



**Important—Updated content: The Cisco Virtualized Multi-tenant Data Center CVD (<http://www.cisco.com/go/vmdc>) provides updated design guidance including the Cisco Nexus Switch and Unified Computing System (UCS) platforms.**

---

# Data Center Design—IP Network Infrastructure

---

Cisco Validated Design

November 2, 2011

## Contents

Introduction	2
Audience	3
Overview	3
Data Center Network Topologies	3
Hierarchical Network Design Reference Model	4
Correlation to Physical Site Design	5
Core Layer	6
Aggregation Layer	8
Access Layer	8
Alternatives for Service Insertion	12
Specialized Topologies	13
Managing Oversubscription	15
Nexus 7000 Virtual Device Contexts	17
VDC Hardware Overview	17
VDC Topology Examples	19
Data Center Logical Topologies	24
Basic Client-Server Model with Services	24



---

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

Copyright © 2009 Cisco Systems, Inc. All rights reserved.

Multi-Tier Application Model	25
Layer 3 Design and Features	27
Overview	27
Layer 3 Feature Best Practices	27
IP Route Summarization	27
IGP Hello and Dead/Hold Timer Settings	28
Router and Interface Definition	29
Routing Protocol Authentication	31
OSPF Reference Bandwidth	31
OSPF Throttle Timers	32
First Hop Redundancy	32
IP Multicast	33
Layer 2 Design and Features	34
Using Layer 2 at the Aggregation and Access Layer	34
Traditional STP Design	35
STP Stability	35
STP Features	37
Other Layer 2 Relevant Features	42
Network Design with STP	45
"Loop Free" Layer 2 Design	51
Virtual Switching System (VSS)	52
Virtual Port Channel (vPC)	61
Conclusion	69
Additional References	69
Cisco Validated Design	70

## Introduction

Cisco provides network design guidance focused on specific areas of the enterprise network, such as data center, campus, branch/WAN, and Internet edge. Hierarchical network design is a common theme across these areas, as is designing the network for high availability, security, scalability, and performance. The data center brings its own specific challenges to network design, based on the density of high-performance end nodes within a very constrained physical space. Trends in the industry (such as the virtualization of servers, services devices, and infrastructure components) provide great flexibility and extend how some layers of the classic hierarchical design are defined. The proliferation of network interfaces and VLANs together pose a design challenge to allow for the physical and logical movement of servers around the data center facility while also providing for network stability and constrained failure domains.

Cisco Data Center 3.0 provides an architecture to unify the virtualization capabilities of individual devices to create a fully virtualized data center. Benefits of this approach include lower total cost of ownership (TCO), increased resilience, and improved agility and responsiveness to changing business needs. Multiple Cisco Validated Design (CVD) documents are available to help network architects

incorporate new Cisco products and features into the data center to realize these benefits. This document focuses specifically on IP network infrastructure design and provides guidance on high-level network topologies as well as details regarding best practices for configuring Layer-3 and Layer-2 features.

## Audience

This design document is intended for network architects who are designing the physical and logical data center IP network topology.

## Overview

The Cisco Data Center 3.0 architecture is built on the concept of investment protection, allowing incremental integration of new technologies and platforms into the existing infrastructure to meet changing business needs. Specific business requirements in the enterprise data center can dictate highly customized topologies, but often many of the same fundamental design principles and technologies can be used to construct them. This document is targeted at providing a reference topology relevant to today's common enterprise data center environment that has been extended to reflect some of the new products and features that are available to the network designer. Best practices in the area of network device configuration for Layer-3 and Layer-2 features are also provided.

The main sections of this design document cover the following:

- **Data Center Network Topologies**—This section discusses the hierarchical network design reference model, insertion of data center services, designs using Nexus 7000 Virtual Device Contexts (VDCs), and the use of logical networking constructs to control the flow of traffic across the topology.
- **Layer 3 Design and Features**—This section discusses IP routing topology design decisions and features. These guidelines pertain primarily to configuration of and interaction between the core and aggregation layers of the network with additional implications on insertion of data center services—such as firewall and server load balancing.
- **Layer 2 Design and Features**—This section discusses Ethernet forwarding within a broadcast domain, with a particular emphasis on Spanning Tree Protocol (STP) and alternative features for forwarding loop prevention such as Virtual Port Channels (vPC) in NX-OS and the Cisco Catalyst 6500 Virtual Switching System (VSS). These guidelines pertain primarily to configuration of and interaction between the aggregation and access layers of the network, but also affect design decisions for service insertion.

Design recommendations specific to data center serverfarm design and security are available in the following two CVD companion documents:

- *Data Center Service Patterns*  
[http://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/DC\\_3\\_0/dc\\_serv\\_pat.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_serv_pat.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](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_sec_design.html)

## Data Center Network Topologies

Data center networking technology is currently an area of rapid change. Higher-performance end nodes and the migration to 10-Gigabit Ethernet for edge connectivity are changing design standards while virtualization capabilities are expanding the tools available to the network architect. When designing the

data center network, a solid hierarchical foundation provides for high availability and continued scalability. This foundation also provides the flexibility to create different logical topologies utilizing device virtualization, the insertion of service devices, as well as traditional Layer-3 and Layer-2 network configuration. The following section describes the hierarchical network design reference model as applied to meet the requirements and constraints commonly found in today's data centers. As a reference model, this topology is flexible and extensible, and may need to be extended or modified to meet the requirements of a specific enterprise data center network.

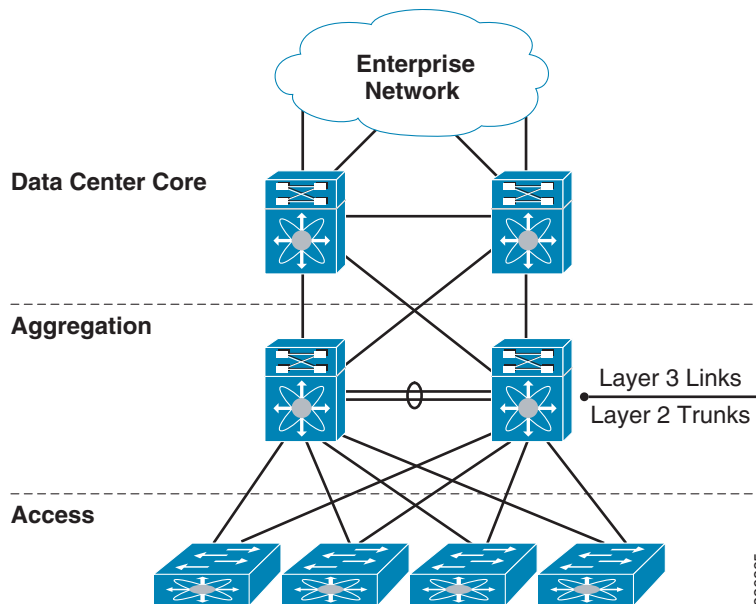
## Hierarchical Network Design Reference Model

Hierarchical network design has been commonly used in enterprise networking for many years. This model uses redundant switches at each layer of the network topology for device-level failover that creates a highly available transport between end nodes using the network. Data center networks often require additional services beyond basic packet forwarding, such as server load balancing, firewall, or intrusion prevention. These services might be introduced as modules populating a slot of one of the switching nodes in the network, or as standalone appliance devices. Each of these service approaches also supports the deployment of redundant hardware to preserve the high availability standards set by the network topology.

A structured data center environment uses a physical layout that correlates tightly to the hierarchy of the network topology. Decisions on cabling types and the placement of patch panels and physical aggregation points must match the interface types and densities of the physical switches being deployed. In a new data center build-out, the two can be designed simultaneously, also taking into consideration the constraints of power and cooling resources. When seeking to avoid significant new investment within an existing data center facility, the pre-existing physical environment of cabling, power, and cooling can strongly influence the selection of switching platforms. Careful planning in conjunction with networking requirements and an eye toward flexibility for the future is critical when designing the physical data center environment. Taking a modular approach to data center design provides flexibility and scalability in both network topology design and utilization of physical resources.

[Figure 1](#) illustrates the primary network switching layers of the hierarchical network design reference model for the data center environment. The overall hierarchical model is similar to the reference topology for enterprise campus design, but the term *aggregation layer* replaces the term *distribution layer*. This denotes the fact that the data center network is less concerned with distributing network access across multiple geographically disparate wiring closets and is more focused on the aggregation of server resource—and providing an insertion point for shared data center services.

**Figure 1** Hierarchical Network Design Reference Model

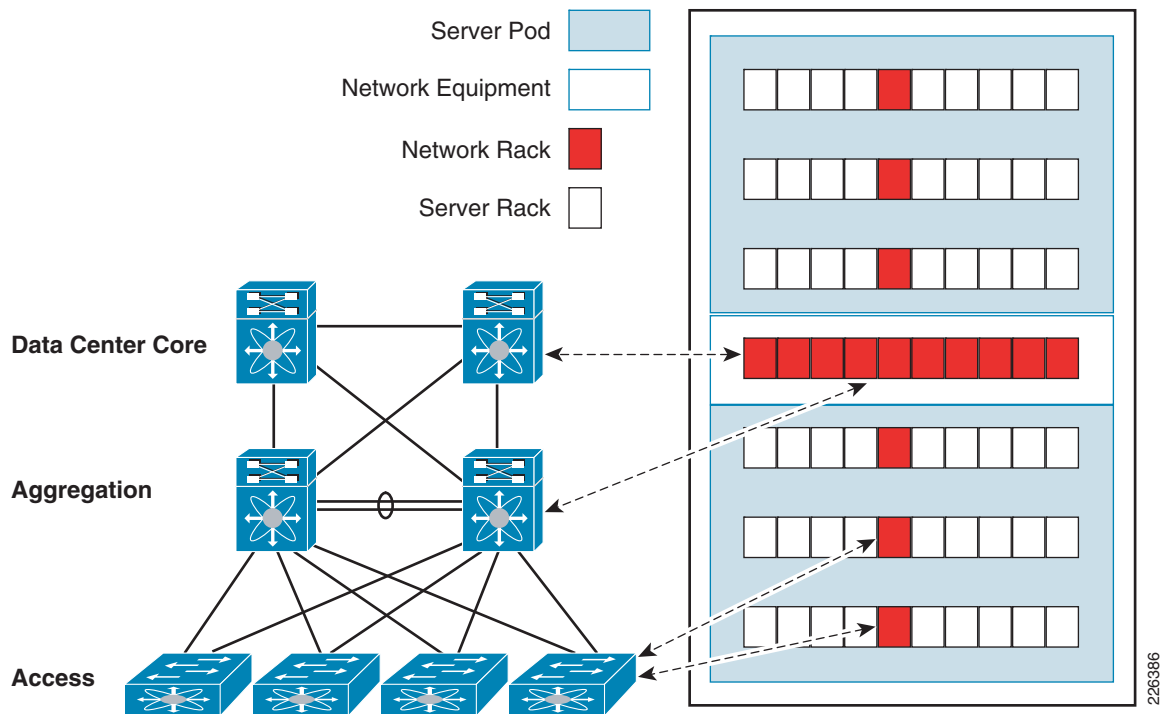


The reference model in [Figure 1](#) illustrates the boundary between Layer-3 routed networking and Layer-2 Ethernet broadcast domains at the aggregation layer. Larger Layer-2 domains increase the physical flexibility of the data center—providing the capability to manually or virtually relocate a server to a different physical rack location with less chance of requiring a change of IP addressing to map to a specific subnet. This physical flexibility comes with a tradeoff. Segregating the network into smaller broadcast domains results in smaller spanning tree domains and failure domains—which improve network stability, reduce convergence times and simplify troubleshooting. When determining how to scale Layer-2 domains, the network architect must consider many factors including the access switching model in use and the nature of the underlying applications being serviced. Cisco has introduced features such as bridge assurance and dispute mechanism into switching products to allow greater scalability of Layer-2 domains with increased stability of the STP. These features are discussed in the [“Layer 2 Design and Features”](#) section on page 34.

## Correlation to Physical Site Design

Designing the data center topology in a structured manner that correlates closely to the physical design provides for simplified support and troubleshooting processes. Using the same approach across multiple physical data center sites within the enterprise provides additional benefits of consistency for data center operations staff responsible for multiple site locations. An example of a modular approach is to build a group of similar server racks, access switches, and associated aggregation switches into a logical unit—often referred to as a *pod* or a *zone*. The data center may then be scaled larger by replicating the model to encompass additional pods, or potentially different types of pods based on server functions or access switch model such as a *top-of-rack (ToR)* or *end-of-row (EoR)* physical access switch model.

**Figure 2 Mapping the Network to DC Floor Plan**



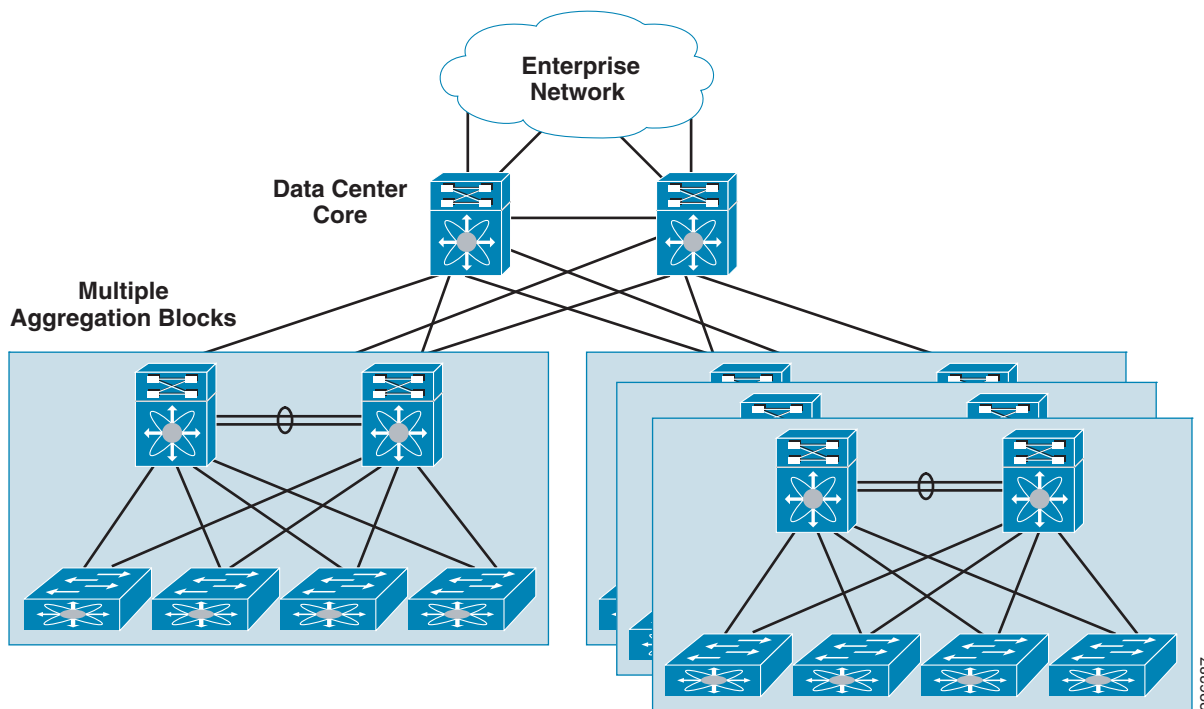
An example of mapping the layers of the reference topology to areas of the physical data center floor plan is illustrated in Figure 2. Many considerations such as the types of servers in use, server virtualization models, location of storage resources, power, and cooling must be assessed when developing a zone or pod model for a specific enterprise data center requirement. In this particular example, a *middle-of-row* (MoR) access-switching model is in use keeping the horizontal cabling required from individual servers within a given row. Aggregation and data center core switching resources are segregated in a centrally located row designated for network equipment. In this case, the aggregation switch pair is considered part of the data center pod. In some cases a multi-tier Layer-2 access model may be in place, or virtualization features may be in use within the access layer that define the boundaries of a pod differently. There are many possible variations to using the pod or zone concept to scale the data center topology, Figure 2 illustrates an example of one possible approach.

## Core Layer

The hierarchical network design model gains much of its stability and high availability characteristics by splitting out switching nodes based on their function, and providing redundant switching units for each functional layer required. The core of a data center network is typically broken out into a pair of high performance, highly available chassis-based switches. In larger or geographically dispersed network environments the core is sometimes extended to contain additional switches, the recommended approach is to scale the network core continuing to use switches in redundant pairs. The primary function of the data center network core is to provide highly available, high performance Layer-3 switching for IP traffic between the other functional blocks of the network such as campus, Internet edge, WAN/branch, and data center. By configuring all links connecting to the network core as point-to-point Layer-3 connections, rapid convergence around any link failure is provided, and the control plane of the core switches is not exposed to broadcast traffic from end node devices or required to participate in STP for Layer-2 network loop prevention.

In small-to-medium enterprise environments, it is reasonable to connect a single data center aggregation block directly to the enterprise switching core for Layer-3 transport to the rest of the enterprise network. Provisioning a separate, dedicated pair of data center core switches provides additional insulation from the rest of the enterprise network for routing stability and also provides a point of scalability for future expansion of the data center topology. As the business requirements expand and dictate two or more aggregation blocks serving separate pods or zones of the data center, a dedicated data center core network provides for scale expansion without requiring additional Layer-3 interfaces to be available on the enterprise core. An illustration of scaling the data center topology with a dedicated core and multiple aggregation blocks is provided in Figure 3.

**Figure 3**      **Scaling the Data Center with a Dedicated Core**



Cisco's premier switching platform for the data center core is the Nexus 7000 Series switch. The Nexus 7000 Series has been designed from the ground up to support the stringent uptime requirements of the data center. The Nexus 7000 Series switches are optimized for support of high density 10-Gigabit Ethernet, providing scalability in the 18-slot chassis up to 128 wire rate 10-Gigabit Ethernet interfaces when ports are configured in a dedicated mode using the N7K-M132XP-12 I/O Module. The Nexus 7000 Series hardware is coupled with Cisco NX-OS, a modular operating system also designed specifically for the requirements of today's data center networks. NX-OS is built on the industry-proven SAN-OS software-adding virtualization, Layer-2, and Layer-3 features and protocols required in the data center environment. NX-OS includes high availability features-such as granular process modularity, In-Service Software Upgrade (ISSU) and stateful process restart that are specifically targeted at the service-level requirements of the enterprise or service provider data center.

When choosing switching platforms to provision layers of the data center network topology, the network architect must be aware of specific features and interface types required by the network design. The Nexus 7000 Series offers unique virtualization features such as Virtual Device Contexts (VDCs) and Virtual Port Channels (vPCs). The Nexus 7000 Series switches also have excellent high availability features, throughput, and 10-Gigabit Ethernet port densities, however, NX-OS does not support some of the features found in Cisco IOS-based switching platforms. Another Cisco switching platform commonly found in the core of today's data centers is the Cisco Catalyst 6500. The Catalyst 6500 offers

software features such as support of Multi Protocol Label Switching (MPLS), VLAN Mapping, and Q-in-Q multiple-level VLAN tagging that may be required in specific designs. The Cisco Catalyst 6500 also offers greater diversity of physical interface types and support for services modules directly installed within the chassis.

## Aggregation Layer

The aggregation layer of the data center provides a consolidation point where access layer switches are connected providing connectivity between servers for multi-tier applications, as well as connectivity across the core of the network to clients residing within the campus, WAN, or Internet. The aggregation layer typically provides the boundary between Layer-3 routed links and Layer-2 Ethernet broadcast domains in the data center. The access switches are connected to the aggregation layer using 802.1Q VLAN trunks to provide the capability of connecting servers belonging to different VLANs and IP subnets to the same physical access switch. An alternate aggregation layer option is to use Layer-3 routed point-to-point links also between the aggregation layer and access layers, referred to as a *routed-access design*. The routed-access design provides deterministic failover characteristics; it features very small broadcast and failure domains contained to a single switch. However, because the routed access model limits a given VLAN or IP subnet to a single access switch, it also severely limits the mobility of servers without requiring IP address changes and can complicate designs—such as Network Interface Card (NIC) teaming approaches where multiple interfaces from a server carry the same IP address.

Traditional models of access-layer connectivity include links from each of the access-layer switches into both switches forming the aggregation-layer redundant switch pair. This approach provides network resiliency in the even of a single link or interface failover, or failure of one of the aggregation switches. The inter-switch link between the two aggregation switches is also an 802.1Q trunk that carries all of the VLANs in use in the serverfarm. The STP is active independently for each VLAN instance using the Rapid Per VLAN Spanning Tree Plus (RPVST+) model, which blocks redundant ports when they are not needed to avoid network loops. Features such as Virtual Port Channels (vPC) on the Cisco Nexus 7000 Series and Virtual Switching System (VSS) on the Catalyst 6500 series have been introduced to allow both switches in the aggregation pair to act as a single switching unit from a STP and port channel perspective. This approach allows all links between an access switch and the aggregation layer to be active as a single port channel instead of having STP blocking a redundant path. More detail on Layer 2 configuration approaches for the aggregation layer is available in the [“Layer 2 Design and Features” section on page 34](#).

The Nexus 7000 Series and Catalyst 6500 Series modular chassis switches are both excellent choices for the enterprise data center. Both platforms have their respective implementations of loop-free alternatives to STP for access layer switch aggregation (Nexus 7000 vPC and Catalyst 6500 VSS). However, the Nexus 7000 Series is positioned as the primary area of focus for new data center and 10-Gigabit Ethernet-oriented features. These capabilities are geared towards the support of a unified fabric architecture supporting combined Fibre Channel over Ethernet (FCoE) and IP traffic on a common infrastructure. The NX-OS operating system on the Nexus 7000 Series is also shared with other Cisco Nexus family switches. The Nexus 5000 Series switches offer unique benefits in the access layer of the data center, providing innovative ToR switching and storage integration options to the network architect.

## Access Layer

### Traditional Models

The access layer of the network provides connectivity for serverfarm end nodes residing in the data center. Design of the access layer is tightly coupled to decisions on server density, form factor, and server virtualization that can result in higher interface count requirements. Traditional data center access layer

designs are strongly influenced by the need to locate switches in a way that most conveniently provides cabling connectivity for racks full of server resources. The most commonly used traditional approaches for data center serverfarm connectivity are *end-of-row*, *top-of-rack*, and integrated switching. Each design approach has pros and cons, and many enterprises use multiple access models in the same data center facility as dictated by server hardware and application requirements.

The *end-of-row* (EoR) switching model consists of a large chassis-based switch used to provide end node connectivity for each group of server racks in a single row of the data center. The chassis switch may alternatively be located more towards the middle of a given row, to decrease the length and bulk of cabling required for server connectivity. EoR access switches can help to keep server-to-server traffic local to the access switch when on the same VLAN, and provide scalable interface utilization with the installation or removal of individual I/O modules in the chassis. The Cisco Nexus 7000 Series and Catalyst 6500 Series switches provide possible platform choices for data center EoR access layer designs.

The *top-of-rack* (ToR) switching model uses smaller 1 to 2 rack unit (RU) switches located in each individual server rack to provide access switching connectivity. This model provides short server cabling runs and is often used for high density 1 RU server environments, or virtualized server deployments where individual physical servers may require a higher density of physical interfaces. Single switches may be used in each rack, two switches are sometimes used to provide redundancy within one rack or shared across a pair of racks for connecting dual-homed servers. The ToR model can increase the aggregate bandwidth between the access and aggregation layers due to the large number of uplinks required to connect the ToR switches. This approach also increases port density requirements and Layer-2 control plane load at the aggregation layer. Some deployments use a "looped-square" topology for aggregation connections where ToR switches are daisy-chained to one another in pairs to reduce aggregation port count requirements. Depending on the consistency of the number of active ports in a rack, the ToR model may at times result in inefficient port utilization since switches designed for ToR use are typically not as scalable as a chassis in terms of port count. The Cisco Nexus 5000 Series with the Nexus 2000 Series Fabric Extenders, or the Catalyst 4900 Series switches are potential platform choices for a data center ToR access layer design.

*Integrated switching* refers to the inclusion of blade switches inside a modular blade server chassis environment. Since these units are integrated into the server housing, this approach sometimes can blur the line between server and access switch from a configuration and support perspective. It is critical to stability of the access layer to ensure these integrated switches are configured and supported with similar operational concepts and best practices as EoR and ToR switches in the areas of STP configuration and VLAN trunking. An alternative approach that is sometimes used in blade server environments is a passthrough approach where the blade servers are cabled to an external ToR switch. Where blade switches for an integrated switching access layer are required, the Cisco Catalyst 3100 Series blade switches provide an excellent option to maintain a consistent Cisco feature set across the data center switching fabric.

