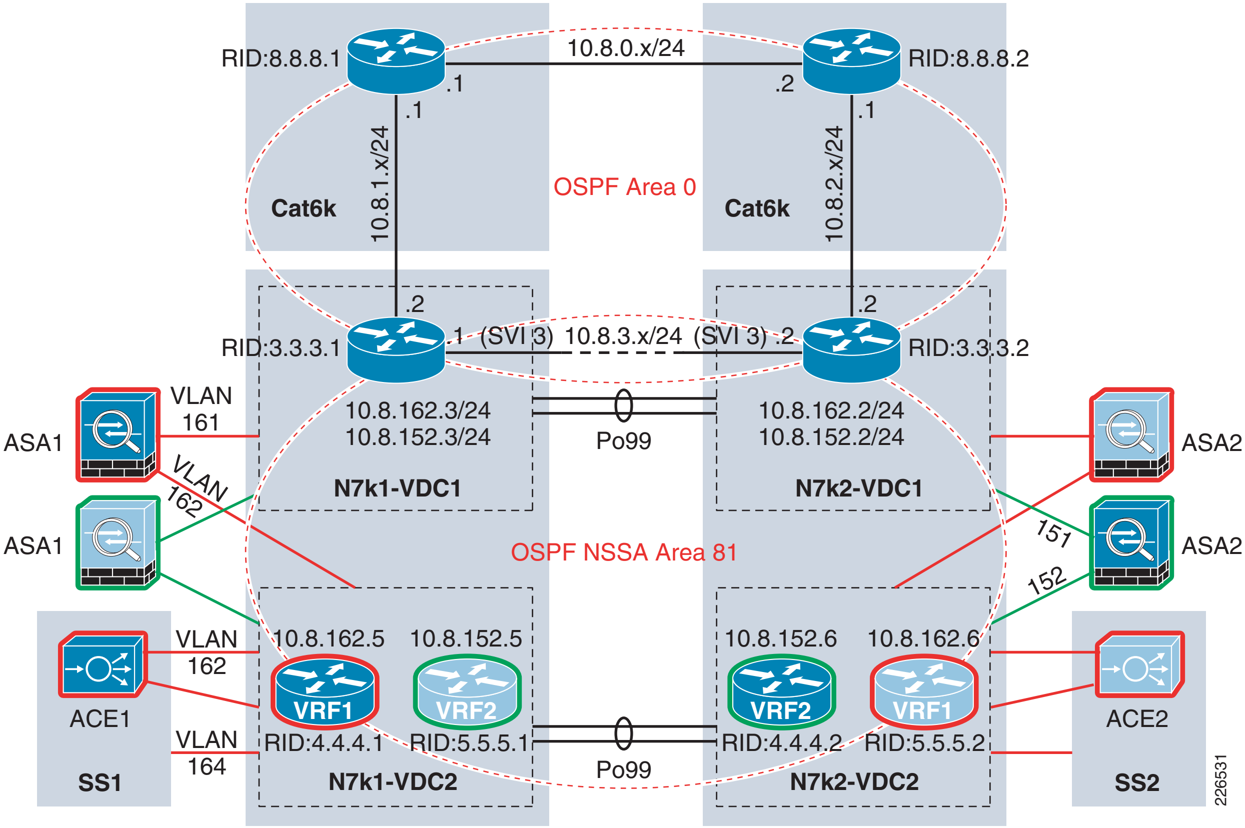
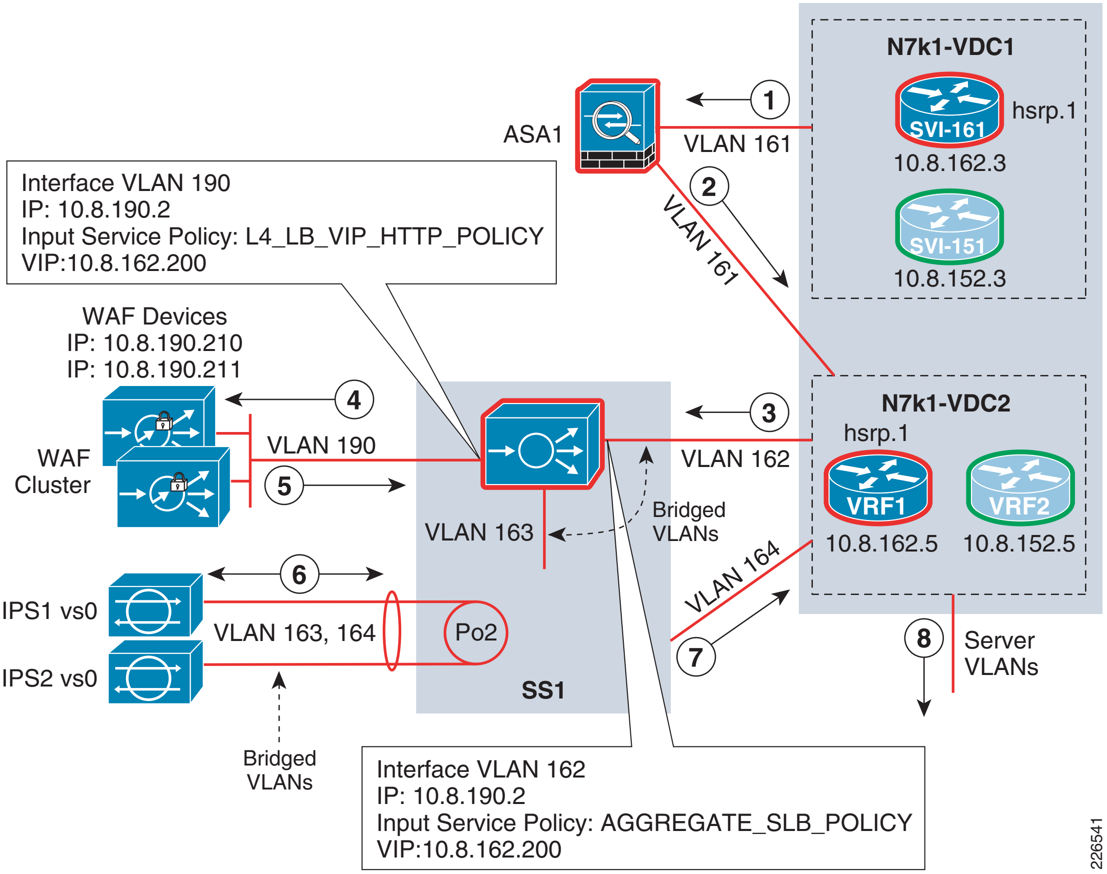


Figure 25 Solution Layer 2 Topology with Services



Aggregation Layer

Virtual Device Contexts

The solution’s aggregation layer consists of two Nexus 7000 platforms. Each Nexus 7000 device supports three virtual contexts using default resource allocation parameters. To create the VDCs, use the following commands from the default administrative context. In this example, **dca-n7k1** identifies the default context.

```
dca-n7k1(config)#vdc vdc1
dca-n7k1(config)#vdc vdc2
```

To validate the VDC instantiation use the **show vdc** command:

```
dca-n7k1# show vdc
vdc_id  vdc_name                          state             mac
------  --------                          -----             ----------
1       dca-n7k1                          active            00:23:ac:64:73:c1
2       vdc1                              active            00:23:ac:64:73:c2
3       vdc2                              active            00:23:ac:64:73:c3
```

VDCs do not share physical interface resources. Interfaces must be explicitly assigned to the VDC in port groups. The VDC physical interfaces can be shown with the **show vdc membership** command.

```
dca-n7k1-vdc1# show vdc membership
vdc_id: 2 vdc_name: vdc1 interfaces:
        Ethernet1/1           Ethernet1/3           Ethernet1/5
        Ethernet1/7           Ethernet1/9           Ethernet1/11
        Ethernet1/13          Ethernet1/15          Ethernet2/2
        Ethernet2/4           Ethernet2/6           Ethernet2/8
```

Note The remainder of the Nexus 700 configurations occurs within one of these two VDC instances, **vdc1** or **vdc2**.
The default administrative VDC may also operate as an aggregation layer device, but was not used as such in this solution.

Layer 3

Figure 24 shows the Layer 3 topology of the entire solution where Area “0” exists between the core Catalyst 6500s and Nexus 7000 VDCs. An NSSA 81 is defined between the Nexus 7000 VDCs and multiple VRFs defined on **vdc2**. Note that the neighboring relationship is created across the VDC entities logically bridging VLANs 161 to 164 as well as VLANs 151 to 152, respectively. For details, refer to “Layer 2” section on page 36.

Enable the Nexus 7000 OSPF feature set on each VDC with the following command:

feature ospf

Note The configuration examples are taken from the Nexus 7000 N7k1.

