BCMSN

Building Cisco Multilayer Switched Networks

Volume 1

Version 3.0

Student Guide

EPGS Production Services: 07.27.06

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.com Website at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Students, this letter describes important course evaluation access information!

Welcome to Cisco Systems Learning. Through the Cisco Learning Partner Program, Cisco Systems is committed to bringing you the highest-quality training in the industry. Cisco learning products are designed to advance your professional goals and give you the expertise you need to build and maintain strategic networks.

Cisco relies on customer feedback to guide business decisions; therefore, your valuable input will help shape future Cisco course curricula, products, and training offerings. We would appreciate a few minutes of your time to complete a brief Cisco online course evaluation of your instructor and the course materials in this student kit. On the final day of class, your instructor will provide you with a URL directing you to a short post-course evaluation. If there is no Internet access in the classroom, please complete the evaluation within the next 48 hours or as soon as you can access the web.

On behalf of Cisco, thank you for choosing Cisco Learning Partners for your Internet technology training.

Sincerely,

Cisco Systems Learning

Table of Contents

Volume 1

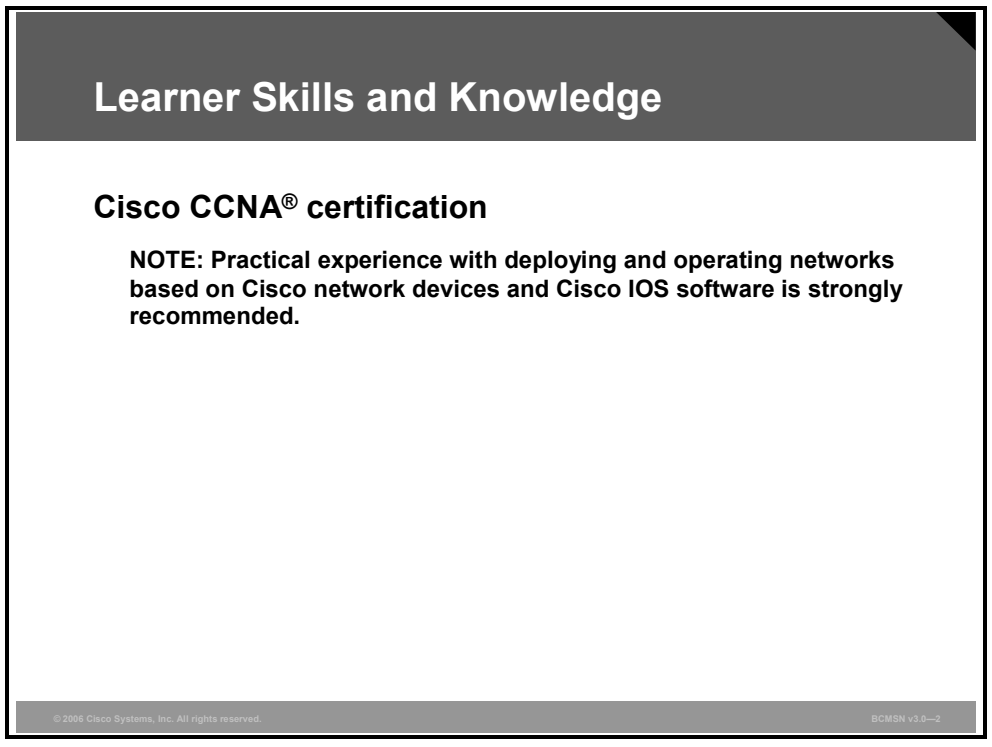
Course Introduction

Overview

Building Cisco Multilayer Switched Networks (BCMSN) v3.0 is recommended training for individuals seeking Cisco CCNP® certification. The course instructs network administrators of campus area network sites on the use of advanced multilayer switches in implementing a scalable topology based upon Cisco Systems technologies. The goal is to train network administrators in the technology and capabilities of multilayer switches to allow for supporting a dramatic increase the number of end stations, and the interleaving of voice, video, and data, while ensuring a reliable network infrastructure.

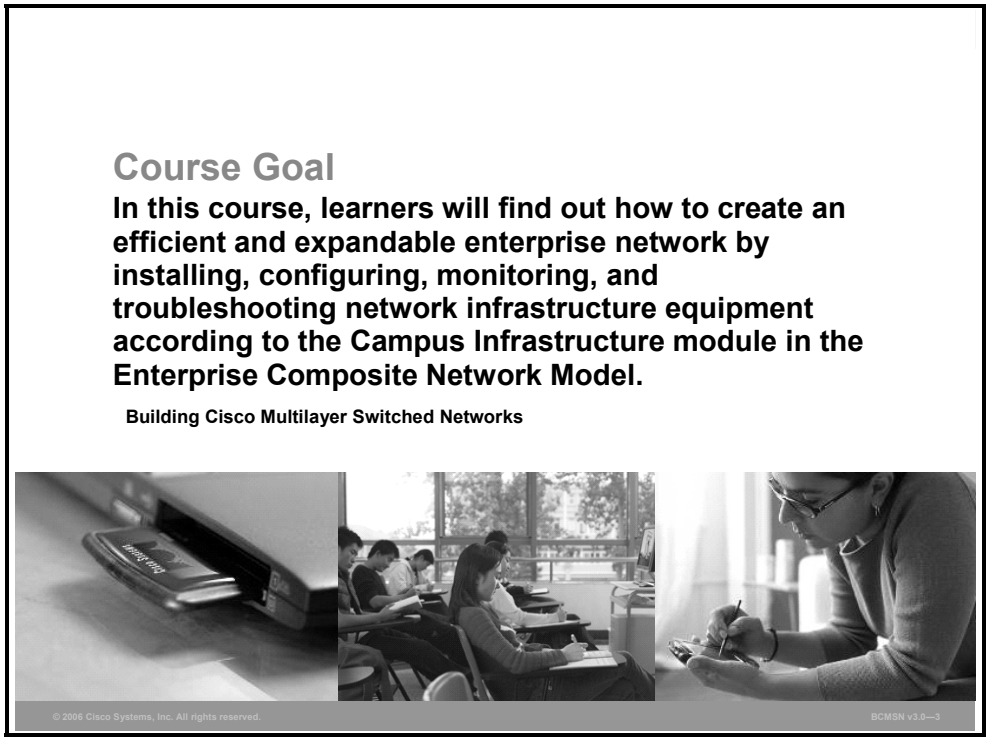
Learner Skills and Knowledge

This topic lists the skills and knowledge that learners must possess to benefit fully from the course. The subtopic also includes recommended Cisco learning offerings that learners should complete to benefit fully from this course.



Course Goal and Objectives

This topic describes the course goal and objectives.

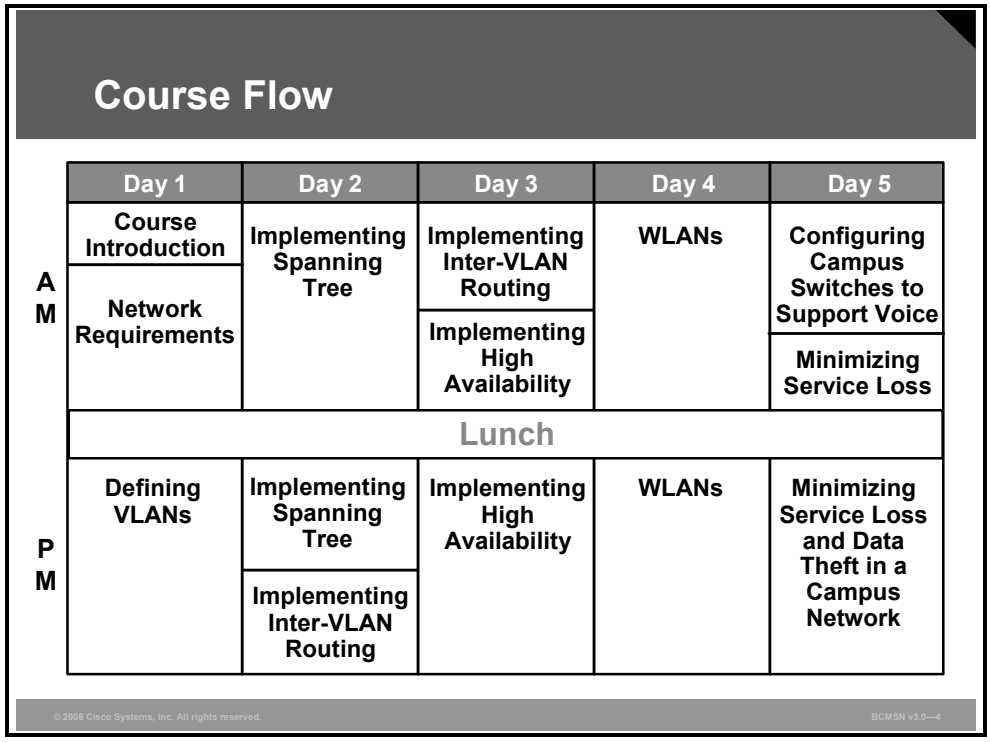


Upon completing this course, you will be able to meet these objectives:

- Describe the Campus Infrastructure module of the ECNM
- Define VLANs to segment network traffic and manage network utilization
- Explain the procedure for configuring both 802.1Q and ISL trunking between two switches so that VLANs that span the switches can connect
- Describe how VLAN configuration of switches in a single management domain can be automated with the Cisco proprietary VTP
- Implement high availability technologies and techniques using multilayer switches in a campus environment
- Describe WLANs
- Describe and configure switch infrastructure to support voice
- Describe and implement security features in a switched network

Course Flow

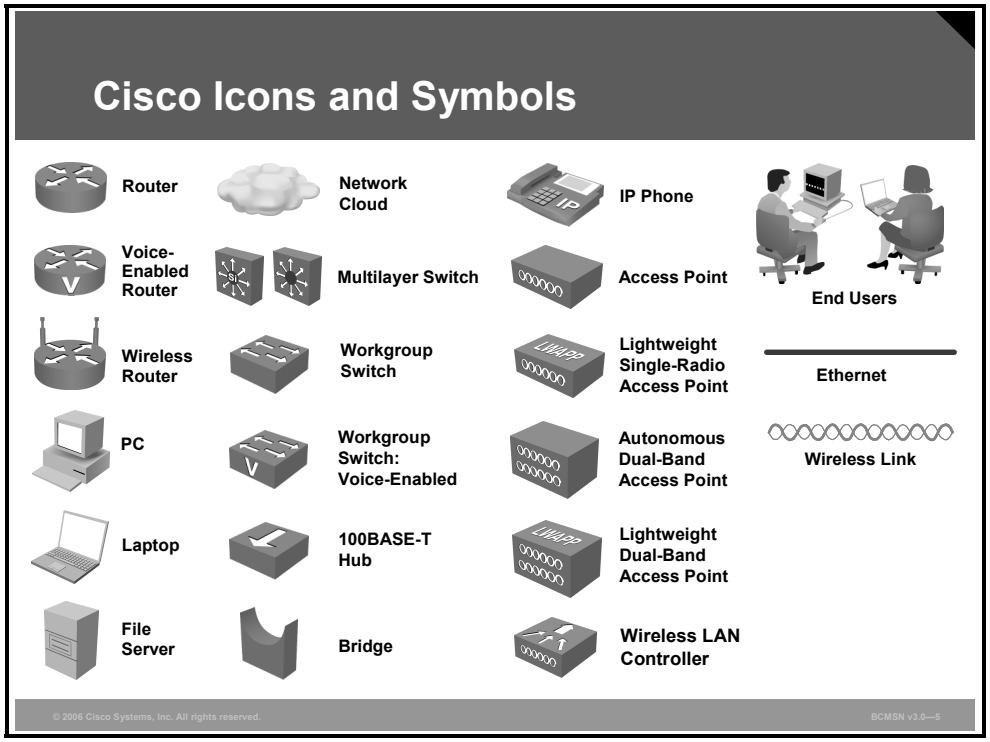
This topic presents the suggested flow of the course materials.



The schedule reflects the recommended structure for this course. This structure allows enough time for the instructor to present the course information and for you to work through the lab activities. The exact timing of the subject materials and labs depends on the pace of your specific class.

Additional References

This topic presents the Cisco icons and symbols used in this course, as well as information on where to find additional technical references.



Cisco Glossary of Terms

For additional information on Cisco terminology, refer to the *Cisco Internetworking Terms and Acronyms* glossary of terms at http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm.

Your Training Curriculum

This topic presents the training curriculum for this course.

You are encouraged to join the Cisco Certification Community, a discussion forum open to anyone holding a valid Cisco Career Certification (such as Cisco CCIE®, CCNA®, CCDA®, CCNP®, CCDP®, CCIP®, CCSP™, or CCVP™).

It provides a gathering place for Cisco-certified professionals to share questions, suggestions, and information about Cisco Career Certification programs and other certification-related topics. For more information, visit http://www.cisco.com/en/US/learning/le3/le2/le37/learning_certification_level_home.html.

Learner Introductions

- **Your name**
- **Your company**
- **Skills and knowledge**
- **Brief history**
- **Objective**

Module 1

Network Requirements

Overview

This module looks at the need for multilayer switches within Cisco's overall network design. A review of Intelligent Information Networks (IIN) and Service-Oriented Network Architectures (SONA) will set the groundwork for the course ahead.

Additionally a quick overview of the characteristics of layer 2 and layer 3 networks will aid in identifying the reasons for using a multi-layer switch. Students will learn how issues that exist in traditionally designed networks can be resolved by applying this state-of-the-art design to their networks.

Module Objectives

Upon completing this module, you will be able to explain the Cisco hierarchical network model as it pertains to the campus network. This ability includes being able to meet these objectives:

- Describe the Campus Infrastructure module of the ECNM

Lesson 1

Introducing Campus Networks

Overview

This lesson begins by discussing operational problems found in nonhierarchical networks at Layers 2 and 3 of the Open Systems Interconnection (OSI) model. The Enterprise Composite Network Model (ECNM) is then introduced, and the features and benefits of ECNM are explained. Students will learn how issues that exist in traditionally designed networks can be resolved by applying this state-of-the-art design to their networks.

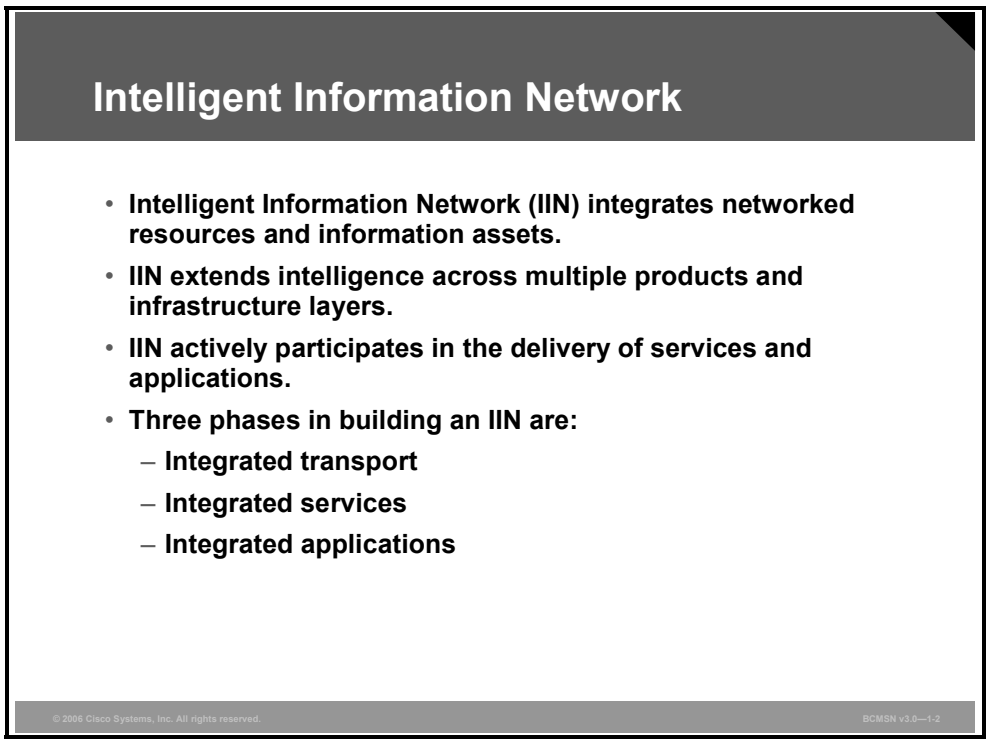
Objectives

Upon completing this lesson, you will be able to describe the Campus Infrastructure module of the ECNM. You will also be able to identify the structure and components used to build or expand a network in the Campus Infrastructure module. This ability includes being able to meet these objectives:

- Define IIN and Cisco SONA frameworks
- Describe the Cisco Enterprise Architecture and how it maps to the traditional three-layer hierarchical network model
- Describe the devices in a nonhierarchical network
- Identify problems that can occur in a nonhierarchical switched network
- Identify problems that can occur in a nonhierarchical routed network
- Define multilayer switches in a nonhierarchical network
- List the issues that occur with multilayer switches and VLANs in a nonhierarchical network
- Describe the Enterprise Composite module, which can be used to divide the enterprise network into physical, logical, and functional boundaries
- List the benefits of the ECNM
- Describe the Campus Infrastructure module of the ECNM
- Identify the two interfaces used to configure Cisco Catalyst switches

IIN and Cisco SONA Framework

This topic describes the Intelligent Information Network (IIN), its features, and the Cisco Service-Oriented Network Architecture (SONA) that guides an evolution of enterprise networks toward IIN.



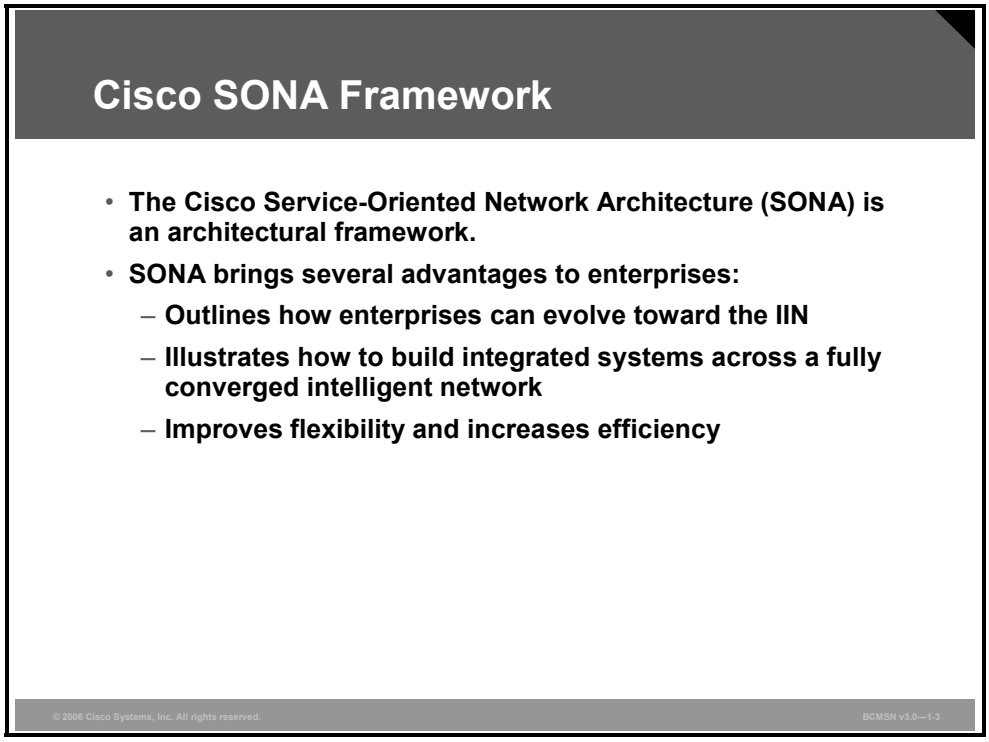
The Cisco vision of the future IIN encompasses these features:

- **Integration of networked resources and information assets that have been largely unlinked:** The modern converged networks with integrated voice, video, and data require that Information Technology (IT) departments more closely link the IT infrastructure with the network.
- **Intelligence across multiple products and infrastructure layers:** The intelligence built into each component of the network is extended network-wide and applies end-to-end.
- **Active participation of the network in the delivery of services and applications:** With added intelligence, the IIN makes it possible for the network to actively manage, monitor, and optimize service and application delivery across the entire IT environment.

With the listed features, the IIN offers much more than basic connectivity, bandwidth for users, and access to applications. The IIN offers end-to-end functionality and centralized, unified control that promotes true business transparency and agility.

The IIN technology vision offers an evolutionary approach that consists of three phases in which functionality can be added to the infrastructure as required:

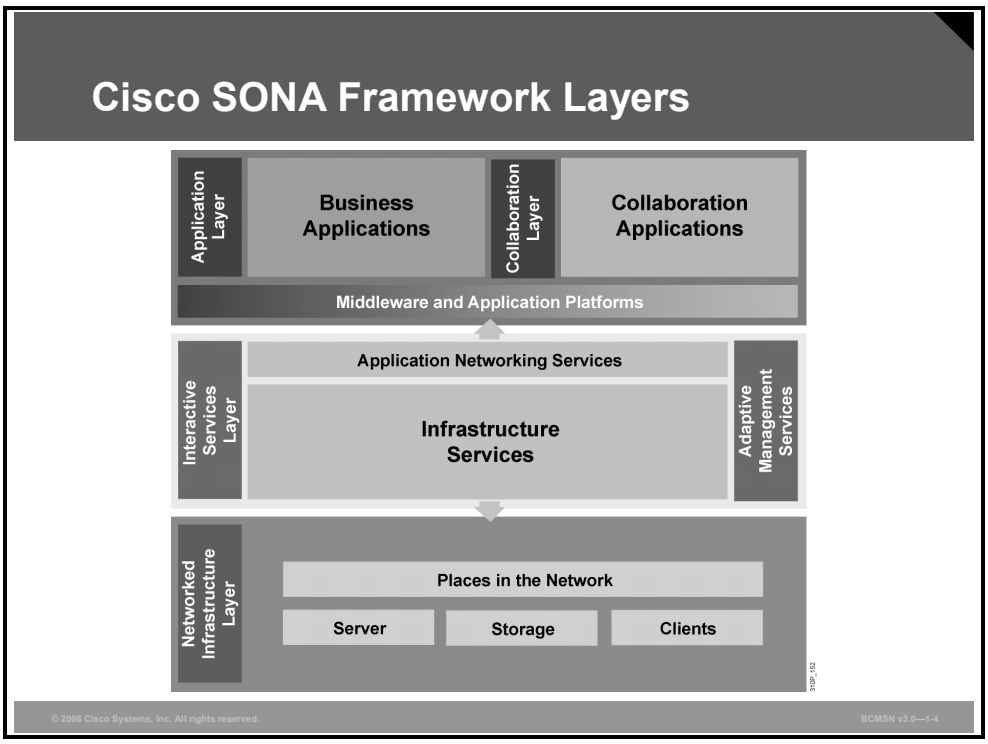
- **Integrated transport:** Everything—data, voice, and video—consolidates onto an IP network for secure network convergence. By integrating data, voice, and video transport into a single, standards-based, modular network, organizations can simplify network management and generate enterprise-wide efficiencies. Network convergence also lays the foundation for a new class of IP-enabled applications delivered through Cisco IP Communications solutions.
- **Integrated services:** After the network infrastructure has been converged, IT resources can be pooled and shared or "virtualized" to flexibly address the changing needs of the organization. Integrated services help to unify common elements, such as storage and data center server capacity. By extending virtualization capabilities to encompass server, storage, and network elements, an organization can transparently use all its resources more efficiently. Business continuity is also enhanced because shared resources across the IIN provide services in the event of a local system failure.
- **Integrated applications:** With Application-Oriented Networking (AON) technology, Cisco has entered the third phase of building the IIN. This phase focuses on making the network "application-aware" so it can optimize application performance and more efficiently deliver networked applications to users. In addition to capabilities such as content caching, load balancing, and application-level security, Cisco AON makes it possible for the network to simplify the application infrastructure by integrating intelligent application message handling, optimization, and security into the existing network.



With its vision of the IIN, Cisco is helping organizations to address new IT challenges, such as the deployment of service-oriented architectures, Web services, and virtualization. Cisco SONA is an architectural framework that guides the evolution of enterprise networks to an IIN. The Cisco SONA framework provides several advantages to enterprises, such as the following:

- Outlines the path towards the IIN
- Illustrates how to build integrated systems across a fully converged IIN
- Improves flexibility and increases efficiency, which results in optimized applications, processes, and resources

Cisco SONA uses the extensive product line services, proven architectures, and experience of Cisco and its partners to help the enterprises achieve their business goals.



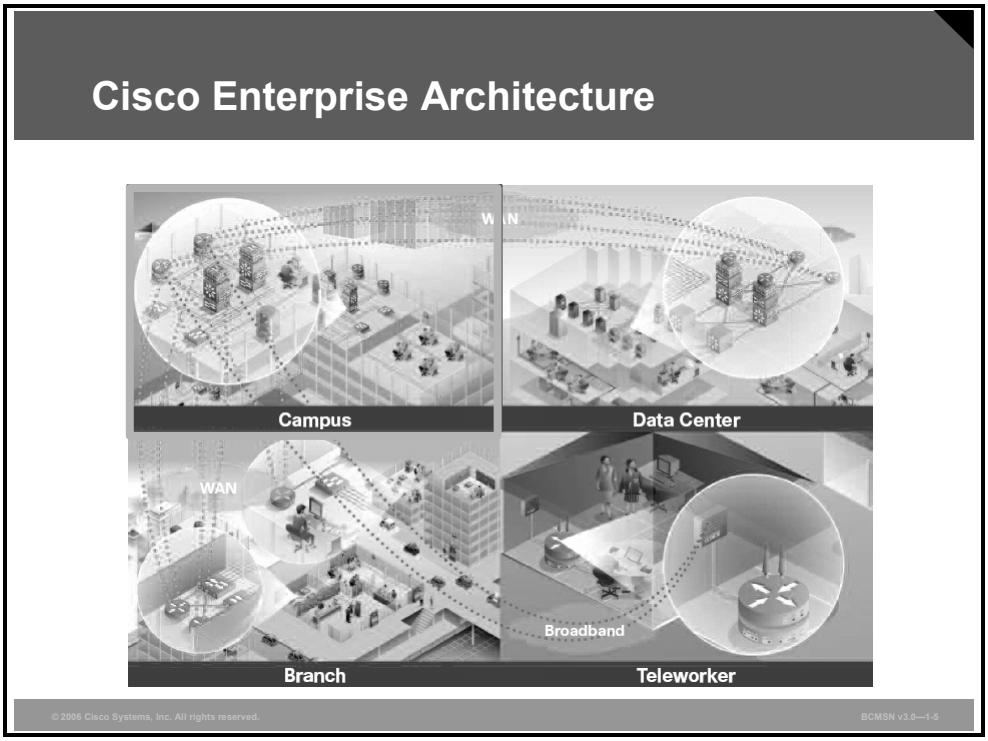
The Cisco SONA framework shows how integrated systems can both allow a dynamic, flexible architecture, and provide for operational efficiency through standardization and virtualization. It brings forth the notion that the network is the common element that connects and enables all components of the IT infrastructure. Cisco SONA outlines these three layers of the IIN:

- **Network infrastructure layer:** This layer is where all the IT resources are interconnected across a converged network foundation. The IT resources include servers, storage, and clients. The network infrastructure layer represents how these resources exist in different places in the network, including the campus, branch, data center, WAN and Metropolitan Area Network (MAN), and teleworker. The objective for customers in this layer is to have anywhere and anytime connectivity.
- **Interactive services layer:** This layer enables efficient allocation of resources to applications and business processes that are delivered through the networked infrastructure. This layer comprises these services:
  - Voice and collaboration services
  - Mobility services
  - Security and identity services
  - Storage services
  - Computer services
  - Application networking services
  - Network infrastructure virtualization
  - Services management
  - Adaptive management services

- **Application layer:** This layer includes business applications and collaboration applications. The objective for customers in this layer is to meet business requirements and achieve efficiencies by leveraging the interactive services layer.

Cisco Network Models

This topic describes Cisco network models with the Cisco Enterprise Architecture and its mapping to traditional three-layer hierarchical network model.



Cisco provides the enterprise-wide systems architecture that helps companies to protect, optimize, and grow the infrastructure that supports business processes. The architecture provides integration of the entire network—campus, data center, WAN, branches, and teleworkers—offering staff secure access to the tools, processes, and services.

- **Cisco Enterprise Campus Architecture:** The Cisco Enterprise Campus Architecture combines a core infrastructure of intelligent switching and routing with tightly integrated productivity-enhancing technologies, including IP Communications, mobility, and advanced security. The architecture provides the enterprise with high availability through a resilient multilayer design, redundant hardware and software features, and automatic procedures for reconfiguring network paths when failures occur.

  Multicast provides optimized bandwidth consumption, and quality of service (QoS) prevents oversubscription to ensure that real-time traffic, such as voice and video, or critical data is not dropped or delayed. Integrated security protects against and mitigates the impact of worms, viruses, and other attacks on the network, even at the port level.

  Cisco enterprise-wide architecture extends support for standards, such as 802.1x and Extensible Authentication Protocol (EAP). It also provides the flexibility to add IP Security (IPSec) and Multiprotocol Label Switching Virtual Private Networks (MPLS VPNs), identity and access management, and VLANs to compartmentalize access. This helps improve performance and security and decreases costs. The enterprise campus architecture will be the focus of this courseware.

- **Cisco Enterprise Data Center Architecture:** The Cisco Enterprise Data Center Architecture is a cohesive, adaptive network architecture that supports the requirements for consolidation, business continuance, and security while enabling emerging service-oriented architectures, virtualization, and on-demand computing.

  IT staff can easily provide departmental staff, suppliers, or customers with secure access to applications and resources. This approach simplifies and streamlines management, significantly reducing overhead. Redundant data centers provide backup using synchronous and asynchronous data and application replication. The network and devices offer server and application load balancing to maximize performance. This solution allows enterprises to scale without major changes to the infrastructure.

- **Cisco Enterprise Branch Architecture:** The Cisco Enterprise Branch Architecture allows enterprises to extend head-office applications and services, such as security, IP Communications, and advanced application performance, to thousands of remote locations and users, or to a small group of branches.

  Cisco integrates security, switching, network analysis, caching, and converged voice and video services into a series of integrated services routers in the branch so that enterprises can deploy new services when they are ready without buying new equipment. This solution provides secure access to voice, mission-critical data, and video applications anywhere, anytime.

  Advanced network routing, VPNs, redundant WAN links, application content caching, and local IP telephony call processing provide a robust architecture with high levels of resilience for all the branch offices. An optimized network leverages the WAN and LAN to reduce traffic and save bandwidth and operational expenses. Enterprises can easily support branch offices with the ability to centrally configure, monitor, and manage devices located at remote sites, including tools, such as AutoQoS, that proactively resolve congestion and bandwidth issues before they affect network performance.

- **Cisco Enterprise Teleworker Architecture:** The Cisco Enterprise Teleworker Architecture allows enterprises to securely deliver voice and data services to remote small or home offices over a standard broadband access service, providing a business resiliency solution for the enterprise and a flexible work environment for employees. Centralized management minimizes IT support costs, and robust integrated security mitigates the unique security challenges of this environment.

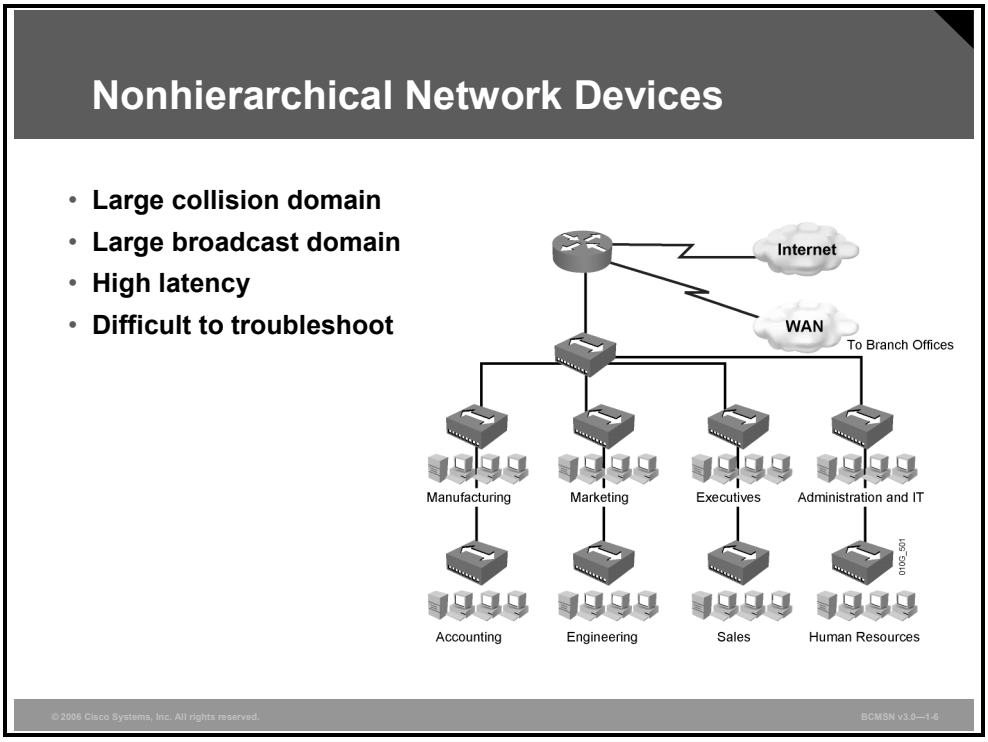
  Integrated security and identity-based networking services enable the enterprise to help extend campus security policies to the teleworker. Staff can securely log into the network over an “always-on” VPN and gain access to authorized applications and services from a single cost-effective platform. The productivity can further be enhanced by adding an IP phone, providing cost-effective access to a centralized IP Communications system with voice and unified messaging services.

- **Cisco Enterprise WAN Architecture:** The Cisco Enterprise WAN Architecture offers the convergence of voice, video, and data services over a single IP Communications network. This approach enables enterprises to cost-effectively span large geographic areas.

  QoS, granular service levels, and comprehensive encryption options help ensure the secure delivery of high-quality corporate voice, video, and data resources to all corporate sites, enabling staff to work productively and efficiently from any location. Security is provided with multiservice VPNs (IPSec and MPLS) over Layer 2 or Layer 3 WANs, hub-and-spoke, or full-mesh topologies.

Describing Nonhierarchical Campus Network Issues

This topic describes devices and their functions in a nonhierarchical network.



The simplest Ethernet network infrastructure is composed of a single collision and broadcast domain. This type of network is referred to as a "flat" network because any traffic that is transmitted within it is seen by all of the interconnected devices, even if they are not the intended destination of the transmission.

The benefit of this type of network is that it is very simple to install and configure, so it is a good fit for home networking and small offices. The downside of a flat network infrastructure is that it does not scale well as demands on the network increase. Following are some of the issues with nonhierarchical networks:

- Traffic collisions increase as devices are added, impeding traffic flow on the network.
- Broadcast traffic increases as devices are added to the network, causing overutilization of network resources.
- Problem isolation on a large flat network can be difficult.

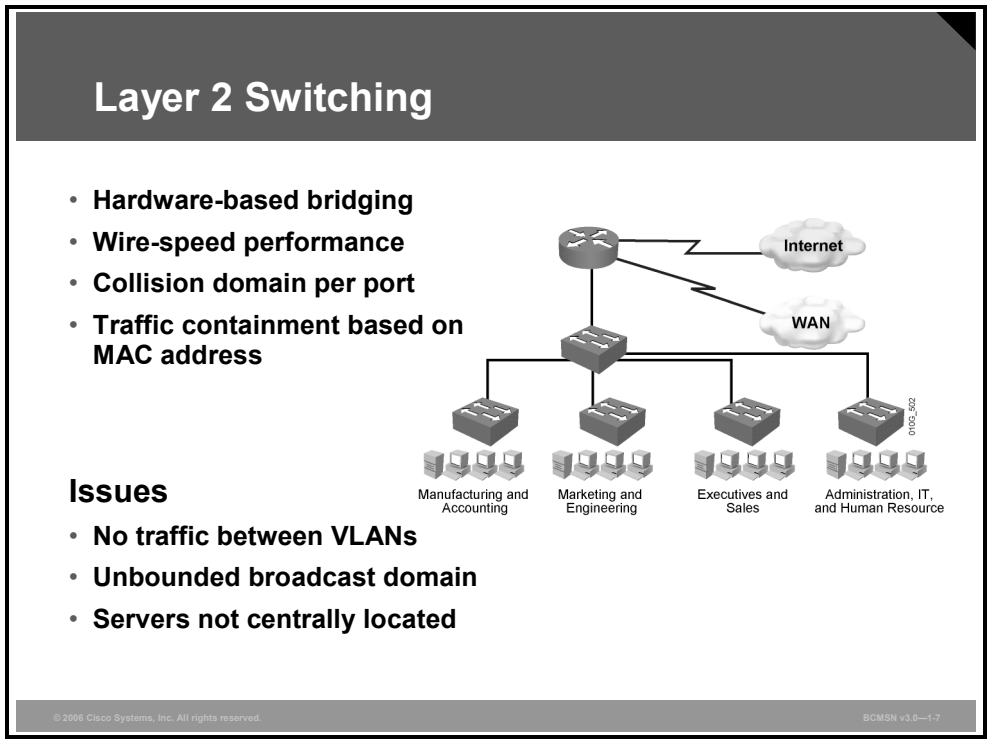
Network Devices

The table shows the key network hardware devices in a nonhierarchical network and the function of each.

Device	Function
Hub	Layer 1 device used to interconnect networking components such as PCs, printers, hubs, and routers. This device creates a single broadcast and collision domain for all networking components to which it is connected. Hubs have been superseded in networks by inexpensive switches.
Switch	Layer 2 device used to interconnect networking components such as PCs, printers, hubs, and routers. In its default configuration, this device creates a single broadcast domain for devices connected to it. Each port acts as a separate collision domain.
Router	Layer 3 device used to create and interconnect network segments or broadcast domains. A router must be configured before traffic can flow through it. Each interface creates a Layer 3 segment and therefore establishes a border for the broadcast and collision domains for all devices on that segment.

Describing Layer 2 Network Issues

This topic describes issues that can occur in a switched network.



Layer 2 switches can significantly improve performance in a carrier sense multiple access collision detect (CSMA/CD) network when used in place of hubs. This is because each switch port represents a single collision domain, and the device connected to that port does not have to compete with other devices to access the media.

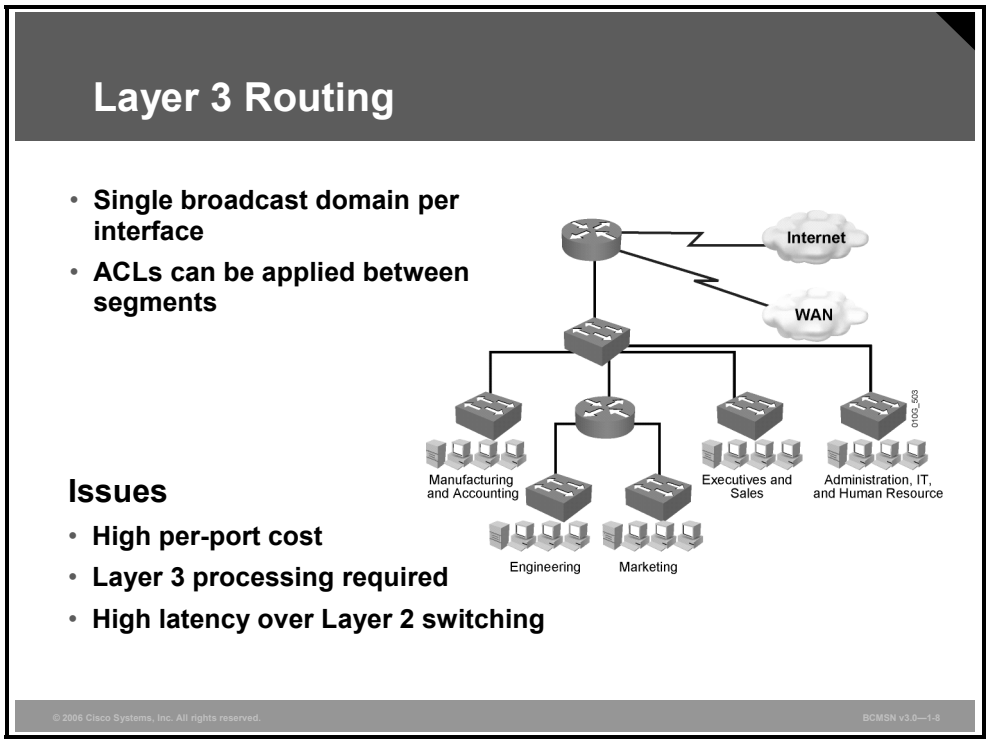
Ideally, every host on a given network segment is connected to its own switch port, thus eliminating all media contention as the switch manages network traffic at Layer 2. An additional benefit of Layer 2 switching is that large broadcast domains can be broken up into smaller segments by assigning switch ports to different VLAN segments.

For all their benefits, some drawbacks still exist in nonhierarchical switched networks.

- If switches are not configured with VLANs, very large broadcast domains may be created.
- If VLANs are created, traffic cannot move between VLANs using only Layer 2 devices.
- As the Layer 2 network grows, the potential for bridge loops increases. Therefore, the use of a Spanning Tree Protocol (STP) becomes imperative.

Describing Routed Network Issues

This topic describes problems that can occur in a Layer 3 network.

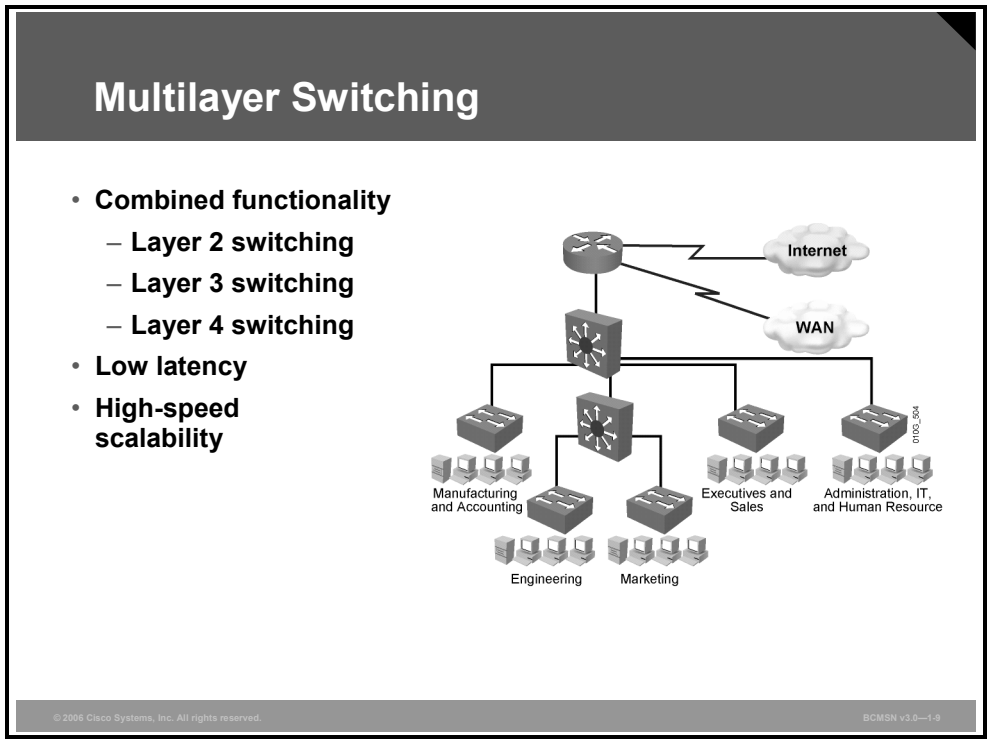


A major limitation of Layer 2 switches is that they cannot switch traffic between Layer 3 network segments (IP subnets for example). Traditionally, this was done using a router. Unlike switches, a router acts as a broadcast boundary and does not forward broadcasts between its interfaces. Additionally, a router provides for an optimal path determination process. The router examines each incoming packet to determine which route the packet should take through the network. Also, the router can act as a security device, manage QoS, and apply network policy. Although routers used in conjunction with Layer 2 switches resolve many issues, some concerns still remain.

- When security or traffic management components, such as access control lists (ACLs), are configured on router interfaces, the network may experience delays as the router processes each packet in software.
- When routers are introduced into a switched network, end-to-end VLANs are no longer supported because routers terminate the VLAN.
- Routers are more expensive per interface than Layer 2 switches, so their placement in the network should be well planned. Nonhierarchical networks by their very nature require more interconnections and, hence, more routed interfaces.
- In a nonhierarchical network, the number of router interconnections may result in peering problems between neighboring routers.
- Because traffic flows are hard to determine, it becomes difficult to predict where hardware upgrades are needed to mitigate traffic bottlenecks.

What Is a Multilayer Switch?

This topic describes multilayer switches in a nonhierarchical network.



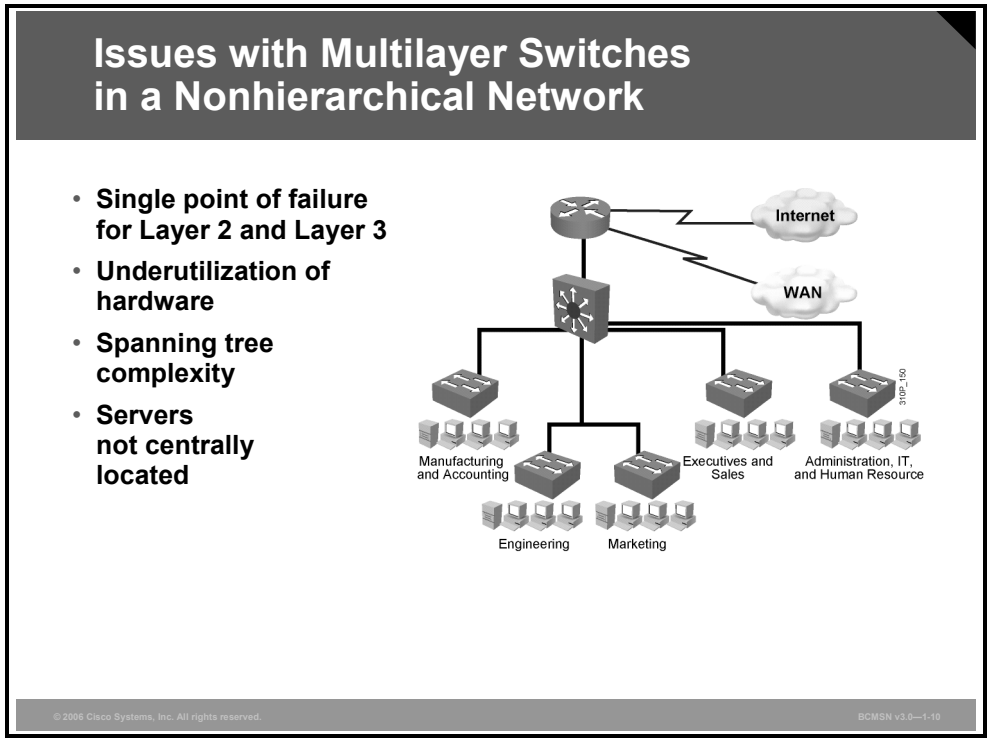
Multilayer switching is hardware-based switching and routing integrated into a single platform. In some cases, the frame and packet forwarding operation is handled by the same specialized hardware ASIC and other specialized circuitry. A multilayer switch does everything to a frame and packet that a traditional switch or router does, including the following:

- Provides multiple simultaneous switching paths
- Segments broadcast and failure domains
- Provides destination-specific frame forwarding based on Layer 2 information
- Determines the forwarding path based on Layer 3 information
- Validates the integrity of the Layer 2 frame and Layer 3 packet via checksums and other methods
- Verifies packet expiration and updates accordingly
- Processes and responds to any option information
- Updates forwarding statistics in the MIB
- Applies security and policy controls, if required
- Provides optimal path determination
- Can (if a sophisticated modular type) support a wide variety of media types and port densities
- Has the ability to support QoS
- Has the ability to support VoIP and inline power requirements

Because it is designed to handle high-performance LAN traffic, a multilayer switch can be placed anywhere within the network, cost-effectively replacing traditional switches and routers. Generally, however, a multilayer switch may be more than is required to provide end systems access to network resources.

Issues with Multilayer Switches and VLANs in a Nonhierarchical Network

This topic describes the issues that occur with multilayer switches and VLANs in a nonhierarchical network.



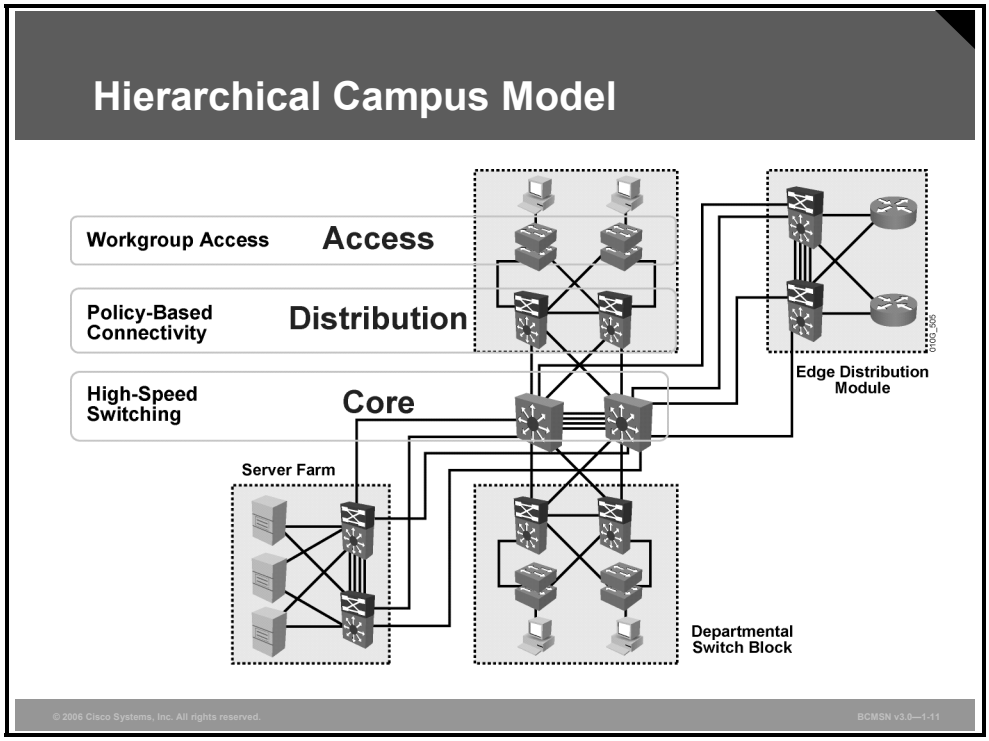
Multilayer switches combine switching and routing on a single hardware platform and can enhance overall network performance when deployed properly. Multilayer switches provide very high-speed Layer 2 and Layer 3 functionality by "caching" much of the forwarding information between sources and destinations.

Here are issues that exist when a multilayer switch is deployed in an improperly designed network.

- Multilayer switches, by condensing the functions of switching and routing in a single chassis, can create single points of failure if redundancy for these devices is not carefully planned and implemented.
- Switches in a flat network are interconnected, creating many paths between destinations. If active, these potential redundant paths will create bridging loops. To control this, the network must run an STP. Networks that use the IEEE 802.1D protocol may experience periods of disconnection and frame flooding during topology change.
- Multilayer switch functionality may be underutilized if a multilayer switch is simply a replacement for the traditional role of a router in a nonhierarchical network.

The Enterprise Composite Network Model

This topic describes the ECNM, which can be used to divide the enterprise network into physical, logical, and functional areas.



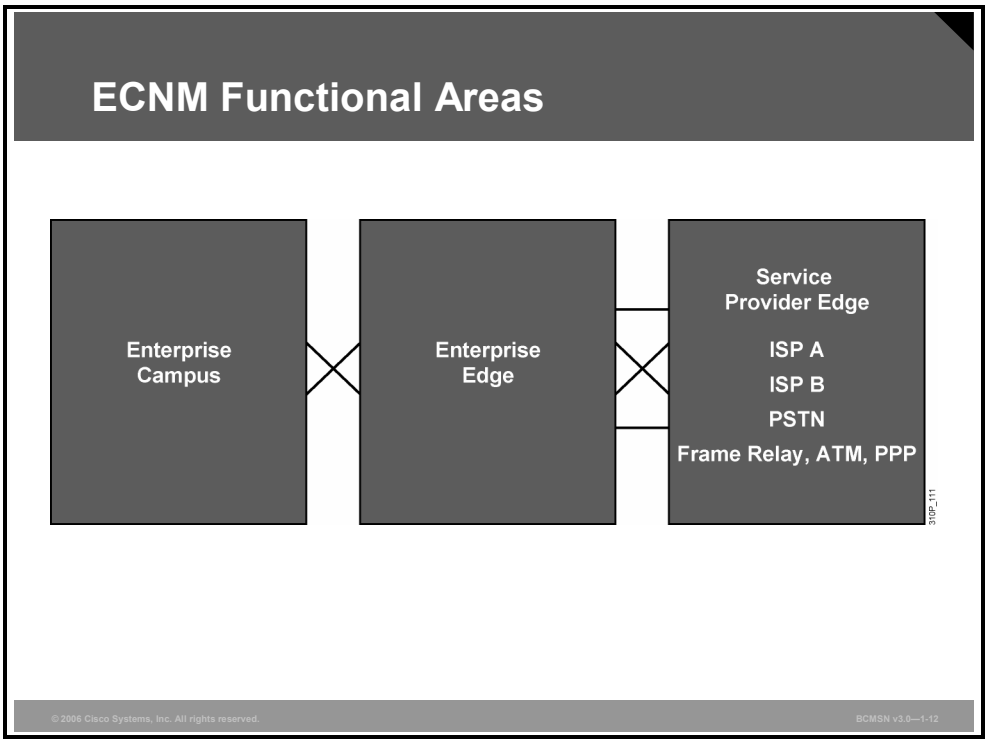
The ECNM provides a modular framework for designing networks. This modularity allows flexibility in network design and facilitates ease of implementation and troubleshooting. The hierarchical model divides networks into the Building Access, Building Distribution, and Building Core layers, as follows:

- **Building Access layer:** The Building Access layer is used to grant user access to network devices. In a network campus, the Building Access layer generally incorporates switched LAN devices with ports that provide connectivity to workstations and servers. In the WAN environment, the Building Access layer at remote sites may provide access to the corporate network across WAN technology.
- **Building Distribution layer:** The Building Distribution layer aggregates the wiring closets and uses switches to segment workgroups and isolate network problems.
- **Building Core layer:** The Building Core layer (also known as the Campus Backbone submodule) is a high-speed backbone and is designed to switch packets as fast as possible. Because the core is critical for connectivity, it must provide a high level of availability and adapt to changes very quickly.

The ECNM divides the enterprise network into physical, logical, and functional areas. These areas allow network designers and engineers to associate specific network functionality on equipment based upon its placement and function in the model.

Enterprise Composite Network Model Functional Areas

This subtopic describes the functional areas of the ECNM.



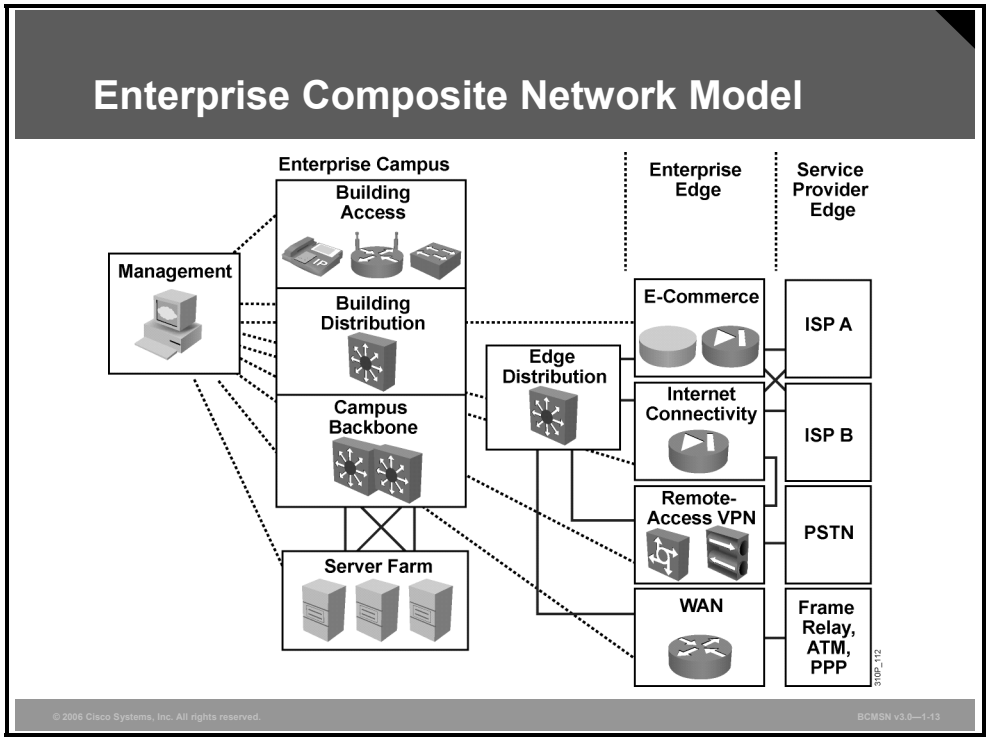
The ECNM introduces modularity by dividing the network into functional areas that ease design, implementation, and troubleshooting tasks. An enterprise campus is defined as one or more buildings, with multiple virtual and physical networks, connected across a high-performance, multilayer-switched backbone.

The ECNM contains these three major functional areas:

- **Enterprise Campus:** The Enterprise Campus functional area contains the modules required to build a hierarchical, highly robust campus network that offers performance, scalability, and availability. This area contains the network elements required for independent operation within a single campus, such as access from all locations to central servers. The Enterprise Campus functional area does not offer remote connections or Internet access.
- **Enterprise Edge:** The Enterprise Edge aggregates connectivity from the various resources external to the enterprise network. As traffic comes into the campus, this area filters traffic from the external resources and routes it into the Enterprise Campus functional area. It contains all of the network elements for efficient and secure communication between the enterprise campus and remote locations, remote users, and the Internet. The Enterprise Edge would replace the Demilitarized Zone (DMZ) of most networks.
- **Service Provider Edge:** This functional area represents connections to resources external to the campus. This area facilitates communication to WAN and Internet service provider technologies.

Benefits of the Enterprise Composite Network Model

This topic describes the benefits of the ECNM.



To scale the hierarchical model, Cisco introduced the ECNM, which further divides the enterprise network into physical, logical, and functional areas. The ECNM contains functional areas, each of which has its own Building Access, Building Distribution, and Building Core (or Campus Backbone) layers.

The ECNM meets these criteria:

- It is a deterministic network with clearly defined boundaries between modules. The model also has clear demarcation points, so that the designer knows exactly where traffic is located.
- It increases network scalability and eases the design task by making each module discrete.
- It provides scalability by allowing enterprises to add modules easily. As network complexity grows, designers can add new functional modules.
- It offers more network integrity in network design, allowing the designer to add services and solutions without changing the underlying network design.

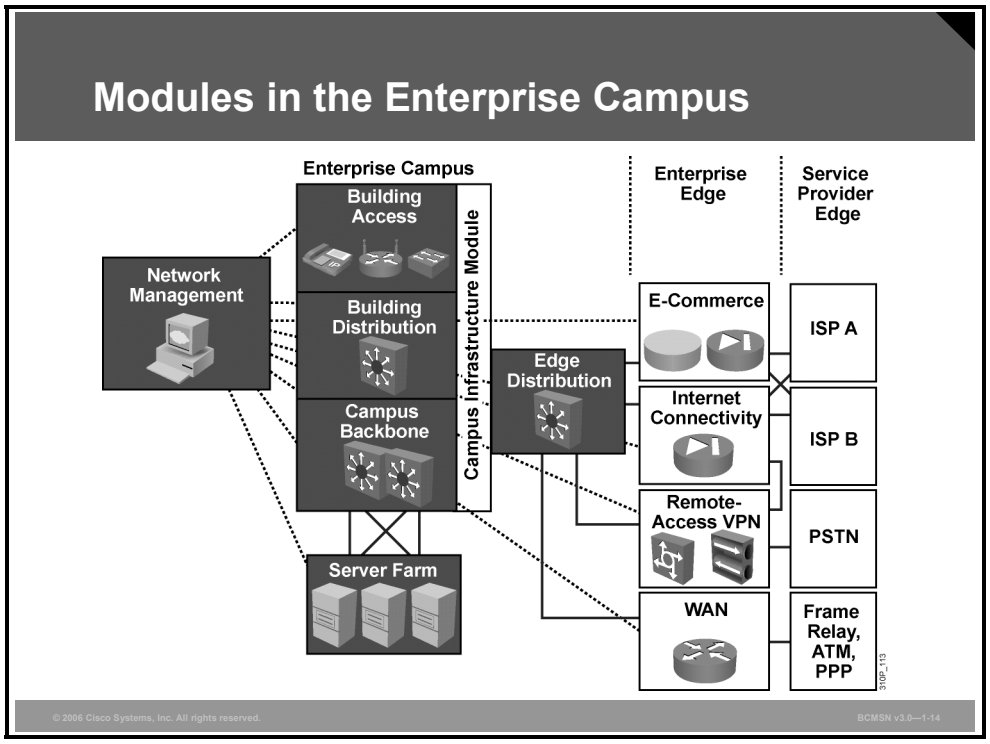
Benefits of ECNM

The table shows the benefits that ECNM offers for each of the submodules where it is implemented.

Submodules	Performance	Scalability	Availability
Building Access	Critical to desktop performance	Provides port density	Important to provide redundancy
Building Distribution	Critical to campus performance	Provides switch modularity	Critical to provide redundancy
Campus Backbone	Critical to overall network performance	Provides switch modularity	Critical to provide redundancy and fault tolerance
Network Management	Monitors performance		Monitors device and network availability
Server Farm	Critical to server performance	Provides switch modularity	Critical to provide redundancy and fault tolerance
Edge Distribution	Critical to WAN and Internet performance	Provides switch modularity	Important to provide redundancy

Describing the Campus Infrastructure Module

This topic describes the Enterprise Campus functional area.

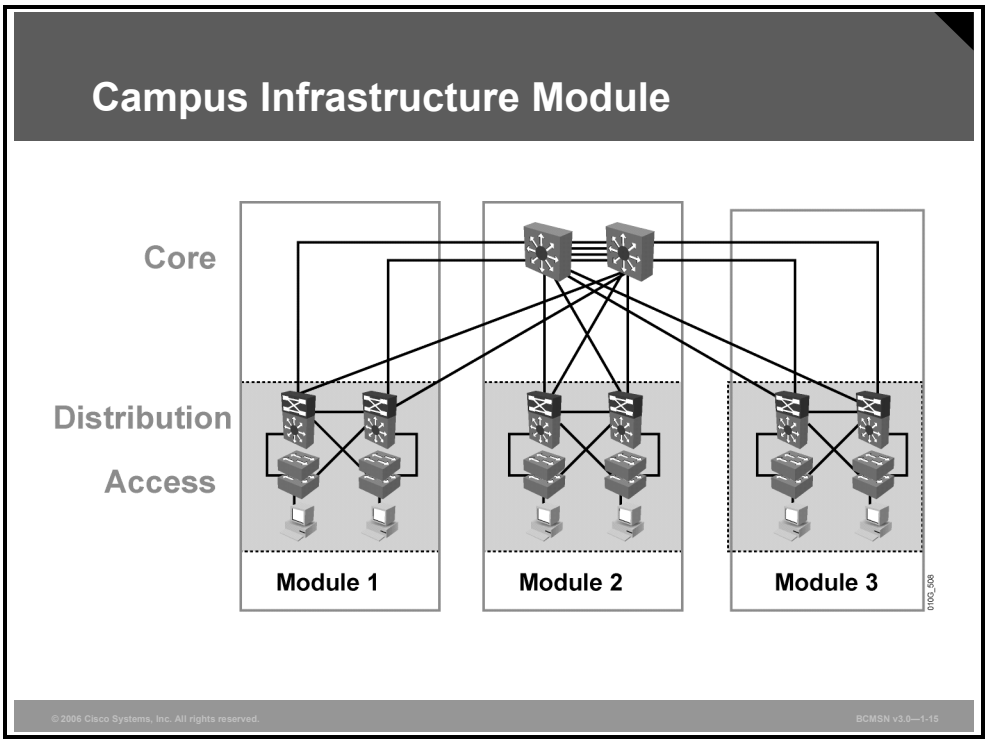


The Enterprise Campus functional area includes the Campus Infrastructure, Network Management, Server Farm, and Edge Distribution modules. Each module has a specific function within the campus network:

- **Campus Infrastructure module:** Includes Building Access and Building Distribution submodules. It connects users within the campus to the Server Farm and Edge Distribution modules. The Campus Infrastructure module is composed of one or more floors or buildings connected to the Campus Backbone submodule.
- **Network Management module:** Performs system logging and authentication as well as network monitoring and general configuration management functions.
- **Server Farm module:** Contains e-mail and corporate servers providing application, file, print, e-mail, and Domain Name System (DNS) services to internal users.
- **Edge Distribution module:** Aggregates the connectivity from the various elements at the Enterprise Edge functional area and routes the traffic into the Campus Backbone submodule.

Campus Infrastructure Module

This topic describes the Campus Infrastructure module of the ECNM.



The Campus Infrastructure module connects users within a campus to the Server Farm and Edge Distribution modules. The Campus Infrastructure module comprises Building Access and Building Distribution switches connected through the Campus Backbone to campus resources.

A Campus Infrastructure module includes these submodules:

- **Building Access submodule (also known as Building Access layer):** Contains end-user workstations, IP phones, and Layer 2 access switches that connect devices to the Building Distribution submodule.

  The Building Access submodule performs services such as support for multiple VLANs, private VLANs, and establishment of trunk links to the Building Distribution layer and IP phones. Each building access switch has connections to redundant switches in the Building Distribution submodule.

- **Building Distribution submodule (also known as Building Distribution layer):** Provides aggregation of building access devices, often using Layer 3 switching. The Building Distribution submodule performs routing, QoS, and access control. Traffic generally flows through the building distribution switches and onto the campus core or backbone.

  This submodule provides fast failure recovery because each building distribution switch maintains two equal-cost paths in the routing table for every Layer 3 network number. Each building distribution switch has connections to redundant switches in the core.

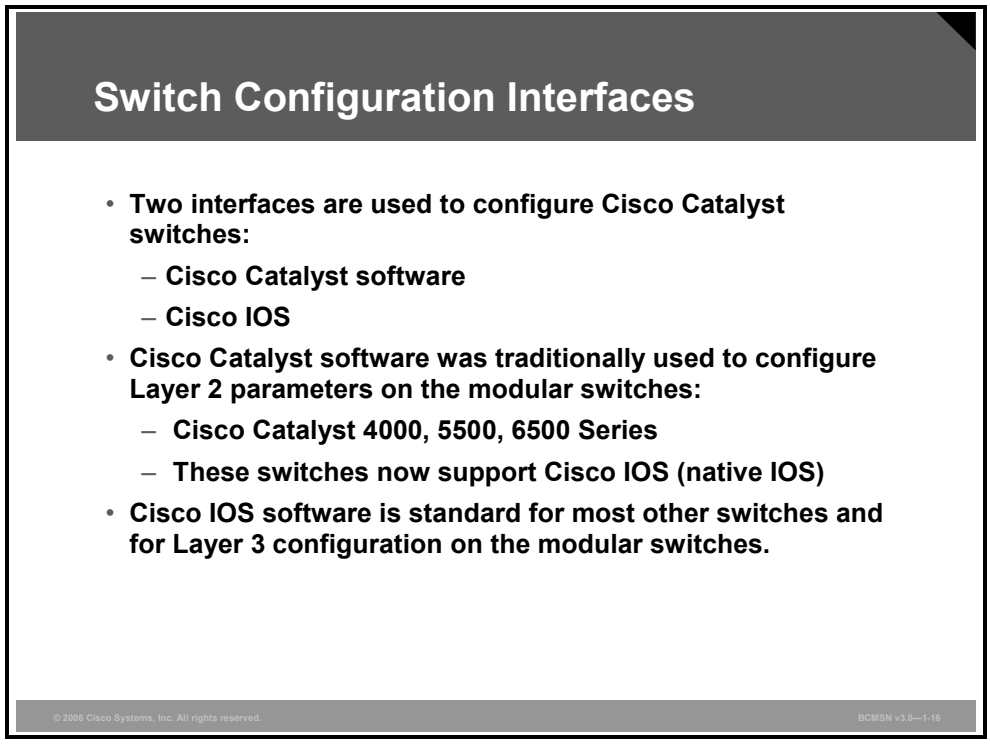
- **Campus Backbone submodule (also known as Building Core layer):** Provides redundant and fast-converging connectivity between buildings and the Server Farm and Edge Distribution modules.