## Virtual Access Evolution

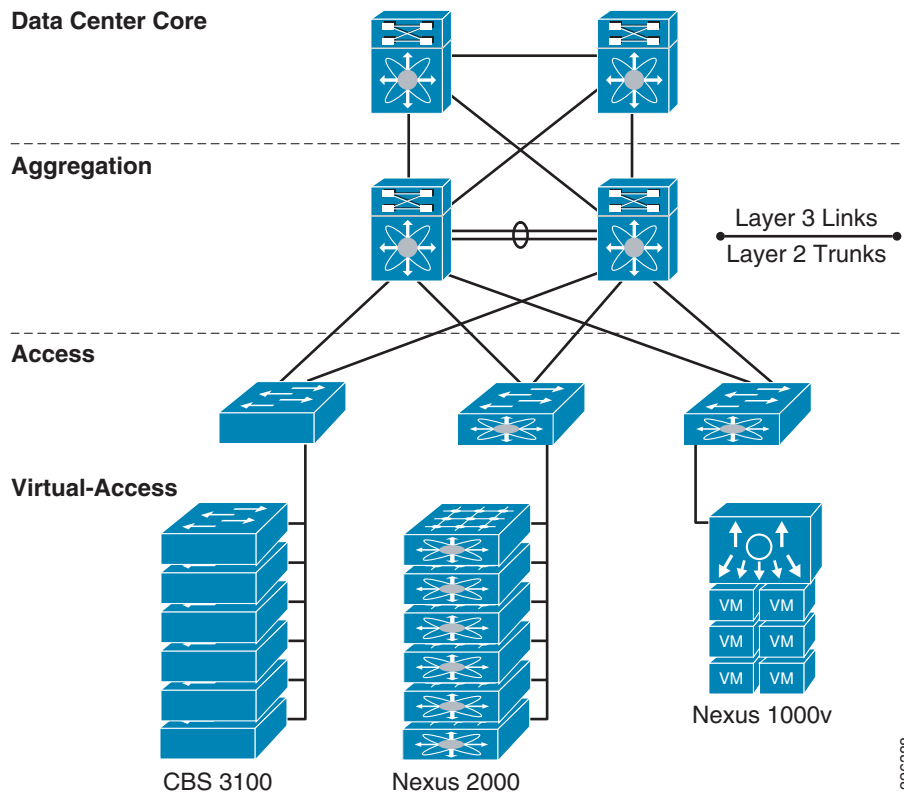
The evolution of networking technology in the data center is most evident at the access layer of the network and within the serverfarm. Several options for building the data center access layer introduce switch virtualization that allows the function of the logical Layer-2 access layer to span multiple physical devices. For example:

- Cisco Catalyst Blade Switch 3100 series include Virtual Blade Switch (VBS) technology that allows multiple physical blade switches to share a common management and control plane by appearing as a single switching node.
- Cisco Nexus 5000 Series switches work in conjunction with the Cisco Nexus 2000 Series Fabric Extenders to act as a single virtual access switch while providing ToR connectivity for servers in multiple racks.

- The software-based switching implementation in the Cisco Nexus 1000V Virtual Distributed Switch also provides virtual access layer switching capabilities designed to operate in server virtualization environments.

Figure 4 illustrates these examples of access-layer virtualization in the data center network. The virtual-access sublayer does not represent an additional level of Layer-2 switching; it conceptually exists as virtual I/O modules or line cards extended from a centralized management and control plane. This approach offers many of the benefits of EoR switching such as reduced aggregation switch port density requirements and fewer points of management, while providing cable-management benefits similar to a ToR model.

**Figure 4** Data Center Virtual-Access Options



The Cisco Nexus 5000 Series switches provide high-density 10-Gigabit Ethernet connectivity and innovative storage integration capabilities for the support of FCoE. With a Layer-2 capable implementation of NX-OS, the Nexus 5000 is optimized for the evolving data center access layer. For customers currently requiring density of 1-Gigabit Ethernet server connectivity, the Nexus 2000 Fabric Extenders may be deployed in conjunction with a Nexus 5000 Series switch and treated as a single virtual chassis in the access layer of the data center topology. This approach may be used to provide ToR switching to multiple racks of servers, with all management functions for the Nexus 2000 Fabric Extenders centralized into their associated Nexus 5000 Series switch. The Nexus 5000 Series located in a *middle-of-row* (MoR) location can also provide nearby 10-Gigabit Ethernet interfaces as servers within the row begin transition to 10-Gigabit Ethernet.

Implementations of hypervisor-based server virtualization systems include software-based logical switching capabilities within the server. Cisco has announced the Nexus 1000V virtual distributed switch to allow the network architect to provide a consistent networking feature set across both physical servers and virtualized servers. This capability is planned for release in the first half of calendar year 2009. The

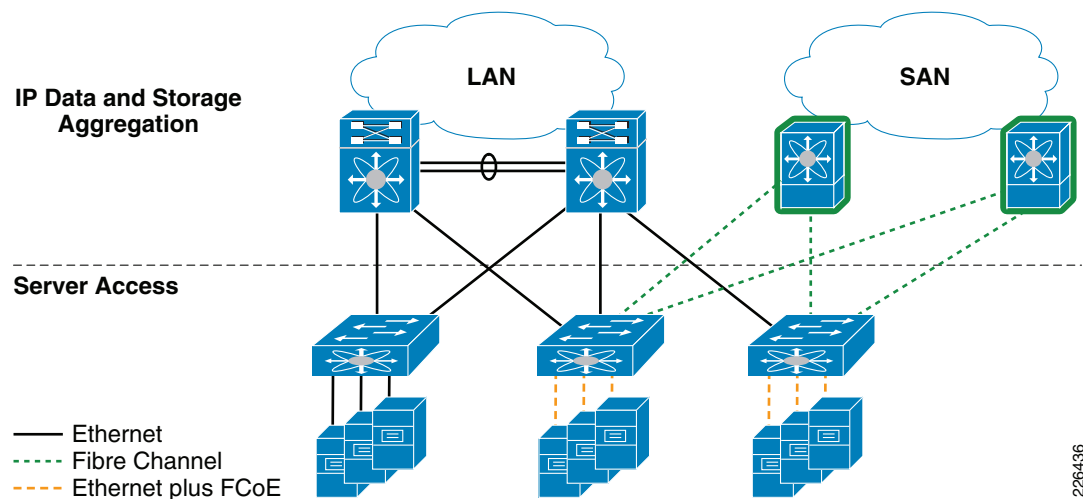
Cisco 1000V operates as a virtualized chassis switch, with Virtual Ethernet Modules (VEMs) resident on the individual virtualized servers managed by a central Virtual Supervisor Module (VSM) that controls the multiple VEMs as one logical modular switch. The VSM provides a centralized point of configuration and policy management for the entire virtual distributed switch. Both the Cisco Nexus 2000 Fabric Extenders and the Cisco Nexus 1000V represent variations on the evolving capabilities of the data center virtual-access sublayer.

## Storage Integration

Another important factor changing the landscape of the data center access layer is the convergence of storage and IP data traffic onto a common physical infrastructure, referred to as a unified fabric. The unified fabric architecture offers cost savings in multiple areas including server adapters, rack space, power, cooling, and cabling. The Cisco Nexus family of switches, particularly the Nexus 5000 Series is spearheading this convergence of storage and data traffic through support of Fibre Channel over Ethernet (FCoE) switching in conjunction with high-density 10-Gigabit Ethernet interfaces. Server nodes may be deployed with converged network adapters (CNAs) supporting both IP data and FCoE storage traffic, allowing the server to use a single set of cabling and a common network interface. The Cisco Nexus 5000 Series also offers native Fibre Channel interfaces to allow these CNA attached servers to communicate with traditional Storage Area Network (SAN) equipment.

At its initial product release, the Cisco Nexus 5000 supports a unified fabric switching approach only at the edge of the data center topology. Over time, the Cisco Nexus family will allow further consolidation of FCoE-based storage traffic into the aggregation layer of the data center. Choosing Cisco Nexus switching platforms for new data center investment today positions the network architect to take advantage of additional I/O consolidation features as they are released across the product family. [Figure 5](#) illustrates a topology with CNA-attached servers running both FCoE traffic and IP data traffic over a common interface to a Nexus 5000 switch. The Nexus 5000 splits out the FCoE traffic and provides native Fibre Channel interface connections back into Fibre Channel switches to connect to the shared SAN.

**Figure 5** Access Layer Storage Convergence with Nexus 5000



## Alternatives for Service Insertion

Integration of network services such as firewall capabilities and server load balancing is a critical component of designing the data center architecture. The aggregation layer is a common location for integration of these services since it typically provides the boundary between Layer 2 and Layer 3 in the data center and allows service devices to be shared across multiple access layer switches. Cisco offers integrated services modules for the Cisco Catalyst 6500 Series switching platform, such as the Firewall Services Module (FWSM) and Application Control Engine (ACE) Module. The Nexus 7000 Series does not currently support services modules. One option for using existing Catalyst 6500 Services Modules in a data center topology with a Nexus 7000-based aggregation layer is to house the modules in separate Catalyst 6500 Services Chassis. Locating services modules in external chassis allows I/O module slot real-estate in the aggregation switches to be dedicated purely to packet forwarding interfaces.

**Figure 6** *Services Chassis Reference Topology*

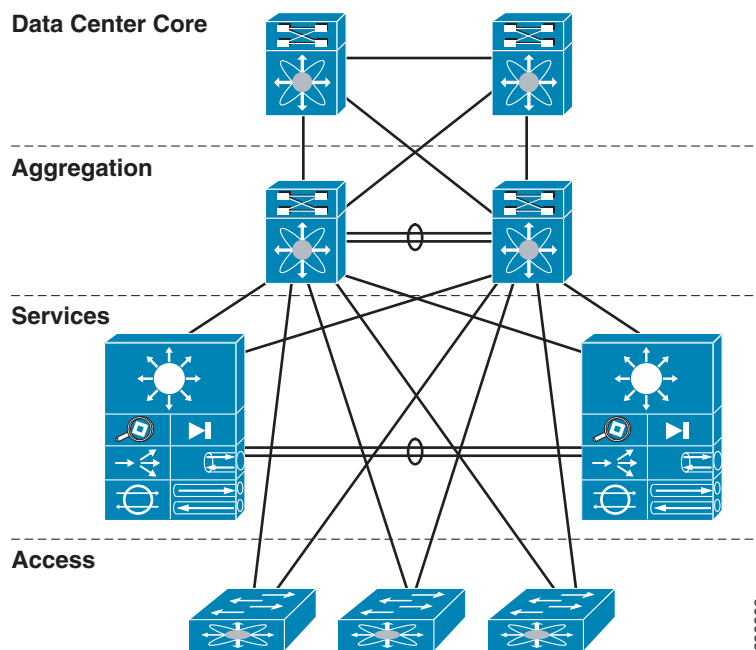
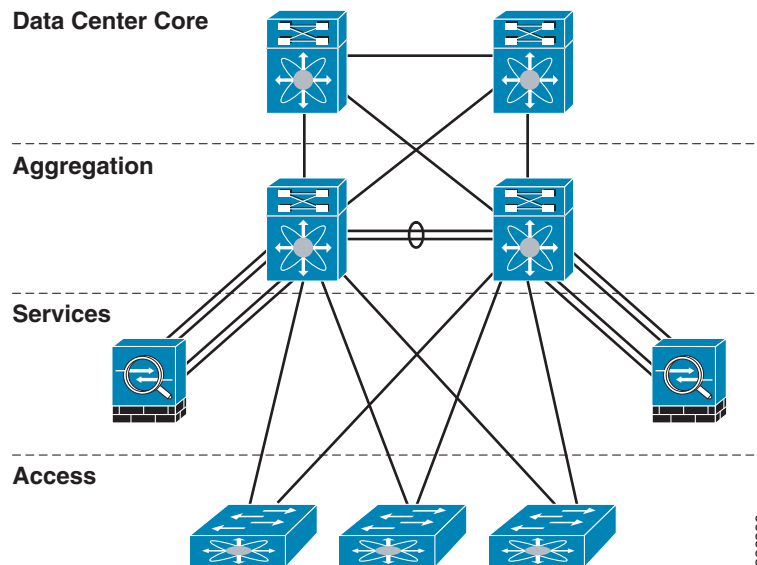


Figure 6 illustrates a reference topology for connecting external Catalyst 6500 Services Chassis to the data center aggregation layer. This topology uses a dual-homed approach for data path connectivity of the Services Chassis into both of the aggregation layer switches. This approach decouples the service modules from dependence on a specific aggregation switch, and provides operational flexibility for system maintenance that may be required to the aggregation switches or the services switches. From a high availability perspective, if one of the aggregation switches fails, traffic can continue to flow through the other aggregation switch to the active service modules without any failover event needing to occur with the service modules themselves. Separate links directly between the Services Chassis are shown, these links are optional and if provisioned should be dedicated to service module fault tolerance and stateful control plane traffic.

Another alternative for introduction of services into the data center topology is to use standalone appliance devices, such as the Cisco Adaptive Security Appliance (ASA) 5580 or the Cisco Application Control Engine (ACE) 4710 Appliance. If external Services Chassis switches are in place, the Services Chassis may be used as a general service aggregation sublayer and provide the connection point for

standalone service appliances. If Services Chassis are not part of the architecture, the service appliance devices may also be attached directly to the data center aggregation switches. An example of a service appliance topology using ASA 5580 devices is illustrated in Figure 7.

**Figure 7** Service Appliances Directly Attached to Aggregation



In Figure 7, each of the ASA-5580 devices is attached only to one of the aggregation switches using two separate 10 Gigabit Ethernet physical interfaces for VLANs that are carrying traffic from inside and outside of the firewall process respectively. The remaining two 1 Gigabit Ethernet physical connections are provisioned to carry failover and state traffic between the two ASA units in separate VLANs, which are extended between the aggregation switches. This type of connectivity has been validated for ASA units deployed in a transparent mode. The fate of each ASA device is tied to its directly attached aggregation switch, and the presence of redundant units provides high availability. ASA systems also support an interface redundancy feature which may be considered to support a dual-homed ASA architecture.



**Note**

Detailed configuration information for a transparent services topology using ASA-5580 devices directly attached to a Nexus 7000 aggregation layer is available in *Implementing Nexus 7000 in the Data Center Aggregation Layer with Services* at the following URL:

[https://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/nx\\_7000\\_dc.html](https://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html)

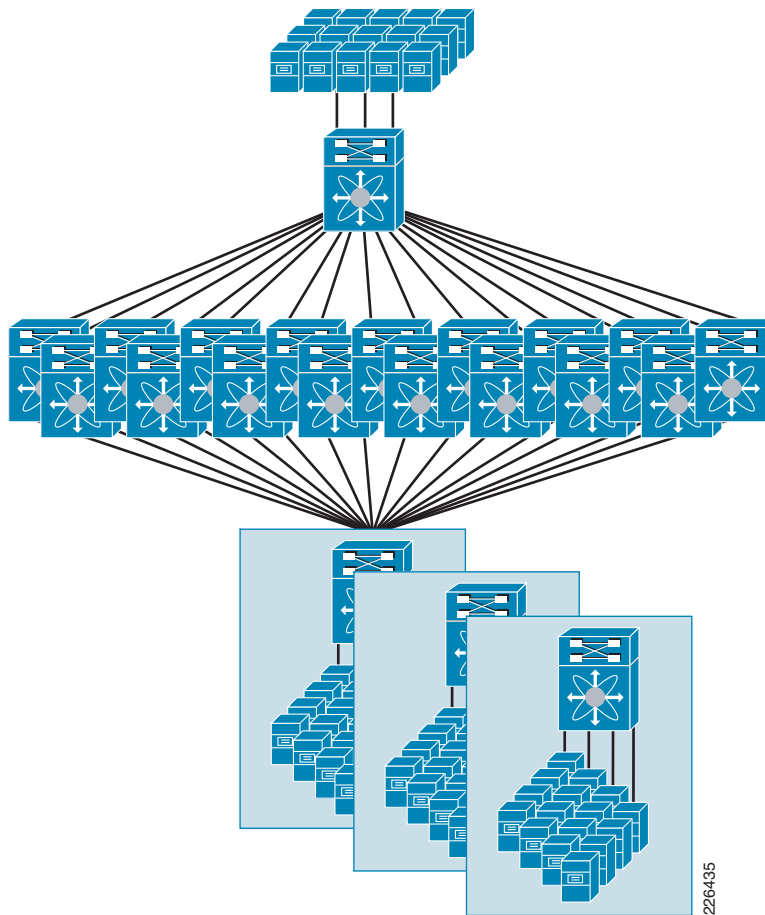
## Specialized Topologies

Server clustering technologies are becoming more prevalent in the data center to support application availability and scalability requirements. Some applications in the enterprise environment use redundant clustered servers to provide basic high availability or load balancing at the server level; these applications fit well into a traditional hierarchical design model. Other applications are more computation intensive, and use compute clusters to scale processing resources using software that allows multiple end nodes to appear as a single entity to the end user. These high performance computing clusters often require highly customized network designs to accommodate the throughput and low latency required to scale the system. In some cases, alternative interconnect technologies such as Infiniband or proprietary solutions are used in the design to meet the required performance characteristics. However, over half of the large clustered supercomputing implementations as listed at

<http://top500.org> use Ethernet as their primary interconnect technology. The Cisco Nexus 7000 and 5000 Series switching products offer multiple characteristics that make them ideal for the design of customized high performance computing cluster topologies.

For building large compute clusters with Layer 3 switching capabilities, the Nexus 7000 Series switches scale to support up to 256 10-Gigabit Ethernet interfaces on the 10-slot chassis, or 512 10-Gigabit Ethernet interfaces on the 18-slot chassis. The 1-Gigabit interfaces scale as high as 768 ports in a single 18-slot chassis using the 48-port I/O modules. The platform supports up to 5 hot-swappable redundant switch fabric modules, and the switch fabric architecture is designed to scale beyond 15 terabits per second (Tbps). The IP routing implementation in NX-OS supports up to 16 paths to the same destination network, and provides Equal Cost Multi Path (ECMP) load balancing for traffic to a given destination. This combined scalability of port density, switch fabric capacity, and layer 3 load balancing provides the flexibility to theoretically scale a routed cluster topology supporting 512 access switches each supporting hundreds of servers with aggregate bandwidth of 160 Gbps between any two access switches. An illustration of this type of cluster scaled with 16-way ECMP is show in [Figure 8](#).

**Figure 8** Nexus 7000 Cluster Topology with 16-Way ECMP



The Nexus 5000 Series switches also offer unique capabilities for design of server clusters requiring Layer 2 switching. The Nexus 5020 platform offers up to 52 line-rate 10-Gigabit Ethernet ports with an overall throughput of 1.04 Tbps. The Nexus 5000 uses a cut-through switching architecture that provides consistent low latency regardless of packet sizes or services enabled on the switch. Lossless forwarding for multiple classes of Ethernet traffic allows convergence of storage traffic onto a common cabling infrastructure. These features allow the network architect to build a very high-performance, low-latency

compute cluster with a single converged network adapter into each server delivering data and FCoE transport. The Nexus 5000 Series switching platform supports installation of modular native Fibre Channel interfaces to provide the connectivity to backend SAN resources.

## Managing Oversubscription

While it is possible to build a completely non-blocking switching fabric using custom server-clustering topologies, most data center topologies are built using some level of oversubscription between layers of the network, which reduces the overall cost. Server end nodes often are not able to transmit traffic at the line rate of their connected network interfaces, and if they can do so, it is only for very short periods of time. Statistically, it is not likely that all servers connected to an access-layer switch, for example, are attempting to transmit at line rate with traffic directed to the aggregation layer uplinks simultaneously. In designing the data center topology, the network architect must consider the characteristics of applications and server architectures being deployed to plan reasonable oversubscription rates in the network that will be cost effective without becoming severe bottlenecks for communication between end nodes.

Server virtualization technology achieves multiple benefits; one of the goals is to increase the average processor and hardware utilization per physical server by running multiple logical server instances. This also translates into a potential increase in the utilization of network interface cards in the server, depending on the number of interfaces installed and how the various VLANs supporting different server instances are allocated. When designing an access layer to support virtualized servers, planning for a higher expected bandwidth utilization per server interface than typically used for non-virtualized servers might be required.

Migration to 10-Gigabit Ethernet also has an impact on expected server utilization of network interfaces. A physical server that previously required 4 1-Gigabit Ethernet interfaces that each used an average of 50 percent may be migrated to a 10-Gigabit Ethernet CNA, where the previous data load now reflects only a 20 percent utilization rate on the interface. However, if FCoE traffic is also introduced over the same CNA, this may add additional load into the equation, and also reduce the overall level of utilization that will be considered practical from the IP data traffic on the interface.

### Oversubscription in Network Layers

In a three-tier data center network architecture, oversubscription is commonly calculated at the access and aggregation layers based on the ratio of network interfaces facing the serverfarm versus the number of interfaces-facing the data center core. This view of oversubscription is primarily switch-level and is relevant to traffic flow from the serverfarm out to clients that reside in the campus, branch/WAN, or over the Internet. Calculation of oversubscription must also take into account links that are blocking between the access and aggregation layers due to STP. With STP in place, if the interface from an access switch to the STP root switch in the aggregation layer fails and the backup link becomes forwarding, then there is no change to the effective oversubscription rate after the link has failed. Technologies such as vPC in NX-OS and VSS on the Catalyst 6500 offer alternatives to STP where all links are normally forwarding, this can reduce the effective oversubscription ratio when all links are active and healthy. From a high availability perspective, if it is necessary to maintain a determined oversubscription rate even in the event of single link failure, additional interfaces may be required since there may be no blocking path to transition to a forwarding state and provide additional bandwidth when a primary link fails.

For example in a simple access layer consisting of multiple 48-port 1-Gigabit Ethernet switches with two 10-Gigabit Ethernet uplinks, the ratio of server interface bandwidth to uplink bandwidth is 48 Gbps/10 Gbps if one of the interfaces is in a STP blocking state for redundancy. This results in a 4.8:1 ratio of oversubscription at the access layer for a given VLAN. Multiple VLANs may be configured on an access switch with their forwarding paths alternating between the two uplinks to distribute load and take better

advantage of the available uplink bandwidth. If vPC or VSS is in use and the two 10 Gbps uplinks are instead forming a port channel where both links are forwarding, the ratio is now 48Gbps/20 Gbps, which would result in an effective oversubscription ratio of 2.4:1 at the access layer when all links are active.

Extend this example to include an aggregation layer built with chassis switches that each have 64 ports of 10-Gigabit Ethernet. Allocating 8 ports from each aggregation switch into each core switch, and an 8-port channel between the two aggregation switches, that leaves 40 ports for actual aggregation of the access layer assuming a simple model as an example without Services Chassis or appliances in the mix. The oversubscription ratio facing the core for each aggregation switch in this example would be 400 Gbps/160 Gbps, which reduces to a ratio of 2.5:1.

## Traffic Flow Considerations

When managing oversubscription in a data center design, the network architect must also take into consideration likely traffic paths within the logical topology that has been created on top of the physical design. Multi-tier application flows create a portion of the traffic that does not pass from the serverfarm out towards the core, instead it passes directly server-to-server. Application specific considerations can affect the utilization of uplinks between switching layers considerably. For example, if servers belonging to multiple tiers of a given application are located on the same VLAN in the same access layer switch, these flows will stay local to the switch and not consume uplink bandwidth to the aggregation layer. If practical, this approach may be even used to contain certain types of inter-server traffic within a rack in a ToR switching traditional access model. If the same two tiers of the application are on the same Layer-2 access layer switch but on different VLANs and IP subnets, routing will be required between tiers which will result in traffic flowing to the aggregation layer and back to move between subnets.

Services devices such as server load balancers or firewalls should also be considered in planning for oversubscription. Often the raw throughput available from these devices is less than the backplane speed of the aggregation layer or services chassis, and these devices may instead become the true transport bottleneck for specific classes of flows. Depending on the service device in question and the nature of the application, other platform limitations such as connections per second may become the most critical planning criteria.

If chassis-based switches are in use, oversubscription for individual I/O modules may also come into play in the performance of the network. The 32-port 10-Gigabit Ethernet I/O module (N7K-M132XP-12) for the Nexus 7000 Series has a backplane connection of 80 Gbps into the switching fabric of the chassis. If all 32 ports are taken into consideration, the I/O module itself is providing 320 Gbps of interface bandwidth into 80 Gbps, or a 4:1 oversubscription rate local to the I/O module. This I/O module may also be used in a dedicated interface mode, where only one interface from each group of 4 ports based on the module architecture is active, effectively becoming an 8-port module that is not oversubscribed at all, but line rate into the switch fabric at each interface. Decisions on where servers providing different functional roles in the topology are connected relative to the physical switches and I/O modules deployed as well as the logical network topology can have a significant effect on the performance of inter-server traffic flows.

In large data centers supporting many different applications, it may not be practical to analyze and manage oversubscription for each individual application deployed. Awareness of these considerations in the overall design however is important in creating a flexible data center topology that does not create traffic bottlenecks for common classes of client-to-server and server-to-server traffic. These factors may also be used to optimize performance for a subset of applications that may have particularly high throughput or low latency requirements.

## Nexus 7000 Virtual Device Contexts

Cisco Nexus 7000 Series switches running NX-OS have introduced the capability to divide a single physical switch into up to four virtual switches, referred to as Virtual Device Contexts (VDCs). Each VDC operates similar to a standalone switch with a distinct configuration file, complement of physical ports, and separate instances of necessary control plane protocols such as routing protocols and spanning tree. This feature provides the potential option to use a single physical switch pair to serve multiple roles within a data center topology. Different VDC design options can use this feature for service integration, enhanced security, administrative boundaries, or flexibility of hardware deployment during changing business needs.