VDC Example—vdc1

Define the OSPF routing instance and its associated parameters. Note the NSSA Area 81 definition.

```
router ospf 8
  router-id 3.3.3.1
  area 81 nssa
  default-information originate
  area 0.0.0.0 range 10.8.0.0/24
  area 0.0.0.0 range 10.8.1.0/24
  area 0.0.0.0 range 10.8.2.0/24
  area 0.0.0.0 range 10.8.3.0/24
  area 0.0.0.81 range 10.8.128.0/18
  area 0.0.0.0 authentication message-digest
  area 0.0.0.81 authentication message-digest
  timers throttle spf 10 100 5000
  timers throttle lsa router 1000
  timers throttle lsa network 1000
  auto-cost reference-bandwidth 10000
```

Note These configurations are based upon Nexus 7000 Layer 3 configuration best practices available in the *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services at the following URL:* http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html

Assign interfaces to their appropriate area leveraging route security. In this example, VLAN 3 and Ethernet 1/1 belong to Area 0 while VLANs 151 and 161 are NSSA 81 members. VLAN 3 uses the port channel existing between the two **vdc1** instances of the Nexus 7000 aggregation layer.

```
vlan 3,151,161
interface Vlan3
  ip ospf authentication message-digest
  ip ospf authentication-key 3 9125d59c18a9b015
  ip ospf dead-interval 3
  ip ospf hello-interval 1
  ip router ospf 8 area 0.0.0.0

interface Vlan151
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip router ospf 8 area 0.0.0.81

interface Vlan161
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip router ospf 8 area 0.0.0.81

interface Ethernet1/1
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 9125d59c18a9b015
  ip ospf dead-interval 3
  ip ospf hello-interval 1
  ip router ospf 8 area 0.0.0.0
```

The use of HSRP provides default routes for the VRFs defined on the each of the **vdc2** instances. The following HSRP configurations were implemented on VLANs 151 and 161:

```
interface Vlan151
ip address 10.8.152.3/24
  hsrp 1
    authentication text c1sc0
    preempt delay minimum 180
    priority 10 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.152.1

interface Vlan161
ip address 10.8.162.3/24
  hsrp 1
    authentication text c1sc0
    preempt delay minimum 180
    priority 20 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.162.1
```

Each of the HSRP groups use authentication to secure communications and different priorities. In this example, VLAN 161 is set with a higher priority, in this case **20**, than its peer and will be active on this VDC **vdc1** on Nexus 7000 **N7k1**. Using alternating priorities helps to reinforce the active-active solution topology. The **show hsrp interface** command verifies the configuration.

```
dca-n7k1-vdc1# show hsrp interface vlan 161
Vlan161 - Group 1 (HSRP-V1) (IPv4)
  Local state is Active, priority 20 (Cfged 20), may preempt
    Forwarding threshold(for VPC), lower: 0 upper: 0
  Preemption Delay (Seconds) Minimum:180
  Hellotime 1 sec, holdtime 3 sec
  Next hello sent in 0.530000 sec(s)
  Virtual IP address is 10.8.162.1 (Cfged)
  Active router is local
  Standby router is 10.8.162.2
  Authentication text "c1sc0"
  Virtual mac address is 0000.0c07.ac01 (Default MAC)
  2 state changes, last state change 4w0d
  IP redundancy name is hsrp-Vlan161-1 (default)
```

VDC Example—vdc2

Enable OSPF feature set on the VDC with the following command:

feature ospf

Define the routing instance and create the VRFs as necessary to support your serverfarms. In this example, there are two VRFs housed in the **vdc2** context. The VRF becomes the default gateway for the servers associated with a VLAN through the membership interface configuration.

```
router ospf 8
  vrf servers1
    router-id 4.4.4.1
    area 81 nssa
    area 0.0.0.81 authentication message-digest
    timers throttle spf 10 100 5000
    timers throttle lsa router 1000
    timers throttle lsa network 1000
  vrf servers2
    router-id 5.5.5.1
    area 81 nssa
    area 0.0.0.81 authentication message-digest
    timers throttle spf 10 100 5000
    timers throttle lsa router 1000
    timers throttle lsa network 1000
```

Assign the appropriate VLAN interfaces to each VRF. In this example, VLANs 164 and 180 are members of VRF **vrf1**, while VLANs 152 and 181 are members of VRF **vrf2**. It is important to note that the 152 and 164 VLANs use a unique HSRP group identifier, in this case **2**, and an IP addressing scheme that compliments the 161 and 151 VLANs found on **vdc1**. This is necessary as VLANs 152 and 164 on **vdc2** are logical extensions of VLANs 151 and 161 defined on **vdc1**. The HSRP groups defined on these VLANs may be used as default gateways by service devices such as the ACE or WAF.

```
interface Vlan152
  no shutdown
  vrf member vrf2
  ip address 10.8.152.5/24
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip router ospf 8 area 0.0.0.81
  ip pim sparse-mode
  ip igmp version 3
  hsrp 2
    authentication text c1sc0
    preempt delay minimum 180
    priority 10 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.152.7

interface Vlan164
  no shutdown
  vrf member vrf1
  ip address 10.8.162.5/24
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip router ospf 8 area 0.0.0.81
  ip pim sparse-mode
  ip igmp version 3
  hsrp 2
    authentication text c1sc0
    preempt delay minimum 180
    priority 20 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.162.7
```

The 180 and 181 VLAN interfaces are part of their respective VRF and each is the default gateway for any number of servers. Again, to reinforce the active-active design, the HSRP *active* priority for each VLAN alternates between the two **vdc2** instances of the aggregation layer.

Note The interfaces are set to passive because the VLAN interface is the Layer 3 boundary for the serverfarm.

```
interface Vlan180
  no shutdown
  vrf member vrf1
  ip address 10.8.180.3/24
  ip ospf passive-interface
  ip router ospf 8 area 0.0.0.81
  ip pim sparse-mode
  ip igmp version 3
  hsrp 1
    authentication text c1sc0
    preempt delay minimum 180
    priority 20 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.180.1

interface Vlan181
  no shutdown
  vrf member vrf2
  ip address 10.8.181.3/24
  ip ospf passive-interface
  ip router ospf 8 area 0.0.0.81
  ip pim sparse-mode
  ip igmp version 3
  hsrp 1
    authentication text c1sc0
    preempt delay minimum 180
    priority 10 forwarding-threshold lower 0 upper 0
    timers  1  3
    ip 10.8.181.1
```

The **show ip ospf neighbors** command verifies that the routing relationships exist between **vdc1** and the VRFs on **vdc2**.

```
dca-n7k1-vdc2# show ip ospf neighbors vrf vrf1
 OSPF Process ID 8 VRF vrf1
 Total number of neighbors: 3
 Neighbor ID     Pri State            Up Time  Address          Interface
 3.3.3.1           1 FULL/DROTHER     3d22h    10.8.162.3       Vlan164
 3.3.3.2           1 FULL/DROTHER     3d22h    10.8.162.2       Vlan164
 4.4.4.2           1 FULL/DR          4d00h    10.8.162.6       Vlan164
dca-n7k1-vdc2# show ip ospf neighbors vrf vrf2
 OSPF Process ID 8 VRF vrf2
 Total number of neighbors: 3
 Neighbor ID     Pri State            Up Time  Address          Interface
 3.3.3.1           1 FULL/DROTHER     1w3d     10.8.152.3       Vlan152
 3.3.3.2           1 FULL/DROTHER     1w3d     10.8.152.2       Vlan152
 5.5.5.2           1 FULL/DR          2w2d     10.8.152.6       Vlan152
```

Layer 2

Figure 25 shows the Layer 2 topology of one active side of the solution. This topology uses all of the transparent services discussed earlier in the document. The Nexus 7000 aggregation layer contains this Layer 2 service domain between two VDCs. The following discusses the Layer 2 implementation of the VDCs in the aggregation layer. The testing of this solution was conducted using Rapid PVST+.

Note These configurations are based upon the Nexus 7000 Layer-2 configuration best practices available in the *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services at the following URL:* http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html.

VDC Example—vdc1

Bridge assurance monitors point-to-point links for BPDU reception, validating the integrity of the forwarding path. Bridge assurance must be enabled on both sides of the link; otherwise, the port will be placed in a blocking state. The following command enables bridge assurance globally on the VDC.

spanning-tree port type network default

Set the spanning tree root for local VLANs. In this example, the root for VLAN 161 is defined as VLAN 163 on the services switch. The root priority is not set for VLAN 161 on the VDC. Following is an example of setting the root priority for local VLANs that do not require the services layer.

```
spanning-tree vlan 99,128,130,132,166,770-771 priority 24576
```

A higher priority setting is used to indicate the Nexus VDC is a secondary switch.

```
spanning-tree vlan 129,131,133 priority 28672
```

The Inter Switch Link (ISL) between the Nexus 7000 VDC **vdc1** supports all of the local and service VLANs. This port channel uses multiple physical links for availability and scalability of the environment.

```
configure interface port-channel99
  spanning-tree port type network
```

The ASA 5580 does not currently support bridge assurance, each of the ASA connections are configured as normal Layer 2 links. Note that the root guard is not enabled on these connections because VDC **vdc1** is not the root for service VLANs.

```
configure interface Ethernet1/3
  description to dca-asa1 vc1
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 161
  spanning-tree port type normal
```

The **show spanning-tree vlan** command shows that the root switch for VLAN 161 is through the ASA 5580 at interface Ethernet1/3.

```
dca-n7k1-vdc1# show spanning-tree vlan 161
VLAN0161
  Spanning tree enabled protocol rstp
  Root ID    Priority    24739
             Address     0021.d72a.c000
             Cost        2004
             Port        131 (Ethernet1/3)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32929  (priority 32768 sys-id-ext 161)
             Address     0023.ac64.73c2
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po99             Desg FWD 1         128.4194 Network P2p
Eth1/3           Root FWD 2         128.131  P2p
```

VDC Example—vdc2

As shown in Figure 25, VLAN 161 exits **vdc1**, enters the ASA virtual context, and returns to the same Nexus 7000 chassis through Ethernet 1/10 carrying VLAN 162 on **vdc2**. The ASA 5580 transparent virtual context bridges the VLANs between the Nexus VDCs. This forces traffic to adhere to the security policies enabled on the ASA context. The root for VLAN 162 is shown at Ethernet 1/2 on **vdc2**. Ethernet 1/2 is a trunk connecting to the active services path while Ethernet 1/4 is a trunk port to the standby services path for VLAN 161-164 services. Port channel 99 is the ISL between the **vdc2** contexts at the aggregation layer.

```
dca-n7k1-vdc2# show spanning-tree vlan 162
VLAN0162
  Spanning tree enabled protocol rstp
  Root ID    Priority    24739
             Address     0021.d72a.c000
             Cost        2002
             Port        130 (Ethernet1/2)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32930  (priority 32768 sys-id-ext 162)
             Address     0023.ac64.73c3
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po99             Desg FWD 1         128.4194 Network P2p
Eth1/2           Root FWD 2         128.130  Network P2p
Eth1/4           Desg FWD 2         128.132  Network P2p
Eth1/10          Desg FWD 2         128.138  P2p
```

Figure 25 indicates that VLAN 162, 163, and 164 are transparently bridged within the services chassis. VLAN 164 returns from the services chassis on Ethernet 1/2 where the VLAN is a member of VRF **vrf1** and the Layer 2 domain is contained. See the "Layer 3" section on page 32 for VRF configuration details.