  The purpose of the Campus Backbone submodule is to switch traffic as fast as possible between Campus Infrastructure submodules and destination resources. Forwarding decisions should be made at the ASIC level whenever possible.

  Routing, ACLs, and processor-based forwarding decisions should be avoided at the core and implemented at building distribution devices whenever possible. High-end Layer 2 or Layer 3 switches are used at the core for high throughput, with optimal routing, QoS, and security capabilities available when needed.

Reviewing Switch Configuration Interfaces

This topic identifies the two interfaces used to configure Cisco Catalyst switches.



In the era of the early high-end Cisco Catalyst switches, the Cisco Catalyst operating system (CatOS) and the command interface were significantly different from the Cisco IOS mode navigation interfaces available on all newer Cisco Catalyst platforms. The two interfaces have different features and a different prompt and CLI syntax.

Note	Desktop Express-based switches use a Cisco Network Assist (GUI interface) not a CLI.

Cisco CatOS

This subtopic describes the Cisco Catalyst Operating System (CatOS).

Cisco Catalyst Software

- **Cisco Catalyst software is used to configure Layer 2 parameters.**
- **Cisco Catalyst software configuration commands are prefaced with the keyword set.**
  - **Console(enable) set port enable 3/5**
- **Layer 3 configuration is implemented on MSFC with the Cisco IOS interface.**
- **Some platforms can now use the Cisco IOS interface to configure both Layer 2 and Layer 3 (native IOS).**

Cisco Catalyst 4000, 5500, and 6500 switches

Cisco Catalyst Software Interface

The original Cisco Catalyst interface is sometimes referred to as the "set-based" or, more recently, "Catalyst software" command-line interface (CLI).

In the Cisco Catalyst software, commands are executed at the switch prompt, which can be either nonprivileged (where a limited subset of user-level commands is available) or at a password-protected privileged mode (where all commands are available). Configuration commands are prefaced with the keyword **set**.

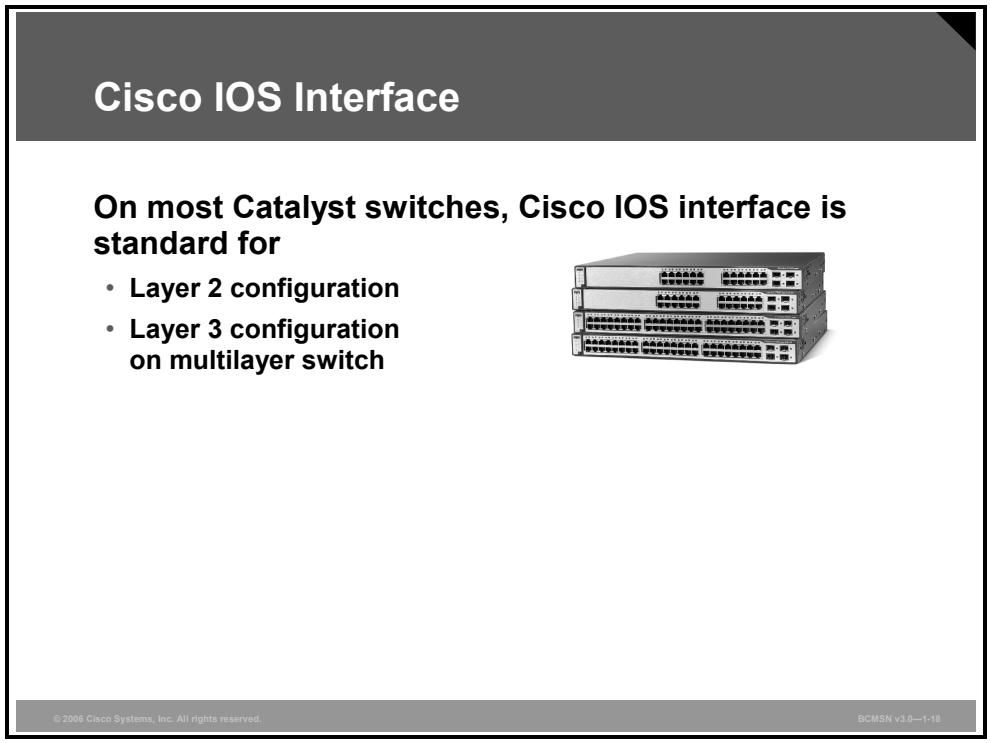
Example: Using Cisco Catalyst Software Commands

In the example, the Cisco Catalyst software commands execute the following: first, show the status of a port; second, move to enable mode that requires a password; third, enable the port.

```
Console> show port 3/5

.

.
Console> enable

Enter password:
Console(enable) set port enable 3/5
```

Cisco IOS Interface

This subtopic describes the Cisco IOS interface that is used on most Cisco Catalyst switches.



Cisco Catalyst switch platforms have had a number of different operating systems and user interfaces. Over the years, Cisco has made great strides in converting the interface on nearly every Cisco Catalyst platform to the Cisco IOS interface familiar to Cisco users on routing platforms. Unlike the Cisco Catalyst software, various modes are navigated to execute specific commands.

Here is an example of how switch port 3 might be enabled on an access layer switch using the Cisco IOS interface and how its status is verified after configuration. Compare how the Cisco IOS interface is navigated here to the previous example, showing how the same function is performed in the Cisco Catalyst software.

Example: Using Cisco IOS Commands

```
Switch# config terminal
Switch(config)# interface fastethernet 0/3
Switch(config-if)# no shut
Switch(config-if)# end
Switch# show interface fastethernet 0/3
```

Configuration Interface Available on Various Cisco Catalyst Platforms

Some widely used Cisco Catalyst switch platforms that support the Cisco IOS interface are 2950, 3500, 3700, 4500*, 6500*, and 8500.

* These platforms have an option to use either Cisco IOS or Cisco Catalyst software for Layer 2 configuration.

The Catalyst software interface exists on several modular Cisco Catalyst platforms, including the Cisco Catalyst 4000, 4500, 5500, 6000, and 6500 Series.

For example, on the Cisco Catalyst 6500, you have the option of using the Cisco Catalyst software, Cisco Catalyst software plus Cisco IOS software, or Cisco IOS software functionality.

Cisco Catalyst 6500 Interfaces

Operating System	Where Installed	Purpose
Cisco Catalyst software	On Cisco Catalyst switch supervisor module.	Cisco Catalyst software interface provided to configure Layer 2 switch functions. Suitable if unit is used in a Layer 2 environment only.
Cisco Catalyst software + Cisco IOS software	If switch contains routing capability, where the supervisors run Cisco Catalyst software, and the Multilayer Switch Feature Card (MSFC) or Route Switch Module (RSM) runs Cisco IOS software.	This allows the Layer 2 switch functionality to be separate from the Layer 3 (and above) Cisco IOS functionality.
Native Cisco IOS	A single instance of Cisco IOS software is installed on the Cisco Catalyst Supervisor Engine, which also controls MSFC.	A single Cisco IOS kernel provides all Multilayer Switching functions (Layers 2 and above).

The Cisco IOS interface is used across a wide variety of Cisco Catalyst switch platforms, particularly the fixed and stackable switches, and is therefore the interface assumed through the remainder of this courseware. Labs may provide direction on the use of specific Cisco Catalyst software commands, depending on the equipment provided.

Summary

This topic summarizes the key points discussed in this lesson.

Summary

- The SONA framework guides the evolution of the enterprise network toward IIN.
- Cisco enterprise architecture with a hierarchical network model facilitates the deployment of converged networks.
- Nonhierarchical network designs do not scale and do not provide the required security necessary in a modern topology.
- Layer 2 networks do not provide adequate security or hierarchical networking.
- Router-based networks provide greater security and hierarchical networking; however, they can introduce latency issues.

Summary (Cont.)

- Multilayer switches combine both Layer 2 and Layer 3 functionality to support the modern campus network topology.
- Multilayer switches can be used in nonhierarchical networks; however, they will not perform at the optimal level.
- The enterprise composite model identifies the key components and logical design for a modern topology.
- Implementation of an ECNM provides a secure, robust network with high availability.
- The Campus infrastructure, as part of an ECNM, provides additional security and high availability at all levels of the campus.
- The two Cisco Catalyst switch interfaces have different features and different font.

Module Self-Check

Use the question here to review what you learned in this module. The correct answer is found in the Module Self-Check Answer Key.

Q1) Which attribute does *not* apply to multilayer switches? (Source: Introducing Campus Networks)

A) combine Layer 2, 3, and 4 switching
B) provide low latency
C) combine Layer 1, Layer 2, and Layer 3 switching
D) provide high-speed scalability

Module Self-Check Answer Key

Q1) C

Module 2

Defining VLANs

Overview

This module defines the purpose of VLANs and describes how VLAN implementation can simplify network management and troubleshooting and can improve network performance. When VLANs are created, their names and descriptions are stored in a VLAN database that can be shared between switches. You will see how design considerations determine which VLANs will span all the switches in a network and which VLANs will remain local to a switch block. The configuration components of this module will describe how individual switch ports may carry traffic for one or more VLANs, depending on their configuration as access or trunk ports. This module explains both why and how VLAN implementation occurs in an enterprise network.

Module Objectives

Upon completing this module, you will be able to define VLANs to segment network traffic and manage network utilization. This ability includes being able to meet these objectives:

- Identify how various technologies are best implemented within the Campus Infrastructure module
- Configure VLANs on access switches to confine traffic to individual VLANs in accordance with the Campus Infrastructure module design
- Explain the procedures for configuring both 802.1Q and ISL trunking between two switches so that VLANs that span the switches can connect
- Describe how VLAN configuration of switches in a single management domain can be automated with the Cisco proprietary VTP
- Identify common VLAN configuration errors and explain the solutions to those errors

Lesson 1

Implementing Best Practices for VLAN Topologies

Overview

This lesson addresses the business and technology needs of an organization and addresses how those needs can be met by applying the appropriate resources to the Campus Infrastructure module.

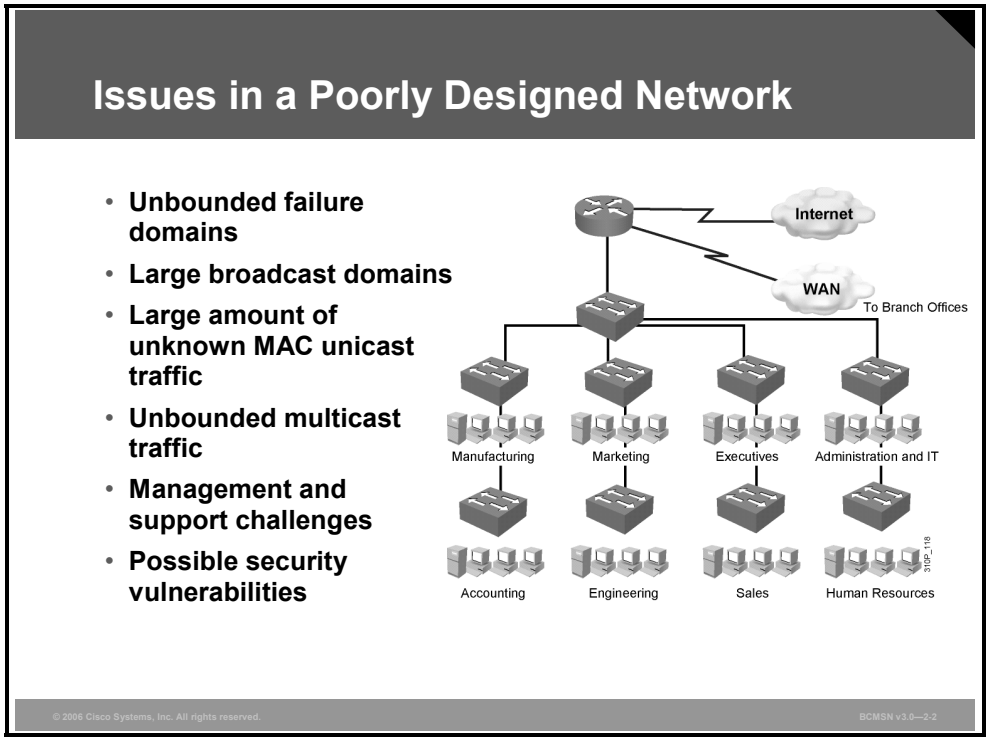
Objectives

Upon completing this lesson, you will be able to identify how various technologies are best implemented within the Campus Infrastructure module. This ability includes being able to meet these objectives:

- List the issues that can occur in a poorly designed network
- Given a sample organization, explain how to designate VLANs for the organization
- Describe the different network interconnection technologies and identify their appropriate usage in a campus network
- Determine the equipment and cabling needs on the various links of VLANs in a campus network
- Map a hierarchical IP addressing scheme to the VLANs in a campus network
- Identify the most common traffic sources and their destination on a campus network

Describing Issues in a Poorly Designed Network

This topic describes the issues that can occur in a poorly designed network.



A poorly designed network has increased support costs, reduced service availability, and limited support for new applications and solutions. Less than optimal performance will affect end users directly and will affect access to central resources. Here are some of the issues that stem from a poorly designed network.

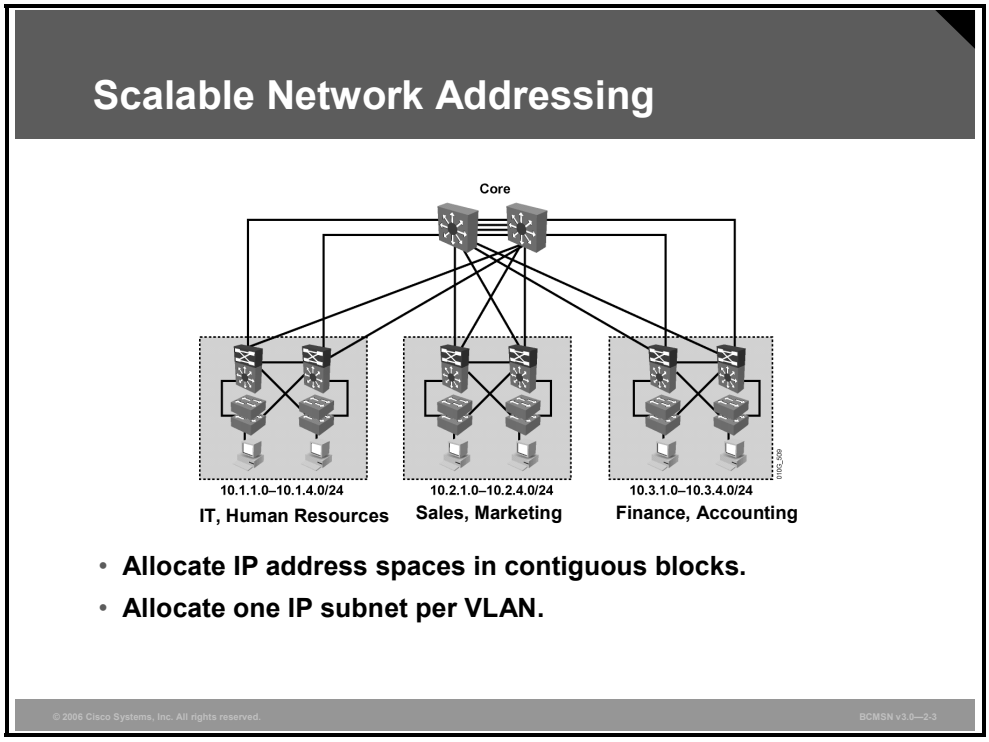
- **Failure domains:** One of the most important reasons to implement an effective design is to minimize the extent of a network problem when it occurs. When Layer 2 and Layer 3 boundaries are not clearly defined, failure in one network area can have a far-reaching effect.
- **Broadcast domains:** Broadcasts exist in every network. Many applications and many network operations require broadcasts to function properly; therefore, it is not possible to completely eliminate them. Just as with failure domains, to minimize the negative impact of broadcasts, broadcast domains should have clear boundaries and include an optimal number of devices.
- **Large amount of unknown MAC unicast traffic:** Cisco Catalyst switches limit unicast frame forwarding to ports associated with the specific unicast address. However, frames that arrive for a destination MAC address that is not recorded in the MAC table are flooded out all switch ports. This is known as an "unknown MAC unicast flooding." Because this causes excessive traffic on switch ports, network interface cards (NICs) have to attend to a larger number of frames on the wire, and security can be compromised as data is being propagated on a wire for which it was not intended.

- **Multicast traffic on ports where not intended:** IP multicast is a technique that allows IP traffic to be propagated from one source to a multicast group that is identified by a single IP and MAC destination group address pair. Similar to unicast flooding and broadcasting, multicast frames will be flooded out all switch ports. A proper design allows for containment of multicast frames while allowing them to be functional.
- **Difficulty in management and support:** A poorly designed network may be disorganized, poorly documented, and lacking easily identified traffic flows, which can cause support, maintenance, and problem resolution to become time-consuming and arduous tasks.
- **Possible security vulnerabilities:** A poorly designed switched network with little attention to security requirements at the access layer can compromise the integrity of the entire network.

A poorly designed network always has a negative impact and becomes a burden for any organization in terms of support and related costs.

Grouping Business Functions into VLANs

This topic describes a best practice for designating VLANs for an organization.



Hierarchical network addressing means that IP network numbers are applied to the network segments or VLANs in an orderly fashion that takes into consideration the network as a whole. Blocks of contiguous network addresses are reserved for, and configured on, devices in a specific area of the network.

Here are some benefits of hierarchical addressing.

- **Ease of management and troubleshooting:** Hierarchical addressing groups network addresses contiguously. Network management and troubleshooting are more efficient because a hierarchical IP addressing scheme will make problem components easier to locate.
- **Minimizing of error:** Orderly network address assignment can minimize error and duplicate address assignment.
- **Reduced number of routing table entries:** In a hierarchical addressing plan, routing protocols are able to perform route summarization, which allows a single routing table entry to represent a collection of IP network numbers. Route summarization makes routing table entries more manageable and provides these benefits:
  - Reduced number of CPU cycles when recalculating a routing table or sorting through the routing table entries to find a match
  - Reduced router memory requirements
  - Faster convergence after a change in the network
  - Easier troubleshooting

Guidelines for Applying IP Address Space in the Enterprise Network

The Enterprise Composite Network Model (ECNM) provides a modular framework for designing and deploying networks. It also provides the ideal structure for overlaying a hierarchical IP addressing scheme. Here are some guidelines to follow.

- Design the IP addressing scheme so that blocks of 4, 8, 16, 32, or 64 contiguous network numbers can be assigned to the subnets in a given building distribution and access switch block. This approach allows each switch block to be summarized into one large address block.
- At the Building Distribution layer, continue to assign network numbers contiguously out toward to the access layer devices.
- Have a single IP subnet correspond with a single VLAN. Each VLAN is a separate broadcast domain.
- Subnet at the same binary value on all network numbers, avoiding variable length subnet masks when possible, to minimize error and confusion when troubleshooting or configuring new devices and segments.

Example: Network Design

A business with approximately 250 employees wants move to the ECNM.

Users per Department

The table shows the number of users in each department.

Department	Number of Users	Location
IT	45	Building A
Human Resources	10	Building A
Sales	102	Building B
Marketing	29	Building B
Finance	18	Building C
Accounting	26	Building C

Six VLANs are required to accommodate one VLAN per user community; therefore, in following the guidelines of the ECNM, six IP subnets are required.

The business has decided to use network 10.0.0.0 as its base address.

The Sales Department is the largest department, which requires a minimum of 102 addresses for its users. Therefore, a subnet mask of 255.255.255.0 (/24) is chosen, giving a maximum number of 254 hosts per subnet.

It has been decided, for future growth, to have one switch block per building as follows:

- Building A is allocated 10.1.0.0/16
- Building B is allocated 10.2.0.0/16
- Building C is allocated 10.3.0.0/16

Building A VLANs and IP Subnets

The table shows the allocation of VLANs and IP subnets within building A.

Department	VLAN	IP Subnet Address
IT	VLAN 11	10.1.1.0/24
Human Resources	VLAN 12	10.1.2.0/24
Unused	-	10.1.3.0 - 10.1.254.0

Building B VLANs and IP Subnets

The table shows the allocation of VLANs and IP subnets within building B.

Department	VLAN	IP Subnet Address
Sales	VLAN 21	10.2.1.0/24
Marketing	VLAN 22	10.2.2.0/24
Unused	-	10.2.3.0 - 10.2.254.0

Building C VLANs and IP Subnets

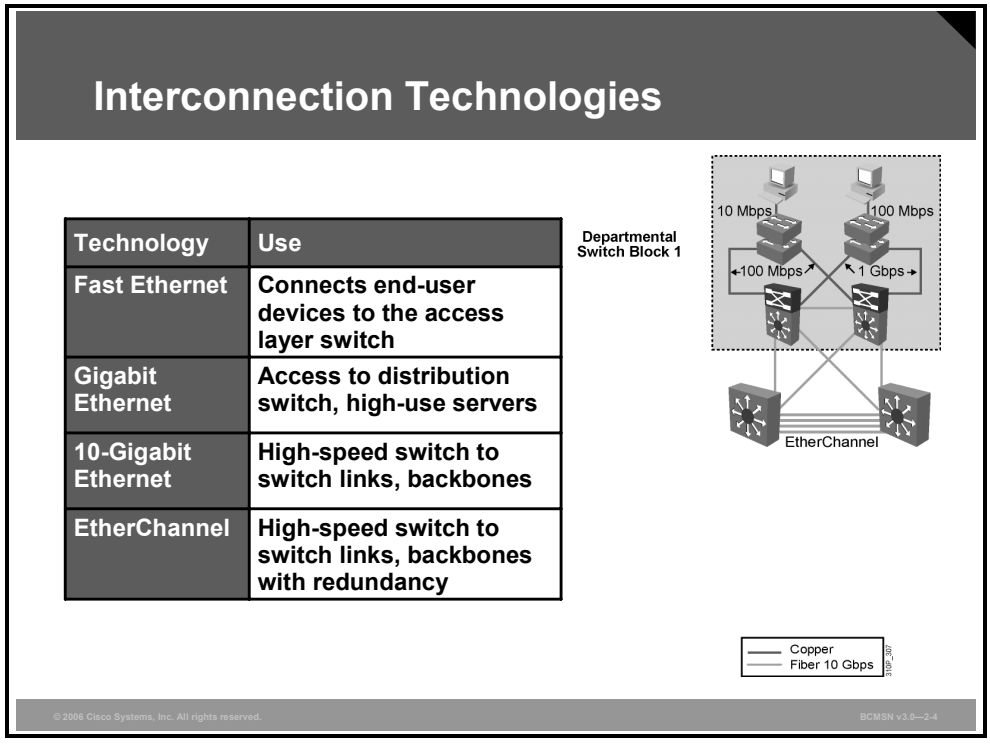
The table shows the allocation of VLANs and IP subnets within building C.

Department	VLAN	IP Subnet Address
Finance	VLAN 31	10.3.1.0/24
Accounting	VLAN 32	10.3.2.0/24
Unused	-	10.3.3.0 - 10.3.254.0

Some of the currently unused VLANs and IP subnets would be used to manage the network devices. If the company decides to implement IP telephony, then some of the unused VLANs and IP subnets would be allocated to the voice VLANs.

Describing Interconnection Technologies

This topic describes the different network interconnection technologies and identifies their appropriate usage in a campus network.



A number of technologies are available to interconnect devices in the campus network. Some of the more common technologies are listed here. The interconnection technology selected will depend on the amount of traffic the link must carry. A mixture of copper and fiber-optic cabling will likely be used, based on distances, noise immunity requirements, security, and other business requirements.

- **Fast Ethernet (100-Mbps Ethernet):** This LAN specification (IEEE 802.3u) operates at 100 Mbps over twisted-pair cable. The Fast Ethernet standard raises the speed of Ethernet from 10 Mbps to 100 Mbps with only minimal changes to the existing cable structure. A switch with port functioning at both 10 and 100 Mbps can move frames between ports without Layer 2 protocol translation.
- **Gigabit Ethernet:** An extension of the IEEE 802.3 Ethernet standard, Gigabit Ethernet increases speed tenfold over that of Fast Ethernet, to 1000 Mbps, or 1 Gbps. IEEE 802.3z specifies operations over fiber optics, and IEEE 802.3ab specifies operations over twisted-pair cable.
- **10-Gigabit Ethernet:** 10-Gigabit Ethernet was formally ratified as an IEEE 802.3 Ethernet standard in June 2002. This technology is the next step for scaling the performance and functionality of an enterprise. With the deployment of Gigabit Ethernet becoming more common, 10-Gigabit will become the norm for uplinks.

- **EtherChannel:** This feature provides link aggregation of bandwidth over Layer 2 links between two switches. EtherChannel bundles individual Ethernet ports into a single logical port or link, providing aggregate bandwidth of up to 1600 Mbps (8-100Mbps links, full duplex) or up to 16 Gbps (8-Gigabit links, full duplex) between two Cisco Catalyst switches. All interfaces in each EtherChannel bundle must be configured with similar speed, duplex, and VLAN membership.

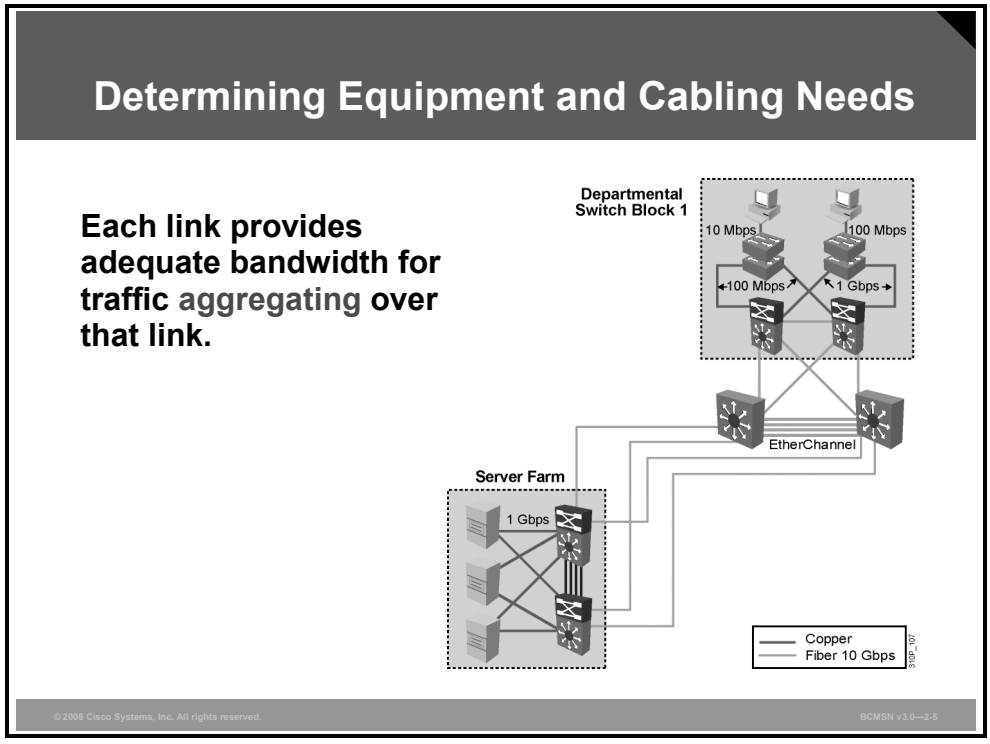
Interconnection Technologies

The table discusses the use of each technology in the Campus Infrastructure module.

Technology	Use in Campus Infrastructure Module
Fast Ethernet	Often used to connect end-user devices to the access layer switch. If user connections are running at 10 Mbps, Fast Ethernet links between access and distribution switches may be adequate. It is adequate for servers in small- to medium-sized networks if full duplex is invoked.
Gigabit Ethernet	High-speed LAN backbones connecting building distribution switches to campus backbone switches. Widely used internal or Internet-accessible servers might be connected via gigabit Layer 2 technology.
10-Gigabit Ethernet	Very high-speed LAN backbone and link aggregation. As gigabit links become more common, 10-Gigabit will be necessary to scale the uplinks.
EtherChannel	Any connection between switches with multiple physical links that requires high bandwidth and redundancy. Links between Building Distribution and Campus Backbone might be Gigabit EtherChannel. Links between access and distribution layer devices might be Fast Ethernet EtherChannel.

Determining Equipment and Cabling Needs

This topic describes how to determine equipment and cabling requirements on various links of VLANs in a campus network.



There are four objectives in the design of any high-performance network: security, availability, scalability, and manageability. The ECNM, when implemented properly, provides the framework to meet these objectives. In the migration from a current network infrastructure to the ECNM, a number of infrastructure changes may be needed, including the replacement of current equipment and the existing cable plant.

This list describes the equipment and cabling decisions that should be considered when altering infrastructure.

- Replace hubs and legacy switches with new switches at the Building Access layer. Select equipment with the appropriate port density at the access layer to support the current user base while preparing for growth. Some designers begin by planning for about 30 percent growth. If the budget allows, use modular access switches to accommodate future expansion. Consider planning for support of inline power and quality of service (QoS) if IP telephony may be implemented in the future.
- When building the cable plant from the Building Access layer to the Building Distribution layer devices, remember that these links will carry aggregate traffic from the end nodes at the access layer to the Building Distribution switches. Ensure that these links have adequate bandwidth capability. EtherChannel bundles can be used here to add bandwidth as necessary.

- At the Building Distribution layer, select switches with adequate performance to handle the load of the current Building Access layer. Also plan some port density for adding trunks later to support new access layer devices. The devices at this layer should be multilayer (Layer 2/Layer 3) switches that support routing between the workgroup VLANs and network resources. Depending on the size of the network, the building distribution layer devices may be fixed chassis or modular. Plan for redundancy in the chassis and in the connections to the access and core layers, as the business objectives dictate.
- The Campus Backbone equipment must support high-speed data communications between other submodules. Be sure to size the backbone for scalability and plan on redundancy.

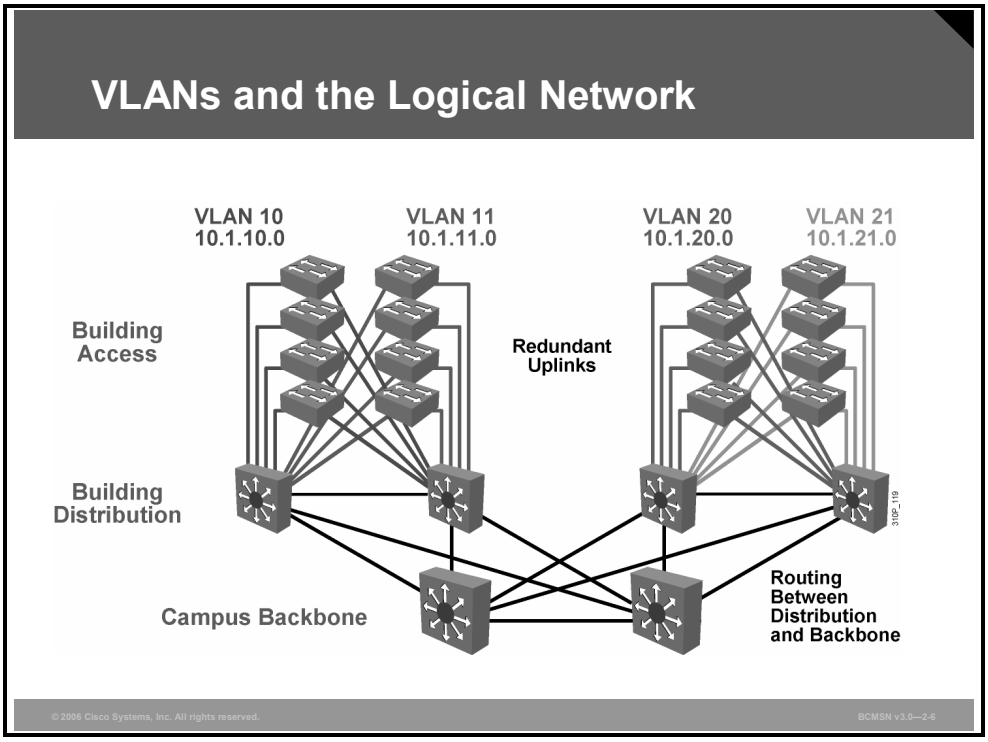
Cisco Systems has online tools to assist designers in making the proper selection of devices and uplink ports based on business and technology needs. Cisco suggests oversubscription ratios that can be used to plan bandwidth requirements between key devices on a network with average traffic flows.

- **Access to distribution layer links:** The oversubscription ratio should be no higher than 20:1. That is, the link can be 1/20 of the total bandwidth available cumulatively to all end devices using that link.
- **Distribution to core links:** The oversubscription ratio should be no higher than 4:1.
- **Between core devices:** There should be little to no oversubscription planning. That is, the links between core devices should be able to carry traffic at the speed represented by the aggregate number bandwidth of all the distribution uplinks into the core.

Caution These ratios are appropriate for estimating average traffic from access layer, end-user devices. They are not accurate for planning oversubscription from the Server Farm or Edge Distribution module. They are also not accurate for planning bandwidth needed on access switches hosting typical user applications with high bandwidth consumption (for example, nonclient server databases or multimedia flows to unicast addresses. Using QoS end to end prioritizes the traffic that would need to be dropped in the event of congestion.

Mapping VLANs in a Hierarchical Network

This topic describes the procedure for mapping subnets to the VLANs in a campus network.

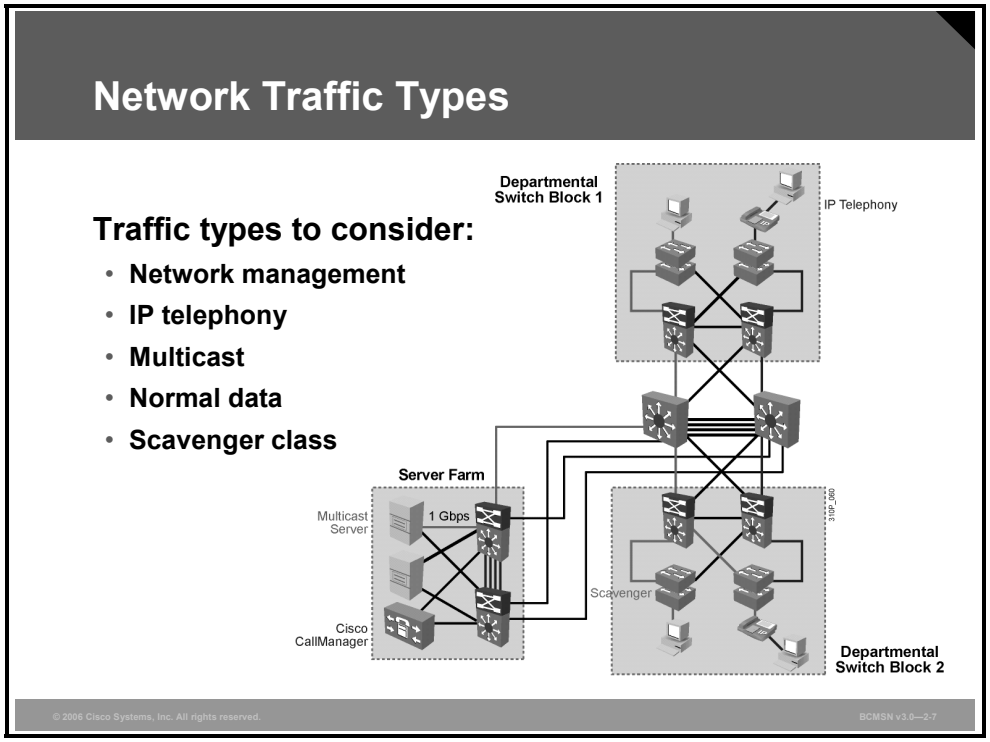


When mapping VLANs onto the new hierarchical network design, keep these parameters in mind.

- Examine the subnetting scheme that has been applied to the network and associate a VLAN to each subnet.
- Configure routing between VLANs at the distribution layer using multilayer switches.
- Make end-user VLANs and subnets local to a specific switch block.
- Ideally, limit a VLAN to one access switch or switch stack. However, it may be necessary to extend a VLAN across multiple access switches within a switch block to support a capability such as wireless mobility.

Considering Traffic Source to Destination Paths

This topic identifies the most common traffic sources and their destinations on a campus network.



This table lists different types of traffic that may exist on the network and that should be considered before device placement and VLAN configuration.

Traffic Types

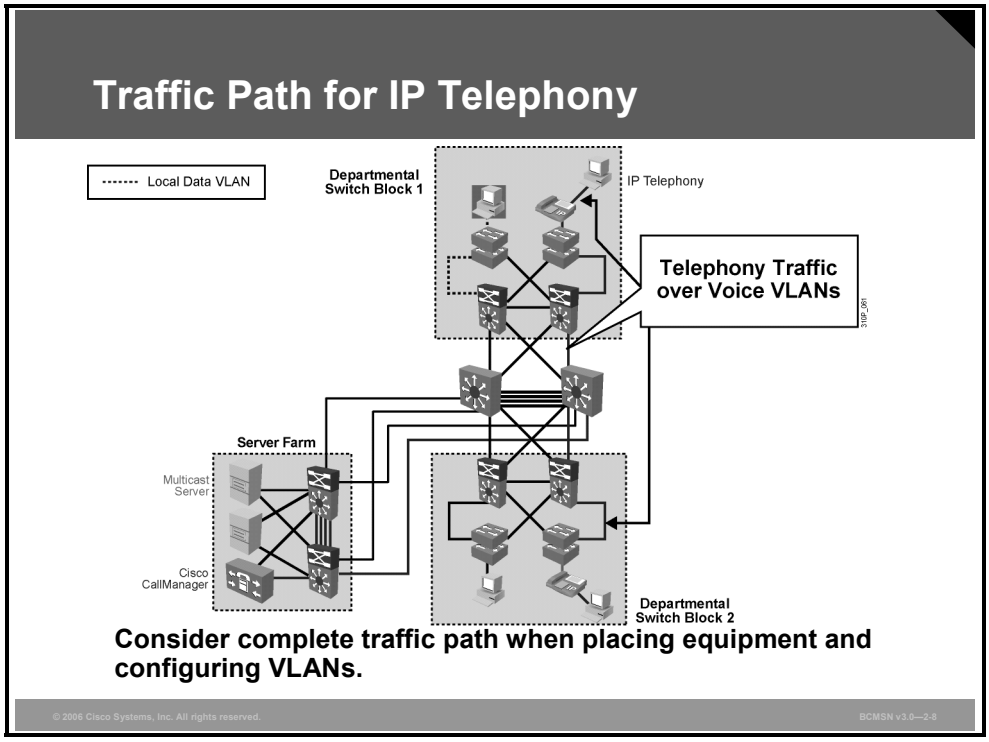
The table describes the different traffic types.

Traffic Type	Description
Network management	Many different types of network management traffic may be present on the network. Examples include bridge protocol data units (BPDUs), Cisco Discovery Protocol (CDP) updates, Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON) traffic. Some designers will assign a separate VLAN to the task of carrying certain types of network management traffic to make network troubleshooting easier.
IP telephony	There are two types of IP telephony traffic: signaling information between the end devices (for example, IP phones, and softswitches, such as Cisco CallManager) and the data packets of the voice conversation itself. Often, the data to and from IP phones will be configured on a separate VLAN for voice traffic because the designer will want to apply QoS measures to give high priority to voice traffic.

Traffic Type	Description
IP multicast	IP multicast traffic is sent from a particular source address to nonunique MAC addresses. Examples of applications that generate this type of traffic are IP/TV broadcasts and imaging software used to configure workstations and servers quickly. Multicast traffic can produce a large amount of data streaming across the network. Switches need to be configured to keep this traffic from flooding to devices that have not requested it, and routers need to ensure that multicast traffic is forwarded to the network areas where it is requested.
Normal data	This is typical application traffic related to file and print services, e-mail, Internet browsing, database access, and other shared network applications. This data may have to be treated in the same or different ways in different parts of the network, based on the volume of each type. Examples of this type of traffic are Server Message Block, Netware Core Protocol (NCP), Simple Mail Transfer Protocol (SMTP), Structured Query Language (SQL), and HTTP.
Scavenger class	Scavenger class includes all traffic with protocols or patterns that exceed their normal data flows. It is used to protect the network from exceptional traffic flows that may be the result of malicious programs executing on end-system PCs. Scavenger class is also used for less than best-effort type traffic, such as peer-to-peer traffic.

Considering IP Telephony

This subtopic describes the IP telephony components and traffic flow in a campus network.



The size of an enterprise network drives the design and placement of certain types of devices. If the network is designed according to the ECNM, there will be distinct devices separating the access, distribution, and backbone areas of the network. The network design and the types of applications supported will determine where certain traffic sources are located. In the case of multicast and IP telephony applications, they do share some common traffic types. Specifically, if a Cisco CallManager is providing music on hold, it may need to multicast that traffic stream.

Consider these points when determining where to place the servers:

- Cisco CallManager servers must be accessible throughout the network at all times. Ensure that there are redundant NICs in the publisher and subscriber servers and redundant connections between those NICs and the upstream switch from the server. It is recommended that voice traffic be configured on its own VLAN. Cisco CallManager servers are typically located within the Server Farm block in the ECNM design.
- VLAN trunks must be configured appropriately to carry IP telephony traffic throughout the network or to specific destinations.

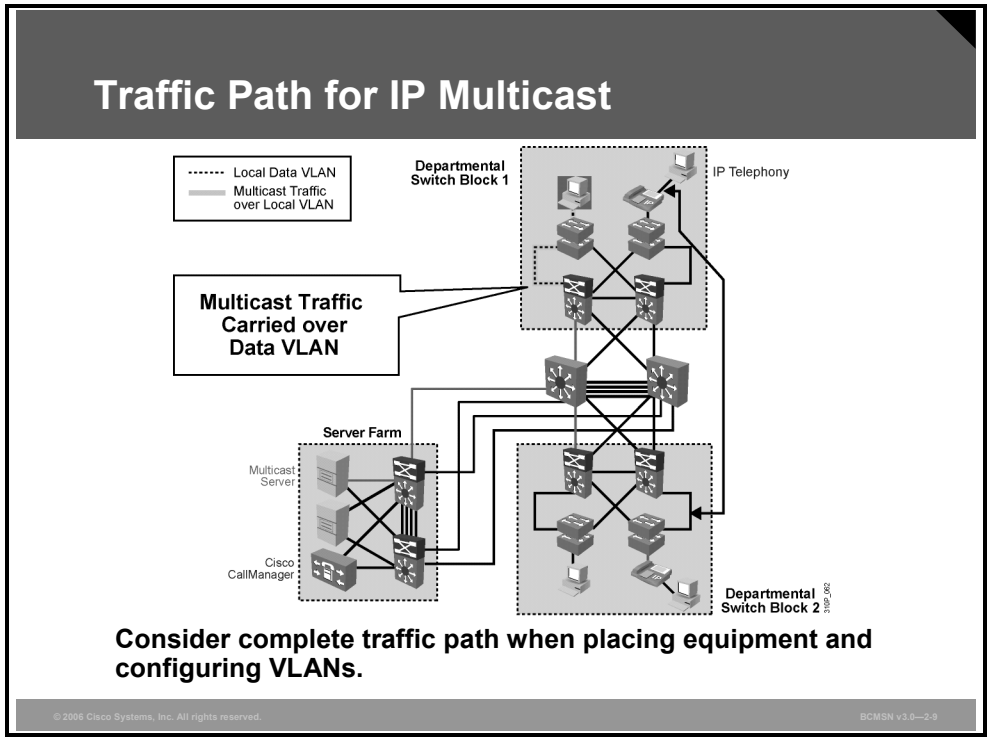
When you deploy voice, Cisco recommends that you enable two VLANs at the access layer: a native VLAN for data traffic and a voice VLAN.

Separate voice and data VLANs are recommended for these reasons:

- Address space conservation and voice device protection from external networks
- QoS trust boundary extension to voice devices
- Protection from malicious network attacks
- Ease of management and configuration

Considering IP Multicast Traffic

This subtopic describes the IP multicast components and traffic flow in a campus network.

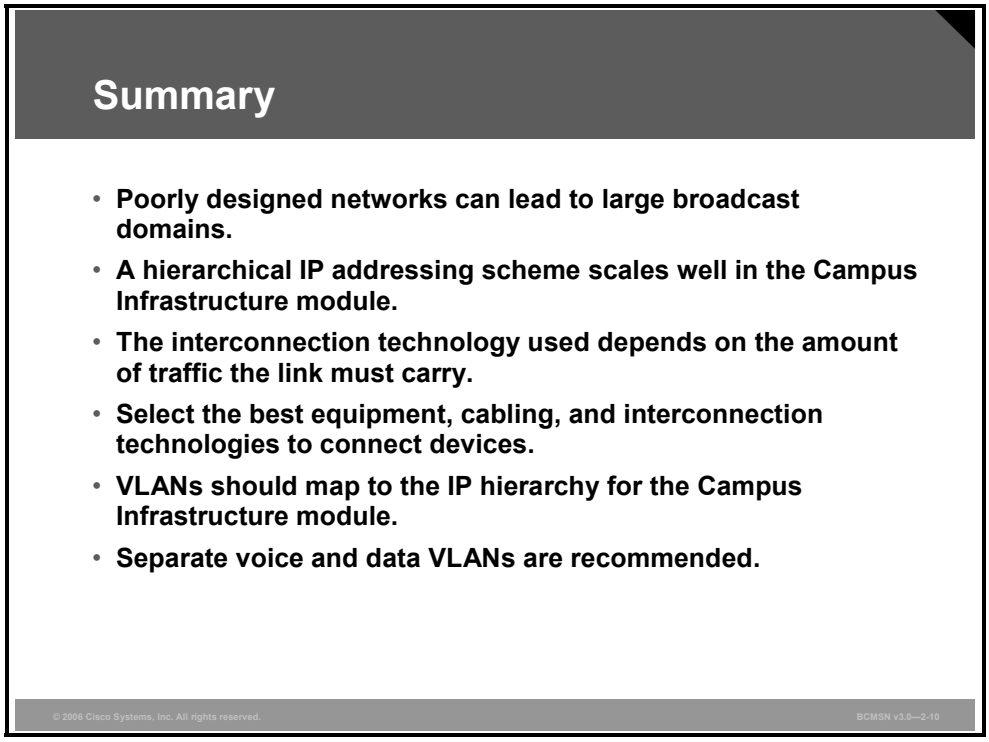


The multilayer campus design is ideal for control and distribution of IP multicast traffic. The Layer 3 multicast control is provided by Protocol Independent Multicast (PIM) routing protocol. Multicast control at the wiring closet is provided by Internet Group Management Protocol (IGMP) snooping or Cisco Group Management Protocol (CGMP). Multicast control is extremely important because of the large amount of traffic involved when several high-bandwidth multicast streams are provided. Consider these guidelines when designing the network for multicast traffic:

- IP multicast servers may exist within a server farm or be distributed throughout the network at appropriate locations.
- Select distribution layer switches to act as PIM rendezvous points (RPs), and place them where they are central to the location of the largest distribution of receiving nodes. RPs are typically used to temporarily connect multicast source and receivers. For further study in PIM, attend the *Building Scalable Cisco Internetworks* course.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 2

Implementing VLANs

Overview

VLANs are used to create logical broadcast domains and Layer 3 segments in a given network. A VLAN is considered a logical segment because the traffic it carries may traverse multiple physical network segments. This lesson will examine how switch ports can be statically configured to belong to one or more VLANs and how various ports on a single switch can belong to different VLANs. End-to-end VLANs will be differentiated from local VLANs. Local VLANs exist within the context of a single switch or switch block, whereas end-to-end VLANs span multiple network segments interconnected by switches.

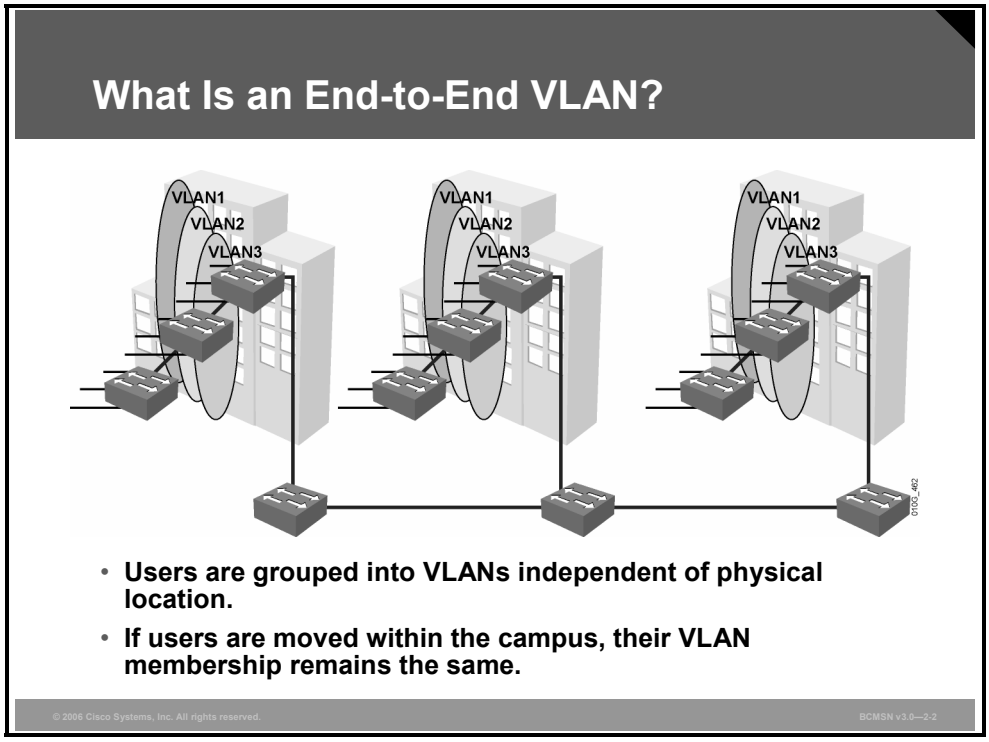
Objectives

Upon completing this lesson, you will be able to configure VLANs on access switches to confine traffic to individual VLANs in accordance with the Campus Infrastructure module design. This ability includes being able to meet these objectives:

- Define an end-to-end VLAN
- Define a local VLAN
- Describe the benefits of implementing local VLANs in a campus network
- Describe the VLAN configuration modes and their functions
- Define a VLAN access port
- List the commands to implement a VLAN
- List the steps to create a VLAN and associate it with an access port

Describing End-to-End VLANs

This topic describes end-to-end VLANs.



The term end-to-end VLAN refers to a single VLAN that is associated with switch ports that are widely dispersed throughout an enterprise network. Traffic for this VLAN is carried throughout the switched network. If many VLANs in a network are end-to-end, special links (trunks) are required between switches to carry the traffic of all the different VLANs.

An end-to-end VLAN has these characteristics:

- The VLAN is geographically dispersed throughout the network.
- Users are grouped into the VLAN regardless of physical location.
- As a user moves throughout a campus, the VLAN membership of that user remains the same.
- Users are typically associated with a given VLAN for network management reasons.
- All devices on a given VLAN typically have addresses on the same IP subnet.

Because a VLAN represents a Layer 3 segment, end-to-end VLANs allow a single Layer 3 segment to be geographically dispersed throughout the network. These could be some of the reasons for implementing this design:

- **Grouping users:** Users can be grouped on a common IP segment, even though they are geographically dispersed.
- **Security:** A VLAN may contain resources that should not be accessible to all users on the network, or there may be a reason to confine certain traffic to a particular VLAN.
- **Applying quality of service (QoS):** Traffic from a given VLAN can be given higher or lower access priority to network resources.

- **Routing avoidance:** If much of the VLAN user traffic is destined for devices on that same VLAN, and routing to those devices is not desirable, users can access resources on their VLAN without their traffic being routed off the VLAN, even though the traffic may traverse multiple switches.
- **Special purpose VLAN:** Sometimes a VLAN is provisioned to carry a single type of traffic that must be dispersed throughout the campus (for example, multicast, voice, or visitor VLANs).
- **Poor design:** For no clear purpose, users are placed in VLANs that span the campus, or even span WANs.

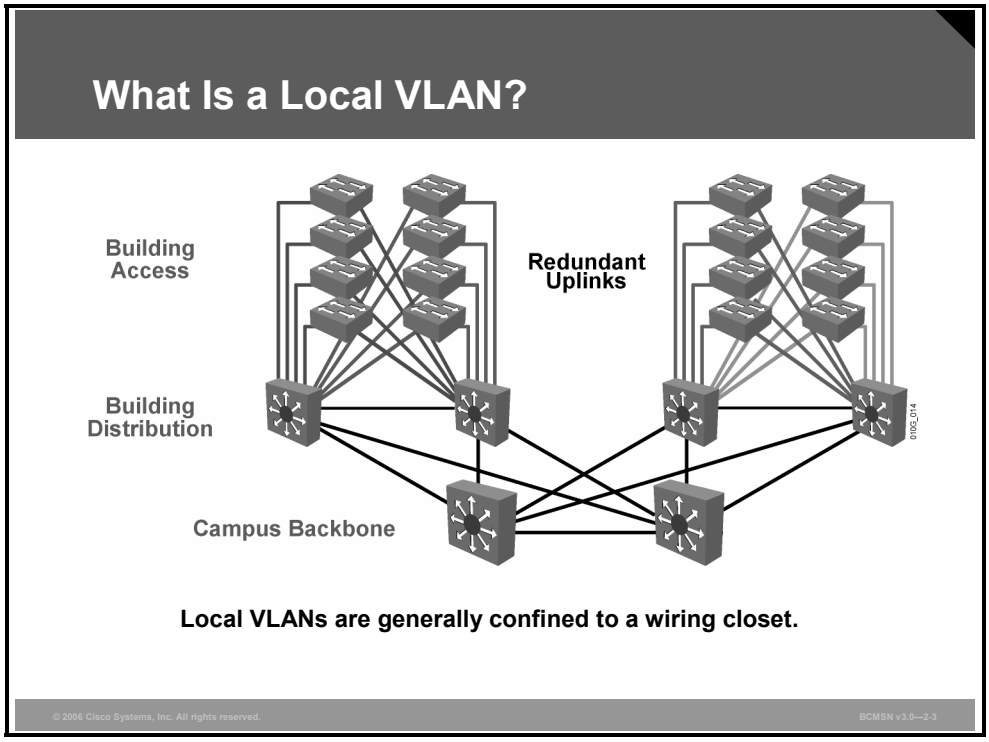
Some items should be considered when implementing end-to-end VLANS. Switch ports are provisioned for each user and associated with a given VLAN. Because users on an end-to-end VLAN may be anywhere in the network, all switches must be aware of that VLAN. This means that all switches carrying traffic for end-to-end VLANs are required to have identical VLAN databases. Also, flooded traffic for the VLAN is, by default, passed to every switch even if it does not currently have any active ports in the particular end-to-end VLAN. Finally, troubleshooting devices on a campus with end-to-end VLANs can be challenging because the traffic for a single VLAN can traverse multiple switches in a large area of the campus.

Example: End-to-End VLAN Implementation

In a military setting, one VLAN is designated to carry top-secret data. Users with access to that data are widely dispersed throughout the network. Because all devices on that VLAN have similar security requirements, security is handled by access lists at the Layer 3 devices that route traffic onto the segment (VLAN). Security can be applied VLAN-wide without addressing security at each switch in the network, which might have only a single user on the top-secret VLAN.

Describing Local VLANs

This topic describes local VLANs.



In the past, network designers attempted to implement the 80/20 rule when designing networks. The rule was based on the observation that, in general, 80 percent of the traffic on a network segment was passed between local devices, and only 20 percent of the traffic was destined for remote network segments. Therefore, end-to-end VLANs were typically used.

Designers now consolidate servers in central locations on the network and provide access to external resources such as the Internet through one or two paths on the network because the bulk of traffic now traverses a number of segments. Therefore, the paradigm now is closer a 20/80 proportion, in which the greater flow of traffic leaves the local segment, so Local VLANs have become more efficient.

In addition, the concept of end-to-end VLANs was very attractive when IP address configuration was a manually administered and burdensome process; therefore, anything that reduced this burden as users moved between networks was an improvement. But, given the ubiquity of DHCP, the process of configuring IP at each desktop is no longer a significant issue. As a result, there are few benefits to extending a VLAN throughout an enterprise.

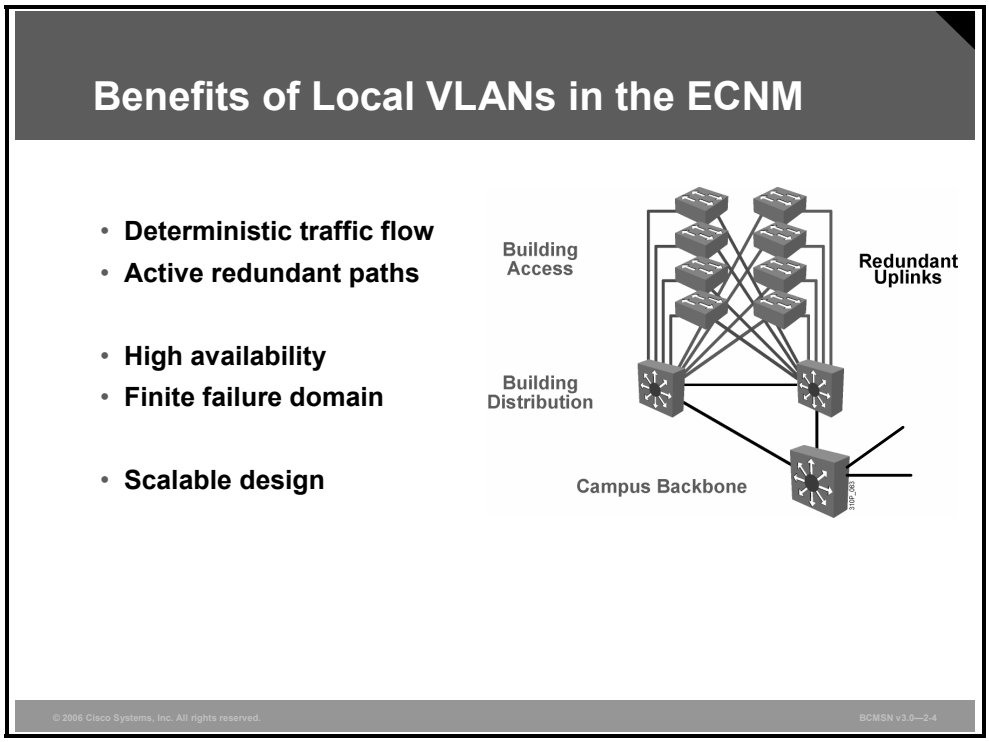
It is often more efficient to group all users of a set of geographically common switches into a single VLAN, regardless of the organizational function of those users, especially from a troubleshooting perspective. VLANs that have boundaries based upon campus geography rather than organizational function are called “local VLANs.” Local VLANs are generally confined to a wiring closet.

Here are some local VLAN characteristics and user guidelines:

- Local VLANs should be created with physical boundaries in mind rather than the job functions of the users on the end devices.
- Traffic from a local VLAN is routed to reach destinations on other networks.
- A single VLAN does not extend beyond the Building Distribution submodule.
- VLANs on a given access switch should not be advertised to all other switches in the network.

Benefits of Local VLANs in an Enterprise Campus Network

This topic describes the benefits of implementing local VLANs in an enterprise campus network.



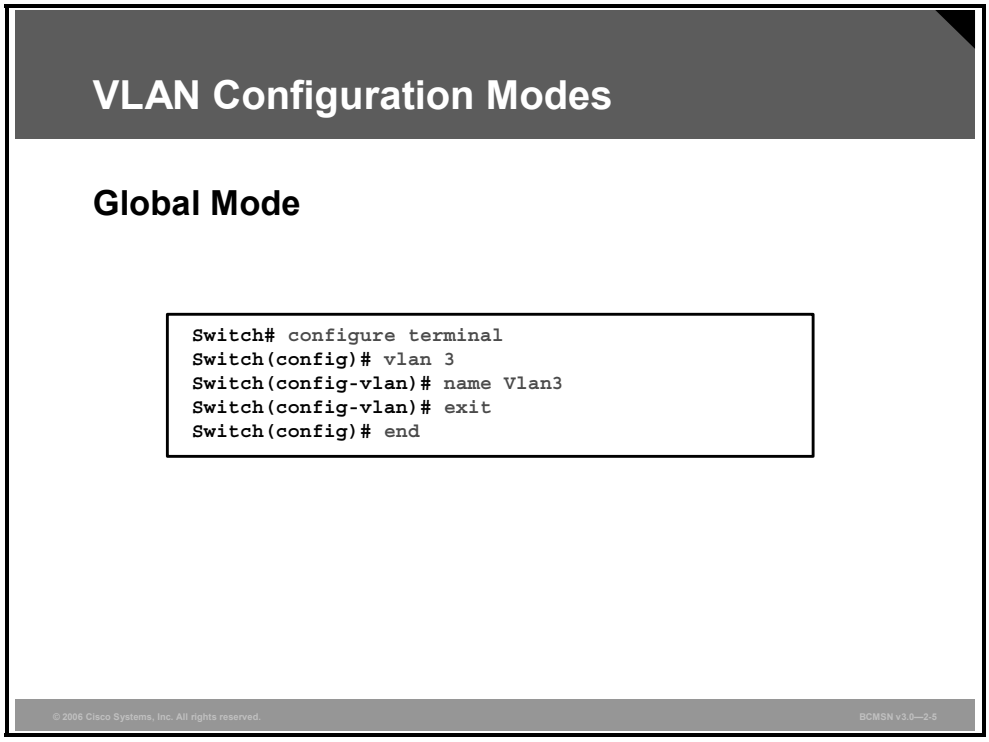
Local VLANs are part of the Enterprise Composite Network Model (ECNM) design where VLANs that are used at the access layer should extend no further than their associated distribution switch. Traffic is routed from the local VLAN as it is passed from the distribution layer into the core. This design can mitigate Layer 2 troubleshooting issues that occur when a single VLAN traverses the switches throughout an enterprise campus network. Implementing the ECNM using local VLANs provides these benefits:

- **Deterministic traffic flow:** The simple layout provides a predictable Layer 2 and Layer 3 traffic path. In the event of a failure that was not mitigated by the redundancy features, the simplicity of the model facilitates expedient problem isolation and resolution within the switch block.
- **Active redundant paths:** When implementing Per VLAN Spanning Tree (PVST) or Multiple Spanning Tree Protocol (MSTP), all links can be used to make use of the redundant paths.
- **High availability:** Redundant paths exist at all infrastructure levels. Local VLAN traffic on access switches can be passed to the building distribution switches across an alternative Layer 2 path in the event of primary path failure. Router Redundancy protocols can provide failover should the default gateway for the access VLAN fail. When both the Spanning Tree Protocol (STP) instance and VLAN are confined to a specific access and distribution block, then Layer 2 and Layer 3 redundancy measures and protocols can be configured to failover in a coordinated manner.

- **Finite failure domain:** If VLANs are local to a switch block, and the number of devices on each VLAN is kept small, failures at Layer 2 are confined to a small subset of users.
- **Scalable design:** Following the ECNM design, new access switches can be easily incorporated, and new submodules can be added when necessary.

VLAN Configuration Modes

This topic describes the VLAN configuration modes and their functions.



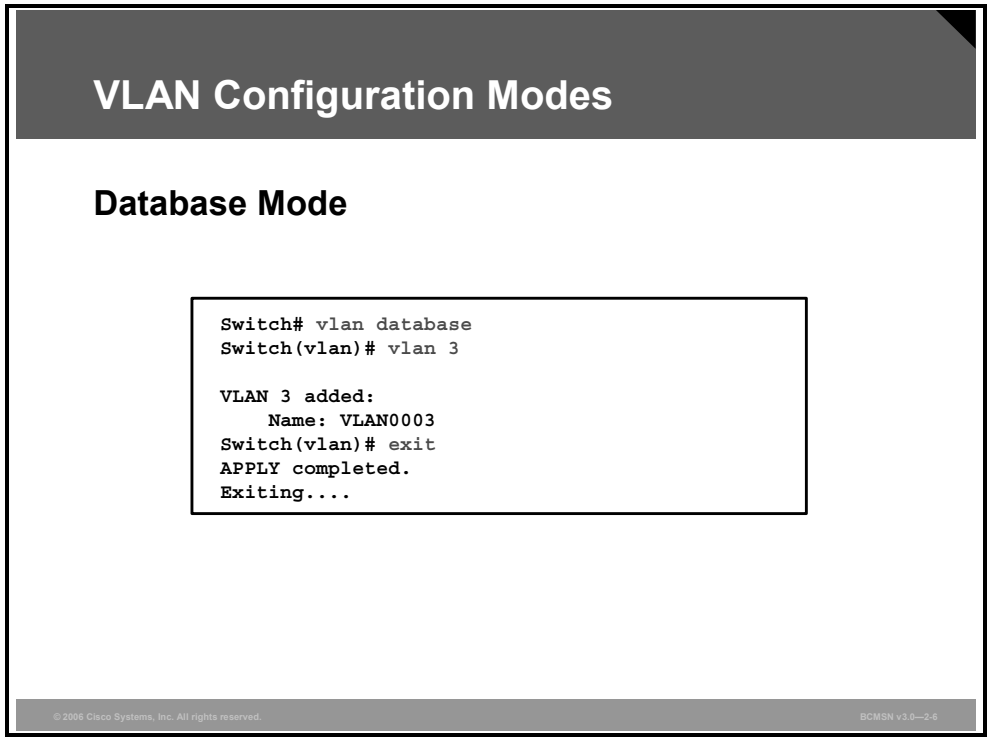
VLANs are created in either global configuration or VLAN database mode on most Cisco IOS software-based switches. Global configuration mode is the preferred way of creating and managing VLANs because the user interface is familiar.

When a VLAN is created or deleted, the change occurs as soon as the user presses the **Enter** key on the VLAN configuration line. The commands in this course will delineate VLAN creation and management using global configuration mode, as shown in the figure.

Note	Global configuration mode can be used to configure VLANs in the range 1 to 1005 and must be used to configure extended-range VLANs (1006 to 4094). The VLAN Trunk Protocol (VTP) configuration revision number is incremented each time a VLAN is created or changed.

VLAN Database Mode

This subtopic describes the VLAN database mode.



Alternatively, VLANs can be created and managed using VLAN database mode.