### VDC Hardware Overview

#### VDCs and the Nexus 7000 Supervisor Module

The Nexus 7000 allows the administrator to set the high availability (HA) policies for non-default VDCs, identifying what action should be taken when an unrecoverable fault occurs in a VDC. The default VDC cannot be reconfigured in this respect, and will always force a full supervisor reload on a single supervisor system, or a switchover on a dual-supervisor system. The VDC HA policies are also specific to whether the system is provisioned with dual supervisors or with a single supervisor.

The default HA policy for non-default VDCs on a system with a single supervisor system is to restart only the processes of the failed VDC itself. For a dual-supervisor system the default configuration is to perform a switchover to the standby supervisor for all VDCs. If certain VDCs are considered less critical or more independence is desired between VDCs, non-default VDCs may be configured as "Bringdown", which places the VDC experiencing the fault in a failed state similar to a single physical switch failure.

When VDCs are created, they are allocated system resources according to a VDC resource template, either one created by the administrator or the default template. The configurable resources include the following:

- Memory for IPv4 and IPv6 unicast and multicast routes
- Items requiring system overhead such as SPAN sessions and port channels
- Virtual items within the VDC, such as VLANs and VRFs

When designing larger data center topologies, default settings such as the maximum memory allocated to the IP routing table within a given VDC may need to be adjusted to support the requirements of the topology. Other settings such as the SPAN sessions and port channels totals are limited at the device level, so if multiple VDCs are in use a decision must be made as to which VDCs get allocation of these capabilities.

#### VDCs and Nexus 7000 I/O Modules

Cisco Nexus 7000 I/O modules provide interfaces that may be allocated to different VDCs within the chassis. Each individual port on an I/O module belongs to a single VDC. Forwarding of traffic between VDCs can only be accomplished through mutual connectivity to an external device, unless the VDCs are directly cabled together. By default, all interfaces on a new I/O module are allocated to the default VDC. From the default VDC, the system administrator can create additional VDCs and allocate the interfaces to become members of the new VDC. Once allocated to a non-default VDC, the interfaces are only exposed for configuration from within the EXEC environment of the VDC where they reside.

**Note**

The 32-port 10-Gigabit Ethernet I/O module (N7K-M132XP-12) has specific constraints based on its architecture where interfaces that are part of the same port group must be allocated to the same VDC. For details, refer to the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.1*, Creating VDCs section:

[http://www.cisco.com/en/US/partner/docs/switches/datacenter/sw/4\\_1/nx-os/virtual\\_device\\_context/configuration/guide/vdc\\_create.html#wp1165079](http://www.cisco.com/en/US/partner/docs/switches/datacenter/sw/4_1/nx-os/virtual_device_context/configuration/guide/vdc_create.html#wp1165079)

Additional resources such as memory space for Layer-3 and Layer-2 forwarding tables also reside on each I/O module. When choosing to allocate interfaces to a new VDC, the cleanest approach from a hardware resource perspective is to allocate all ports on a given I/O module to the same VDC. With this arrangement, it is clear that all resources on the module are at the disposal of one VDC, and from a support perspective if the I/O module requires replacement for any reason, it may be removed from the chassis without impacting other VDCs. However, depending on hardware constraints and how the VDCs will be used, this type of approach is not always practical. If an I/O module will be shared between two VDCs, ensure that the limits of the shared resources such as Forwarding Information Base capacity (essentially the memory to store the Layer 3 forwarding entries) are not exceeded by the combined load of multiple VDCs. The FIB capacity of the existing 32-port 10-Gigabit Ethernet (N7K-M132XP-12) and 48-port Gigabit Ethernet (N7K-M148GT-11, N7K-M148GS-11) I/O modules is able to support 56,000 IPv4 unicast routes per module as of NX-OS software version 4.1(3).

Another reason that it is desirable to allocate all ports of a given module to a single VDC is management of control plane traffic. Control plane policing may be configured within the default VDC. As of NX-OS version 4.1(3), this is not a VDC-specific feature; therefore, it applies to all VDCs on the switch. This feature helps to prevent the Supervisor Module from becoming flooded by too many control plane packets in the event of a network event such as a loop condition or a malfunctioning peer device. Control plane policing limits are applied in hardware on a per-forwarding-engine basis and operate independently on each I/O module. If a network event causes the policing limit to be reached for one VDC, it is possible that valid control plane packets belonging to another VDC could be dropped by the policing function if the two VDCs have interfaces on a common I/O module. Loss of valid control plane packets could cause interruption of proper peering between neighboring devices and possible loss of data plane traffic.

## Default VDC and VDC Management

When accessing the console port of a Nexus 7000 switch, one is automatically placed in the operating context of the default VDC. The default VDC has unique capabilities; it is from this context that additional VDCs may be created and deleted, currently up to a total of four VDCs per switch including the default VDC. Once logged into the default VDC with an account holding network-admin rights, the user may also easily change into the operating context of the other VDCs using **switchto vdc vdcname** command, and return to default by using the **switchback** command. A Telnet or secure shell terminal session into the default VDC also provides the network-admin the capability to create VDCs, allocate interfaces and resources to VDCs, and navigate to and from the operating context of other VDCs.

The IP management interface on the Supervisor Module of the Nexus 7000 Series is different from the interfaces on I/O modules. The management interface exists within all VDCs, and carries a unique IP address in each VDC. The management interface is also by default placed into a separate Virtual Routing and Forwarding instance (VRF) on each VDC called *management* to ensure that out-of-band network management traffic is forwarded according to a separate default gateway than data interfaces that are placed in the VRF called *default*. In this way, distinct management IP addresses may be provided for administration of a single VDC if needed to segregate administrative control of portions of the switch. A Telnet or secure shell terminal session directly into one of the non-default VDCs on the switch cannot navigate directly to the default VDC or other VDCs.

The default VDC has several other unique and critical roles in the function of the switch:

- System-wide parameters such as control plane policing, VDC resource allocation, and Network Time Protocol (NTP) may be configured from the default VDC.
- Licensing of the switch for software features is controlled from the default VDC.
- Software installation must be performed from the default VDC, all VDCs run the same version of software.
- Reloads of the entire switch may only be issued from the default VDC. Non-default VDCs may be reloaded independently of other VDCs as of NX-OS version 4.1(2).

If it is anticipated that a switch may be used in a multi-VDC configuration, it is recommended to reserve the default VDC for administrative functions and to configure production network connections in non-default VDC. This approach will provide flexibility and higher security. Administrative access into the non-default VDCs to perform configuration functions may easily be granted without exposing access to reload the entire switch or change software versions. No Layer 3 interfaces in the default VDC need to be exposed to the production data network, only the management interface needs to be accessible through an out-of-band (OOB) management path. Unused interfaces may be retained in a shutdown state in the default VDC as a holding area, until they are needed in the configuration of one of the non-default VDCs. The default VDC in this way may be maintained as an administrative context requiring console access or separate security credentials.



#### Note

If it is preferred to use the switch in a non-virtualized model or all four VDCs are required for production data based on design constraints, the default VDC is a fully functional context and may also be configured as part of the production network topology. If accounts are required which have access to administer only the configuration of the default VDC but should not be able to affect other VDCs, those users may be assigned accounts with only the **vdc-admin** level of privileges.

## VDC Topology Examples

Cisco has validated several topology examples that use VDCs to serve different roles in the network design. Since a VDC has similar characteristics and capabilities to a separate physical switch, these are not VDC-specific topologies and they could also be built with separate dedicated switches in the roles occupied by VDCs. The topologies shown have been validated through lab testing, but suitability for the requirements of a particular enterprise network design must be evaluated by the network architect in terms of throughput requirements, control plane load, physical constraints, and business needs. This is not a comprehensive list of all possible uses of VDCs, but merely shows some validated models and discusses some of the considerations relevant to the network architect who chooses to deploy a VDC-based topology.



#### Note

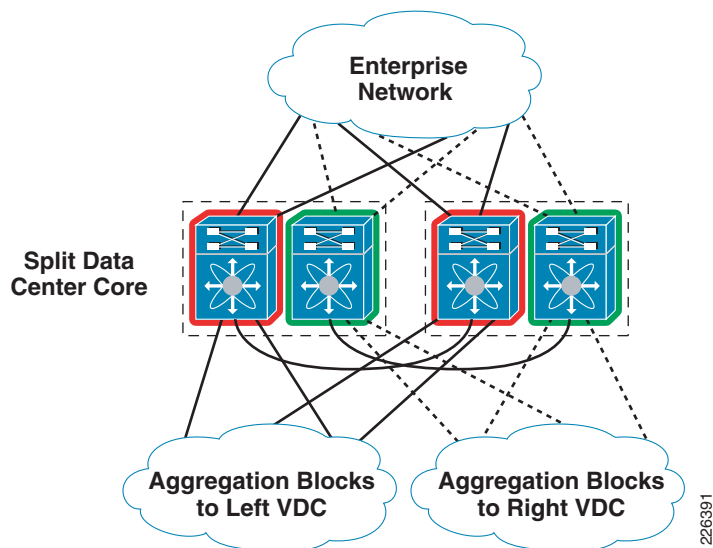
Cisco does not recommend using two VDCs from the same physical switch to construct any single layer of a hierarchical network design. Distinct, physical box redundancy within a network layer is a key characteristic that contributes to the high availability of the hierarchical network design reference model.

### Split-Core Topology

VDCs can be useful tools to facilitate migration when networking requirements are growing faster than budget, or the enterprise network needs to expand to support mergers and acquisitions. The ability to field a logically distinct pair of virtual switches through software administration and cabling without

provisioning additional power capacity and rack space provides greater agility to the network architect. The topology shown in [Figure 9](#) illustrates a single pair of Nexus 7000 switches being used to build two logically distinct data center cores.

**Figure 9** *Split Data Center Core Using VDCs*



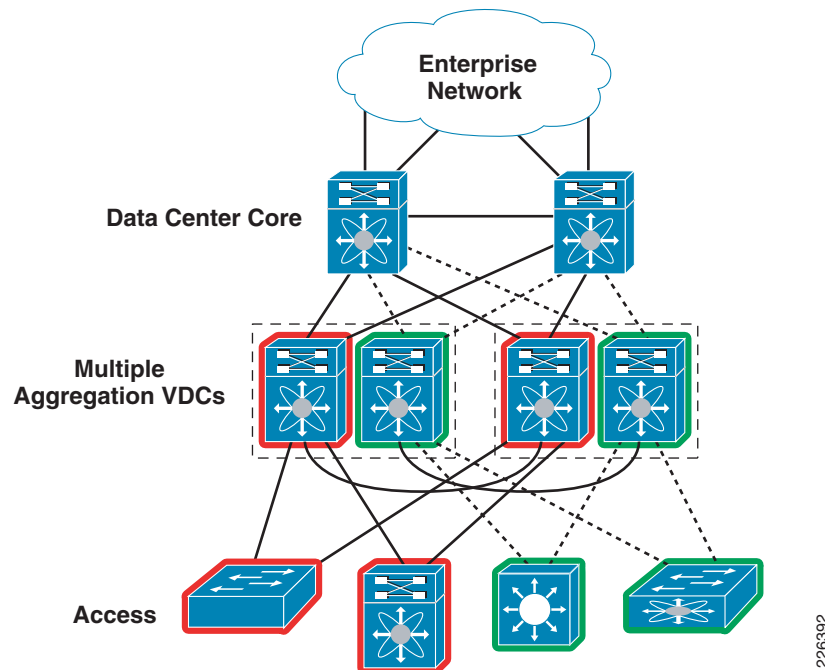
Business requirements driving such a possible need in the network architecture might include a pending merger or sale of a business unit, provisioning a separate data center topology to support application development and testing work, or providing a unique IP routing context for links to a hot-standby site located nearby. Considerations when designing this type of topology include the following:

- As a core layer, the connections to external devices should be purely Layer 3-based, so the control plane load on the device will not be required to perform processing for Layer 2 specific protocols such as STP. This specificity of function will enhance VDC stability.
- The size of the required routing tables being propagated from the enterprise core should be evaluated to ensure routing table memory capacity is adequate.
- If individual I/O modules are to be split between VDCs, the total routing capacity of the FIB TCAM on the module is shared between all VDCs with interface on the module.
- The only connectivity between the two virtual data center cores is through the enterprise network core as shown, unless separate physical cabling is provisioned to facilitate bypassing the enterprise core.
- [Figure 9](#) only illustrates two VDCs forwarding data. These may be configured as non-default VDCs, maintaining the default VDC as an administrative space.

## Multiple Aggregation Blocks

At the aggregation layer of the data center network, a single aggregation block consists of a pair of aggregation switches for redundancy and their associated access layer switches. If an enterprise has a business requirement to deploy separate aggregation blocks for different business units, the use of VDCs may be considered to accomplish this logical segregation without needing to deploy separate physical switches. Like the split core, this approach may also be used during times of growth, mergers and acquisitions, or to create a partitioned development data center network. An example of the use of VDCs to build multiple aggregation blocks from a single pair of aggregation switches is shown in [Figure 10](#).

**Figure 10** Multiple Aggregation Blocks Using VDCs



The red colored VDC in the aggregation layer of [Figure 10](#) aggregates only the red colored access switches. The green VDC only aggregates the green colored access switches, effectively creating two separate aggregation blocks that operate as if separate physical aggregation switches had been deployed. Considerations when designing this type of topology include:

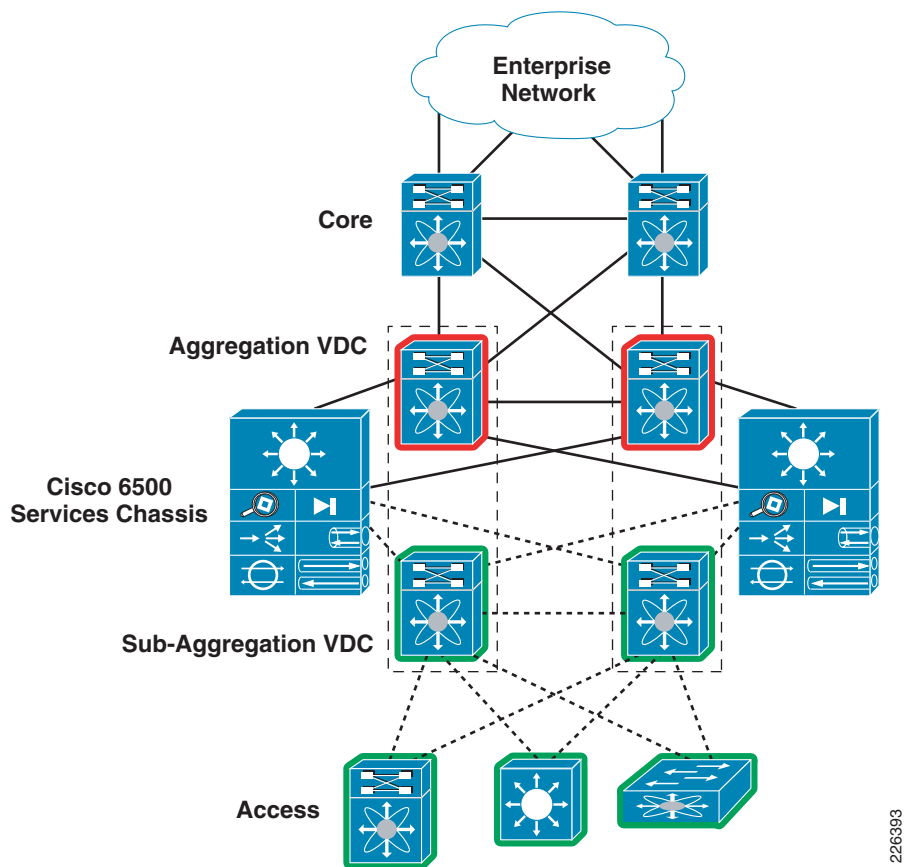
- Not only will Layer 3 control plane load be generated at the aggregation layer, but Layer 2 control plane load also. This model can work well when neither of the two VDCs is pushing the limits of its capabilities in terms of scale factors such as routing table entries and STP virtual ports.
- Allocating complete I/O modules to a VDC instead of splitting the interfaces of a given module across more than one VDC can provide greater stability and serviceability to the system. Based on the complement of hardware available in the design, this approach needs to be weighed against other best practices, such as spanning the port channel between aggregation switches across multiple I/O modules for high availability.
- The only connectivity between the red and green VDCs is through an additional Layer 3 hop through the data center core. This completely partitions the subnets and broadcast domains between the two aggregation blocks, any server moving between aggregation blocks would potentially require a new IP address.
- Depending on the I/O modules used in the aggregation of the access layer and whether dedicated mode ports are in use on a 10-Gigabit I/O module, the network administrator can now move one access switch or group of access switches between aggregation blocks without any re-cabling. (Within the constraints of I/O module architecture.)
- [Figure 10](#) only illustrates two VDCs forwarding data. These may be configured as non-default VDCs, maintaining the default VDC as an administrative space.

## Services VDC Sandwich

Data center service insertion requirements may include server load balancing devices, security devices such as firewall and intrusion prevention, and others. Multiple approaches exist for the integration of these services into the data flow. Design decisions include using modules in external Services Chassis, using appliances, and whether to run the service devices in a transparent or routed mode. One very flexible design approach is to use all services in transparent mode, but to insert an additional layer of routing instances between the serverfarm subnets and the services devices. This approach has previously been shown in design guidance using VRFs, and the deployment of multiple VRFs also provides the capability to direct traffic independently through multiple virtual contexts on the service devices, leveraging the virtualization of both the routing functions and the services devices in the design.

The VDC capability of the Nexus 7000 Series enables the network architect to leverage another type of virtualization in the design, to improve ease of configuration, supportability, and security. A secondary virtual switching layer called the *sub-aggregation* can be created using VDCs, located between the services devices and the access switches. This topology is referred to as a services VDC sandwich. An example of this topology using services modules located in external Catalyst 6500 chassis is shown in Figure 11.

**Figure 11** Services Sandwiched Between VDCs



All of the access layer switches shown in attach only to the sub-aggregation VDCs. Different classes of servers could also be attached to access-layer switches that connect directly to the main aggregation layer above the Services Chassis, if they either do not require services or are serviced by a different group of services devices. Additional considerations when designing this type of topology include the following:

- Similar designs have been deployed only using a single pair of switches with separate VLANs and VRFs to provide the routing instance below the Services Chassis. The insertion of a separate set of VDCs into the design still represents using a single physical pair of switches to perform these functions but provides better isolation between the routing environments above and below the Services Chassis. This conceptually provides for easier support and configuration, without increasing the impact of a single-switch failure due to the introduction of a second set of VDCs.
- The security model is more robust, since the operating environment of the sub-aggregation VDCs is completely separate from the primary aggregation layer. Instead of being only separate VLANs and VRFs on the same switch, they are separate virtual switches with completely different sets of processes and physical ports.
- Additional interfaces may be required for the VDC sandwich topology as compared to a VRF sandwich topology. The Services Chassis must have separate physical connections into both sets of VDCs as opposed to VLANs sharing the same trunks. Additional interface count must also be provisioned to support the inter-switch link between the two sub-aggregation VDCs.
- This model has been validated by Cisco using Firewall Services Module (FWSM) and Application Control Engine (ACE) modules running in transparent mode, where the two layers of VDCs are direct IP routing peers. Layer 3 control plane load on the VDC below the services may be limited by using static routes pointing to an HSRP address shared between the primary aggregation VDCs to support IP unicast traffic flows. IP multicast traffic is not supported over a combination of static routes and HSRP addresses, if IP multicast is a requirement then an IGP such as OSPF or EIGRP may be used.
- VDCs provide the distinction between the routing instances of the aggregation and the sub-aggregation layers, however, the use of multiple VRFs in the sub-aggregation layer may be utilized to support additional virtualization capabilities. Distinct VRFs in the sub-aggregation layer may be mapped using VLANs to separate contexts within the virtualized service devices such as the FWSM and ACE, allowing active contexts to be split between both Services Chassis. If services are required between layers of a multi-tier application architecture, placing these tiers in subnets belonging to separate VRFs will allow for powerful, multi-context service insertion between tiers.
- A services VDC sandwich using external Services Chassis provides independent connectivity between the services and both aggregation switches, if the aggregation switch on the left side of the topology fails, the services on the left side have dual connectivity and can maintain a primary role. Service appliances run in transparent mode such as the Adaptive Security Appliance (ASA) 5580 that only support single connections to carry a given VLAN will not be dual-homed if they are attached directly to the aggregation, but can still be deployed in a highly available manner by using redundant appliances.

**Note**

For more detail on the VDC services sandwich architecture with virtualized services, refer to the *Data Center Service Patterns* document at the following URL:

[http://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/DC\\_3\\_0/dc\\_serv\\_pat.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_serv_pat.html)

## Data Center Logical Topologies

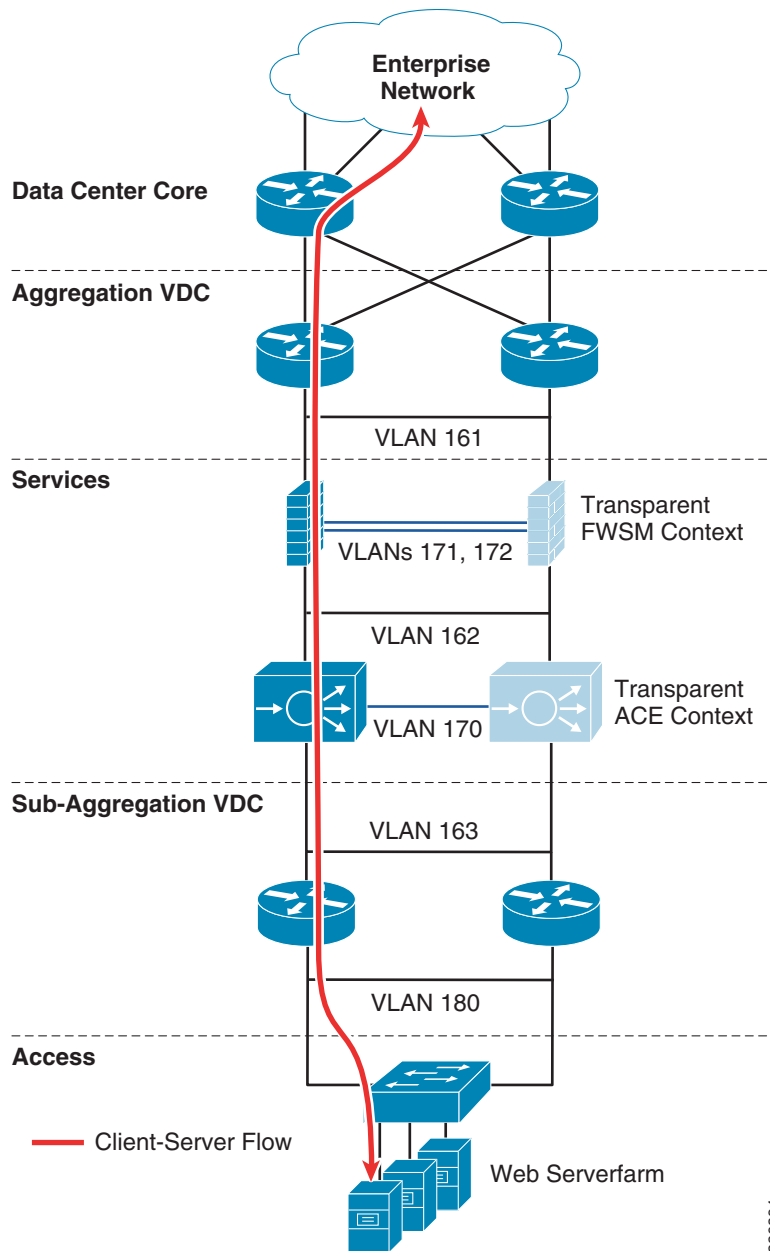
The hierarchical network design model describes a reference for physical switching layers of the network, dictates dual-homed connections for link redundancy, and presents alternatives for the insertion of data center services. This physical topology provides a flexible framework upon which to build the logical topology of the data center network, which controls flows of traffic through allocation of VLANs, subnets and routing instances. Multiple logical topologies to support a variety of specific application requirements may be configured concurrently across the same shared physical data center infrastructure. To provide examples of logical data center topology configurations, this section will use a hierarchical network design with external Catalyst 6500 Services Chassis built using a VDC services sandwich model as shown in [Figure 11](#) as a base physical topology.

### Basic Client-Server Model with Services

The most basic model for client-server computing communications is a single client communicating directly with a single server that holds all resources to fulfill the client request. An example of such an application would be a simple web server with locally stored content responding to a browser client, or a basic file server with directly attached storage. For purposes of discussion, we will reference the web server model and provision both firewall and load balancing services to scale the web services across multiple physical servers while providing additional security to the serverfarm.

The Services Chassis in this model provides purely Layer-2 connectivity to forward VLAN traffic between the aggregation layer VDCs, the service modules, and the sub-aggregation layer VDCs. Using IP routing in the sub-aggregation layer VDCs also creates a Layer-3 boundary for serverfarm broadcast domains from the access layer, and allows the sub-aggregation layer to provide default gateway services for the serverfarm IP subnets. This basic client-server logical topology including both firewall services from FWSMs and load balancing services from ACE modules is illustrated in [Figure 12](#). Implementing the services in a transparent mode allows a model that can easily be extended to a multi-context configuration without altering the routing topology of the existing service contexts.