Services Layer

This section describes the services layer implementation used during solution testing. The following Cisco service devices and services are detailed:

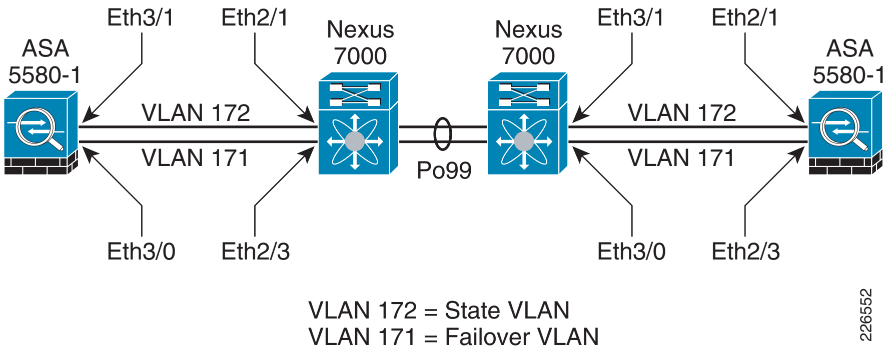
- ASA
- Services Chassis
- Application Control Engine Services Module
- WAF
- NetFlow Services
- ERSPAN

A major advantage to the services layer is the ability to introduce new services in a controlled manner using predictable traffic patterns. The services highlighted in this section are not all encompassing.

ASA

All of the ASA virtual contexts in this design are configured between a pair of physical devices. Each context was deployed in transparent mode. Figure 26 depicts the physical connectivity of the ASA 5580 appliances to the Nexus 7000 aggregation layer switches. The state and failover interfaces are Gigabit attached to the Nexus switches. The ASA failover and state VLANs are trunked across port channel 99, the ISL.

Figure 26 ASA Failover Topology



The following is the failover configuration for the ASA platforms. Of primary importance is the use of two failover groups, 1 and 2, to create active traffic patterns on each physical ASA device. The virtual contexts are subsequently placed in alternating failover groups to split flows across the infrastructure.

```
interface GigabitEthernet3/0
 description LAN Failover Interface
!
interface GigabitEthernet3/1
 description STATE Failover Interface
!
failover
failover lan unit primary
failover lan interface failover GigabitEthernet3/0
failover polltime unit 1 holdtime 5
failover key *****
failover replication http
failover link state GigabitEthernet3/1
failover interface ip failover 10.8.171.1 255.255.255.0 standby 10.8.171.2
failover interface ip state 10.8.172.1 255.255.255.0 standby 10.8.172.2
failover group 1
  preempt
failover group 2
  secondary
  preempt 1
context dca-vc1
  allocate-interface Management0/0.1
  allocate-interface TenGigabitEthernet5/0.161 outside
  allocate-interface TenGigabitEthernet5/1.162 inside
  config-url disk0:/dca-vc1.cfg
  join-failover-group 1
!
context dca-vc2
  allocate-interface Management0/0.2
  allocate-interface TenGigabitEthernet7/0.151 outside
  allocate-interface TenGigabitEthernet7/1.152 inside
  config-url disk0:/t
  join-failover-group 2
```

```
!
```

Virtual Context Example—vc1

The ASA virtual context bridges traffic between VLANs 161 and 162. The "System" context allows one to create the proper trunking parameters for the virtual context; for example, VLAN 161 and 162 are assigned to the physical interfaces connected to a Nexus platform.

```
interface TenGigabitEthernet5/0.161
 vlan 161
!
interface TenGigabitEthernet5/1.162
 vlan 162
!
```

From the virtual contexts, the interfaces are referenced by name. In the previous section, the *outside* and *inside* interface names are assigned to the trunk ports. The transparent virtual context also assigns security levels to the interfaces.

```
interface outside
 nameif north
 security-level 100
!
interface inside
 nameif south
 security-level 0
!
```

The firewall context uses an IP address within the same subnet range as the Nexus 7000 **vdc1** VLAN interface 161, indicating the traffic is truly being bridged at Layer 2 between VLANs.

```
ip address 10.8.162.10 255.255.255.0 standby 10.8.162.11
```

The virtual context is defined as transparent with the following command:

firewall transparent

For more information on ASA security configuration, refer to the configuration guides at the following URL:

http://www.cisco.com/en/US/docs/security/asa/asa81/config/guide/config.html

Services Chassis

The services chassis is a platform for integrated network services and independent appliances. In this example, the services chassis supports an integrated ACE and NAM modules, in addition to ACE WAF and IPS appliances. These services use the underlying Layer 2 configuration. The Service Chassis is designated as root for those services considered active. Aligning services with the Layer 2 and Layer 3 structures provides for optimal traffic flows under normal operating conditions. The traffic patterns become well known and allow for predictable, reliable service insertion.

In Figure 25, the services chassis is root for VLAN 163 and therefore root for 161,162, and 164. The configuration is as follows:

```
spanning-tree mode rapid-pvst
spanning-tree pathcost method long
spanning-tree vlan 163,170-172,3001 priority 24576
```

To provide access for the various service modules on the platform, assign VLANs to each module. In this example, the ACE uses **autostate** to monitor interface-availability on the services chassis.

```
svclc autostate
! Allow for multiple VLANs to be accessible by the service modules
svclc multiple-vlan-interfaces
! Module 8 is the ACE
svclc module 8 vlan-group 1,2,150,160,190
! Management VLAN group
svclc vlan-group 1  146
! ACE Failover VLAN group
svclc vlan-group 2  170
! Bridged VLAN group for standby ACE virtual context
svclc vlan-group 150  152,153
! Bridged VLAN group for active ACE virtual context
svclc vlan-group 160  162,163
! WAF VLAN group for active ACE virtual context
svclc vlan-group 190  190
! Module 9 is the NAM, 146 is the Mgmt VLAN group
analysis module 9 management-port access-vlan 146
```

By following the STP path, note that the root for VLAN 162 is through the ACE virtual context defined on module 8 of the services chassis.

```
show spanning-tree vlan 162
VLAN0162
  Spanning tree enabled protocol rstp
  Root ID    Priority    24739
             Address     0021.d72a.c000
             Cost        2000
             Port        897 (TenGigabitEthernet8/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32930  (priority 32768 sys-id-ext 162)
             Address     0021.d72a.c000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 480
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Te1/1               Desg FWD 2000      128.1    P2p Network
Te1/2               Desg FWD 2000      128.2    P2p Network
Te8/1               Root FWD 2000      128.897  P2p
```

The ACE module performs its application optimization functions (see below) and bridges the traffic to VLAN 163 where the inline IPS devices reside off of port channel 2. Port channel 2 is a static port channel to a series of IPS 4270 devices using VLAN pairing interfaces. Essentially, the IPS is transparently bridging VLANs 163 and 164.

```
interface Port-channel2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 163,164
 switchport mode trunk
 switchport nonegotiate
 mtu 9216
end
```

By consistently applying a source and destination IP address load-balancing hashing scheme, port channel 2 will allow for flow persistence to a specific IPS used by the channel.

```
show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip enhanced
```

The **show spanning-tree** command verifies the bridging capabilities of the IPS channel as the root priority matches that of VLAN 163.

```
show spanning-tree vlan 164
```

```
VLAN0164
  Spanning tree enabled protocol rstp
  Root ID    Priority    24739
             Address     0021.d72a.c000
             Cost        2000
             Port        1665 (Port-channel2)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32932  (priority 32768 sys-id-ext 164)
             Address     0021.d72a.c000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 480
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Te1/1               Desg FWD 2000      128.1    P2p Network
Te1/2               Desg FWD 2000      128.2    P2p Network
Po2                 Root FWD 2000      128.1665 P2p Network
```

Note For more information on using IPS/IDS devices in the data center please go to “David’s Paper”.

Application Control Engine Services Module

This section discusses the necessary configurations to enable the active-active ACE service module design. In addition, the ACE feature-specific implementations are described for the example virtual ACE context shown in Figure 25.

Admin Context—Failover Configuration

Define the VLAN interface used for the state and ACE configuration synchronization. In this solution, the ISL between services chassis containing the ACE service modules supports VLAN 170. The local and remote peering ACE have an IP address on this subnet.

```
ft interface vlan 170
  ip address 10.8.170.1 255.255.255.0
  peer ip address 10.8.170.2 255.255.255.0
  no shutdown
```

Establish the failover peer communication parameters as follows:

```
ft peer 1
  heartbeat interval 100
  heartbeat count 10
  ft-interface vlan 170
```

Define two fault-tolerant groups, in this case 2 and 3, where the ACE virtual context are assigned. The fault-tolerant groups will only be active on one ACE service-module at a time. This active peer has the higher priority setting. The virtual contexts **vc1** and **vc2** are assigned to different fault-tolerant groups 2 and 3 respectively with alternating active peers.

```
ft group 1
  peer 1
  priority 150
  peer priority 50
  associate-context vc1
  inservice
ft group 2
  peer 1
  priority 50
  peer priority 150
  associate-context vc2
  inservice
```

The **show ft group brief** command verifies the active-active deployment of the ACE virtual contexts across the fault-tolerant peer groups.

```
show ft group brief
FT Group ID: 1  My State:FSM_FT_STATE_ACTIVE
Peer State:FSM_FT_STATE_STANDBY_COLD
                Context Name: vc1       Context Id: 1
FT Group ID: 2  My State:FSM_FT_STATE_STANDBY_HOT
Peer State:FSM_FT_STATE_ACTIVE
                Context Name: vc2       Context Id: 2
```