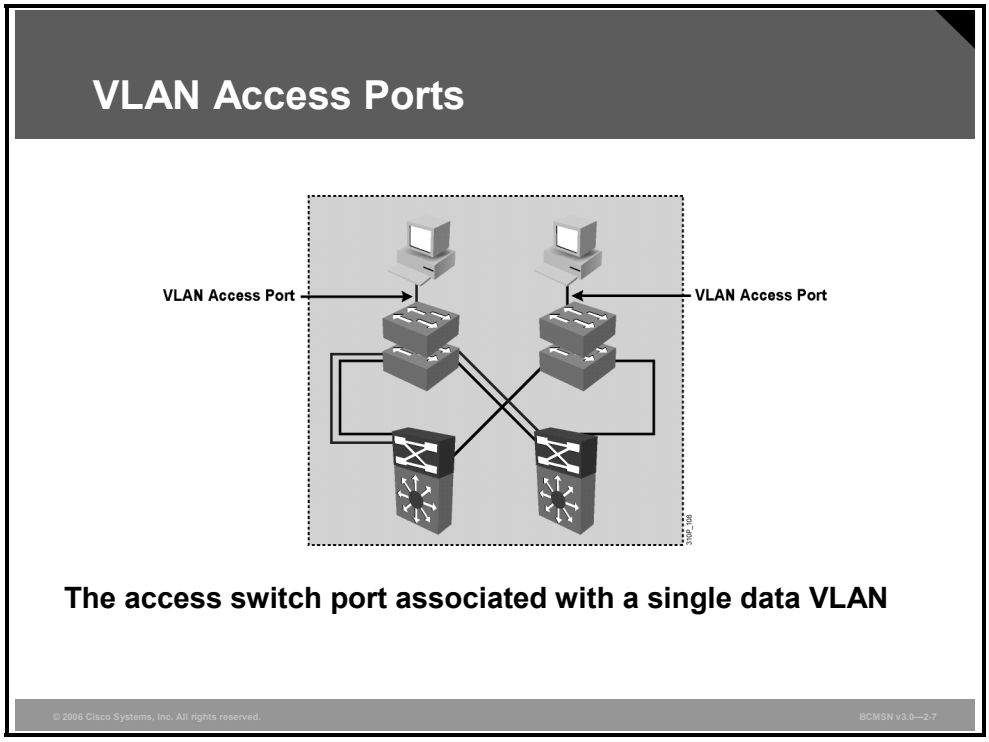
VLAN database mode is session oriented. When you add, delete, or modify VLAN parameters, the changes are not applied until you enter the **apply** or **exit** command. You can also exit VLAN database mode without applying the changes by entering the **abort** command.

To access this mode, the **vlan database** command is executed from privileged EXEC mode. From this mode, you can add, delete, and modify VLAN configurations for VLANs in the range 1 to 1005.

Note	This mode has been deprecated and will be removed in some future release.

Explaining VLAN Access Ports

This topic describes VLAN access ports.



When an end system is connected to a switch port, it needs to be associated with a VLAN, in accordance with the network design. To associate a device with a VLAN, the switch port to which the device connects will be assigned to a single data VLAN and thus becomes an access port. A switch port can become an access port through static or dynamic configuration.

On most switches, VLAN membership results from execution of a specific **switchport** configuration command. In a local VLAN strategy, the switch port is associated with the same VLAN as the other devices on that same switch or switch cluster.

Attributes and characteristics of access ports:

- An access port is associated with a single VLAN.
- The VLAN to which the access port is assigned must exist in the VLAN database of the switch, or the port will be associated with an inactive VLAN that does not forward frames.
- Because an access switch port is part of a VLAN or broadcast domain, that port will receive broadcasts, multicasts, unicast floods, and so forth that are sent to all ports in the VLAN.
- The end device will typically have an IP address in a subnet that is common to all other devices on the same access VLAN.

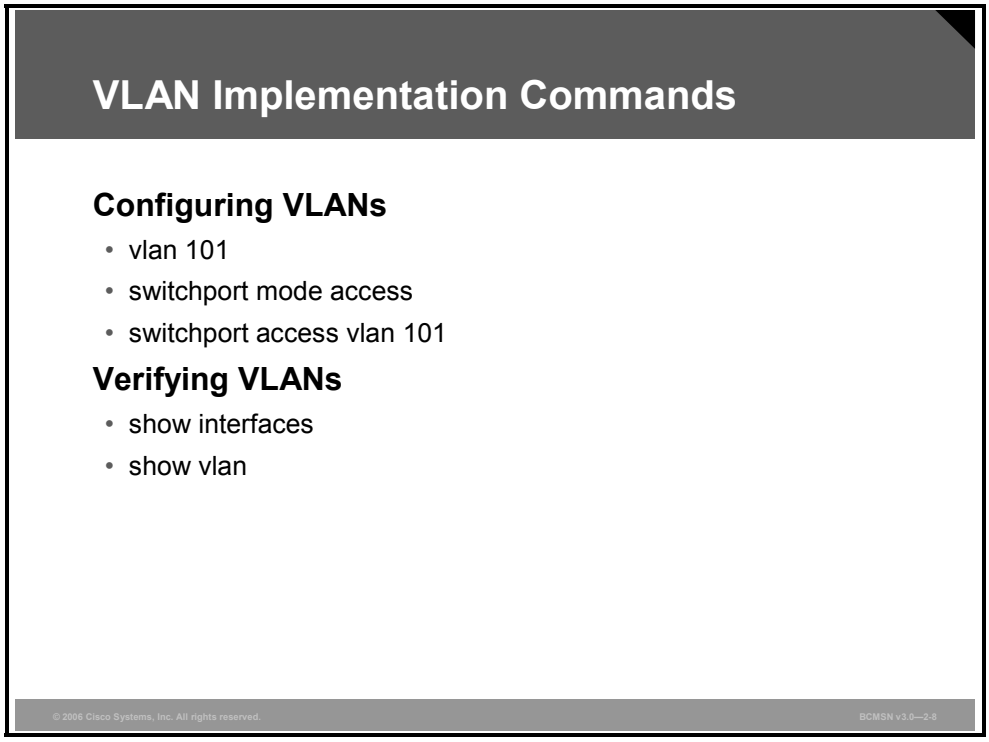
Dynamic Access Port Association

Switch ports can be dynamically associated with a given VLAN based upon the MAC address of the device connecting on that port. This requires that the switch query a VLAN Membership Policy Server (VMPS) to determine what VLAN to associate with a switch port, when a specific source MAC address is seen on the switch port.

This might be beneficial with a set of workstations that rove throughout the enterprise. Regardless of what switch or switch port the workstation connected to, that switch port would become an access port on a single, specific VLAN. Some security situations may require dynamic VLAN associations. However, dynamic VLANs are not consistent with the ECNM and will not be discussed further in this course.

Describing VLAN Implementation Commands

This topic describes the commands to configure a VLAN.



VLAN Implementation Commands

The table describes the primary commands used to implement VLANs and to verify their configuration in the Cisco Catalyst switch IOS interface.

Command	Description
`switch(config)#` **`vlan`** *`vlan-id`*	Creates VLAN *vlan-id.*
`switch(config)#` **`no vlan`** *`vlan-id`*	Deletes VLAN *vlan-id*.
`switch(config-vlan)#` **`name`** *`vlan_name`*	Assigns a specific name to the VLAN.
`switch(config-if)#` **`switchport mode access`**	Specifies that this port is to function at Layer 2 and places the port in VLAN access mode. Typically followed by the **switchport access vlan** *vlan-id* command.
`switch(config-if)#` **`switchport access vlan`** *`vlan-id`*	Associates a single switch port as an access port to a single VLAN *vlan-id.*
`switch(config-if)#` **`no shutdown`**	Enables a switch port or interface.
`switch(config)#` **`shutdown vlan`** *`vlan-id`*	Suspends local traffic on the specified VLAN. Does not change the VLAN information in the VTP database, and the switch still advertises VTP information.

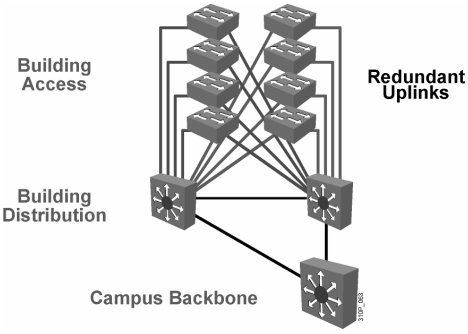
Command	Description
switch# **show vlan**	Displays the parameters for all configured VLANs or for one VLAN on the switch.
switch# **show interfaces** *type slot/port* **switchport**	Displays the switchport configuration of the interface

Implementing a VLAN

This topic describes the procedure to create a VLAN and associate it with an access port.

How to Implement a VLAN

- **Create or configure a VLAN.**
- **Verify VLAN configuration.**
- **Associate switch ports with the VLAN.**
- **Verify switch port configuration.**
- **Test VLAN connectivity.**
- **Implement VLAN and switch security.**



To create or configure a VLAN and associate switch ports, follow these steps:

Step 1 Create the VLAN.

Step 2 Verify the VLAN configuration.

Step 3 Associate switch ports with the VLAN.

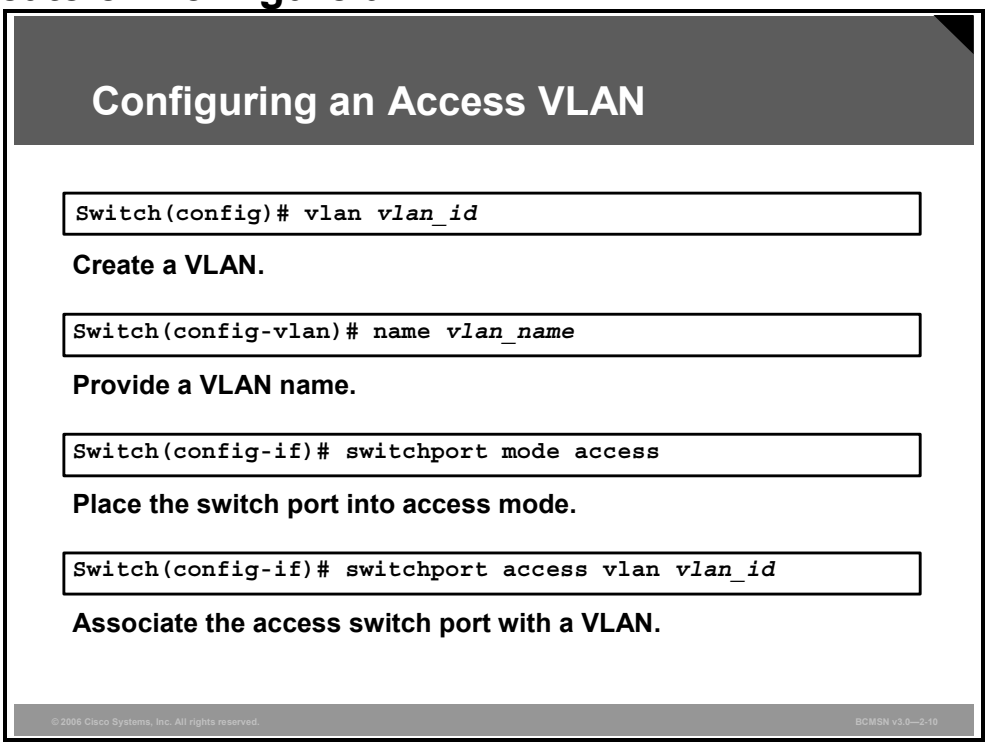
Step 4 Verify the switch port configuration.

Step 5 Test VLAN connectivity.

Step 6 Implement switch and VLAN security measures.

These steps are explained in greater detail in the remainder of this topic.

1. Create or Configure a VLAN



Before assigning a switch port to a specific VLAN, the VLAN may need to be created. The example that follows shows the syntax for creating a VLAN using the Cisco IOS interface.

To create a VLAN or enter VLAN configuration mode, use the **vlan** command:

```
Switch(config)# vlan vlan_id
```

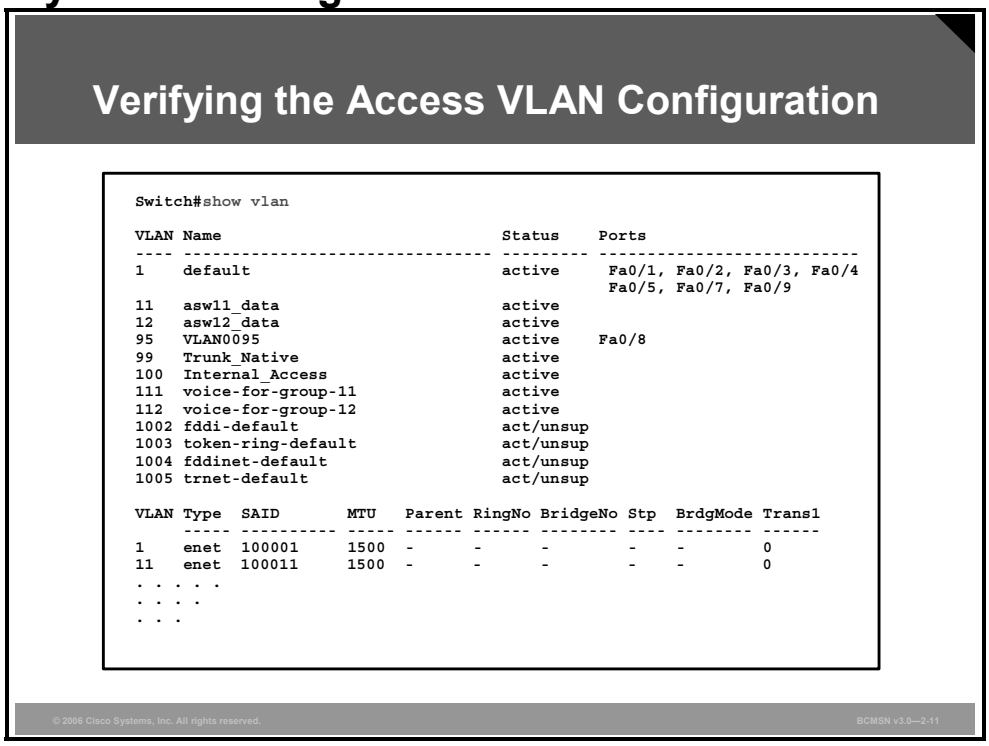
VLAN Creation Arguments

Argument	Description
vlan_id	Required: Any valid VLAN number from 1-4094 if accepted by the switch platform, 1-1024 if not. If VLAN does not presently exist, a VLAN with this *vlan_id* will be created, and prompt will change to VLAN config mode. If the VLAN already exists, prompt will change, and VLAN parameters can be altered for this *vlan_id.*

To enter a description of the VLAN from the VLAN configuration mode:

```
Switch(config-vlan)# description vlan_description
```

2. Verify VLAN Configuration



Show VLAN

Execute the **show vlan** command from privileged EXEC mode. It displays information about a particular VLAN. The fields in the **show vlan** command output are described in the table.

Field	Description
VLAN	VLAN number
Name	Name of the VLAN, if configured
Status	Status of the VLAN (active or suspended)
Ports	Ports that belong to the VLAN
Type	Media type of the VLAN
SAID	Security association ID value for the VLAN
MTU	Maximum transmission unit size for the VLAN
Parent	Parent VLAN, if one exists
RingNo	Ring number for the VLAN, if applicable
BrdgNo	Bridge number for the VLAN, if applicable
STP	Spanning Tree Protocol type used on the VLAN
BrdgMode	Bridging mode for this VLAN
Trans1	Translation bridge 1
AREHops	Maximum number of hops for all-routes explorer frames
STEHops	Maximum number of hops for spanning tree explorer frames

3. Associate Switch Ports with the VLAN

Switch ports that are to function at Layer 2 and carry traffic for a single VLAN are configured as access switch ports and are assigned an access VLAN.

To configure a Layer 2 switch port as an access port:

```
Switch(config-if)# switchport mode access
```

Switch Port Parameters

This table describes the parameters for the **switchport mode access** command.

Parameter	Description
switchport	Required: Configures the interface to function as a Layer 2 port only. On many switches, this is the default. **No switchport** would reverse this process and, on some switch platforms, convert the port to a Layer 3 port.
mode access	Required: Switch port must be configured in access mode if you will next assign a specific access VLAN. Alternative mode options are available for nonaccess port functionality.

To assign the access port to a specific VLAN:

```
Switch(config-if)# switchport access vlan vlan_id
```

Switch Port Access Parameters

Parameter	Description
switchport	Required: Indicates further configuration of Layer 2 functionality of switch port.
access	Required: Indicates further configuration of access features of the switch port.
vlan *vlan_id*	Required: Indicates what single VLAN number is to be associated with this access port. On some switch platforms, this command will create a VLAN, not just associate an ID.

4. Verify Switch Port Configuration

These commands are useful for verifying that a switch port is configured as intended:

show interface *type slot/port* **switchport**

show running-config interface *type slot/port*

show vlan

Show Running-Config interface

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
!
Current configuration :33 bytes
interface FastEthernet 5/6
 switchport access vlan 200
```

```
 switchport mode access
end
```

5. Test VLAN Connectivity

After placing a device on the configured switch port, these steps will help verify if the device is connecting to the VLAN as intended:

Step 1 Ensure that the connected device has a correctly configured IP address and a subnet mask that places it on the same network as the default gateway.

Step 2 Ping the default gateway.

Step 3 Check to see if the ping to the default gateway is successful. If so, the VLAN configuration and the IP address configuration have been verified.

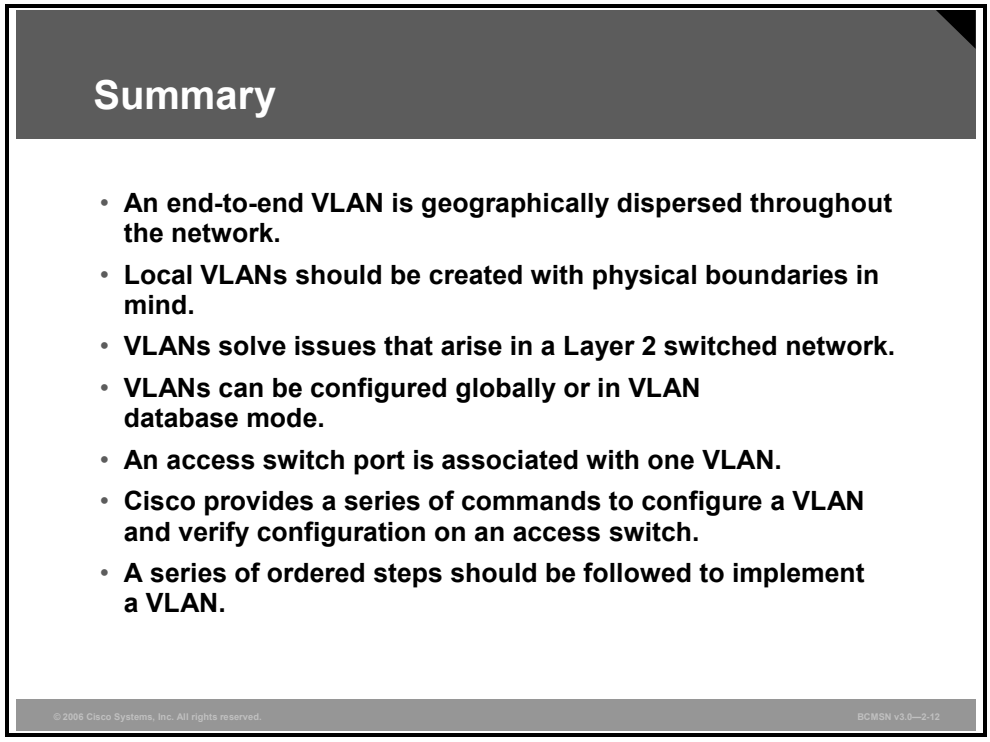
6. Implement Switch and VLAN Security Measures

When implementing VLANs, you should consider a few measures to secure the VLAN and the switch itself. The security policy of the organization will likely have more detailed recommendations, but these can provide a foundation.

- Create a “parking-lot” VLAN with a VLAN ID (VID) other than VLAN1, and place all unused switch ports in this VLAN. This VLAN may provide the user with some minimal network connectivity. (Check on the security policy of your organization before implementing.)
- Disable unused switch ports, depending on the security policy of the organization.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 3

Implementing Trunks

Overview

Switch ports carrying traffic for multiple VLANs are called trunk ports. As frames from multiple VLANs traverse trunk ports, the switch must identify each frame to associate it with a given VLAN. This lesson will examine the differences between Inter-Switch Link (ISL) and 802.1Q, two protocols used to mark frames on a trunk link.

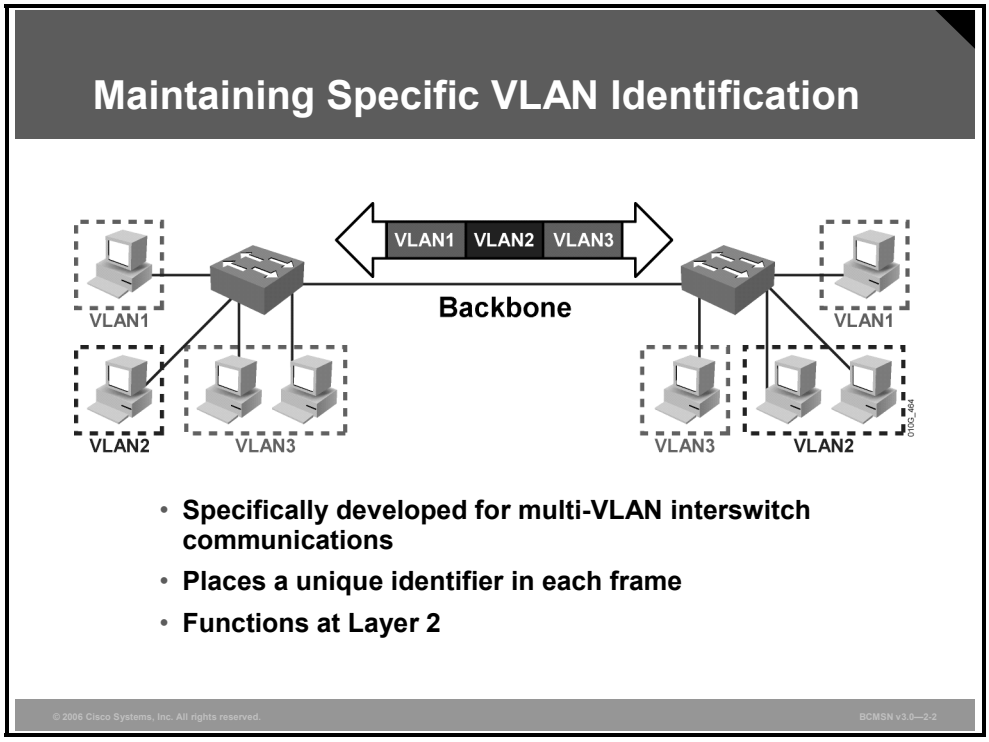
Objectives

Upon completing this lesson, you will be able to explain the procedure for configuring both 802.1Q and ISL trunking between two switches so that VLANs that span the switches can connect. This ability includes being able to meet these objectives:

- Describe a VLAN trunk in an enterprise network
- Describe ISL trunking
- Describe 802.1Q trunking
- Define an 802.1Q native VLAN
- Explain VLAN ranges and their usage
- Identify the commands used to configure trunking
- Explain the procedure to configure trunking

Explaining VLAN Trunks

This topic describes a VLAN trunk in an enterprise network.



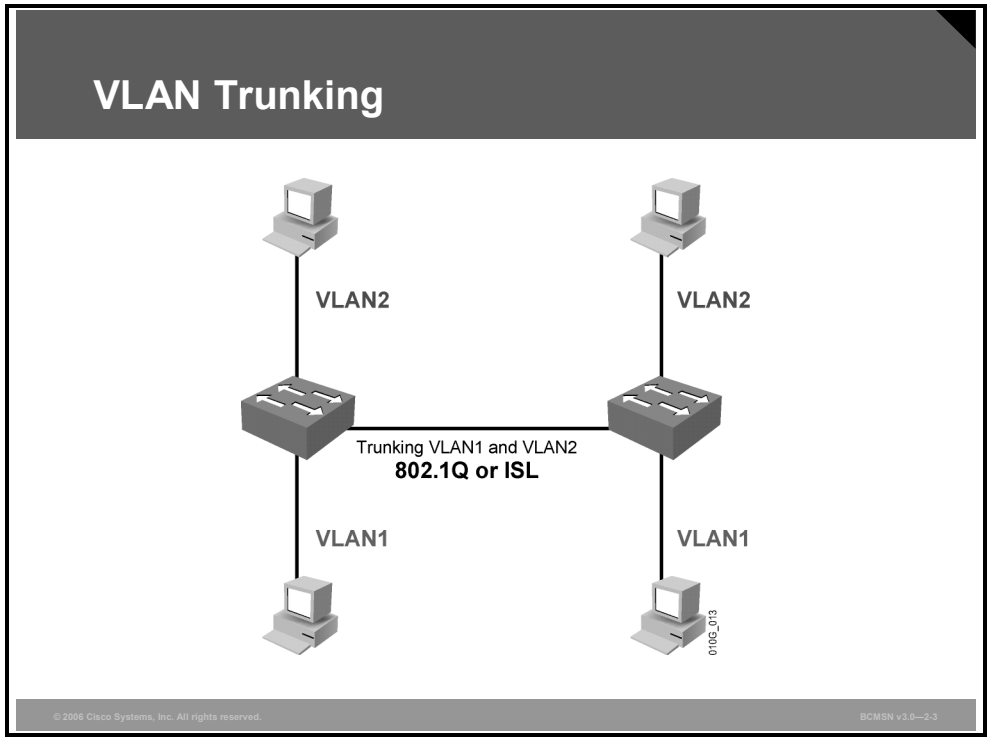
Multiple VLANs are supported between switches through the use of VLAN trunks. A trunk is a Layer 2 link between switches that are running a specialized trunking protocol. Trunks carry the traffic of multiple VLANs over physical links (multiplexing) and enable the extension of a single Layer 2 VLAN between switches.

If frames from a single VLAN traverse a trunk link, a trunking protocol must mark the frame to identify its associated VLAN as the frame is placed onto the trunk link. The receiving switch then knows the frame's VLAN of origin and can process the frame accordingly.

On the receiving switch, the VLAN ID (VID) is removed when the frame is forwarded onto an access link associated with its VLAN.

VLAN Trunking Protocols

This subtopic introduces two examples of trunking protocols.



A special protocol is required to establish a trunk link between two devices. A trunk link may exist between these devices:

- Two switches
- A switch and a router
- A switch and a trunk-capable network interface card (NIC) in a node such as a server

If a single physical link carries traffic for multiple VLANs, each frame must be "marked" with a VID so it is differentiated from frames coming from other VLANs. This marking or frame identification is accomplished through the implementation of a trunking protocol. Frame identification uniquely assigns an ID, referred to as a VID, to each frame. Each receiving switch examines this VID to determine the destination VLAN of the frame.

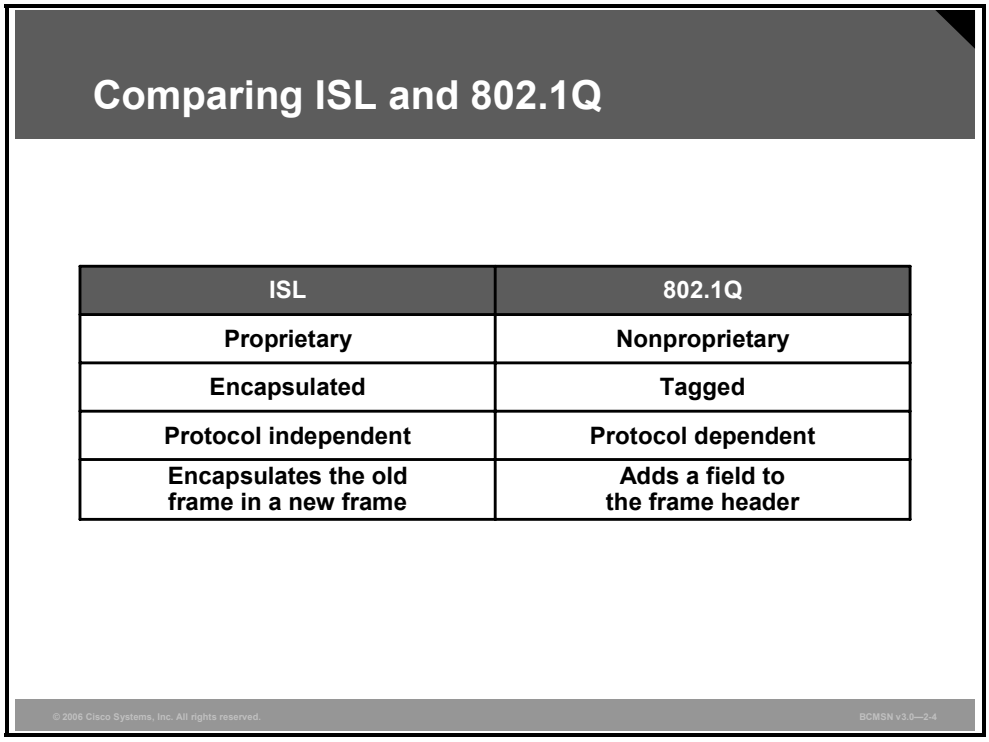
VIDs are associated with only those frames that traverse a trunk link. When a frame enters or exits the switch on an access link, no VID is present. The ASIC on the switch port assigns the VID to a frame as it is placed on a trunk link and also strips off the VID if the frame exits an access switch port.

Trunk links should be managed so that they carry traffic for intended VLANs only. This practice keeps unwanted VLAN data traffic from traversing links unnecessarily. Trunk links are used between the access and distribution layers of the campus switch block. These are the trunk protocols used to carry multiple VLANs over a single link:

- ISL: Cisco ISL
- 802.1Q: IEEE standard trunking protocol

Comparing ISL and 802.1Q Trunking Protocols

This subtopic compares the features of ISL and 802.1Q trunking protocols.



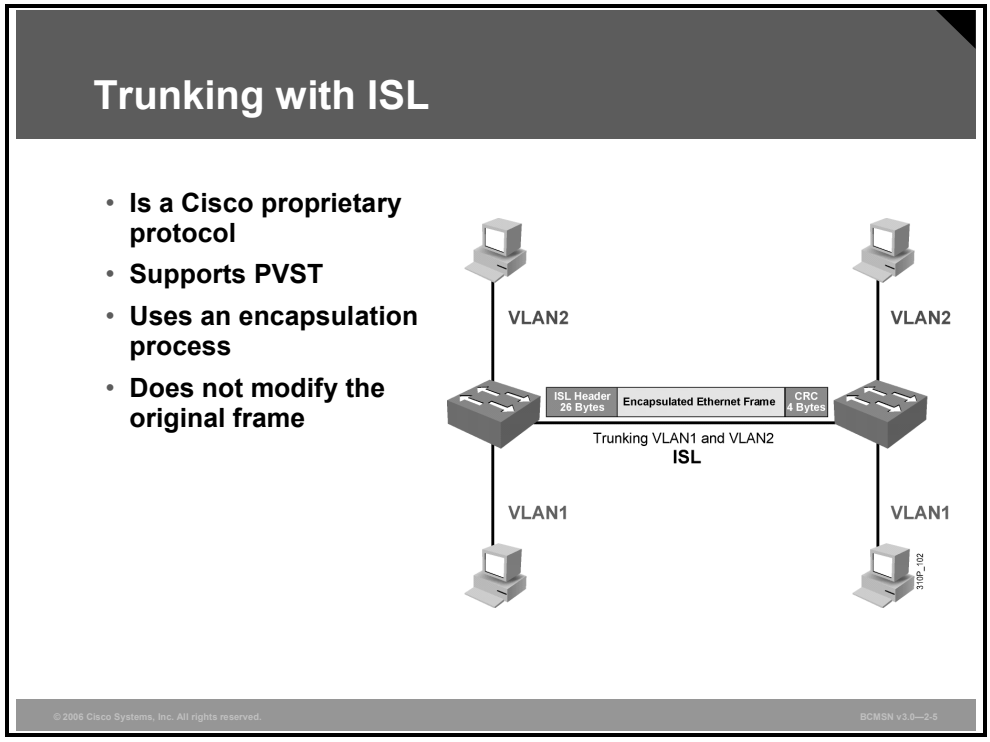
Depending on the trunking protocol, data frames sent across a trunk link are either encapsulated or tagged. The purpose of encapsulating or tagging frames is to provide the receiving switch with a VID to identify the VLAN from which the frame originated. The trunking protocol ISL, a Cisco Systems proprietary protocol, encapsulates frames, whereas IEEE 802.1Q inserts a tag into the original Layer 2 data frame.

802.1Q is not proprietary and can be deployed in any Ethernet standards-based Layer 2 device. It is specific to a single Layer 2 protocol (Ethernet) because it modifies the Layer 2 Ethernet frame by inserting a tag between two specific fields of the frame and therefore must be aware of the frame header details.

ISL is Layer 2-protocol independent. Because the original Layer 2 frame is fully encapsulated and not altered, ISL can transport data frames from various Layer 2 media types.

Describing ISL Trunking

This topic describes ISL trunking.



ISL is a Cisco proprietary protocol option for configuring Layer 2 trunk links. It is the original standard for trunking between switches and predates IEEE trunking standards. ISL takes original Layer 2 frames and encapsulates them with a new ISL header and trailer, cyclic redundancy check (CRC), before placing them on the trunk link.

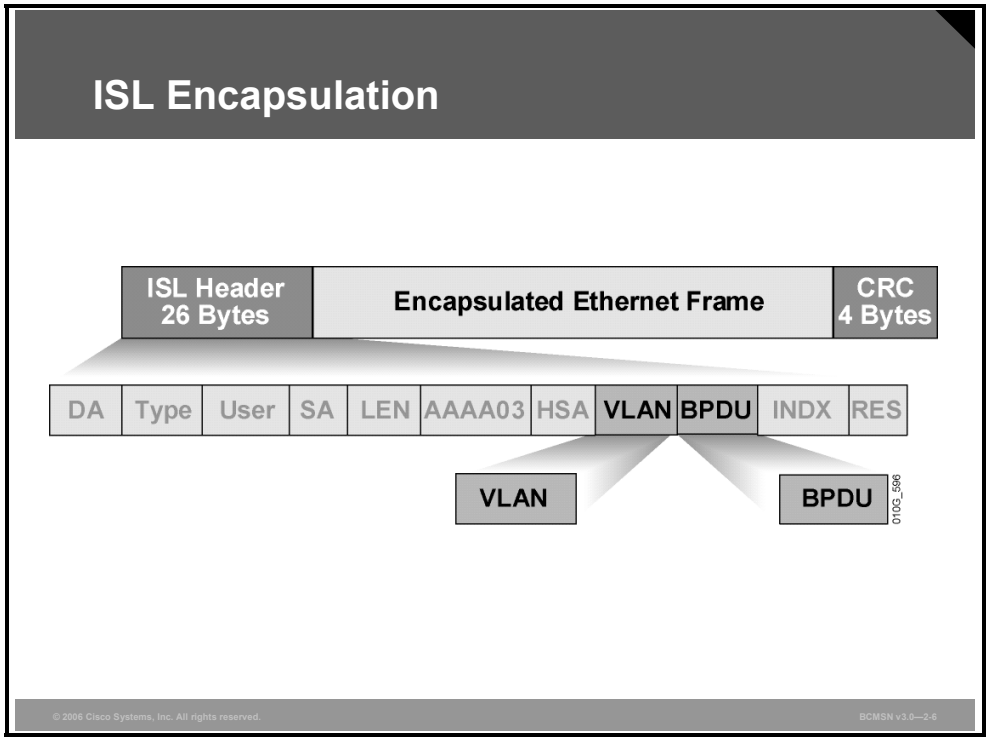
Because an entirely new header is appended to the original frame, the header offers some features not found in 802.1Q, an alternative trunking protocol.

These are some features of the ISL protocol:

- It supports multiple Layer 2 protocols (Ethernet, Token Ring, FDDI, and ATM).
- It supports Per VLAN Spanning Tree (PVST) protocol.
- It does not use a native VLAN, so it encapsulates every frame.
- The encapsulation process leaves original frames unmodified.

ISL Encapsulation Process

This subtopic describes how the ISL trunking protocol encapsulates frames.



When a switch port is configured as an ISL trunk port, the entire original Layer 2 frame, including header and frame check sequence (FCS) trailer, will be encapsulated before it traverses the trunk link. Encapsulation is the process of placing an additional header in the front and a trailer at the end of the original Layer 2 frame. The ISL header will contain the VID of the VLAN where the frame originated. At the receiving end, the VID is read, the header and trailer are removed, and the original frame is forwarded like any regular Layer 2 frame on that VLAN.

Only ISL trunk ports can properly receive ISL encapsulated frames. A non-ISL port receiving an ISL frame may consider the frame size to be invalid or may not recognize the fields in the header. The frame will likely be dropped and counted as a transmission error when received by a non-ISL port.

ISL Header

The ISL header contains various fields with values that define attributes of the original Layer 2 data within the encapsulated frame. This information is used for forwarding, media identification, and VLAN identification. The population of the fields within the ISL header varies, based on the type of VLAN and the media of the link. The ASIC on an Ethernet port encapsulates the frames with a 26-byte ISL header and a 4-byte FCS. This 30-byte ISL encapsulation overhead is consistent among the Layer 2 protocols supported on Cisco Catalyst switches, but the overall size of the frame will vary and be limited by the maximum transmission unit (MTU) of the original Layer 2 protocol.

The ISL Ethernet frame header contains these information fields:

- **DA (destination address):** 40-bit destination address. This is a multicast address and is set at 0x01-00-0C-00-00 or 0x03-00-0c-00-00. The first 40 bits of the DA field signal to the receiver that the packet is in ISL format.
- **Type:** 4-bit descriptor of the encapsulated frame types: Ethernet (0000), Token Ring (0001), FDDI (0010), and ATM (0011).
- **User:** 4-bit descriptor used as the Type field extension or to define Ethernet priorities; it is a binary value from 0, the lowest priority, to 3, the highest priority. The default User field value is "0000." For Ethernet frames, the User field bits "0" and "1" indicate the priority of the packet as it passes through the switch.
- **SA (source address):** 48-bit source MAC address of the transmitting Cisco Catalyst switch port.
- **LEN (length):** 16-bit frame-length descriptor minus DA, Type, User, SA, LEN, and CRC.
- **AAAA03:** Standard Subnetwork Access Protocol (SNAP) 802.2 logical link control (LLC) header.
- **HSA (high bits of source address):** First 3 bytes of the SA (manufacturer or unique organizational ID).
- **VID:** 15-bit VID. Only the lower 10 bits are used for 1024 VLANs.
- **BPDU (bridge protocol data unit):** 1-bit descriptor identifying whether the frame is a spanning tree BPDU. It also identifies if the encapsulated frame is a Cisco Discovery Protocol (CDP) or VLAN Trunk Protocol (VTP) frame and indicates if the frame should be sent to the control plane of the switch.
- **INDX (index):** 16 bits to indicate the port index of the source of the packet as it exits the switch. It is used for diagnostic purposes only and may be set to any value by other devices. It is a 16-bit value and is ignored in received packets.
- **RES:** 16 bits reserved for Token Ring and FDDI frames.
- **Encapsulated Ethernet Frame:** Encapsulated data packet, including its own CRC value, completely unmodified. The internal frame must have a CRC value that is valid when the ISL encapsulation fields are removed. A receiving switch may strip off the ISL encapsulation fields and use this ENCAP FRAME field as the frame is received (associating the appropriate VLAN and other values with the received frame as indicated for switching purposes).

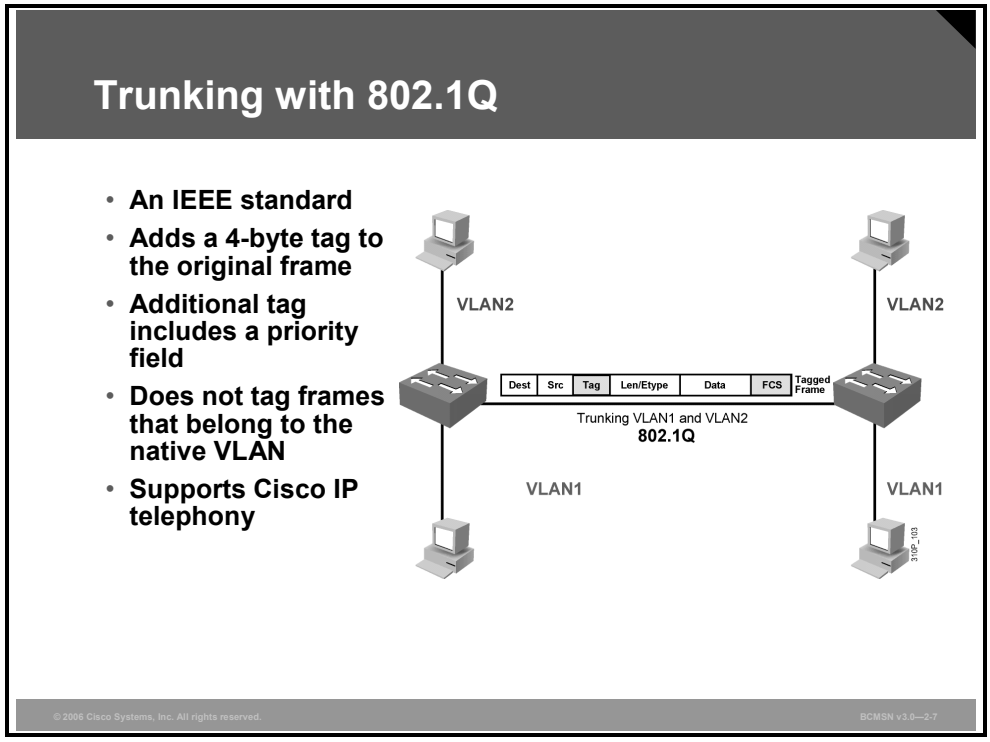
ISL Trailer

The trailer portion of the ISL encapsulation is an FCS that carries a CRC value calculated on the original frame plus the ISL header as the ISL frame was placed onto the trunk link. The receiving ISL port recalculates this value. If the CRC values do not match, the frame is discarded. If the values match, the switch discards the FCS as a part of removing the ISL encapsulation so that the original frame can be processed. The ISL trailer consists of these frame checks:

- **FCS:** Consists of 4 bytes. This sequence contains a 32-bit CRC value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames. The FCS is generated over the DA, SA, LEN, Type, and Data fields. When an ISL header is attached, a new FCS is calculated for the entire ISL packet and added to the end of the frame.

Describing 802.1Q Trunking

This topic describes 802.1Q trunking.



Like ISL, 802.1Q is a protocol that allows a single physical link to carry traffic for multiple VLANs. It is the IEEE standard VLAN trunking protocol. Rather than encapsulating the original Layer 2 frame in its entirety, 802.1Q inserts a tag into the original Ethernet header, then recalculates and updates the FCS in the original frame and transmits the frame over the trunk link.

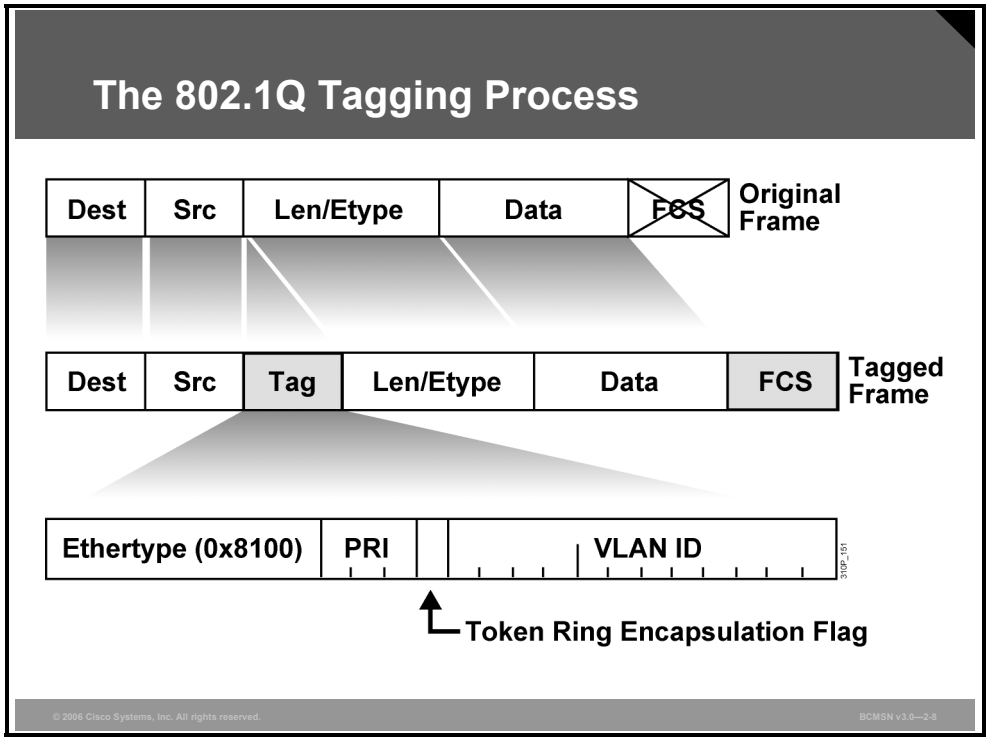
The 802.1Q protocol, often referred to as “dot-1Q,” offers the clear benefit of being the first IEEE standards-based trunking protocol for Ethernet. It allows multiple VLANs to traverse infrastructure equipment where cross-vendor links exist.

These are features of the 802.1Q protocol:

- Support for Ethernet and Token Ring
- Support for 4096 VLANs
- Support for Common Spanning Tree (CST), Multiple Spanning Tree Protocol (MSTP) protocol, and Rapid Spanning Tree Protocol (RSTP)
- Point-to-multipoint topology support
- Support for untagged traffic over the trunk link via native VLAN
- Extended quality of service (QoS) support
- Growing standard for IP telephony links

802.1Q Tagging Process

This subtopic describes the 802.1Q process for tagging frames that traverse a trunk link.



To identify a frame with a given VLAN, the 802.1Q protocol adds a tag, or a field, to the standard Layer 2 Ethernet data frame. The components of this tag are shown in the figure. Because inserting the tag alters the original frame, the switch must recalculate and alter the FCS value for the original frame before sending it out the 802.1Q trunk port. In contrast, ISL does not modify the original frame at all.

The new 802.1Q Tag field has these components:

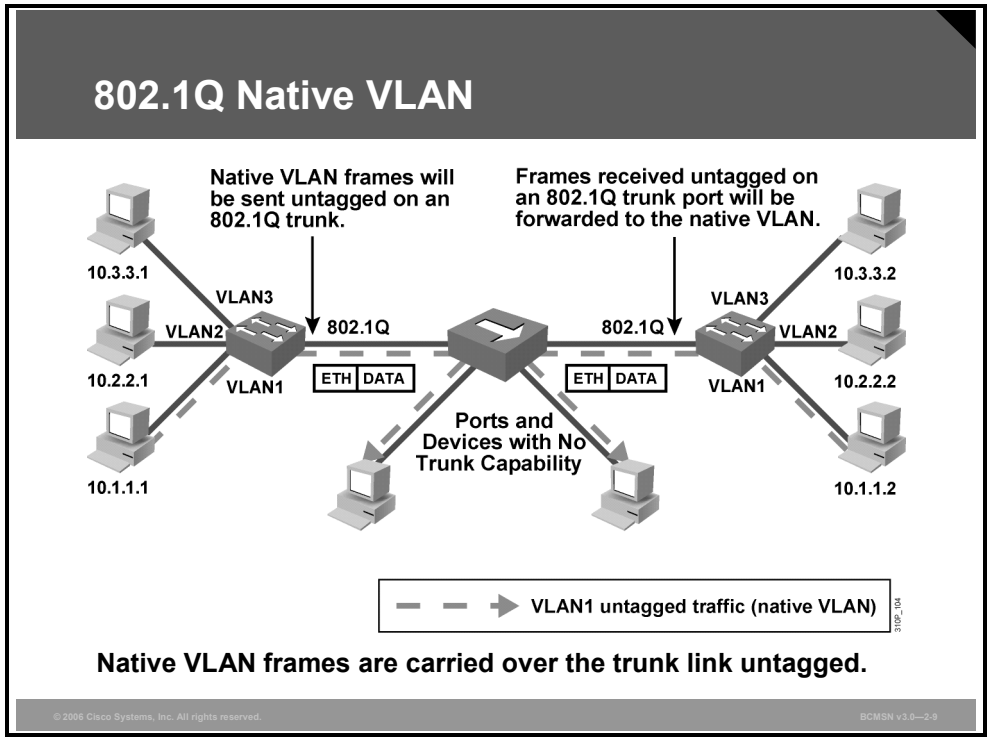
- **EtherType:** Uses EtherType 0x8100 to indicate that this is an 802.1Q frame.
- **PRI:** 3 bits; carries priority information for the frame.
- **Token Ring Encapsulation Flag:** Indicates the canonical interpretation of the frame if it is passed from Ethernet to Token Ring. This value is always set to zero for Ethernet switches.
- **VID:** VLAN association of the frame. By default, all normal and extended-range VLANs are supported.

If a non-802.1Q-enabled device or an access port receives an 802.1Q frame, the tag data is ignored, and the packet is switched at Layer 2 as a standard Ethernet frame. This allows for the placement of Layer 2 intermediate devices, such as other switches or bridges, along the 802.1Q trunk path. To process an 802.1Q tagged frame, a device must allow an MTU of 1522 or higher.

Note An Ethernet frame that has a larger MTU than expected (1518 by default for Ethernet), but no larger than 1600 bytes, will register as a Layer 2 error frame called a “baby giant.” For ISL, the original frame plus ISL encapsulation can generate a frame as large as 1548 bytes, and 1522 bytes for an 802.1Q tagged frame.

Explaining 802.1Q Native VLANs

This topic defines an 802.1Q native VLAN.



When configuring an 802.1Q trunk, a matching native VLAN must be defined on each end of the trunk link. A trunk link is inherently associated with tagging each frame with a VID. The purpose of the native VLAN is to allow frames that are not tagged with a VID to traverse the trunk link. An 802.1Q native VLAN is defined as one of these:

- The VLAN that a port is associated with when not in trunking operational mode
- The VLAN that is associated with untagged frames that are received on a switch port
- The VLAN to which Layer 2 frames will be forwarded if received untagged on an 802.1Q trunk port

Compare this to ISL, in which no frame may be transported on the trunk link without encapsulation, and any unencapsulated frames received on a trunk port are immediately dropped.

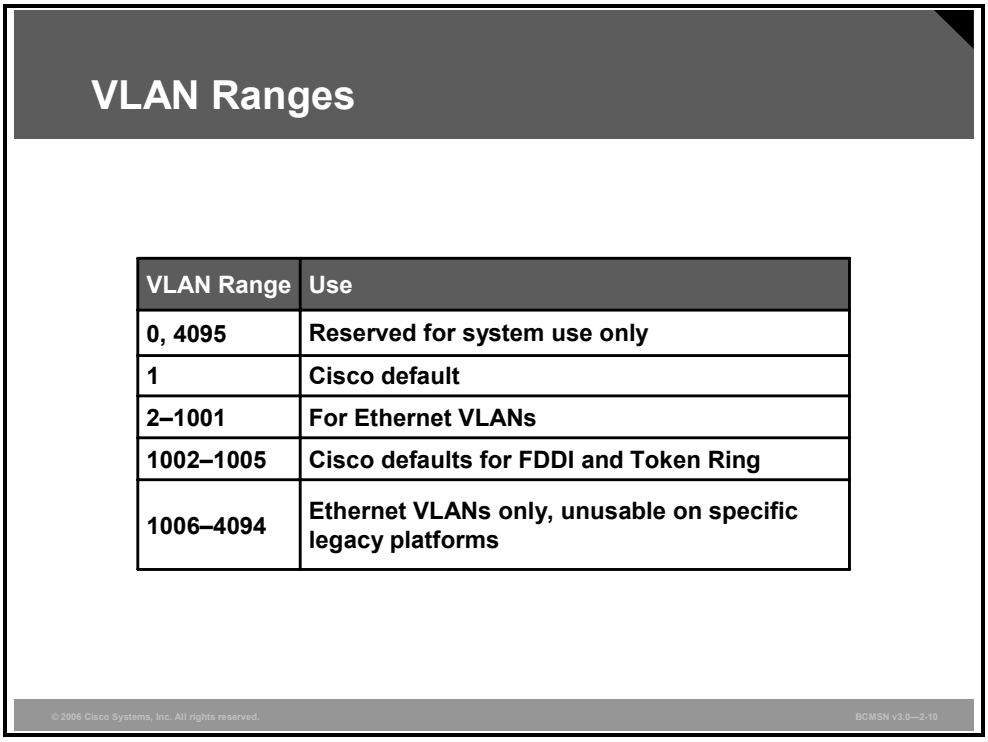
Each physical port has a parameter called a port VLAN ID (PVID). Every 802.1Q port is assigned a PVID value equal to the native VID. When a port receives a tagged frame that is to traverse the trunk link, the tag is respected. For all untagged frames, the PVID is considered the tag. This allows the frames to traverse devices that may be unable to read VLAN tag information.

Native VLANs have these attributes:

- A trunk port will support only one native active VLAN per operational mode. The modes are access and trunk.
- By default, on Cisco Catalyst switches, all switch ports and native VLANs for 802.1Q are assigned to VLAN1.
- The 802.1Q trunk ports connected to each other via physical or logical segments must all have the same native VLAN configured to operate correctly.
- If the native VLAN is misconfigured for trunk ports on the same trunk link, Layer 2 loops can occur due to diverting Spanning Tree Protocol (STP) BPDUs from their correct VLAN.

Explaining VLAN Ranges

This topic explains VLAN ranges and their uses.



Each VLAN on the network must have a unique VID. The valid range of user-configurable ISL VLANs is 1 to 1024. The valid range of VLANs specified in the IEEE 802.1Q standard is 1 to 4094.

VLAN Ranges

This table describes VLAN ranges and their use.

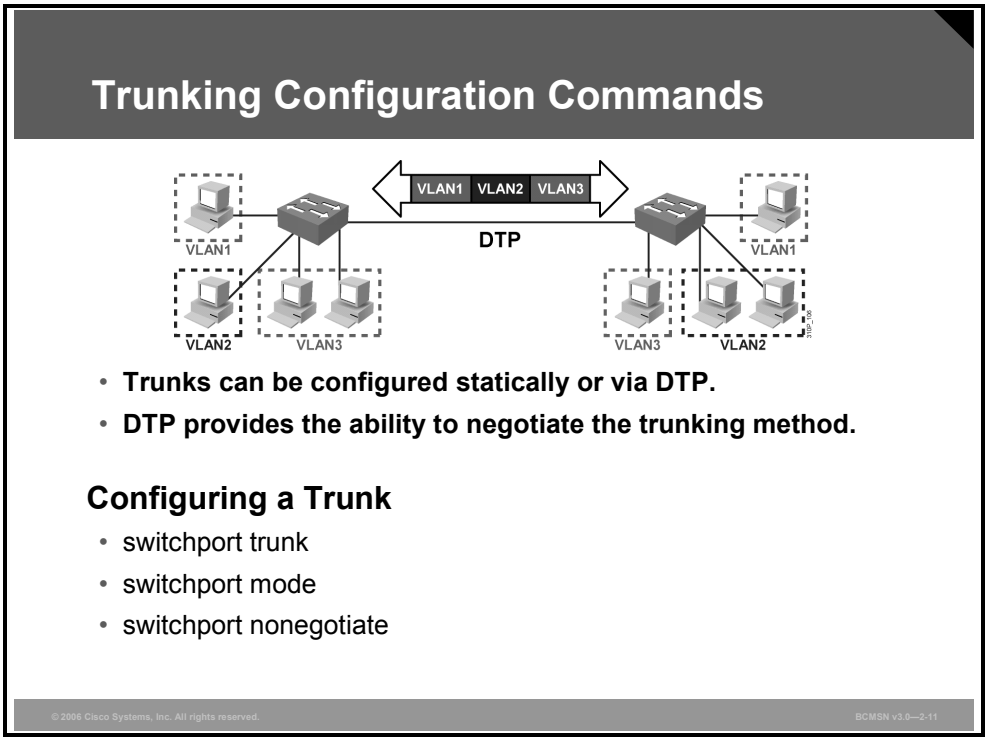
VLAN Ranges	Range	Use	VTP Propagated
0, 4095	Reserved	For system use only. VLANs cannot be seen or used.	—
1	Normal	Cisco default VLAN. This VLAN can be used but not modified or deleted.	Yes
2–1001	Normal	These VLANs can be created, used, and deleted.	Yes
1002–1005	Normal	Cisco defaults for FDDI and Token Ring. These cannot be deleted.	Yes

VLAN Ranges	Range	Use	VTP Propagated
1006–4094	Extended	For Ethernet VLANs only. Layer 3 ports and some software features require internal VLANs. Internal VLANs are allocated from 1006 and up. You cannot use a VLAN that has been allocated for such use. To display the VLANs used internally, enter the **show vlan internal usage** command. Switches running Cisco Catalyst product series software do not support configuration of VLANs 1006-1024. If you configure VLANs 1006-1024, ensure that the VLANs do not extend to any switches running Cisco Catalyst product series software. You must enable the extended system ID to use extended-range VLANs.	No

As a best practice, assign extended VLANs beginning with 4094 and work downward because some switches use extended-range VIDs for internal use, starting at the low end of the extended range. Refer to "Configuring Extended-Range VLANs" in the software configuration guide associated with your switch platform and software release.

Describing Trunking Configuration Commands

This topic identifies the commands used to configure trunking.



Commands for configuring a trunk will vary, depending on the operating system version of your switch. The commands shown here are for a Cisco IOS software-based switch.

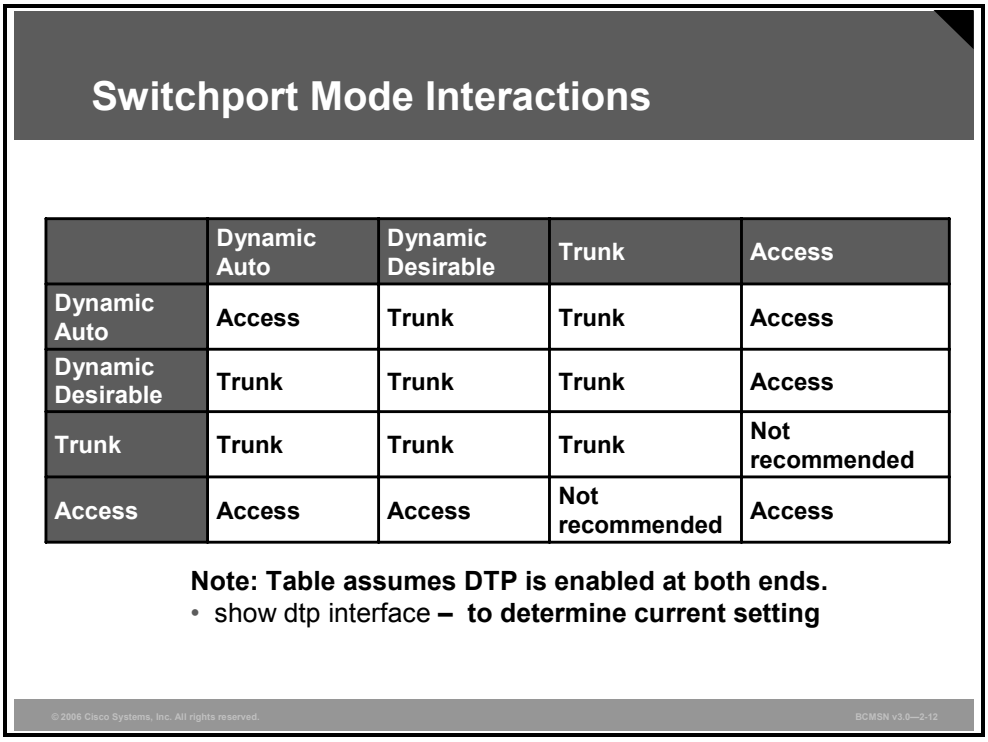
Trunking Commands

The table describes commands for configuring a trunk on a switch that is running Cisco IOS software.

Command	Description
`Switch(config)#` **`interface`** *`number`*	Selects the interface to configure.
`Switch(config-if)#` **`switchport trunk`** [***`allowed`*** *`vlan range or list`*]	Sets the trunk characteristics when the interface is in trunking mode. Use the **no** form of this command to reset all trunking characteristics to the defaults. A range of VIDs to be carried on the trunk can optionally be specified.
`Switch(config-if)#` **`switchport trunk native vlan`**	Sets the native VLAN for sending and receiving untagged traffic when the interface is in 802.1Q trunking mode. Valid IDs are from 1 to 4094 when the Enhanced Image (EI) software is installed, and 1 to 1001 when the Standard Image (SI) software is installed. Do not enter leading zeros.
`Switch (configure-if)#` **`switchport nonegotiate`**	Disables the sending of the Dynamic Trunking Protocol (DTP) packets on the interface.

Command	Description
Switch(config-if)# **switchport mode** {**access** | **dynamic** {**auto** | **desirable**} | **trunk**}	Configures the mode of a port to trunk or access. The **dynamic auto** and **dynamic desirable** options are used to autonegotiate the trunk.
Switch# **show interfaces switchport**	Displays the administrative and operational status of a switching (nonrouting) port.

Identifying the Modes for Trunking



Trunk links should be configured statically whenever possible. However, Cisco Catalyst switch ports run DTP, which can automatically negotiate a trunk link. This Cisco proprietary protocol can determine an operational trunking mode and protocol on a switch port when it is connected to another device that is also capable of dynamic trunk negotiation.

DTP mode can be configured to turn the protocol off or to instruct it to negotiate a trunk link under only certain conditions, as described in the table.

Switchport Modes

Mode	Function
Dynamic Auto	Creates the trunk based on the DTP request from the neighboring switch.
Dynamic Desirable	Communicates to the neighboring switch via DTP that the interface would like to become a trunk if the neighboring switch interface is able to become a trunk.
Trunk	Automatically enables trunking regardless of the state of the neighboring switch and regardless of any DTP requests sent from the neighboring switch.
Access	Trunking is not allowed on this port regardless of the state of the neighboring switch interface and regardless of any DTP requests sent from the neighboring switch.
Nonegotiate	Prevents the interface from generating DTP frames. This command can be used only when the interface switchport mode is access or trunk. You must manually configure the neighboring interface as a trunk interface to establish a trunk link.
Dynamic Auto	Creates the trunk based on the DTP request from the neighboring switch.

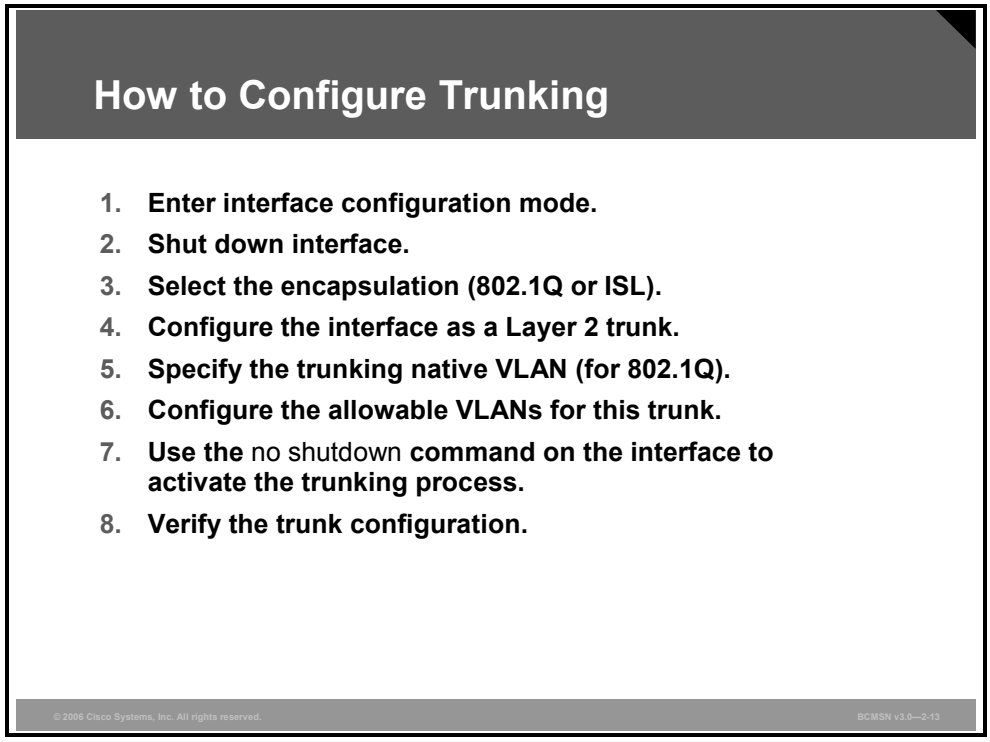
The default DTP mode is Cisco IOS and platform dependent. To determine the current DTP mode, issue the command **show dtp interface**.