Figure 12 Basic Client-Server Model with Services



## Multi-Tier Application Model

The basic client-server model serving one set of serverfarm subnets may be extended to support multi-tier applications. For example, consider extending the simple web server example into a two-tiered application with a web/application front end serverfarm that requires a layer of firewall services to provide security for a clustered backend database application. The logical topology to serve this application may easily be created on the same physical infrastructure already in place. To provision this topology, migrate to a fully virtualized services model with the addition of new services contexts on the FWSM and ACE modules, and addition of separate VRF instances within the sub-aggregation VDC layer. Serverfarm subnets that are terminated on separate pairs of VRF instances that do not peer directly

at Layer-3 will need to pass through their respective services layers to the main aggregation VDC in order to communicate with one another. The addition of VRFs in this design is required to keep the multiple tiers of the application segregated between services contexts.

Figure 13 Two-Tier Application Model with Services

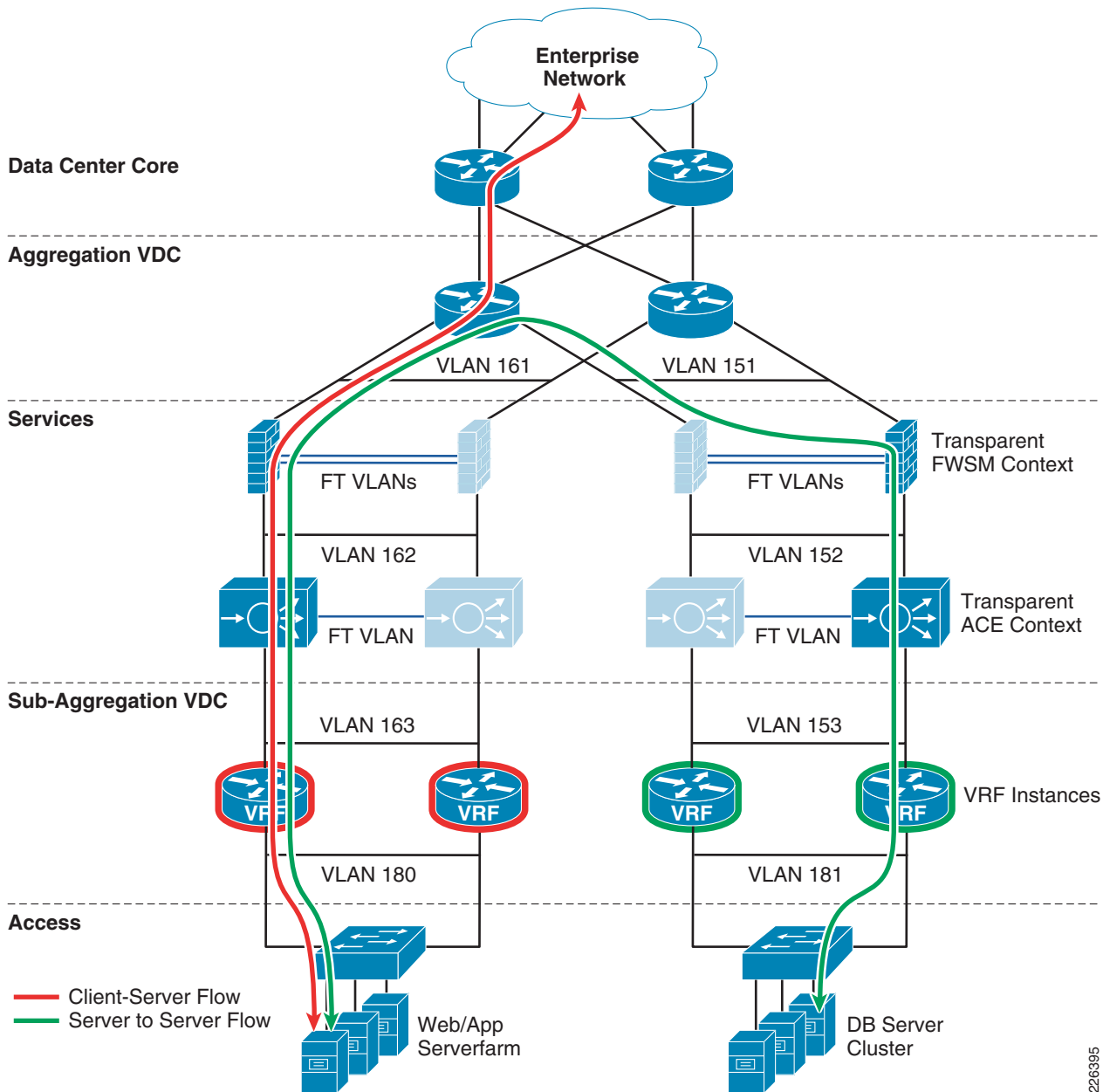


Figure 13 illustrates an example of a logical topology to support a two-tier application. In this example, the database server cluster and web/application serverfarm must live on separate IP subnets correlated to the different VRF instances to force the server-to-server traffic through the services modules. The separate services contexts supporting the database server cluster are shown as running their primary context in the right side of the physical topology as opposed to the left side, along with the VRF instance providing primary HSRP default gateway for the subnet. In a virtualized model these service contexts may be set as primary active on different sides of the physical network, since each set of virtual contexts

and VRFs represents a separate logical topology overlaid upon the physical infrastructure of the data center. This allows processing load to be distributed in a dual-active manner across both sets of services modules on a per-context basis, while still providing the fault tolerance of redundant physical hardware.

**Note**

A detailed description of this logical topology including configuration examples is available in the *Data Center Service Patterns* at the following URL:

[http://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/DC\\_3\\_0/dc\\_serv\\_pat.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_serv_pat.html)

**Note**

To build this topology using transparent services devices and Nexus 7000 switching in the aggregation layer, separate VDCs should be used above and below the services to ensure that the VLAN interfaces configured carry unique MAC addresses on the Layer-2 broadcast domains. NX-OS 4.1 and earlier releases do not allow static MAC address configuration on VLAN interfaces.

## Layer 3 Design and Features

### Overview

Layer 3 IP routing configuration is required in the data center core and aggregation layers, and potentially also in conjunction with services depending on the logical topology design. Some of the common Layer 3 features required in the data center include the ability to run an Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Enhanced Interior Gateway Routing Protocol (EIGRP), IP multicast routing support using Protocol Independent Multicast (PIM), and the ability to provide first-hop gateway redundancy with a protocol such as Hot Standby Router Protocol (HSRP). This section describes best practices configuration for Layer 3 features in a typical hierarchical data center topology, with configuration examples provided primarily using NX-OS version 4.1.3 syntax for reference, except where noted.

### Layer 3 Feature Best Practices

#### IP Route Summarization

Routing protocol summarization is a common IP networking practice used to keep routing tables small for faster convergence and greater stability. In the data center hierarchical network, summarization may be performed at the data center core or the aggregation layer. Summarization is recommended at the data center core if it is a dedicated layer that is separate from the enterprise core. The objective is to keep the enterprise core routing table as concise and stable as possible to limit the impact of routing changes happening in other places in the network from impacting the data center, and vice versa. If a shared enterprise core is used, summarization is recommended at the data center aggregation layer. In order to enable summarization, proper IP address allocation must have been used in the assignment of subnets to allow them to be summarized into a smaller number of routes.

OSPF uses the **area<area-id>range** command under the router configuration to summarize addressing at an area border router (ABR). OSPF area range statements can be configured and are displayed using a slash with a number representing the length of the network address as opposed to a mask (for example, 10.8.128.0/18). Use of the slash syntax as an alternative to dotted decimal network masks is an option throughout the NX-OS configuration.

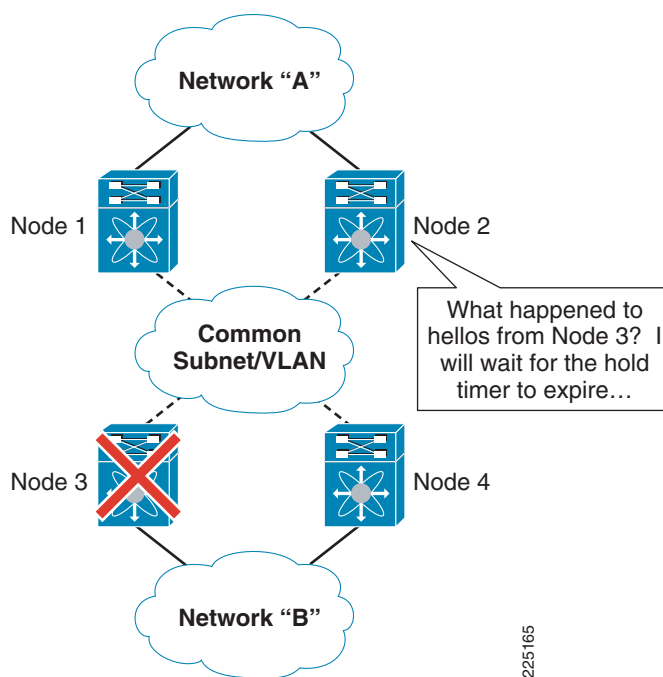
Cisco NX-OS supports EIGRP summary aggregate addresses at the interface-level using the **ip summary-address eigrp** command. EIGRP auto-summarization is an IOS feature that is commonly disabled by network administrators when EIGRP is configured. EIGRP auto-summarization is not supported with NX-OS.

## IGP Hello and Dead/Hold Timer Settings

### Routing Peer Failure Triggers

Common design guidance for configuration of IGP such as EIGRP and OSPF in the enterprise can include some tuning of default hello and dead/hold timers. The hello timer is effectively a keepalive packet between routing protocol neighbors that assures a routing instance that the given neighbor is still there. If no hello packets are received from a specific neighbor for a period of time as defined by the *dead* timer in OSPF or *hold* timer in EIGRP, the neighbor is considered to be down and the adjacency must be removed. When routers peer across a shared Layer 2 cloud or VLAN with multiple end nodes as shown in Figure 14, the expiration of a dead or hold timer can be the key value that determines when a given neighbors routes are removed from the table during an outage. For example, if Node 3 in Figure 14 fails or its physical link to the Layer 2 cloud, Nodes 1 and 2 need to wait for its dead/hold timer to expire before removing Node 3's routes to Network B from their tables.

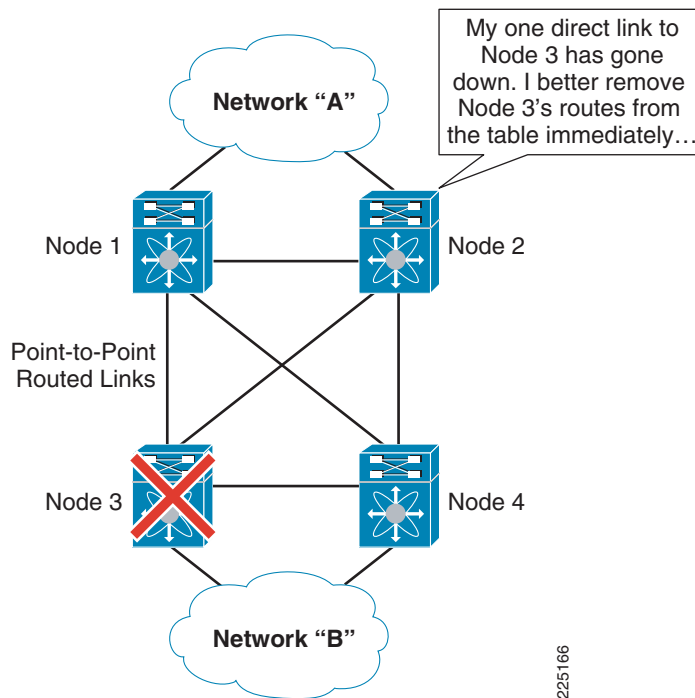
**Figure 14** IGP Peering Across Shared Layer 2 Cloud



If hierarchical network design best practices are followed with the Layer 2 and 3 boundary at the aggregation layer, the dead/hold timer for an IGP is normally not the value that triggers a failover of routing paths in the topology. The core and aggregation layer should be connected by a full-mesh of

point-to-point routed links as shown in Figure 15. Each link carries a separate IP subnet and is a dedicated fiber connection or Layer 3 port channel between switches so that if a switch at one end fails or the link itself completely fails, the remaining active node(s) immediately removes the associated routes from the table, since the only physical path leading to the next-hop neighbor is down.

**Figure 15** IGP Peering Across Point-to-Point Links



If all routing nodes that are peering with a dynamic IGP use dedicated point-to-point links for their communication, tuning the hello and dead/hold timers to smaller values is not required to decrease convergence time and reduce packet loss during device or link failures.

### IGP Timers for Single-Supervisor Systems

While some Cisco platforms support tuning of hello and dead timers down into the millisecond range, such aggressive timers may add significant control plane load in a data center environment where hundreds of VLANs and Layer 3 interfaces might exist in a single switch. NX-OS supports tuning of hello timers as low as one second, and dead/hold timers as low as 3 seconds for OSPF and EIGRP. These current configuration minimums have been validated using Nexus 7000 switches with single-supervisor systems in the aggregation layer of a data center topology. Tuning these timers only improves convergence times if the data center infrastructure includes devices that IGP peer over a Layer 2 cloud. A common example of this is a data center service such as a firewall that may be implemented in a routed mode running an IGP.

### Router and Interface Definition

A traditional best practice recommendation for routing protocol configuration with Cisco IOS in the data center is to use the **passive interface default** command, this allows the administrator to only enable the routing protocol on specific interfaces as opposed to having all interfaces covered that are included in the scope of the **network** command. NX-OS takes a slightly different approach to interface assignment in routing protocol configuration.

Creation of the routing instance in NX-OS is handled identically to Cisco IOS using the **router<eigrp|ospf><as number|process id>** command. To enable routing on specific interfaces, each Layer 3 interface to be active in the routing protocol must be identified under the interface configuration using the **ip router<eigrp|ospf><as number|process id>** command. EIGRP by default requires very little further configuration. Other common routing protocol commands for OSPF areas and summarization are very similar to Cisco IOS.

**Note**

Additional detail for advanced configuration of routing protocols in NX-OS may be found in the Cisco *NX-OS Unicast Routing Configuration Guide*, Release 4.0 at the following URL:  
[http://www.cisco.com/en/US/partner/docs/switches/datacenter/sw/4\\_0/nx-os/unicast/configuration/guide/l3\\_nxos-book.html](http://www.cisco.com/en/US/partner/docs/switches/datacenter/sw/4_0/nx-os/unicast/configuration/guide/l3_nxos-book.html)

The following is an example of OSPF router configuration and interface definition using the **ip router interface** command:

```
router ospf 8
  router-id 3.3.3.1
  area 81 nssa
  redistribute static route-map STAT-MAP
  area 0 range 10.8.0.0/18
  area 81 range 10.8.128.0/18
  area 0 authentication message-digest
  timers throttle spf 10 100 5000
  timers throttle lsa router 1000
  timers throttle lsa network 1000
  auto-cost reference-bandwidth 10000
interface Ethernet1/1
  description <to core1>
  ip address 10.8.1.2/24
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip ospf dead-interval 3
  ip ospf hello-interval 1
  ip router ospf 8 area 0
  no shutdown
interface Vlan128
  no shutdown
  ip address 10.8.128.3/24
  ip ospf passive-interface
  ip router ospf 8 area 81
```

Many interfaces at the aggregation layer are VLAN interfaces, which serve as the default gateways for serverfarm VLANs. It is desirable to have the subnets of these interfaces included in the routing table, but not to have the IGP peering across every routed interface through the access layer. To eliminate this undesirable peering in NX-OS, add the **ip<ospf|eigrp>passive-interface** command under the interface configuration, as shown in the configuration of interface VLAN 128 above.

**Note**

Many capabilities in Cisco NX-OS are not enabled by default, and must be specifically turned on before their configuration syntax will be available to the administrator. This approach is referred to as *conditional services* and is a by-product of the granular process and feature-level modularity in NX-OS. Examples of these features used in validation testing include EIGRP, OSPF, VLAN interfaces, HSRP, PIM, and LACP.

## Routing Protocol Authentication

Requiring authentication of IGP peers with a pre-shared key helps provide basic protection against the data center routing table being populated by routes from either malicious or inadvertent devices. OSPF and EIGRP configuration of authentication is virtually identical in both Cisco NX-OS and IOS. One difference is that the authentication keys are encrypted by default in the display once configured in NX-OS, instead of being shown in clear text.

The following example shows an interface in NX-OS configured for OSPF with authentication:

```
interface Ethernet1/1
  description <to core1>
  ip address 10.8.1.2/24
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 3 b2255cb5a7107f1b
  ip ospf dead-interval 3
  ip ospf hello-interval 1
  ip router ospf 8 area 0
  no shutdown
```

The following example shows an interface in NX-OS configured for EIGRP with authentication. The key chain definition must be globally configured on the switch first before being assigned to the individual interface:

```
key chain eigrp
  key 7
    key-string 7 070c285f4d06

interface Ethernet1/1
  description <to core1>
  ip address 10.8.1.2/24
  ip router eigrp 8
  ip authentication mode eigrp 8 md5
  ip authentication key-chain eigrp 8 eigrp
  ip hold-time eigrp 8 8
  ip hello-interval eigrp 8 2
  no shutdown
```

## OSPF Reference Bandwidth

The default reference bandwidth of 100 Mbps used in Cisco IOS is a legacy from prevalent network speeds at the time the OSPF version 2 protocol was originally developed. This reference bandwidth results in 10-Gigabit, 1-Gigabit, and 100 Mbps interfaces to have the same cost. A best practice in router configuration is to set the reference bandwidth to a more reasonable number in the context of currently available link speeds. By configuring the reference bandwidth to a larger number such as 10,000 Mbps, it is the equivalent of 10 Gigabits. Therefore, a 10-Gigabit Ethernet interface will have a cost of 1 and a 1-Gigabit interface will have a cost of 10.

Cisco NX-OS automatically implements a more reasonable reference bandwidth by default of 40,000 Mbps. This value provides greater flexibility with the development of 40 Gbps and 100 Gbps interfaces on the horizon in the Nexus 7000 platform. In a data center network with both NX-OS and IOS devices, the reference bandwidth setting should be adjusted so that all devices within an OSPF area use a consistent value. Configuration of reference bandwidth for OSPF in Cisco NX-OS is identical to Cisco IOS; it is done with the use of the **auto-cost reference-bandwidth** command.

## OSPF Throttle Timers

OSPF configuration provides the ability to customize the default timers that control the pacing of SPF algorithm execution and LSA pacing. In Cisco IOS, it is recommended to reduce these values from their default values to improve network convergence, while assuring that multiple iterations have a dampening effect applied to reduce processor utilization if intermittent or flapping connectivity occurs in the network.

In Cisco NX-OS, the default SPF timers have been significantly reduced. Common deployments of NX-OS platforms are in a high-speed data center requiring fast convergence, as opposed to a wide area network (WAN) deployment with lower speed links where slower settings might still be more appropriate. To optimize OSPF for fast convergence in the data center, the default throttle timers in NX-OS can be updated using the same command syntax (**timers throttle spf 10 100 5000**) as the Cisco IOS under the router definition. This is consistent with design guidance provided for the Cisco Catalyst 6500s running Cisco IOS and improves convergence times in a hierarchical network design over the default values.

Cisco NX-OS supports slightly different command syntax for manipulating LSA timers. Cisco NX-OS supports the manipulation of the LSA hold interval at both the network and router level. In validation cycles, the OSPF LSA hold timer was reduced at the network and router levels from the default value of 5000ms down to 1000ms, using the **timer throttle lsa<router|network> 1000** commands.

## First Hop Redundancy

Most end-node systems such as servers do not run a routing protocol, but instead support configuration of a single-destination IP default gateway address for traffic destined off of the local subnet. To provide redundancy for IP default gateway services, several protocols exist which are commonly referred to together as First Hop Redundancy Protocols (FHRPs). Cisco NX-OS supports implementations of multiple FHRPs: Hot Standby Router Protocol (HSRP), Gateway Load Balancing Protocol (GLBP), and Virtual Router Redundancy Protocol (VRRP). Configuration differences and best practices for use of HSRP as the FHRP on Cisco NX-OS are provided in the following subsection.

### Hot-Standby Router Protocol

In the classic hierarchical network design with the Layer 2/3 boundary at the aggregation layer, IP default gateway services are provided for servers and end-nodes on Layer 3 VLAN interfaces configured with HSRP. In addition, for services integration or other instances where static routing is used, a single next-hop address can be configured on static routes and that next-hop address is shared between two redundant physical devices running HSRP. In some cases, it is desirable to have multiple next-hop HSRP addresses active between different pairs of switches on the same subnet. HSRP groups can be used to differentiate multiple instances of HSRP on a single IP subnet. Use of HSRP authentication is also recommended to help ensure that only the intended devices running HSRP on a given subnet are able to establish a relationship.

HSRP uses a priority mechanism to define which interface is active for the shared HSRP IP address and its associated MAC address. A best practice for configuration is to set the highest priority HSRP interface in the same switch where the spanning tree root is defined for a given VLAN. This allows Layers 2 and 3 forwarding paths to be aligned, reducing unnecessary use of the link between aggregation switches. Preemption is an optional configuration that allows this highest priority interface to resume active status when recovering from an outage, where a lower priority redundant interface had assumed active responsibilities. Delay can also be configured on the preemption, in this example 180 seconds of delay is used to provide some time for the switch to fully complete a bootup process and ensure that all interfaces are active before asserting preemption to claim active HSRP status on an interface. The full

duration of bootup time may vary in a specific configuration. To ensure that adequate preemption time is provisioned when using this parameter, the typical time to boot and enable all interfaces and protocols with the hardware and configuration in question should be validated before configuration.

HSRP configuration syntax and concepts are very similar between Cisco IOS and Cisco NX-OS. NX-OS uses the command **hsrp**<group id> instead of **standby**<group id>. Instead of requiring that the keyword be reiterated by the administrator prior to entering each HSRP command, NX-OS implements an HSRP configuration mode under the interface so that all associated commands can be entered directly and is displayed with an indent under the **hsrp** keyword and group number. The following is an example of IOS-based HSRP configuration from a Cisco Catalyst 6500:

```
interface Vlan128
 ip address 10.7.128.3 255.255.255.0
 ip pim sparse-mode
 ip igmp version 3
 standby 1 ip 10.7.128.1
 standby 1 timers 1 3
 standby 1 priority 20
 standby 1 preempt delay minimum 180
 standby 1 authentication c1sc0
```

The following is an HSRP configuration of a comparable interface in NX-OS:

```
interface Vlan128
 no shutdown
 ip address 10.8.128.3/24
 ip ospf passive-interface
 ip router ospf 8 area 81
 hsrp 1
 authentication c1sc0
 preempt delay minimum 180
 priority 20
 timers 1 3
 ip 10.8.128.1
```

HSRP in NX-OS also supports object tracking that allows dynamic alteration of HSRP interface priority based on the status of a tracked object. An example of a tracked object is a local interface in the switch or the availability of an IP route. One possible application of this feature is to track the logical interfaces to the core of the network and decrement HSRP priority if interfaces fail. Ideally, the connectivity to the core should be built with redundant port channels spread across multiple I/O modules, so that no single port or I/O module failure can cause isolation of the aggregation switch from the core of the network.

## HSRP Timer Configuration

Common guidance for optimization of HSRP for fast failover is to reduce the hello and hold timers from their defaults of 3 and 10 seconds, respectively. NX-OS does support HSRP version 2 with millisecond timers; however, a hello timer of 1-second and hold timer of 3 seconds provides fast failover without creating too high of a control plane load in networks with a large number of VLAN interfaces. Also, when using hello and hold timers that match those of the routing protocol, the default gateway services failover with similar timing to the IGP neighbor relationships. HSRP hello and hold timers of 1 and 3 seconds are recommended for fast failover, and were validated in support of this document with 175 active HSRP interfaces configured.

## IP Multicast

Cisco NX-OS supports IP multicast routing by default and does not require the **ip multicast routing** command used in Cisco IOS. Protocol Independent Multicast (PIM) works in conjunction with the unicast routing table and Rendezvous Point (RP) definition to manage forwarding of multicast traffic

from sources to receivers. PIM is enabled on the interface level in NX-OS using the **ip pim sparse-mode** command. Dense-mode PIM is not typically recommended for enterprise networks because of its periodic flooding behavior and is not supported in NX-OS.

When configuring IPv4 sparse mode PIM for multicast support, all of the routers in the local multicast domain need to be aware of the address of a Rendezvous Point (RP) for the group addresses to be used for communication. Using either Auto-RP or Bootstrap Router (BSR) is recommended to provide automated discovery of the RP's IP address instead of manual configuration. Anycast-RP may also be used as an approach to allow two or more routers to share RP responsibilities and provide redundancy.

IP multicast traffic can pose challenges for services integration in the data center. If multicast traffic is routed through a services layer, the simplest configuration is to keep service devices in transparent mode. If services are integrated using dynamic routing on services devices or with a Services Chassis MSFC, multicast traffic will flow, but specific attention should be focused on the configuration of the services devices to ensure that replication of multicast traffic does not place undue load on the devices when multiple interfaces are configured. If static routes pointing to HSRP addresses as next-hop are used for services integration, multicast traffic must be configured to bypass the services layer because the use of static routes directed to HSRP addresses is not supported for multicast forwarding.