Example Virtual Context—Client Side

```
interface vlan 162
  description ** North Side facing N7K-1 vdc2 **
  bridge-group 161
  no normalization
  no icmp-guard
  access-group input BPDU
  access-group input ALLOWED_TRAFFIC
  service-policy input aggregate-slb-policy
  no shutdown
```

The ACE virtual context enforces access controls via ACLs. By default, ACE interfaces deny all traffic forcing network and security administrators to define those traffic patterns that are acceptable. In the above configuration, the access group **ALLOWED_TRAFFIC** defines acceptable IP traffic but perhaps even more significant is the use of the BPDU access group. The BPDU group permits the forwarding of BPDUs across the transparent ACE virtual context. This allows STP to operate properly in the service layer domain. This is the access list required to allow the flow of BPDUs across the ACE interface, the same access group must be applied to the server-facing VLAN interface.

```
access-list BPDU ethertype permit bpdu
```

The following class maps define the Virtual IP address available on the ACE virtual context. Note that the IP address are within the save Layer 2 subnet as the 161 VLAN interface on VDC1, which has already been bridged to VLAN 162 across the ASA virtual context. Each of these VIPs is listening on a different port, 443 and 80 respectively, but these could actually be any number of ports.

```
class-map match-all L4_HTTPS_VIP_ADDRESS
  2 match virtual-address 10.8.162.200 tcp eq https
class-map match-all L4_HTTP_VIP_ADDRESS
  2 match virtual-address 10.8.162.200 tcp eq www
```

The client interface has a single ingress service policy applied to interesting traffic, meaning the traffic must meet the access requirements and then the class definitions for the policy for additional ACE features to be applied. Traffic that does no match the service policy but is permissible according to the access rules will simply bridged between the client and server (VLANs 162 and 163) interfaces. In the following sample configuration, the **aggregate-slb-policy** command was applied to VLAN 162 during testing. Note that both encrypted and non-encrypted traffic are subject to a policy map named **pm-waf**.

```
policy-map multi-match aggregate-slb-policy
 class L4_HTTP_VIP_ADDRESS
    loadbalance vip inservice
    loadbalance policy pm-waf
    loadbalance vip icmp-reply
```

```
  class L4_HTTPS_VIP_ADDRESS
    loadbalance vip inservice
    loadbalance policy pm-waf
    loadbalance vip icmp-reply
    ssl-proxy server SSL_PSERVICE_CRACKME
```

Note **L4_HTTPS_VIP_ADDRESS** uses an SSL proxy service that is defined in the "Example Virtual Context—SSL Termination" section on page 46.

The **pm-waf** policy map references a sticky serverfarm, **wafstkygrp**, which inserts HTTP cookies to maintain flow persistence. The individual WAF cluster nodes are defined in a serverfarm named **sf_waf** used by the **wafstkygrp** sticky group. In this example, cookie-based persistence is used, but the ACE module supports sticky-based on IP, Layer 4 payload, HTTP headers, HTTP content in addition to SSL session-id persistence and others.

```
policy-map type loadbalance http first-match pm-waf
  class class-default
    sticky-serverfarm wafstkygrp
    insert-http ACEForwarded header-value "%is"
```

Note The **insert-http** command retains the client source IP address in the HTTP header sent to the server and is available for logging purposes.

The sticky group definition references the WAF cluster nodes and replicates the sticky table to the standby ACE virtual context through the fault-tolerant VLAN 170 defined within the admin context. Sticky replication improves the resiliency of the solution and user experience.

```
sticky http-cookie wafcookie wafstkygrp
  cookie insert
  replicate sticky
  serverfarm sf_waf
```

Example Virtual Context—WAF Side

The ACE WAF cluster is positioned adjacent to the bridged traffic flows crossing the ACE fabric. This one-arm deployment model allows the ACE to redirect only traffic to the WAF farm that is interesting namely, HTTP-based flows. In this example, the cluster resides on the 190 VLAN where the ACE interface defines a service policy named **L4_LB_VIP_HTTP_POLICY**. Note that access rules are required but there is not a BVI.

```
interface vlan 190
  description ** WAF Cluster Interface **
  ip address 10.8.190.2 255.255.255.0
  alias 10.8.190.1 255.255.255.0
  peer ip address 10.8.190.3 255.255.255.0
  no normalization
  no icmp-guard
  access-group input ALLOW_TRAFFIC
  service-policy input L4_LB_VIP_HTTP_POLICY
  no shutdown
```

The **policy-map** is applied to ingress traffic from the WAF cluster, meaning the VIPs assigned to this map will correspond to the destinations defined on the WAF cluster. Note that the same VIP or class, **L4_HTTP_VIP_ADDRESS**, is employed saving configuration space while simultaneously aligning the client and WAF VLAN service policies, making the flow from client, to WAF node, and to real server easier to follow. The **pm-webbank** policy defines a sticky serverfarm referencing the real servers.

```
policy-map multi-match L4_LB_VIP_HTTP_POLICY
  class L4_HTTP_VIP_ADDRESS
    loadbalance vip inservice
    loadbalance policy pm-webbank
    loadbalance vip icmp-reply

policy-map type loadbalance first-match pm-webbank
  class class-default
    sticky-serverfarm bnkstygrp

sticky http-cookie bankcookie bnkstygrp
  cookie insert
  replicate sticky
  serverfarm sf_bank
```

The **sf_bank** serverfarm is used by the **bnkstygrp** sticky serverfarm. Note the use of the **UBER** probe to monitor the health of these servers. The following section on high availability details the use of probes by the ACE virtual context to not only monitor real servers but the WAF cluster nodes as well.

```
serverfarm host sf_bank
  probe UBER
  rserver tbox1 8081
    inservice
  rserver uber0 8081
    inservice
  rserver uber1 8081
  rserver uber2 8081
    inservice
  rserver uber3 8081
    inservice
  rserver uber4 8081
    inservice
  rserver uber5 8081
    inservice
```

Example Virtual Context—Server Side

The server-side VLAN interface is the other member of the bridge group consisting of itself and the client-side VLAN interface 162. The server interface applies similar access controls lists, but this is not a requirement but often a reality of symmetric traffic flows. Note the BPDU access groups is present to allow spanning tree communications. In this example, the use of a service policy on the server interface was not required.

```
interface vlan 163
  description ** South Side facing Servers **
  bridge-group 161
  no normalization
  no icmp-guard
  access-group input BPDU
  access-group input ALLOW_TRAFFIC
  no shutdown
```

The definition of the BVI used by the client and server VLANs (162 and 163) describes the Layer 3 attributes of the interface. Note that the subnet 10.8.162.x/24 is also used by the BVI. The alias IP address is a virtual address shared between the active ACE virtual context and the standby peer.

```
interface bvi 161
  ip address 10.8.162.20 255.255.255.0
  alias 10.8.162.22 255.255.255.0
  peer ip address 10.8.162.21 255.255.255.0
  no shutdown
```

The routes for the ACE virtual context only apply to traffic originating from the device, for the most part probes. As the serverfarms are not Layer-2 adjacent to the ACE network administrators should configure static routes to those subnets. In this example, the 10.8.180.0 subnet contains a serverfarm probed by the ACE. The route directs traffic to the HSRP standby address of **vdc2 vrf1**, 10.8.162.7. This VRF terminates the Layer-2 service domain on the Nexus aggregation switches. Only one default route is required to direct traffic to the "norther" VDC in the Nexus **vdc1**.

```
ip route 0.0.0.0 0.0.0.0 10.8.162.1
! Route to server VLAN 180 on vdc2 vrf1
ip route 10.8.180.0 255.255.255.0 10.8.162.7
```

Example Virtual Context—Fault Tolerance and High Availability

Object tracking allows the ACE virtual context to determine its status based on the status of objects external to the ACE. The active-active services chassis design tracked the bridge group VLAN interfaces. In the example below, VLAN 163 is the server facing or "southern" interface on the ACE context. As shown by the priority setting, the failure of the VLAN on the services chassis would result in this ACE context failing over to its peer device.

Note Interface tracking permits the ACE context to act on autostate messages received from the services chassis supervisor engine.

```
ft track interface  TrackVlan163
  track-interface vlan 163
  peer track-interface vlan 163
  priority 150
  peer priority 50
```

As mentioned earlier, the ACE is capable of monitoring the health of both real servers and network-based services. In this solution, the ACE virtual context monitors the state of the application servers using an HTTP-based probe. This probe is only successful if the probe receives a 200 HTTP status response from the server.

```
probe http UBER
  port 8081
  interval 2
  passdetect interval 5
  request method get url /Kelev/view/home.php
  expect status 200 200
```

The virtual context also needs to verify that the WAF cluster nodes are not only available but that they are capable of reaching the "real" destinations for which they are supplying security and proxy services. Therefore, the probe as configured to force a full-session down to the web servers, ironically the same web servers the ACE is monitoring with the previous prove.

```
probe http CRACKME
  port 81
  interval 2
  passdetect interval 5
  request method get url /Kelev/view/home.php
  expect status 200 200
```

Note These probes could have been combined into one probe instance but for demonstration purposes two distinct probes are used.

Example Virtual Context—SSL Termination

The ACE virtual context in this solution topology acts as an SSL-termination point, providing secure transport from the client. The ACE can create a Certificate Signing Request (CSR) for approval by the enterprises Certificate Authority (CA). The following is the CSR parameters defined for the virtual context.

```
crypto csr-params CSR_PARAMS_1
  country US
  state North Carolina
  locality RTP
  organization-name ESE
  organization-unit BANK VAULT
  common-name crackme.com
```

The **ssl-proxy** defines the termination service; in this case it is named **SSL_PSERVICE_CRACKME**. This proxy will be assigned to a service policy and associated with VIP. The proxy references the key and certificate that were used to generate the CSR and the certificate assigned by the CA.

```
ssl-proxy service SSL_PSERVICE_CRACKME
  key my2048RSAkey.PEM
  cert crackme-cert.pem
```

The client-side VLAN 162 references the SSL proxy under the HTTPS VIP assigned through the **aggregate-slb-policy**.

```
policy-map multi-match aggregate-slb-policy
class L4_HTTPS_VIP_ADDRESS
    loadbalance vip inservice
    loadbalance policy pm-waf
    loadbalance vip icmp-reply
    ssl-proxy server SSL_PSERVICE_CRACKME
```

The ACE module may also act as an SSL client; this will be explored in future documentation.