```
ASW11#show dtp interface fa0/1
DTP information for FastEthernet0/1:
  TOS/TAS/TNS:                              TRUNK/DESIRABLE/TRUNK
  TOT/TAT/TNT:                              802.1Q/802.1Q/802.1Q
  Neighbor address 1:                       001646FA9B01
  Neighbor address 2:                       000000000000
  Hello timer expiration (sec/state):       17/RUNNING
  Access timer expiration (sec/state):      287/RUNNING
```

Note General best practice is to set the interface to **trunk** and **nonegotiate** when a trunk link is required. On links where trunking is not intended, DTP should be turned off.

Configuring Trunking

This topic explains the procedure to configure trunking.



Switch ports are configured for trunking using Cisco IOS commands. To configure a switch port as an 802.1Q or an ISL trunking port, follow these steps on each trunk interface.

Step 1 Enter interface configuration mode.

Step 2 Shut down the interface to prevent the possibility of premature autoconfiguration.

Step 3 Select the trunking encapsulation. Note that some switches support only ISL *or* 802.1Q.

Step 4 Configure the interface as a Layer 2 trunk.

Step 5 Configure the trunking native VLAN number for 802.1Q links. This number *must* match at both ends of an 802.1Q trunk.

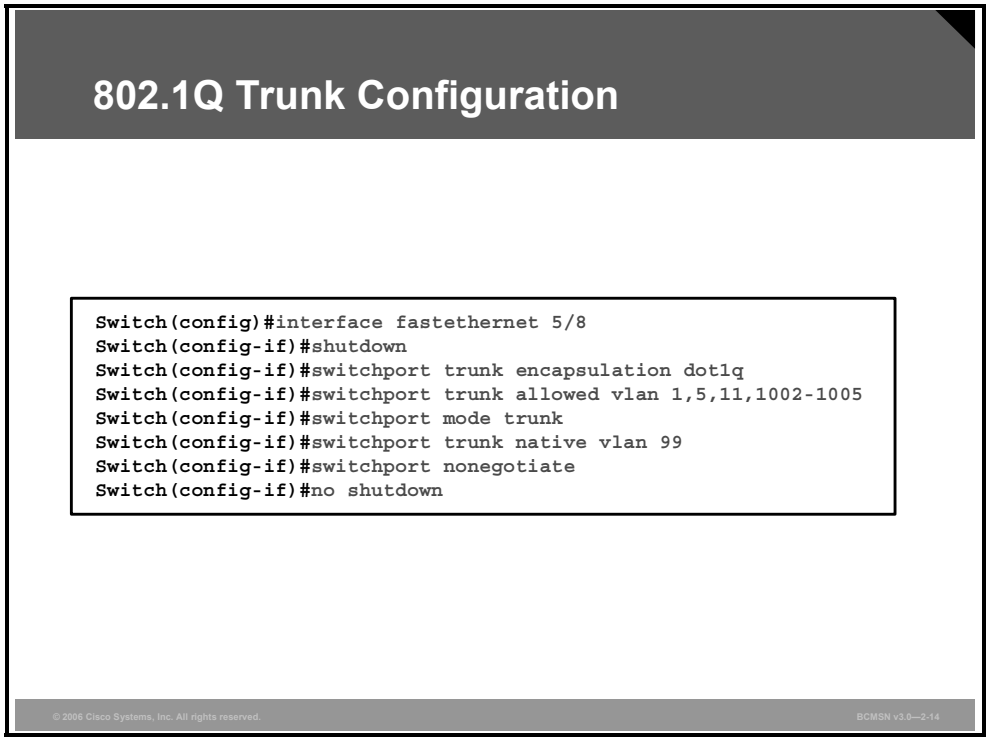
Step 6 Configure the allowable VLANs for this trunk. This is necessary if VLANs are restricted to certain trunk links. This is best practice with the Enterprise Composite Network Model (ECNM) and leads to the correct operation of VLAN interfaces.

Step 7 Use the **no shutdown** command on the interface to activate the trunking process.

Step 8 Verify the trunk configuration using **show** commands.

Configuring an 802.1Q Trunk

This subtopic describes how to configure an 802.1Q trunk link.



The example shows how to configure interface Fast Ethernet 5/8 as an 802.1Q trunk. Frames from VLANs 1, 5, 11, and 1002 to 1005 will be allowed to traverse the trunk link. The switchport mode for the interface is trunk (on), and no DTP messages will be sent on the interface.

Note	For security reasons, the native VLAN has been configured to be an “unused” VLAN. This will be discussed in more detail later.

Configuration of Switch Port as 802.1Q Trunk Link

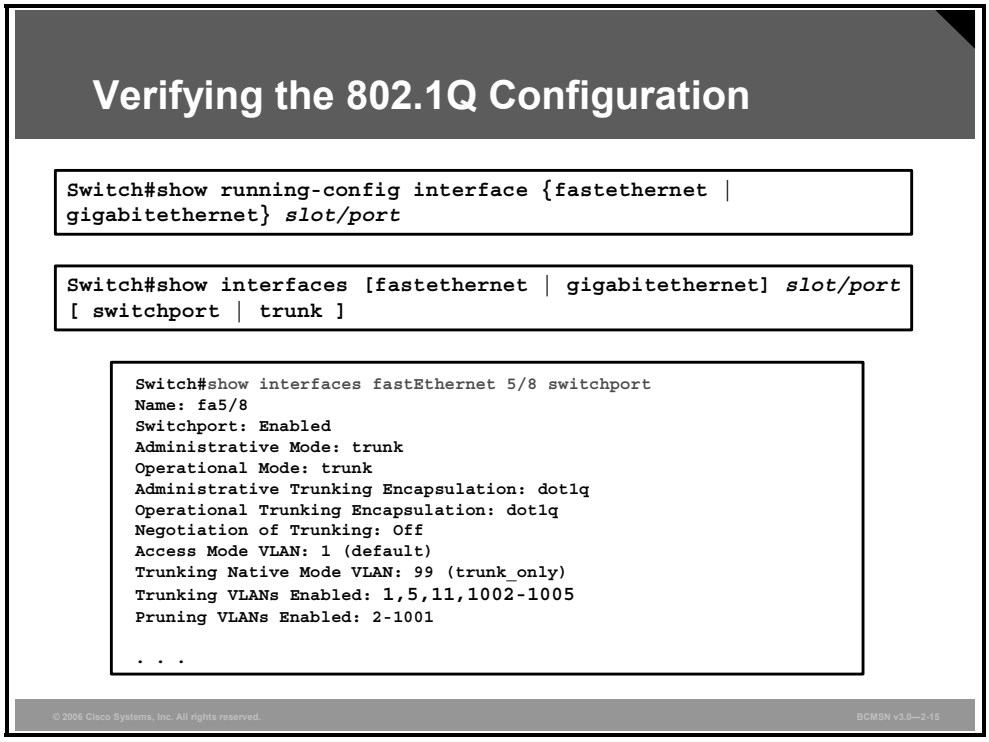
The table describes the commands used to configure a switch port as an 802.1Q trunk link.

Step	Action	Notes
1.	Enter interface configuration mode. `Switch(config)#`**`interface`** {**`fastethernet`** | **`gigabitethernet`**} *`slot/port`*	Select the interface to configure.
2.	Select the encapsulation. `Switch(config-if)#`**`switchport trunk encapsulation`** {**`isl`** | **`dot1q`** | **`negotiate`**}	(Optional) If switchport trunk mode is configured, this command must be used with either the **isl** or the **dot1q** argument; **negotiate** is the default.
3.	Configure the interface as a Layer 2 trunk. `Switch(config-if)#`**`switchport mode`** {**`dynamic`** {**`auto`** | **`desirable`**} | **`trunk`**}	The switchport mode of the interface determines if the link will perform trunking.
4.	Specify the native VLAN. `Switch(config-if)#`**`switchport trunk native vlan`** *`vlan_number`*	The default native VLAN is VLAN1. For security reasons, it should be set to an "unused" VLAN.
5.	Configure the allowable VLANs for this trunk. `Switch(config-if)#`**`switchport trunk allowed vlan`** {**`add`** | **`except`** | **`all`** | **`remove`**} *`vlan_num1`*`[,`*`vlan_num`*`[,`*`vlan_num`*`[,...]]]`	If not specified, all VLANs are allowed on the trunk. VLANs can be specifically allowed or disallowed.

Caution Ensure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If there is a native VLAN mismatch, traffic will not be transmitted correctly on the trunk.

Verify the 802.1Q Configuration

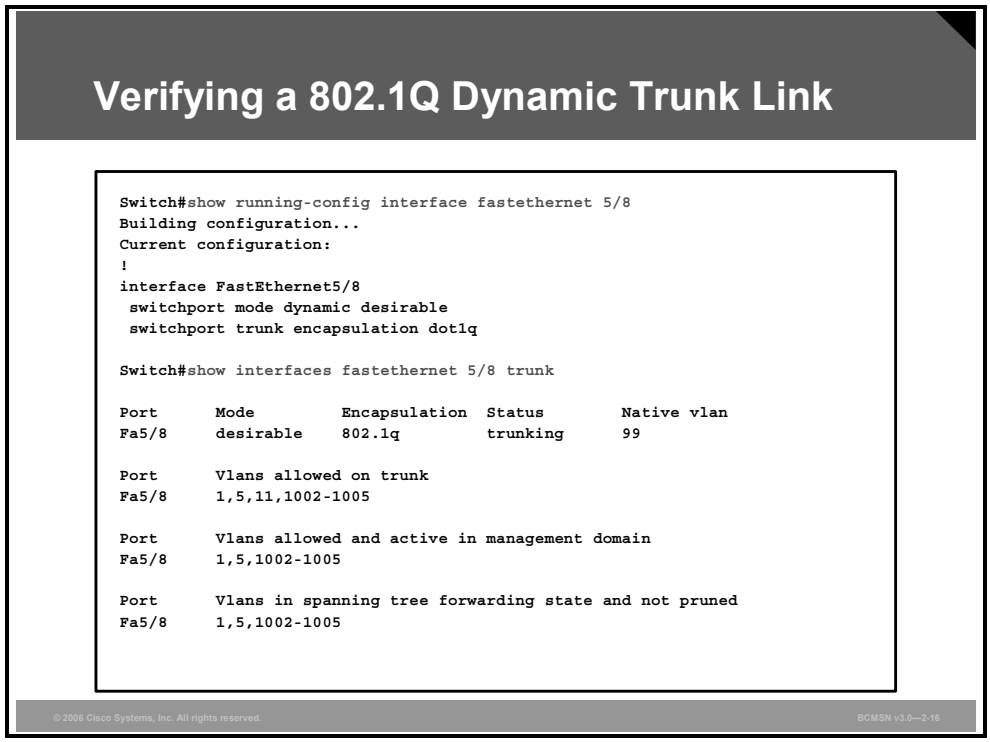
This subtopic describes how to verify an 802.1Q link.



Use **show** commands to display port information, switch port information, or trunking information.

Example: Configure and Display Port Information for an 802.1Q Dynamic Trunk Link

This subtopic describes how to verify an 802.1Q dynamic trunk link.

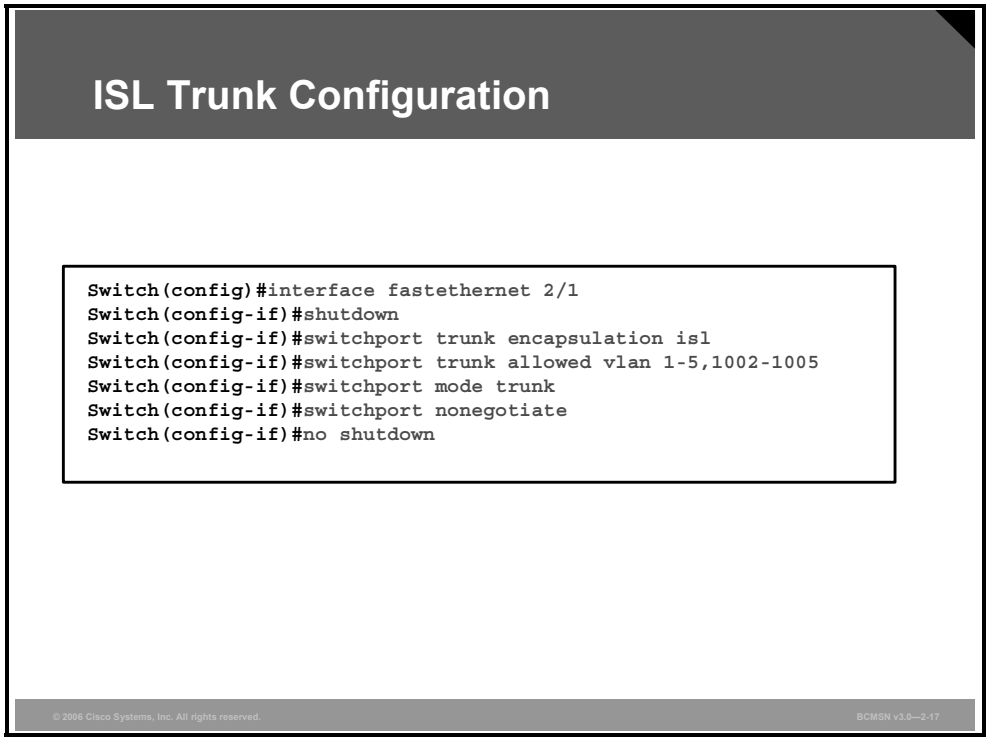


The output in the figure shows that DTP has negotiated with the other switch to enable 802.1Q trunking.

Also note that the native VLAN has been configured to be VLAN99. It is best practice that the native VLAN is not left at the default of VLAN1 and should be an “unused” VLAN. This will be discussed in more detail later.

Configuring an ISL Trunk

This subtopic discusses the configuration of an ISL trunk.



In the example, interface Fast Ethernet 2/1 has been configured as a trunk link for ISL that is permanently on. DTP negotiation is not allowed. The trunk link will carry VLAN traffic for VLANs 1-5 and 1002-1005. VLANs 2-5 are configured on various access ports on the switch, and the trunk links need to carry the frames for these VLANs in addition to the frames for the system VLANs 1002-1005.

Configuration of Switch Port as ISL Trunking Port

The table describes commands for configuring a switch port as an ISL trunking port.

Step	Action	Notes
1.	Enter interface configuration mode. `Switch(config)# interface {fastethernet | gigabitethernet} slot/port`	Select the interface to configure.
2.	Select the encapsulation (if multiple encapsulations are supported). `Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}`	This command is optional, unless you configure the port in switchport trunk mode. In that case, you must use this command with either the *isl* or *dot1q* argument. **Negotiate** is the default. This command is only supported if the switch hardware supports both ISL and dot-1Q encapsulation.
3.	Configure the allowable VLANs for this trunk. `Switch(config-if)#switchport trunk allowed vlan {add | except | all | remove} vlan_num1[,vlan_num[,vlan_num[,...]]]`	If not specified, all VLANs are allowed on the trunk. VLANS can be specifically allowed or disallowed.

Step	Action	Notes
4.	Configure the interface as a Layer 2 trunk. `Switch(config-if)# switchport mode {dynamic {auto | desirable} | trunk}`	The switchport mode of the directly connected interface helps determine if the link will perform trunking.

Note It is best practice to shut down an interface while configuring trunking attributes so that premature autonegotiation cannot occur.

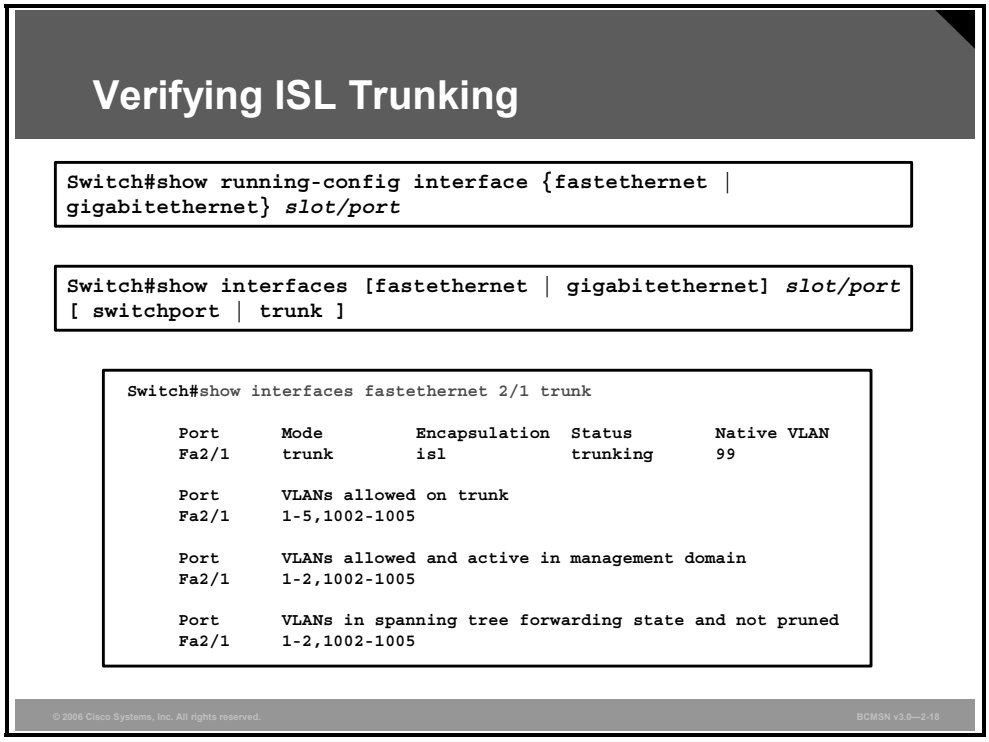
Configuring a Port for ISL Trunking with No DTP

When configuring the Layer 2 trunk to not use DTP, this syntax is used so that the trunk mode is set to "on," and no DTP messages are sent on the interface.

- First, enter the **shutdown** command in the interface mode.
- Enter the **switchport trunk encapsulation** command.
- Enter the **switchport mode trunk** command.
- Enter the **switchport nonegotiate** command.
- Finally, enter the **no shutdown** command.

Verifying the ISL Trunk Configuration

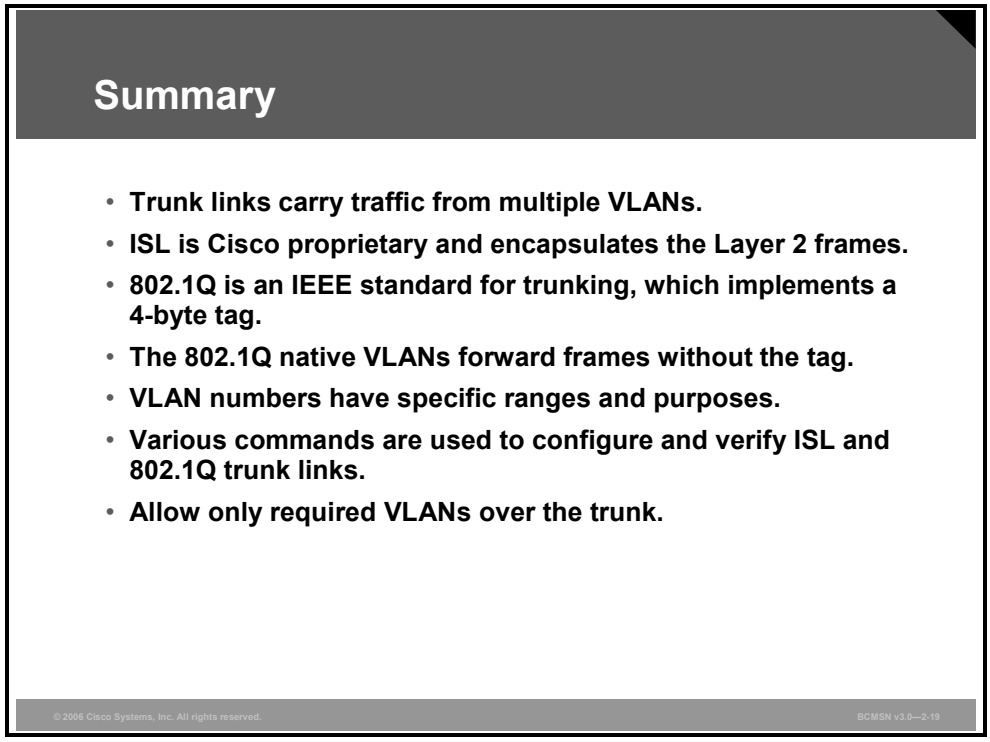
This subtopic describes how to verify an ISL trunk link.



Use **show** commands to display port information, switch port information, or trunking information.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 4

Propagating VLAN Configurations with VTP

Overview

When VLANs span multiple switches, a protocol is needed to accurately manage VLAN information at each switch. This protocol is referred to as VLAN Trunk Protocol (VTP) and is used to ensure that all switches in a given group, or VTP domain, have the same information about the VLANs that are present in that domain. This lesson will examine VTP and how it allows each switch to participate in the VTP domain. The VTP mode determines if and when updates are sent by a switch.

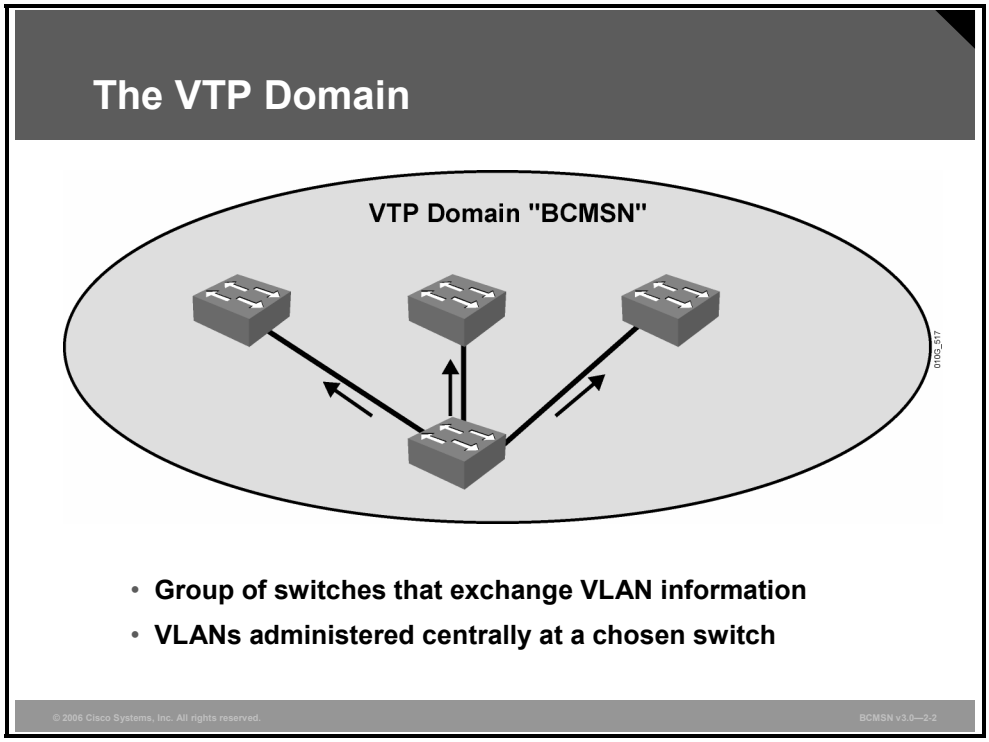
Objectives

Upon completing this lesson, you will be able to describe how VLAN configuration of switches in a single management domain can be automated with the Cisco Systems proprietary VTP. This ability includes being able to meet these objectives:

- Define a VTP domain in a campus network
- Define VTP
- Describe the three different VTP modes
- Describe VTP Pruning
- Describe how VTP distributes and synchronizes VLAN information
- Describe the commands used to configure and verify a VTP management domain
- Describe the procedures to configure a VTP management domain
- Describe the procedure to add a new switch to an existing VTP domain

Explaining VTP Domains

This topic defines a VTP domain in a campus network.



In an enterprise network with many interconnected switches, maintaining a consistent list of VLANs across those switches can be administratively cumbersome and potentially error prone. The VTP is designed to automate this administrative task.

Switches that share common VLAN information are organized into logical groups called VTP management domains. The VLAN information within a VTP domain is propagated through trunk links and is updated via the VTP, allowing all switches within a particular domain to maintain identical VLAN databases.

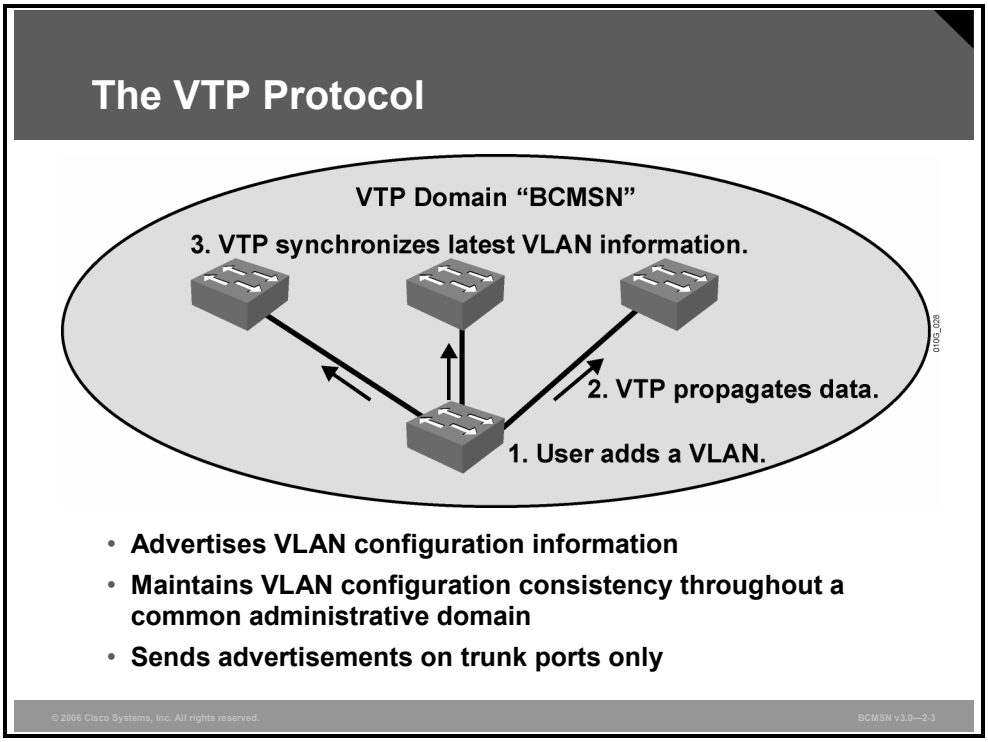
Only “global” VLAN information regarding VLAN number, name, and description is exchanged. Information on how ports are assigned to VLANs on a given switch is kept local to the switch and is not part of a VTP advertisement.

These are the attributes of a VTP domain:

- A switch may be in only one VTP domain.
- A VTP domain may be as small as only one switch.
- VTP updates will be exchanged only with other switches in the same domain.
- The way VLAN information is exchanged between switches in the same domain depends upon the VTP mode of the switch.
- By default, a Cisco Catalyst switch is in the no-management-domain state until it receives an advertisement for a domain over a trunk link, or until a management domain is configured.

Describing the VTP

This topic defines the VTP.



Switches in a single VTP domain exchange VTP updates to distribute and synchronize VLAN information. VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the additions, deletions, and name changes of VLANs on all switches in a VTP domain.

VTP runs over trunk links allowing interconnected switches to exchange Layer 2 frames, synchronizing a single list of configured VLANs. This reduces the manual configuration required at each switch; VLANs can be created on one switch and then propagated to others.

These are the attributes of VTP:

- VTP is a Cisco proprietary protocol.
- VTP will advertise VLANs 1–1005 only.
- VTP updates are exchanged only across trunk links.
- Each switch operates in a given VTP mode that determines how VTP updates are sent from and received by that switch.

VTP Versions

Currently, Cisco Catalyst switches run VTP versions 1, 2, or 3. Version 2 is the most prevalent, although within version 2 the default operating mode of the switch is version 1.

Version 2 provides these features:

- Support for Token Ring switches
- Consistency checks on new VTP and VLAN configuration parameters
- Propagation of VTP updates that have an unrecognized type, length, or value
- Forwarding of VTP updates from transparent mode switches without checking the version number

VTP version 3 is now available on some switches that use the Cisco Catalyst software operating system version.

When enabled, VTP version 3 provides these enhancements to previous VTP versions:

- Support for extended VLANs
- Support for the creation and advertising of private VLANs
- Support for VLAN instances and Multiple Spanning Tree (MST) mapping propagation instances
- Improved server authentication
- Protection from the wrong database accidentally being inserted into a VTP domain
- Interaction with VTP version 1 and VTP version 2
- Ability to be configured on a per-port basis

Caution	VTP version 1 and VTP version 2 are not interoperable on switches in the same VTP domain. Every switch in the VTP domain must use the same VTP version. Do not enable VTP version 2 unless every switch in the VTP domain supports version 2.

VTP in the Campus Infrastructure Module

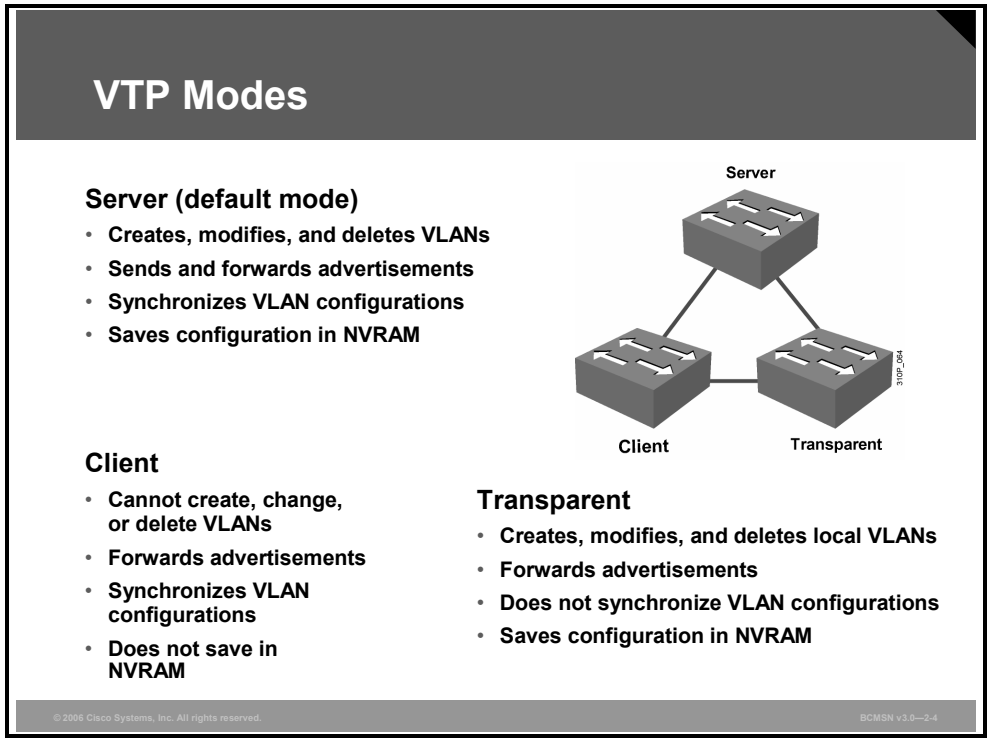
There are some guidelines to using VTP within the Campus Infrastructure module.

- The VTP domain is restricted to building switch blocks.
- VTP keeps VLAN information consistent between Building Distribution layer and Building Access layer switches.
- VLAN configuration errors or failures will be confined to the distribution and access layer switch blocks.
- Knowledge of all VLANs does not need to exist on all switches within the Campus Infrastructure module. Use of VTP is optional, and in high-availability environments it is best practice to set all switches to ignore VTP updates.

Caution	VLANs deleted on one switch may be deleted on all switches in the VTP domain, and thus all ports removed from that VLAN. Delete VLANs with caution on a switch that is participating in a VTP domain with other switches.

VTP Modes

This topic describes the three modes in which VTP operates.



On each switch, VTP can be configured to operate in one of three modes: server, client, or transparent. The default VTP mode is server. The mode will determine if VLANs can be created on the switch and how the switch will participate in sending and receiving VTP advertisements. The number of VLANs that can be configured on a switch will vary by mode.

VTP Mode Features

The table describes the features of the VTP client, server, and transparent modes.

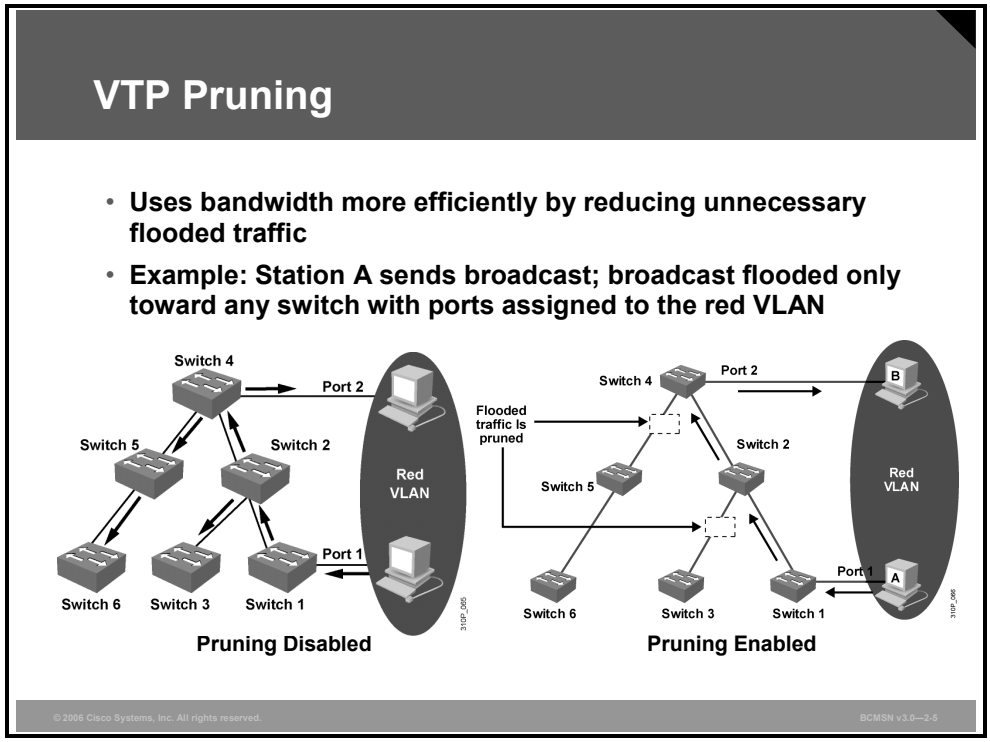
VTP Mode	Features
Server	■ Creates, modifies, and deletes VLANs at the command-line interface (CLI) ■ Generates VTP advertisements and forwards advertisements from other switches in the same management domain ■ May update its own VLAN database with information received from other servers in the management domain ■ Saves VLAN configuration information in “vlan.dat” file in Flash memory
Client	■ Cannot create, modify, or delete VLANs at the CLI ■ Forwards VTP advertisements received ■ Synchronizes its own VLAN database with latest information received from VTP servers in the management domain ■ VLAN information in RAM only, not stored in NVRAM or Flash memory; must be repopulated from a VTP server if switch is powered cycled

VTP Mode	Features
Transparent	■ Creates, modifies, and deletes VLANs for the VLAN database on the local switch only ■ Does not generate VTP advertisements ■ Does not update its VLAN database with information received from VTP servers in the same management domain ■ Forwards VTP advertisements received from VTP servers in the same VTP domain ■ Always has a configuration revision number of 0 ■ Saves VLAN configuration in NVRAM

Caution Before adding a VTP client or server to a VTP domain, always verify that its VTP configuration revision number is lower than the configuration revision number of the other switches in the VTP domain. Switches in a VTP domain always use the VLAN configuration of the switch with the highest VTP configuration revision number. If you add a switch in server or client mode that has a revision number that is higher than the revision number in the VTP domain, it can erase all VLAN information from the VTP server and VTP domain. To reset the VTP revision number on the switch that is being added, either modify the VTP domain name or set the VTP mode to transparent.

Describing VTP Pruning

This topic describes VTP Pruning.



VTP Pruning uses VLAN advertisements to determine when a trunk connection is flooding traffic needlessly.

By default, a trunk connection carries traffic for all VLANs in the VTP management domain. Commonly, some switches in an enterprise network do not have local ports configured in each VLAN. In the example, only switches 1 and 4 support ports that are statically configured in the red VLAN.

VTP Pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices.VLAN1 is always ineligible for pruning; traffic from VLAN1 cannot be pruned.

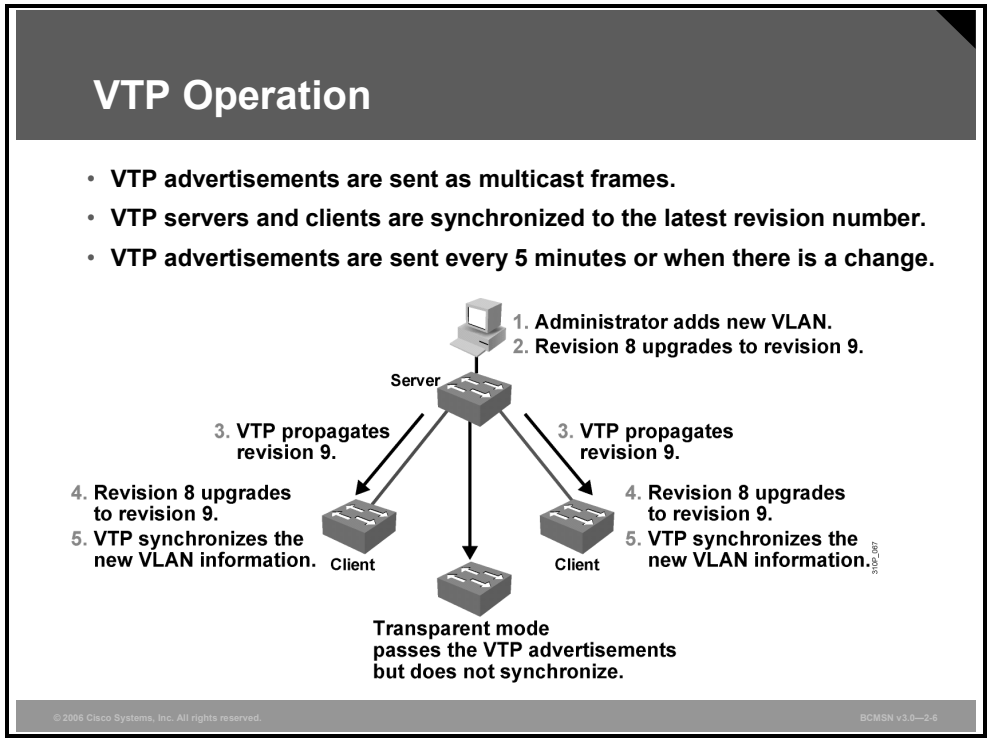
The figure shows, on the left, a switched network without VTP Pruning enabled. Port 1 on switch 1 and port 2 on switch 4 are assigned to the red VLAN. A broadcast is sent from the host connected to switch 1. Switch 1 floods the broadcast, and every network device in the network receives it, even though switches 3, 5, and 6 have no ports in the red VLAN. With VTP Pruning enabled, the broadcast traffic from station A is not forwarded to switches 3, 5, and 6 because traffic for the red VLAN has been pruned on the links indicated on switches 2 and 4.

Note	You can implement VTP Pruning on only VTP servers, not on clients. Consider VTP Pruning support to minimize traffic on trunk links.

Note	A switch runs an instance of spanning tree for each VLAN that it is aware of, even if no ports are active or if VTP Pruning is enabled. VTP Pruning prevents unnecessary flooded traffic but does not eliminate the switch knowledge of pruned VLANs.

Describing VTP Operation

This topic describes how VTP distributes and synchronizes VLAN information.



Switches within a VTP management domain synchronize their VLAN databases by sending and receiving VTP advertisements over trunk links. VTP advertisements are flooded throughout a management domain by switches that are running in specific modes of operation.

Advertisements are sent every 5 minutes or whenever there is a change in VLAN configuration. VTP advertisements are transmitted over VLAN1, using a Layer 2 multicast frame. VLAN advertisements are not propagated from a switch until a management domain name is specified or learned.

VLAN Synchronization Over VTP

The table shows the general order of VLAN synchronization over VTP.

Step	Action
1	Configure the VTP domain, VTP mode, and VTP password (optional) on each switch. This configuration proactively determines which switches will send updates.
2.	Switches running VTP server mode then send VTP updates across trunk links.
3.	A device that receives a VTP advertisement will check that the VTP management domain name and password in the advertisement match those configured in the local switch.
4.	If a match is found, a switch further inspects the VTP update to see the configuration revision number.
5.	If the configuration revision number of the message is greater than the number currently in use, and the switch is running in VTP server or client mode, the switch overwrites its current VLAN information with that in the received update.
6.	The switch may also request more information.

Configuration Revision Number

One of the most critical components of VTP is the configuration revision number. When initially configured, the VTP configuration revision number is set to 0. Each time a VTP server modifies its VLAN information, it increments the VTP configuration revision number by one. It then sends out a VTP advertisement referencing the new configuration revision number. If the configuration revision number being advertised is higher than the number stored on other switches in the VTP domain, they will overwrite their VLAN configurations with the new information.

Caution Given this overwrite process, if all the VLANs on a VTP server are deleted, the VTP server will then send an advertisement with a higher revision number; the receiving devices in the VTP domain will accept the advertisement and delete those VLANs also.

VTP Advertisement Types

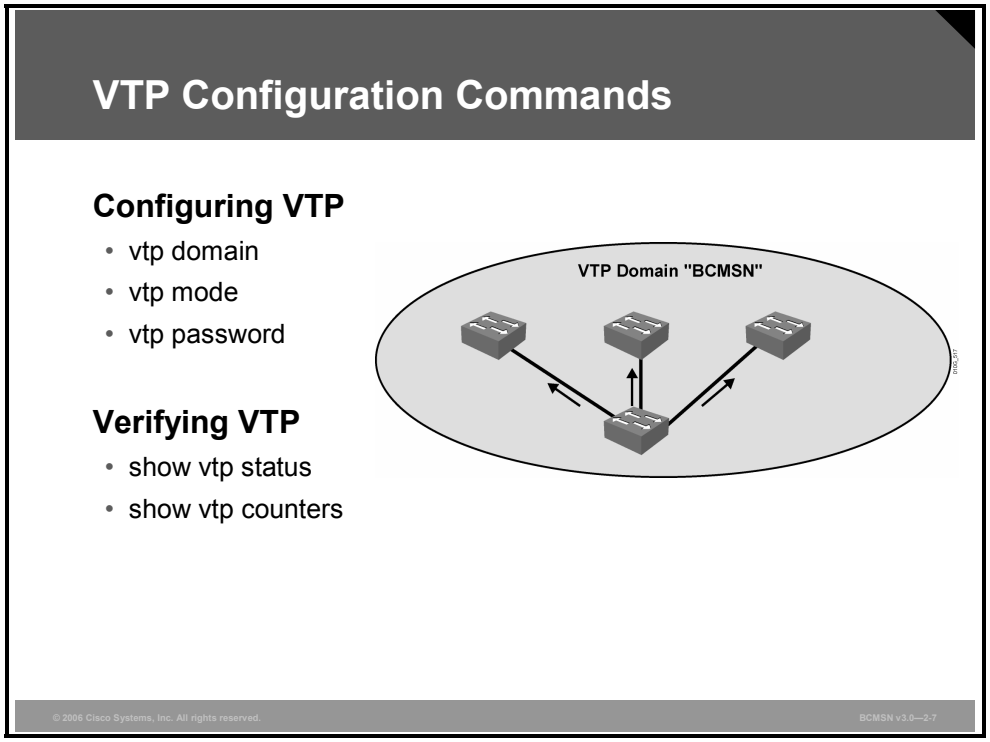
Three types of VTP advertisements are exchanged between switches.

- **Summary advertisements:** An update sent by VTP servers every 300 seconds or when a VLAN database change occurs. Among other things, this advertisement lists the management domain, VTP version, domain name, configuration revision number, time stamp, and number of subset advertisements. If the advertisement results from a VLAN database change, one or more subset advertisements will follow.
- **Subset advertisements:** An update that follows a summary advertisement resulting from a change in the VLAN database. A subset advertisement cites the specific change that was made to a specific VLAN entry in the VLAN database. One subset advertisement will be sent for each VLAN ID (VID) that encountered a change.
- **Advertisement requests from clients:** An update sent by a switch requesting information to update its VLAN database. If a client hears a VTP summary advertisement with a configuration revision number higher than its own, the switch may send an advertisement request. A switch operating in VTP server mode then responds with summary and subset advertisements.

Note VTP advertisements are associated with VLAN database information only, not with VLAN information configured on specific switch ports. Likewise, on a receiving switch, the receipt of new VLAN information does not change the VLAN associations of trunk or access ports on that switch.

Describing VTP Configuration Commands

This topic describes the commands used to configure and verify a VTP management domain.



The **vtp** configuration command is used to configure VTP characteristics for a switch. All switches in the same VTP domain will share the same VTP domain name and VTP password, if one is configured. It is a good idea to set the VTP mode to "client" if switches are being added to an existing switched network.

The **show VTP** commands are used to verify the current VTP parameter values.

VTP Commands

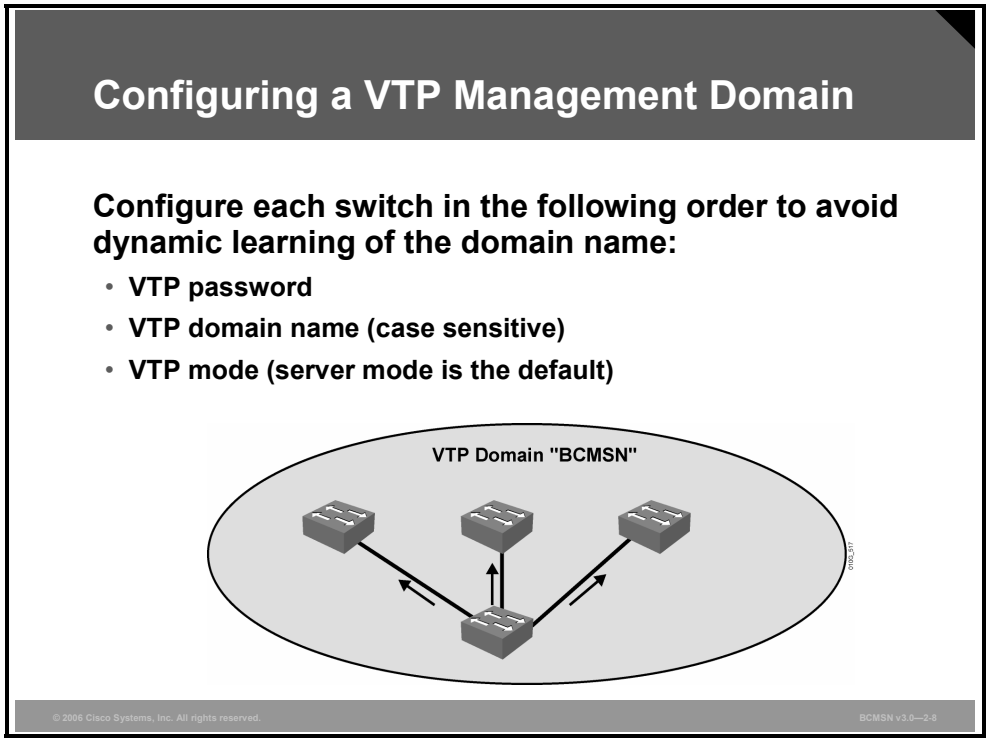
The table describes the commands that are used to configure VTP.

Command	Description
`Switch(config)#` **`vtp domain`** ***`domain_name`***	Sets the VTP domain name. Enter an ASCII string from 1 to 32 characters to identify the VTP administrative domain for the switch. The domain name is case sensitive.
`Switch(config)#` **`vtp password`** ***`password`***	The 16-byte secret value used in Message Digest 5 (MD5) digest calculation to be sent in VTP advertisements and to validate received VTP advertisements. The password can be an ASCII string from 1 to 32 characters and is case sensitive.
`Switch(config)#` **`vtp v2-mode`**	Enables VTP version 2 in the administrative domain.

Command	Description
Switch(config)# **vtp mode client**	Places the switch in VTP client mode. VLANs cannot be configured on a switch that is configured in this mode. When a VTP client starts up, it does not send VTP advertisements until it receives advertisements to initialize its VLAN database.
Switch(config)# **vtp mode server**	Places the switch in VTP server mode. The switch is enabled for VTP and sends advertisements. VLANs can be configured on the switch. The switch can recover all the VLAN information in the current VTP database from nonvolatile storage after reboot.
Switch(config)# **vtp mode transparent**	Places the switch in transparent mode. It cannot affect VLAN configurations on other devices in the network. The switch receives VTP advertisements and forwards them on all trunk ports except the one on which the advertisement was received.
Switch# **show vtp status**	Verifies the VTP name, mode, revision number, and other information.
Switch# **show vtp counters**	Indicates if VTP updates are being sent and received by the switch.
Switch# **show vlan**	Displays the parameters for all configured VLANs. On a client, it will indicate if VTP updates are being received by the switch.

Configuring a VTP Management Domain

This topic describes the procedure to configure a VTP management domain.



Default VTP configuration values depend on the switch model and the software version. The default values for the Cisco Catalyst 2900, 4000, and 6000 Series switches are as follows:

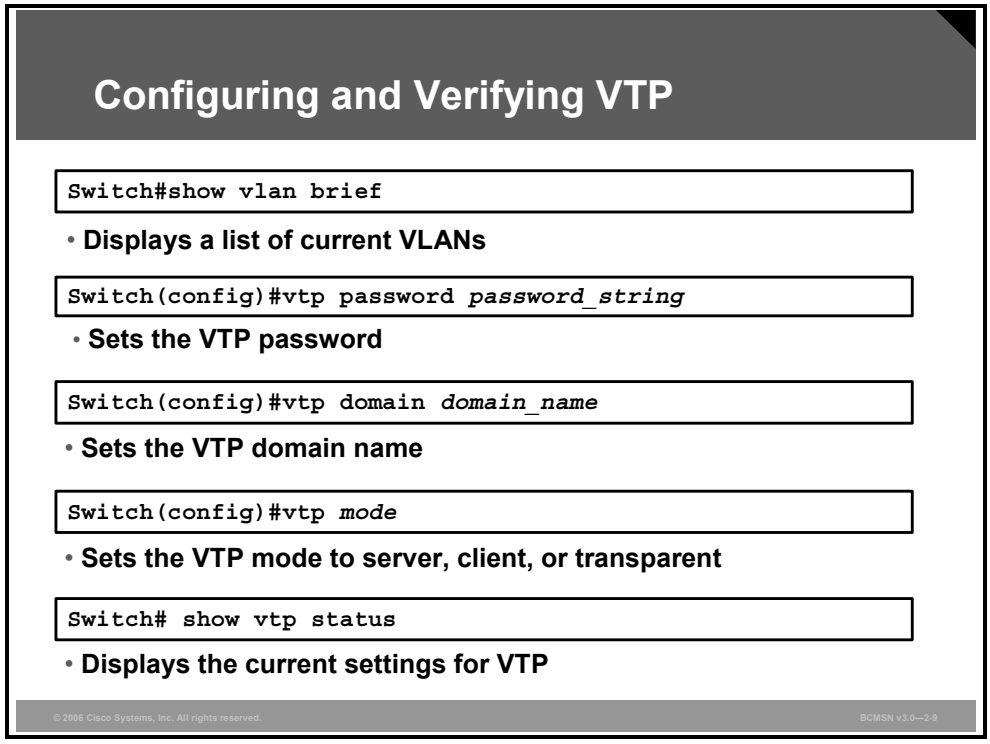
- **VTP domain name**: None
- **VTP mode**: Server
- **VTP password**: None
- **VTP trap**: Disabled (Simple Network Management Protocol [SNMP] traps communicating VTP status)

The VTP domain name can be specified or learned from VTP updates that are seen from other switches. By default, the domain name is not set.

A password can be set for the VTP management domain. The password must be the same for all switches in the domain for the VLAN database to be synchronized among switches.

Configuring VTP on a Switch

This subtopic lists the steps used to configure VTP.



The steps for configuring VTP will vary per design and switch mode, but the general steps for configuring a switch are as follows:

Step 1 Establish a design specifying what switches will be server, client, or transparent, and what the boundaries are for the VTP domain.

Step 2 Verify the current VLAN information on any switch that will be configured as server.

Step 3 Specify the VTP password (optional).

Step 4 Specify the version number, if other than default.

Step 5 Specify the VTP domain name (case sensitive).

Step 6 Configure the VTP mode.

Step 7 Verify the configuration.

Step 8 Verify that updates are being sent from or received by the switch as intended.

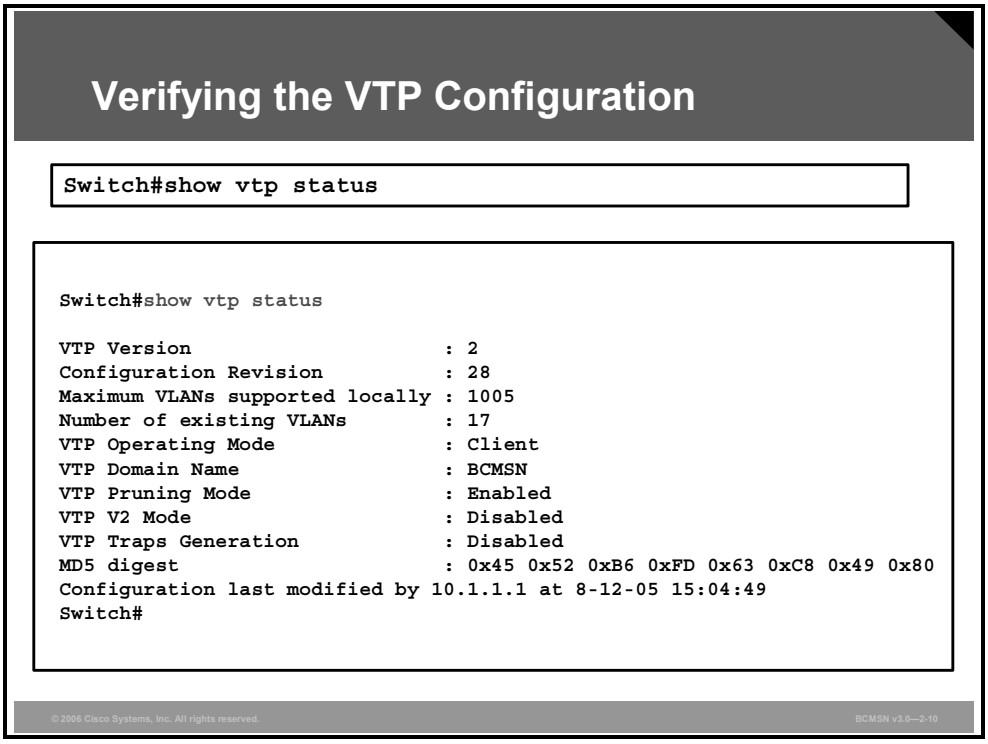
VTP Configuration Commands for a Switch

The table describes the commands used to configure a switch to become part of a VTP domain. Follow these steps from privileged EXEC mode.

Step	Action	Notes
1	Display list of VLANS. `Switch# show vlan brief`	Determine if the list of VLANs displays before configuration. If this switch is about to be configured as a server, this list will overwrite VLAN information on client switches.
2.	Enter global configuration mode. `Switch# configure terminal`	
3.	Specify a VTP password. `Switch(config)#vtp password password_string`	Sets a password for the VTP domain, which can be from 1 to 34 characters and is case sensitive. The password can be set only on switches operating in server or client mode. Use **no vtp password** to clear the password.
4.	Configure the domain name. `Switch(config)#vtp domain domain_name`	Defines the VTP domain name, which can be up to 32 characters long. Domain name case must match other switches in the domain for updates to occur properly.
5.	Enable VTP version 2. `Switch(config)#vtp v2-mode`	To revert to VTP version 1, enter **vtp v1-mode.**
6.	Configure the VTP mode. `Switch(config)#vtp mode mode`	Enter server, client, or transparent. To revert to the default (server), enter **no vtp mode.**
7.	Exit global configuration mode. `Switch(config)#exit`	
8	Display list of VLANS. `Switch# show vlan brief`	Determine if the list of VLANs shown is as anticipated, given the mode of the switch.

Verifying the VTP Configuration

This subtopic identifies the state of the VTP configuration using the output to the **show vtp status** command.

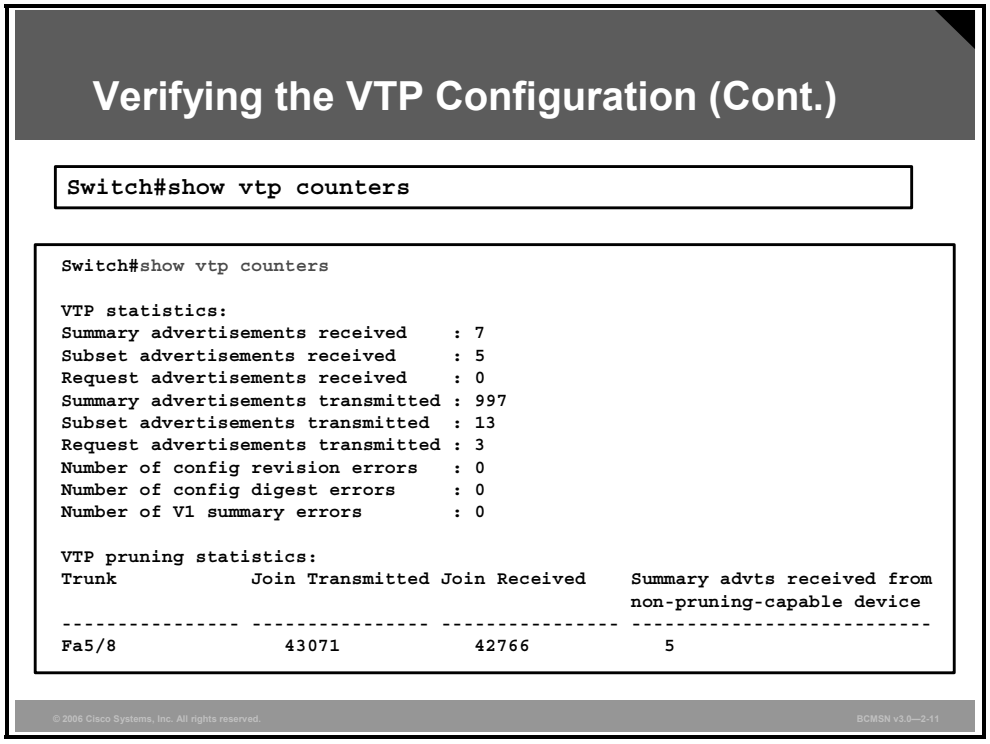


Use the **show vtp status** command to verify the VTP configuration.

When initially configuring switches in a VTP domain, pay close attention to the configuration revision number. Check to see that it increments only when changes are made at intended VTP servers.

In the figure, **Configuration last modified by 10.1.1.1** specifies the IP address of the switch that last updated the VLAN database of this switch.

Note	In this example, VTP version 2 is available (as shown by the “VTP Version” line of the output) but not enabled (as shown by the “VTP V2 Mode” line of the output).



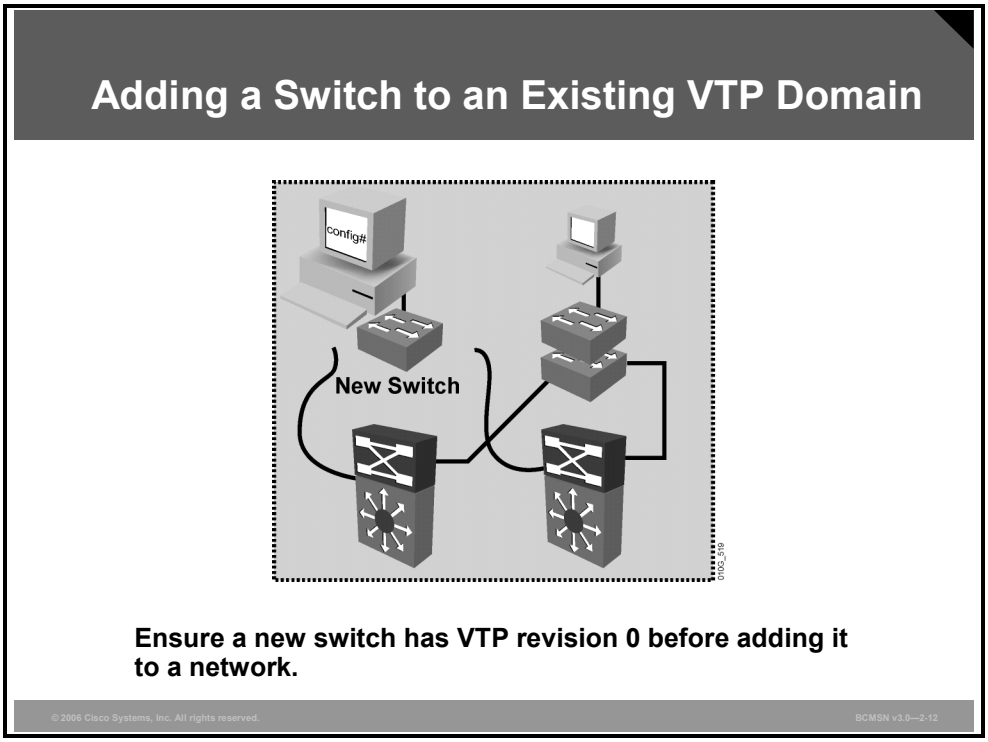
VTP Counters

Use the **show vtp counters** command to display statistics about VTP operation.

Output from this command verifies if VTP updates are being sent and are being received by the switch, and records the number of updates that have been seen.

Adding New Switches to an Existing VTP Domain

This topic describes the procedure to add a new switch to an existing VTP domain.



The configuration revision number is used when determining if a switch should keep its existing VLAN database or overwrite it with the VTP update sent by another switch in the same domain with the same password. Therefore, when a switch is added to a network, it is important that it does not inject spurious information into the domain.

Caution	This overwrite occurs whether the switch is a VTP client or a VTP server. A VTP client can erase VLAN information on a VTP server. One indication that information has been erased is when many of the ports in the network go into inactive state because the ports are now assigned to a nonexistent VLAN.

Note	An example of a VTP client overwriting a VTP server will be shown later.

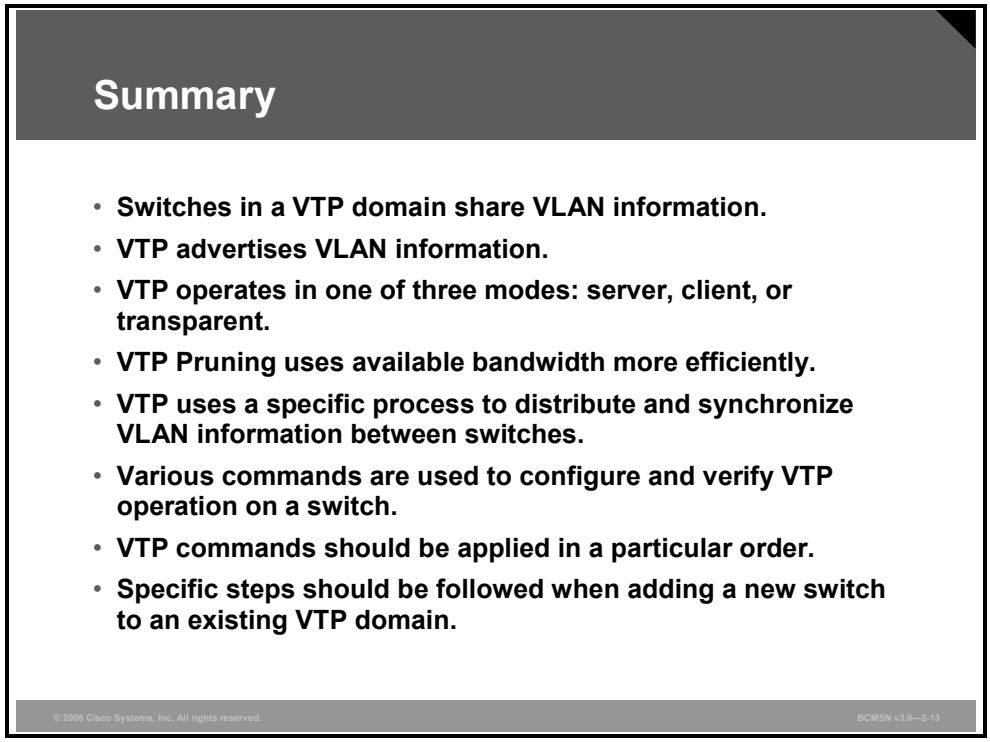
Adding a New Switch to an Existing Network

The table describes the procedure for adding a new switch to a network. It is critical to VLAN stability to add a switch in this manner.

Step	Action	Notes
1.	Ensure that there is no connectivity between the new switch and the network, and power the switch on.	This keeps updates from this switch from overwriting the VLAN databases of other switches in the domain before the switch is configured properly.
2.	Change the switch VTP mode to transparent.	This will set the configuration revision number to 0 and ensure that no updates are received if the switch is connected to the network out of sequence in subsequent steps.
3.	Delete vlan.dat.	This will remove any VLAN information from the switch.
4.	Change the VTP domain name to something unconventional, and change the mode to client.	This will keep the switch from dynamically learning the domain name upon reload.
5.	Reload or power cycle the switch.	This will remove all VLAN information from RAM.
6.	Verify the switch VTP and VLAN configuration.	Use **show** commands to verify that the switch is in client mode and that it is at configuration revision 0.
7.	Configure the switch with a valid VTP domain name and password.	Switch is configured to participate in the network as intended.
8.	Connect the switch to the network.	Switch should receive accurate, current VLAN information.
9.	Verify VLAN database.	Use **show** commands to verify that the switch has received VLAN information as intended.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 5

Correcting Common VLAN Configuration Errors

Overview

When VLANs span multiple switches, there are configuration challenges and issues to be overcome. VLAN configuration problems include security issues related to the 802.1Q native VLAN and Dynamic Trunking Protocol (DTP).

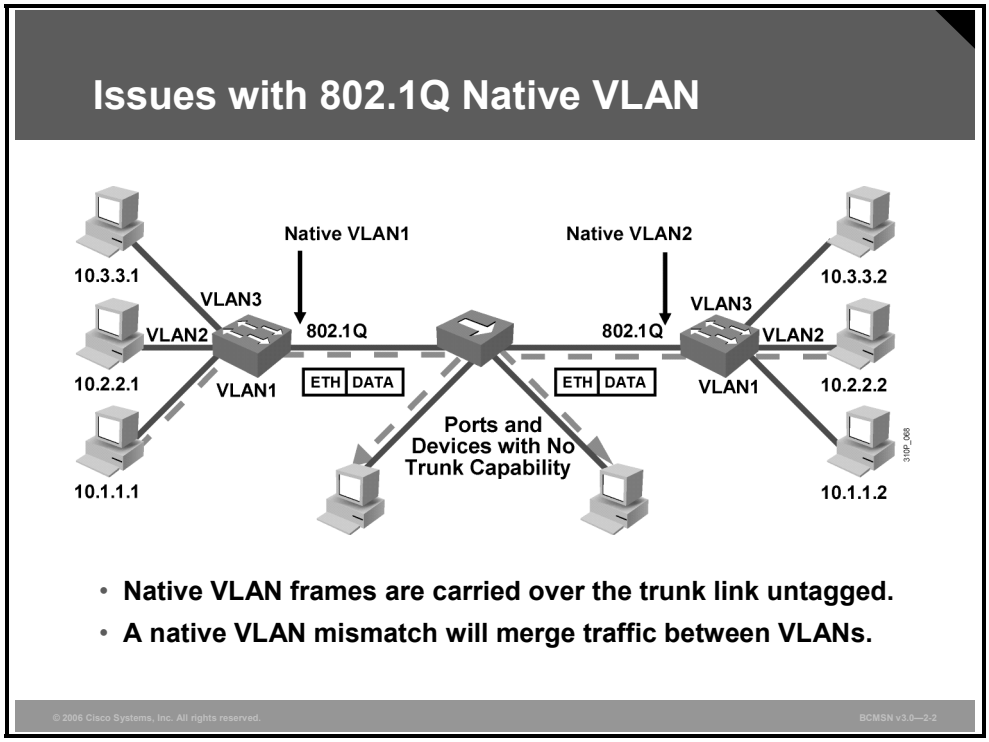
Objectives

Upon completing this lesson, you will be able to identify common VLAN configuration errors and explain the solutions to those errors. This ability includes being able to meet these objectives:

- Identify the security issues with 802.1Q native VLANs
- Describe how to resolve the security issues with 802.1Q native VLANs
- List key problems that result from trunk link configuration
- Identify best practices for resolving trunk link problems
- Identify common problems with VTP configuration
- Describe best practice for VTP configuration

Describing Issues with 802.1Q Native VLANs

This topic describes the security issues with 802.1Q native VLANs.



Different Native VLANs

This is a frequent configuration error. The native VLAN that is configured on each end of an 802.1Q trunk must be the same. Remember that a switch receiving an untagged frame will assign it to the native VLAN of the trunk. If one end is configured for native VLAN1 and the other for native VLAN2, a frame sent in VLAN1 on one side will be received on VLAN2 on the other. VLAN1 and VLAN2 have been segmented and merged. There is no reason this should be required, and connectivity issues will occur in the network.

Cisco Systems switches use Cisco Discovery Protocol (CDP) to warn of a native VLAN mismatch.

Untagged Frames

In the example, the PCs connected to the hub will send untagged frames. Because the frames are untagged, on the switch on the left, they will become part of VLAN1; on the switch on the right, they will become part of VLAN2.

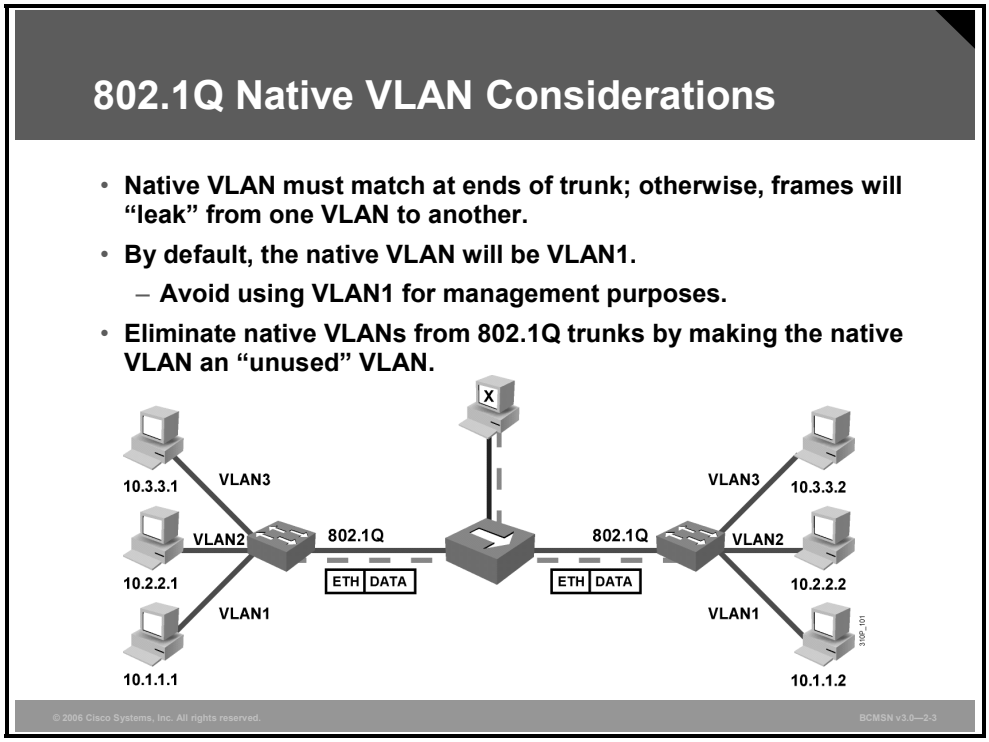
Issues and Mitigation of 802.1Q Native VLANs

The table describes the issue with and mitigation of 802.1Q native VLANs.

Issue	Description	Mitigation
Different native VLANs	Untagged frames from different native VLANs will merge between the two native VLANs.	Ensure that if the native VLAN is being used, it matches at both ends of the trunk.
Untagged frames	Any untagged frames that an 802.1Q trunk receives will be forwarded to any ports in the native VLAN, which could be a security issue.	This issue can be avoided by assigning an unused VLAN number to the native VLAN so that any untagged frames that an 802.1Q trunk receives will not be forwarded to any user ports.

Resolving Issues with 802.1Q Native VLANs

This topic describes how to resolve the security issues with 802.1Q native VLANs.



Consider these issues when you are configuring the native VLAN on an 802.1Q trunk link:

- The native VLAN interface configurations must match at both ends of the link or the trunk may not form.
- By default, the native VLAN will be VLAN1. For the purpose of security, the native VLAN on a trunk should be set to a specific VLAN ID (VID) that is not used for normal operations elsewhere on the network.