## Layer 2 Design and Features

The recommended designs all include a Layer 2 interconnect between the aggregation and access layers. This part details the reasons why bridging is desirable in the data center, while summarizing the dangers associated with this technology.

Two different kinds of solutions have been validated, both using the latest features introduced on Cisco platforms:

- Redundancy handled by the STP, enhanced with bridge assurance and the dispute mechanism
- Redundancy provided by distributed channeling technology, using the Catalyst 6000s VSS or the Nexus 7000 vPC.



### Note

The use of Layer 2 for Data Center Interconnection (DCI) between different aggregation blocks was not validated for this document.

## Using Layer 2 at the Aggregation and Access Layer

There are many arguments in favor of using large Layer 2 domains in a data center. One of the most compelling reason why bridging is predominant in the data center is because several applications or services rely on it. The list includes server clustering for HA and load balancing, Layer 2 keepalives between appliances, database clustering, virtual server failure recovery, and others. But there is an even more fundamental justification for keeping bridging in a routed world: Layer 2 networking operates below Layer 3. This truism is what really makes transparent bridging so attractive. Introducing some bridging functionality in a router is the only way of connecting several hosts in a single subnet to it. If furthermore this subnet needs to spread across different physical devices, a bridge will likely be required. If on the top of that redundancy is expected, not only a bridge but some kind of Layer 2 protection mechanism will be necessary. Virtualization is one of the most promising technologies being currently introduced in the data center. Server virtualization allows multiple instances of server operating systems and/or applications to run on a single server hardware platform. Virtualized servers are called *virtual machines* (VMs). A virtual machine can be stopped and moved across the network almost instantly. When this virtual machine is reactivated, it expects to be able to communicate immediately using the IP

address that it was using in its previous location. Bridging is currently the best way of achieving this, because it operates transparently to the routing layer. Layer 2 provides flexibility and mobility from the perspective of Layer 3.

There are also good reasons why to avoid Layer 2 in the network. The traditional way of doing transparent bridging requires the computation of a spanning tree for the data plane. *Spanning* means that there will be connectivity between any two devices that have at least one path physically available between them in the network. *Tree* means that the active topology will use a subset of the links physically available so that there is a single path between any two devices ( i.e., there is no loop in the network). Note that this requirement is related to the way frames are forwarded by bridges, not to the Spanning Tree Protocol (STP) that is just a control protocol in charge of building such a tree. Unless a specific entry exists in the filtering database (also known as MAC address table or CAM table), a bridge will flood a frame to all the ports belonging to the active topology. Again, because the topology is a tree, this behavior will result in a single copy being delivered to all the nodes in the network.

This approach has the following two main drawbacks:

- Network-wide failure domain—Unless filtered frames are flooded. A single source can send traffic that is propagated to all the links in the network. If an error condition occurs and the active topology includes a loop, because Ethernet frames do not include a time-to-live (TTL) field, traffic might circle around endlessly, resulting in network-wide flooding and link saturation.
- No multipathing—Because the forwarding paradigm requires the active topology to be a tree, only one path between any two nodes is used. That means that if there are  $N$  redundant paths between two devices, all but one will be simply ignored. Note that the introduction of a per-VLAN trees allows working around this constraint to a certain extent.

Port channels are a way of introducing redundancy and load-balancing while keeping a logical spanning tree in the data plane. The vPC and VSS are Cisco enhancements that reduce the constraint associated with port channels, and their application to the design of a data center will be introduced in [“Loop Free” Layer 2 Design” section on page 51](#).

Transparent interconnection of lots of links (TRILL) will be the first technology supported by Cisco that really breaks away from the spanning tree-based frame forwarding by *routing* frames at Layer 2. TRILL is not presented in this document. TRILL provides a flexible Layer 2 with the reliability of Layer 3.

## Traditional STP Design

This section introduces some general rules on spanning tree stability. The latest tools, like the dispute mechanism and bridge assurance, are presented along with the other relevant features. Eventually, the example of a possible network design is used to offer the latest recommendation.

### STP Stability

STP is a very mature protocol, benefiting from years of development and production deployment. However, STP makes some assumption on the quality of the network and can fail. Those failures are generally high profile because of the extent to which they impact the network. This part summarizes the reasons why STP is more fragile than its Layer-3 control protocol counterparts, and introduces some rules aimed at reducing the risk associated with this technology.

### Differences Between Routing and Bridging

Transparent bridging is the result of a long technological evolution that was guided by the desire to keep the property of the thick coaxial cable that was the base for the original Ethernet networks. *Transparent* means that the stations using the service are not aware that the traffic they are sending is bridged; they

are not participating in the bridging effort. The technology is similarly transparent to the user, and a high end Ethernet switch running STP is still supposed to be plug-and-play, just like a coaxial cable or a repeater were. As a result, unlike routers, bridges have to discover whether their ports are connected to peer bridges or plain hosts. In particular, in the absence of control message reception on a port, a bridge will assume that it is connected to a host and will provide connectivity. Therefore, the most significant differences between routing and bridging with STP are as follows:

- A routing protocol identifies where to send packets.
- STP identifies where not to send frames.

The obvious consequence is that if a router fails to receive the appropriate updates, the parts of the network that were relying on this router for communication will not be able to reach each other. This failure tends to be local, as the communication within those distant network parts is not affected. If a bridge misses control information, it will instead open a loop. As it has been observed, this will most likely impact the whole bridging domain.

## STP Failures

### Two Kinds of Failures

The essential goal of the STP is to build an active topology for bridging that is a spanning tree. As a result, the following two kinds of failures are encountered:

- The failure to produce a "spanning" topology. This is what happens when the protocol ends up blocking too many ports. This failure is relatively simple to troubleshoot. This is a matter of exploring the path between devices that should have connectivity and checking why STP blocks a port on it.
- The failure to compute a "tree". This is a well known and much more severe problem. As observed, it generally has network-wide consequences, which makes the identification of the source of the issue all the more difficult.

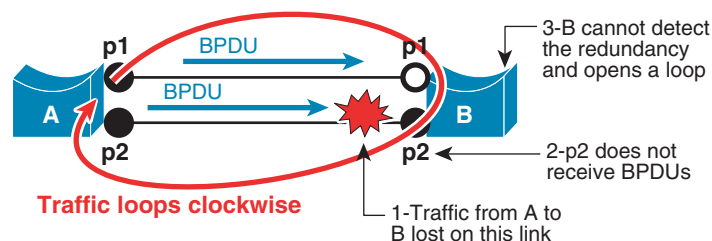
### How STP Opens a Loop

STP is designed to never open, even temporarily, a loop during its operation. However, like any protocol, it is based on some assumptions that might not be true in the network. Below are some known problems that can lead to a loop in a redundant network protected by the STP:

- Unidirectional link failure in the network—STP assumes that a link is bidirectional, which means that if the link is up, traffic can go both ways. This property is not always satisfied, particularly on fiber links. The Bridge Protocol Data Unit (BPDU) that would lead the redundant port to be blocked could be lost because of such a failure, resulting in a loop (that only occurs in the other direction).

**Figure 16** below provides an example of a loop resulting from a uni-directional link. Bridge B has the responsibility of blocking the bottom link between bridges A and B. However, this link drops traffic sent from bridge A toward bridge B, and in particular, p2 on bridge B does not receive any BPDU. As a result, bridge B does not detect the redundancy and opens both of its ports, leading to a traffic loop (clockwise only, as the link is uni-directional).

**Figure 16 Loop Induced by a Uni-directional Link**



- A "brain-dead" bridge in the network—This is a device or group of devices that forward data traffic but terminate BPDUs. STP was first designed in a time when bridges were software-based devices, and if the CPU was not able to handle the control protocol, it was not able to handle the data traffic either. Modern switches implement STP in software (running on a CPU), while traffic is generally handled in hardware by ASICs. It is then possible that the bridge is not able to run STP while it is still able to forward traffic. A bridge that would not be able to run STP would be invisible from its peers and considered as a host, which could result in a loop. This condition could occur because of a bug, because the spanning tree process has crashed, because the spanning tree process is not able to exchange BPDUs with its neighbor (queues stuck), or because the CPU is over utilized, for example. In fact, any device that drops BPDUs while transmitting traffic will almost certainly open a loop.

These failures that can result in STP opening a loop are always related to BPDUs not being sent, being lost, or ignored. This is because, in order to detect a redundant link, STP needs to continuously receive control information through this link.

#### Rules for STP Network Stability

Most of the design recommendations for STP are directly targeted at reducing the risks mentioned above. This should be mainly to:

- Minimize the number of alternate ports in the design (i.e., reduce dependency on STP).
- Restrict the freedom of STP to introduce forwarding ports in the topology. Several features, described below, allow falling back to an STP discarding state when some policies are violated, introducing a safer Layer 3-like behavior.

## STP Features

New features for increased spanning tree stability have been integrated in the design of NX-OS and recently added in IOS. This part introduces the dispute mechanism and bridge assurance, but also gives an overview of the existing Cisco spanning tree tools relevant to the data center, with an emphasis on how they differ from the new features.

### Portfast (Edge Port)

This feature is probably the best known and the most important spanning tree option. It was introduced more than 10 years ago by Cisco and allows STP to skip the listening and learning stages on ports connected to hosts. When adopted by the IEEE, the concept of portfast was renamed as *edge* port. Cisco command line interface was recently updated to reflect use of this new terminology:

- Nexus—spanning-tree port type edge
- IOS—spanning-tree portfast edge

Portfast is not a stability feature. It is only mentioned in this document to warn about the CLI change and to stress how important it is for RSTP and MST. During their normal operation, those protocols may have to ensure that all the ports of a bridge are in *sync*. This *sync* status may require moving the non-edge ports to a discarding state. Ports that are connected to an RSTP/MST peer can then instantly move away from this discarding state using the proposal/agreement mechanism. However, ports connected to hosts (like stations or routers) have no RSTP/MST peer and will only transition slowly to forwarding if not explicitly designated as portfast edge. Thus, identifying edge ports is critical to the efficient operation of MST and RSTP.

## Loopguard

In a spanning tree network, a part of any BPDU exchanged between bridges is specific to the root bridge. Non-root bridges receive the root information on their "root" port, and propagate it down toward the edge of the tree through their "designated" ports. On a given segment, the designated port is the port that has the best spanning tree information, and it keeps advertising this information as long as it is designated (i.e., as long as no other port is able to beat it). [“How STP Opens a Loop” section on page 36](#), provides some example of problems that can lead to a bridging loop. The scenarios described have in common that a port supposed to handle the redundancy fails to receive BPDUs and transitions to a designated role. Assuming an initial stable state, preventing those failures should be as simple as preventing any port from entering the designated role. But of course, there are many legitimate reasons for a port to enter the designated role during normal STP operation. Loopguard is a feature that is only restricting, not preventing, the ability of a port to move to designated forwarding.

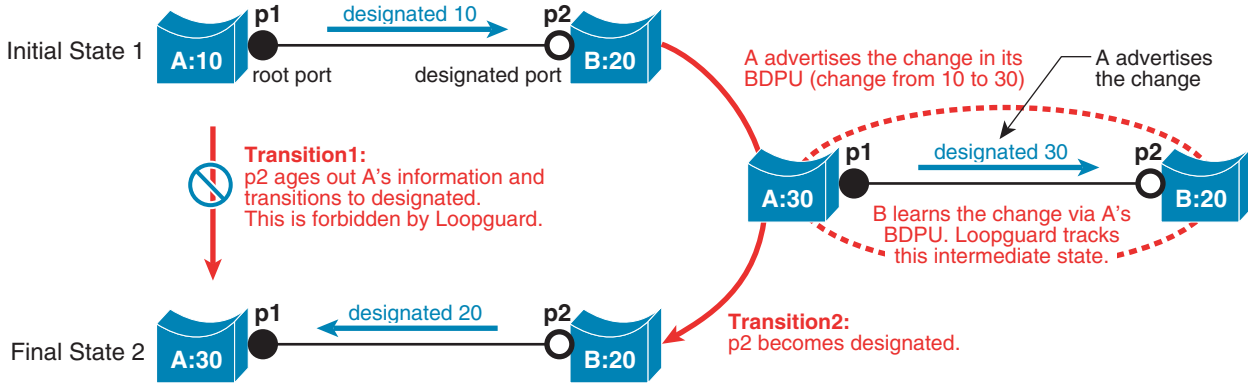
Let us focus on the case of a point-to-point link, a segment with only two ports p1 and p2, where p1 is the designated port. As long as p1's information is better than p2's, p1 should keep sending BPDUs. This condition can be only exited if:

- p2's information becomes better than p1's, in which case p2 becomes designated and starts sending BPDUs to p1.
- p1's information becomes worse than p2's. In that case, p1 will transmit a BPDU with its updated information and p2 will become designated.

This means that as long as p2 is not designated, it must keep receiving BPDUs from p1. This property is the base of Loopguard's consistency check. On a point-to-point link, a non-designated port will revert back to discarding should it fail receiving the periodic BPDUs from the designated port.

[Figure 17](#) below illustrates how Loopguard restricts the ability of a port to move to designated port to forwarding. Transition1 would be the result of p2 failing to receive any BPDUs from p1, most likely because of a uni-directional link failure or brain-dead neighbor. This dangerous transition is prevented by Loopguard. On the other hand, should A or B's information change, it is perfectly normal for p2 to become designated. Transition2 illustrates the result of A sending worse information after a configuration or a change in the topology.

**Figure 17 Loopguard Consistency Check**



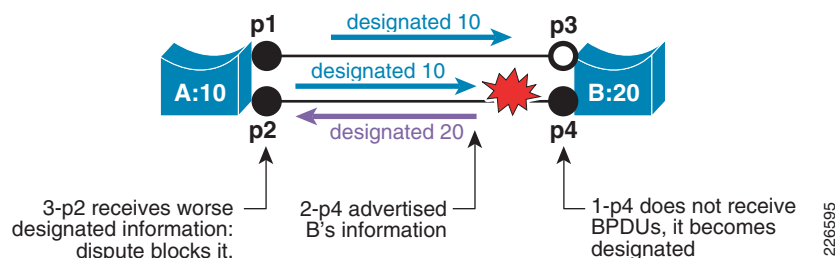
Loopguard is a very efficient mechanism because it protects from both the unidirectional link and the brain dead bridge failure cases. Note that port p2 can only check on the health of the designated port p1 if it has received at least one BPDU from p1. If an error condition existed at linkup, port p2 would never have detected the existence of p1 and would have transitioned to designated forwarding. In a typical spanning tree way, the weakness of Loopguard is linked to its plug and play aspect: port Y has no knowledge of its connection to another spanning tree port and must first detect it by receiving a BPDU.

**Dispute Mechanism**

RSTP officially introduced the concept of port roles. A port can be root, alternate, backup or designated. On a particular segment, as mentioned above, the port that is sending the best BPDU is the designated port. This port connects the segment to the root bridge. There should be a unique designated port on a given segment; if there were two designated ports on a segment, this segment would be introducing a forwarding loop as it would have two active paths to the root. This rule allows a powerful consistency check in RSTP and MST because the BPDU has been enhanced to carry the role of the port that transmitted it.

Figure 18 is just an enhanced version of Figure 16 on page 37 that shows the BPDUs exchanged in the case of a unidirectional link scenario. Port p4 does not receive the information sent by p2. Thus p4 considers it is the designated port, and sends information tagged as *designated* on the segment. Port p2 on bridge A sees a neighbor pretending to be designated, with a worse information (20 is worse than 10). Port p2 can then deduce that there is a communication failure with its peer, and it goes to a discarding state, until it stops receiving the conflicting message from its neighbor.

**Figure 18 Uni-directional Link Blocked by the Dispute Mechanism**

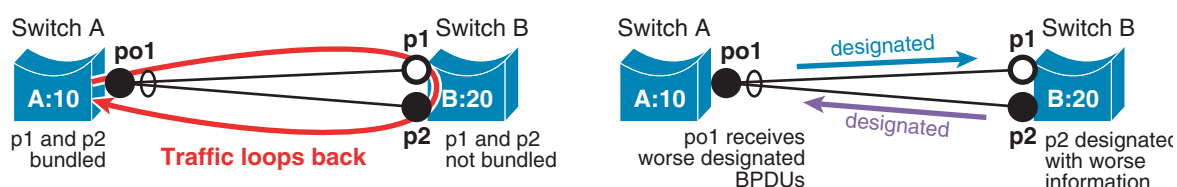


Note that even if this feature is intelligent enough to only act when p4 is (or close to) forwarding, there are some legitimate scenarios where the dispute mechanism can be triggered temporarily. However, a port permanently blocked in a dispute state is almost certainly the indication of a problem.

Loopguard cannot detect a unidirectional link failure unless the peer designated port has been able to send at least one BPDU successfully. The dispute mechanism does not suffer from this limitation and is thus much more efficient at protecting against that kind of problem. On the other hand, the dispute mechanism is not able to detect the brain-dead bridge scenario, as this one does not introduce any inconsistency in the BPDUs. Another slight limitation of the dispute mechanism is that it is limited to Rapid-PVST (RSTP) and MST as it requires the port role to be encoded in the BPDUs.

The example in Figure 19 shows the power of the dispute mechanism in a less trivial case of an EtherChannel inconsistency (EtherChannels are discussed in more details in “Port Channels/EtherChannels” section on page 42). On switch A, the two ports (p1 and p2) are bundled in a channel po1. Because of a configuration error, port p1 and p2 of switch B are not bundled. This could lead to a half loop: a frame transmitted by po1 to port p1 in switch B would be switched back to po1 through p2.

**Figure 19 EtherChannel Error Detected by the Dispute Mechanism**



Suppose that switch A generates better STP information and that po1 is designated, and that it sends periodic information on p1 (BPDUs are sent on a single physical link on a port channel). On switch B, p1 is a root port, but p2 does not receive any BPDU and becomes designated. This port p2 on switch B starts sending worse information to po1, triggering the dispute mechanism on switch A. Note that the result would have been identical if switch B had better information than switch A.

The dispute mechanism is a built-in RSTP and MST. There is no associated configuration command. The feature has been included in NX-OS since the first release for both Nexus 5000 and Nexus 7000. It was introduced on the Catalyst 6500 series in Cisco IOS Release 12.2(18)SXF for MST and in Cisco IOS Release 12.2(33)SXI for Rapid-PVST.

## Bridge Assurance

To ensure that the STP can be run on low-end devices, the IEEE standard specifies that “the memory requirements associated with each bridge port are independent of the number of bridges and LANs in the network”. This rule precludes the creation of adjacencies between bridges, as there can be an arbitrary number of peers connected to a particular port. On a particular segment, only the designated port ends up sending periodic BPDUs. Other peers can (and in fact are likely to) remain silent and no information is collected about them. As a result, there is no consistency check that can be applied to them either. The typical dangerous bridging behavior that makes the phrase “STP identify where not to send frames” (described in “Differences Between Routing and Bridging” section on page 35) thus also indirectly stems from that rule.

Bridge assurance is a very simple feature that introduces a counterpart to portfast configuration. With bridge assurance, a port can be now be classified in two categories:

- Edge ports—ports connected to end stations (existing portfast)
- Network ports—ports connected point to point to another bridge

This is not really an additional configuration burden. In order to reach their full potential, RSTP and MST already required a network with point-to-point links between bridges, and a strict identification of the edge ports.

Bridge assurance forces a bridge to keep sending BPDUs on a network port, even if it is not a designated port. Conversely, a network port expects receiving periodic BPDUs, even if it is designated. If no BPDU is received, the port stays in (or reverts to) a discarding state. Bridge assurance introduces stability in the network by making bridging closer to routing: a bridge is no longer forwarding traffic by default. The network port behavior can also be compared to Loopguard without its plug-and-play aspect. The main weakness of Loopguard was that it was not started until it had received at least a BPDU and detected a peer. Bridge assurance does not suffer from this problem as the existence of the peer is identified by configuration.

Note that this simple consistency check can only be implemented on point-to-point links because a bridge, contrarily to a router, has no way of sending traffic to a subset of its neighbors on a shared segment.

As mentioned in “[Portfast \(Edge Port\)](#)” section on page 37, the CLI has been enhanced to support bridge assurance:

- NX-OS—spanning-tree port type {edge [trunk], network, normal}
- IOS—spanning-tree portfast {edge [trunk], network, normal}

The **normal** setting allows keeping the current existing spanning tree behavior and will be necessary on shared segments or port connected to a bridge that does not support bridge assurance.

The recommended setting for bridge assurance is to consider all ports as network ports by default, using the following global configuration command:

- NX-OS—**spanning-tree port type network default**
- IOS—**spanning-tree portfast network default**

This mode will force the network administrator to visit the configuration of each port and help reveal the most common configuration errors (like non-identified edge ports, or bridge assurance not enabled on a neighbor). But most of all, it is safer to have spanning tree block too many ports than not enough and introduce discarding as the default port state enhances the overall stability of the network.



#### Note

Bridge assurance is the ultimate brain-dead bridge detection mechanism. When paired to dispute mechanism (which is enabled by default), it enforces a very strict consistency check on the connection a bridge can open to a neighbor. Loopguard is a plug-and-play subset of bridge assurance that should not be enabled along with bridge assurance. Loopguard could still make sense if bridge assurance is not supported locally, or on the remote designated bridge.

Bridge assurance was introduced in 4.0(1a)N2(1) on the Nexus 5000, in the initial Release 4.0(1) on the Nexus 7000 and IOS Release 12.2(33)SXI on the Catalyst 6500 Series.

## Rootguard/BPDUGuard

Those features are often incorrectly depicted as loop prevention mechanisms. Neither Rootguard nor BPDUGuard enhance the ability of the STP of detecting redundant connections; they are just administrative tools that change the way STP reacts to certain events. In some cases, it might be more desirable to lose connectivity rather than letting a compromised or mis-configured device in the network.

### BPDUGuard

BPDUGuard is an administrative tool that shut down a port when it receives a BPDU. This feature is commonly enabled on edge ports that are not supposed to have an STP peer. Note that running STP is enough to prevent loops. BPDUGuard simply enforces a security policy that will not only prevent STP interaction, but make the violation of this policy very clear.

## Rootguard

Rootguard places, in a discarding state, a port that receives better information from its neighbor. In particular, it prevents a port from becoming a root port, a port on the path to the root bridge.

The most straightforward use of Rootguard is to enforce some trust policy with a peer. The port configured for Rootguard still runs spanning tree with the peer. In particular, it will accept topology change information and agreements as an answer to a proposal; however, the peer is not trusted enough to be allowed to inject spanning tree information in the network. In general, spanning tree interaction with non-trusted peers is discouraged, and BPDUguard is preferred for this kind of scenario. However, Rootguard can still make sense on a port connected to a trusted device in order to protect against configuration error. For example, configuring Rootguard on the aggregation ports leads to the access will help to isolate an access bridge claiming to be the root.

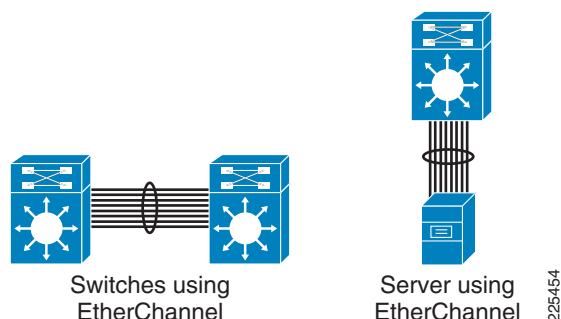
Rootguard can be used in a more subtle way in order to enhance the robustness of the network. As mentioned in “[Rules for STP Network Stability](#)” section on page 37, it is better to restrict as much as possible the freedom of the spanning tree. Suppose that there are 100 possible network topologies that could use port X as a root port and that, out of these topologies, 99 percent are the result of a spanning tree failure or misconfiguration and only 1 percent is the result of a legitimate case. If this only valid scenario is not very practical and/or not very likely, it might be better to sacrifice it for the benefit of the added safety that the rejection of the 99 invalid cases brings. The consequences of enabling Rootguard have network-wide impact and must be evaluated at this level. A practical example of this tradeoff is described in the “[Rootguard and BPDUguard Configuration](#)” section on page 47.

## Other Layer 2 Relevant Features

### Port Channels/EtherChannels

EtherChannel allows multiple physical ports in a single switch to be aggregated forming one logical port. This logical port may be defined as a Layer 2 or Layer 3 interface consisting of a maximum eight ports. The EtherChannel interfaces are called *port channels*, and the technology is also referred to as *port aggregation* or *link aggregation*. The ports comprising the EtherChannel interface are often spread across the line cards of a modular switch to provide a more resilient connection. In general, EtherChannel improves the overall scalability and availability of the network and is a well-documented best practice within the data center. As [Figure 20](#) illustrates, an EtherChannel may be defined between two switches or a server and switching platform.