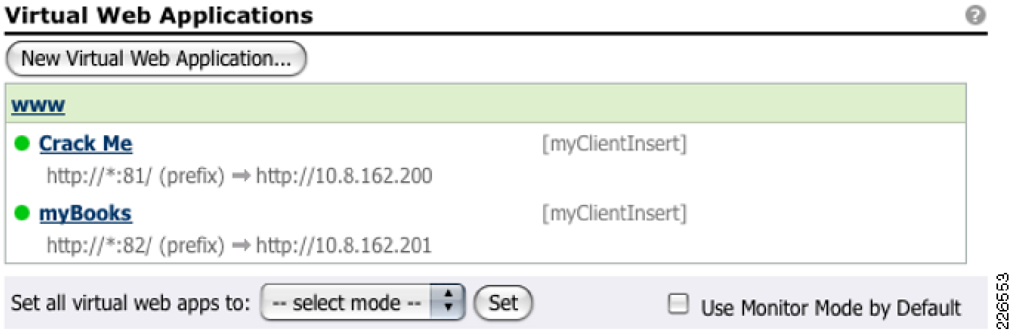
WAF

The WAF cluster is positioned adjacent to the ACE service module. The ACE leverages the WAF as a serverfarm, redirecting traffic associated with a VIP defined on the ACE. As a cluster, the WAF farm is easily configured from a centralized management node. The management node pushes policies to the cluster node, while the ACE service module can monitor the actual health of the WAF cluster farm.

To readily scale the WAF cluster the solution uses port-based listeners, not URL-based, meaning we are expecting to proxy and secure web-application traffic on ports 81 and 82. Figure 28 is a snapshot of two web applications secured by the WAF cluster leveraging ACE services. Remember the ACE virtual context would forward traffic to the IP addresses of the WAF cluster nodes on these ports. The cluster node, listening on the interesting port, would secure the traffic and proxy the request. In this example the request is forwarded to another set of VIPs located on the ACE interface VLAN 190. In Figure 27, the ACE VIPs 10.8.162.200 and 10.8.162.201 are referenced by their respective virtual web applications named **Crack Me** and **myBooks**.

Note The WAF is capable of performing URL-based listening.

Figure 27 Sample Virtual Web Applications



Virtual Web Applications

New Virtual Web Application...

www

● Crack Me [myClientInsert]
http://*:81/ (prefix) → http://10.8.162.200

● myBooks [myClientInsert]
http://*:82/ (prefix) → http://10.8.162.201

Set all virtual web apps to: -- select mode -- Set

□ Use Monitor Mode by Default

Note Beyond initial CLI configuration, most of the WAF cluster configuration is performed through a secure web interface. For guidance, it is recommended to review the configuration guides available at the following URL: http://www.cisco.com/en/US/products/ps9586/products_installation_and_configuration_guides_list.html

NetFlow Services

NetFlow allows for in-depth traffic analysis. In the tested solution, the Cisco Nexus 1000v is positioned as a NetFlow exporter, while the Cisco NetFlow Collector is positioned in an out-of-band management network as a NetFlow collector. The default export interface for the Nexus 1000v is the management 0 interface.

Define the NetFlow collector, for example 10.x.x.x, in the Nexus 1000v export configuration. The collector's listening port, in this case 3000, must also be defined.

```
flow exporter lnxnf
  description Cisco Netflow Collector
  destination 10.x.x.x
  transport udp 3000
  source mgmt0
  version 9
    template data timeout 300
    option exporter-stats timeout 120
```

Defining a custom flow record is also possible, but in this example the default "netflow-original" record is being used. The Nexus 1000v supports NetFlow v9.

```
flow monitor ESE-flow
  description Flow to Collector
  record netflow-original
  exporter lnxnf
  timeout active 1800
  cache size 4096
```

Finally, the NetFlow should be enabled at a port group level. In this example, port profile VM 180 uses the previously defined NetFlow record definition. The VM 180 port group is actually the vNIC access ports for VMs residing in VLAN 180.

```
port-profile vm180
  vmware port-group pg180
  switchport mode access
  switchport access vlan 180
  ip flow monitor ESE-flow input
  ip flow monitor ESE-flow output
  no shutdown
  state enabled
```

Note Use the **show flow** commands to verify the NetFlow configuration on the Nexus 1000v.

ERSPAN

ERSPAN allows for remote monitoring of system devices. In the tested solution, the services chassis is a destination for Nexus 1000v ERSPAN traffic. Figure 19 illustrates this flow from the access layer through the aggregation layer to the services layer. The following examples configurations are derived from this topology.

Services Switch Example

To configure this model, use the following configurations on the Services switch. Allow access to the management VLAN of the NAM.

```
! Module 9 is the NAM, 146 is the Mgmt VLAN group
analysis module 9 management-port access-vlan 146
```

Create a Layer 3 destination for the ERSPAN traffic. In this example, SVI 3001 serves as an ERSPAN interface. Note that MTU setting is large to allow for the additional headers of the GRE ERSPAN tunnel. This jumbo frame setting should be standardized across the tunnels path to optimize TCP flows.

```
interface Vlan3001
 description ERSPAN Source Interface
 mtu 9216
 ip address 10.8.33.4 255.255.255.0
 load-interval 30
!
```

Create a monitor session that references the ERSPAN interface as a source. The ERSPAN interface can support monitored traffic from multiple sources; the ERSPAN ID field identifies each one allowing the traffic to be directed to various destinations. In this case, the ERSPAN source from an ESX server using Nexus 1000v DVS (see below) is directed to the NAM service module located in slot 9 of the services chassis.

```
monitor session 1 type erspan-source
 description <** N1k ERSPAN - originating from dcesx4n1 monitor session 1 **>>
 source vlan 3001
 destination
  erspan-id 1
  ip address 10.8.33.4
!
monitor session 3 type erspan-destination
 description <** N1k ERSPAN to NAM - originating from dcesx4n1  **>>
 destination analysis-module 9 data-port 2
 source
  erspan-id 1
  ip address 10.8.33.4
!
```

The same SVI supports another ERSPAN session, ID 2, whose flows will be directed to a virtual IDS sensor, allowing remote intrusion detection of a VM at the services layer. In this example, the virtual IDS sensor is connected through Gigabit Ethernet interface 3/26.

```
monitor session 2 type erspan-source
 description <** N1k ERSPAN - originating from dcesx4n1 monitor session 2 **>>
 source vlan 3001
 destination
  erspan-id 2
  ip address 10.8.33.4
!
monitor session 4 type erspan-destination
 description <** N1k ERSPAN to IDS-1 - originating from dcesx4n1  **>>
 destination interface Gi3/26
 source
  erspan-id 2
  ip address 10.8.33.4
!
```

Nexus 1000v Example

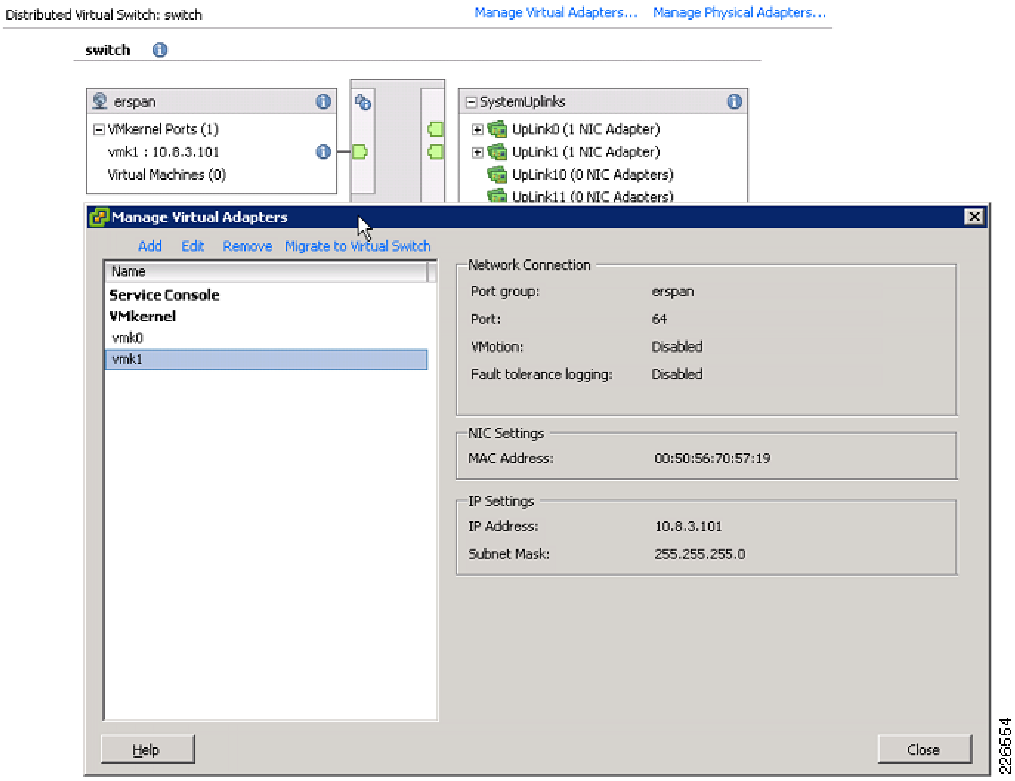
The Nexus 1000v supports ERSPAN. The ERSPAN source traffic originating from the vNIC interfaces of the VMs is destined for both intrusion detection and network analysis devices (see above) supported by the services layer. Figure 19 illustrates the use of the ERSPAN ID to direct traffic appropriately.