```
Switch(config-if)#switchport trunk native vlan vlan-id
```

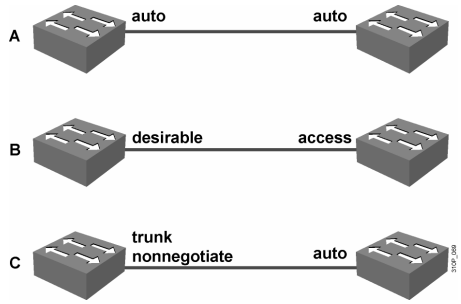
- If there is a native VLAN mismatch on an 802.1Q link, CDP (if used and functioning) will issue a "native VLAN mismatch" error.
- On select versions of Cisco IOS software, CDP may not be transmitted or will be automatically turned off if VLAN1 is disabled on the trunk.
- If there is a native VLAN mismatch on either side of an 802.1Q link, Layer 2 loops may occur because VLAN1 Spanning Tree Protocol (STP) bridge protocol data units (BPDUs) are sent to the IEEE STP MAC address (0180.c200.0000) untagged.
- When troubleshooting VLANs, note that a link can have one native VLAN association when in access mode and another native VLAN association when in trunk mode.

Describing Trunk Link Problems

This topic describes the key problems that can result from trunk link configuration.

Explaining Trunk Link Problems

- **Trunks can be configured statically or autonegotiated with DTP.**
- **For trunking to be autonegotiated, the switches must be in the same VTP domain.**
- **Some trunk configuration combinations will successfully configure a trunk, some will not.**



- **Will any of the above combinations result in an operational trunk?**

These elements determine whether or not an operational trunk link is formed and also determine the type of trunk the link becomes: the trunking mode, the trunk encapsulation type, the VLAN Trunk Protocol (VTP) domain, and the hardware capabilities of two connected ports.

Notice that with the default switchport mode set to **dynamic auto** and with DTP enabled, if another switch is connected and is set to **switchport mode trunk**, the switch will automatically convert the link to a trunk. This could have security implications because it might start accepting traffic destined for any VLAN. Therefore, a malicious user could start communicating with other VLANs through that compromised port.

Note To see the different DTP messages and recommended actions, see http://www.cisco.com/univercd/cc/td/doc/product/lan/cat2950/12111yj/2950smg/msg_desc.htm#xtocid3

Examples of Trunk Negotiation

An explanation follows for the three different examples shown in the figure.

Example A

If both ends of the link are set to **switchport mode auto**, the link will not become a trunk. The ports will remain as access ports.

```
1ASW3#show interface fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
```

Example B

If one end of the link is set **to switchport mode dynamic desirable** and the other end of the link is set to **switchport mode access**, both ports will remain as access ports.

```
1ASW3#show interface fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On

1DSW1#show interfaces g1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
```

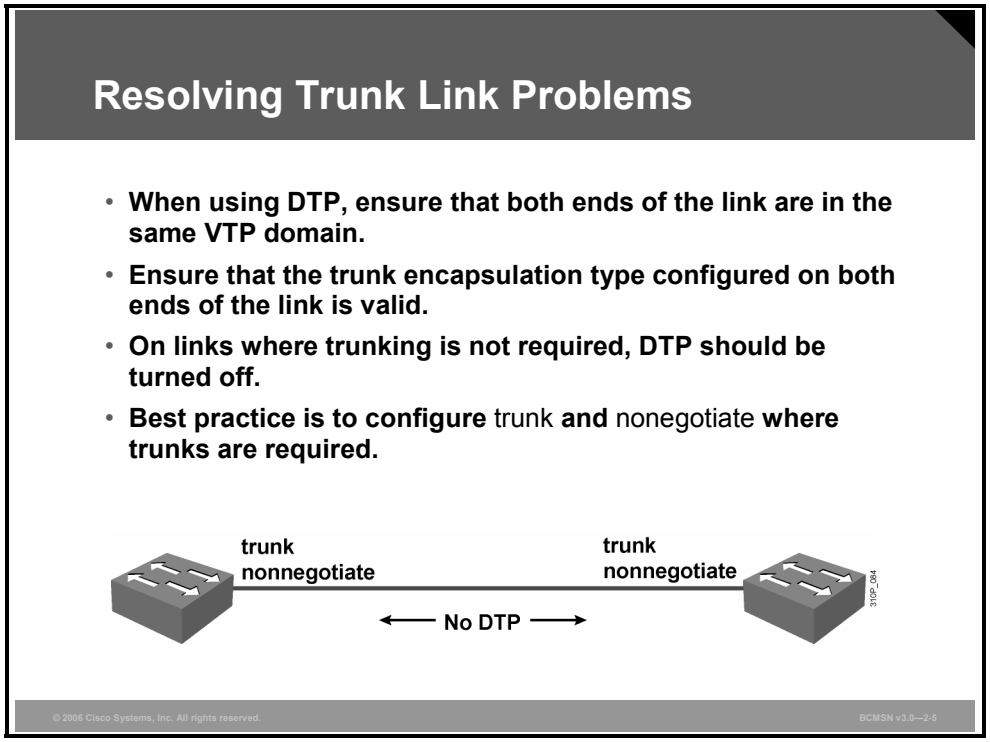

Example C

If one end of the link is set to **switchport mode trunk** and **switchport nonegotiate**, and the other end of the link is set to **switchport mode auto**, a mismatch will occur. This is because the switch on the left is not sending any DTP frames and so the port that is set to **switchport mode auto** on the switch on the right will default to being an access port.

```
1ASW3#show int fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off

1DSW1#show interfaces g1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
```

Resolving Trunk Link Problems

This topic identifies best practices for resolving trunk link problems.



Trunk negotiation is managed by the DTP, which is a point-to-point protocol. When using DTP to configure trunks, ensure that both ends of the link are in the same VTP domain.

Because DTP is a Cisco proprietary protocol, some internetworking devices do not support DTP frames, which could cause misconfigurations. To avoid this potential problem, when you configure an interface that is connected to a device that does not support DTP, do not forward DTP frames to that device. In other words, turn off DTP.

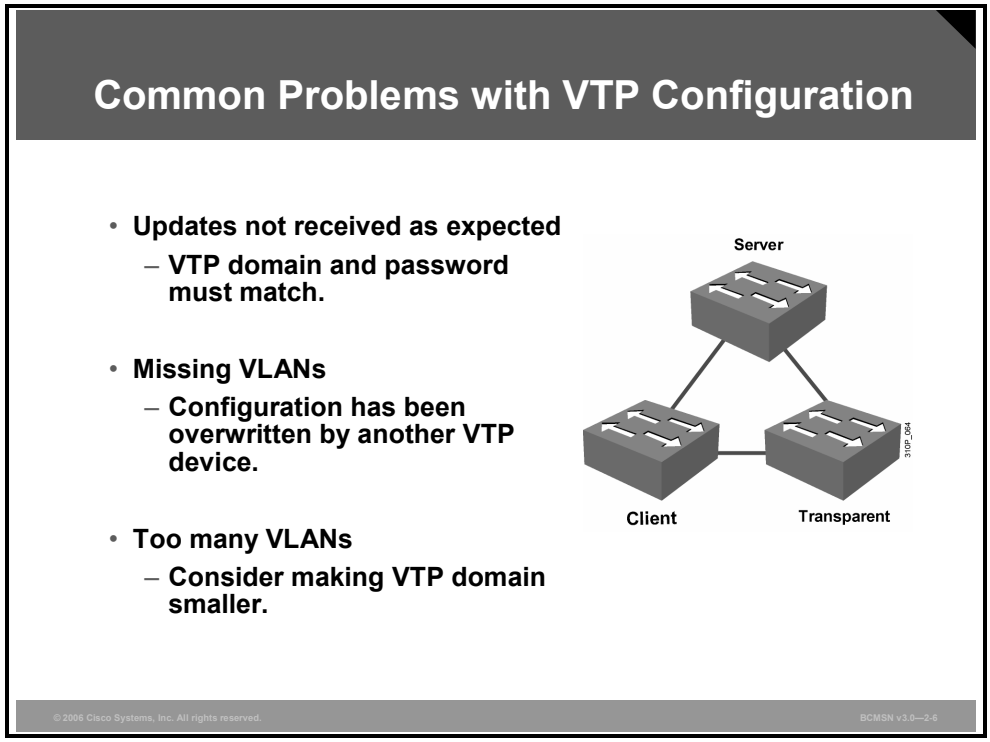
Use these commands to configure the ports into the appropriate mode:

- If you do not intend to trunk across those links, use the **switchport mode access** interface configuration command to disable trunking.
- To enable trunking to a device that does not support DTP, use the **switchport mode trunk** and **switchport nonegotiate** interface configuration commands to cause the interface to become a trunk but to not generate DTP frames.
- Use the **switchport trunk encapsulation** *isl* or **switchport trunk encapsulation** *dot1q* interface to select the encapsulation type on the trunk port.

Note	Whether a device supports DTP or not, general best practice is to configure trunks statically by configuring the interface to **trunk** and **nonegotiate**.

Common Problems with VTP Configuration

This topic identifies common problems with VTP configuration.



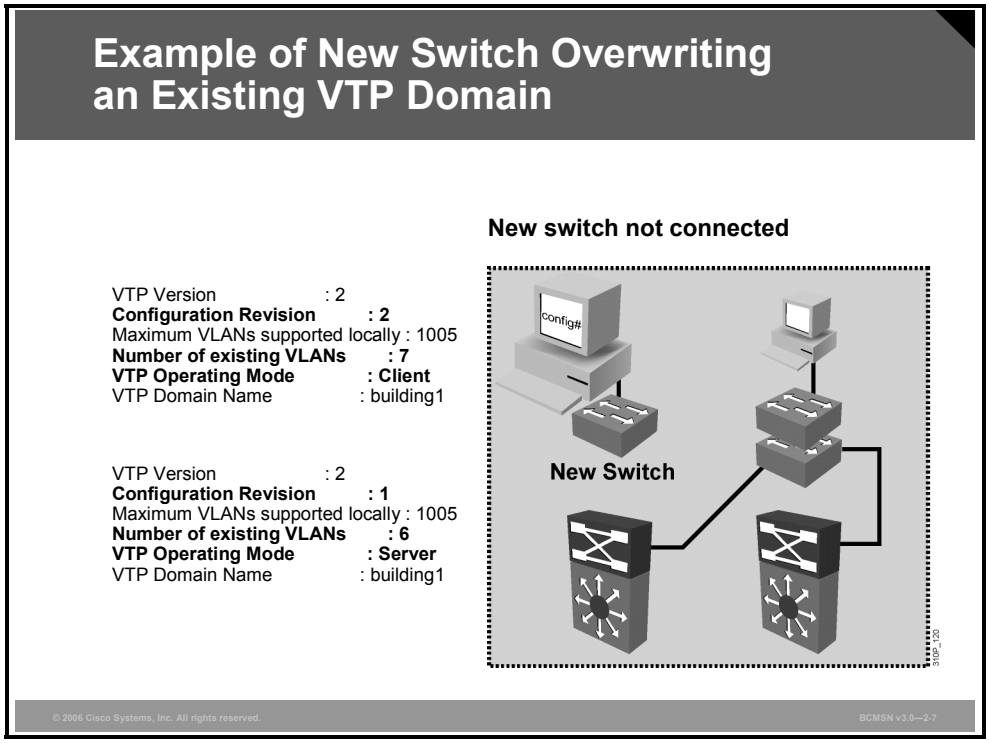
Common Problems with VTP Configuration

The table describes some unexpected results that can occur after VTP configuration.

Problem	Possible Causes
Updates not received as expected	■ VTP domain name and password must match on a given switch to receive updates from a VTP server. The domain name is case sensitive. ■ VTP version must be compatible with other switches in the domain. ■ Ensure that there is at least one server in the domain. ■ Check that a trunk link exists to VTP server.
Missing VLANs	■ Upon initial configuration, the VTP server may have had a partial VLAN database, and it overwrote the existing, more complete, database on the existing switch. ■ VLANs were deleted individually at the VTP server, and those deletions will be propagated in the domain. (To avoid this, ensure that any switch becoming a VTP server has a complete VLAN list.) ■ Not all Cisco switches support the same extended-range VLANs (those numbered higher than 1005). This information is not learned or propagated through VTP, so it may vary in a switched network.
Too many VLANs	■ The VTP server has a VLAN list that is more complete than the list needed by other switches in the domain.

Example of a Switch Overwriting an Existing VTP Domain

This subtopic describes how a switch can overwrite an existing VTP domain.



The configuration revision number is used when determining if a switch should keep its existing VLAN database or overwrite it with the VTP update sent by another switch in the same domain with the same password. Therefore, when a switch is added to a network, it is important that it does not inject spurious information into the domain.

Example of a VTP Client Overwriting a VTP Server

This is an example of a VTP client overwriting a VTP server when correct procedures were not followed:

The VTP server, 1DSW1, is currently at configuration revision 1 and knows of six VLANs.

```
1DSW1#show vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 6
VTP Operating Mode              : Server
VTP Domain Name                 : building1
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x0B 0xED 0x6C 0xE2 0x16
0xE9 0x3D 0x3C
Configuration last modified by 172.16.1.111 at 3-1-93 00:29:26
```

```
Local updater ID is 172.16.1.111 on interface Vl1 (lowest
numbered VLAN interface found)
```

The new switch, 1ASW3, has not yet been connected to the network. It is a VTP client with a configuration revision of 2 and knows of seven VLANs.

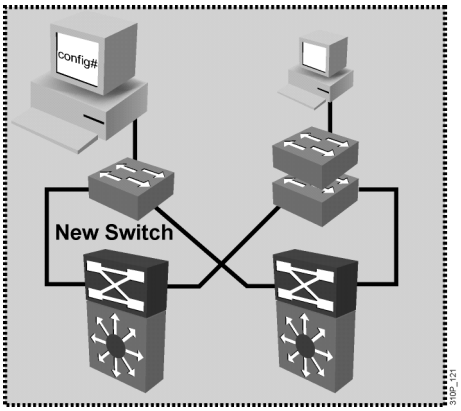
```
1ASW3#show vtp status
VTP Version                     : 2
Configuration Revision          : 2
Maximum VLANs supported locally : 250
Number of existing VLANs        : 7
VTP Operating Mode              : Client
VTP Domain Name                 : building1
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x7C 0x2A 0x2B 0xF1 0x2C
0x90 0x5D 0xB2
Configuration last modified by 172.16.1.11 at 3-1-93 00:34:17
```

Example of New Switch Overwriting an Existing VTP Domain (Cont.)

New switch connected

VTP Version : 2
Configuration Revision : 2
Maximum VLANs supported locally : 1005
Number of existing VLANs : 7
VTP Operating Mode : Client
VTP Domain Name : building1

VTP Version : 2
Configuration Revision : 2
Maximum VLANs supported locally : 1005
Number of existing VLANs : 7
VTP Operating Mode : Server
VTP Domain Name : building1



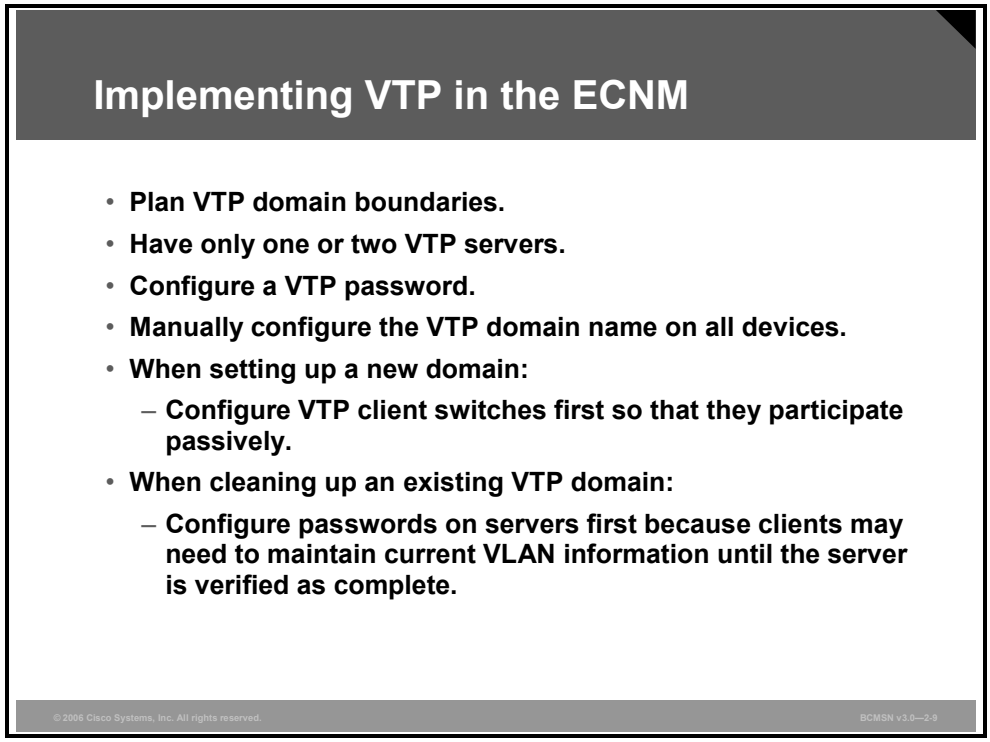
The link between 1DSW1 and 1ASW3 is now connected, and the VTP client overwrites the VTP server because of the higher configuration revision number.

```
1DSW1#
00:43:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1,
changed state to up


1DSW1#show vtp status
VTP Version                     : 2
Configuration Revision          : 2
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : building1
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x7C 0x2A 0x2B 0xF1 0x2C
0x90 0x5D 0xB2
Configuration last modified by 172.16.1.11 at 3-1-93 00:34:17
Local updater ID is 172.16.1.111 on interface Vl1 (lowest
numbered VLAN interface found)
```

To prevent a VTP domain from being overwritten, always add a new switch either in VTP transparent mode or as a VTP client with a revision number that is lower than the revision number in the existing VTP domain.

Best Practice for VTP Configuration

This topic describes best practice for VTP configuration.

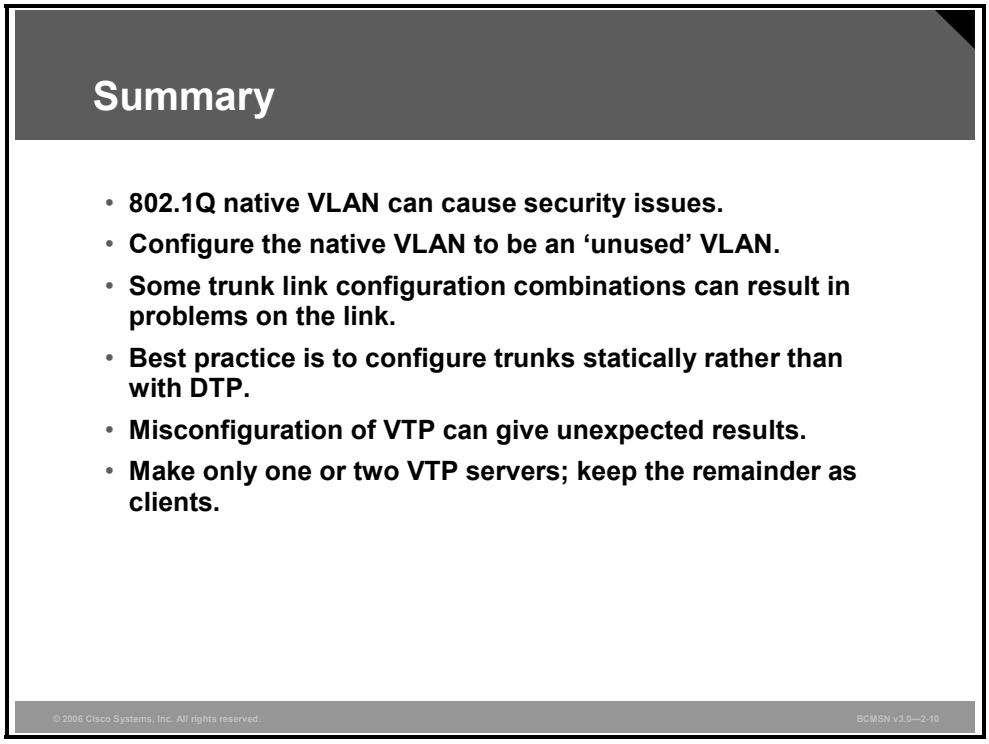


Here is a list of general best practices with regard to configuring VTP in the Enterprise Composite Network Model (ECNM):

- Plan boundaries for the VTP domain. Not all switches in the network need information on all VLANs in the network. In the ECNM, the VTP domain should be restricted to redundant distribution switches and the access switches that they serve.
- Have only one or two switches specifically configured as VTP servers and the remainder as clients.
- Configure a password so that no switch can join the VTP domain with a domain name only (which can be derived dynamically).
- Manually configure the VTP domain name on all switches that are installed in the network so that the mode can be specified and the default mode of server on all switches can be overwritten.
- When you are setting up a new domain, configure VTP client switches first so that they participate passively; then configure servers to update client devices.
- In an existing domain, if you are performing VTP cleanup, configure passwords on servers first. Clients may need to maintain current VLAN information until the server contains a complete VLAN database. After the VLAN database on the server is verified as complete, client passwords can be configured to be the same as the server passwords. Clients will then accept updates from the server.

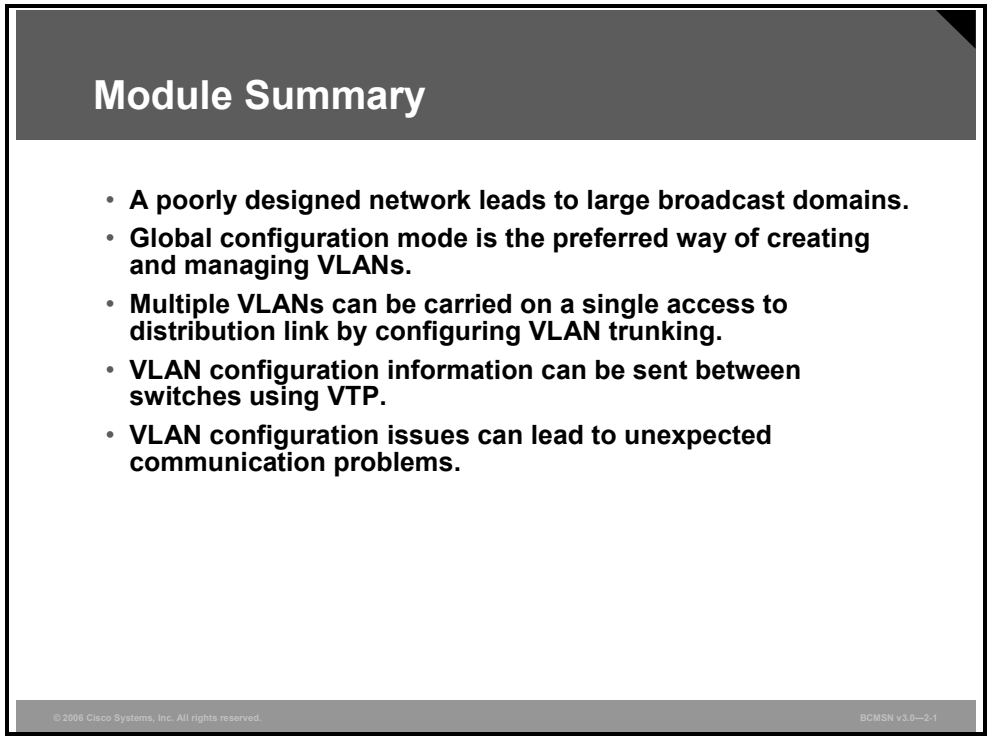
Summary

This topic summarizes the key points discussed in this lesson.



Module Summary

This topic summarizes the key points discussed in this module.



This module examined the function of VLANs and how they are implemented in a switched campus network. Depending on its configuration as an access or trunk port, each switch port can be associated with one or many VLANs.

The Inter-Switch Link (ISL) and 802.1Q protocols are used to establish trunk links carrying traffic for multiple VLANs.

Trunk links between switches can also carry VLAN Trunk Protocol (VTP) information, which allows VLAN names and descriptions contained in a VLAN database to be shared between switches.

References

For additional information, refer to these resources.

- Cisco Systems, Inc., *VLAN Trunking Protocols: Inter-Switch Link and IEEE 802.1Q Frame Format:* http://www.cisco.com/en/US/tech/tk389/tk390/technologies_tech_note09186a0080094665.shtml
- Cisco Systems, Inc., *Catalyst 3550 Multilayer Switch Software Configuration Guide, 12.1(13)EA1: Configuring VTP:* http://www.cisco.com/en/US/products/hw/switches/ps646/products_configuration_guide_chapter09186a008014f376.html
- Cisco Systems, Inc., *Products & Services: Tool Index*: http://www.cisco.com/en/US/products/prod_tools_index.html
- Cisco Systems, Inc. *Virtual LANs/VLAN Trunking Protocol (VLANs/VTP): Understanding and Configuring VLAN Trunk Protocol (VTP)*: http://cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094c52.shtml
- Cisco Systems, Inc., *Catalyst 6500 Series Software Configuration Guide, 8.1: Configuring VTP:* http://www.cisco.com/en/US/partner/products/hw/switches/ps708/products_configuration_guide_chapter09186a008019f048.html#wp1017196

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) Which two interconnection technologies are considered appropriate for servers in the Enterprise Composite Network Model? (Choose two.) (Source: Implementing Best Practices for VLAN Topologies)

A) Fast Ethernet
B) Gigabit Ethernet
C) 10-Gigabit Ethernet
D) 10-Gigabit EtherChannel

Q2) Which two statements identify network benefits provided by VLANs? (Choose two.) (Source: Implementing VLANs)

A) VLANs divide the network into larger broadcast domains or subnets.
B) VLANs reduce the impact of network problems.
C) VLANs help to isolate problem employees.
D) VLANs can transmit frames to all ports in all VLANs.
E) VLANs allow you to segregate frames that contain sensitive or critical information.

Q3) Which two features belong to the 802.1Q trunking protocol? (Choose two.) (Source: Implementing Trunks)

A) It encapsulates Ethernet frames.
B) It alters the existing Ethernet frame.
C) It supports native VLANs.
D) It does not support native VLANs.

Q4) What are the two guidelines that you must follow when you are adding a new switch to an existing VTP management domain? (Choose two.) (Source: Propagating VLAN Configurations with VTP)

A) Set the switch to transparent mode before connecting it to the network.
B) Set the switch to server mode before connecting it to the network.
C) Change the VTP domain information while the switch is in server mode.
D) Change the VTP domain name to reset the configuration revision number.

Q5) A link between two switches in configured as **switchport mode dynamic desirable** at one end and **switchport mode access** at the other end. What mode will the link be operating in? (Source: Correcting Common VLAN Configuration Errors)

A) It will be operating in trunk mode.
B) It will be operating in access mode.
C) The link will not work at all.
D) It will be operating in access mode but there could be problems with VLAN hopping.

Module Self-Check Answer Key

Q1) A, B

Q2) B, E

Q3) B, C

Q4) A, D

Q5) B

Module 3

Implementing Spanning Tree

Overview

This module introduces the fundamentals of Spanning Tree Protocol (STP) operation in a switched network. The root bridge will be explained as well as how the root bridge and its backup are elected. Features for enhancing the performance of STP will be covered—namely, Rapid STP (RSTP) and Multiple STP (MSTP).

You will discover how EtherChannel is configured and how it interoperates with STP. The module also provides guidelines on improving STP resiliency when network faults occur.

Module Objectives

Upon completing this module, you will be able to implement spanning tree operation in a hierarchical network. This ability includes being able to meet these objectives:

- Explain the operation of STP to include enhancements to it, such as RSTP, PVST+, PVRST, and MSTP
- Describe RSTP and the procedure for implementing it in an existing network
- Describe MSTP and the procedure for implementing it in an existing network
- Configure link aggregation with EtherChannel

Lesson 1

Describing the STP

Overview

In a campus network where there are redundant links between switches, Spanning Tree Protocol (STP) manages which links will provide an active Layer 2 path, which ones will be inactive, and which ones will provide redundancy in the case of active path failure. This lesson will examine the general components and operation of STP in a switched network.

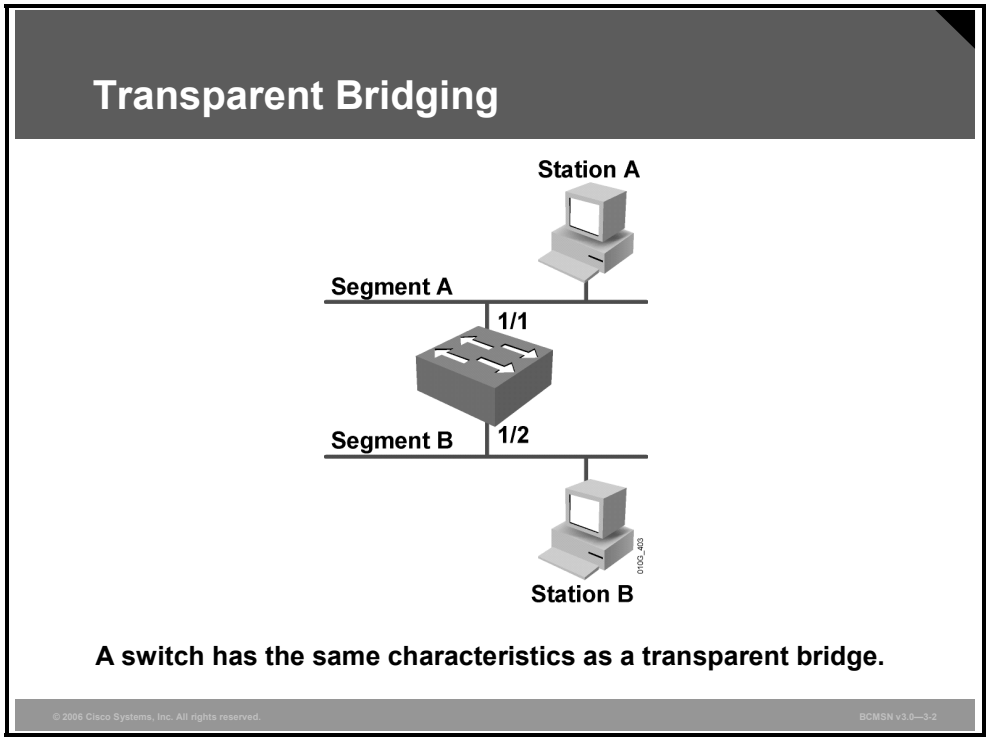
Objectives

Upon completing this lesson, you will be able to explain the operation of the spanning tree protocols to include enhancements to it, such as RSTP, PVST+, PVRST, and MSTP. This ability includes being able to meet these objectives:

- Describe a transparent bridge
- Identify the traffic patterns in a bridge loop
- Define a loop-free network
- Describe the 802.1D STP
- Define a root bridge
- Describe the four port roles
- Describe PortFast, PVST+, RSTP, MSTP, and PVRST

Describing Transparent Bridges

This topic describes transparent bridges.



Because switches have replaced bridges as the network device for implementing transparent bridging in modern networks, the basic functionality of a switch is identical to that of a transparent bridge on a per-VLAN basis. To understand STP, it is important first to look at the behavior of a transparent bridge without spanning tree.

By definition, a transparent bridge has these characteristics:

- It must not modify the frames that are forwarded.
- It learns addresses by "listening" on a port for the source address of a device. If a source MAC address is read in frames coming in a specific port, the bridge assumes that frames destined for that MAC address can be sent out of that port. The bridge then builds a table that records what source addresses are seen on what port. A bridge is always listening and learning MAC addresses in this manner.
- It must forward all broadcasts out all ports, except for the port that initially received the broadcast.
- If a destination address is unknown to the bridge, it forwards the frame out all ports except for the port that initially received the frame. This is known as unicast flooding.

Transparent bridging, by definition, must be transparent to the devices on the network. End stations require no configuration. The existence of the bridging protocol operation is not directly visible to them—hence, the term transparent bridging.

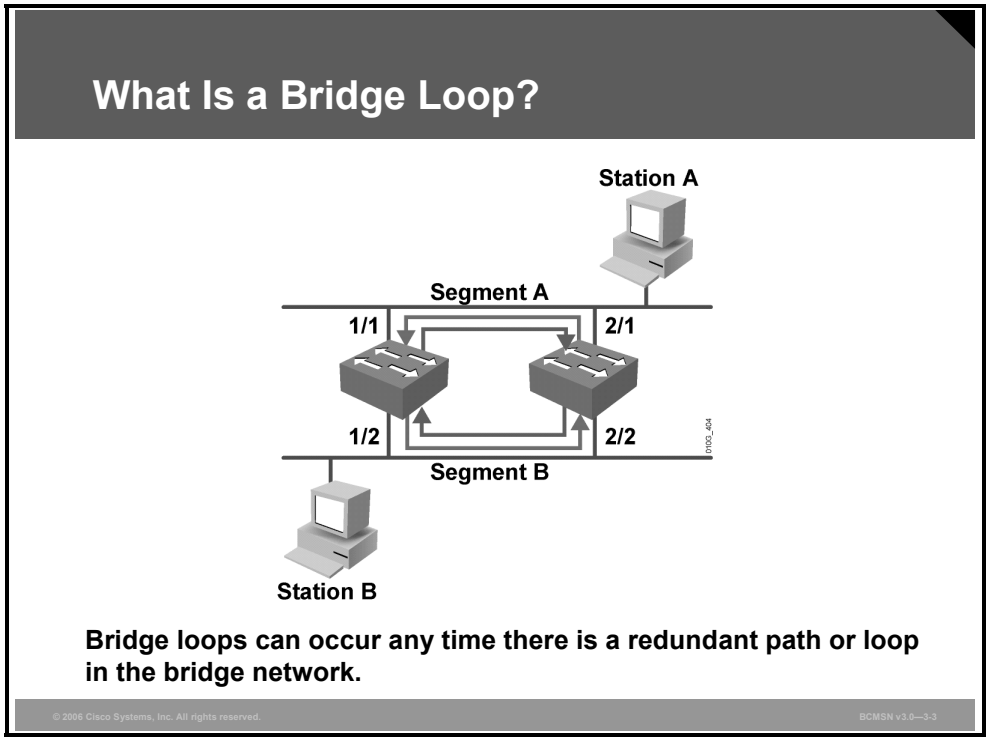
As with traditional shared Ethernet, transparent bridges inherently lack the capability to provide redundancy. The STP inserts a mechanism into the Ethernet transparent bridge environment to dynamically discover the network topology and ensure only one path through the network.

Without STP, there is no way to make a transparent bridge environment redundant. STP also protects a network against accidental miscablings because it prevents unwanted bridging loops in the transparent bridging environment.

Note	Be aware that the spanning tree algorithm is implemented in other media types, such as Token Ring. STP has a different purpose and function in Token Ring than in Ethernet because bridging loops can be desirable in Token Ring.

Identifying Traffic Loops

This topic identifies the traffic patterns in a bridge loop.



A bridge loop is observed when a frame that is forwarded circulates cyclically and redundantly; this occurs where there is no mechanism to manage the redundant Layer 2 paths.

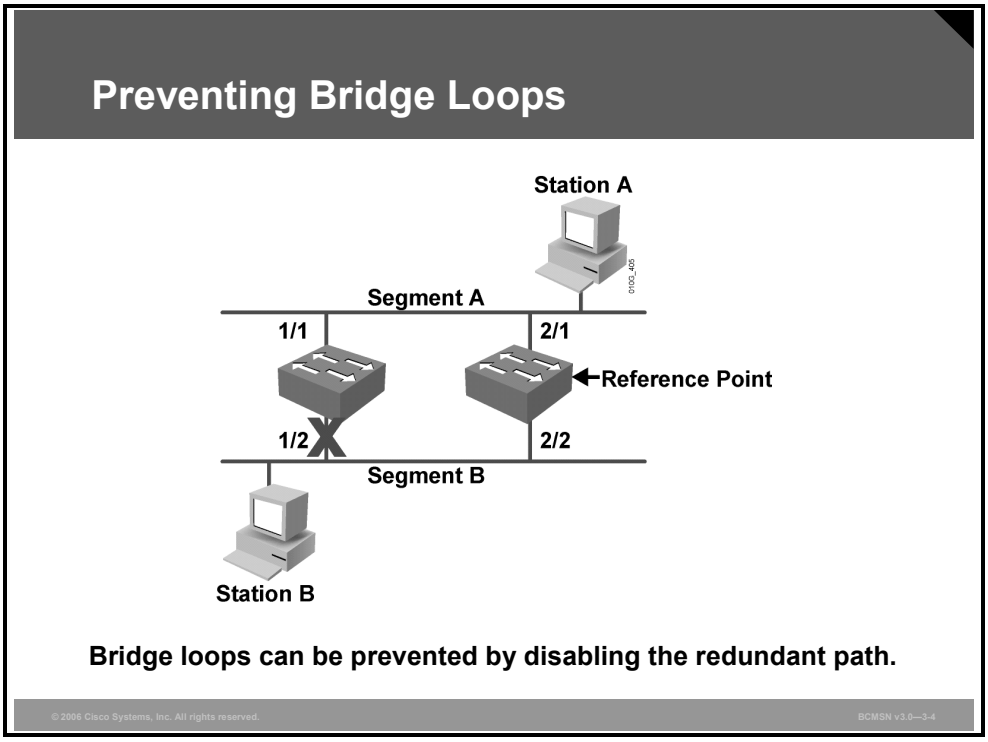
Example: Flooded Unicast Frames and Bridge Loops

Station A has two potential paths to station B by way of the two intermediate bridges. Here is a description of what happens if station A sends frames to station B, if there were no provisions enabled to deal with redundant paths.

Step	Action
1.	Station A transmits the frame destined for station B onto segment A. Both bridges on segment A pick up the frame on their bridge ports 1/1 and 2/1. Both bridges populate their respective MAC tables, indicating that station A resides on segment A, on bridge ports 1/1 and 2/1.
2.	Both bridges forward the frame to segment B. Station B receives the frame, and both bridges also see the same frame, with the MAC address of station A in the Source Address (SA) field, coming from the other bridge. The bridges will now incorrectly forward all frames for station A to segment B. When station B responds to station A, the frame will be dropped by both bridges because it will be received on the same bridge ports that it considers the destination of station A.
3.	If station A, or any station, sends a broadcast, the effects of the Layer 2 loop would be much worse. The destination MAC address would be FF-FF-FF-FF-FF-FF. This would cause each bridge to forward the frame out all bridge ports except the bridge port upon which the frame was received. The broadcast frame would also be forwarded to the originating bridge, which would again forward the same broadcast out all bridge ports. This broadcast would continue until the loop was shut down or until the bridge could no longer handle the load.

Explaining a Loop-Free Network

This topic defines a loop-free network.



A loop-free network is one in which no Layer 2 loops exist; therefore, the network cannot create Layer 2 broadcast storms or flooded unicast storms. A loop-free network can be achieved manually by shutting down or disconnecting all redundant links between bridges. However, this leaves no redundancy in the network and requires manual intervention in the event of a link failure.

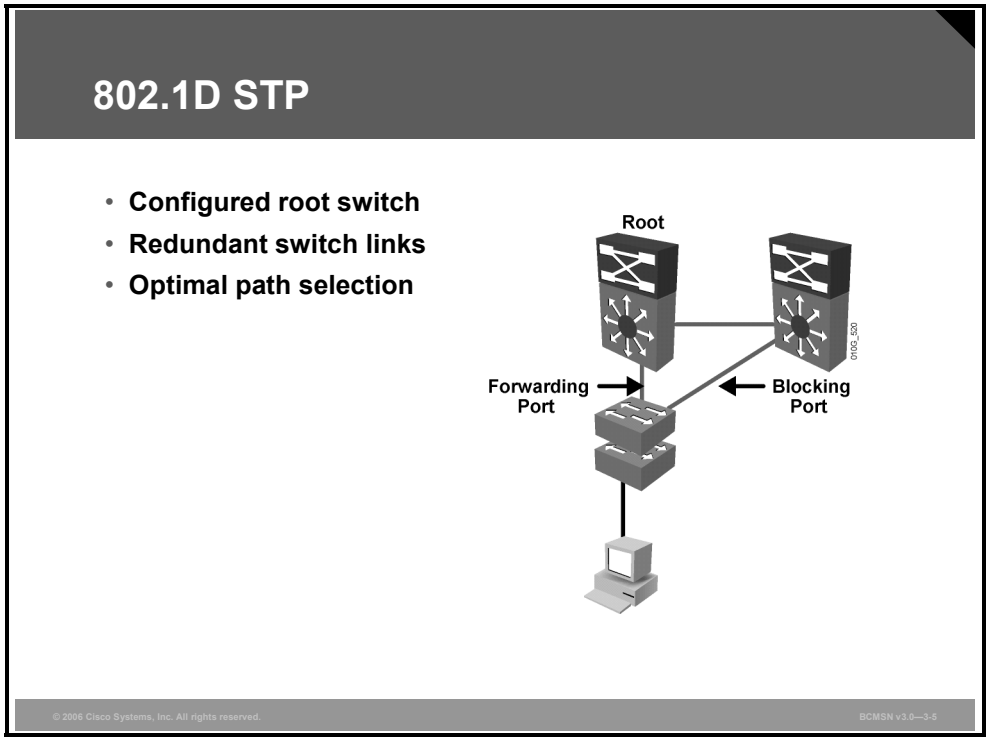
STP resolves this problem. Where there are alternative links to a destination on a switch, only one link will be used to forward data, unless there is a failure on that link. The switch ports associated with alternative paths remain aware of the topology of the network and can be enabled if a failure occurs on a primary link. In the case of primary link failure, the switch will begin forwarding frames over an alternative link.

The spanning tree algorithm (STA) runs on each switch to activate or block redundant links. To find the redundant links, the STA chooses a reference point in the network and determines if there are redundant paths to that reference point. If the STA finds a redundant path, it chooses which path will forward frames and which redundant path or paths will be blocked. This effectively severs the redundant links within the network until they are needed when the primary link toward the reference point fails.

Spanning tree standards often refers to a “bridge,” but it is likely that all the devices exchanging spanning tree information will be Layer 2 switches.

Describing the 802.1D STP

This topic describes the 802.1D STP.

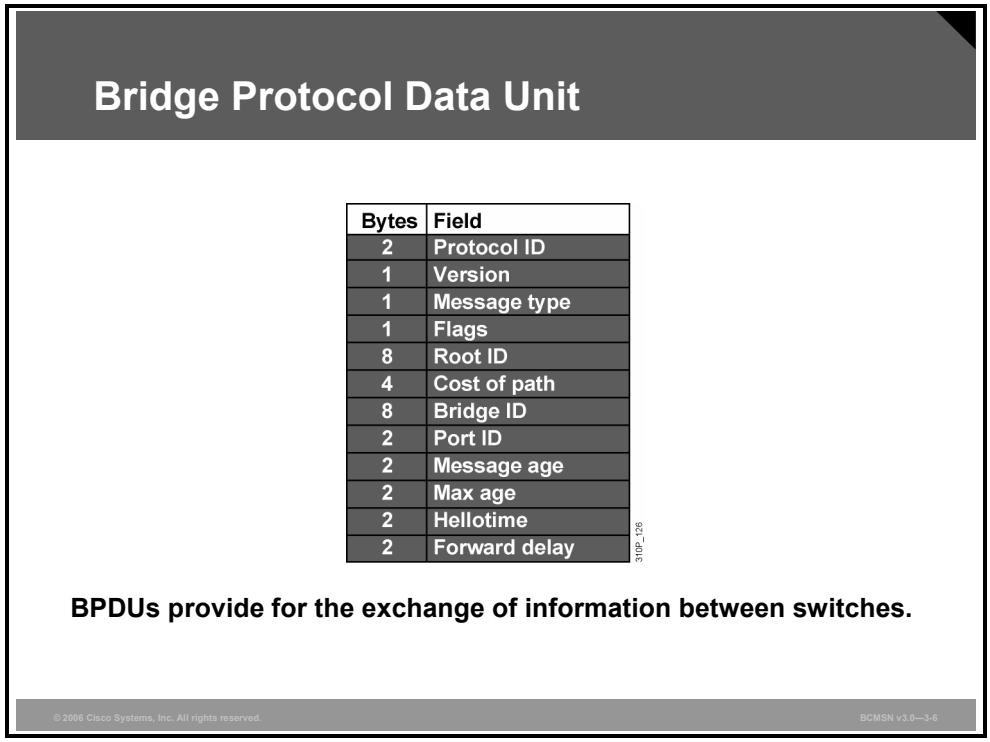


The 802.1D STP provides a mechanism for switches to reconfigure the paths over which they forward frames, making possible a loop-free path when there are redundant switch paths through the network. This is accomplished by forwarding traffic over specific ports and by disabling other ports to prevent frames from being sent repeatedly or in a loop. STP prevents loops by using these mechanisms:

- STP is implemented through the exchange of bridge protocol data unit (BPDU) messages between adjacent switches.
- A single "root bridge" is elected to serve as the reference point from which a loop-free topology is built for all switches exchanging BPDUs.
- Each switch, except for the root bridge, determines a “root port” that provides the best path to the root bridge.
- In a triangular design similar to the one in the figure, on the link between the two nonroot switch ports, a port on one switch will become a designated port, and the port on the other switch will be in a blocking state, not forwarding frames. This effectively breaks any loop. Typically, the designated port will be on the switch with the best path to the root bridge.
- Any port state change on any switch is considered a network topology change (for example, if a port goes up or down and the STA must be run on all switches to adapt to the new topology).

Spanning Tree Communication

This subtopic identifies the information contained in a BPDU that is used to send spanning tree information between switches.



STP sends configuration messages out every port of the bridge. These messages are called BPDUs.

Here is some of the information provided in a BPDU:

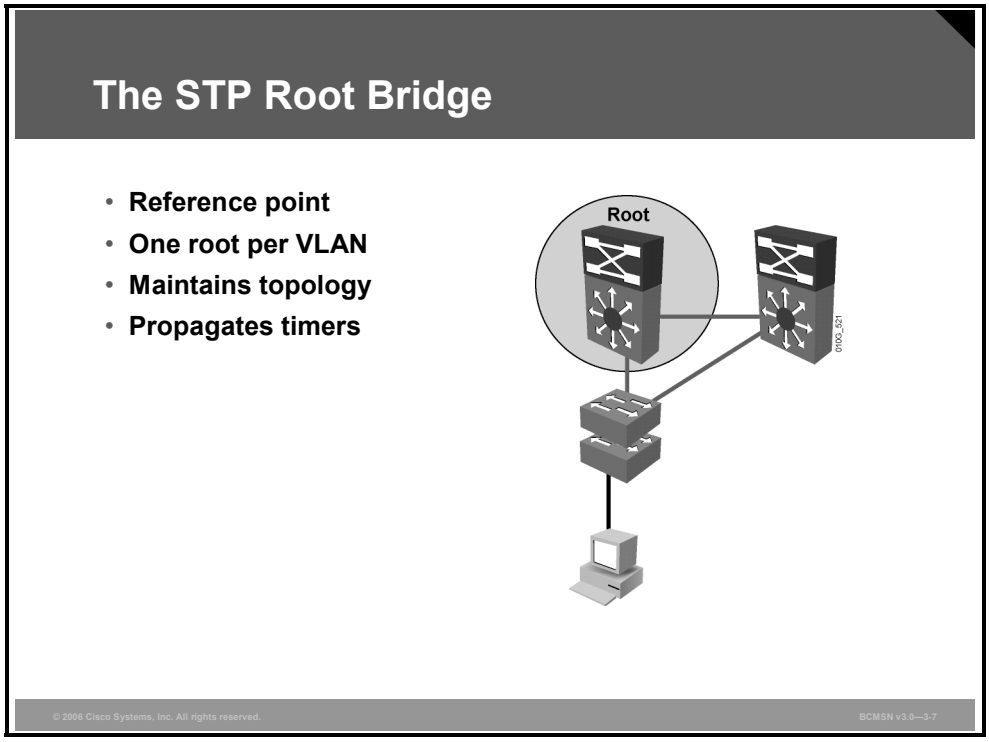
- **Root ID:** The lowest bridge ID (BID) in the topology
- **Cost of path:** Cost of all links from the transmitting switch to the root bridge
- **BID:** BID of the transmitting switch
- **Port ID:** Transmitting switch port ID
- **STP timer values:** Max age, hellotime, forward delay

BPDUs contain the appropriate information for STP configuration. The Type field for the BPDU message is 0x00, and it uses the multicast MAC address 01-80-C2-00-00-00.

The switch compares the BPDUs received on all ports to its own values to determine what role the receiving switch and its ports will play in the STP topology.

Describing the Root Bridge

This topic defines a root bridge.



STP uses the concepts of root bridge, root ports, and designated ports to establish a loop-free path through the network. The first step in creating the loop-free spanning tree is to elect a root bridge. The root bridge is the reference point that all switches use to establish forwarding paths that will avoid loops in the Layer 2 network.

The main information to be concerned with is the root ID (bridge that the transmitting bridge thinks is the root), BID, and cost (which is the cost to the root bridge). The STP topology is considered converged after a root bridge has been selected and each bridge has selected its root port, designated bridge, and the ports that will participate in the STP topology. STP uses these configuration messages (BPDUs) as it transitions port states to achieve convergence.

Spanning tree elects one bridge on the LAN to be the master bridge. This bridge is called the root bridge. The root bridge is special because all the path calculation through the network is based on the root. The bridge is elected based on the BID, which consists of a 2-byte Priority field plus a 6-byte MAC address. In spanning tree, lower BID values are preferred. In a default configuration, the Priority field is set at 32768.

Because the default Priority field is the same for all the bridges, the root selection is based on the lowest MAC address. One method of selecting a specific bridge to be the root is to manually alter the Priority field to a lower value. Regardless of what the MAC address is, the Priority field decides which bridge is going to be the root, assuming that all bridges do not have the same priority value.

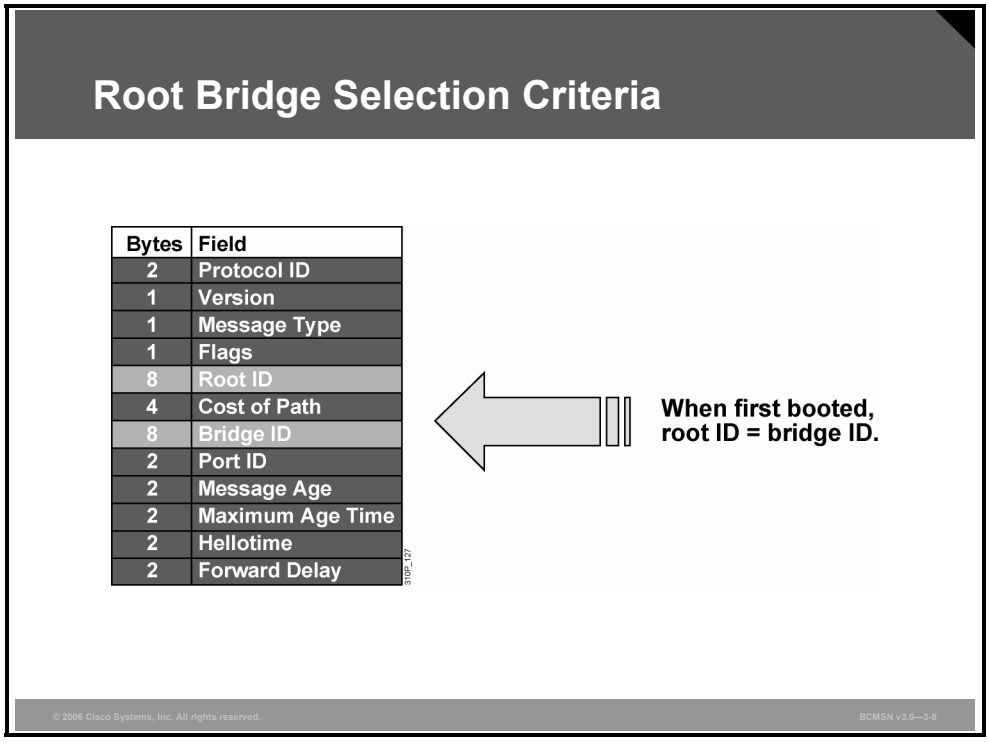
When a topology change occurs as a result of switch link state changes, the root will send messages throughout the tree regarding the topology change. This allows the content addressable memory (CAM) tables to adjust and to provide for a new path that may be used toward end host devices.

Timer information is also sent by the root bridge to nonroot bridges, informing them of the intervals to use as the ports transition through the spanning tree port states.

The root bridge maintains the stability of the forwarding paths between all switches for a single STP instance. A spanning tree instance is a configuration in which all switches exchanging BPDUs and participating in spanning tree negotiation are associated with a single root. If this is done for all VLANs, it is called a Common Spanning Tree (CST) instance. There is also a Per VLAN Spanning Tree (PVST) implementation that provides one instance, and therefore one root bridge, for each VLAN.

BPDU Fields Associated with Root Bridge Selection

This subtopic describes the criteria used to determine which device will be elected as the root.



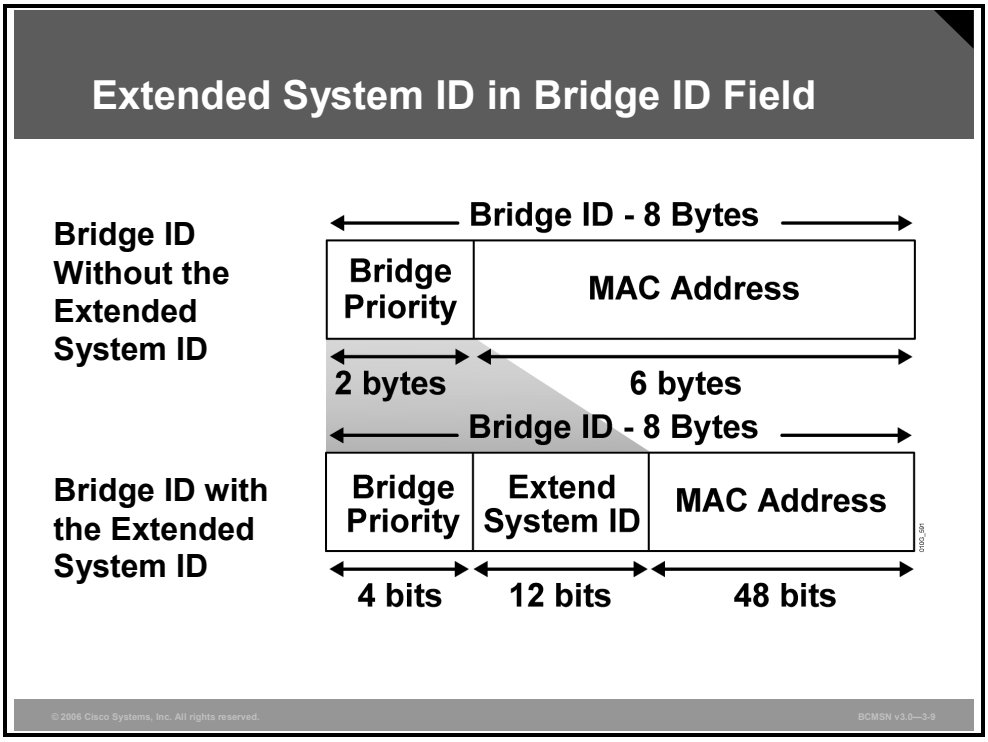
The BID and root ID are both 8-byte fields carried in a BPDU. These values are used to complete the root bridge election process. A switch identifies the root bridge by evaluating the Root ID field in the BPDUs it receives. The unique BID of the root bridge will be carried in the Root ID field of the BPDUs that are sent by each switch in the tree.

When a switch first boots and begins sending BPDUs, it has no knowledge of a root ID, so it will populate the Root ID field of outbound BPDUs with its own BID.

The switch with the lowest numerical BID will assume the role of root bridge for that spanning tree instance. Upon receipt of BPDUs with lower BIDs than its own, a switch will place the lowest value seen in all BPDUs into the Root ID field information of its outbound BPDUs.

BID Field in the BPDU

This subtopic describes the BID field content.



Spanning tree operation requires that each switch have a unique BID. In the original 802.1D standard, the BID was composed of the Priority Field and the MAC address of the switch, and all VLANs were represented by a CST. Because PVST requires that a separate instance of spanning tree run for each VLAN, the BID field is required to carry VLAN ID (VID) information. This is accomplished by reusing a portion of the Priority field as the extended system ID to carry a VID.

To accommodate the extended system ID, the original 802.1D 16-bit Bridge Priority field is split into two fields, resulting in these components in the BID:

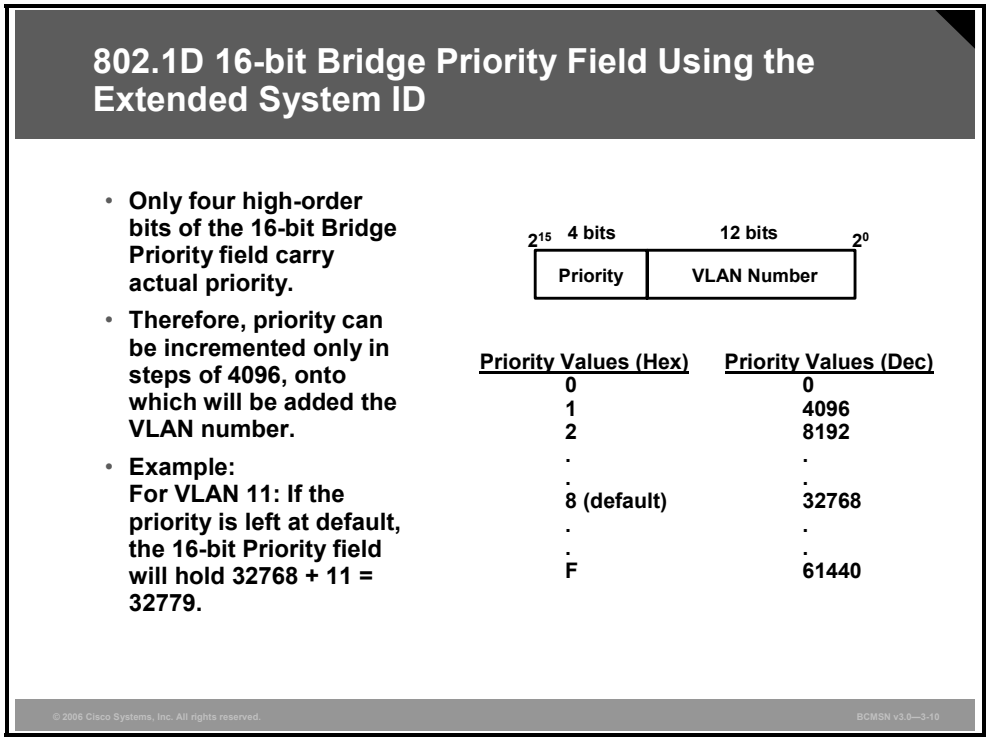
- **Bridge Priority:** A 4-bit field still used to carry bridge priority. Because of the limited bit count, priority is now conveyed in discreet values in increments of 4096 rather than discreet values in increments of 1, as they would be with the full 16-bit field available. The default priority, in accordance with IEEE 802.1D, is 32,768, which is the midrange value.
- **Extended System ID:** A 12-bit field carrying, in this case, the VID for PVST.
- **MAC Address:** A 6-byte field with the MAC address of a single switch.

By virtue of the MAC address, a BID is always unique. When the priority and extended system ID are appended to the switch MAC address, each VLAN on the switch can be represented by a unique BID.

If no priority has been configured, every switch will have the same default priority, and the election of the root for each VLAN will be based on the MAC address. This is a fairly random means of selecting the ideal root bridge; for this reason, it is advisable to assign a lower priority to the switch that should serve as root bridge.

Priority Field in the BPDU

This subtopic describes the priority field content.



Only four bits are now used to set the bridge priority. Because of the limited bit count, priority is now configurable only in increments of 4096.

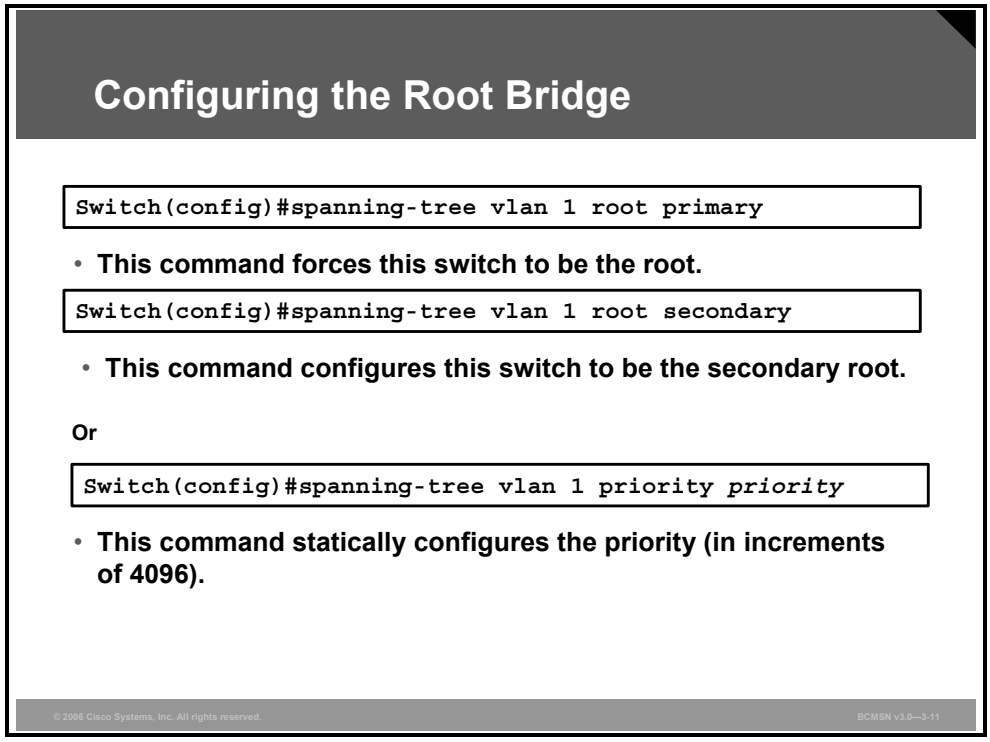
A switch responds with possible priority values if an incorrect value is entered:

```
Switch(config)#spanning-tree vlan 1 priority 1234
% Bridge Priority must be in increments of 4096.
% Allowed values are:
  0     4096  8192  12288 16384 20480 24576 28672
  32768 36864 40960 45056 49152 53248 57344 61440
```

- If no priority has been configured, every switch will have the same default priority of 32768.
- Assuming that all other switches are at default priority, the **spanning-tree vlan** *vlan-id* **root primary** command will set a value of 24576.
- Assuming that all other switches are at default priority, the **spanning-tree vlan** *vlan-id* **root secondary** command will set a value of 28672.

How to Configure a Root Bridge

This subtopic identifies the commands for configuring a switch as the root bridge.



The switch with the lowest BID becomes the root bridge for a VLAN. Specific configuration commands are used to determine which switch will become the root bridge.

A Cisco Catalyst switch running PVST maintains an instance of spanning tree for each active VLAN that is configured on the switch. A unique BID is associated with each instance. For each VLAN, the switch with the lowest BID becomes the root bridge for that VLAN. Whenever the bridge priority changes, the BID also changes. This results in the recomputation of the root bridge for the VLAN.

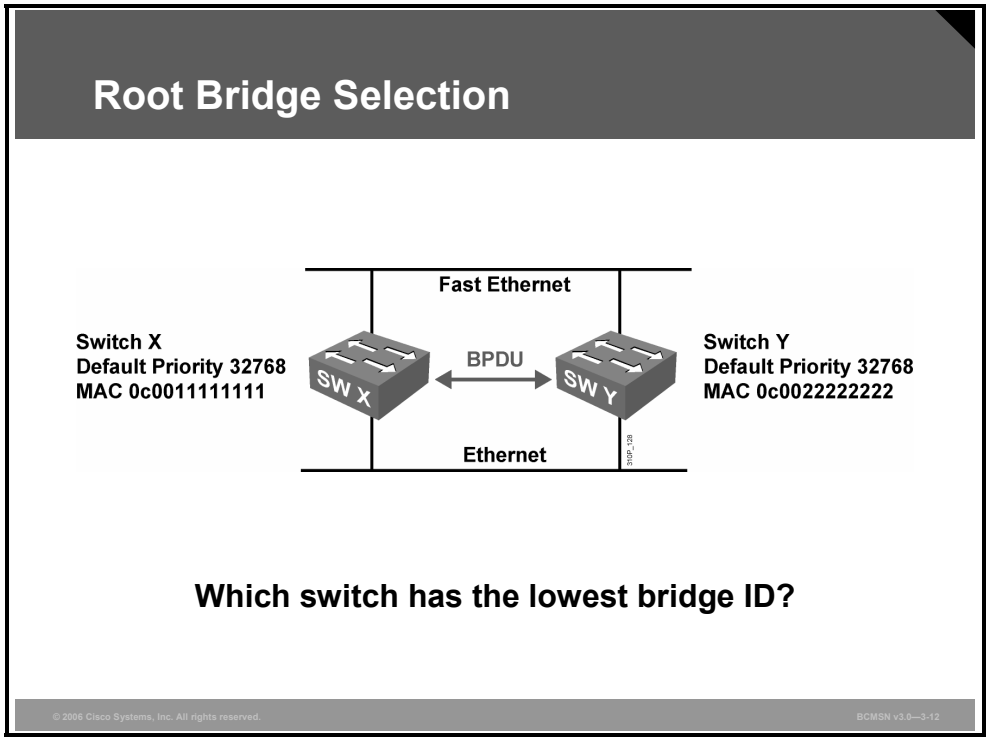
To configure a switch to become the root bridge for a specified VLAN, use the command **spanning-tree vlan** *vlan-ID* **root primary**.

Caution	Spanning tree commands take effect immediately, so network traffic will be disrupted while reconfiguration occurs.

A secondary root is a switch that may become the root bridge for a VLAN if the primary root bridge fails. To configure a switch as the secondary root bridge for the VLAN, use the command **spanning-tree vlan** *vlan-ID* **root secondary**. Assuming that the other bridges in the VLAN retain their default STP priority, this switch will become the root bridge in the event that the primary root bridge fails. This command can be executed on more than one switch to configure multiple backup root bridges.

Identifying the Root Selection Process

This subtopic describes the process by which a root bridge is elected.



BPDUs are exchanged between switches, and the analysis of the BID and root ID information from those BPDUs determines which bridge is selected as the root bridge.

In the example shown, both switches have the same priority for the same VLAN. The switch with the lowest MAC address will, therefore, be elected root bridge. In the example, switch X is the root bridge for VLAN1, with a BID of 0x8001:0c0011111111.

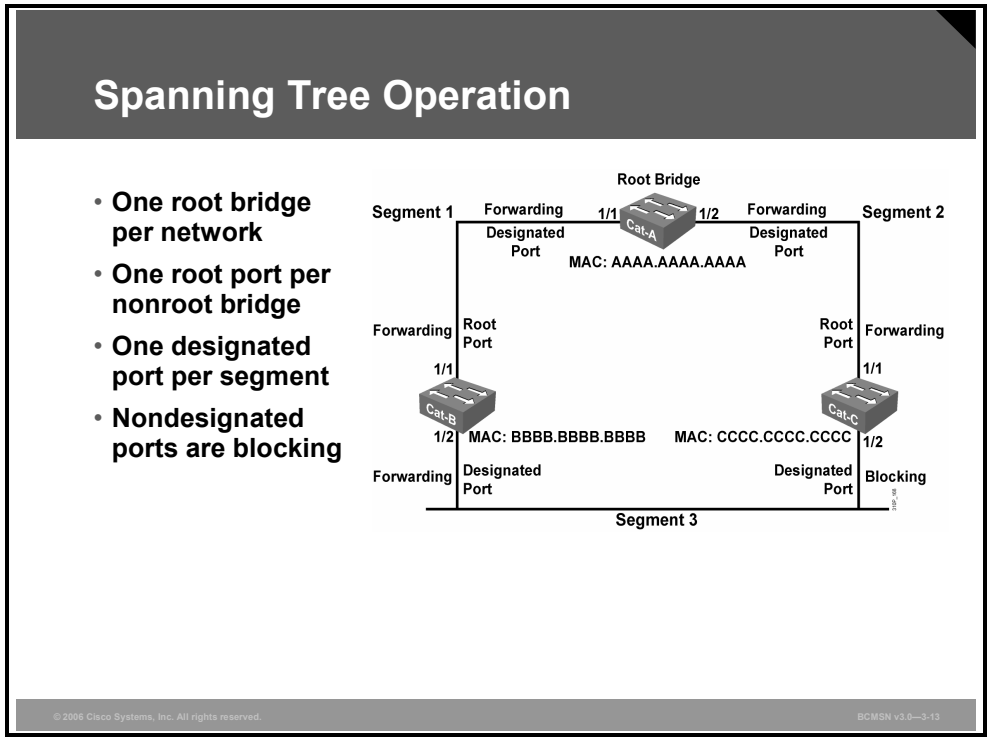
Election of a Root Bridge

These steps show how a root bridge election occurs.

Step	Action
1.	Upon startup, each switch transmits BPDUs out all enabled interfaces on a per-VLAN basis. At startup, each switch sets the root ID equal to its own BID. During this time, the switch ports are not used to forward standard data frames.
2.	As the BPDU goes out through the network, each switch compares the root BPDU it sent out to the one it received. The exact BPDU fields and how they are compared are outlined in the next topic.
3.	If the received root ID is superior, the switch will propagate it; otherwise, it will continue to send its own BID as the root BID in transmitted BPDUs.
4.	On the root bridge, all ports are designated ports in a forwarding state.
5.	Nonroot bridges must determine an optimal path to the root.

Describing Port Roles

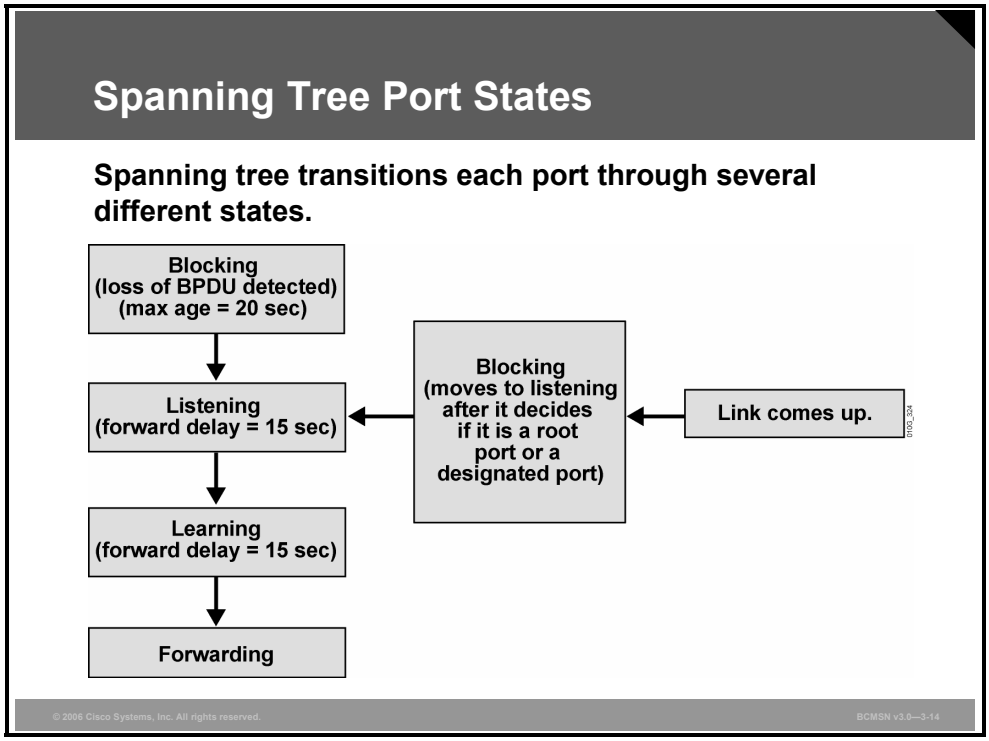
This topic describes the four port roles.



On a nonroot bridge, as spanning tree receives BPDUs on various ports, it determines the roles that each port will fill in the topology. There are four 802.1D port roles.

Port Role	Description
Root port	This port exists on nonroot bridges and is the switch port with the best path to the root bridge. Root ports forward traffic toward the root bridge, and the source MAC address of frames received on the root port is capable of populating the MAC table. Only one root port is allowed per bridge.
Designated port	This port exists on root and nonroot bridges. For root bridges, all switch ports are designated ports. For nonroot bridges, a designated port is the switch port that will receive and forward frames toward the root bridge as needed. Only one designated port is allowed per segment. If multiple switches exist on the same segment, an election process determines the designated switch, and the corresponding switch port begins forwarding frames for the segment. Designated ports are capable of populating the MAC table.
Nondesignated port	The nondesignated port is a switch port that is not forwarding (blocking) data frames and not populating the MAC address table with the source addresses of frames seen on that segment.
Disabled port	The disabled port is a switch port that is shut down.

By examining the switch port roles on a switch, spanning tree can determine the most desirable forwarding path for data frames.



Each Layer 2 port on a switch running STP exists in one of these five port states:

- **Blocking:** In this state, the Layer 2 port is a nondesignated port and does not participate in frame forwarding. The port receives BPDUs to determine the location and root ID of the root switch and which port roles (root, designated, or nondesignated) each switch port should assume in the final active STP topology. By default, the port spends 20 seconds in this state (max age).
- **Listening:** In this state, spanning tree has determined that the port can participate in frame forwarding according to the BPDUs that the switch has received so far. At this point, the switch port is not only receiving BPDUs, it is also transmitting its own BPDUs and informing adjacent switches that the switch port is preparing to participate in the active topology. By default, the port spends 15 seconds in this state (forward delay).
- **Learning:** In this state, the Layer 2 port prepares to participate in frame forwarding and begins to populate the CAM table. By default, the port spends 15 seconds in this state (forward delay).
- **Forwarding:** In this state, the Layer 2 port is considered part of the active topology; it forwards frames and also sends and receives BPDUs.
- **Disabled:** In this state, the Layer 2 port does not participate in spanning tree and does not forward frames.

STP uses timers to determine how long to transition ports. STP also uses timers to determine the health of neighbor bridges and how long to cache MAC addresses in the bridge table.

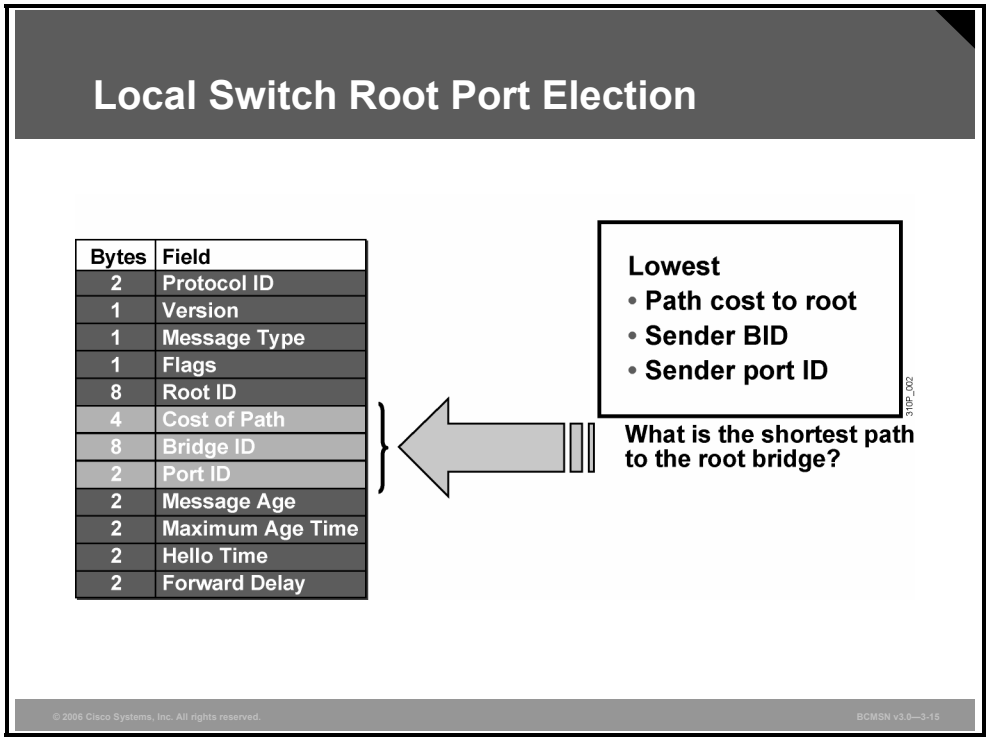
The explanation of the timers is as follows:

- **Hello timer:** 2 seconds. This timer is used to determine how often the root bridge sends configuration BPDUs.
- **Maximum Age (Max Age):** 20 seconds. This timer tells the bridge how long to keep ports in the blocking state before listening.
- **Forward Delay (Fwd Delay):** 15 seconds. This timer determines how long to stay in the listening state before learning, and in the learning state before forwarding.

The STP timers can be tuned based on network size. These parameters are designed to give STP ample opportunity to ensure a loop-free topology. Mistuning these parameters can cause serious network instability. When a bridge sees BPDUs with a better path to the root, it recalculates STP. This approach allows ports to transition when appropriate.

Forming an Association with the Root Bridge

This subtopic identifies methods by which switch ports determine their role in STP.



Nonroot bridges place various ports in their proper roles by listening to BPDUs as they come in on all ports. Receiving BPDUs on multiple ports indicates a redundant path to the root bridge.

The switch looks at these components in the BPDU to determine which switch ports will forward data and which switch ports will block data:

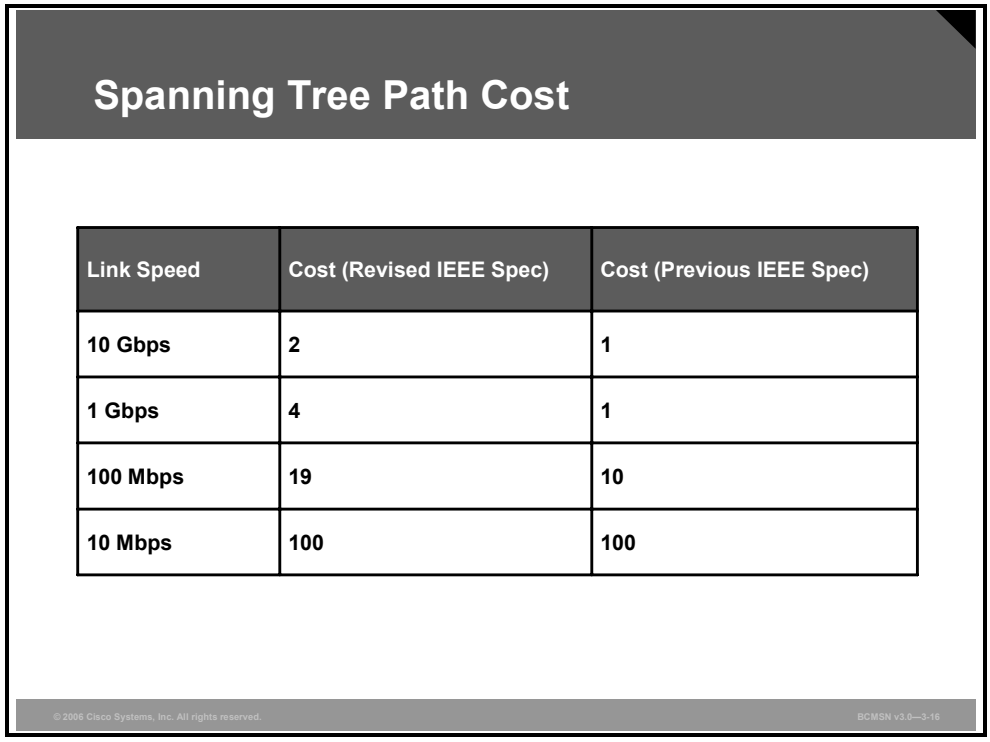
- Lowest path cost
- Lowest sender BID
- Lowest sender port ID

The switch looks at the path cost first to determine which port is receiving the lowest-cost path. The path is calculated on the basis of the link speed and the number of links the BPDU traversed. If a port has the lowest cost, that port is eligible to be placed in forwarding mode. All other ports that are receiving BPDUs continue in blocking mode.

If the path cost and sender BID are equal, as with parallel links between two switches, the switch goes to the port ID as a "tiebreaker." The port with the lowest port ID forwards data frames, and all other ports continue to block data frames.

Path Cost

This subtopic identifies the forwarding path between a device and the root bridge.

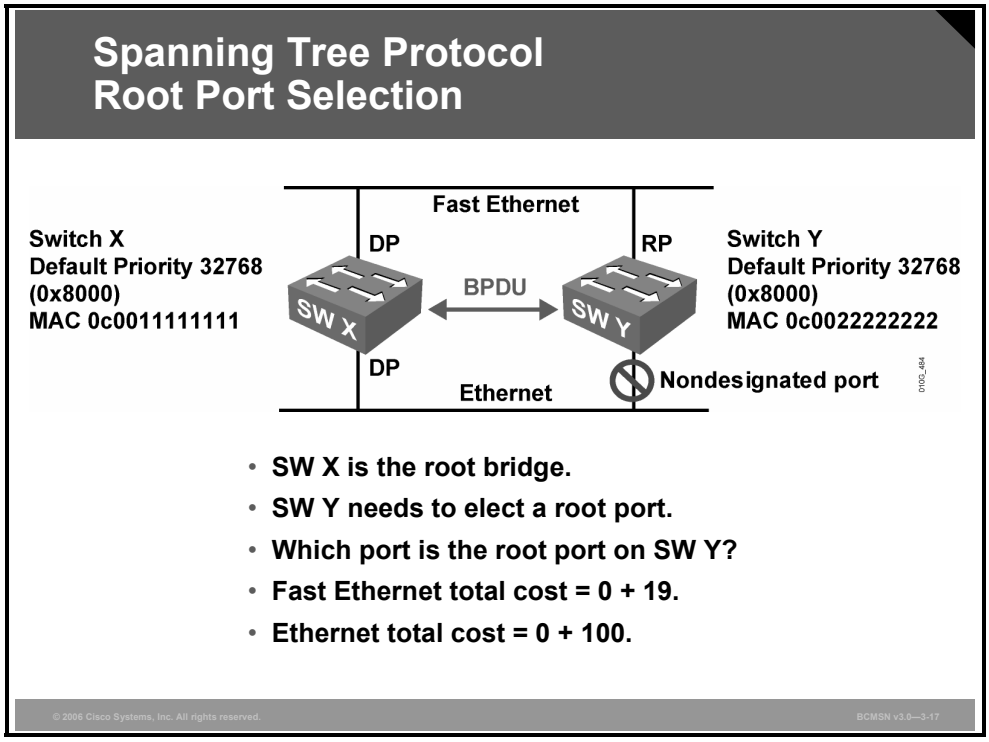


The spanning tree path cost is a value advertised in the BPDU by each bridge. This is a value representing the cumulative cost of all the links from the root bridge to the switch sending the BPDU. Path cost value is used by the receiving switch to determine the best path to the root bridge. The lowest cost is considered to be the best path.

Port cost values per link are shown in the table under the “Revised IEEE Spec” heading, with the lower values being associated with higher bandwidth and therefore being the more desirable paths. This new specification uses a nonlinear scale with port cost values as shown. In the previous IEEE specification, the cost value was calculated based on Gigabit Ethernet being the maximum Ethernet bandwidth, with an associated value of 1, from which all other values were derived in a linear manner.

Selecting the Root Port

This subtopic identifies how a switch port is selected as the root port.

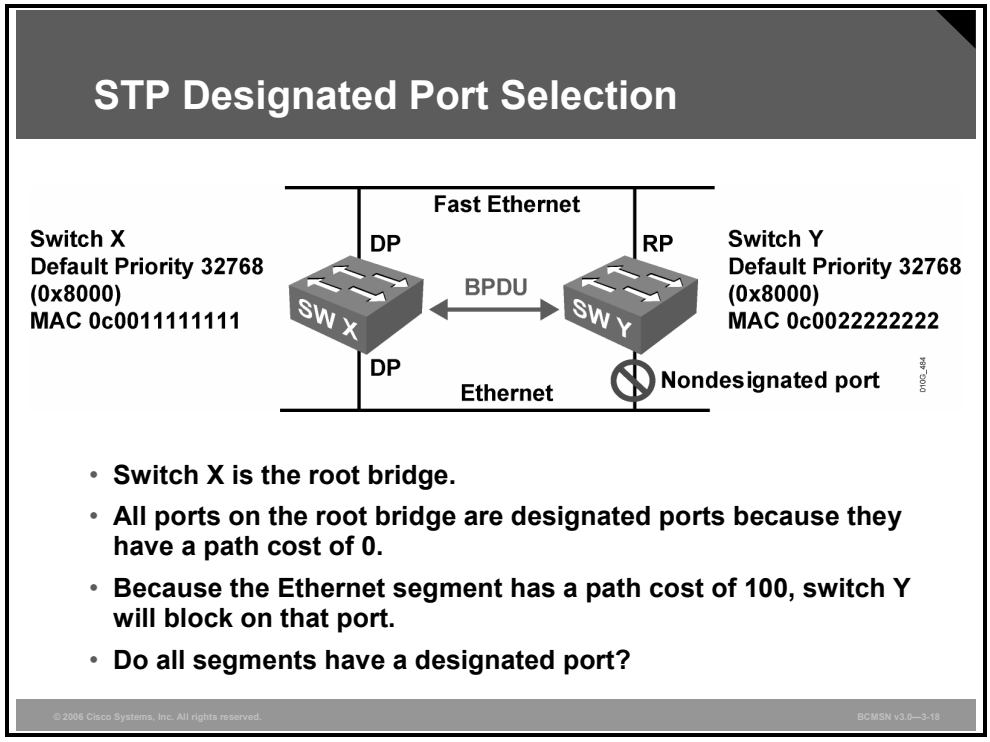


Switch Y receives a BPDU from the root bridge (switch X) on its switch port on the Fast Ethernet segment and another BPDU on its switch port on the Ethernet segment. The root path cost in both cases is zero.

The local path cost on the Fast Ethernet switch port is 19, whereas the local path cost on the Ethernet switch port is 100. As a result, the switch port on the Fast Ethernet segment has the lowest path cost to the root bridge and is elected the root port for switch Y.

Selecting the Designated Port

This subtopic identifies the features that apply to designated switch ports.



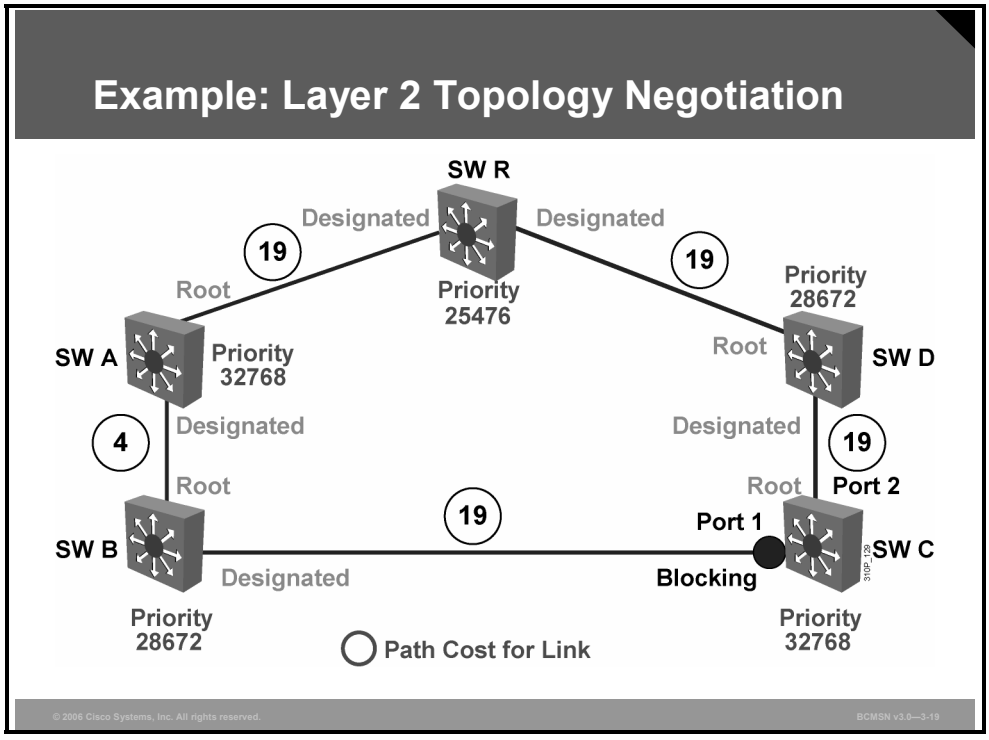
STP selects one designated port per segment to forward traffic. Other switch ports on the segment typically become nondesignated ports and continue blocking, or the switch port could be a root port and continue forwarding, as shown in the figure.

The nondesignated ports receive BPDUs but block data traffic and do not forward data traffic to prevent loops. The switch port on the segment with the lowest path cost to the root bridge is elected the designated port. If multiple switch ports on a switch have the same path cost and are connecting to the same neighbor switch, then the switch port with the lowest sender port ID becomes the designated port.

Because ports on the root bridge all have a root path cost of zero, all ports on the root bridge are designated ports.

Example: Determining the Active Topology

This subtopic identifies the features that apply to BPDU messaging.



Here is a scenario with switches running STP and exchanging information as shown in the figure. From this information, exchange will yield these final results:

- The election of a root bridge as a Layer 2 topology point of reference
- The determination of the best path to the root bridge from each switch
- The election of a designated switch and corresponding designated port for every switched segment
- The removal of loops in the switched network by transitioning some switch links to a blocked state
- Determination of the "active topology" for each instance or VLAN running STP

The active topology is the final set of communication paths that are created by switch ports forwarding frames. After the active topology has been established, using topology change notifications (TCNs), the switched network must reconfigure the activity topology if a link failure occurs.

Topology Changes in STP

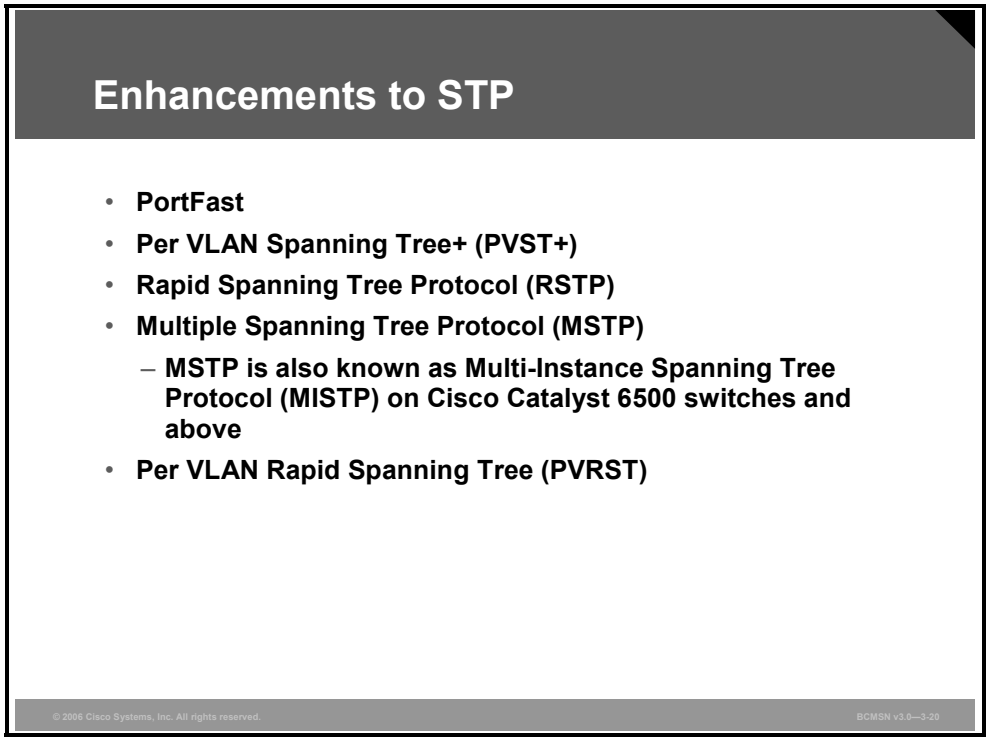
The other type of STP BPDU that needs to be discussed is TCN. The TCN BPDU is generated when a bridge discovers a change in topology, usually because of a link failure, bridge failure, or a port transitioning to forwarding state.

The TCN BPDU is set to 0x80 in the Type field and is subsequently forwarded on the root port toward the root bridge. The upstream bridge responds with acknowledgment of the BPDU in the form of topology change acknowledgment (TCA). The least significant bit is for TCN, and the most significant bit is for TCA in the Flag field.

The bridge sends this message to its designated bridge. Remember, the designated bridge is the closest neighbor to the root of a particular bridge (or the root, if it is directly connected). The designated bridge acknowledges the topology change back to the sending neighbor and sends the message to its designated bridge. This process repeats until the root bridge gets the message. The root learns about the topology changes in the network in this way.

Explaining Enhancements to STP

This topic describes PortFast, PVST+, RSTP, MSTP, and PVRST.



The 802.1D STP standard was developed long before VLANs were introduced. The 802.1D standard has limitations that can be improved by the Cisco Systems proprietary PVST, which allows separate instances of spanning tree and Cisco proprietary features such as PortFast and UplinkFast, which provide much faster convergence.

802.1Q has defined standards-based technologies for handling VLANs. To reduce the complexity of this standard, the 802.1 committee specified only a single instance of spanning tree for all VLANs. Not only does this provide a considerably less flexible approach than the PVST adopted by Cisco, it creates an interoperability problem.

To address both these issues, Cisco introduced the Per VLAN Spanning Tree+ (PVST+) protocol in 4.1 code on the Cisco Catalyst 5000 Series. (All Cisco Catalyst 4000 and 6000 series switches support PVST+.) This feature allows the two schemes to interoperate in a seamless and transparent manner in almost all topologies and configurations.

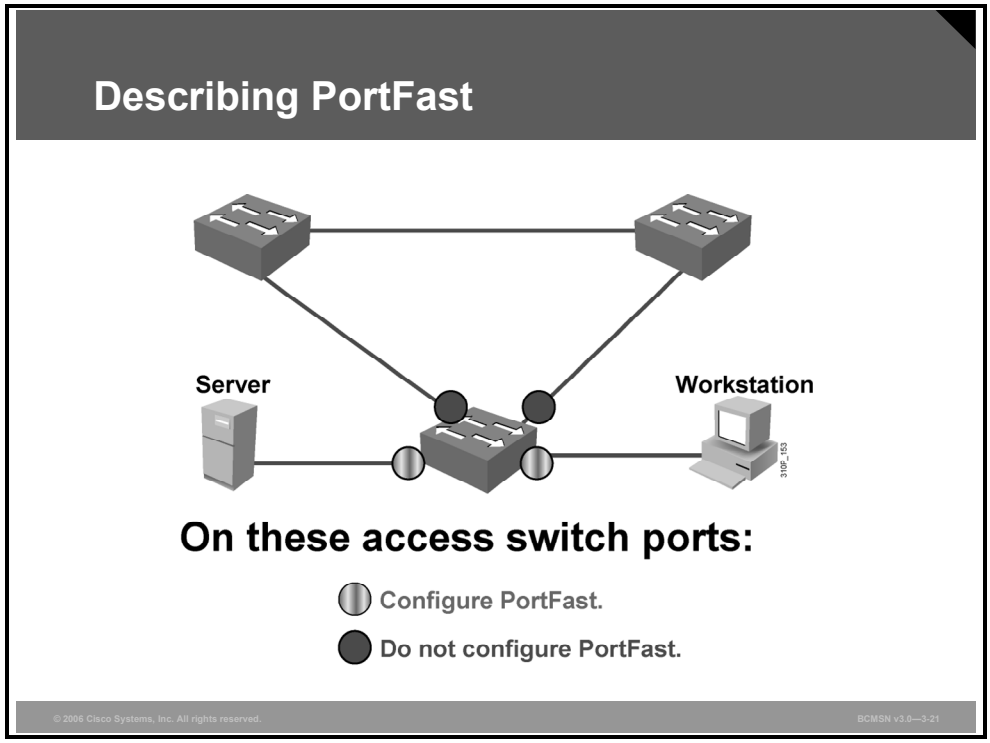
There are both advantages and disadvantages to using a single spanning tree. An advantage is that it allows switches to be simpler in design and place a lighter load on the CPU. A disadvantage is that a single spanning tree precludes load balancing and can lead to incomplete connectivity in certain VLANs (the single STP VLAN might select a link that is not included in other VLANs). Given these tradeoffs, most network designers have concluded that the disadvantages of having one spanning tree outweigh the benefits.

Two new IEEE standards, RSTP (802.1w) and MSTP (802.1s), improve on the original 802.1D STP standard. They provide similar functionality to the Cisco proprietary features. Rapid Spanning Tree Protocol (RSTP) provides much faster convergence, and Multiple Spanning Tree Protocol (MSTP) allows for multiple instances of spanning tree.

Per VLAN Rapid Spanning Tree (PVRST) allows Rapid Spanning Tree (RST) to be implemented, giving faster convergence, while still using the Cisco proprietary PVST.

Describing PortFast

This subtopic identifies the features of PortFast.



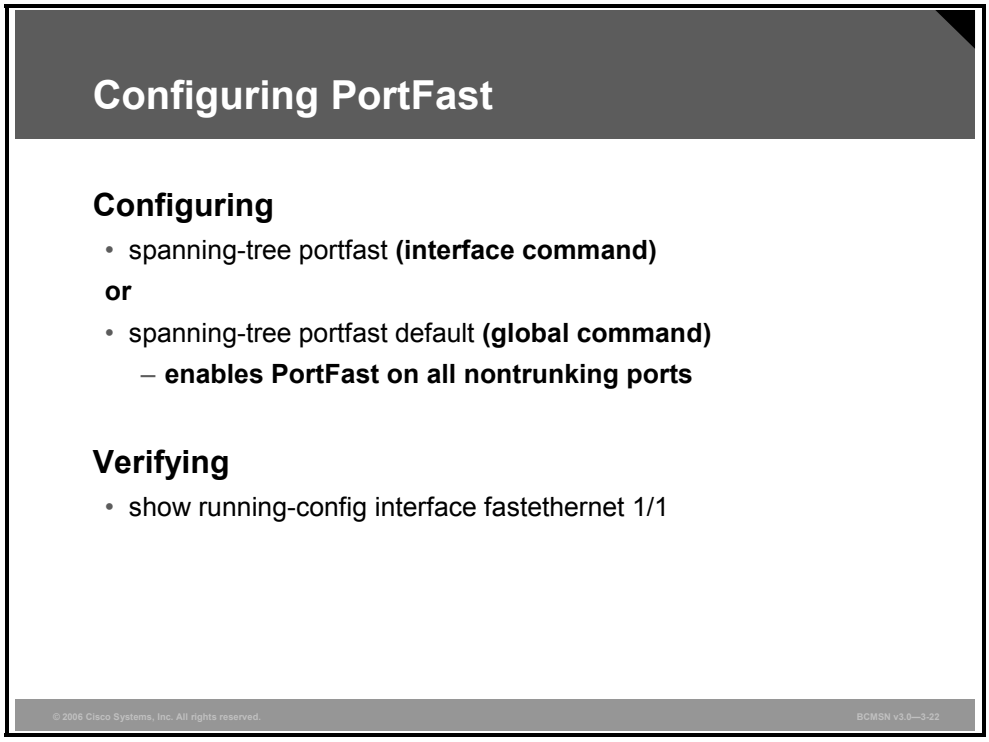
Spanning tree PortFast causes an interface configured as a Layer 2 access port to transition from blocking to forwarding state immediately, bypassing the listening and learning states. You can use PortFast on Layer 2 access ports that are connected to a single workstation or to a server to allow those devices to connect to the network immediately rather than waiting for spanning tree to converge.

If an interface that is configured with PortFast receives a BPDU, then spanning tree can put the port into the blocking state by using a feature called BPDU guard.

Caution	Because the purpose of PortFast is to minimize the time that access ports must wait for spanning tree to converge, it should be used only on access ports. If you enable PortFast on a port connecting to another switch, you risk creating a spanning tree loop.

Configuring PortFast

This subtopic identifies the commands used to configure port-level tuning with PortFast.



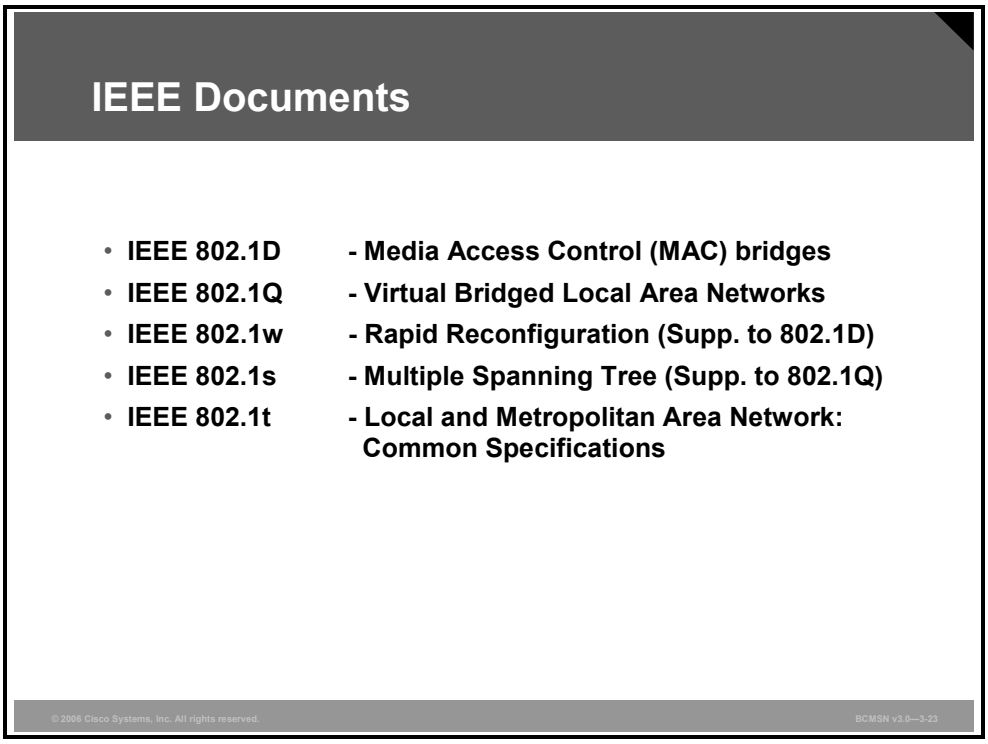
PortFast Commands

The table lists the commands used to implement and verify PortFast on an interface.

Argument	Description
`Switch(config-if)#` **`spanning-tree portfast`**	Enables PortFast on a Layer 2 access port and forces it to enter the forwarding state immediately.
`Switch(config-if)#` **`no spanning-tree portfast`**	Disables PortFast on a Layer 2 access port. PortFast is disabled by default.
`Switch(config)#`**`spanning-tree portfast default`**	Globally enable the PortFast feature on all nontrunking ports. When the PortFast feature is enabled, the port changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes.
`Switch#` **`show running-config interface`** *`type slot/port`*	Indicates if PortFast has been configured on a port. It can also be used to show if configuration has occurred on an EtherChannel link by specifying *port-channel and channel_number* in the place of *type slot/port.*

IEEE Documents

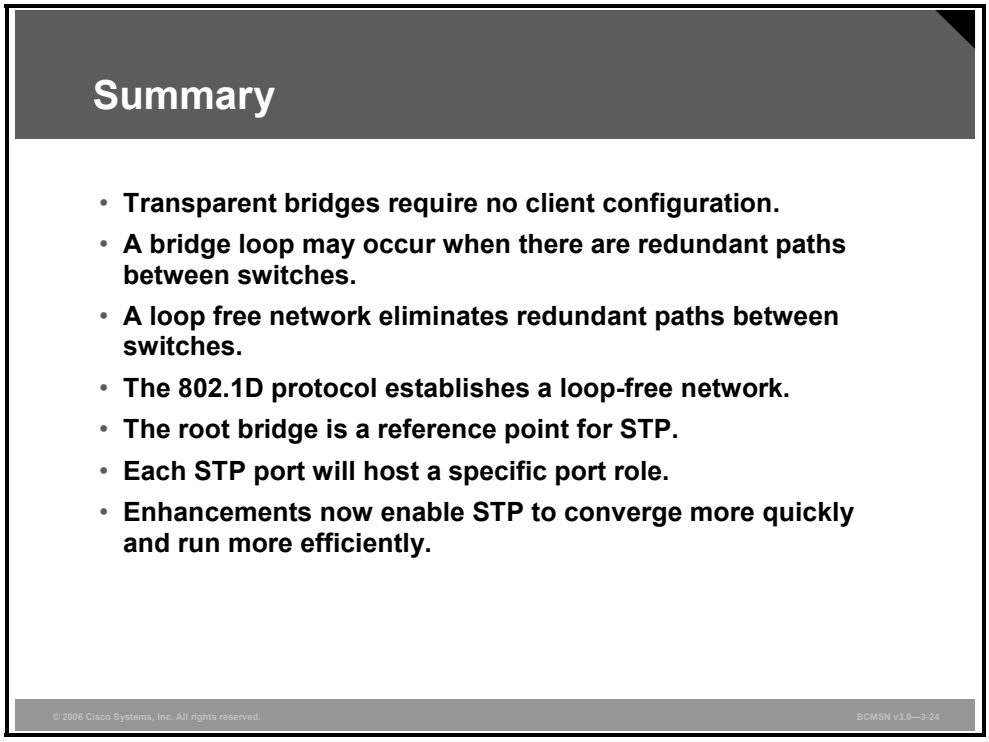
This subtopic lists the IEEE documents related to this lesson.



The documents listed are available on the IEEE Web site, http://www.ieee.org.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 2

Implementing RSTP

Overview

Rapid Spanning Tree Protocol (RSTP) is an improvement on the original 802.1D Spanning Tree Protocol (STP) standard. RSTP provides much faster convergence when topology changes (TCs) occur in a switched network. Through the use of specific port states, port roles, and link types, RSTP very quickly adapts to network topology transitions.

A proposal and agreement process between neighbor switches is unique to RSTP. Also, topology change notifications (TCNs) are transferred in a very different manner than they are in 802.1D STP operation. Configuration of RSTP is much the same as in 802.1D, except for a few variations and identifiable characteristics in the spanning tree verification commands.

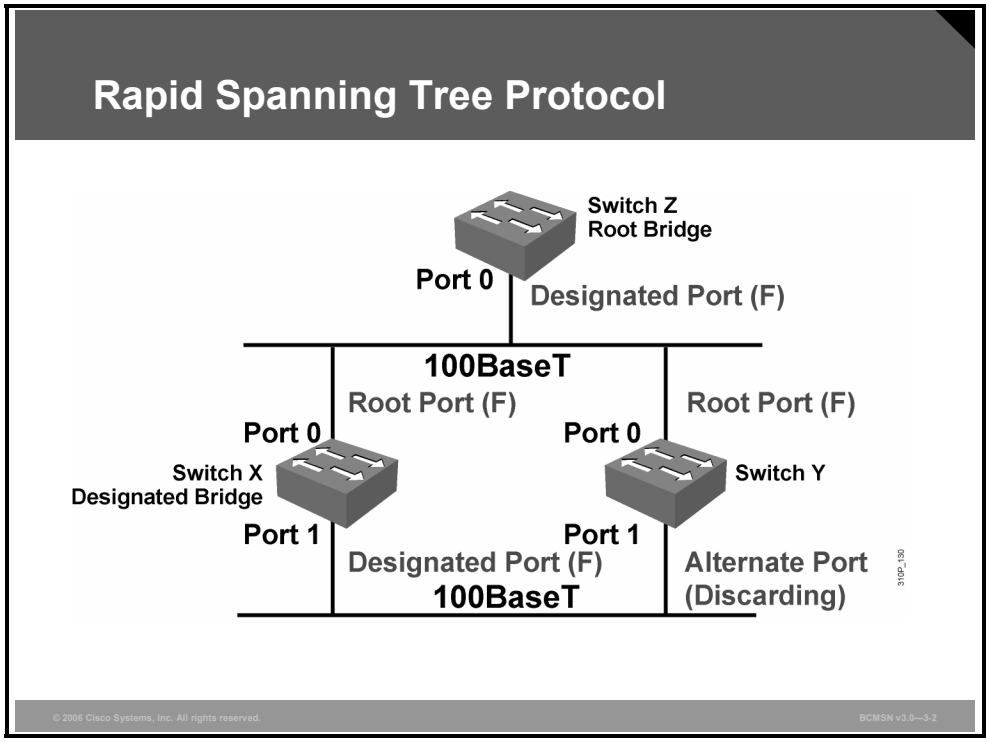
Objectives

Upon completing this lesson, you will be able to describe RSTP and the procedure for implementing it in an existing network. This ability includes being able to meet these objectives:

- Describe the RSTP
- Describe the three RSTP port states
- Describe the five different RSTP port roles
- Explain an edge port
- Describe the function of the different RSTP link types
- Differentiate the 802.1w use of the BPDU from 802.1D
- Describe the stages of the RSTP proposal and agreement process
- Describe the process that RSTP uses to notify all bridges in the network of a TC
- Describe the commands used to implement RSTP
- Explain the procedure to implement RSTP in a switched network

Describing the RSTP

This topic describes the RSTP.



The immediate hindrance of STP is convergence. Depending on the type of failure, it takes anywhere from 30 to 50 seconds to converge the network. RSTP helps with convergence issues that plague legacy STP. RSTP has additional features similar to UplinkFast and BackboneFast that offer better recovery at Layer 2 compared to STP.

RSTP is based on IEEE 802.1w standard. Numerous differences exist between RSTP and STP. RSTP requires full-duplex point-to-point connection between adjacent switches to achieve fast convergence. Half duplex, generally speaking, denotes a shared medium whereby multiple hosts share the same wire; a point-to-point connection cannot reside in this environment. As a result, RSTP cannot achieve fast convergence in half-duplex mode.

STP and RSTP also have port designation differences. RSTP has alternate and backup port designations, which are absent from the STP environment. Ports that are not participating in spanning tree are known as edge ports. Edge ports can be statically configured or will be recognized by the PortFast parameter. The edge port becomes a nonedge port immediately if a bridge protocol data unit (BPDU) is heard on the port. Nonedge ports participate in the spanning tree algorithm; hence, only nonedge ports generate TCs on the network when transitioning to forwarding state only. TCs are not generated for any other RSTP states. In legacy STP, TCNs were generated for any active port that was not configured for PortFast.

RSTP speeds the recalculation of the spanning tree when the Layer 2 network topology changes. It is an IEEE standard that redefines STP port roles, states, and BPDUs.

RSTP is proactive and therefore negates the need for the 802.1D delay timers. RSTP (802.1w) supersedes 802.1D, while still remaining backward compatible. Much of the 802.1D

terminology remains, and most parameters are unchanged. In addition, 802.1w is capable of reverting back to 802.1D to interoperate with legacy switches on a per-port basis.

The RSTP BPDU format is the same as the IEEE 802.1D BPDU format, except that the Version field is set to 2 to indicate RSTP, and the Flags field makes use of all 8 bits.

In a switched domain, there can be only one forwarding path toward a single reference point; this is the root bridge. The RSTP spanning tree algorithm (STA) elects a root bridge in exactly the same way as 802.1D elects a root.

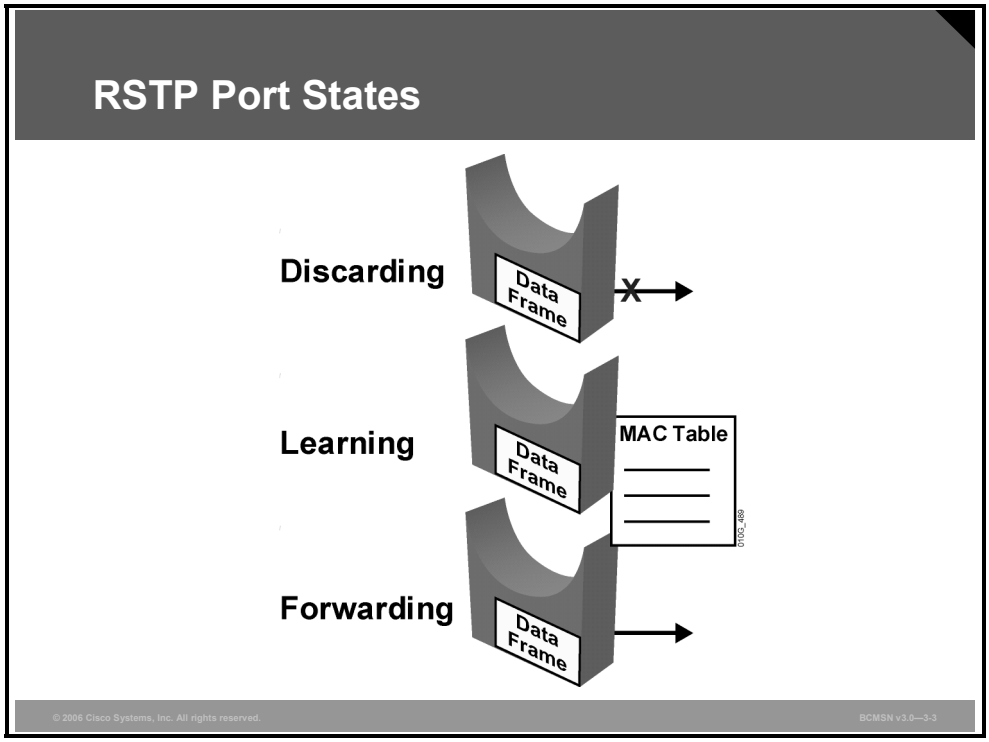
However, there are critical differences that make RSTP the preferred protocol for preventing Layer 2 loops in a switched network environment. Many of the differences stem from the Cisco Systems proprietary enhancements. The Cisco RSTP enhancements have these characteristics:

- They are integrated into the protocol at a low level.
- They are transparent.
- They require no additional configuration.
- They generally perform better than the Cisco proprietary 802.1D enhancements.
- BPDU carries information about port roles and is sent to neighbor switches only.

Because the RSTP and the Cisco proprietary enhancements are functionally similar, features such as UplinkFast and BackboneFast are not compatible with RSTP.

Describing RSTP Port States

This topic describes the three RSTP port states.



RSTP provides rapid convergence following the failure or re-establishment of a switch, switch port, or link. An RSTP TC will cause a transition in the appropriate switch ports to the forwarding state through either explicit handshakes or a proposal and agreement process and synchronization.

With RSTP, the role of a port is separated from the state of a port. For example, a designated port could be in the discarding state temporarily, even though its final state is to be forwarding.

The RTSP port states correspond to the three basic operations of a switch port: discarding, learning, and forwarding.

Characteristics of Port States

The table describes the characteristics of RSTP port states.

Port State	Description
Discarding	This state is seen in both a stable active topology and during topology synchronization and changes. The discarding state prevents the forwarding of data frames, thus “breaking” the continuity of a Layer 2 loop.
Learning	This state is seen in both a stable active topology and during topology synchronization and changes. The learning state accepts data frames to populate the MAC table in an effort to limit flooding of unknown unicast frames.
Forwarding	This state is seen only in stable active topologies. The forwarding switch ports determine the topology. Following a TC, or during synchronization, the forwarding of data frames occurs only after a proposal and agreement process.

In all port states, a port will accept and process BPDU frames.

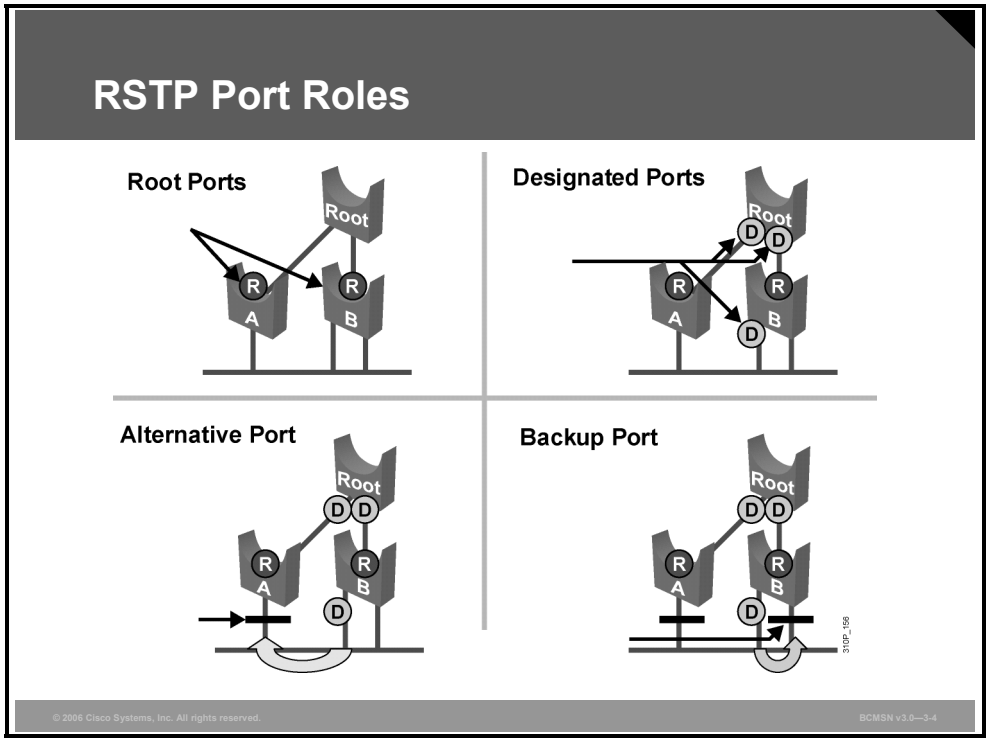
Ports States Between STP and RSTP

The table describes STP and RSTP port states.

Operational Port State	STP Port State	RSTP Port State
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding
Disabled	Disabled	Discarding

Describing RSTP Port Roles

This topic describes the different RSTP port roles.



The port role defines the ultimate purpose of a switch port and the way it handles data frames. Port roles and port states are able to transition independently of each other. RSTP uses these definitions for port roles.

Port Role Definitions

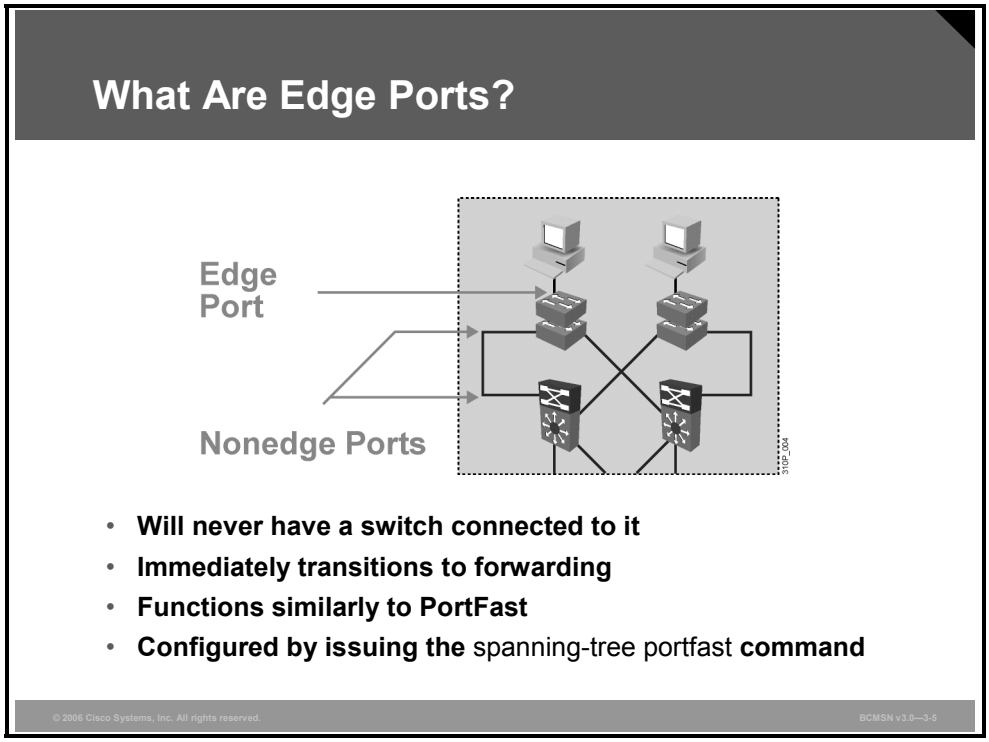
The table defines port roles.

Port Role	Description
Root port	The root port is the switch port on every nonroot bridge that is the chosen path to the root bridge. There can be only one root port on every switch. The root port assumes the forwarding state in a stable active topology.
Designated port	Each segment has at least one switch port as the designated port for that segment. In a stable, active topology, the switch with the designated port receives frames on the segment that are destined for the root bridge. There can be only one designated port per segment. The designated port assumes the forwarding state. All switches that are connected to a given segment listen to all BPDUs and determine the switch that will be the designated switch for a particular segment.
Alternative port	The alternative port is a switch port that offers an alternative path toward the root bridge. The alternative port assumes a discarding state in a stable, active topology. An alternative port is present on nondesignated switches and makes a transition to a designated port if the current designated path fails.
Backup port	The backup port is an additional switch port on the designated switch with a redundant link to the segment for which the switch is designated. A backup port has a higher port ID than the designated port on the designated switch. The backup port assumes the discarding state in a stable, active topology.

Establishing the additional port roles allows RSTP to define a standby switch port before a failure or TC. The alternative port moves to the forwarding state if there is a failure on the designated port for the segment.

Explaining Edge Ports

This topic explains how edge ports function.



An RSTP edge port is a switch port that is never intended to be connected to another switch device. It immediately transitions to the forwarding state when enabled.

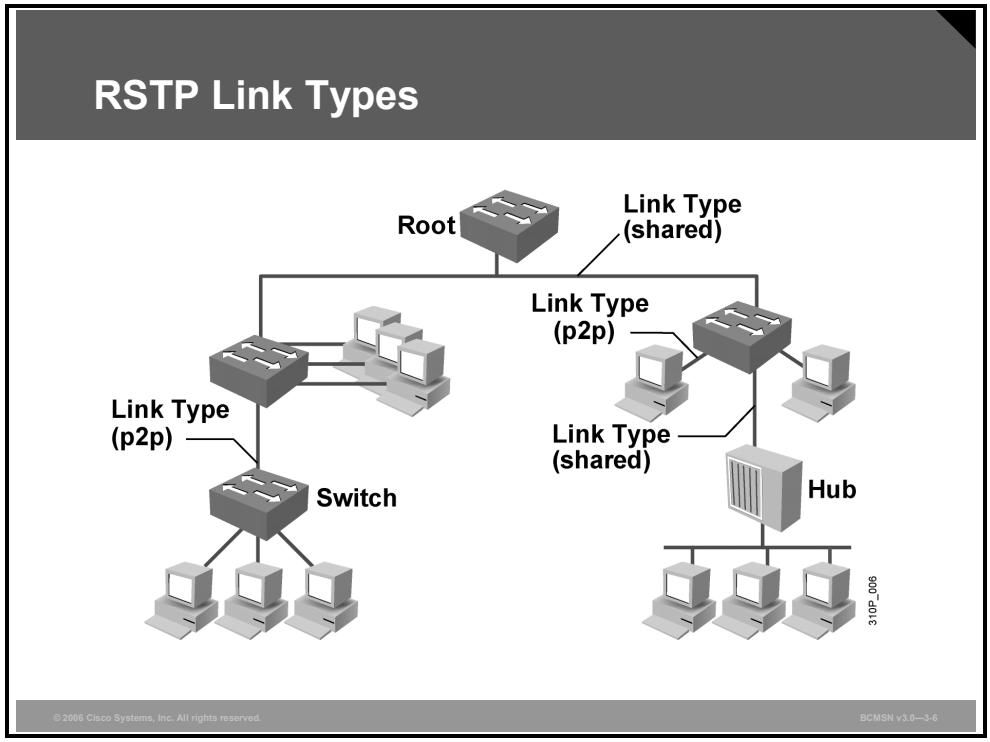
The edge port concept is well known to Cisco spanning tree users because it corresponds to the PortFast feature. All ports that are directly connected to end stations anticipate that no switch device will be connected to them, and so they immediately transition to the STP forwarding state, thereby skipping the time-consuming listening and learning stages. Neither edge ports nor PortFast-enabled ports generate TCs when the port transitions to a disabled or enabled status.

Unlike PortFast, an edge port that receives a BPDU immediately loses its edge port status and becomes a normal spanning tree port. When an edge port receives a BPDU, it generates a TCN.

The Cisco RSTP implementation maintains the PortFast keyword for edge port configuration, thus making an overall network transition to RSTP more seamless. Configuring an edge port where the port will be attached to another switch can have negative implications for RSTP when it is in the “sync” state.

Describing RSTP Link Types

This topic describes the function of the different RSTP link types.



Link type provides a categorization for each port participating in RSTP. The link type can predetermine the active role that the port plays as it stands by for immediate transition to a forwarding state, if certain parameters are met. These parameters are different for edge ports and nonedge ports. Nonedge ports are categorized into two link types. Link type is automatically determined but can be overwritten with an explicit port configuration.

RSTP Link Types

The table defines link types.

Link Type	Description
Point-to-point	Port operating in full-duplex mode. It is assumed that the port is connected to a single switch device at the other end of the link.
Shared	Port operating in half-duplex mode. It is assumed that the port is connected to shared media where multiple switches might exist.

Edge ports, the equivalent of PortFast-enabled ports, and point-to-point links are candidates for rapid transition to a forwarding state. Before the link type parameter can be considered for the purpose of expedient port transition, RSTP must determine the port role.

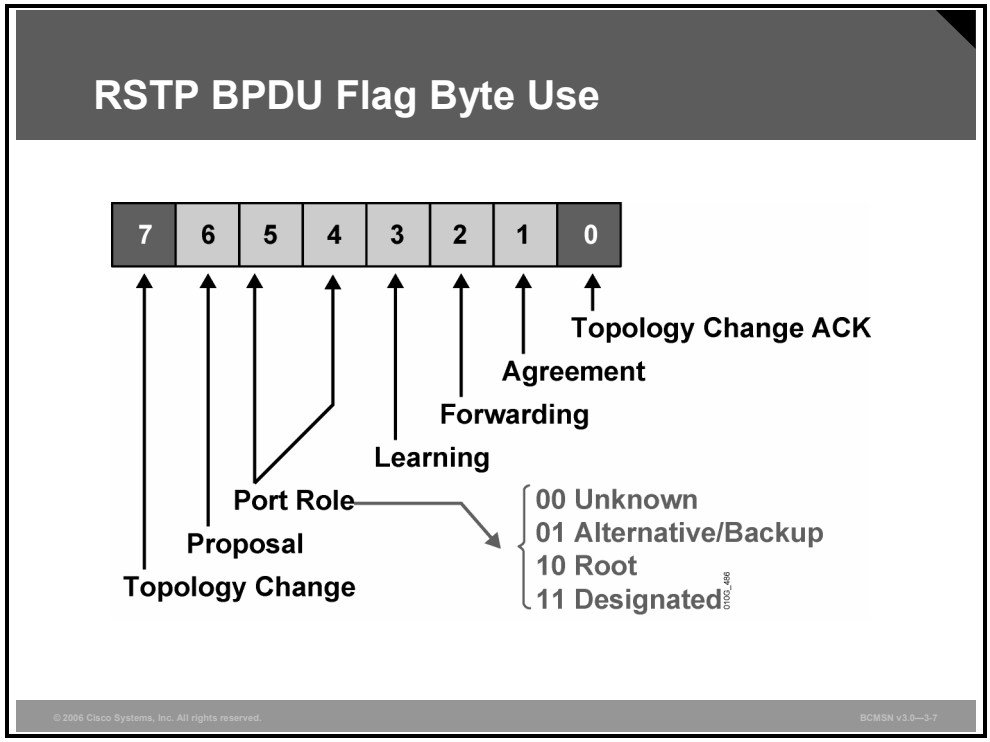
Root ports: Do not use the link type parameter. Root ports are able to make a rapid transition to the forwarding state as soon as the port is in "sync."

Alternative and backup ports: Do not use the link type parameter in most cases.

Designated ports: Make the most use of the link type parameter. Rapid transition to the forwarding state for the designated port occurs only if the link type parameter indicates a point-to-point link.

Examining the RSTP BPDU

This topic differentiates the 802.1w use of the BPDU from 802.1D.



RSTP (802.1w) uses type 2, version 2 BPDUs, so an RSTP bridge can communicate with 802.1D on any shared link or with any switch running 802.1D. RSTP sends BPDUs and populates the flag byte in a slightly different manner than the manner used by 802.1D.

- An RSTP bridge sends a BPDU with its current information every hellotime period (2 seconds by default), even if it does not receive any BPDUs from the root bridge.
- Protocol information can be immediately aged on a port if hellos are not received for three consecutive hellotimes or if the max age timer expires.
- Because BPDUs are now used as a “keepalive” mechanism, three consecutively missed BPDUs indicate lost connectivity between a bridge and its neighboring root or designated bridge. This fast aging of the information allows quick failure detection.

RSTP uses the flag byte of version 2 BPDU as shown in the figure.

- Bits 0 and 7 are used for TCN and acknowledgement (ACK), as they are in 802.1D.
- Bits 1 and 6 are used for the proposal and agreement process.
- Bits 2–5 encode the role and state of the port originating the BPDU.

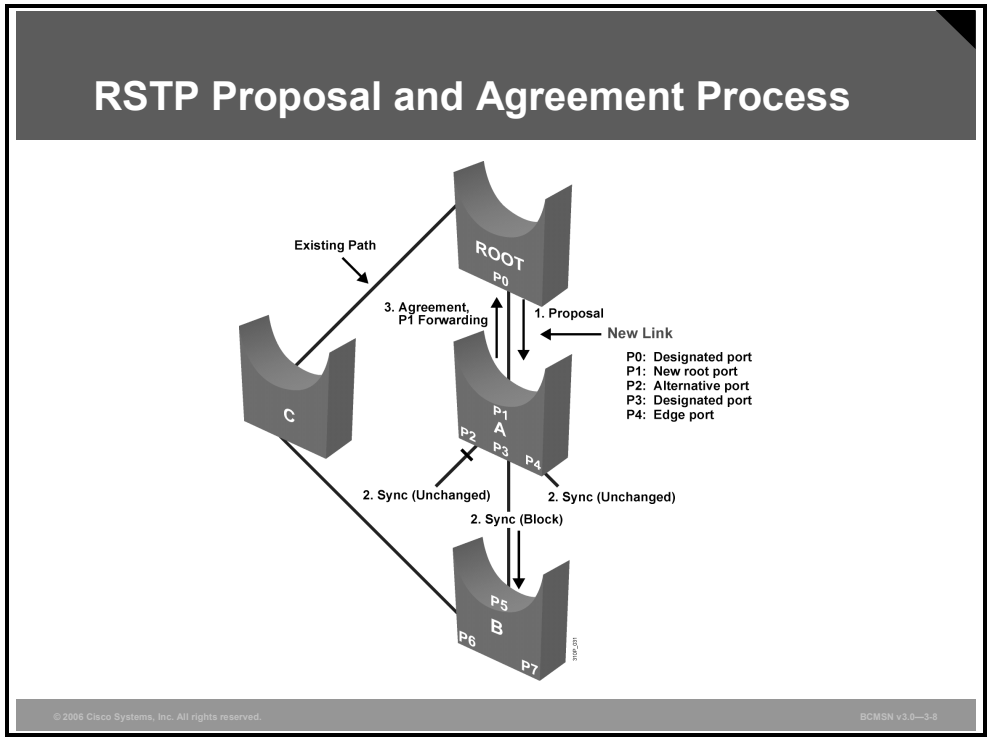
The Flag field in the STP BPDU packet contained TCN and TCA. In RSTP, the Flag field, 1 byte long, has been modified to accommodate port designations and proposal/agreement between adjacent switches. BPDUs are sent every 2 seconds. Unlike in legacy STP, in RSTP, each switch generates its own BPDUs, regardless of whether or not it hears BPDUs from the root.

In legacy STP, BPDUs were generated by only the root and propagated throughout the spanning tree domain. As a result, when a switch did not receive a configuration BPDU, it did not know where the failure occurred.

In RSTP mode, the switch needs direct interactions with only its immediate neighbors. Hence, BPDUs also serve as keepalive mechanisms between adjacent switches. If the switch does not hear three consecutive BPDUs from its downstream neighbor, it will transition appropriate ports to converge the network.

Identifying the RSTP Proposal and Agreement Process

This topic describes the stages of the RSTP proposal and agreement process.



In 802.1D, when a port has been selected by spanning tree to become a designated port, it must wait two times the forward delay before transitioning the port to a forwarding state. RSTP significantly speeds up the recalculation process after a TC in the network because it converges on a link-by-link basis and does not rely on timers expiring before ports can transition. Rapid transition to a forwarding state can be achieved on only edge ports and point-to-point links. In RSTP, this condition corresponds to a port with a designated role that is in a blocking state. The figure illustrates, step by step, how rapid transition is achieved.

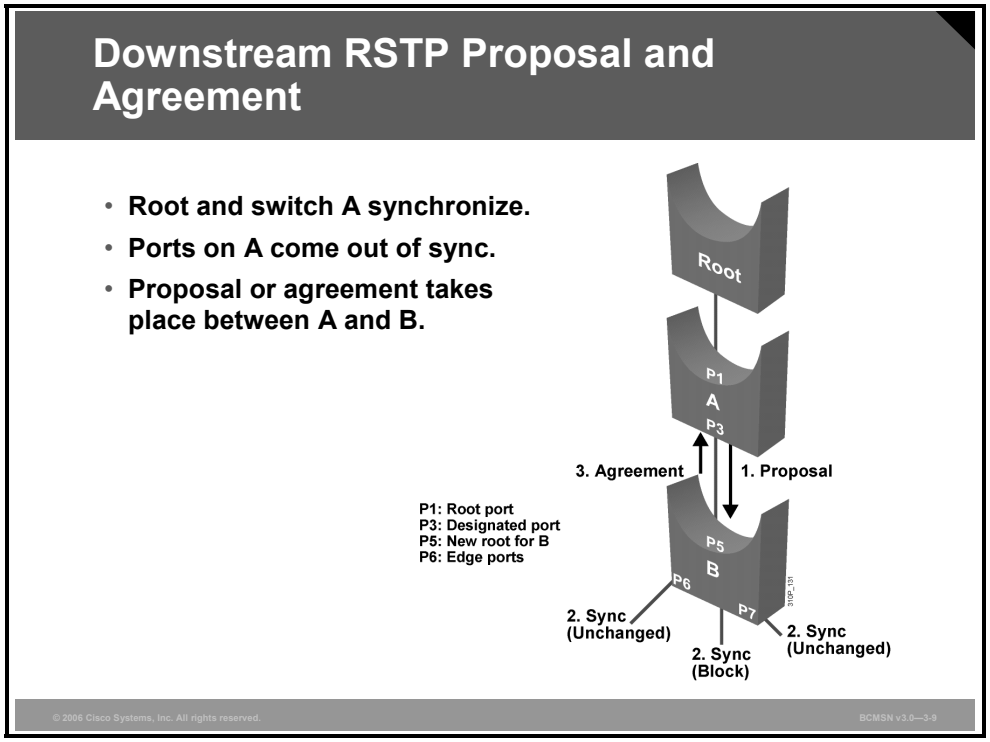
1. Switch A had a path to the root via switch B and switch C. A new link is then created between the root and switch A, and both ports are in designated blocking state until they receive a BPDU from their counterparts. When a designated port is in a discarding or learning state (and only in this case), it sets the proposal bit on the BPDUs it sends out. This is what happens for port P0 of the root bridge.

2. Switch A sees the proposal BPDU with a superior path cost. It blocks all nonedge designated ports other than the one over which the proposal and agreement process are occurring. This operation, called “sync,” prevents switches below A from causing a loop during the proposal and agreement process. Edge ports need not be blocked and remain unchanged during sync.

3. Bridge A explicitly sends an agreement that allows the root bridge to put the root port P0 in forwarding state. Port P1 becomes the root port for A.

Downstream RSTP Proposal Process

This subtopic discusses the steps in RSTP proposal acknowledgement downstream from the root bridge.

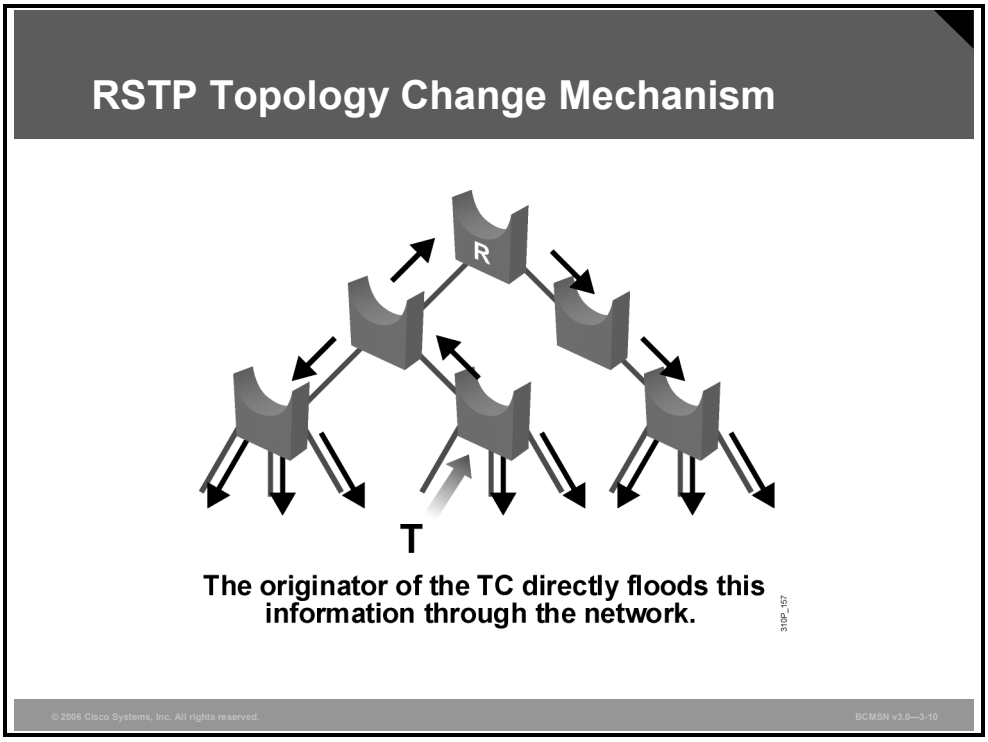


After switch A and the root bridge are synchronized, the proposal and agreement process continues on switch A out of all of its downstream-designated, nonedge ports, as shown in the figure.

1. Switch B on P5 will see that switch A is discarding and will also transition to the designated discarding state. Switch A then sends its proposal BPDU down to B with the root ID of the root bridge.
2. Switch B sees a proposal with the superior BPDU from A and blocks all nonedge-designated ports other than the one over which the proposal and agreement process is occurring.
3. Switch B sends a BPDU with the agreement bit set, and switch A P3 transitions to forwarding state. The synchronization process continues with switches downstream from B.

Identifying the RSTP TCN Process

This topic describes the process that RSTP uses to notify all bridges in the network of a TC.



In 802.1D, any port state change generates a TCN. When an 802.1D bridge detects TC, it sends TCNs toward the root bridge. The root bridge sets the TC flag on the outbound BPDUs that are relayed to switches down from the root.

When a bridge receives a BPDU with the TC flag bit set, the bridge reduces its bridge-table aging time to forward delay seconds. This ensures a relatively quick flushing of the MAC address table.

In RSTP, only nonedge ports moving to the forwarding state cause a TC. Loss of connectivity is not considered to be a TC, and, under these conditions, a port moving to the blocking state does not generate a TC BDPU.

RSTP Actions

When an RSTP bridge detects a TC, it performs these actions.

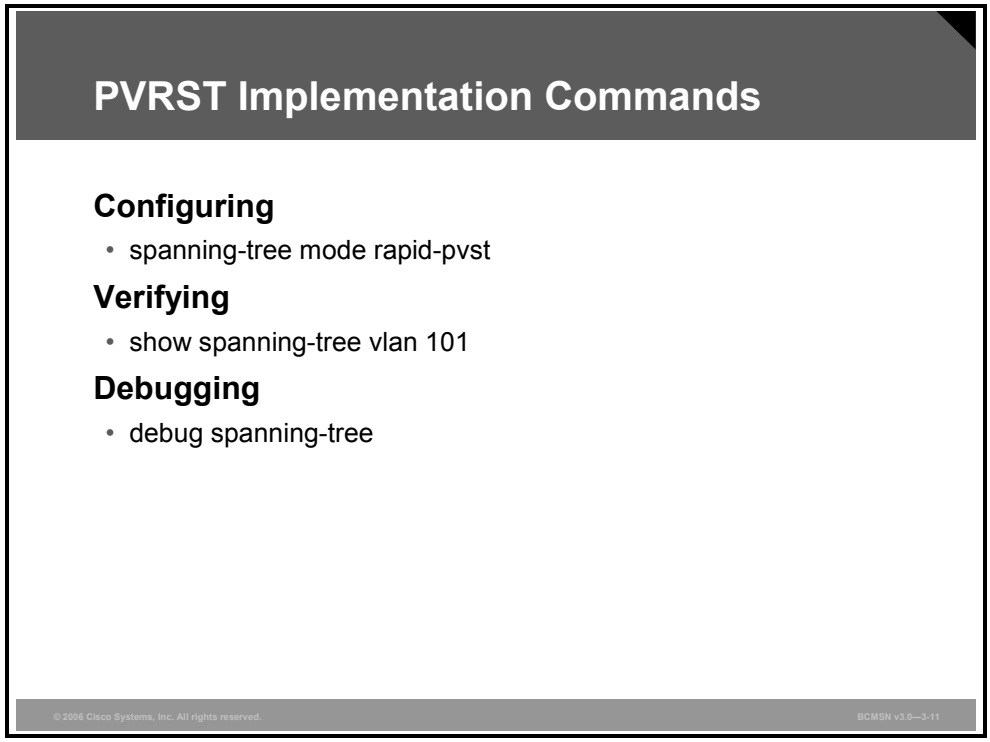
Step	Action	Notes
1.	The RSTP bridge starts the TC-While timer.	RTSP sets the TC-While timer with a value equal to twice the hellotime for all its nonedge designated ports and the root port, if necessary.
2.	The RSTP bridge flushes the MAC addresses associated with all these ports.	
3.	The TC flag bit is set on all outbound BPDUs.	BPDUs are sent on the root port as long as the TC-While timer is active.
4.	The bridge receives a BPDU with the TC bit set from a neighbor and clears the MAC addresses on all ports.	The port that received the TC BPDU retains learned MAC addresses.
5.	The bridge starts the TC-While timer and sends BPDUs, with a TC bit set, out of all its designated ports and root port.	RSTP does not use the specific TCN BPDU, unless a legacy bridge needs to be notified.

The TCN is flooded across the entire network, one switch at a time, from the switch that is the source of the change rather than from the root bridge. The TC propagation is now a one-step process. There is no need for each switch port to wait for the root bridge to be notified and then maintain the TC state for the value of the max age plus forward delay seconds.

If the port consistently keeps receiving BPDUs that do not correspond to the current operating mode for two periods of hellotime, the port switches to the mode that is indicated by the BPDUs.

Describing PVRST Implementation Commands

This topic describes the commands used to implement Per VLAN Rapid Spanning Tree (PVRST).



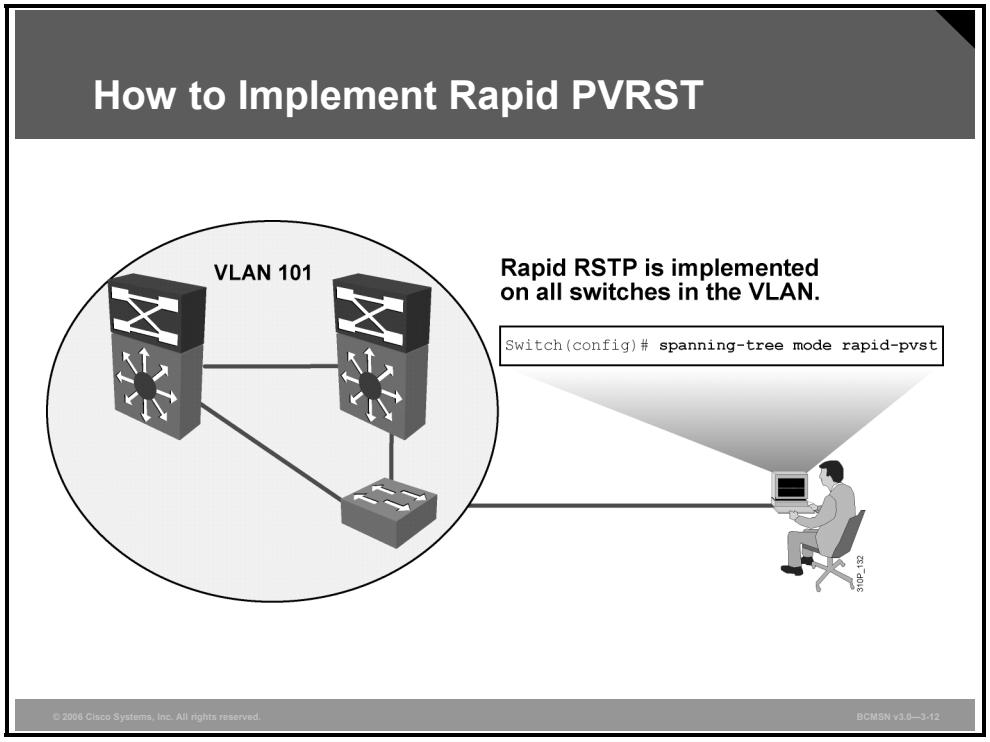
PVRST Commands

The table describes the commands that enable PVRST.

Command	Description
`Switch(config)#` **`spanning-tree mode rapid-pvst`**	Sets spanning tree mode to PVRST
`Switch#` **`show spanning-tree vlan`** *`vlan-number`* **`[detail]`**	Shows commands that are VLAN-based rather than instance-based
`Switch#` **`debug spanning-tree pvst+`**	Debugs PVST events
`Switch#` **`debug spanning-tree switch state`**	Debugs port state changes

Implementing PVRST Commands

This topic explains the procedure to implement RSTP in a switched network.



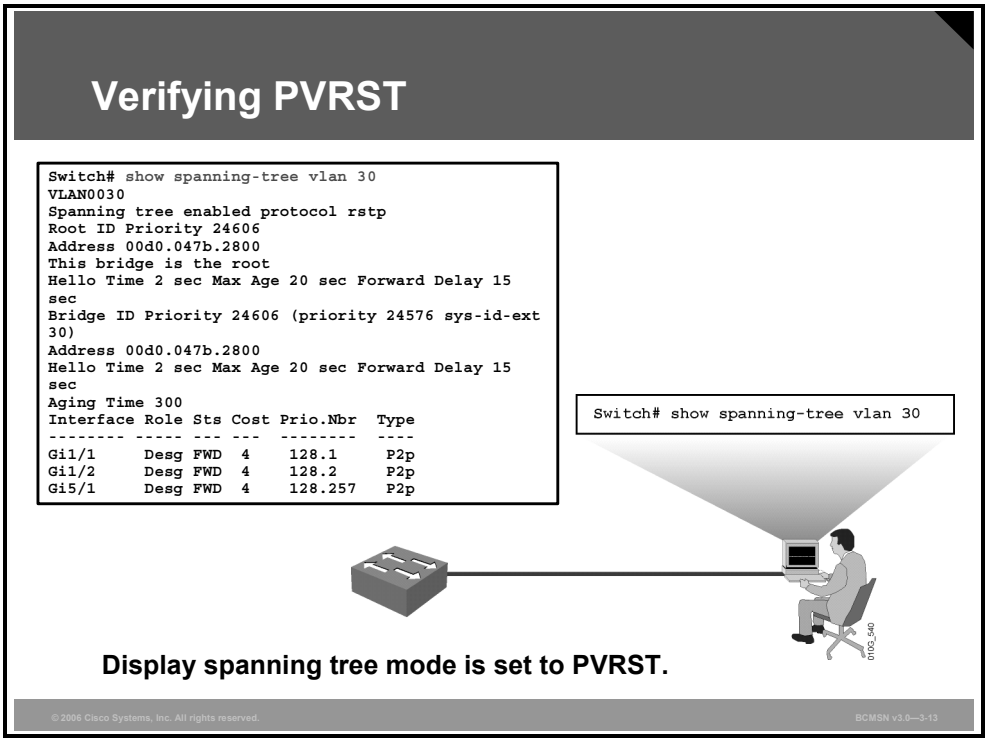
Configuring PVRST

The table describes how to configure PVRST.

Step	Description
1.	If spanning tree is disabled, enable it for a VLAN. `Switch(config)# spanning-tree vlan vlan-range`
2.	Set spanning tree mode to Rapid PVST+. Default is 802.1D (shows as “ieee”). `Switch(config)# spanning-tree mode rapid-pvst`

Verifying the PVRST Configuration

This subtopic identifies the commands that verify a rapid spanning tree (RST) configuration for a VLAN.



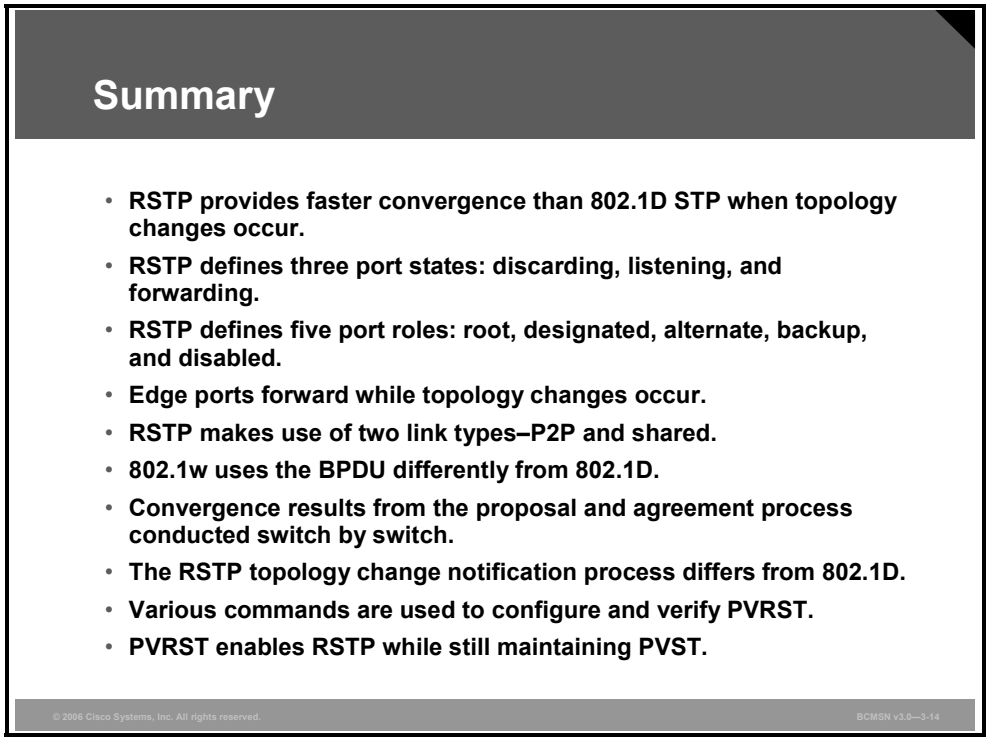
A variety of **show** commands can be used to display configuration and operation information about spanning tree. The **show spanning-tree** command takes several arguments to display a variety of information about the STP configuration. Without any arguments, it will display general information about all STP configurations. The complete syntax is as follows:

Switch#**show spanning-tree** [*bridge-group* | **active** | **backbonefast** | {**bridge** [*id*]}| **detail** | **inconsistentports** | {**interface** *interface interface-number*} | **root** | **summary** [**total**] | **uplinkfast** | {**vlan** *vlan-id*} | {**port-channel** *number*} | **pathcost-method**]

Refer to your software documentation for a complete explanation of each parameter.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 3

Implementing MSTP

Overview

Per VLAN Spanning Tree (PVST) creates a single instance of spanning tree for each VLAN in the network. This may impose a processing load on a switch when many VLANs are present. Multiple Spanning Tree Protocol (MSTP) reduces this loading by allowing a single instance of spanning tree to run for multiple VLANs. Specific configuration and verification steps must be followed to properly implement MSTP.

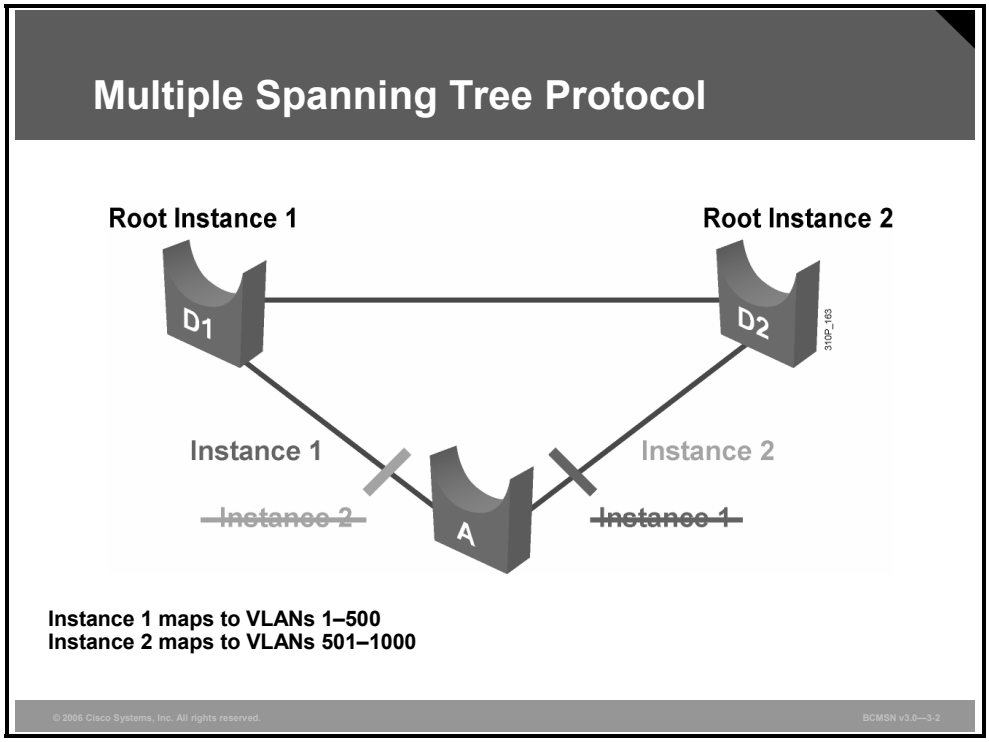
Objectives

Upon completing this lesson, you will be able to describe MSTP and the procedure for implementing it in an existing network. This ability includes being able to meet these objectives:

- Describe MSTP
- Describe the characteristics of an MST region
- Describe changes to the Bridge Priority field to accommodate the MSTP instance number
- Describe how MSTP operates with CST
- Describe the commands used to implement MSTP
- Explain the procedure to implement MSTP in a switched network

Explaining MSTP

This topic describes MSTP.



The main purpose of MSTP is to reduce the total number of spanning tree instances to match the physical topology of the network and thus reduce the CPU loading of a switch. The instances of spanning tree are reduced to the number of links (that is, active paths) that are available.

If the example in the diagram were implemented via Per VLAN Spanning Tree+ (PVST+), there could potentially be 4094 instances of spanning tree, each with its own bridge protocol data unit (BPDU) conversations, root bridge election, and path selections.

In this example, the goal is to achieve load distribution, with VLANs 1-500 using one path and VLANs 501-1000 using the other path, with only two instances of spanning tree. The two ranges of VLANs are mapped to two MSTP instances, respectively. Rather than maintaining 1000 spanning trees, each switch needs to maintain only two.

Implemented in this fashion, MSTP converges faster than PVST+ and is backward compatible with 802.1D STP, 802.1w Rapid Spanning Tree Protocol (RSTP), and the Cisco PVST+ architecture. Implementation of MSTP is not required if the Enterprise Composite Network Model (ECNM) is being employed because the number of active VLAN instances, and hence the STP instances, would be small and very stable due to the design.

MSTP allows you to build multiple spanning trees over trunks by grouping VLANs and associating them with spanning tree instances. Each instance can have a topology independent of other spanning tree instances. This architecture provides multiple active forwarding paths for data traffic and enables load balancing.

Network fault tolerance is improved over Common Spanning Tree (CST) because a failure in one instance (forwarding path) does not necessarily affect other instances. This VLAN-to-MSTP grouping must be consistent across all bridges within an MST region.

In large networks, you can more easily administer the network and use redundant paths by locating different VLAN and spanning tree assignments in different parts of the network. A spanning tree instance can exist only on bridges that have compatible VLAN instance assignments.

You must configure a set of bridges with the same MSTP configuration information, which allows them to participate in a specific set of spanning tree instances. Interconnected bridges that have the same MSTP configuration are referred to as a Multiple Spanning Tree (MST) region. Bridges with different MSTP configurations or legacy bridges running 802.1D are considered separate MST regions.

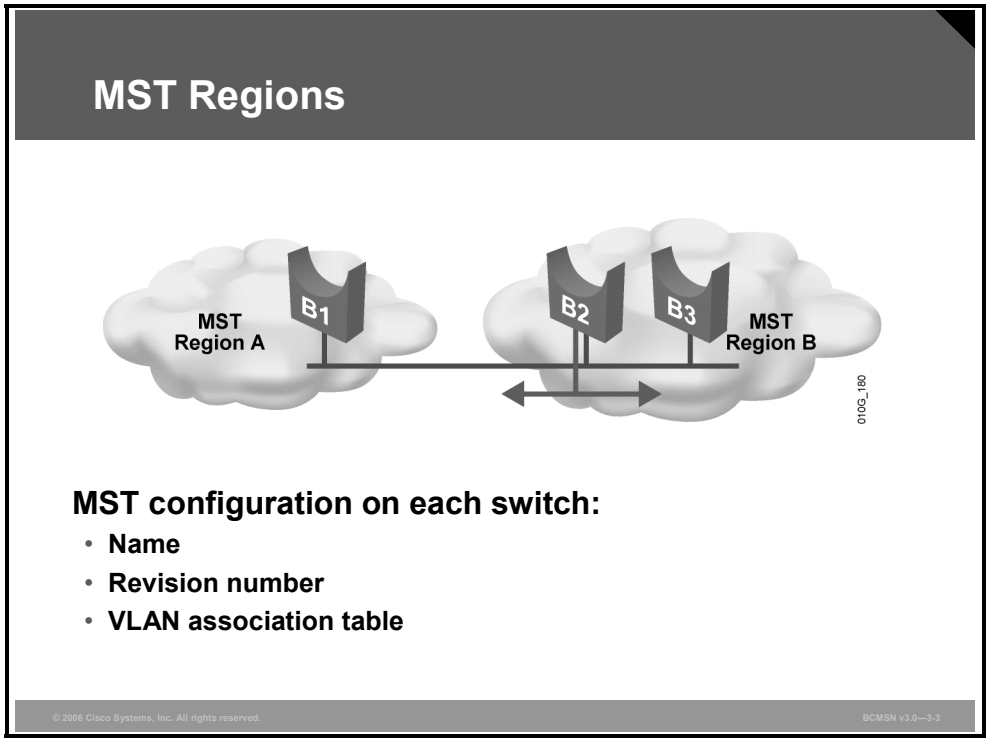
In a Cisco PVST+ environment, the spanning tree parameters are tuned so that half of the VLANs are forwarding on each uplink trunk. This is easily achieved by electing bridge D1 to be the root for VLANs 501–1000, and bridge D2 to be the root for VLANs 1–500. In this configuration, these statements are true:

- Optimum load balancing is achieved.
- One spanning tree instance for each VLAN is maintained, which means 1000 that there are instances for only two different logical topologies. This consumes resources for all the switches in the network (in addition to the bandwidth used by each instance sending its own BPDUs).

Note The MST implementation in Cisco IOS Release 12.2(25)SEC is based on the IEEE 802.1s standard. The MST implementations in earlier Cisco IOS releases are prestandard. For the standard and prestandard to work together, additional commands are required, as will be shown later in this lesson.

Describing MST Regions

This topic describes the characteristics of the MST region.



MSTP differs from the other spanning tree implementations in that it combines some, but not necessarily all, VLANs into logical spanning tree instances. This raises the problem of determining what VLAN is to be associated with what instance. More precisely, this means tagging BPDUs so that receiving devices can identify the instances and the VLANs to which they apply.

The issue is irrelevant in the case of the 802.1D standard, in which all instances are mapped to a unique and common instance CST. In the PVST+ implementation, different VLANs carry the BPDUs for their respective instances (one BPDU per VLAN), based on the VLAN tagging information.

To provide this logical assignment of VLANs to spanning trees, each switch running MSTP in the network has a single MSTP configuration that consists of three attributes:

- An alphanumeric configuration name (32 bytes)
- A configuration revision number (two bytes)
- A 4096-element table that associates each of the potential 4096 VLANs supported on the chassis with a given instance

To be part of a common MST region, a group of switches must share the same configuration attributes. It is up to the network administrator to properly propagate the configuration throughout the region.

Currently, this step is possible only by means of the command-line interface (CLI) or through Simple Network Management Protocol (SNMP). Other methods can be implemented in the future because the IEEE specification does not explicitly mention how to accomplish this step.

Note	If two switches differ on one or more configuration attributes, they are part of different regions.

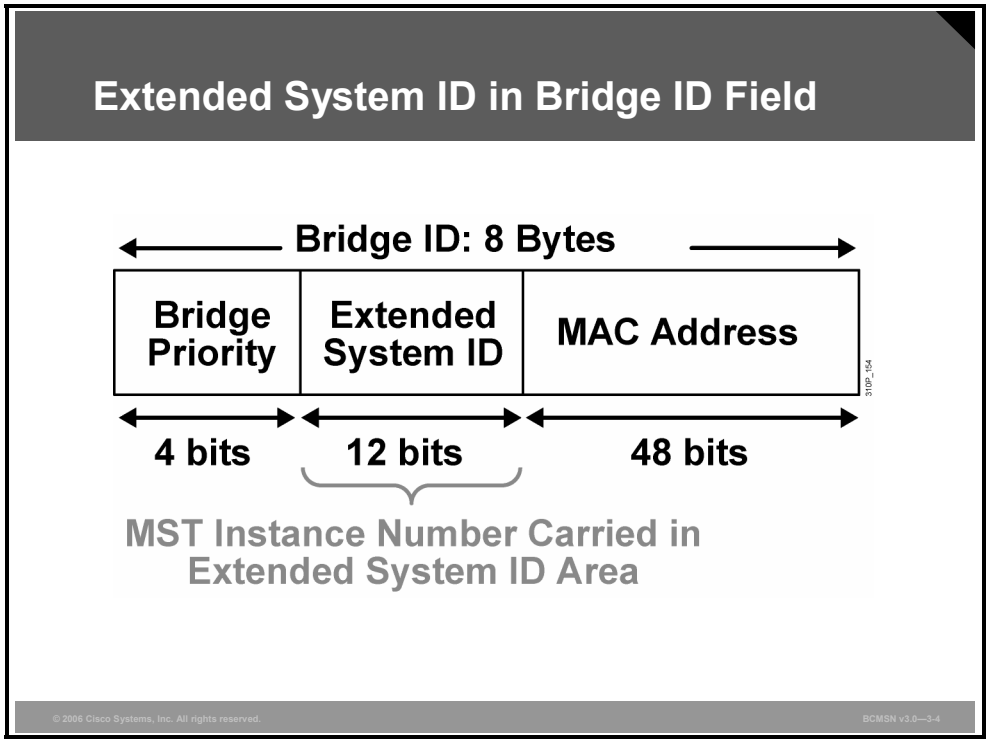
To ensure a consistent VLAN-to-instance mapping, it is necessary for the protocol to be able to exactly identify the boundaries of the regions. For that purpose, the characteristics of the region are included in BPDUs. The exact VLAN-to-instance mapping is not propagated in the BPDU because the switches need to know only whether they are in the same region as a neighbor.

Therefore, only a digest of the VLAN-to-instance mapping table is sent, along with the revision number and the name. After a switch receives a BPDU, it extracts the digest (a numerical value derived from the VLAN-to-instance mapping table through a mathematical function) and compares it with its own computed digest. If the digests differ, the mapping must be different, so the port on which the BPDU was received is at the boundary of a region.

In generic terms, a port is at the boundary of a region if the designated bridge on its segment is in a different region or if it receives legacy 802.1D BPDUs. In the figure, the port on B1 is at the boundary of region A, whereas the ports on B2 and B3 are internal to region B.

Describing the Extended System ID

This topic describes changes to the Bridge Priority field to accommodate the MSTP instance number.



As with PVST, the 12-bit Extended System ID field is used in MSTP. In MSTP, this field carries the MSTP instance number.

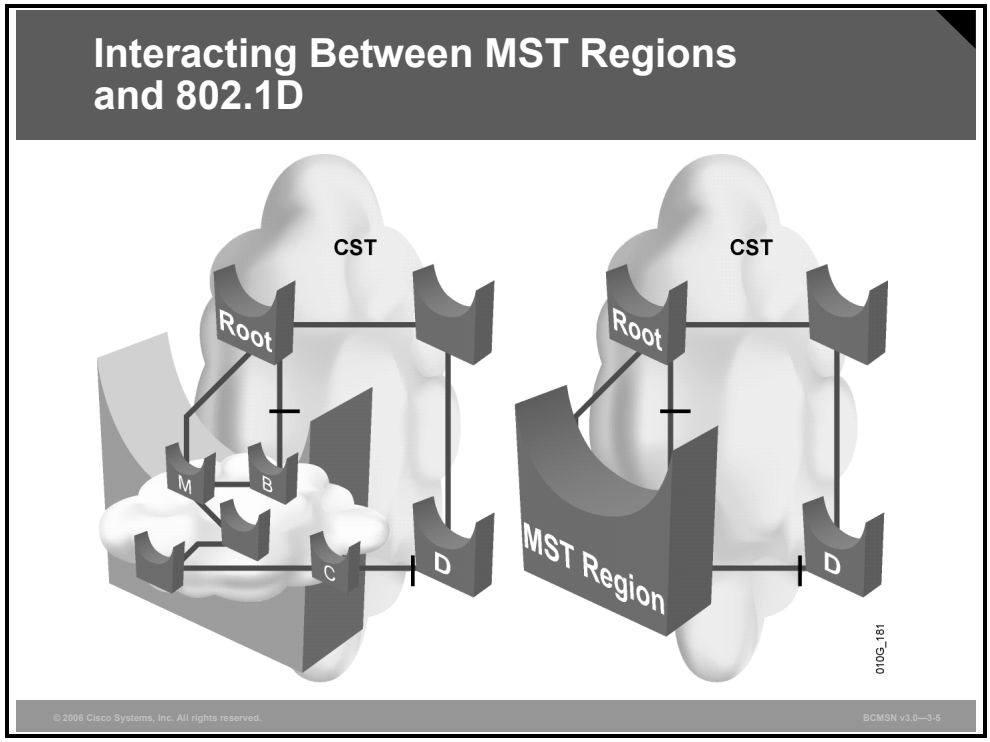
The 802.1D protocol states that each bridge must have a unique bridge identifier. In PVST, each VLAN is considered to be a different logical bridge. Therefore, each VLAN needs a unique bridge identifier. Before supporting 4000 VLANs, Cisco Systems supported a maximum of 1024 VLANs, which required 1024 bridge identifiers.

MAC address reduction is a feature that ensures bridge ID uniqueness for all 4000 VLANs, even when there are only 1024 or 64 MAC addresses available on the switch. It accomplishes this uniqueness by making the 16-bit Bridge Priority field in the BPDU unique for each VLAN. Before this feature, the Bridge Priority field was fully configurable and did not have to be unique for each VLAN because the appending 48-bit MAC address was unique for each VLAN.

MAC address reduction splits the 16-bit field into two fields: a configurable 4-bit field and a nonconfigurable 12-bit field. The nonconfigurable 12-bit field carries the VLAN ID (VID) or, with MSTP, the MSTP instance number. The two fields are merged to create the unique Bridge Priority field for a particular VLAN or, in this case, an MSTP instance. The appending MAC address remains the same for all instances.

Interacting Between MST Regions and 802.1Q

This topic describes how MSTP operates with CST.



One issue that arises from MSTP design is interoperability with the CST implementation in 802.1D. According to the IEEE 802.1s specification, an MSTP switch must be able to handle at least one Internal Spanning Tree (IST). The MST region consists of one IST and an arbitrary number of MSTP instances.

These are two functionally equivalent diagrams. Notice the location of the different blocked ports. In a typically bridged network, you expect to see a blocked port between Switch M and Switch B. Instead of blocking on Switch D, you expect to have the second loop broken by a blocked port somewhere in the middle of the MST region. However, due to the IST, the entire region appears as one virtual bridge that runs a single spanning tree (CST). This approach makes it possible to understand that the virtual bridge blocks an alternate port on Switch B. Also, that virtual bridge, which is on the Switch-C-to-Switch D segment, causes Switch D to block its port.

The MSTP instances are simple RSTP instances that exist only inside a region. These instances run the RSTP automatically by default, without any extra configuration. Unlike the IST, MSTP instances never interact with devices outside the region. Remember that MSTP runs only one spanning tree outside the region. Therefore, except for the IST instance, regular instances inside the region have no outside counterpart. In addition, MSTP instances do not send BPDUs outside a region, only the IST does.

MSTP instances do not send independent individual BPDUs. Inside the MST region, bridges exchange MSTP BPDUs that can be seen as normal RSTP BPDUs for the IST while containing additional information for each MSTP instance.

The IST (instance 0) runs on all bridges within an MST region. An important characteristic of the IST is that it provides interaction at the boundary of the MST region with other MST regions, and, more important, it is responsible for providing compatibility between the MST regions and the spanning tree of 802.1D (CST) and PVST+ networks connected to the region.

IST receives and sends BPDUs to the CST for compatibility with 802.1D. IST is capable of representing the entire MST region as a CST virtual bridge to switched networks outside the MST region.

- The MST region appears as a single virtual bridge to the adjacent CST and MST regions. The MST region uses RSTP port roles and operation.
- MSTP switches run IST, which augments CST information with internal information about the MST region.
- IST connects all the MSTP switches in the region and any CST switched domains.
- MSTP establishes and maintains additional spanning trees within each MST region. These spanning trees are termed MSTP instances. The IST is numbered 0, and the MSTP instances are numbered 1, 2, 3, and so on, up to 15. Any MSTP instance is local to the MST region and is independent of MSTP instances in another region, even if the MST regions are interconnected.
- The M-Record is a subfield, within the BPDU of MSTP instances, that contains enough information (root bridge and sender bridge priority parameters) for the corresponding instance to calculate the final topology. It does not contain any timer-related parameters (such as hellotime, forward delay, and max age) that are typically found in a regular IEEE 802.1D BPDU because these timers are derived from the IST BPDU timers. It is important to note that within an MST region, all spanning tree instances use the same parameters as the IST.
- MSTP instances combine with the IST at the boundary of MST regions to become the CST, as follows:
  - M-records are always encapsulated within MSTP BPDUs. The original spanning trees, which are called “M-trees,” are active only within the MST region. M-trees merge with the IST at the boundary of the MST region and form the CST.
- MSTP supports some of the PVST extensions, as follows:
  - PortFast is supported.
  - BPDU filter and BPDU guard are supported in MSTP mode.
  - Loop guard and root guard are supported in MSTP.
  - For private VLANs (PVLANs), you must map a secondary VLAN to the same instance as the primary.

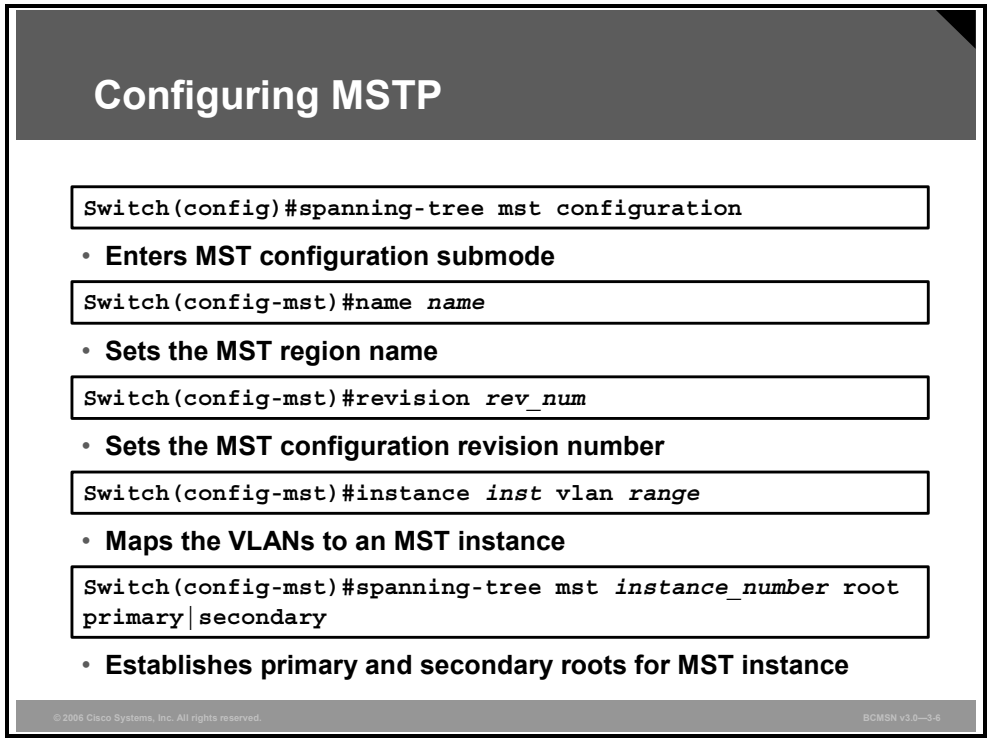
Each switch in the network that runs MSTP has a single MSTP configuration consisting of these attributes:

- An alphanumeric configuration name (32 bytes)
- A configuration revision number (two bytes)
- A 4096-element table that associates each of the potential 4096 VLANs supported on the chassis to a given instance

To be part of a common MST region, a group of switches must share the same configuration attributes. It is up to the network administrator to properly propagate the configuration throughout the region.

Describing MSTP Implementation Commands

This topic describes the commands used to implement MSTP.



Given the following steps, all switches would be configured with the spanning tree MSTP and extend system-id syntax, and only the distribution switches that terminate the VLANs would have their priority changed.

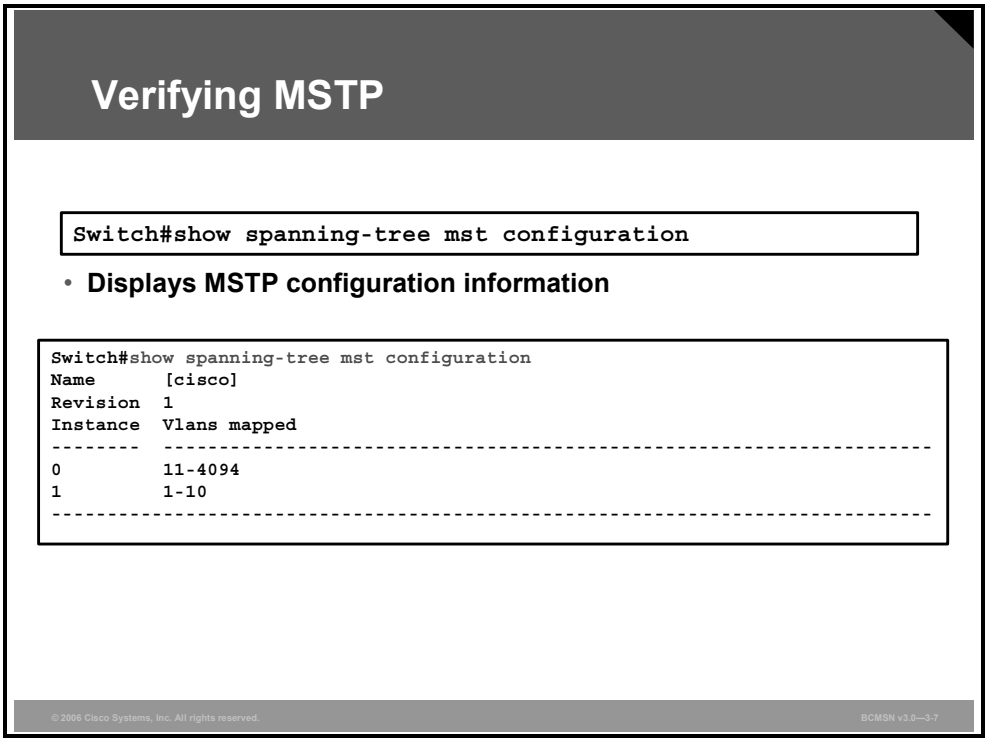
Step	Description	Notes and Comments
1.	Enter MSTP configuration submode. `Switch(config)# spanning-tree mst configuration`	You can use the **no** keyword to clear the MSTP configuration.
2.	Display the current MSTP configuration. `Switch(config-mst)#show current`	
3.	Set the MST region name. `Switch(config-mst)#name name`	
4.	Set the MSTP configuration revision number. `Switch(config-mst)#revision revision_number`	The revision number can be any unassigned 16-bit integer. It is not incremented automatically when you commit a new MSTP configuration.
5.	Map the VLANs to an MSTP instance. `Switch(config-mst)#instance instance_number vlan vlan_range`	If you do not specify the **vlan** keyword, you can use the **no** keyword to unmap all the VLANs that were mapped to an MSTP instance. If you specify the **vlan** keyword, you can use the **no** keyword to unmap a specified VLAN from an MSTP instance.
6.	`Switch(config-mst)#show pending`	Display the new MSTP configuration to be applied.

Step	Description	Notes and Comments
7.	`Switch(config-mst)#`**`end`**	Apply the configuration and exit MSTP configuration submode.
8.	`Switch(config-mst)#`**`spanning-tree mst`** *`instance_number`* **`root primary|secondary`**	Assign root bridge for MSTP instance. This syntax makes the switch root primary or secondary (only active if primary fails). It sets primary priority to 24576 and secondary to 28672.
9.	`Switch(config)#` **`spanning-tree extend system-id`**	This enables MAC address reduction, also known as extended system ID in Cisco IOS software.
10.	`Switch(config-if)#` **`spanning-tree mst pre-standard`**	This command is required if the neighboring switch is using a prestandard version of MSTP.

Note The MST implementation in Cisco IOS Release 12.2(25)SEC is based on the IEEE 802.1s standard. The MST implementations in earlier Cisco IOS releases are prestandard.

Configuring and Verifying MSTP

This topic explains the procedure to implement MSTP in a switched network.



Use the **show spanning-tree mst** command to display MSTP information.

Example: Displaying MSTP Configuration Information

This example shows how to display MSTP configuration information. This includes MST region name, revision number, and VLAN-to-MSTP instances mapping.