**Figure 20** Switch and Server EtherChannel Examples



At Layer 2, a port channel is presented to the control plane as a single logical link. This means that the STP does not detect any redundancy and does not need to block. Channels are thus helpful in reducing the dependency on STP redundancy, as recommended in the “[Rules for STP Network Stability](#)” section

on page 37. As long as there is a single member active in a channel, adding or removing a link has no impact on STP either. This reduces the STP computation and the associated MAC address table flushes and flooding that can result from it.



#### Note

IOS computes the STP cost of a channel dynamically based on the bandwidth available. If a member link of a channel goes down, the bandwidth is updated as well as the spanning tree cost for the interface. This could result in the disruptive computation of a new topology. This behavior can however be overridden by explicitly configuring a spanning tree cost on the channel interface. The configuration is not necessary on NX-OS, that does not update the channel cost based on link up/link down.

At last, port channels are able to distribute the traffic and provide granular load-balancing. Switches compute a hash in hardware, based on Layers 2, 3, and 4 fields present in the frames in order to determine which channel member to use when forwarding traffic.

The channeling solution is simple and robust because its scope is limited to the simplest network possible, comprising only two switches connected redundantly. The mechanism operates at the port level, relying on the link status to determine whether a port is operational or not, with the addition of an optional control plane (LACP or PAgP) that also runs below STP.

Link Aggregation Control Protocol (LACP) introduced by IEEE 802.3ad is a standards-based mechanism for two switches to negotiate the building of these bundled links. Many Cisco products also support the use of Port Aggregation Protocol (PAgP) to provide this function. NX-OS does not currently support PAgP. The use of LACP is the recommended solution for configuration of port channel interfaces to the Nexus 7000 over static configuration as it provides configuration sanity check and monitoring of the health of the channel. LACP is configured using the keywords **active** and **passive** in the interface configuration. At least one end of the port channel connection must be placed in **active** mode for channel negotiation to occur. Since LACP is the only supported negotiation protocol in NX-OS, the **channel-protocol** command used in Cisco IOS is not required. Other than this difference, the rest of the configuration is similar to that of Cisco IOS.

The following is an example of a port channel interface configuration in NX-OS and one of its associated physical interfaces:

```
interface port-channel122
  description < to ssl >
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 162-163,180-183
  logging event port link-status
  logging event port trunk-status

interface Ethernet1/19
  description < to ssl >
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 162-163,180-183
  no shutdown
  channel-group 122 mode active
```

Previous best practices guidance for the Cisco Catalyst 6500 configuration has also included the configuration of the adaptive port channel hash-distribution algorithm. This configuration optimized the behavior of the port ASICs of member ports upon the failure of a single member. The Nexus 7000 performs this optimization by default, and does not require or support this command. NX-OS supports the customization of the load-balancing criteria on port channels through the **port-channel load-balance ethernet** command, either for the entire device or on a per-module basis.

**Note**

Hashing algorithms are configured on a per-hop basis, and do not need to match on both sides of a port channel. For more detail on configuration of port channel interfaces in NX-OS, refer to the *Cisco NX-OS Interfaces Configuration Guide, Release 4.1* at the following URL:

[http://www.cisco.com/en/US/docs/switches/datacenter/sw/4\\_1/nx-os/interfaces/configuration/guide/if\\_cli.pdf](http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_1/nx-os/interfaces/configuration/guide/if_cli.pdf)

The Port Aggregation Protocol (PAgP) is the Cisco proprietary predecessor of LACP. It has been recently extended to be used along with the Virtual Switching System ( see “EPAgP” section on page 58) and is still a viable option.

**UDLD (Uni-Directional Link Detection)**

The UDLD protocol allows devices connected through fiber-optic or copper (for example, Category 5 cabling) Ethernet cables connected to LAN ports to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including bridging loops. UDLD should be enabled globally on all switches in the data center topology. Global UDLD only enables the protocol on fiber optic interfaces, because it is common for inter-switch links and cabling problems, cross-wiring and single fiber failures. UDLD is a peer-to-peer protocol that must be run at both ends of a link to be operational. As servers do not support UDLD; there is no value in running it on server ports.

Several spanning tree features can detect unidirectional link failures and react to them. UDLD is different in the way that it operates on physical ports, at a lower level:

- A port channel is a single logical port from the perspective of STP, and STP thus cannot isolate a single faulty physical link out of the bundle. Only a port specific protocol like UDLD and LACP) can achieve that.
- In case of unidirectional link failure, the mechanisms introduced by STP might prevent a loop from happening at the expense of connectivity, by keeping a port logically blocked. By bringing down the offending link, UDLD allows spanning tree to use a potential alternate port and re-establish connectivity.

UDLD performs a sort of three way handshake that allows a port to check that its neighbors can see it. However, if UDLD stops receiving protocol messages from its neighbors, it takes not action. The aggressive option of UDLD allows reacting to this particular failure and turns UDLD into a keepalive mechanism. The aggressive mode can detect a loss of connectivity that does not bring physically the link down. It can also generate false positives and link flapping triggered by the peer switch not responding quick enough which can be caused by momentary high CPU utilization. Aggressive UDLD does not protect against a failure scenario that would have the network-wide impact of a loop and is thus not considered necessary in the data center environment.

UDLD configuration is slightly different on Cisco NX-OS from Cisco IOS. The UDLD commands are not exposed by default; UDLD must be enabled as a feature using the feature **udld** command. Once the feature is enabled, it is turned on globally; therefore, the **udld enable** command is not required. NX-OS provides a **udld reset** command option to reset all ports shut down by UDLD. This command is not available in Cisco IOS.

## Network Design with STP

The network used in the validation effort included the following major data center switching platforms offered by Cisco:

- Nexus 7000 in the core and aggregation
- Nexus 2000, 5000, 7000, Catalyst 6500, and Catalyst 4948 in the access

To the exception of the latest bridge assurance and dispute mechanism, which are only available in the Nexus and Catalyst 6500 platforms, all the other features used are equally supported across the different range of switches. No significant difference was noted in the convergence time; thus, this section does not focus on the difference in the hardware devices validated.

**Figure 21** STP Configuration Overview

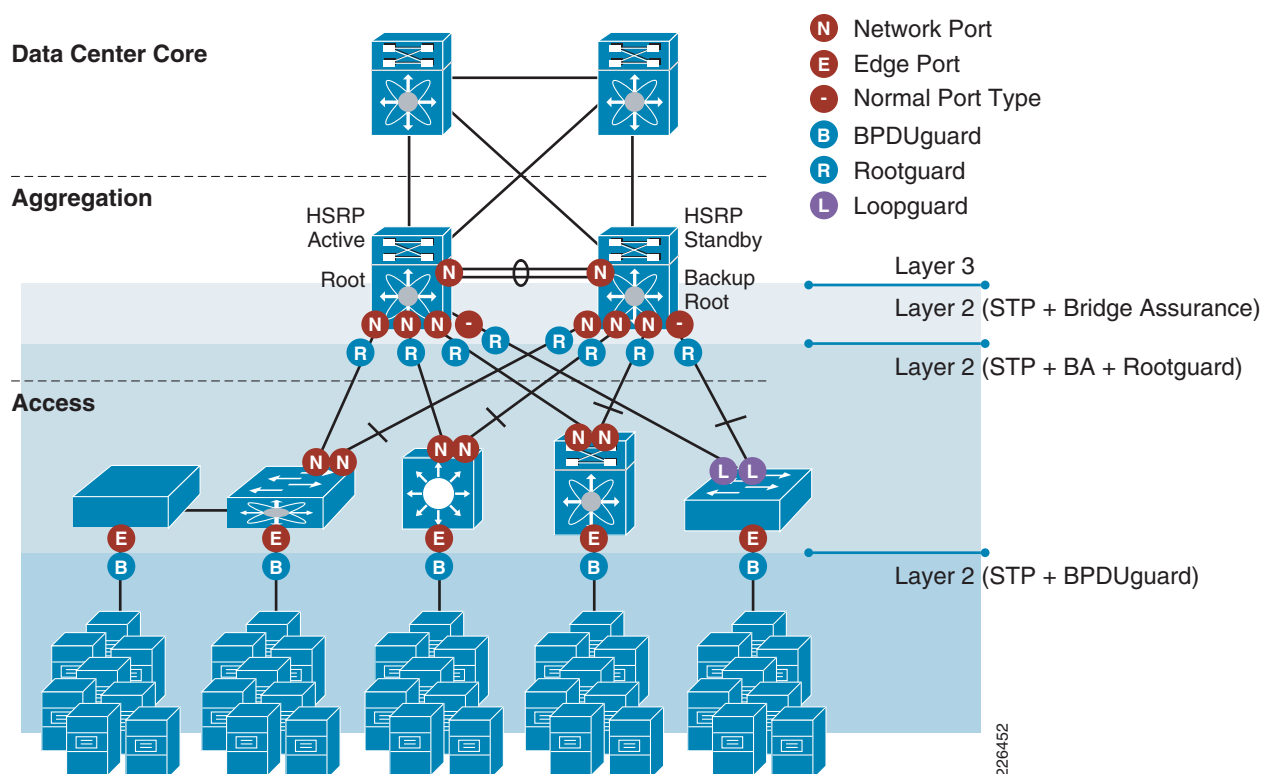


Figure 21 represents the typical data center design for Layer 2 using STP. The location of the different root bridge (and its backup) as well as the HSRP active (and its standby counterpart) are represented for a particular VLAN. Load balancing is achieved by simply switching the role of the aggregation devices on a per VLAN basis. Figure 21 also shows where the different spanning tree features are configured in the network. On the top of the boundary between Layer 3 and Layer 2, on the aggregation devices, two additional boundaries are represented for STP.

- The downstream interfaces leading to the access switches are configured for Rootguard on the aggregation device. The access relies on the STP information from the aggregation, but not the other way around. The justification for this demarcation is detailed in Rootguard on downlink of aggregation and BPDUguard on access ports (possibly port security). For more information, refer to “Rootguard and BPDUguard Configuration” section on page 47.
- The server ports are configured for BPDUguard and do not accept any STP information.

The access switch on the right-end side does not support bridge assurance. Instead, Loopguard is configured on its uplinks and bridge assurance is disabled on the corresponding ports on the aggregation switches.

## Bridge-Specific Recommendations

This section lists some recommendations that are to be applied to a subset of the switches.

### Determine Explicitly the Location of the Root Bridge

STP will compute a topology centered on the root bridge, meaning that the ports that are kept forwarding in the final topology are the ones with the best path (evaluated with a metric depending on the bandwidth) to the root bridge. In the data center, Layer 2 is used for its flexibility and mainly bridges frames for Layer 3 traffic between some hosts and their default gateway. Locating the root bridge as close as possible to this default gateway is enough to get STP to compute the optimal topology for the network in a single command.

The root bridge is the bridge with the smallest bridge ID in the network. The most significant 16 bits of the bridge ID can be set using the **priority** command. The last 48 bits of the bridge ID are MAC address from the switch chassis, so that a root bridge can always be elected in a deterministic way, even in the case when the best priority in the network is configured on several bridges.

By locating the bridge with the second best priority in the network very close to the root bridge, the failure of the root bridge will have minimal impact on the overall network topology.

The recommendation is to make minimal priority configuration, for the sake of simplicity. It is enough to identify the root bridge with the best priority, and the bridge that will serve as a backup root bridge with the second best priority. All other bridges will keep the default priority. The default priority value is 32768. The global configuration mode commands required are listed below.

For the root bridge, the command is as follows:

```
spanning-tree vlan 1 priority 24576
```

for the bridge backing up the root bridge, the command is as follows:

```
spanning-tree vlan 1 priority 28672
```

The priority is usually settable by increment of 4096, so the values suggested above are the minimal changes to this default priority and can still be overridden if necessary.



#### Note

The **spanning-tree vlan 1 root primary** command is often suggested to set the root bridge. The root command is in fact a macro that is expanded into several individual commands adjusting not only the bridge priority but also the *forward-delay* and *max-age* timer values. The convergence time does not depend on timer tuning in Rapid-PVST or MST mode, and the only relevant configuration that the root macro provides, in order to influence the root election, is setting the bridge priority to the above recommended values.



#### Note

There is a single root bridge at a time for STP, and the protocol does not even identify the bridge with the second best bridge ID. This is why the term root and backup root was preferred to primary root and secondary root in this document.

### Rootguard and BPDUguard Configuration

The [Rules for STP Network Stability, page 37](#), suggest that STP should be granted a minimum level of freedom to put a port into forwarding. The configuration of Rootguard and BPDUguard in the data center, justified below, follows this sound principle.

As mentioned in [“Rootguard” section on page 42](#), configuring this feature is generally a trade-off between connectivity versus stability.

- Connectivity drawback:

The recommended design features Rootguard on the aggregation downlink ports toward the access switches. The immediate consequence of this configuration is that the aggregation switches will not be able to elect any of those ports as a root port. The non-root aggregation switch needs to elect a root port. In stable conditions, this is port that is directly connected to the root bridge. Should this inter switch link fail, the non-root aggregation switch would have to connect to the root bridge through an access switch. The Rootguard configuration will prevent that, and this is the connectivity drawback. This drawback is acceptable because:

- Having an access switch connecting the aggregation devices is not a realistic scenario. If one access switch could sustain the traffic that is potentially exchanged between the aggregation boxes, it would be considered itself an aggregation switch (then, indeed, Rootguard should not be configured on the link leading to it.)
- When a VLAN does not have any port forwarding traffic on a switch, a feature called autostate brings down the corresponding VLAN interface. This will happen in the backup root aggregation switch if all the access switches are connected redundantly to the aggregation devices. If the VLAN interface goes down, the routing protocol will stop advertising the corresponding IP subnet. This will prevent traffic from the core targeted at the access to be sent to the non-root aggregation switch. So as soon as autostate is triggered, the root bridge will handle the traffic between the access and the core in both directions.

- Stability benefit:

In the network depicted in [Figure 21](#), only the access switches are blocking some ports. It means that the only risk of bridging loop is coming from an access bridge opening both its uplinks. The following explains how Rootguard, combined with the dispute mechanism and bridge assurance, introduces the most stringent consistency check on the ports leading to the access.

A forwarding port can only have two STP roles: root or designated. A loop could only occur if an access switch places:

- Case 1: both uplinks in designated role.
- Case 2: one uplink in designated and one uplink in root role.
- Case 3: both uplinks in root role.

If an access switch does not advertise the role of its port by sending BPDUs, bridge assurance on the aggregation switch blocks the port. If an access switch advertise a role of designated, the aggregation port will block (Rootguard will block if the information is superior, the dispute mechanism will block if the information is inferior).

Cases 1 and 2 will result in the aggregation bridge blocking the loop, even if the access switch wanted to open one.

Case 3 could still result in a loop. This would be a severe problem, as STP can only elect a single root port on a given bridge. There is, however, a high-level of confidence that the software should be able to avoid this, as it is purely a local decision.

- Recommendations:
  - The link between the aggregation devices must be resilient. A port channel spread across different line cards is highly desirable.
  - For a particular VLAN, no edge device must be connected to the aggregation bridges, and all the access switches be connected to both aggregation devices. This way, in case of the failure of the inter-aggregation link, the autostate feature will bring down the VLAN interfaces on the backup root switch and prevent the black holing of "southbound" traffic.




---

**Note** Attention must be paid to summarization in order to benefit from the autostate feature. If the VLAN interface that is going down had its associated subnet advertised as part of a route summary by the aggregation device, it might still attract and black-hole traffic for this VLAN.

---

BPDUguard also enforces a very restrictive policy to the access ports. Those ports are connected to servers that are generally not controlled by the networking team. It is not expected that any form of STP is run on or through a server. As a result, bringing down a port that would violate this policy is an appropriate action. It will immediately alert the network administrator, while protecting the network. Eventually, there is still the possibility for a server to bridge traffic between its NICs, while filtering BPDUs. This behavior, which could be assimilated as an attack, cannot be detected by STP. Some other features, like port security, could help detect the resulting loop. This document does not however recommend the use of port security on the access switch as it could impede the ability of the network to support some promising features like virtual machine mobility.

### HSRP Configuration

The goal is to align the location of the spanning tree root and the HSRP active in order to ensure that the topology provides the best connectivity to the default gateway of the servers in the VLAN, as mentioned in the [“Determine Explicitly the Location of the Root Bridge” section on page 46](#).

On the aggregation switch, this is achieved by configuring the primary with the highest priority. Example for NX-OS provided below:

#### NX-OS Active HSRP Device

```
interface Vlan161
  hsrp 1
    authentication text c1sc0
    priority 20
    timers 1 3
  ip 10.8.162.1
```

#### NX-OS Standby HSRP Device

```
interface Vlan161
  hsrp 1
    authentication text c1sc0
    priority 10
    timers 1 3
  ip 10.8.162.1
```

## Network-Wide Recommendations

### STP Mode Selection

The spanning tree protocol was first specified in 802.1D. When Cisco started to support VLANs, each VLAN was assigned its own instance of STP, in a model called Per VLAN Spanning Tree (PVST). The concept of VLAN was then standardized in 802.1Q, but the IEEE did not adapt 802.1D, and the topology used by all the newly introduced 802.1Q VLANs was still governed by a single VLAN-unaware instance of STP. Because the IEEE model did not provide for per-VLAN load-balancing, Cisco kept its PVST model and introduced PVST+. PVST+ is just a slight update providing interaction with third party 802.1Q bridges running a single instance of STP (note that PVST+ and PVST are used interchangeably, as all Cisco switches now run PVST+).

In order to provide fast convergence, independent of any timer, 802.1w amended 802.1D and replaced STP with the Rapid Spanning Tree Protocol (RSTP). RSTP was still computing a single topology for all the VLANs so the project 802.1s, an amendment to 802.1Q, was started to create a VLAN-aware spanning tree protocol, able to compute several instances. The resulting new protocol, Multiple Spanning Trees (MST), is meant to be a superset RSTP. In fact, it is expected that 802.1D will eventually be merged into 802.1Q. Cisco platforms support up to three different STP modes:

- Per VLAN Spanning Tree PVST(PVST+)—This mode runs a classic spanning tree instance per VLAN. This mode is not supported in NX-OS as it is superseded by Rapid-PVST.
- Rapid-PVST—This mode runs an instance of Rapid Spanning Tree Protocol (RSTP) per VLAN.
- Multiple Spanning Trees (MST).

STP and RSTP, as mentioned above, are only able to compute a single topology. In PVST modes, in order to achieve VLAN load balancing, the whole spanning tree protocol is thus run per-VLAN. Each VLAN protocol instance operates independently and sends its own BPDUs. On the other hand, a single MST process takes care of all the VLANs. MST computes different topologies by exchanging a single BPDU with its peers. Each VLAN is then mapped to one of those resulting topologies.

The choice between Rapid-PVST and MST, two fast protocols that provide VLAN load-balancing, thus requires more attention.

The advantages of MST over Rapid-PVST are as follows:

- MST is an IEEE standard.
- MST is more resource efficient. In particular, the number of BPDUs transmitted by MST does not depend on number of VLANs, as in Rapid-PVST.

The advantages of Rapid-PVST over MST are as follows:

- MST introduces some administrative overhead: a consistent configuration has to be strictly enforced across the domain in order to benefit from the load balancing.
- MST does not interact seamlessly with service appliances/service modules in transparent mode.

This latter issue is the reason why MST might not be appropriate in some data center environments for now. A service module in transparent mode connects one VLAN to another. In PVST mode, it also connects two spanning tree instances, which share a same root bridge for example. MST does not run per-VLAN but per-physical link. On a given physical port, MST only sends a single untagged BPDU, regardless of the VLANs. There is nothing much the service appliance can do with this BPDU. If it sent it back, MST would detect a self-looped port and block all the VLANs. As a result, most service modules or service appliances simply cannot propagate MST BPDUs. Running STP through the service modules/appliances in transparent mode is an important stability feature to protect against some dual-active failure scenarios. Until those service devices have some form of MST support, it is recommended to stick to Rapid-PVST when they are deployed in the network.

Rapid-PVST is the default spanning-tree mode in NX-OS. The global configuration command in both NX-OS and IOS is as follows:

```
spanning-tree mode rapid-pvst
```



#### Note

The interaction between MST and Rapid-PVST is sub-optimal in term of convergence time and resource utilization. Whether MST or Rapid-PVST, it is recommended to use a consistent spanning tree mode across the whole bridged domain.

### Pathcost Method Long

The STP BPDUs carry a 32-bit field representing the cost to reach the root bridge. The initial implementation of STP, however, was only using 16 bits out of those 32 available, allowing a cost in the range of 0 to 65535.

The spanning tree privilege links with the least cost to the root bridge, it is incremented by the cost of the links traversed on the path to the root. By default, the cost is tied to the bandwidth of a link. Encoded over 16-bit, the cost was not granular enough for modern networks: a 100 MB link had a cost of 19, a Gigabit links 4 and 10-Gigabit links 2. The IEEE decided to update the protocol so that it takes full advantage of the 32 bits reserved for the cost in the BPDU. This change is enabled by the following command:

```
spanning-tree pathcost method long
```

It is recommended to configure this cost method consistently across the whole bridged domain. “[Determine Explicitly the Location of the Root Bridge](#)” section on page 46 details how assigning the root bridge to a sensible location in the network was generally the only spanning tree configuration required in order to get the desired forwarding topology. A mismatch would not cause a failure of STP, but would certainly invalidate this property, and some additional STP cost configuration would be required.

### Network Type by Default for Bridge Assurance

Bridge assurance is a recent feature that is supported on the most common data center switches (Nexus 2000, Nexus 5000, Nexus 7000) and has been introduced on the and Catalyst 6500. On these switches, the following global configuration command enabling the feature is the default:

```
spanning-tree bridge assurance
```

Bridge assurance introduce classifies a given port as **network**, **edge**, or **normal**. The default port type is **normal**, which means that bridge assurance will not apply. As mentioned in “[Bridge Assurance](#)” section on page 40, the recommendation is to get away from the risks associated to a plug-and-play STP and to configure the default port mode to **network**. To change this default to **network**, enter the following command:

```
NX-OS:spanning-tree port type network default
IOS: spanning-tree portfast network default
```

With this best practice, the network benefits from the very conservative behavior introduced by bridge assurance. Edge ports need to be correctly configured for optimal RSTP/MST operation. Bridge assurance will make the network ports safer and will force the identification of edge ports.



#### Note

Bridge assurance will most likely block ports that are not correctly configured. When moving the default port type to **network**, be sure that edge ports are correctly identified and that the network ports are connected point-to-point to a network port peer that is running bridge assurance.

### Respect the Scalability Limits

All the switching platforms provide guidelines in term of maximum number of supported instance ports. They should be enforced in order to keep STP stable (the precise figures are available in the release notes for the particular product in use).

The count, in term of virtual port, is used to estimate the CPU utilization related to STP. Note that the computation of the topology has almost no impact on the process. The main constraint is related to the transmission/reception of BPDUs, and this is particularly an issue for PVST modes. Some other considerations, like memory utilization are taken into account when generating the advertised limit on certain platforms.

The Catalyst 6000 Series of switches are also enforcing an additional limit depending on the line card.



#### Note

The logical port count is the number of logical ports that STP is handling on the platform. It is computed from the number of ports times the number of VLANs with STP enabled on each of those ports. A port channel interface with one VLAN only represents a single port to STP. On the other hand, the per-port limit introduced for the Catalyst 6000 is used to evaluate the stress on the line card. This limit is calculated using the physical ports used by STP, so that a channel would be broken down in the individual physical ports it comprises. A port channel with four members carrying a single VLAN would represent four logical port for this calculation.

#### LACP Active

The use of LACP is beneficial to the stability of the Layer 2 network. Even if a port channel is hiding a loop from to STP, there is still a loop in the physical connection between the switches (the [“Dispute Mechanism” section on page 39](#), provides an example of channeling misconfiguration that can lead to a loop). LACP enforces some consistency checks (locally and between peers) and has the final word as to which physical link will join the channel. It also monitors the health of the individual members. Both ends of a port channel can be configured as LACP active for the sake of simplicity.

#### UDLD

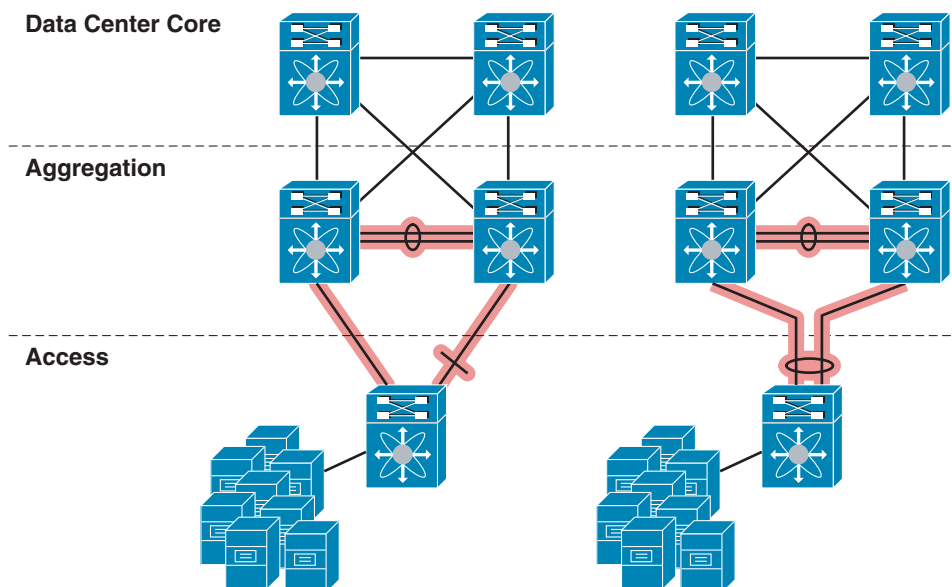
The use of normal (non-aggressive) UDLD is recommended network wide, as explained in the [“UDLD \(Uni-Directional Link Detection\)” section on page 44](#).