To configure ERSPAN on the Nexus 1000v, define a port profile with Layer 3 functionality. A VMkernel NIC on each ESX host will reference this port group to establish the ERSPAN GRE tunnel with the remote destination switch.

```
port-profile erspan
  capability l3control
  vmware port-group
  switchport access vlan 3000
  no shutdown
  system vlan 3000
  state enabled
```

Figure 28 depicts the configuration from VMware’s vSphere client. Essentially, the Nexus 1000v port group on this ESX host is assigned an IP address of 10.8.3.101. This will be a source of ERSPAN traffic from this ESX host. Note that the **erspan** port group appears as a VMkernel port associated with the Nexus 1000v DVS.

Figure 28 VMkernel ERSPAN NIC Example


Manage Virtual Adapters...
Manage Physical Adapters...
Manage Virtual Adapters
Add
Edit
Remove
Migrate to Virtual Switch
Name
Help
Close
226554

Distributed Virtual Switch: switch

switch

erspan
VMkernel Ports (1)
vmk1 : 10.8.3.101
Virtual Machines (0)

SystemUplinks
UpLink0 (1 NIC Adapter)
UpLink1 (1 NIC Adapter)
UpLink10 (0 NIC Adapters)
UpLink11 (0 NIC Adapters)

Service Console
VMkernel
vmk0
vmk1

Network Connection
Port group: erspan
Port: 64
VMotion: Disabled
Fault tolerance logging: Disabled

NIC Settings
MAC Address: 00:50:56:70:57:19

IP Settings
IP Address: 10.8.3.101
Subnet Mask: 255.255.255.0

The **show interface vethernet** command verifies successful communication between VMware's vSphere infrastructure and the Nexus 1000v. The following example shows the newly created VMkernel adapter from the Nexus 1000v perspective as virtual Ethernet interface. Note that the module, or VEM, is indicated.

```
show interface vethernet 3
Vethernet3 is up
    Hardware is Virtual, address is 0050.5670.5719
    Owner is VMware VMkernel, adapter is vmk1
    Active on module 3
    VMware DVS port 64
    Port-Profile is erspan
    Port mode is access
    Rx
    48126340 Input Packets 48125182 Unicast Packets
    6 Multicast Packets 1152 Broadcast Packets
    18129098526 Bytes
    Tx
    2347778 Output Packets 1164 Unicast Packets
    2343150 Multicast Packets 3464 Broadcast Packets 2346614 Flood Packets
    145553584 Bytes
    0 Input Packet Drops 0 Output Packet Drops
```

The port group is assigned to the VMkernel NICs which are visible as virtual Ethernet interfaces on the Nexus 1000v. In the example below, three virtual Ethernet interfaces are enabled under the **erspan** port profile. These three interfaces are active on three different VEMs or ESX hosts that comprise the Nexus 1000v DVS. Use the **show port-profile** command to verify the configuration.

```
show port-profile name erspan
port-profile erspan
```

```
  description:
  status: enabled
  capability uplink: no
  capability l3control: yes
  system vlans: 3000
  port-group: erspan
  max-ports: 32
  inherit:
  config attributes:
    switchport access vlan 3000
    no shutdown
  evaluated config attributes:
    switchport access vlan 3000
    no shutdown
  assigned interfaces:
    Vethernet3
    Vethernet5
    Vethernet7
```

Create a monitor session of **type erspan-source** on the Nexus 1000v DVS. Below are two different ERSPAN sessions that are destined for the same services chassis SVI (10.8.33.4) but use distinct ERSPAN IDs to differentiate the traffic source at the destination.

```
monitor session 1 type erspan-source
  description - to SS1 NAM via VLAN 3000
  source interface Vethernet1 both
  source interface Vethernet8 both
  …
  source interface Vethernet13 both
  destination ip 10.8.33.4
  erspan-id 1
  ip ttl 64
  ip prec 0
  ip dscp 0
  mtu 1500
  no shut

monitor session 2 type erspan-source
  description - to SS1 IDS1 via VLAN 3000
  source interface Vethernet1 both
  source interface Vethernet8 both
…
  source interface Vethernet13 both
  destination ip 10.8.33.4
  erspan-id 2
  ip ttl 64
  ip prec 0
  ip dscp 0
  mtu 1500
  no shut
```

ESX VMkernel routes should be verified on each ESX host using the **esxcfg-route** and **vmkping** commands. Below is an example output from the ESX command line that validates the route to the ERSPAN destination services chassis.

```
#esxcfg-route -l
VMkernel Routes:
Network           Netmask           Gateway
10.8.3.0          255.255.255.0     Local Subnet
10.8.15.0         255.255.255.0     Local Subnet
10.8.33.0         255.255.255.0     10.8.3.1
default           0.0.0.0           10.8.15.1
```

```
#vmkping 10.8.33.4
PING 10.8.33.4 (10.8.33.4): 56 data bytes
64 bytes from 10.8.33.4: icmp_seq=0 ttl=254 time=0.777 ms
64 bytes from 10.8.33.4: icmp_seq=1 ttl=254 time=0.482 ms
64 bytes from 10.8.33.4: icmp_seq=2 ttl=254 time=0.734 ms

--- 10.8.33.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.482/0.664/0.777 ms
```

Access Layer

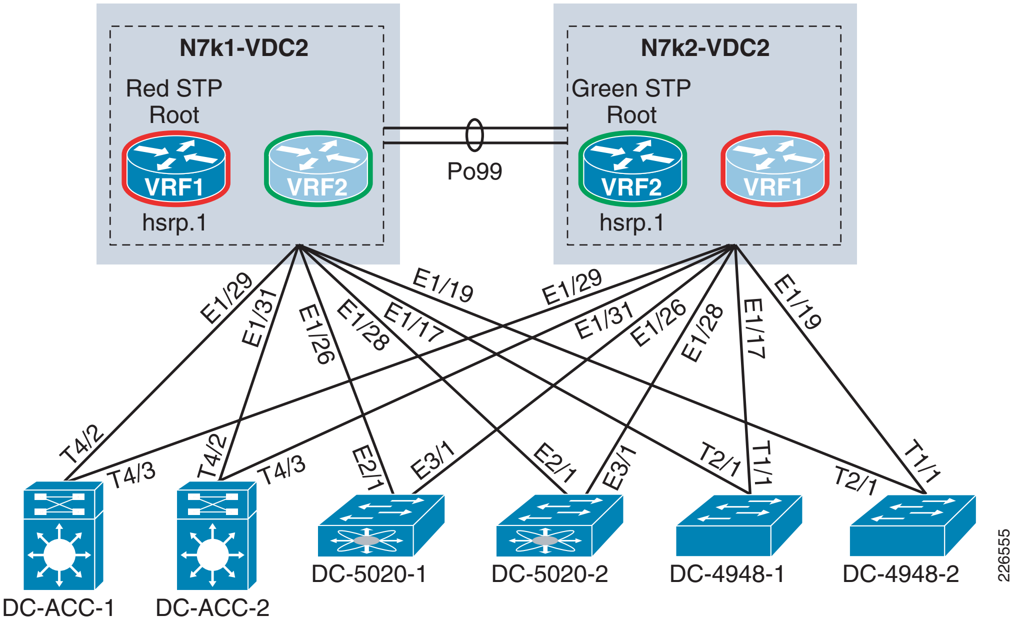
The access layer provides endpoint connectivity to the greater enterprise data center. The following section highlights the uplink and edge port configurations used on the numerous switching platforms validated with this design including:

- Catalyst 4900M
- Catalyst 6500 (including Virtual Switching System)
- Nexus 5000
- Catalyst 3120 (including Virtual Blade Switches)
- Nexus 1000v

Figure 29 can be used to reference the Nexus 7000 uplink ports for the Catalyst 6500, 4900M, and Nexus 5000 switches. Each of these switches used 10-Gigabit Ethernet dot1q trunks in the traditional “triangle” looped topology. ISL **Po99** completes the Layer 2 loop for all of the VLANs. All switches employ RPVST+ to address the physical redundancy in the topology. The path cost calculation method is “long” to optimize convergence time and Root Guard configured on the Nexus 7000 uplink ports. Bridge assurance is enabled by default where possible and Loopguard is enabled on those switches that do not currently use IOS images supporting bridge assurance.

Note Nexus 7000 virtual port channeling (vPC) was not available at the time of this solution testing.