```
Switch#show spanning-tree mst configuration
Name      [cisco]
Revision  1
Instance  Vlans mapped
--------  ---------------------------------------------
0         11-4094
1         1-10
---------------------------------------------------------
```

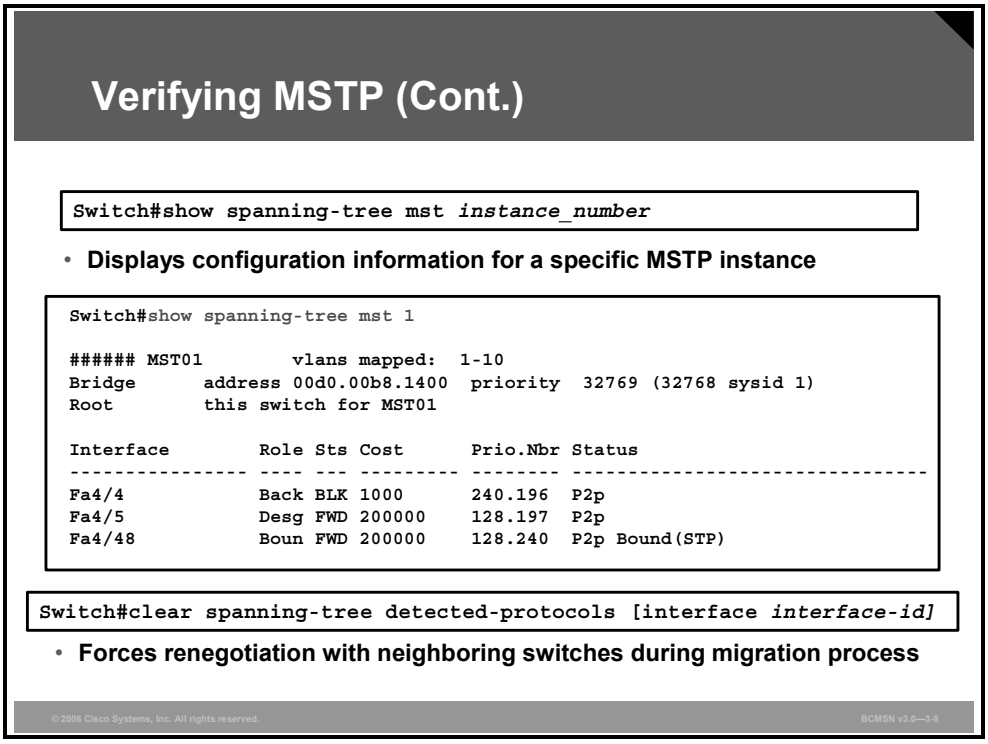
Example: Displaying General MSTP Information

This example shows how to display general MSTP information. Notice that the output is grouped by MSTP instances, starting with the IST.

```
Switch#show spanning-tree mst
 ###### MST00        vlans mapped:  11-4094
Bridge      address 00d0.00b8.1400  priority  32768 (32768 sysid 0)
Root        address 00d0.004a.3c1c  priority  32768 (32768 sysid 0)
            port    Fa4/48          path cost 203100
IST master  this switch
Operational hello time 2, forward delay 15, max age 20, max hops 20
Configured  hello time 2, forward delay 15, max age 20, max hops 20

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- ------------------------
Fa4/4            Back BLK 1000      240.196  Point-to-point
Fa4/5            Desg FWD 200000    128.197  Point-to-point
Fa4/48           Root FWD 200000    128.240  Point-to-point Bound(STP)

###### MST01        vlans mapped:  1-10
Bridge      address 00d0.00b8.1400  priority  32769 (32768 sysid 1)
Root        this switch for MST01

Interface        Role Sts Cost      Prio.Nbr Status

Fa4/4            Back BLK 1000      240.196  Point-to-point
Fa4/5            Desg FWD 200000    128.197  Point-to-point
Fa4/48           Boun FWD 200000    128.240  Point-to-point Bound(STP)
```


Example: Displaying MSTP Information for a Specific Instance

This example displays spanning tree information for a specific MSTP instance—particularly port status, costs, and forwarding role.



A switch running the Per VLAN Rapid Spanning Tree+ (PVRST+) protocol or the MSTP supports a built-in protocol migration mechanism that enables it to interoperate with legacy IEEE 802.1D switches.

If a PVRST+ switch or an MSTP switch receives a legacy IEEE 802.1D configuration BPDU with the protocol version set to 0, it sends only IEEE 802.1D BPDUs on that port. An MST switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, an MST BPDU (version 3) associated with a different region, or an RSTP BPDU (version 2).

However, the switch does not automatically revert to the PVRST+ or MSTP mode if it no longer receives IEEE 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. Use this command in this situation:

```
Switch# clear spanning-tree detected-protocols
```

Example: Displaying MSTP Information for a Specific Instance

This example displays MSTP information for a specific instance.

```
Switch#show spanning-tree mst 1

###### MST01        vlans mapped:  1-10
Bridge      address 00d0.00b8.1400  priority  32769 (32768 sysid 1)
Root        this switch for MST01

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- ----------------
Fa4/4            Back BLK 1000      240.196  Point-to-point
Fa4/5            Desg FWD 200000    128.197  Point-to-point
Fa4/48           Boun FWD 200000    128.240  Point-to-point Bound(STP)
```


Example: Displaying MSTP Information for a Specific Interface

This example displays MSTP information for a specific interface.

```
Switch#show spanning-tree mst interface fastethernet 4/4

FastEthernet4/4 of MST00 is backup blocking
Edge port:no             (default)        port guard :none        (default)
Link type:point-to-point (auto)           bpdu filter:disable     (default)
Boundary :internal                        bpdu guard :disable     (default)
Bpdus sent 2, received 368

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- ------------------------
0        Back BLK 1000      240.196  11-4094
1        Back BLK 1000      240.196  1-10
```

Example: Displaying MSTP Information for a Specific Instance and Interface

This example displays MSTP information for a specific interface and a specific MSTP instance.

```
Switch#show spanning-tree mst 1 interface fastethernet 4/4

FastEthernet4/4 of MST01 is backup blocking
Edge port:no             (default)        port guard :none        (default)
Link type:point-to-point (auto)           bpdu filter:disable     (default)
Boundary :internal                        bpdu guard :disable     (default)
Bpdus (MRecords) sent 2, received 364

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- -------------------------
1        Back BLK 1000      240.196  1-10
```

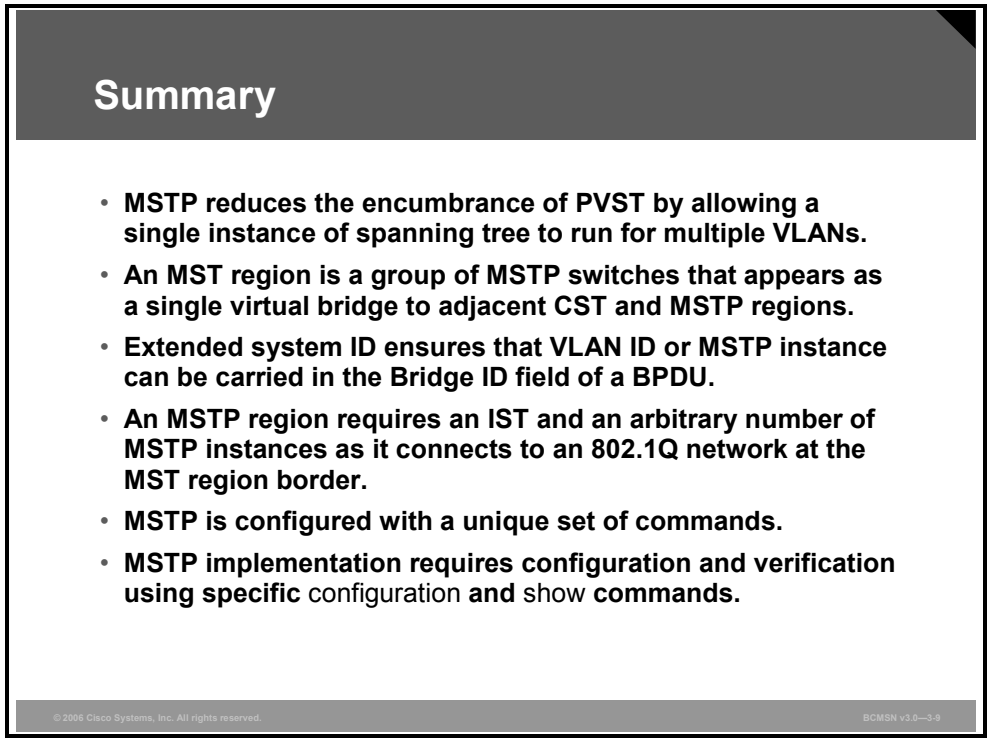
Example: Displaying Detailed MSTP Information

This example displays detailed MSTP information for a specific instance.

```
Switch#show spanning-tree mst 1 detail

###### MST01        vlans mapped:  1-10
Bridge      address 00d0.00b8.1400  priority  32769 (32768 sysid 1)
Root        this switch for MST01

FastEthernet4/4 of MST01 is backup blocking
Port info             port id        240.196  priority   240  cost        1000
Designated root       address 00d0.00b8.1400  priority 32769  cost           0
Designated bridge     address 00d0.00b8.1400  priority  32769 port id  128.197
Timers:message expires in 5 sec, forward delay 0, forward transitions 0
Bpdus (MRecords) sent 123, received 1188

FastEthernet4/5 of MST01 is designated forwarding
Port info             port id        128.197  priority   128  cost      200000
Designated root       address 00d0.00b8.1400  priority 32769  cost           0
Designated bridge     address 00d0.00b8.1400  priority 32769  port id  128.197
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 1188, received 123

FastEthernet4/48 of MST01 is boundary forwarding
Port info             port id        128.240  priority   128  cost      200000
Designated root       address 00d0.00b8.1400  priority 32769  cost           0
Designated bridge     address 00d0.00b8.1400  priority 32769  port id  128.240
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 78, received 0
```


Summary

This topic summarizes the key points discussed in this lesson.



Lesson 4

Configuring Link Aggregation with EtherChannel

Overview

When multiple physical links exist between two switches, these links can be bundled into a single logical link that provides high aggregate bandwidth and fault tolerance for inter-switch connectivity. This lesson will examine the specifics of EtherChannel.

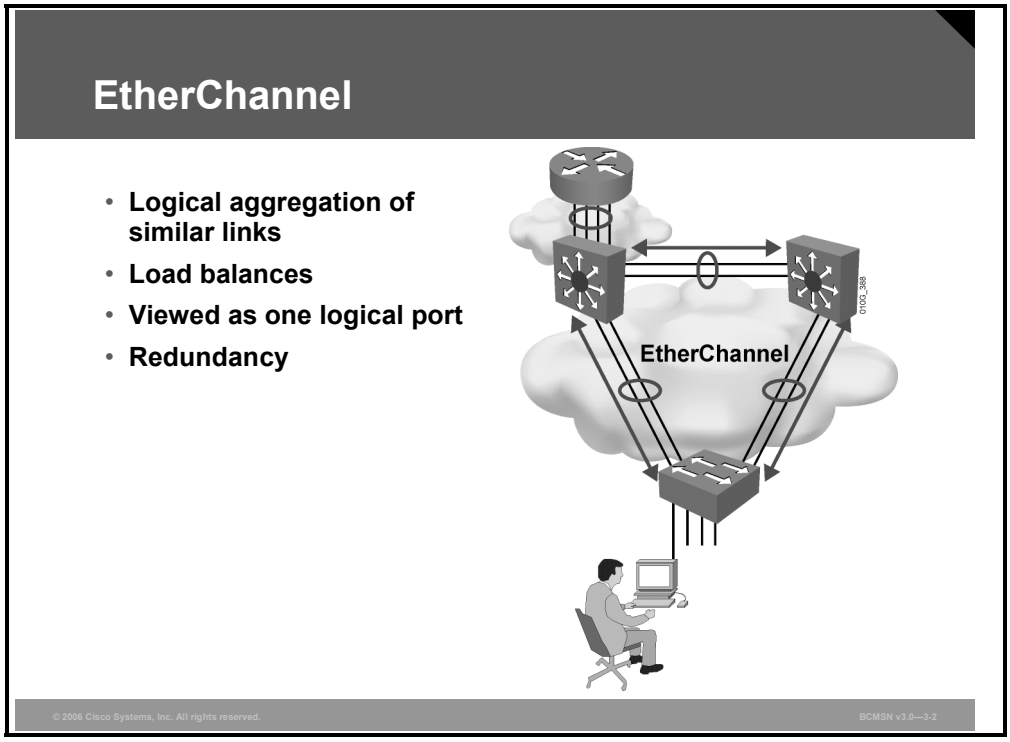
Objectives

Upon completing this lesson, you will be able to configure link aggregation with EtherChannel. This ability includes being able to meet these objectives:

- Describe EtherChannel
- Compare the PAgP and LACP
- Describe the commands used to configure EtherChannel
- Describe the guidelines and best practices for configuring port channels using EtherChannel
- Configure load balancing among the ports included in an EtherChannel

Describing EtherChannel

This topic describes EtherChannel.



Companies require greater and cheaper bandwidth to run their networks. Users are becoming more impatient with any sort of latency that occurs in the network. The insatiable appetite of customers for faster networks and higher availability of the networks has made the competition intense between vendors. Some years ago, Cisco Systems came up with a method to not only provide substantially higher bandwidth but to provide it with lower cost overhead.

EtherChannel is a technology that was originally developed by Cisco as a LAN switch-to-switch technique of inverse multiplexing of multiple Fast or Gigabit Ethernet switch ports into one logical channel. Its benefit is that it is effectively cheaper than higher-speed media while using existing switch ports.

EtherChannel has developed into a cross-platform method of load balancing between servers, switches, and routers. EtherChannel can bond two, four, or eight ports (Cisco Catalyst 6500) to develop one logical connection with redundancy. The three major aspects to EtherChannel are as follows:

- Frame distribution
- Management of EtherChannel
- Logical port

EtherChannel does not do frame-by-frame forwarding in a round-robin fashion on each of the links. The load-balancing policy or frame distribution used is contingent upon the switch platform used. For instance, in a Cisco Catalyst 5500 switch platform, the load-balancing operation performs an X-OR calculation on the two lowest-order bits of the source and destination MAC address. An X-OR operation between a given pair of addresses will use the same link for all frames.

One of the primary benefits of the X-OR operation is to prevent out-of-order frames on the downstream switch. The other advantage is redundancy. If the active channel used by a connection is lost, the existing traffic can traverse over another active link on that EtherChannel.

The one disadvantage to X-OR operation is that the load on the channels might not be equal because the load-balancing policy is done on a specific header as defined by the platform or user configuration.

On a Cisco Catalyst 6500 switch, the load-balancing operation can be performed on MAC address, IP address, or IP + TCP/User Datagram Protocol (UDP), depending on the type of Supervisor/Policy Feature Card (PFC) used. Use the **show port capabilities** command to check the module for EtherChannel feature.

The default frame distribution behavior for the Cisco Catalyst 6500 is IP.

EtherChannel bundles individual Ethernet links into a single logical link that provides bandwidth up to 1600 Mbps (Fast EtherChannel, full duplex) or 16 Gbps (Gigabit EtherChannel) between two Cisco Catalyst switches. All interfaces in each EtherChannel must be the same speed and duplex, and they must be configured as either Layer 2 or Layer 3 interfaces.

If a link within the EtherChannel bundle fails, traffic previously carried over the failed link will be carried over the remaining links within the EtherChannel.

The configuration applied to the individual physical interfaces that are to be aggregated by EtherChannel affects only those interfaces. Each EtherChannel has a logical port channel interface. A configuration applied to the port channel interface affects all physical interfaces assigned to that interface. (Such commands can be Spanning Tree Protocol [STP] commands or commands to configure a Layer 2 EtherChannel as a trunk.)

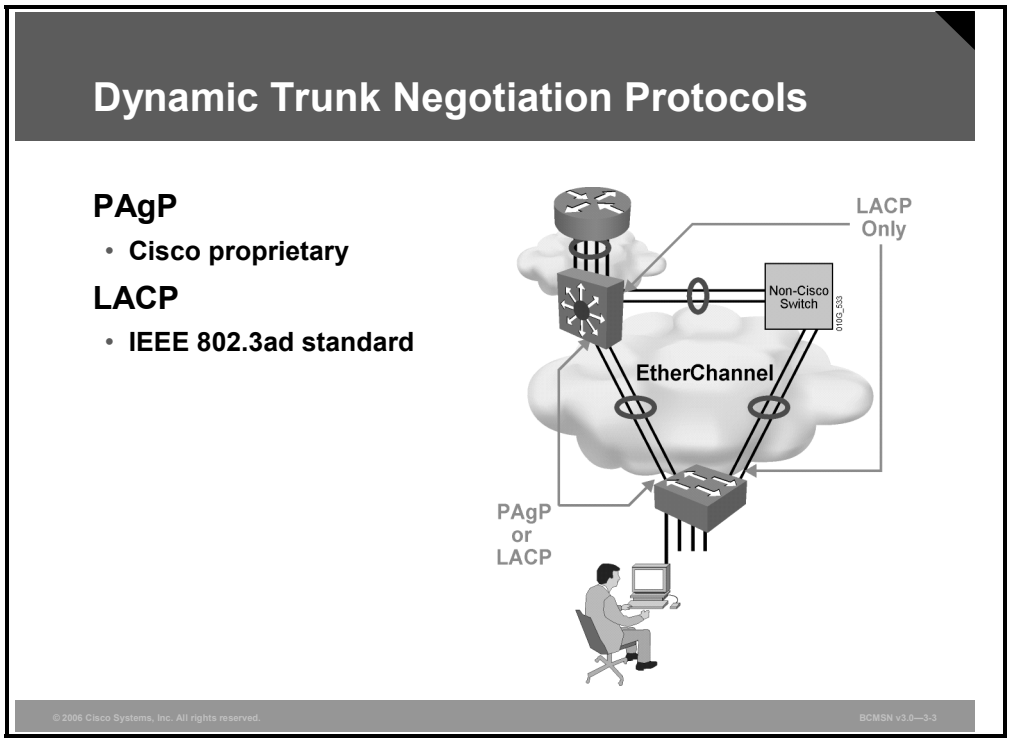
EtherChannel Features and Benefits

Here are advantages of EtherChannelare:

- It allows for the creation of a very high-bandwidth logical link.
- It load balances among the physical links involved.
- It provides automatic failover.
- It simplifies subsequent logical configuration (configuration is per logical link instead of per physical link).

Describing the PAgP and LACP Protocols

This topic compares the Port Aggregation Protocol (PAgP) and Line Aggregation Control Protocol (LACP). They allow ports with similar characteristics to form a channel through dynamic negotiation with adjoining switches.



The PAgP aids in the automatic creation of Fast EtherChannel links. PAgP packets are sent between Fast EtherChannel–capable ports to negotiate the forming of a channel. When PAgP identifies matched Ethernet links, it groups the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

The management of the EtherChannel is done by PAgP. PAgP packets are sent every 30 seconds, using multicast group MAC address 01-00-0C-CC-CC-CC with protocol value 0x0104. PAgP checks for configuration consistency and manages link additions and failures between two switches. It ensures that when EtherChannel is created all ports have the same type of configuration. In EtherChannel, it is mandatory that all ports have the same speed, duplex setting, and VLAN information. Any port modification after the creation of the channel will also change all the other channel ports.

The last component of EtherChannel is the creation of the logical port. The logical port, or Agport, is composed of all the links that make up the EtherChannel. The actual functionality and behavior of the Agport is no different than that of any other port. For instance, the spanning tree algorithm treats Agport as a single port.

LACP is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer. It performs a similar function as PAgP with Cisco EtherChannel. Because LACP is an IEEE standard, it can be used to facilitate EtherChannels in mixed-switch environments.

Interface Modes

Interfaces can be set in any of several modes to control EtherChannel formation.

Comparison of Interface Modes

The table shows the different settings for PAgP and LACP.

PAgP	LACP
Auto: This PAgP mode places an interface in a passive negotiating state in which the interface responds to the PAgP packets that it receives but does not initiate PAgP negotiation (default).	**Passive:** This LACP mode places a port in a passive negotiating state. In this state, the port responds to the LACP packets that it receives but does not initiate LACP packet negotiation (default).
Desirable: This PAgP mode places an interface in an active negotiating state in which the interface initiates negotiations with other interfaces by sending PAgP packets. Interfaces configured in the “on” mode do not exchange PAgP packets.	**Active**: This LACP mode places a port in an active negotiating state. In this state, the port initiates negotiations with other ports by sending LACP packets.
On: This mode forces the interface to channel without PAgP or LACP.	**On:** This mode forces the interface to channel without PAgP or LACP.

LACP Parameters

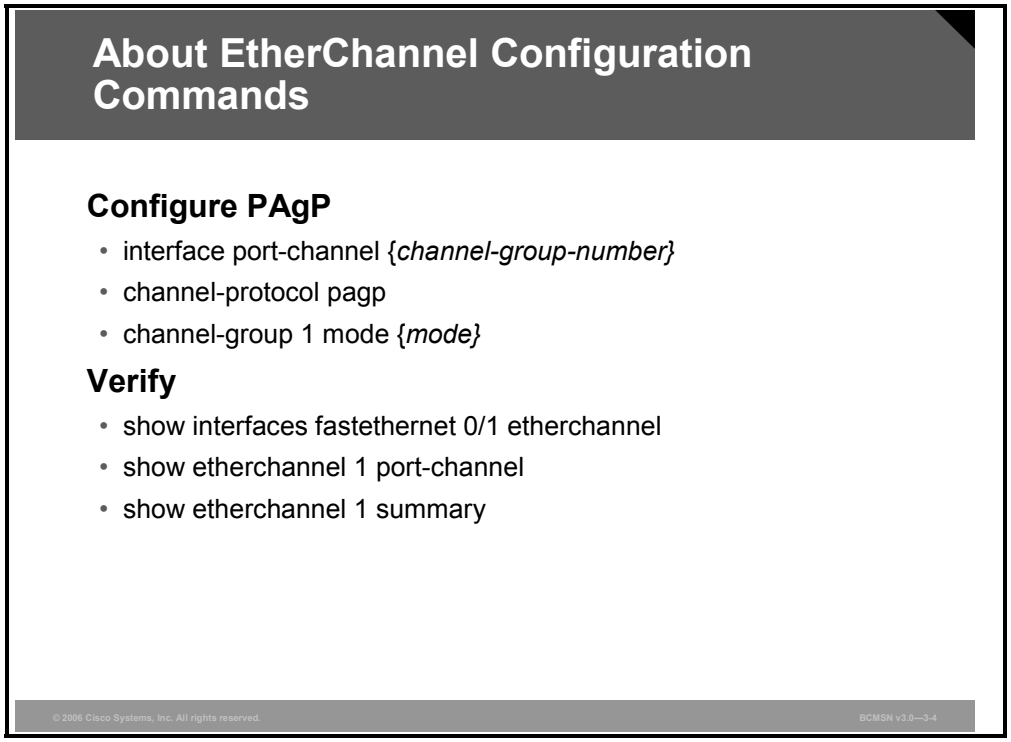
These parameters are used in configuring LACP:

- **System priority:** Each switch running LACP must have a system priority. The system priority can be specified automatically or through the command-line interface (CLI). The switch uses the MAC address and the system priority to form the system ID.
- **Port priority:** Each port in the switch must have a port priority. The port priority can be specified automatically or through the CLI. The port priority and the port number form the port identifier. The switch uses the port priority to decide which ports to put in standby mode when a hardware limitation prevents all compatible ports from aggregating.
- **Administrative key:** Each port in the switch must have an administrative key value, which can be specified automatically or through the CLI. The administrative key defines the ability of a port to aggregate with other ports, determined by these factors:
  - The port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
  - The configuration constraints that you establish

When enabled, LACP attempts to configure the maximum number of compatible ports in a channel. In some instances, LACP is not able to aggregate all the ports that are compatible; for example, the remote system might have more restrictive hardware limitations. When this occurs, all the ports that cannot be actively included in the channel are put in hot standby state and used only if one of the channeled ports fails.

Describing EtherChannel Configuration

This topic describes the commands used to configure EtherChannel.



These commands are used to configure and verify EtherChannel.

EtherChannel Configuration Commands

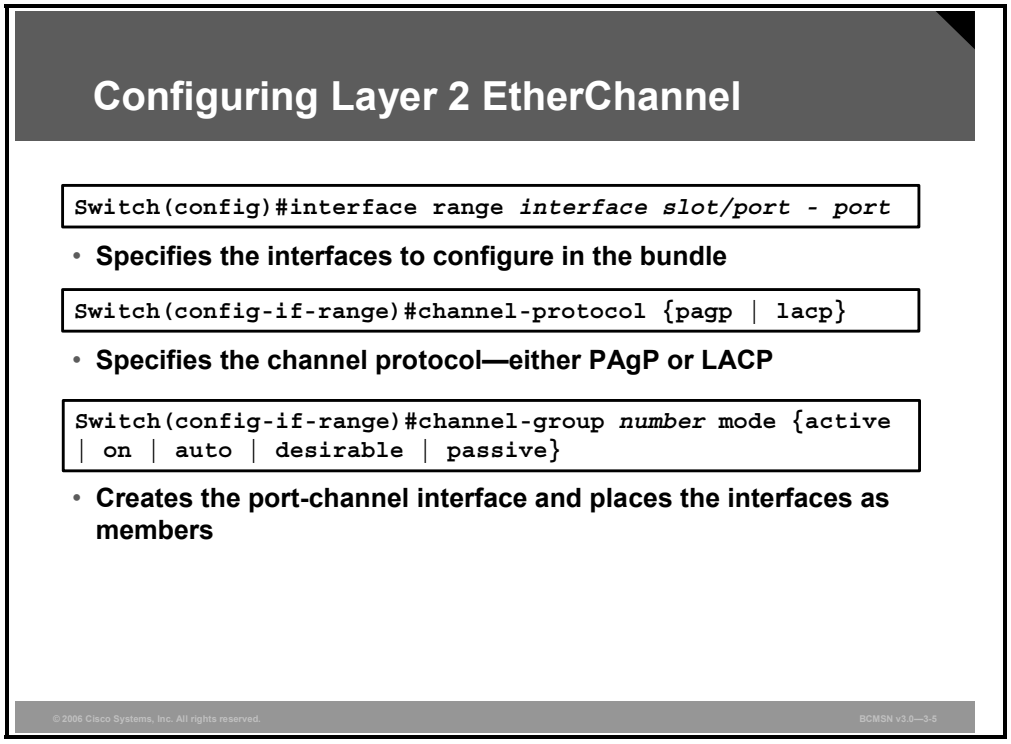
You can use these commands:

Command	Description
`Switch(config)#` **`interface port-channel`** *`port-channel-number`*	Creates a port-channel interface and moves to port-channel configuration mode, allowing the configuration of port-channel interface configuration parameters
`Switch(config-if)#` **`interface`** *`media-type slot/port`*	Moves to configure physical ports into EtherChannel bundles
`Switch(config-if)#` **`channel-group`** *`number`* **`mode`** *`mode_type`*	Associates an interface with a specific port-channel group and specifies if negotiation is to occur
`Switch(config)#` **`Port-channel load-balance`** *`load-balance-type`*	Tells the switch how to load balance traffic over the individual links in the EtherChannel bundle
`Switch#` **`Show running-config interface port-channel`** *`channel_number`*	Shows the running configuration for a specific port-channel interface
`Switch#` **`show running-config interface`** *`type mod/port`*	Shows the running configuration for a specific physical interface

Command	Description
Switch# **show interfaces** *type mod/port* **etherchannel**	Displays information on a physical interface that is specific to its role in an EtherChannel bundle
Switch# **show etherchannel** *num* **port-channel**	Displays information on the current state of the port-channel interface
Switch# **show etherchannel** *num* **summary**	Displays a one-line summary per channel-group

Configuring Port Channels Using EtherChannel

This topic describes the guidelines and best practices for configuring port channels using EtherChannel.



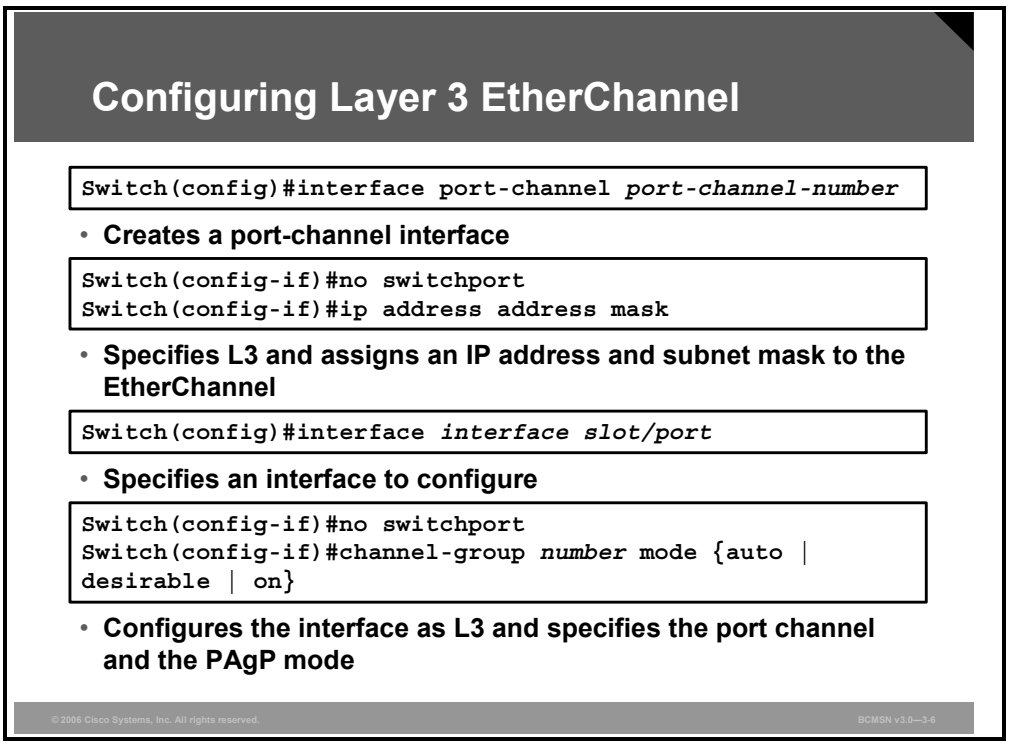
Configure a Layer 2 EtherChannel Bundle

This table shows the steps for configuring and verifying an EtherChannel interface.

Step	Action	Notes
1.	`Switch(config)#` **`interface range fastethernet`** [***`interface_range`***]	Specifies the interfaces that will compose the EtherChannel group
2.	`Switch(config-if-range)#` **`channel-protocol {pagp | lacp}`**	Specifies the channeling protocol to be used
3.	`Switch(config-if-range)#` **`channel-group 2 mode desirable`**	Creates the port-channel interface, if necessary, and assigns the specified interfaces to it

Configuring Layer 3 EtherChannel

This subtopic shows the commands for configuring a Layer 3 EtherChannel.



Configure EtherChannel

The table shows the steps for configuring and verifying a Layer 3 EtherChannel interface.

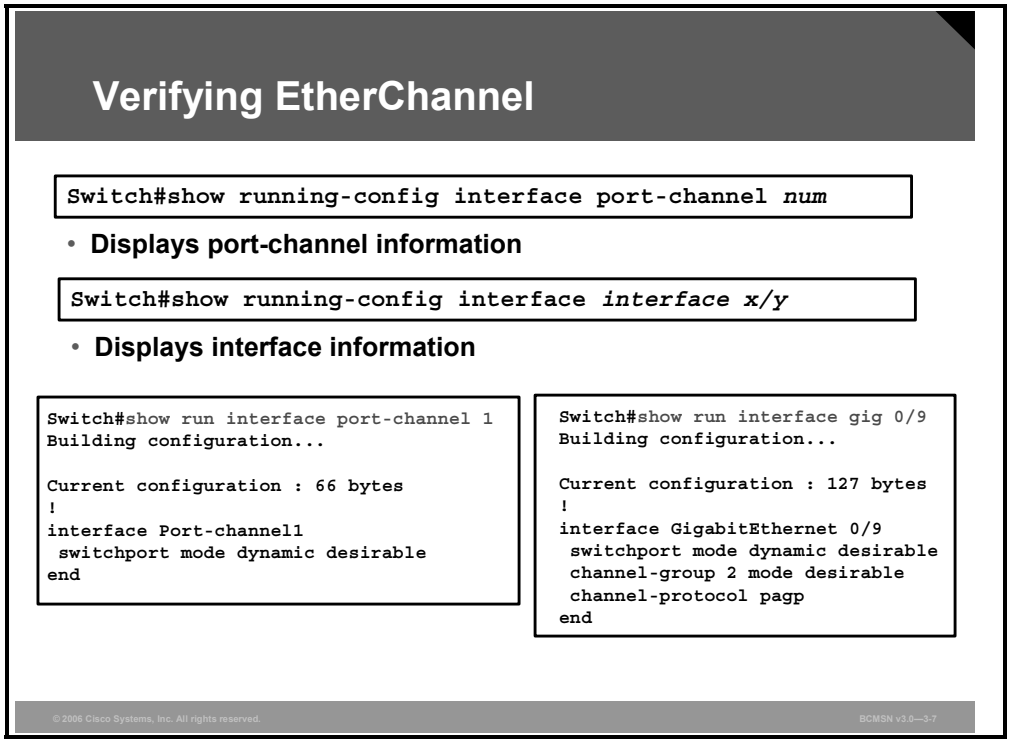
Configure a Layer 3 EtherChannel Bundle

Step	Action	Notes
1.	Create the port-channel interface. `Switch(config)#` **`interface port-channel 1`**	Creates a virtual Layer 2 interface.
2.	`Switch(config-if)#` **`no switchport`**	Changes interface to Layer 3 to enable the use of the IP address command.
3.	Assign an IP address to the port-channel interface because this will be a Layer 3 interface. `Switch(config-if)#` **`ip address 172.32.52.10 255.255.255.0`**	Assigns an IP address to the port-channel interface.
4.	Navigate to the interface that is to be associated with the EtherChannel bundle. `Switch(config)#` **`interface range fastethernet 5/4 - 5`**	This example shows navigation to a range of interfaces with the port-channel. Individual interfaces can be used also.

Step	Action	Notes
5.	Prepare interface. `Switch(config-if-range)#` **`no switchport`** `Switch(config-if-range)#` **`channel-protocol pagp`**	The independent Layer 2 and Layer 3 functionality of the port must be removed so that the port can function as part of a group. Optionally, can specify the channel protocol.
6.	Associate physical interfaces with the port-channel. `Switch(config-if-range)#` **`channel-group 1 mode desirable`**	Assigns all of the physical interfaces in the range to the EtherChannel group.

Verifying EtherChannel

This subtopic addresses the commands that can be used to verify the EtherChannel configuration.



Use the **show interfaces** [*interface*] [*num*] **etherchannel** command to display information about the port channel and the specific EtherChannel interfaces.

Example: Verifying the Configuration of a Layer 3 EtherChannel

This example shows a verification command and its output:

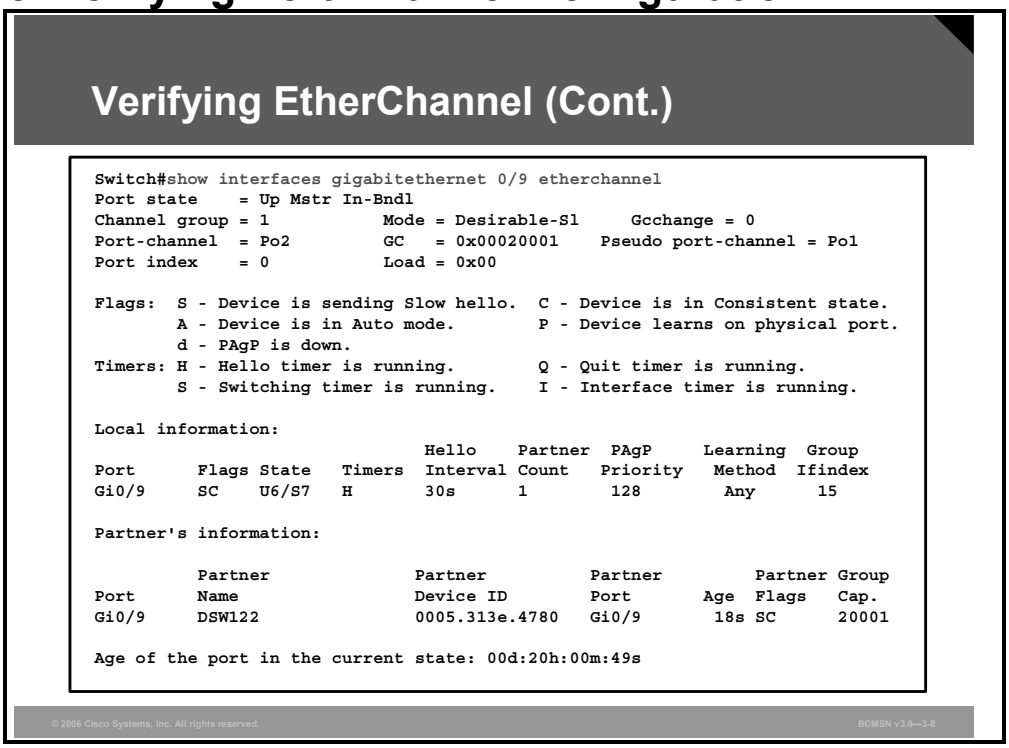
```
Switch#show interfaces fastethernet 5/4 etherchannel