## "Loop Free" Layer 2 Design

A data center is generally made of similar simple building blocks, replicated at will to achieve the desired level of scalability. The solution provides for redundancy, which means that devices in a particular position are at least duplicated. The traditional network design is a typical example of that: a pair of aggregation switches to which as many access switches as necessary are connected in a redundant way. The two main drawbacks of this solution are as follows:

- There is no Layer-2 multipathing for a particular VLAN, and the per-VLAN load balancing that allows using both uplinks of an access switch needs user configuration. There is no way of escaping this constraint as it dictated by the way bridging requires a spanning tree in the data plane.
- The dependency on STP for the creation of a spanning tree topology in the data plane, introducing delay in the convergence and potential risks.

Port channel technology is solving those remaining issues for the very specific case of the interconnection of two switches (see [“Port Channels/EtherChannels” section on page 42](#).) Alone, link aggregation cannot be used to create a fully redundant data center, as it does not protect against the failure of a single switch. Cisco has recently introduced two technologies that lift this latter limitation. Both VSS (on the Catalyst 6000) and vPC (on the Nexus 7000) allow creating a Layer-2 port channel interface distributed across two different physical switches. This limited step-up in the channeling capability is enough to provide the simple building block required to build an entire data center with no dependency on the spanning tree model. [Figure 22](#) shows a high level use of the solution.

**Figure 22 Loop-Free Network**

226451

The left part of [Figure 22](#) illustrates the current model, where the redundancy is handled by STP. The right part of [Figure 22](#) represents the solution introduced by distributing the end of a channel across the two aggregation switches.

The logical view shows that the redundancy has been hidden from STP. As far as the “[Rules for STP Network Stability](#)” section on [page 37](#), are concerned, the right side of [Figure 22](#) shows the best solution, where the following are depicted:

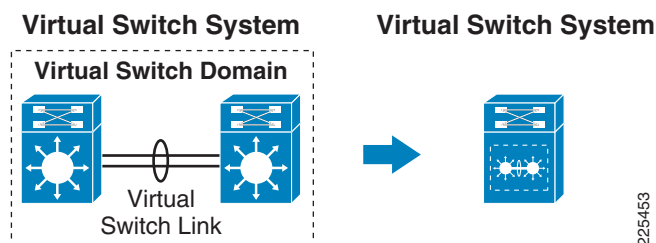
- The number of blocked ports has been eliminated
- The freedom of STP has been also entirely removed, as it cannot open a loop even if it wanted to.

However, the recommendation is to keep STP on as a backup mechanism. Even if the redundancy has been hidden to STP, it is still there, at a lower layer. It is just handled by a different mechanism. STP will help protect against a configuration error that breaks a channel into individual links, for example.

## Virtual Switching System (VSS)

### Concept

The VSS technology allows the grouping of two Catalyst 6500 switches into a single virtual switch. The left side of [Figure 23](#) represents the physical layout of the VSS: two Catalyst 6500s 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 [Figure 23](#) shows this construct forms a single logical switch with a single control plane, a VSS.

**Figure 23 Virtual Switching System**

The primary benefits of this logical grouping include the following:

- Increased operational efficiency by simplifying the network through virtualization
- Increased availability and forwarding performance through Inter-chassis Stateful Switchover (SSO) and Nonstop Forwarding (NSF)
- Increased availability and forwarding performance through Multichassis EtherChannel (MEC)

## Components

This subsection describes the fundamental building blocks of the VSS functionality including:

- [Virtual Switch Domain \(VSD\)](#)
- [Virtual Switch Link \(VSL\)](#)
- [Multichassis EtherChannel \(MEC\)](#)

### Virtual Switch Domain (VSD)

A VSD consists of two Catalyst 6500 chassis as members and is identified by a unique system ID. Although the number of member switches is limited to two per domain, the number of domains is not; there are 255 unique domain IDs available.

The VSS employs an active/standby control topology where the active VSS domain switch performs all control plane functions including the following:

- Layer 2 (EtherChannel, PAgP, LACP, STP)
- Layer 3 (EIGRP, OSPF, VRF, etc.)
- First Hop Redundancy Protocols (HSRP, VRRP, etc.)
- Management Protocols (SNMP, Telnet, SSH, etc.)

The active virtual switch is chosen during the initialization of the domain using the Role Resolution Protocol (RRP) across the newly active VSL. In addition, the initialization of the domain requires that all hardware and software requirements are met and configurations are synchronized between the virtual switch domain members. These are all functions of the Virtual Switch Link Protocol (VSLP) that executes between the two members.

When in a normal operating state, the VSS data plane is active/active; both switches in the domain are forwarding traffic.

### Virtual Switch Link (VSL)

As shown in [Figure 23](#), the VSL is an inter-switch link that forms the backbone of the VSS. The VSL supports control traffic between domain switches allowing the VSS to form and operate. In addition, normal data traffic may also use the VSL connection as a valid forwarding path. The VSL benefits from the high availability and scalability features of the Cisco EtherChannel.

The communication between VSS members across the VSL uses the VSLP. The VSLP includes the following protocols:

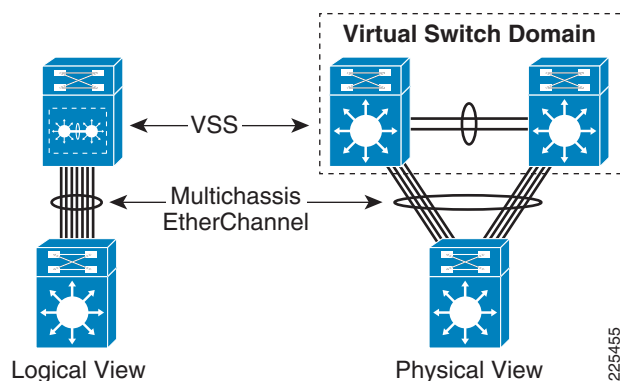
- Link Management Protocol (LMP)
- Role Resolution Protocol (RRP)

LMP manages the VSL, providing for the exchange of domain member identities and switch parameters necessary to form the VSS. RRP validates the capabilities of the domain members and coordinates the active switch election process. In addition to these functions, VSLP monitors the state of the VSL connection through probe messages.

### Multichassis EtherChannel (MEC)

Figure 20 exemplified the one-to-one relationship the traditional EtherChannel entails. VSS allows an EtherChannel to exist across two physical chassis that are logically one. As shown in Figure 24, the VSS from a control plane perspective, is a single switch and thus the traditional switch may use EtherChannel. The right side of Figure 24 shows that the EtherChannel is supported across two VSS-enabled chassis. This MEC is forwarding on all ports, from a logical Layer-2 perspective there is no loop. Convergence is no longer dependent on the implementation of spanning tree, but on the resilience of EtherChannel itself. The forwarding fabric is expanded and the Layer 2 topology simplified.

**Figure 24** VSS Multichassis EtherChannel Logical and Physical Switching Topology



## Configuration

This section provides a high-level introduction to the configuration of VSS. The configuration guidance for implementing VSS between the aggregation and access layer of the data center is detailed in the “[Network Design with VSS](#)” section on page 61.

### Configuring Virtual Switching System (VSS)

A VSS environment consists of two member switches. The switches obtain virtual switch IDs during the VSS conversion process. These VSS-enabled switches may then join a larger construct called a switching domain. The switching domain forms the boundaries of the virtual switching functionality of VSS. The following is the virtual switch domain configuration used during testing:

```
switch virtual domain 100
switch mode virtual
switch 1 priority 110
switch 2 priority 100
mac-address use-virtual
!
```

Note that two VSS-enabled switches are referenced by an ID (1 and 2) and assigned a priority (100 and 110). The default switch priority value is 100. The switch with the higher priority value assumes the active role during role-resolution events such as during system startup or if preemption is enabled. It is not recommended to use preemption in the data center; typically, the VSS member switches use identical line card configurations and reside in a dual-homed environment, where favoring one system over another is irrelevant.

The **mac-address use-virtual** command permits a virtual MAC address to be used by the system. This command accesses a pool of reserved addresses that incorporate the virtual switch domain. Without this command, the VSS will use the MAC address of the active VSS member at boot time, as determined by the active supervisors EEPROM. The standby VSS member will inherit this address upon failure or reboot of the one active switch.

### Configuring the Virtual Switch Link (VSL)

The two members of the VSS domain communicate using the VSL. In this example, the VSL is formed using an EtherChannel. Because the two physical switches are merged into a single virtual switch, both ends of this channel will appear in the virtual switch configuration. In this example, the port channels 98 and 99 are the ends of VSL, and they are assigned through the **switch virtual link** command. The QoS policies are assigned automatically to the VSL that prioritizes VSS traffic.

```
interface Port-channel98
  description <<*** Etherchannel to Agg2 ***>>
  no switchport
  no ip address
  switch virtual link 2
  mls qos trust cos
  no mls qos channel-consistency
!
interface Port-channel99
  description <<*** Etherchannel to Agg1 ***>>
  no switchport
  no ip address
  switch virtual link 1
  mls qos trust cos
  no mls qos channel-consistency
!
```

The physical interfaces used by each port channel reside on one of the two VSS member chassis'. The **show etherchannel summary** command confirms that VSS member switch 2 possesses the physical ports comprising port channel 98 and that switch 1 provides the interface resources for port channel 99.

```
dca-vss#show etherchannel summary
Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
98     Po98(RU)      -         Te2/7/4(P)   Te2/13/4(P)
99     Po99(RU)      -         Te1/7/4(P)   Te1/13/4(P)
```



#### Note

The VSS interface references include the VSS switch ID as the first identifier for all interfaces. In the above example, **Te2/7/4** describes the 10-Gigabit Ethernet interface on VSS member switch 2, slot 7, port 4.

The interfaces are configured as statically configured EtherChannels using the default load balancing hash algorithm result of the source and destination IP address. The following is a sample configuration:

```
interface TenGigabitEthernet1/7/4
  description <<*** to DCA-Agg1 ***>>
  no switchport
  no ip address
  mls qos trust cos
```

```
channel-group 99 mode on
!
```

### Configuring Multichassis EtherChannels

MEC simplifies the traffic patterns in the data center as [Figure 24](#) illustrates from a Layer 2 perspective, despite physically redundant paths there are no Layer 2 forwarding loops. Cisco MEC is designed to limit VSL utilization by preferring local MCE interfaces.



#### Note

A maximum of 128-port channels are supported in IOS Release 12.2(33)SXH. This number has been increased to 512 in IOS Release 12.2(33)SXI.

VSS MEC functionality supports both dynamic and static port aggregation techniques. Dynamic formation of EtherChannels through PAgP or LACP is achieved by using a common device identifier between the VSS peers and the remote switch. It is important to note that the remote non-VSS switch is unaware that two distinct switching platforms are truly being used in the EtherChannel.

Cisco devices allocate traffic across members of an EtherChannel bundle using a hash distribution mechanism. The following sample configuration highlights the use of a Layer 4 hashing algorithm to maximize load distribution across a MEC formed through PAgP and another through LACP. In each case, the port channel is similarly defined, supporting the necessary VLANs each access layer switch requires.

```
port-channel hash-distribution adaptive
port-channel load-balance src-dst-mixed-ip-port
interface TenGigabitEthernet1/13/1
 channel-protocol pagp
 channel-group 51 mode desirable
!
interface TenGigabitEthernet1/13/2
 channel-protocol lacp
 channel-group 52 mode active
!
interface Port-channel51
 description <<*& To Access Switch 1 *&>>
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 128-133,164-167,180-183,300-399
 switchport mode trunk
interface Port-channel52
 description <<*& To Access Switch 2 *&>>
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 128-133,164-167,180-183,300-399
 switchport mode trunk
```

## Packet Flow/Convergence

### Traffic Flow

The VSL can be seen as an extension to the backplane between the two members of the virtual switch, the medium allowing a single switch supervisor to control two physically independent switching platforms. However, the link is not limited to only VSS messaging. The VSL may also provide a transport for data traffic in the following conditions:

- Layer 2 traffic flooded over a VLAN
- Packets requiring software processing by the active supervisor engine where the ingress interface is on the chassis with the standby supervisor

- The packet destination is on the peer chassis, such as the following examples:
  - Traffic within a VLAN where the only known destination interface is on the peer chassis
  - Traffic that is replicated for a multicast group and some multicast receivers are only attached to the peer chassis
  - The known unicast destination MAC address is only on the peer chassis
  - The packet is a MAC-notification frame destined for a port on the peer chassis

The network administrator must provide enough capacity to operate under normal and failure conditions. The VSS switch members will always prefer a local forwarding path and it is highly recommended to dual-home all entities to the VSS through MEC to reduce the presence of data traffic on the VSL. To verify the VSL use the **show switch virtual link** command. Note that in the following example, interface **Te2/7/4** supports the control traffic between VSS peers.

```
#show switch virtual link
VSL Status : UP
VSL Uptime : 1 day, 44 minutes
VSL SCP Ping : Pass
VSL ICC Ping : Pass
VSL Control Link : Te2/7/4
```

To verify the VSS configuration, use the **show switch virtual role** command as follows:

```
#show switch virtual role
Switch  Switch Status  Preempt    Priority  Role      Session ID
      Number          Oper (Conf) Oper (Conf) Oper (Conf) Local  Remote
-----
LOCAL   2      UP         FALSE(N ) 2 (2 )    ACTIVE   0      0
REMOTE  1      UP         FALSE(N ) 1 (1 )    STANDBY 2221   6924
```

Note that the status for the **Local** and **Remote** switch is **UP** and that preempt is not enabled.

### VSS High Availability

VSS improves the accessibility of data center resources by being a highly available entity itself. SSO is a requirement for each peer in the VSS configuration. SSO allows the standby switch to readily assume responsibility for the switching fabric. This feature is enabled through the following configuration statements:

```
redundancy
mode sso
auto-sync running-config
!
```

The following primary failure events can occur in VSS deployment:

- Failure of the active VSS switch
- Failure of the standby VSS switch
- Failure of the VSL

The failure of the active VSS peer switch is managed through SSO. The VSL allows state information to be passed from the active to the hot-standby peer supervisor. Upon detection of the active peer failing, the standby VSS assumes the active switching role. There is no spanning tree convergence or routing topology change. During testing, the newly active VSS peer assumes control of the aggregation layer switching fabric in less than one second.

The second failure scenario, where the standby VSS switch fails is essentially a non-event. There is no convergence as the active VSS peer continues to operate normally. The data center aggregation layer loses capacity, but the forwarding fabric for all dual-homed devices is available.




---

**Note** Devices single-homed to the aggregation layer are at risk of isolation; it is recommended to dual-home to any VSS device whenever possible.

---

The final failure scenario involves the loss of the VSL between the VSS peers. The VSL connection is vital to the operation of the virtual system. Failure of this link without proper detection mechanisms would result in a dual-active scenario, where both VSS member switches assume an active posture creating an unstable Layers 2 and 3 environment.

The following are dual-active mitigation techniques:

- Enhanced PAgP (EPAgP)
- Bidirectional Forwarding Detection (BFD)
- Fast Hello Packets

EPAgP

EPAgP uses MEC as a medium to detect an active-active VSS environment. EPAgP introduces a new type length value (TLV) to the PAgP messages shared between the switches forming the MEC. The VSS enabled switches place the ID of the active switch in this field. If the TLV value received differs from that of the active switch, the active switch will enter recovery mode. By default, recovery mode on a VSS switch means all interfaces, except for the VSL links, will be disabled. EPAgP is the recommended method to address potential dual-active conditions.




---

**Note** EPAgP support is available on the Cisco Catalyst 6500 platforms using Cisco IOS Release 12.2(33)SXH. It is important to verify EPAgP support on the non-VSS switch being used for dual-active detection.

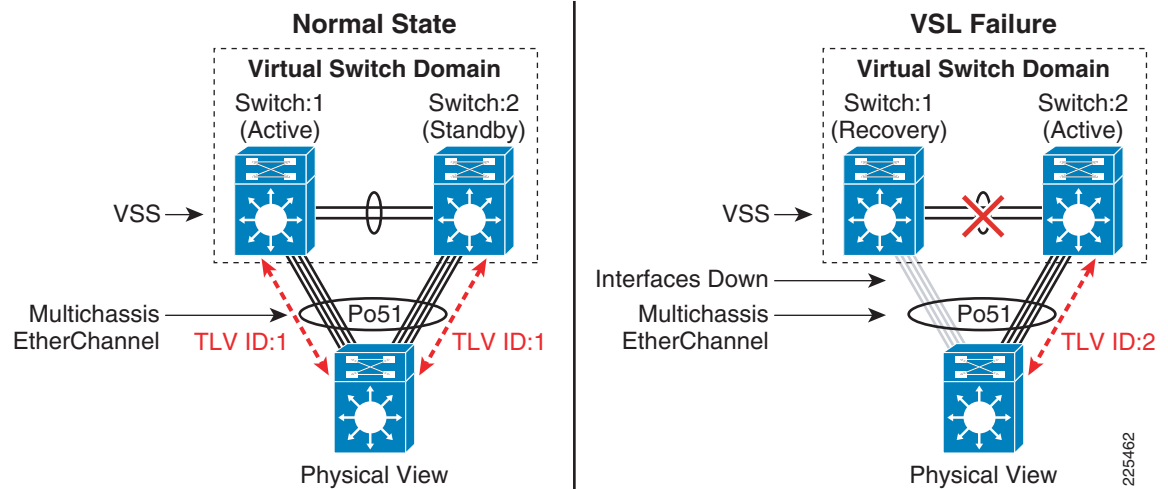
---

[Figure 25](#) illustrates an EPAgP implementation where a third Cisco switch supporting EPAgP provides a means for dual-active detection. In this example a MEC, port channel 51, exists between the VSS aggregation switches and an access layer switch. The TLV associated with VSS member switch **1** will be the active ID passed under normal operating conditions. The following sample configuration was used during testing indicating that the trusted MEC is port channel 51:

```
switch virtual domain 100
  dual-active detection pagp trust channel-group 51
```

Upon failure of the VSL, VSS switch 2 assumes the active role and immediately advertises its TLV value as it considers itself the active VSS switch. When the new TLV value reaches the VSS switch 1, switch 1 identifies that a dual-active condition exists and immediately brings down all of its interfaces and enters a recovery mode. This aggressive action by the previously active switch 1 provides network stability by simply removing itself from the active/active network topology. Testing indicated convergence times in the range of 400 to 500 milliseconds.

**Figure 25 Dual-Active Mitigation: EPAgP Example**



The **show switch virtual dual-active** command provides further insight into the dual-active detection configuration. Note that in the example below, BFD is not enabled and no interfaces are excluded from the shutdown procedure performed by the VSS switch when a dual-active condition is detected. Typically, management interfaces may be excluded from this process to allow access to the VSS platform. The received and expected fields indicate the different TLV values that initiated the dual-active recovery mode.

```
#show switch virtual dual-active summary
Pagp dual-active detection enabled: Yes
Bfd dual-active detection enabled: No
No interfaces excluded from shutdown in recovery mode
In dual-active recovery mode: Yes
  Triggered by: PAgP detection
  Triggered on interface: Te2/13/1
  Received id: 000d.662e.7d40
  Expected id: 000d.662e.7840
```



**Note**

To manage the VSS, the test team used console access and MEC to a dedicated out-of-band (OOB) management network that were not excluded from the dual-active recovery mode.

*Bidirectional Forwarding Detection (BFD)*

BFD dual-active detection requires a dedicated interface on each VSS member switch. These interfaces are not active unless the VSL link goes down. The following is a sample configuration:

Switch 1:

```
interface GigabitEthernet1/5/1
  no switchport
  ip address 10.7.230.1 255.255.255.0
  bfd interval 100 min_rx 100 multiplier 50
!
```

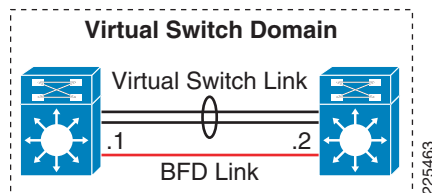
Switch 2:

```
interface GigabitEthernet2/5/1
  no switchport
  ip address 10.7.230.2 255.255.255.0
  bfd interval 100 min_rx 100 multiplier 50
!
```

Figure 26 depicts the physical connectivity of the VSL and BFD links. The VSS domain defines the use of BFD dual-active detection. As the following configuration shows, the GigabitEthernet interfaces 1/5/1 and 2/5/1 are trusted by the VSS for VSL failure detection:

```
switch virtual domain 100
  dual-active pair interface GigabitEthernet1/5/1 interface GigabitEthernet2/5/1 bfd
```

**Figure 26** VSS BFD Example



#### *Fast Hello Packets*

To use the dual-active fast hello packet detection method, a direct Ethernet connection must be provisioned between the two switches. Up to four non-VSL links can be dedicated for this purpose.

The two chassis periodically exchange special Layer 2 dual-active hello messages containing information about the switch state. If the VSL fails and a dual-active scenario occurs, each switch recognizes from the peer's messages that there is a dual-active scenario and initiates recovery. If a switch does not receive an expected dual-active fast hello message from the peer before the timer expires, the switch assumes that the link is no longer capable of dual-active detection.

Cisco IOS Release 12.2(33)SXI and later releases support the dual-active fast hello method.

## Network Design with VSS

**Figure 27** STP Configuration for VSS Loop-Free Design

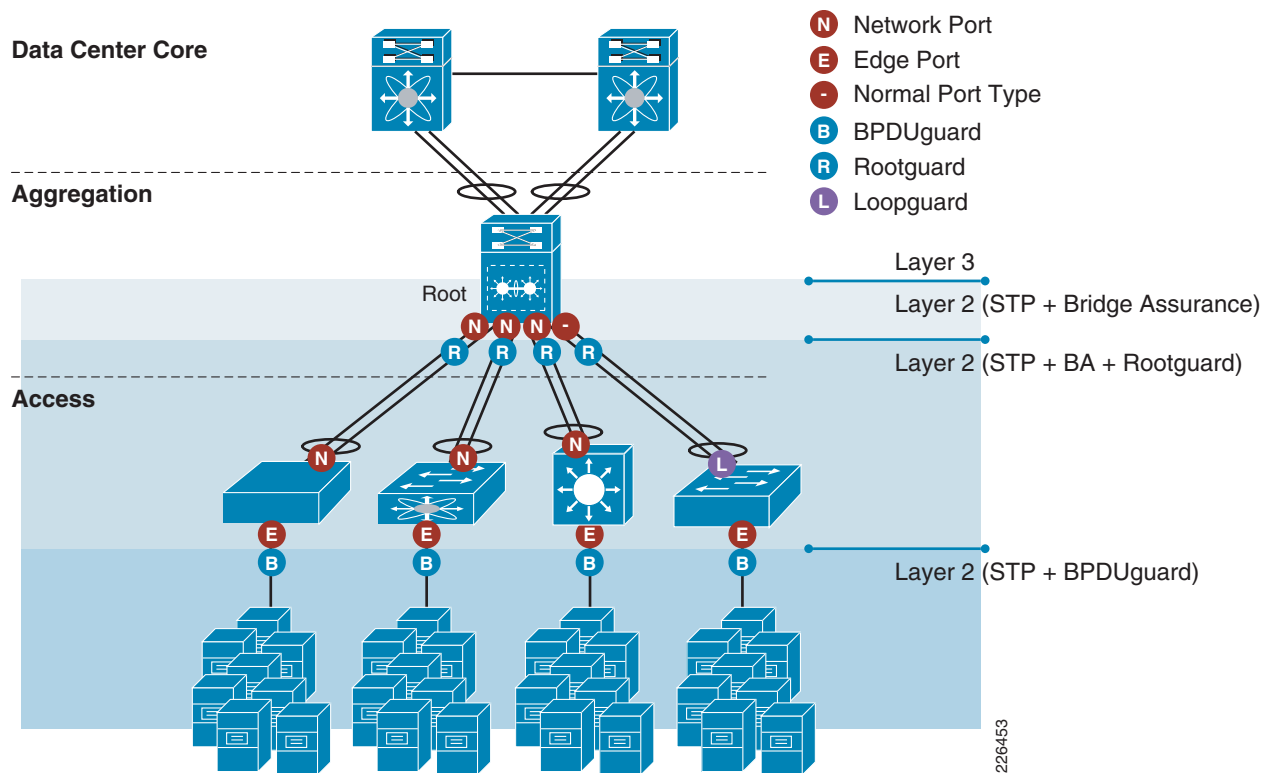


Figure 27 illustrates the resulting data center network after the implementation of VSS on the aggregations switches. VSS simplifies the Layer-2 domain configuration to its bare minimum, as the resulting topology is a logical star.