Figure 29 Access Layer Uplinks to Nexus 7000 for Catalyst 6500, Nexus 5000 and Catalyst 4900M Platforms



Catalyst 4900M

Uplink Ports to Nexus 7000

The 4900M uplink ports support any or all VLANs associated with either VRF located on the Nexus 7000 VDC. Loopguard is enabled on the uplink ports.

```
spanning-tree mode rapid-pvst
spanning-tree pathcost method long

interface TenGigabitEthernet1/1
 switchport trunk allowed vlan 50,51,142
 switchport mode trunk
 spanning-tree guard loop
end
```

The 4900M IOS code tested in this solution does not currently support the bridge assurance feature. As a result, the corresponding Nexus 7000 port disables bridge assurance with the **spanning-tree port type normal** command. Root Guard is enabled on all interfaces to harden the STP domain.

```
interface Ethernet1/17
  description dc20-4948-1
  switchport mode trunk
  switchport trunk allowed vlan 50-51,142
  spanning-tree port type normal
  spanning-tree guard root
```

Edge Ports

The 4900M access switches supported the following devices:

- Cisco Unity
- CSA Management Center

- Networked Attached Storage (NAS) Device

The endpoint ports were configured as access ports to support the Cisco Unity and CSA Management Center in the data center. This is an example of one of these access ports that is defined as an edge port with BPDU Guard enabled:

```
interface GigabitEthernet3/19
 switchport access vlan 50
 switchport mode access
 spanning-tree portfast
 spanning-tree bpduguard enable
!
```

Catalyst 6500

Uplink Ports to Nexus 7000

The Catalyst 6500 uplink ports support any or all VLANs associated with either VRF located on the Nexus 7000 VDC. The Catalyst access switches are using Loopguard on the uplinks, as the IOS version for these switches did not support bridge assurance. If the IOS image supports bridge assurance, enable it globally with the **spanning-tree portfast network default** command; after doing so, all edge ports must be explicitly configured as such.

```
spanning-tree mode rapid-pvst
spanning-tree pathcost method long
interface TenGigabitEthernet4/3
 description ** to Nexus 7000 **
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 128-133,164-169,180-183,300-399
 switchport mode trunk
 spanning-tree guard loop
end
```

The Nexus 7000 trunk port configuration is as follows:

```
interface Ethernet1/29
  description to 6k access
  switchport mode trunk
  switchport trunk allowed vlan 128-133,164-169,180-183,300-399
  spanning-tree port type normal
  spanning-tree guard root
end
```

Edge Ports

The following is an example access port configuration on the Catalyst 6500.

```
interface GigabitEthernet1/7
 switchport
 switchport access vlan 183
 switchport mode access
 spanning-tree portfast
 spanning-tree bpduguard enable
```

Nexus 5000

Uplink Ports to Nexus 7000

The Nexus 5000 uplink trunk ports use bridge assurance.

```
spanning-tree pathcost method long
spanning-tree port type network default
interface Ethernet2/1
  description ** to Nexus 7000 **
  switchport mode trunk
  switchport trunk allowed vlan 15,98,142,180-183,3000,3002-3003
  spanning-tree port type network
```

Edge Ports

The ESX server facing trunk ports on the Nexus 5000 do not support bridge assurance so these are configured as type **edge**. Note that in this example, the VEM management and control packets VLANs (3002, 3003) are supported in band, as well as, the VMotion VLAN (15).

```
interface Ethernet1/1
  switchport mode trunk
  logging event port link-status
  logging event port trunk-status
  switchport trunk allowed vlan 15,98,142,180-183,3000,3002-3003
  spanning-tree port type edge trunk
```

Note The use of trunks to the server has traditionally been “frowned” upon, but with improved hardware performance and the cost savings realized via consolidation through virtualization trunking to the server has become a more generally accepted practice.

Nexus 1000v

Figure 30 illustrates the logical connectivity of the Nexus 1000v Virtual Ethernet Modules (VEMs) to the Nexus 5000 access layer. The Nexus 1000v was tested using Asymmetric Port Channeling (APC). APC allows the aggregation of local VEM ports to form a channel to distinct Cisco Discovery Protocol (CDP)-enabled physical switches. APC uses source-based MAC hash to load balance traffic originating from the virtual machines connected to each VEM across the uplinks. This hash prevents any MAC address conflict conditions from occurring at the Nexus 5000 access layer. The Nexus 5000 switches are unaware of the port aggregation occurring at each VEM and simply recognize the ports as trunks.

Note APC requires that the upstream switches support CDP.

Figure 30 Nexus 1000v Connectivity

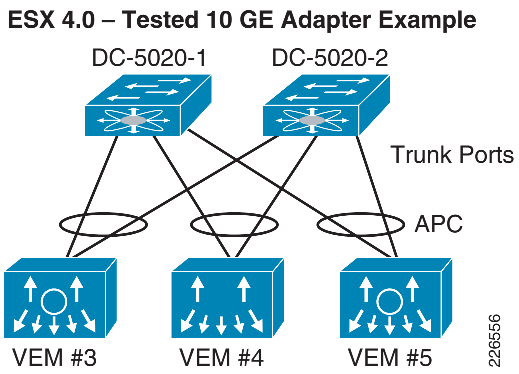


Figure 30 shows three VEMs, or three ESX 4.0 Nexus 1000v-enabled hosts. The Nexus 1000v confirms this using the **show module** command. Module 1 is the Nexus 1000v supervisor.

```
dcvsm# show module
Mod  Ports  Module-Type                      Model              Status
---  -----  -------------------------------- ------------------ ------------
1    0      Virtual Supervisor Module        Nexus1000V         active *
3    248    Virtual Ethernet Module                             ok
4    248    Virtual Ethernet Module                             ok
5    248    Virtual Ethernet Module                             ok
```

The connectivity to the Nexus 5000 access layer can be verified with the **show cdp neighbor** command. Each ESX host, or VEM instance, is dual-homed to the access layer. The "Local Interface" field indicates the Ethernet interface VEM or slot number and port.

```
dcvsm# show cdp neighbor
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute


Device ID              Local Intrfce    Hldtme  Capability  Platform      Port ID
dc10-5020-1(FLC12270066)Eth3/3           157     S I s       N5K-C5020P-BF Eth1/5
dc10-5020-2(FLC12320055)Eth3/4           130     S I s       N5K-C5020P-BF Eth1/5
dc10-5020-2(FLC12320055)Eth4/3           130     S I s       N5K-C5020P-BF Eth1/1
dc10-5020-1(FLC12270066)Eth4/4           157     S I s       N5K-C5020P-BF Eth1/1
dc10-5020-2(FLC12320055)Eth5/3           130     S I s       N5K-C5020P-BF Eth1/2
dc10-5020-1(FLC12270066)Eth5/4           148     S I s       N5K-C5020P-BF Eth1/2
```

Uplink Ports to Nexus 5000

The Nexus 1000v uplink port profile is created with the same VLANs as defined on the Nexus 5000 switches. The "capability uplink" permits the profile to be assigned to physical interfaces on the ESX host. The APC configuration requires that the **channel-group auto** command reference CDP gleaned information to manage the traffic flows across the two different Nexus 5000 switches.

```
port-profile n1kuplinks
  capability uplink
  vmware port-group SystemUplinks
  switchport mode trunk
  switchport trunk allowed vlan 15,98,180-183,3000,3002-3003
  channel-group auto mode on sub-group cdp
  no shutdown
  system vlan 3002-3003
  state enabled
```

The port-profile is subsequently applied to the physical interfaces of the VEMs. Note that there is a different port channel formed for each VEM module. Port channels can not be formed across VEMs. In this example, three distinct port channels are created, across three VEMs and their associated physical interfaces.

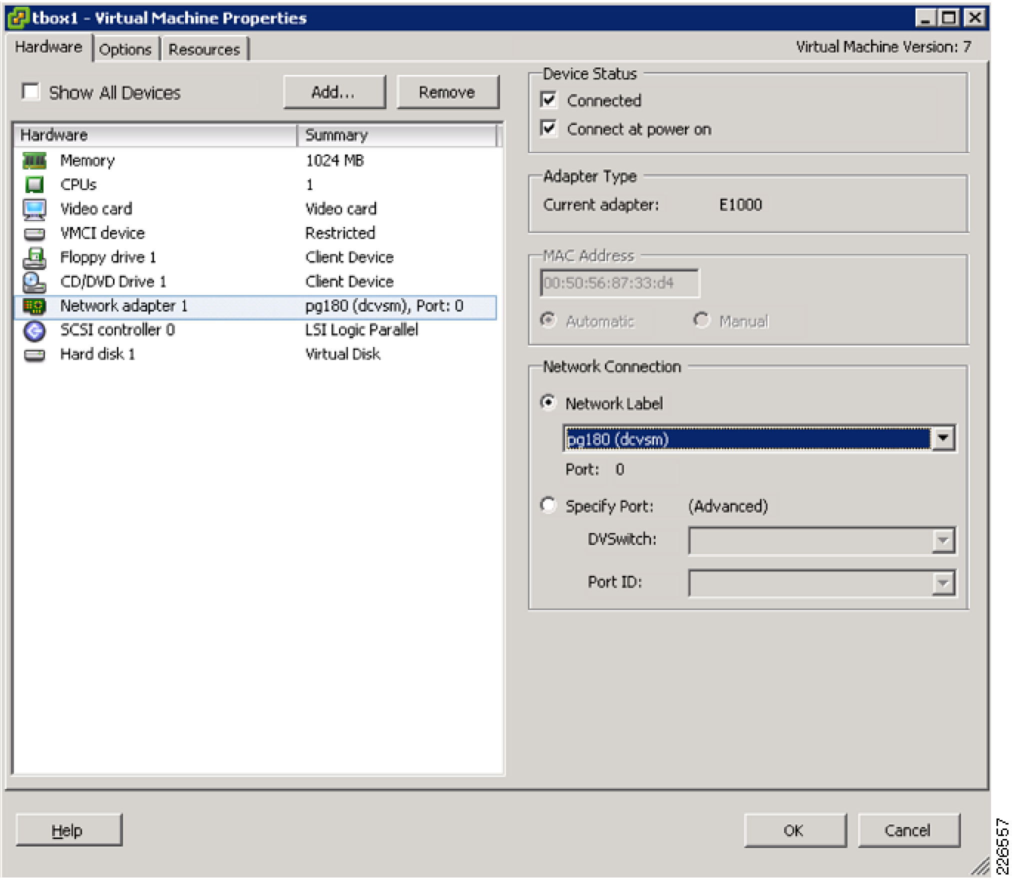
```
interface Ethernet3/3
 channel-group 1
  inherit port-profile n1kuplinks

interface Ethernet3/4
 channel-group 1
  inherit port-profile n1kuplinks

interface Ethernet4/3
  channel-group 2
  inherit port-profile n1kuplinks

interface Ethernet4/4
  channel-group 2
  inherit port-profile n1kuplinks

interface Ethernet5/3
  channel-group 3
  inherit port-profile n1kuplinks

interface Ethernet5/4
  channel-group 3
  inherit port-profile n1kuplinks
```

Edge Ports

The use of port profiles allows for rapid deployment of virtual machine vNICs. This example shows an access port available to the ESX administrator in vSphere as **pg180**. The access VLAN is 180 and there is NetFlow monitoring enabled on all virtual Ethernet interfaces created with using this profile.

```
port-profile vm180
  vmware port-group pg180
  switchport mode access
  switchport access vlan 180
  ip flow monitor ESE-flow input
  ip flow monitor ESE-flow output
  no shutdown
```

Figure 31 Example of Nexus 1000v Port Group Available to vSphere Client



tbox1 - Virtual Machine Properties
Hardware
Options
Resources
Virtual Machine Version: 7
Add...
Remove
Help
OK
Cancel
226557

□ Show All Devices

Hardware	Summary
Memory	1024 MB
CPUs	1
Video card	Video card
VMCI device	Restricted
Floppy drive 1	Client Device
CD/DVD Drive 1	Client Device
Network adapter 1	pg180 (dcvsm), Port: 0
SCSI controller 0	LSI Logic Parallel
Hard disk 1	Virtual Disk

Device Status

☑ Connected

☑ Connect at power on

Adapter Type

Current adapter: E1000

MAC Address

00:50:56:87:33:d4

● Automatic ○ Manual

Network Connection

● Network Label

pg180 (dcvsm)

Port: 0

○ Specify Port: (Advanced)

DVSwitch:

Port ID:

For example, the **show interface** command output on the VSM indicates that VEthernet1 below is assigned to virtual machine “tbox1” that leverages the “vm180” port profile described above.