The general network design recommendations provided in the “[Network Design with STP](#)” section on [page 45](#) still apply. STP still runs as a safeguard against configuration error or channel failures. The root bridge is naturally located on the now unique aggregation switch. There is no need for HSRP anymore.

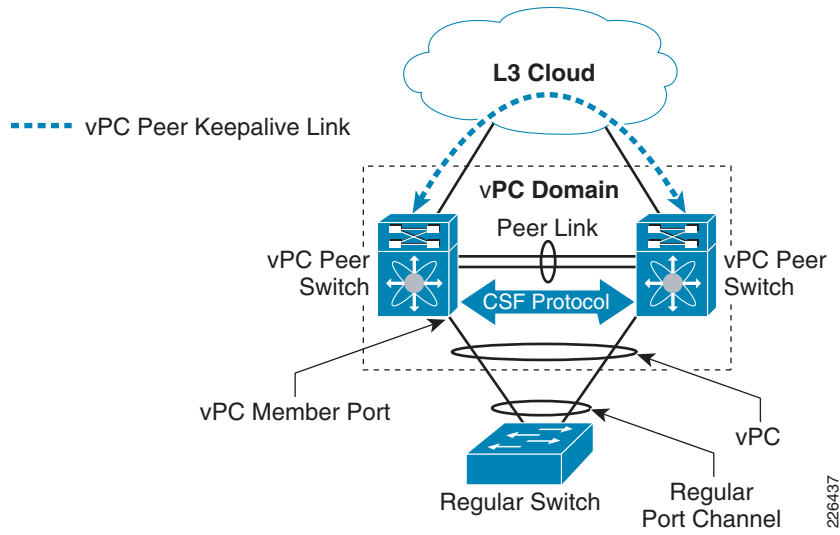
## Virtual Port Channel (vPC)

### Concept

A vPC allows creating a Layer-2 port channel between two Cisco Nexus 7000 Series devices on one end and a third device (switch, router, server, etc) on the other end. The concept is slightly different from VSS in the way that the two Nexus 7000 are still independent switches, with different control and forwarding planes. In term of Layer 2 design, vPC implements an alternate way of providing MEC, the most important feature of VSS.

Figure 28 represents the two Nexus 7000 handling a vPC to a non-vPC access switch.

**Figure 28 Virtual Port Channel Elements**



## Components

### Virtual Port Channel Domain

A vPC domain is group of two vPC peers switches running vPC. The domain has a unique identifier.

### Virtual Port Channel Peer Switches

The two switches that are aggregated by the vPC feature as the end of a distributed channel. Only two switches can be part of this peer relationship, but as mentioned for VSS, this simple property is enough to build a loop-free redundant and scalable data center.

### Virtual Port Channel

A vPC is port channel with at least one end distributed across two vPC peer switches.

### Virtual Port Channel Peer Link

This is the link between the vPC peer switches. The whole architecture relies heavily on the availability of this link, so it is recommended to configure it as a port channel with members spread across different line cards. That aside, the peer link is just a regular interface with the ability to tag packets as having originated on the local peer.

### Virtual Port Channel Member Port

A vPC member port is a physical port on a vPC member that is associated to a vPC.

### Virtual Port Channel Peer Keepalive Link

When the vPC peer link is down, it is important to differentiate between the failure of the vPC peer link alone and the failure of the directly attached vPC peer as a whole. The peer-keepalive link allows the vPC peers to exchange heartbeat messages without using the peer-link. This mechanism is critical to prevent dual-active scenarios in case of peer-link failure. This is similar to the VSS BFD design.

### Cisco Fabric Services (CFS) Protocol

The CFS protocol is a reliable messaging protocol designed to support rapid stateful configuration message passing and synchronization. The vPC uses CFS to transfer a copy of the system configuration for a comparison process, and to synchronize protocol state information between the two vPC peer switches.

## Configuration

The configuration of vPC is relatively straightforward and does not add much to a regular port channel configuration.

### Configuring the vPC Domain

The vPC feature must first be globally enabled using the **feature vpc** command as shown in the following example:

```
feature vpc
vpc domain 3
    role priority 32000
    peer-keepalive destination 172.28.193.95
```

Currently, a vPC pair represented by two Nexus 7000 devices or two VDCs carved in two physically independent Nexus 7000 devices can only participate in a single VPC domain. However, the CLI provides the option to configure a VPC domain number in the range of 1 to 1000. Each VPC has a configured and operational role. The configured role depends on the role priority. The role priority is a 16-bit integer with a default value of 32767. The bridge with the lowest configured role priority will come up as a VPC primary, while the other is a VPC secondary. In case the priority is identical on both peers, some additional tie-breakers including a MAC address, will be used in order to select a primary in a deterministic way. The primary VPC member is responsible for handling the control plane associated with the VPCs in a centralized way. Currently, there is not much value in the role configuration because the operational role can change as a result of the failure of the primary without any possibility for the configured primary to preempt its role back. However, it is recommended to configure a lower priority on a VPC peer in order to identify easily which switch will become the primary vPC device after the system boots up.

The peer-keepalive is identified by a source IP address (local), a destination IP address (remote) and a VRF that will carry the peer-keepalive traffic.

### Configuring the vPC Peer Link

The peer link is a regular port channel that is just identified by the additional **vpc peer-link** command. The following is an extract from the configuration of a vPC peer-link of the test data center used for this document.

```
interface port-channel99
    description < peer link to n74b >
    switchport
    switchport mode trunk
    switchport trunk allowed vlan 151,153,163-167,180-181
    vpc peer-link
    spanning-tree port type network
```

The peer link is critical to the operation of vPC. Even if only a very small part of the network traffic should use this link in normal condition (see [“Packet Flow/Convergence” section on page 64](#)), it is important to reserve enough bandwidth on this channel as it will be used to forward traffic to a neighbor that would have lost its physical connection to one of the vPC peers. The peer link should also be resilient

to the failure of a single module, as its loss would result in degraded network performances. The recommendation is thus to use at least two dedicated 10-Gigabit interfaces in dedicated mode from two different modules.

### Configuring the vPCs

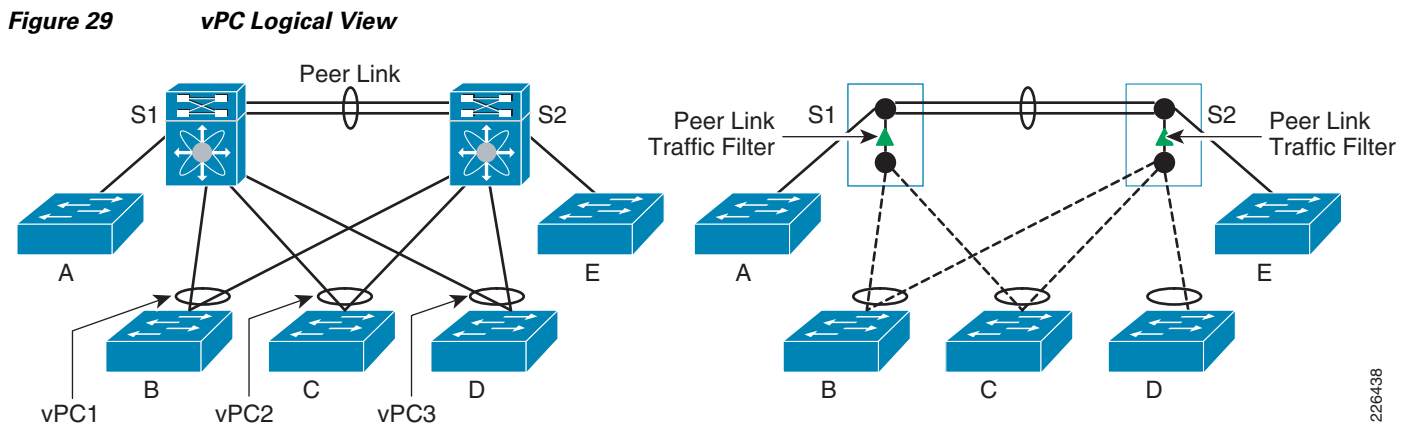
The vPC configuration is just a matter of identifying the vPC member ports on the vPC peers. On each vPC peer, this is a regular port channel configuration with the addition of the **vpc <vpc\_number>** command, as shown in the following example:

```
interface port-channel68
  description < vpc to c68 access >
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 164-167
  vpc 68
  spanning-tree port type network
  logging event port link-status
  logging event port trunk-status
```

The above configuration identifies vPC number 68. This ID is exchanged between the vPC peers and is used to bind the two separate local port channel configuration into a distributed vPC virtual port. The number listed in the **vpc <vpc\_number>** command does not have to match the port channel number. Still, it is considered a best practice to configure those values the same.

### Packet Flow/Convergence

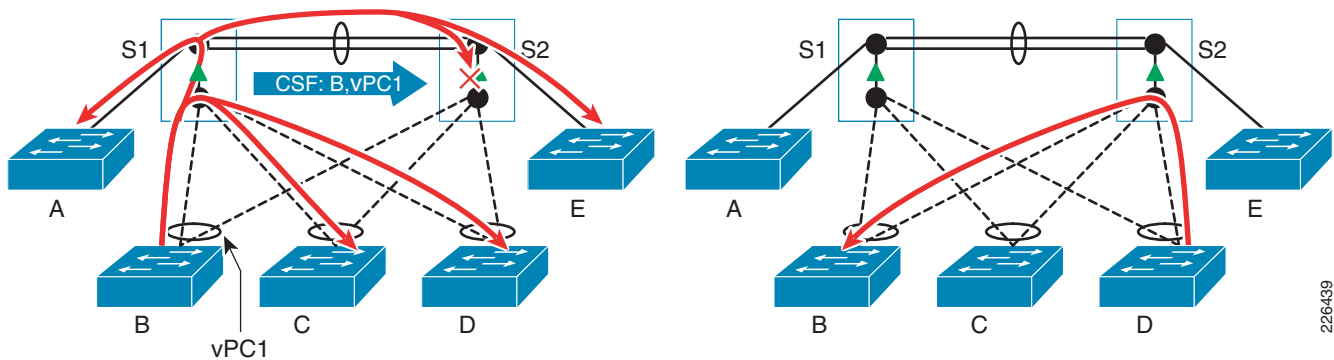
The hardware support for required vPC is illustrated in [Figure 29](#).



The left part of [Figure 29](#) represents five switches connected to a pair of vPC peers S1 and S2. Switches A and E are connected through a single link to the vPC domain, while switches B, C, and D are connected through a channel to the vPC peer switches. The principal characteristic of the model is that all the devices attached to a vPC can be reached directly from a vPC peer. For example S1 can send traffic directly to B, C, or D; however, S1 can only reach switches E through S2. One of the goals of vPC is to send traffic directly to its destination, thus minimizing the use of the peer link. The right part of [Figure 29](#) is a representation of the same network design that introduces an artificial "peer link traffic filter" between the lower vPC member ports and the upper ports (of course, this filter is just a logical representation that has no relation with the real hardware implementation.) Traffic that has crossed the peer link is tagged internally and will not be allowed to be forwarded through the filter. This mechanism

will allow the traffic received from the vPC member port to be locally forwarded, while still providing connectivity to the ports that are not part of a vPC. Figure 30 shows the example of an exchange between switches B and D.

**Figure 30** vPC Traffic Flow

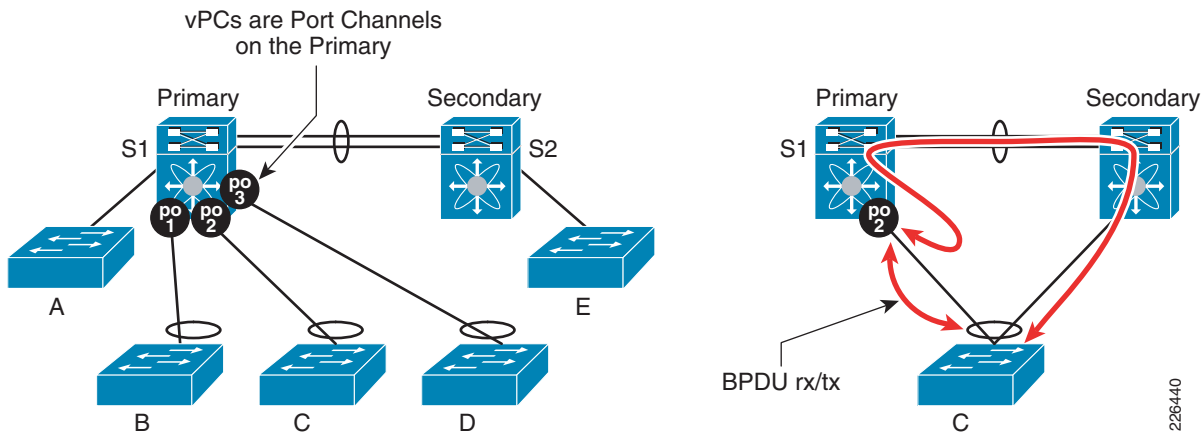


When switch B sends a frame to switch D, the destination address for switch D is unknown and the traffic must be flooded. Again, all the devices belonging to a vPC can be reached directly and S1 replicates the frame to the vPC member ports leading to switches C and D. However, the frame must also be flooded to the non-vPC members. When it is sent on the peer link, an internal header carrying a special bit is added to the frame in order to specify that this traffic has already been sent to the vPC members. As a result, when vPC peer S2 receives the frame, the filter prevents it from being duplicated to its local vPC members and it is only forwarded to switch E. At the same time, a software update carried by CFS advertises to S2 that MAC address B was learnt on vPC. This information will allow S2 to send the reply from switch D to switch B directly on its local vPC member port, even if S2 never received traffic from switch B on this port.

### Spanning Tree Representation

In the case of VSS, the pair of aggregated switches is sharing a unique control plane. The vPC takes a different approach as vPC devices are managed independently and separate instances of network protocols exist on the vPC peers. During the vPC domain setup, a vPC peer is elected as primary. This peer will be responsible for running STP on all the vPC ports of the vPC domain. So logically, as depicted on the left part of Figure 31, a vPC is a simple channel located on the primary vPC peer switch from the perspective of STP. The state of the vPC member ports located on the secondary peer is controlled remotely by the primary.

**Figure 31** *Spanning Tree Logical View*



Still, BPDUs can be exchanged on all the physical links belonging to a vPC. The right part of [Figure 31](#) illustrate how a vPC, logically represented as port channel 2 on the primary switch S1 can send and receive BPDUs on both paths available to bridge C. Switches S1 and S2 are programmed so that the BPDUs can be switched in hardware toward their final destination.

Note that some STP information about the vPCs is still available on the secondary vPC peer, but it is just replicated from the primary.

**Interaction with HSRP/PIM Enhancement**

A number of enhancements have been made to the vPC solution in order to integrate with the Layer 3 features. Specifically, HSRP and PIM interaction are modified to improve scalability and system resiliency.

With HSRP, the improvement was made to the forwarding engine to allow local Layer 3 forwarding at both the active HSRP peer and at the standby HSRP peer. This provides in effect an active/active HSRP configuration with no changes to current HSRP configuration recommendations or best practices and no changes to the HSRP protocol either. The HSRP control protocol still acts like an active/standby pair, such that only the active device responds to ARP requests, but a packet destined to the shared HSRP MAC address is accepted as local on either the active or standby HSRP device.

**Figure 32** HSRP and vPC

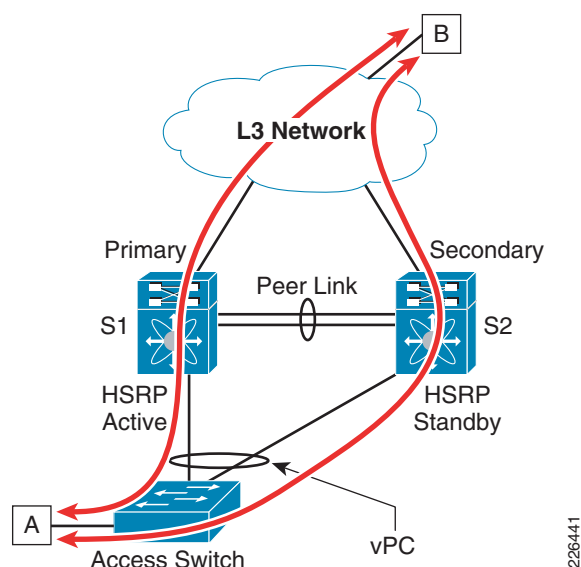


Figure 32 shows an example of this process. The traffic from A to B is load balanced by the hashing algorithm implemented by the port channel of the access switch. Even when a frame is directed toward S2, the HSRP standby is able to intercept it and route it directly to its final destination, thus saving the vPC peer link. The traffic in the other direction will be load balanced to S1 or S2 by the ECMP mechanism of Layer 3, and will reach A using the directly attached vPC member port in the vPC.

The other key Layer-3 interaction is in conjunction with PIM multicast routing. A vPC supports an enhancement to PIM SSM and PIM ASM routing to further improve resiliency in the case of a peer level failure, and in order to guarantee the most efficient multicast forwarding into our out of the vPC attached environment.

Specifically, since there are by requirement two Layer 3 instances (one on each peer switch) it is necessary to have 2 PIM routers configured, one on each device. Normally, both devices would assert for the role of the designated router on the local Layer 2 multicast environment. In this case, the router that wins the assert process then joins the Layer-3 multicast environment, and starts forwarding multicast traffic to any joined clients, or starts forwarding any multicast traffic source on the local network if there is a client request in the Layer 3 network. The router that lost assert will not actually join the Layer 3 network, and will instead wait until the next assert interval, at which point it will again try to win the assert test. In the case of vPC, the PIM process was enhanced so that both routers will join the Layer 3 network, but only the router on the primary vPC peer will win the assert process. The other device will then act as an active standby, ready to forward packets into the multicast network if the primary device fails to assert after the default reassert interval passes. This model provides faster recovery in the case of a forwarding failure, and since the Layer 2 state is already synchronized between both devices, it is not necessary for the standby DR to rebuild the local Layer 2 forwarding state either.

### Convergence

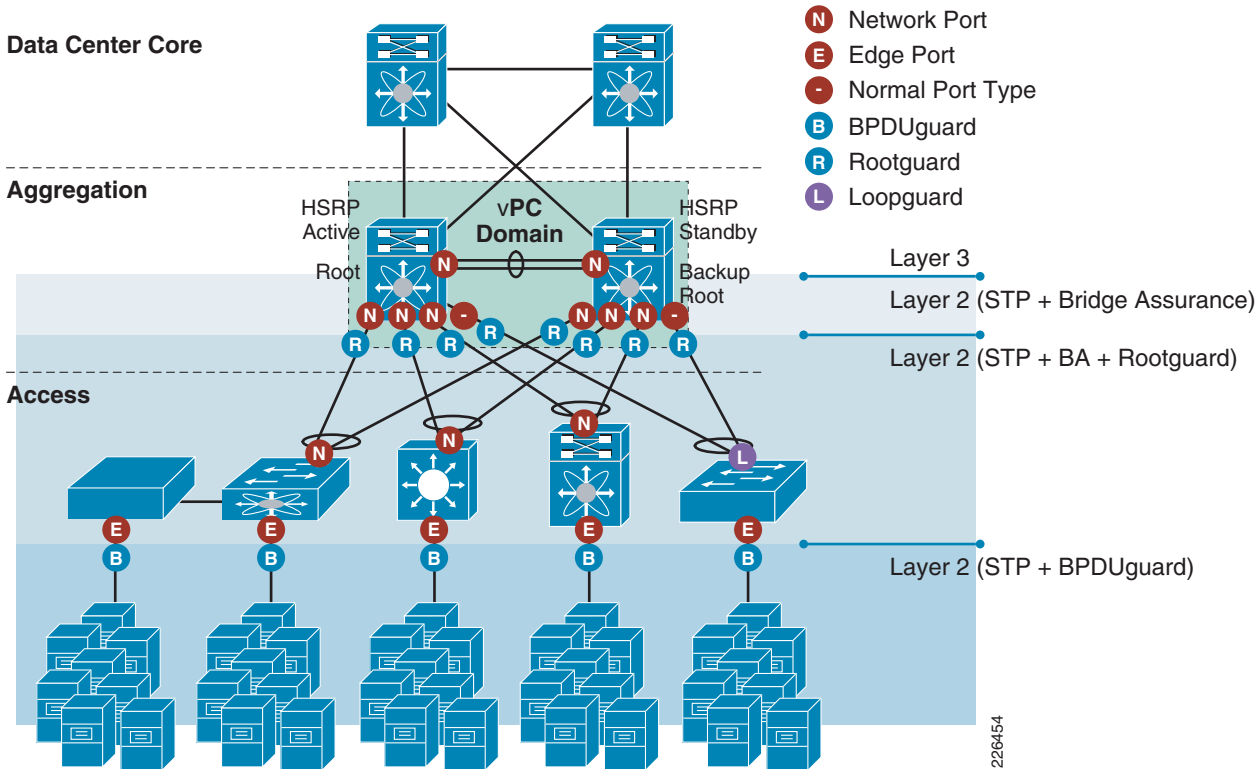
The advantage of the vPC approach is that in the event of a port channel member link failure, recovery uses the port channel recovery mechanism rather than STP. When a vPC member port fails, if there is no other local vPC member port on the vPC peer, a notification is sent to the remote vPC peer so that it will start sending traffic over the vPC peer link for that particular vPC.

The other significant convergence scenario to consider is the peer link failure. The vPC management system will use the peer-keepalive interface to determine if the failure is a link level failure or if in fact the remote peer has failed entirely. In the case that the remote peer is still alive (peer-keepalive messages

are still being received), the vPC secondary switch will disable its vPC member ports and any Layer 3 interfaces attached to a vPC associated VLAN. If the peer-keepalive messages are not being received, then the peer continues to forward traffic, because it assumes that it is the last device available in the network. In case of recovery of the peer link, the systems will resynchronize any MAC addresses learned while communications was disrupted, and the system will continue forwarding normally.

### Network Design with vPC

Figure 33 vPC Loop-Free Design



Even if there is no real use for STP and even to HSRP to a certain extent, vPC still requires those features to be configured in the network. As a result, Figure 33, which represents the vPC loop-free data center design, is very close to Figure 21, which illustrates the STP-based solution. From an operational perspective, however, the vPC solution provides the same benefit as VSS. The following are additional vPC-specific recommendations:

- It is recommended to dual-attach all the access switches to the vPC peers. A switch or host that would only connect to one vPC peer would consume precious bandwidth on the peer link. In case of failure of the peer link or a vPC peer, it would also risk being entirely isolated from the network. Another reason is that such single homing setup will bring orphan ports in the configuration and can trigger Layer-3 SVIs mapped to such VLAN to come up before the vPCs are formed and peer-link is ready, thus causing black-holing for traffic coming from the core towards the access.
- Because the primary vPC peer always handle STP for all the vPCs in the domain, trying to distribute the root bridge across the two vPC peers is not useful. Locating the root bridge for all the spanning tree instances on the primary vPC peer is the recommended configuration, at least because it is the most simple. In order to keep the default gateway collocated on the root bridge, the same vPC primary peer will be configured as an HSRP active for all the VLANs.

- Use LACP (over static link aggregation) when possible on vPCs and peer-link to detect mis-configurations and to provide more graceful failover handling.
- Use UDLD and bridge assurance on the vPC peer-links to detect unidirectional and bidirectional failures on this link
- It is highly discouraged to carry the peer-keepalive communication over the peer-link. They must be separate to ensure vPC system resiliency. A deadlock condition will occur when both peer-link and peer-keepalive are down. In order to form or recover a peer-link a working peer-keepalive link must first exist, that is why they should be independent.
- Use a separate VRF and front panel ports for the peer-keepalive link (1G is more than adequate). An alternate is to use the management interface(s) for the vPC peer-keepalive link access. The peer-keepalive link is a layer 3 logical link. By default it runs on top of the management network between the two vPC peers, and consumes very little bandwidth, allowing regular management traffic to leverage the same ports. Redundant physical peer-keepalive ports can be used to ensure logical keepalive connectivity.
- When using redundant supervisors, only one management interface is active at a time on a given vPC peer. The management interfaces thus cannot be connected back-to-back between vPC peers in order to create the peer-keepalive link. An intermediate switch is needed so that the peer-keepalive heartbeat can be forwarded regardless of the active supervisor.

## Conclusion

Cisco Validated Design Guides (CVDs) often cover narrowly defined designs with specific configuration examples for networking devices deployed in a certain way. The focus of this design guide has been to provide an overview of topologies, best practices, new products and features to the network architect to enhance an existing data center network or prepare for a Greenfield data center implementation. Data center networking is currently an area of intensive development for Cisco, new products and features will continue to be released over the coming months to expand the tools and options available when designing a data center topology. The Cisco CVD program will continue to provide both generalized design overview documentation as well as specific configuration examples in more narrowly defined documents to assist our customer base in keeping their data center topologies prepared for ever-changing business needs.

## Additional References

- *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](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](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/vssdc_integrate.html)
- *Data Center Service Patterns*  
[http://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/DC\\_3\\_0/dc\\_serv\\_pat.html](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_3_0/dc_serv_pat.html)

# Cisco Validated Design

The Cisco Validated Design Program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information visit [www.cisco.com/go/validateddesigns](http://www.cisco.com/go/validateddesigns).

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

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

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0807R)