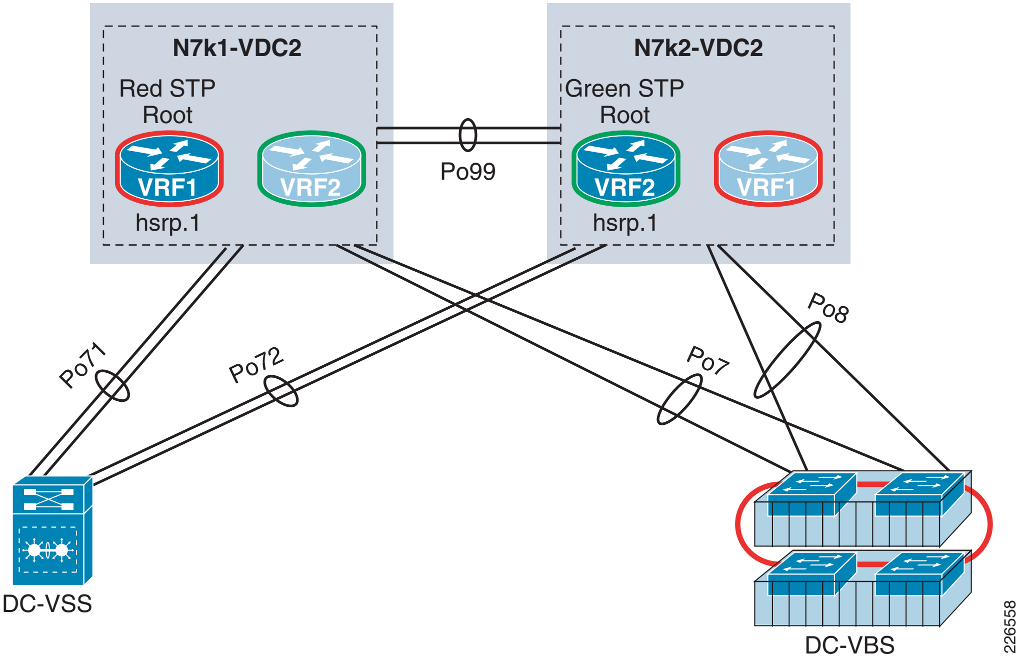
```
Vethernet1 is up
    Hardware is Virtual, address is 0050.5687.33d4
    Owner is VM "tbox1", adapter is Network Adapter 1
    Active on module 3
    VMware DVS port 0
    Port-Profile is vm180
    Port mode is access
    Rx
    5430983 Input Packets 5416452 Unicast Packets
    103 Multicast Packets 14428 Broadcast Packets
    6930761161 Bytes
    Tx
    9660234 Output Packets 9360512 Unicast Packets
    143 Multicast Packets 299579 Broadcast Packets 300281 Flood Packets
    746392601 Bytes
    30 Input Packet Drops 12 Output Packet Drops
```

Catalyst 3120 with VBS

Figure 32 illustrates the connectivity of the Virtual Blade Switch (VBS) to the Nexus aggregation layer. The VBS is a virtual switch stack that allows up to eight physical blade switches to appear as one logical switch. This obviously simplifies spanning tree and the operational requirements in the data center.

Note The Nexus 7000 layer under test does not support virtual port channeling. The vPC combined with VBS would eliminate Layer 2 loops.

Figure 32 Access Layer Uplinks to Nexus 7000 for Catalyst 6500 VSS and Catalyst 3120 VBS Platforms



Uplink Ports to Nexus 7000

The VBS switch is dual-homed to the Nexus aggregation switches. In reality, there are four physical ten Gigabit Ethernet switches providing uplink capacity. These uplinks are distributed across two physical 3120 blade switches, but this number could be increased to eight switches for improved availability and system uplink capacity. Each of the VBS port-channel connects to a single Nexus 7000 aggregation layer VDC.

The VBS port channel uses Loopguard and is configured as such:

```
interface Port-channel7
 description ** to Nexus 7000 **
 switchport trunk allowed vlan 180-183
 switchport mode trunk
 spanning-tree guard loop
end
```

The member interfaces employ LACP to form the channel with the aggregation switch.

```
interface TenGigabitEthernet4/0/1
 switchport trunk allowed vlan 180-183
 switchport mode trunk
 channel-group 7 mode active
end
```

Upon further review of Figure 32, it is apparent that each Nexus 7000 leverages identical port channel configurations, in this case port channels 7 and 8. The **show etherchannel summary** command verifies the connectivity of each to the Nexus aggregation switches.

```
dc07-3120-vbs# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        …
Number of channel-groups in use: 6
Number of aggregators:           6
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
7      Po7(SU)         LACP      Te2/0/1(P)  Te4/0/1(P)
8      Po8(SU)         LACP      Te2/0/2(P)  Te4/0/2(P)
```

Each of the Nexus 7000 VBS port channels are configured identically, with normal spanning tree settings and the appropriate VLANs.

```
interface port-channel7
  description to vbs
  switchport mode trunk
  switchport trunk allowed vlan 180-183
  spanning-tree port type normal
```

Member interfaces confirm the LACP active mode is enabled.

```
interface Ethernet1/20
  description dc07-3120-vbs Ten2/0/1
  switchport mode trunk
  switchport trunk allowed vlan 180-183
  spanning-tree port type normal
  spanning-tree guard root
  channel-group 7 mode active
```

The uplink capacity of the VBS may meet or exceed the ISL at the Nexus aggregation layer. This could potentially result in a ROOT_INC* state if the root and secondary root priorities for each VLAN are not set properly in the aggregation layer. As the VBS path may have less of a STP cost associated with it when compared to the Nexus ISL. To avoid this situation, lower the STP cost associated with the Nexus ISL. In this example, apply the **spanning-tree cost** command to each side of the ISL port-channel configuration.

```
spanning-tree cost 500
```

The **show spanning-tree vlan** command verifies that the ISL at the aggregation layer is the preferred root path based on cost and not the VBS destined port channel.

```
dca-n7k2-vdc2# show spanning-tree vlan 180
VLAN0180
  Spanning tree enabled protocol rstp
  Root ID    Priority    24756
             Address     0023.ac64.73c3
             Cost        1000
             Port        4194 (port-channel99)
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32948  (priority 32768 sys-id-ext 180)
             Address     0023.ac64.7443
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po8              Desg FWD 1000      128.4103 P2p
Po99             Root FWD 500      128.4194 Network P2p
Eth1/26          Desg FWD 2000      128.154  Network P2p
```

Edge Ports

The use of VBS allows the blade servers within the chassis to use 2 to 4 physical adapters per server (i.e., NIC teaming us readily available). There are many NIC teaming methods, but the ability to improve the availability of the server and/or its network capacity are fundamental drivers for enabling NIC teaming. The VBS switch will support static EtherChannel, dynamic port aggregation, active-standby, and trunking schemes and more importantly the motivation for NIC teaming on the servers.

Catalyst 6500 with VSS

Figure 32 illustrates two port channel uplinks to the Nexus aggregation layer. The port channels physically reside on the two chassis comprising the VSS. Each of the Nexus aggregation boxes employ a single port channel to connect to the VSS access layer switch.

Note This solution was not tested using the Nexus 7000 vPC feature, which only adds to the simplicity of the design.

Uplink Ports to Nexus 7000

The following configuration was implemented on the VSS to connect to the Nexus aggregation layer. As of IOS Release 12.2(33)SXI, VSS supports spanning tree bridge assurance.

```
interface Port-channel71
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 15,142,180-183,300-399,3002,3003
 switchport mode trunk
 spanning-tree portfast network
end
```

The Nexus 7000 port channel to the VSS access layer is nearly identical to its counterpart with the exception of Root Guard being enabled.

```
interface port-channel71
  switchport mode trunk
  switchport trunk allowed vlan 15,142,180-183,300-399,3002-3003
  spanning-tree port type network
  spanning-tree guard root
```

Edge Ports

For information regarding the use of VSS in the data center access layer, refer to the *Integrating the Virtual Switching System in Cisco Data Center Infrastructure* document at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/vssdc_integrate.html

Additional References

- Data Center Design—IP Network Infrastructure

  http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/DC-3_0_IPInfra.html

- Security and Virtualization in the Data Center http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_sec_design.html

- Integrating the Virtual Switching System in Cisco Data Center Infrastructure

  http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/vssdc_integrate.html

- Implementing Nexus 7000 in the Data Center Aggregation Layer with Services

  http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html