Port state    = EC-Enbld Up In-Bndl Usr-Config
Channel group = 1           Mode = Desirable     Gcchange = 0
Port-channel  = Po1         GC   = 0x00010001    Pseudo-port-channel = Po1
Port indx     = 0           Load = 0x55

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Local information:
                              Hello    Partner  PAgP     Learning  Group
Port      Flags State   Timers  Interval Count   Priority   Method  Ifindex
Fa5/4     SC    U6/S7           30s      1        128        Any      55

Partner's information:
          Partner              Partner          Partner          Partner Group
Port      Name                 Device ID        Port       Age   Flags   Cap.
Fa5/4     JAB031301            0050.0f10.230c   2/45         1s SAC     2D

Age of the port in the current state: 00h:54m:52s
```

Example: Verifying the Configuration of a Layer 2 EtherChannel

These two examples show how to verify the configuration of Fast Ethernet interface 5/6:

```
Switch#show running-config interface fastethernet 5/6

Building configuration...
Current configuration:
!
interface FastEthernet5/6
switchport access vlan 10
switchport mode access
channel-group 2 mode desirable
end

Switch#show interfaces fastethernet 5/6 etherchannel
Port state    = EC-Enbld Up In-Bndl Usr-Config
Channel group = 1           Mode = Desirable     Gcchange = 0
Port-channel  = Po1         GC   = 0x00010001
Port indx     = 0           Load = 0x55

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Local information:
                              Hello    Partner  PAgP     Learning  Group
Port      Flags State   Timers  Interval Count   Priority   Method  Ifindex
Fa5/6     SC    U6/S7           30s      1        128        Any      56
Partner's information:

          Partner              Partner          Partner         Partner Group
Port      Name                 Device ID        Port       Age  Flags   Cap.
Fa5/6     JAB031301            0050.0f10.230c   2/47        18s SAC     2F

Age of the port in the current state: 00h:10m:57s
```


Example: Verifying Port-Channel Configuration



Use the **show etherchannel** command to display port-channel information after configuration.

This example shows how to verify the configuration of port-channel interface 1 after the interfaces have been configured.

```
Switch#show etherchannel 1 port-channel


Channel-group listing:
----------------------
Group: 1
------------


Port-channels in the group:
----------------------
Port-channel: Po1
------------


Age of the Port-channel   = 01h:56m:20s
Logical slot/port   = 10/1          Number of ports = 2
GC                  = 0x00010001      HotStandBy port = null
Port state          = Port-channel L3-Ag Ag-Inuse


Ports in the Port-channel:
```

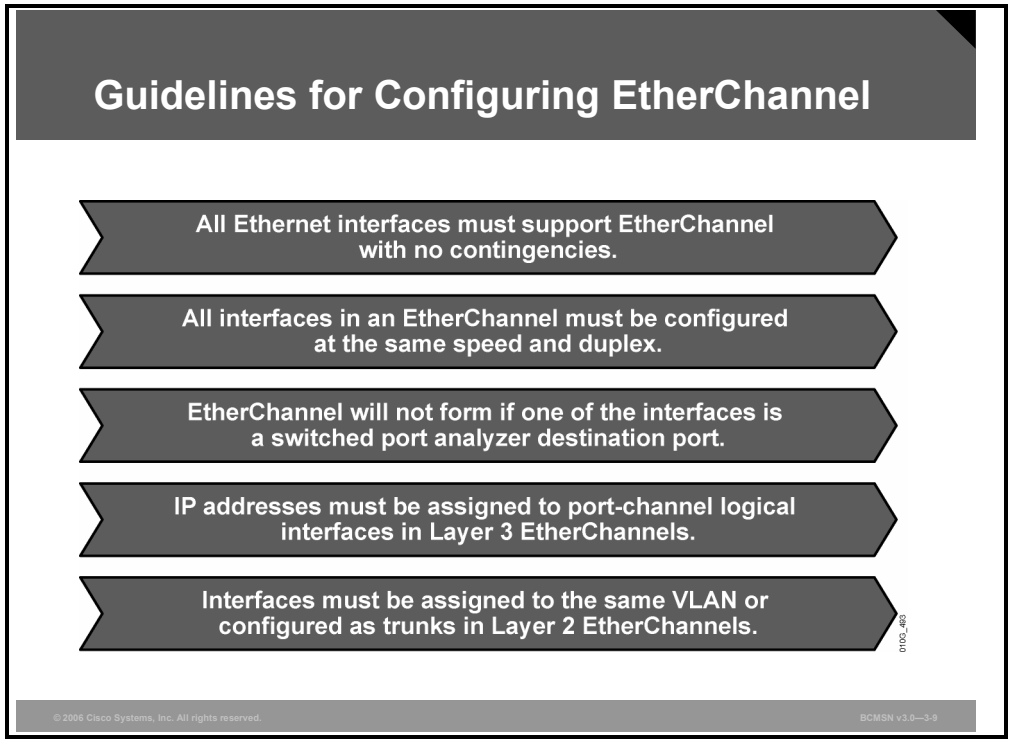
```
Index   Load   Port
------------------
1     00      Fa5/6
0     00      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6

Switch#
```

This example shows how to verify the configuration of port-channel interface 1 (a Layer 2 EtherChannel) after the interfaces have been configured.

```
Switch#show etherchannel 1 port-channel

Port-channels in the group:
----------------------

Port-channel: Po1
------------

Age of the Port-channel   = 00h:23m:33s
Logical slot/port   = 10/2           Number of ports in agport = 2
GC                  = 0x00020001      HotStandBy port = null
Port state          = Port-channel Ag-Inuse

Ports in the Port-channel:

Index   Load   Port
------------------
1     00      Fa5/6
0     00      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6
```

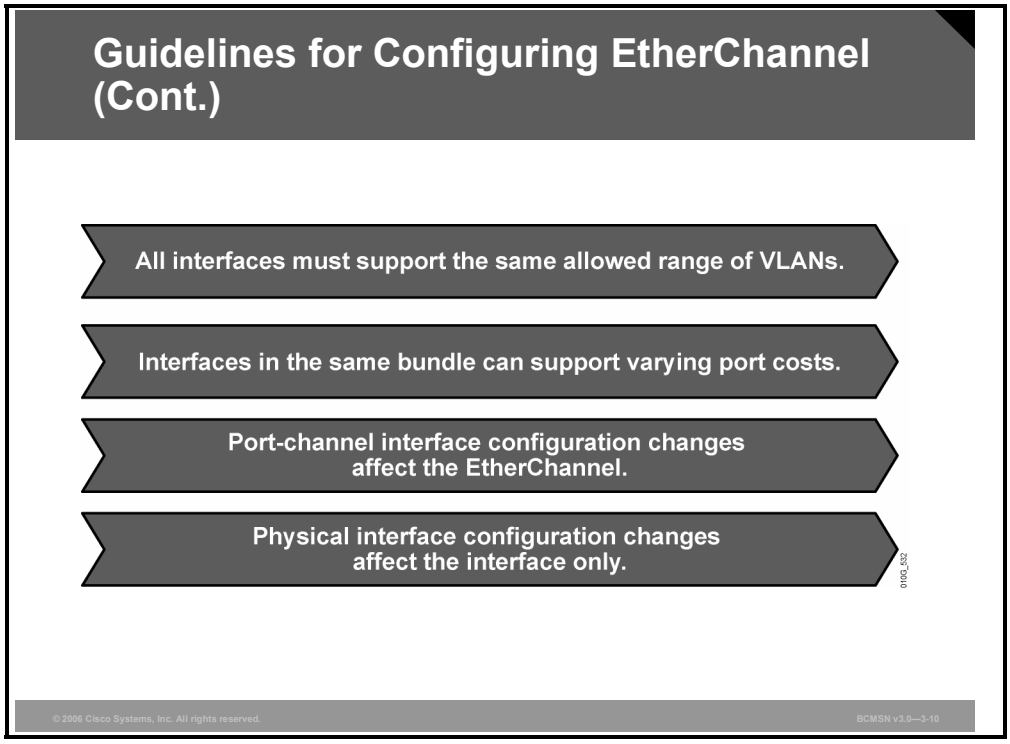

Guidelines and Best Practices for Configuring EtherChannel

This subtopic addresses guidelines and best practices for configuring EtherChannel.



Follow these guidelines and restrictions when configuring EtherChannel interfaces.

- **EtherChannel support:** All Ethernet interfaces on all modules support EtherChannel (maximum of eight interfaces), with no requirement that interfaces be physically contiguous or on the same module.
- **Speed and duplex:** Configure all interfaces in an EtherChannel to operate at the same speed and in the same duplex mode. Also, if one interface in the bundle is shut down, it is treated as a link failure, and traffic will traverse other links in the bundle.
- **Switched port analyzer (SPAN) and EtherChannel:** An EtherChannel will not form if one of the interfaces is a SPAN destination port.
- **For Layer 3 EtherChannels:** Assign Layer 3 addresses to the port-channel logical interface, not to the physical interfaces in the channel.
- **VLAN match:** All interfaces in the EtherChannel bundle must be assigned to the same VLAN or be configured as a trunk.



- **Range of VLANs:** An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking Layer 2 EtherChannel.

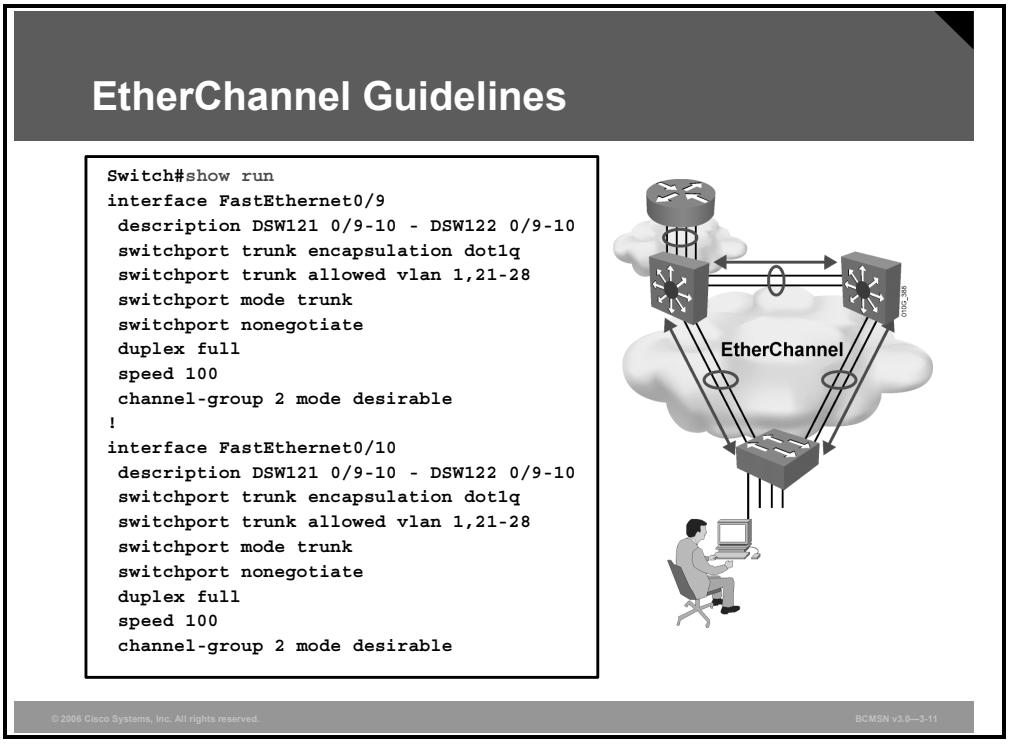
  If the allowed range of VLANs is not the same, the interfaces do not form an EtherChannel, even when set to auto or desirable mode. For Layer 2 EtherChannels, either assign all interfaces in the EtherChannel to the same VLAN or configure them as trunks.

- **STP path cost:** Interfaces with different STP port path costs can form an EtherChannel as long they are otherwise compatibly configured. Setting different STP port path costs does not, by itself, make interfaces incompatible for the formation of an EtherChannel.

- **Port channel versus interface configuration:** After you configure an EtherChannel, any configuration that you apply to the port-channel interface affects the EtherChannel. Any configuration that you apply to the physical interfaces affects only the specific interface you configured.

Guidelines and Best Practices Example

This subtopic describes an example of the guidelines and best practices for configuring port channels using EtherChannel.

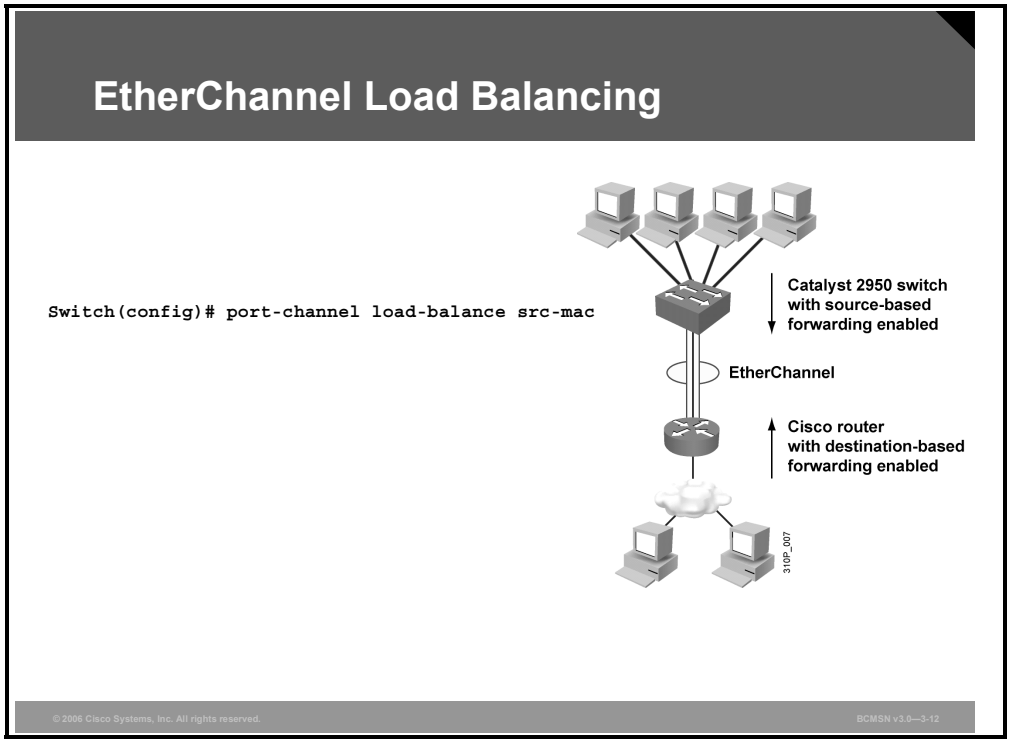


This example shows how to configure an EtherChannel following the guidelines:

- **Speed and duplex:** Configure all interfaces in an EtherChannel to operate at the same speed and in the same duplex mode.
- **VLAN match:** All interfaces in the EtherChannel bundle must be assigned to the same VLAN or be configured as a trunk.
- **Range of VLANs:** An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking Layer 2 EtherChannel.

Configuring Load Balancing over EtherChannel

This topic describes the configuration of load balancing among the ports included in an EtherChannel.



In this figure, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router.

The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel.

Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going to only a single MAC address, using the destination-MAC address always chooses the same link in the channel; using source addresses might result in better load balancing.

EtherChannel Load-Balancing Characteristics

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding.

With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet.

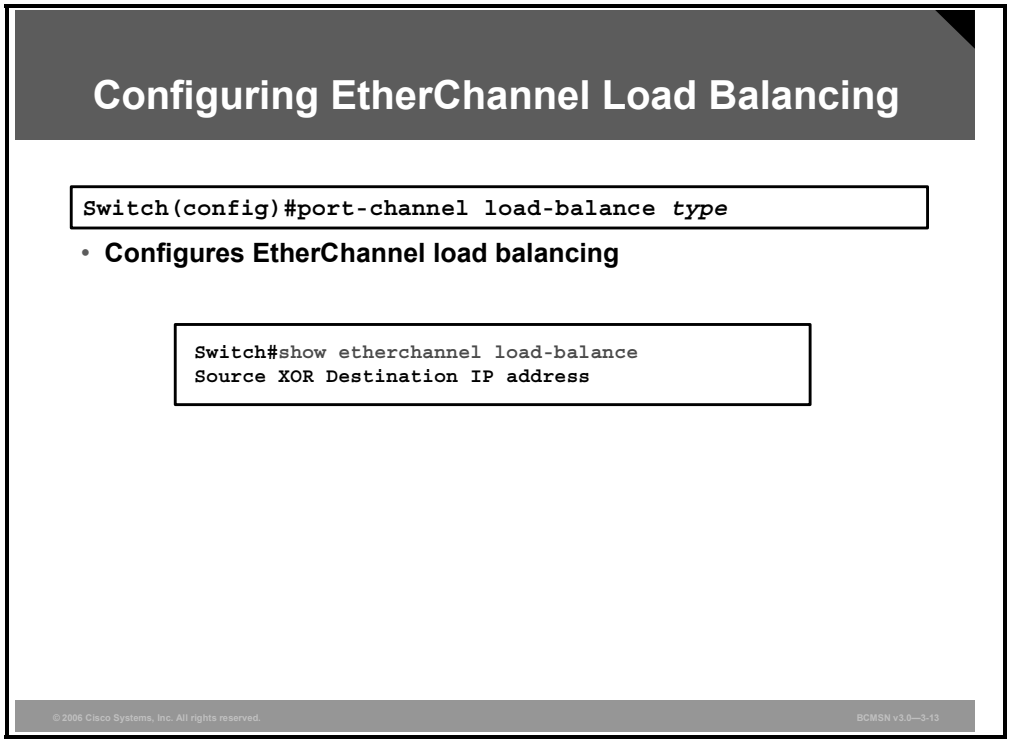
Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel (and the MAC address learned by the switch does not change).

With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination-MAC address of the frame.

Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

EtherChannel Configuration

This subtopic describes the command to configure EtherChannel load balancing.



EtherChannel balances traffic load across the links in a channel. The default and the load balancing method varies among the Cisco Catalyst models.

Load balancing is applied globally for all EtherChannel bundles in the switch. To configure EtherChannel load balancing, use the **port-channel load-balance** command. Load balancing can be based on these variables. The load-balancing keywords are as follows:

- **src-mac:** Source MAC addresses
- **dst-mac:** Destination MAC addresses
- **src-dst-mac:** Source and destination MAC addresses
- **src-ip:** Source IP addresses
- **dst-ip:** Destination IP addresses
- **src-dst-ip:** Source and destination IP addresses (default)
- **src-port:** Source TCP/User Datagram Protocol (UDP) port
- **dst-port:** Destination TCP/UDP port
- **src-dst-port:** Source and destination TCP/UDP port

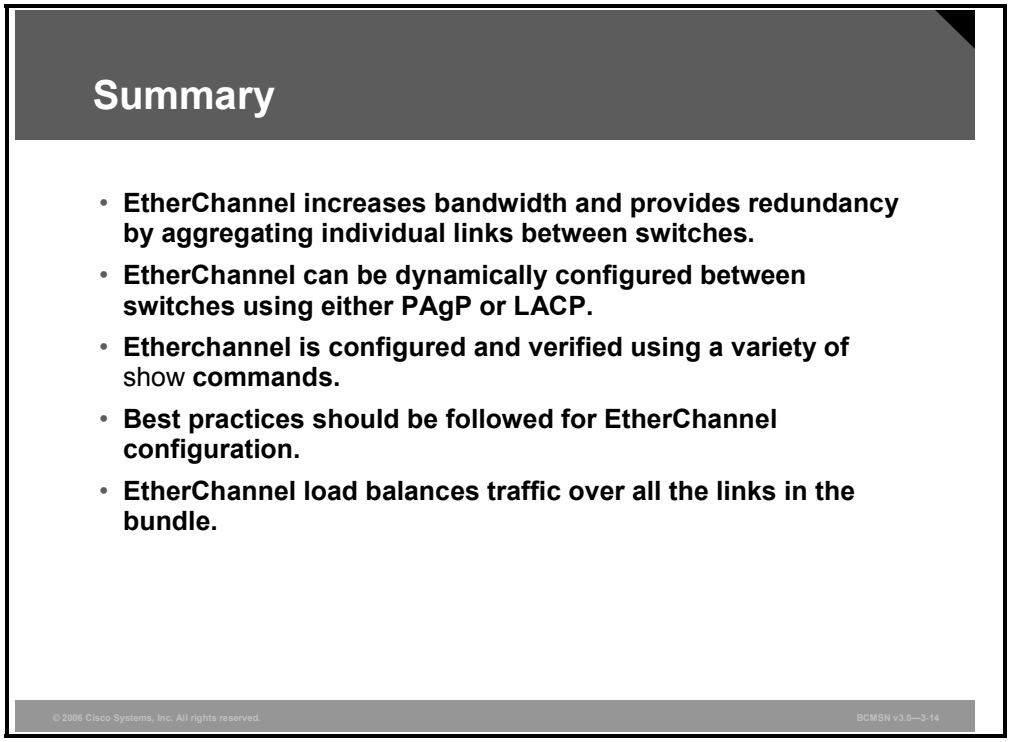
Configuring and Verifying EtherChannel Load Balancing

This example shows how to configure and verify EtherChannel load balancing.

```
Switch(config)# port-channel load-balance src-dst-ip
Switch(config)# exit
Switch# show etherchannel load-balance

Source XOR Destination IP address
```

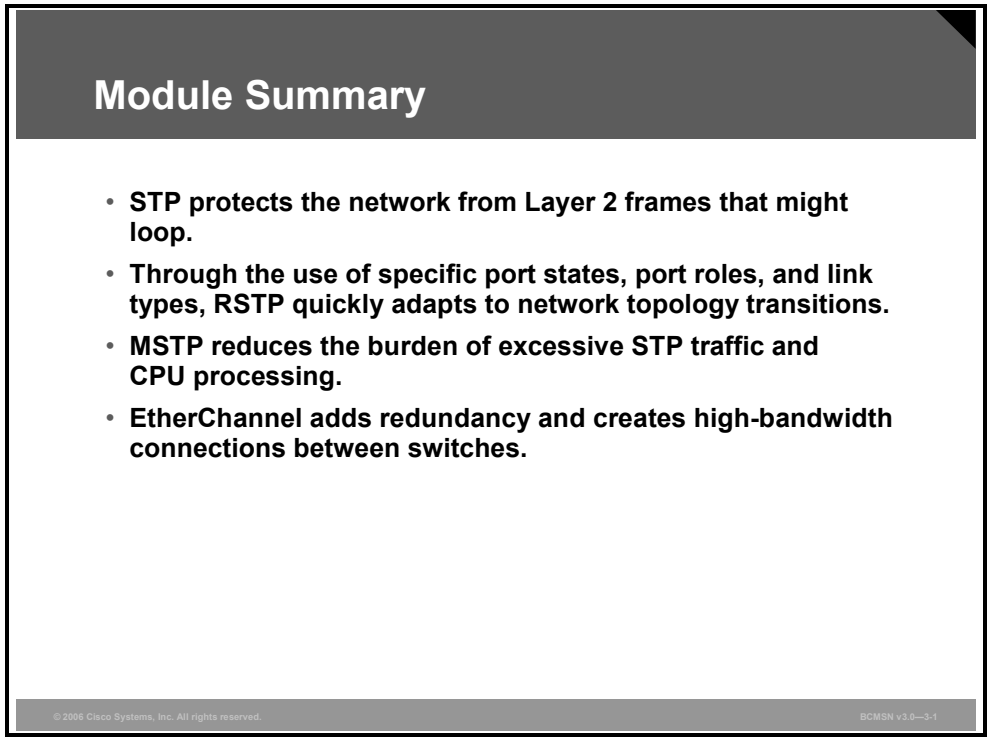
Summary

This topic summarizes the key points discussed in this lesson.



Module Summary

This topic summarizes the key points discussed in this module.



This module reviewed the fundamentals of the Spanning Tree Protocol (STP) operation in a switched network. Many enhancements have been made to the original 802.1D STP protocol.

A switched network can quickly adapt to topology changes by implementing Rapid Spanning Tree Protocol (RSTP).

Multiple Spanning Tree Protocol (MSTP) implements a minimal number of STP instances in a switched environment. Best practices and guidelines for EtherChannel were examined.

References

For additional information, refer to these resources:

- Cisco Systems, Inc., *Spanning-Tree Protocol Enhancements using Loop Guard and BPDU Skew Detection Features:* http://www.cisco.com/en/US/partner/tech/tk389/tk621/technologies_tech_note09186a0080094640.shtml
- Cisco Systems, Inc., *Understanding Spanning-Tree Protocol Topology Changes:* http://www.cisco.com/en/US/partner/tech/tk389/tk621/technologies_tech_note09186a0080094797.shtml
- Cisco Systems, Inc., *Understanding and Configuring Backbone Fast on Catalyst Switches:* http://www.cisco.com/en/US/partner/tech/tk389/tk621/technologies_tech_note09186a00800c2548.shtml

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) Which two statements best describe how spanning tree uses STA to prevent bridging loops in a redundant network? (Choose two.) (Source: Describing the STP)

A) One or more ports are blocked.
B) Some ports send out a broadcast message.
C) Alternative switches are introduced.
D) Some redundant paths are blocked.

Q2) Which two are valid port roles in RSTP? (Choose two.) (Source: Implementing RSTP)

A) alternative port
B) backup port
C) listening port
D) learning port

Q3) Which two features apply to MSTP? (Choose two.) (Source: Implementing MSTP)

A) It groups a set of instances to a single VLAN.
B) It can group a set of VLANs to a single spanning tree instance.
C) A failure in one instance can cause a failure in another instance.
D) The total number of spanning tree instances should match the number of redundant switch paths.

Q4) Which two protocol choices do you have when you are implementing an EtherChannel bundle? (Choose two.) (Source: Configuring Link Aggregation with EtherChannel)

A) PAgP
B) PAgD
C) LACP
D) LAPD

Module Self-Check Answer Key

Q1) A, D

Q2) A, B

Q3) B, D

Q4) A, C

Module 4

Implementing Inter-VLAN Routing

Overview

A switch with multiple VLANs requires a means of passing Layer 3 traffic between those VLANs. This module describes both the process and various methods of routing traffic from VLAN to VLAN. A router that is external to the Layer 2 switch that is hosting the VLANs can provide the inter-VLAN routing. When routing occurs within a Cisco Catalyst multilayer switch, Cisco Express Forwarding (CEF) is deployed to facilitate Layer 3 switching through hardware-based tables, providing an optimal packet-forwarding process. When CEF is implemented, routing is enabled between VLANs through the configuration of switch virtual interfaces (SVIs) that are associated with the various VLANs on the multilayer switch.

Module Objectives

Upon completing this module, you will be able to implement and verify inter-VLAN routing. This ability includes being able to meet these objectives:

- Explain how routing occurs between VLANs
- Implement and verify SVI and routed ports on Cisco Catalyst switches
- Describe the operation of CEF in a multilayer switch environment

Lesson 1

Describing Routing Between VLANs

Overview

Layer 2 switching involves processing frames with respect to their data link layer headers. Information from those headers is stored within the content addressable memory (CAM) table in the switch, which in turn provides the information required to make the forwarding decisions as frames traverse the switch. When multiple Layer 2 VLANs are configured on a switch, a Layer 3 process is required for inter-VLAN communication. VLAN-to-VLAN packet transfer can occur on a Layer 3 device external to the switch.

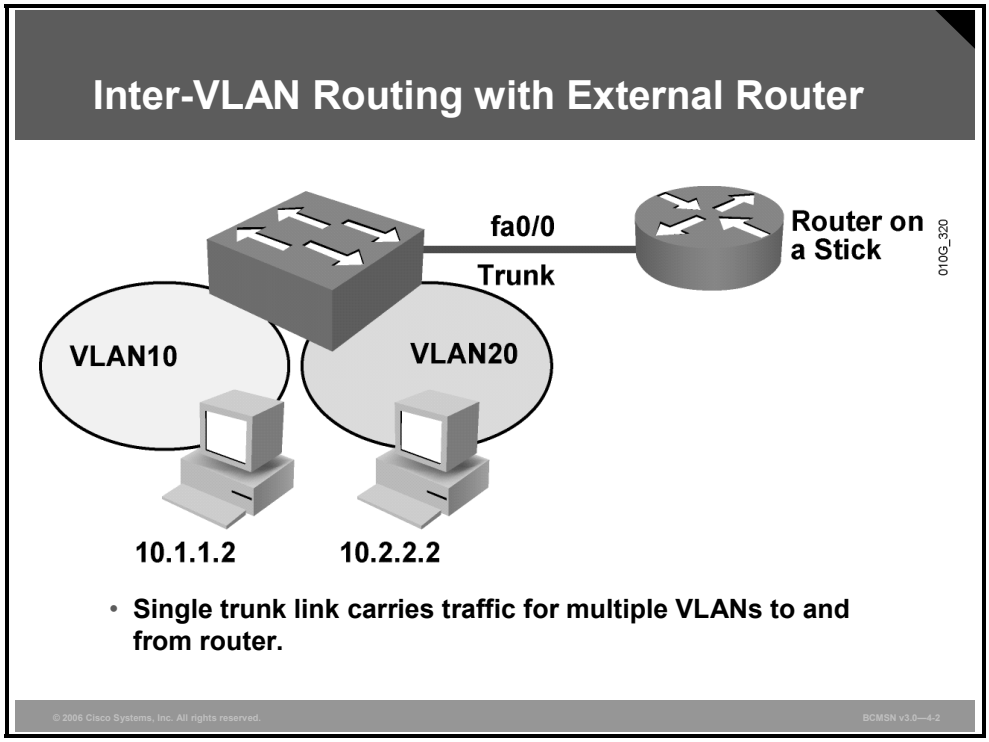
Objectives

Upon completing this lesson, you will be able to implement and verify inter-VLAN routing. This ability includes being able to meet these objectives:

- Describe how inter-VLAN routing works using an external router
- Describe the commands used to configure inter-VLAN routing using an external router
- Explain the procedure to configure inter-VLAN routing using an external router
- Explain how switching interfaces use the forwarding engine to implement Layer 2 and Layer 3 switching
- Describe the frame rewrite process

Inter-VLAN Routing Using an External Router

This topic describes how inter-VLAN routing works using an external router.



If a switch supports multiple VLANs but has no Layer 3 capability to route packets between those VLANs, the switch must be connected to a router external to the switch. This setup is most efficiently accomplished by providing a single trunk link between the switch and the router that can carry the traffic of multiple VLANs, which can in turn be routed by the router. This single physical link must be Fast Ethernet or greater to support Inter-Switch Link (ISL) encapsulation, but 802.1Q is supported on 10-Mb Ethernet router interfaces.

In the figure, the clients on VLAN10 need to establish sessions with a server that is in VLAN20. This will require that traffic be routed between the VLANs as described in this table.

How Traffic Is Routed Between VLANs Using an External Router

The table describes the actions necessary for traffic to be routed between VLANs using an external router.

Step	Action
1.	The router accepts the packets from VLAN10 on its subinterface in that VLAN.
2.	The router performs Layer 3 processing based on the destination network address.
3.	Because the destination network is associated with a VLAN accessed over the trunk link, the router applies the appropriate VLAN identification to the Layer 2 header.
4.	The router then routes the packet out the appropriate subinterface on VLAN20.

In the figure, the router can receive packets on one VLAN and forward them to another VLAN. To perform inter-VLAN routing functions, the router must know how to reach all VLANs that are being interconnected.

The router must have a separate logical connection (subinterface) for each VLAN that is running between the switch and the router and ISL, or 802.1Q trunking must be enabled on the single physical connection between the router and switch.

The routing table will show as directly connected to all the subnets associated with the VLANs that are configured on the router subinterfaces. The router must learn routes to networks that are not configured on directly connected interfaces through dynamic routing protocols.

There are both advantages and disadvantages of inter-VLAN routing on an external router.

The advantages are as follows:

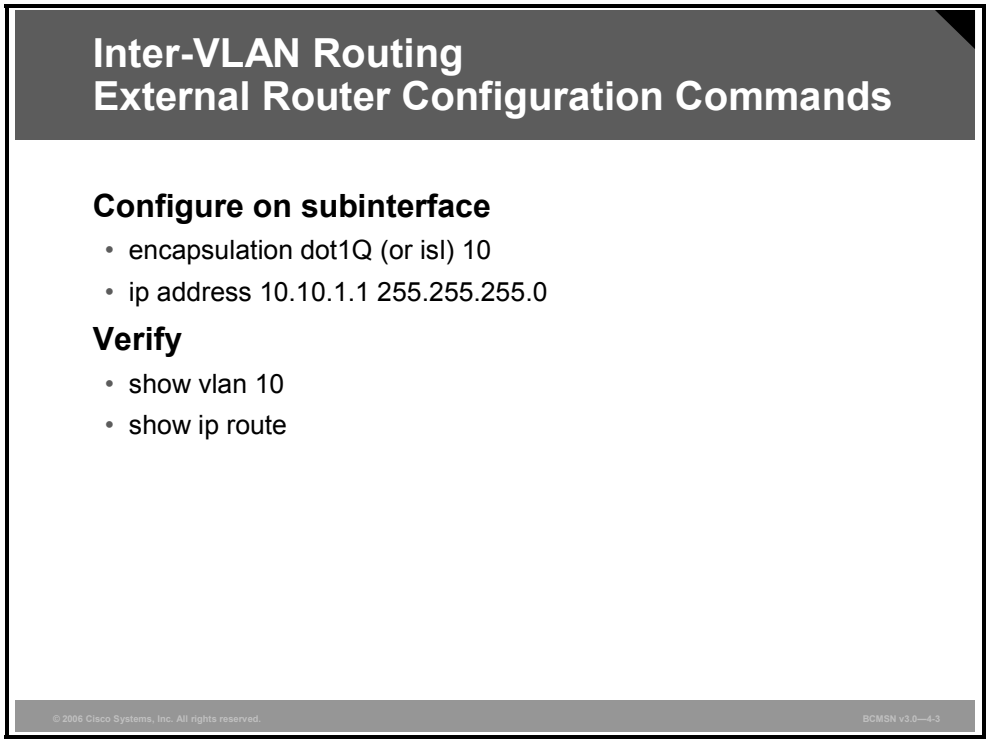
- Implementation is simple.
- Layer 3 services are not required on the switch.
- The router provides communication between VLANs.

The disadvantages are as follows:

- The router is single point of failure.
- Single traffic path may become congested.
- Latency may be introduced as frames leave the switch chassis.

Describing Inter-VLAN Routing Using External Router Configuration Commands

This topic describes the commands used to configure inter-VLAN routing on an external router.



Inter-VLAN routing can be configured using an external router over either ISL or 802.1Q trunks. The commands for configuring the trunk interface on the router are shown in the table.

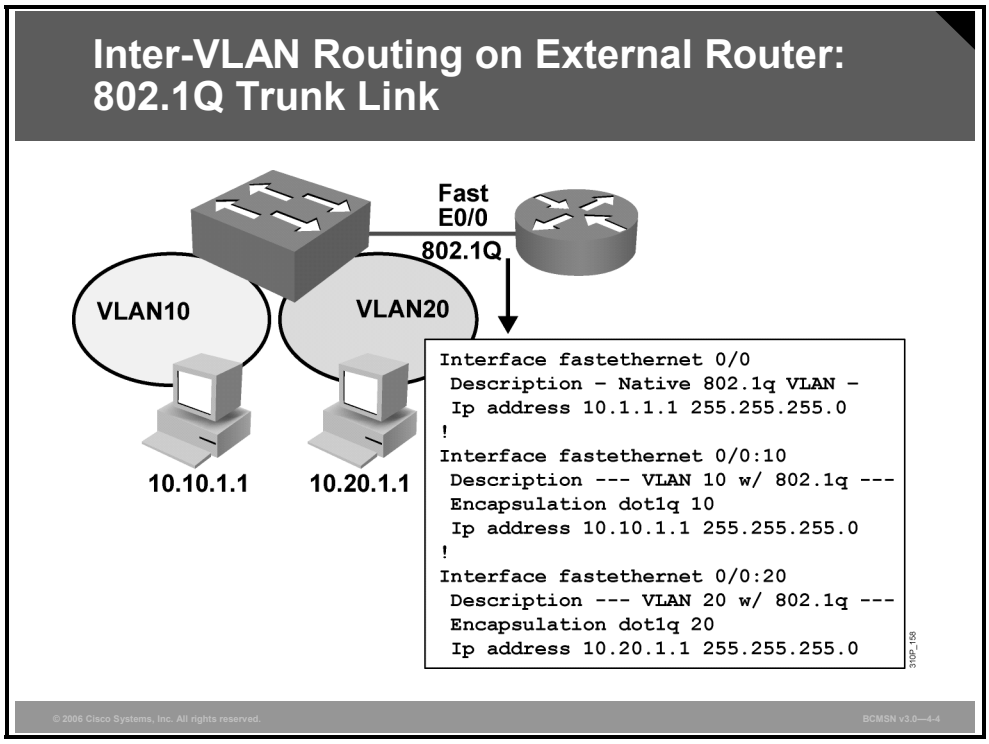
Commands for Inter-VLAN Routing on an External Router

The table provides a description of the commands used to perform inter-VLAN routing on an external router.

Command	Description
`Router(config-subif)#` **`encapsulation dot1Q`** *`vlan_ID`*	This command sets the encapsulation to 802.1Q on the subinterface and associates that subinterface with a particular VLAN. This command is repeated for each VLAN instance on the trunk.
`Router(config-subif)#` **`encapsulation isl`** *`vlan_ID`*	This command sets the encapsulation to ISL on the subinterface and associates that subinterface with a particular VLAN. This command is repeated for each VLAN instance on the trunk.
`Router#` **`show vlan`** [*`vlan_ID`*]	This command displays the subinterface-to-VLAN mappings, the trunk protocol configured, and the number of packets received and transmitted on each VLAN through the configured subinterfaces.
`Router#` **`show ip protocols`**	This command displays the configured routing protocols on the router, along with the interfaces and networks configured for routing.
`Router#` **`show ip route`**	This command displays the routing information that will be used to move data between VLANs on this device.

Configuring Inter-VLAN Routing Using an External Router

This topic illustrates how to configure routing between VLANs using an external router that uses 802.1Q.



A router interface providing inter-VLAN routing on a trunk link must be configured with a subinterface for each VLAN that will be serviced across the link. Each subinterface on the physical link must then be configured with the same trunk encapsulation protocol. That protocol, either 802.1Q or ISL, is typically determined by what was configured on the switch side of the link.

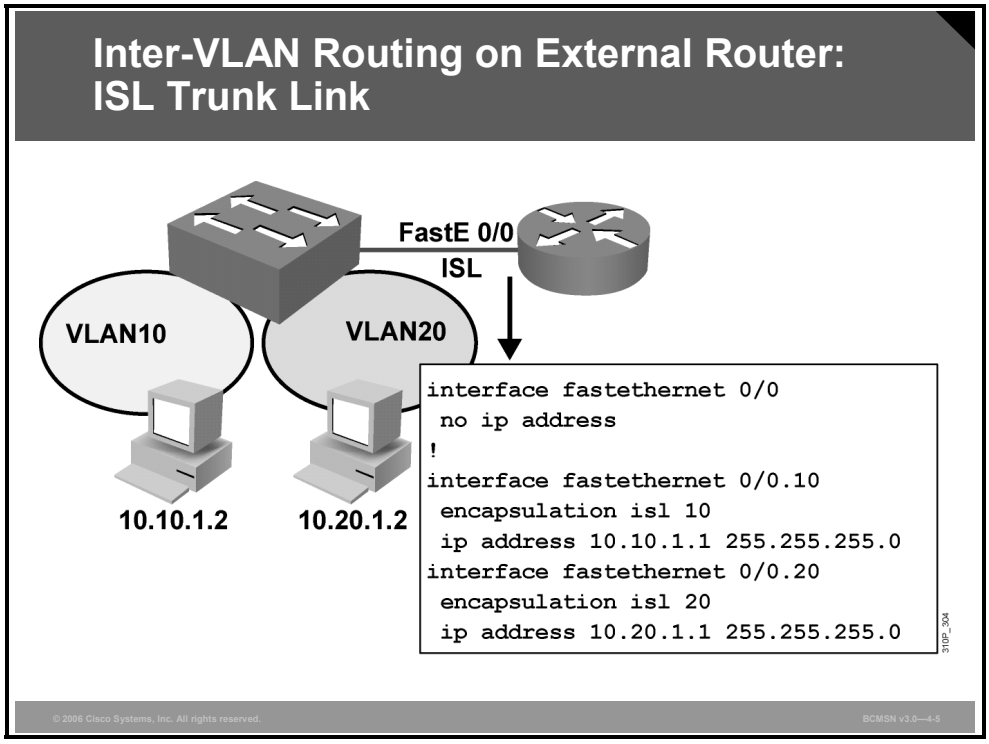
Use the **encapsulation dot1q** subinterface configuration command to enable 802.1Q encapsulation on a router subinterface. The subinterface number need not match the dot-1Q VLAN number, but it is good practice to do so.

Because traffic on the native VLAN is not tagged, all native VLAN frames will be received as normal Ethernet frames, so there is no need to define a specific encapsulation tag for those networks. Some versions of IOS allow for the creation of a subinterface for the native VLAN. If the native VLAN is configured as a subinterface, the command **encapsulation dot1q <*vlan*> native** should be used. All other, non-native VLANs do have an 802.1Q tag inserted into their frames. These non-native VLANs should always be configured as subinterfaces on the router, and the VLANs must be defined as 802.1Q tagged frames, with the VLAN that is associated with them being identified. The subinterface command **encapsulation dot1q <*vlan*>** accomplishes this task.

The subnets of the VLANs are directly connected to the router. Routing between these subnets does not require a dynamic routing protocol because the subnets are directly connected. Routes to the subnets that are associated with each VLAN will appear in the routing table as directly connected interfaces.

Configuring an External Router Using ISL

This subtopic illustrates how to configure routing between VLANs using an external router using ISL.



Configuring an External Router Using ISL Encapsulation

Use the **encapsulation isl** *vlan_id* subinterface configuration command to enable ISL trunking on a router subinterface.

The **native** keyword is not used on the **encapsulation ISL** subinterface command because ISL does not have the concept of a native VLAN.

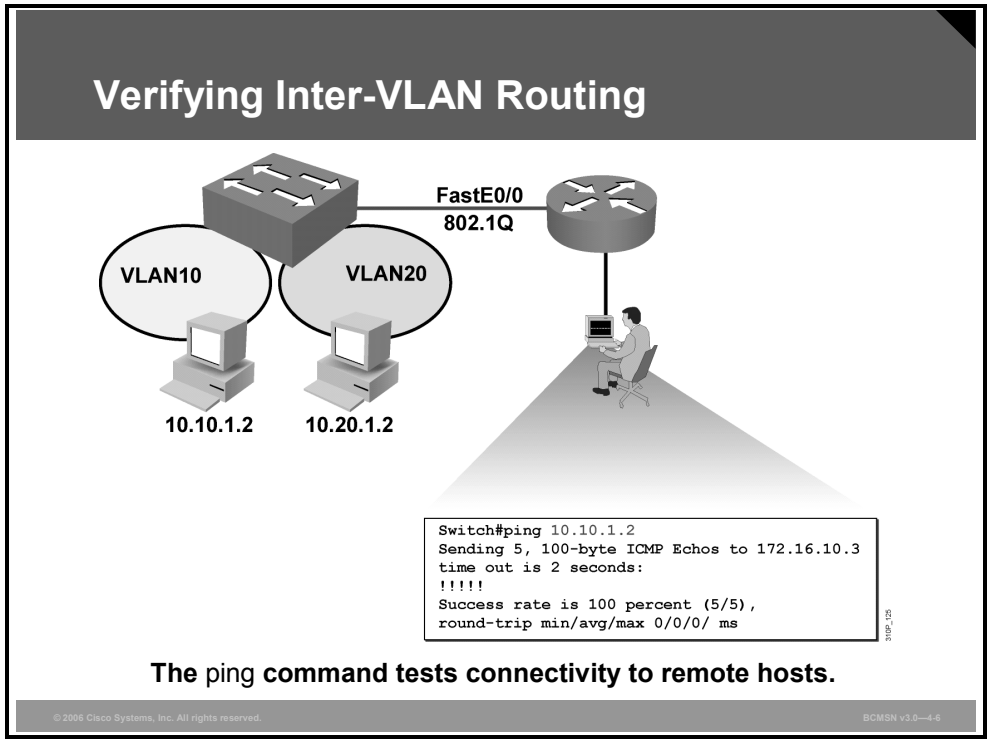
ISL Encapsulation Commands for External Routers

The table describes the actions needed to perform ISL encapsulation on external routers.

Step	Action
1.	Enable ISL trunking on the switch port connecting to the router.
2.	Enable ISL encapsulation on the router Fast Ethernet subinterface.
3.	Assign a network layer address to each subinterface.

Verifying the Inter-VLAN Routing Configuration Using ping

This subtopic discusses how to verify the inter-VLAN routing configuration using **ping**.



After the router is properly configured and connected to the network, the router or the switch can communicate with other nodes on the network.

To test connectivity to remote hosts, use the **ping** command from privileged mode:

```
Switch#ping destination-ip-address
```

Step 1 From the router, attempt to ping a host address on each VLAN to verify router connectivity.

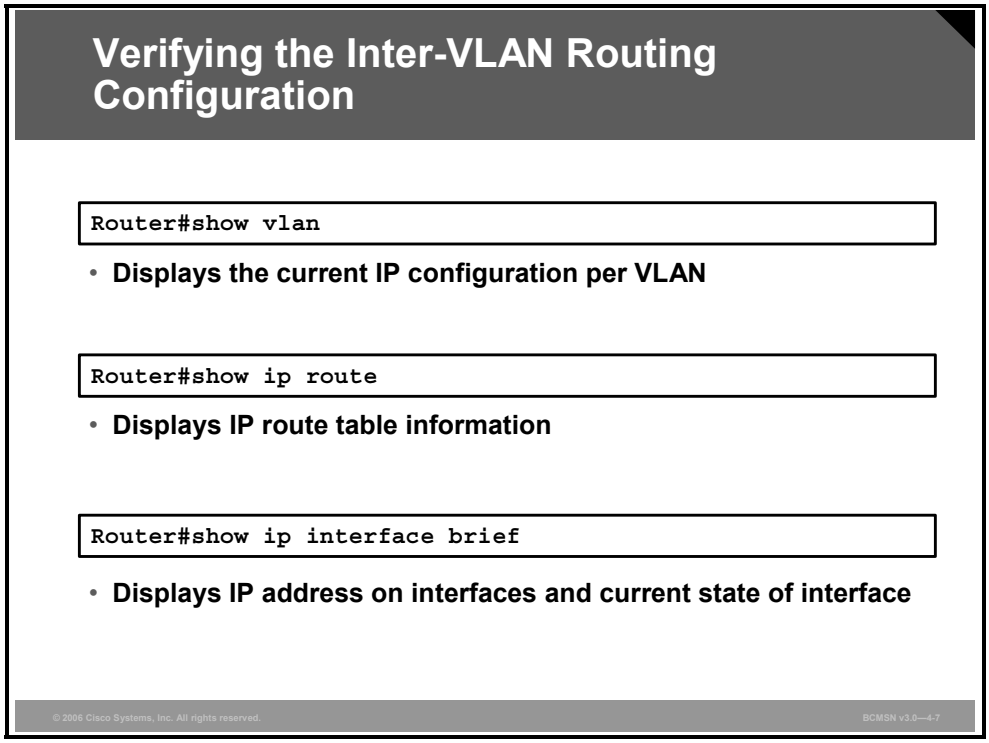
Step 2 From a host on a particular VLAN, attempt to ping a host on another VLAN to verify routing across the external router.

The **ping** command will return one of these responses:

- **Success rate is 100 percent or *ip-address* is alive:** This response occurs in 1 to 10 ms, depending on network traffic and the number of Internet Control Message Protocol (ICMP) packets sent.
- **Destination does not respond:** No answer message is returned if the host does not respond.
- **Unknown host:** This response occurs if the targeted host cannot be resolved.
- **Destination unreachable:** This response occurs if the default gateway cannot reach the specified network or is being blocked.
- **Network or host unreachable:** This response occurs if the Time to Live (TTL) times out. The TTL default is 2 seconds.

Verifying the Inter-VLAN Routing Configuration

This subtopic describes commands used to verify inter-VLAN routing configuration.



Use the **show** commands to display the current (running) configuration, IP routing information, and IP protocol information. This will verify if the routing table represents the subnets of all VLANs.

Example: Displaying Inter-VLAN Configuration Information

The following examples of the **show** commands displays first, a snapshot of inter-VLAN status, and second, the routing table in use:

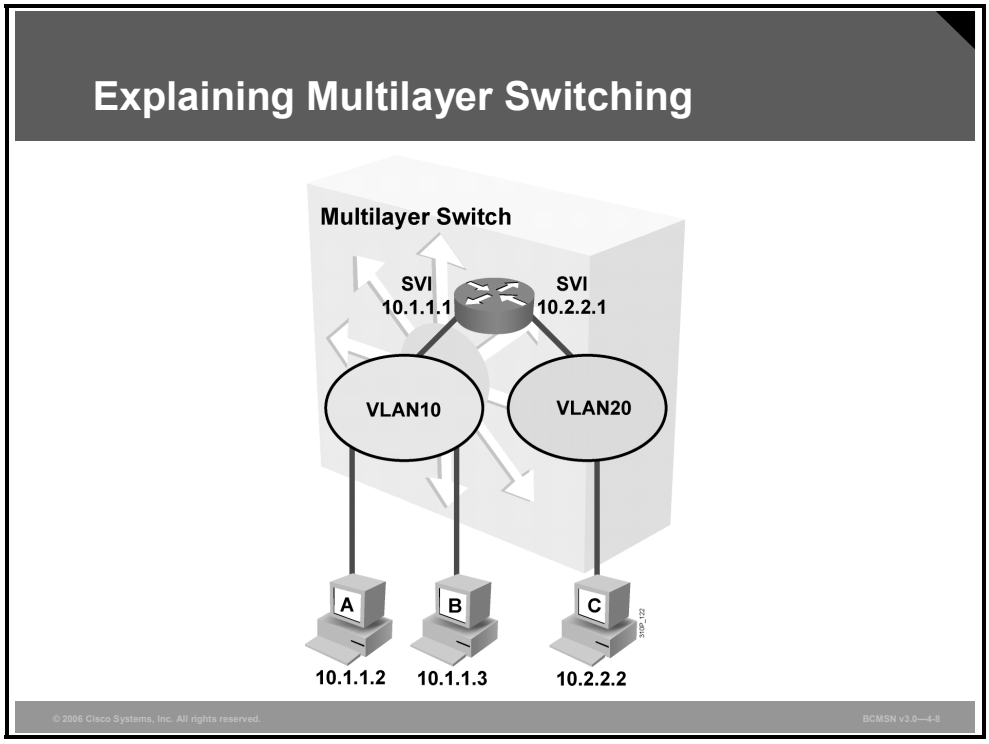
```
Router#show vlans
Virtual LAN ID:  10 (Inter Switch Link Encapsulation)
   vLAN Trunk Interface:   FastEthernet0/0.10
   Protocols Configured:   Address:              Received:        Transmitted:
           IP              10.10.1.1                    0          20

Virtual LAN ID:  20 (Inter Switch Link Encapsulation)
   vLAN Trunk Interface:   FastEthernet0/0.20
   Protocols Configured:   Address:              Received:        Transmitted:
           IP              10.20.1.1                    0          20
```

Example: Displaying Routing Table Information

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
     10.0.0.0/24 is subnetted, 2 subnets
C       10.10.1.0 is directly connected, FastEthernet0/0.10
C       10.20.1.0 is directly connected, FastEthernet0/0.20
```

Explaining Multilayer Switching

This topic explains how switching interfaces use the forwarding engine to implement Layer 2 and Layer 3 switching.



Traditionally, a switch makes forwarding decisions by looking at the Layer 2 header, whereas a router makes forwarding decisions by looking at the Layer 3 header.

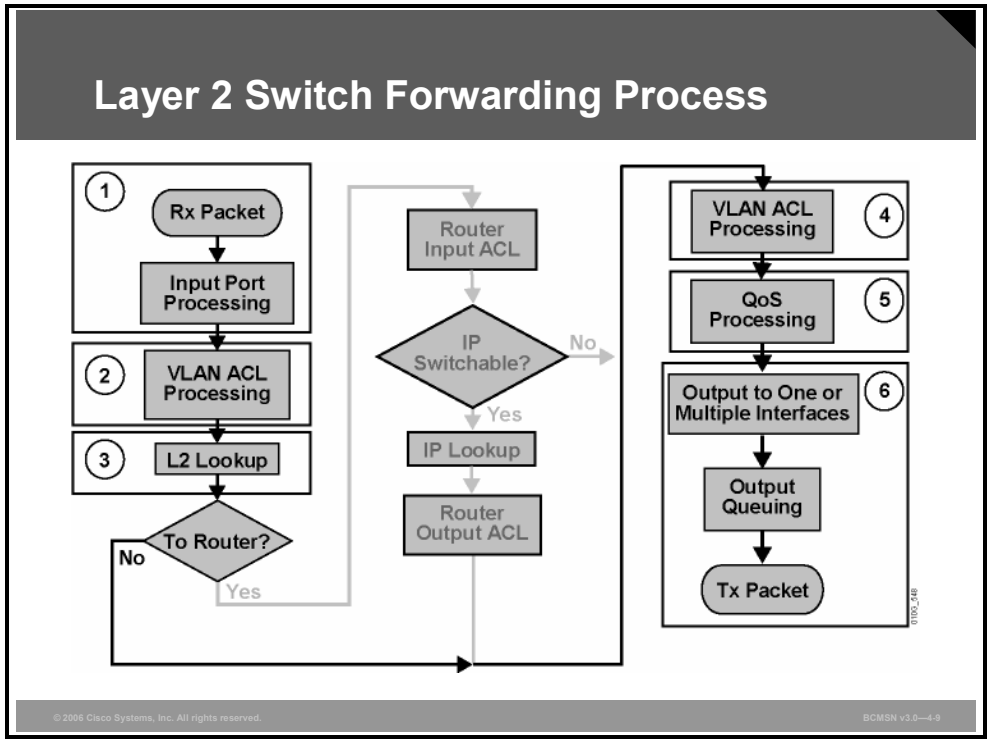
A multilayer switch combines the functionality of a switch and a router into one device, thereby enabling the device to switch traffic when the source and destination are in the same VLAN and to route traffic when the source and destination are in different VLANs (that is, different subnets).

In the figure, traffic between PC A and PC B will be switched at Layer 2, whereas traffic between PC B and PC C will be switched at Layer 3.

Multilayer switches forward frames and packets at wire speed by using ASIC hardware. Specific Layer 2 and Layer 3 components, such as routing tables or access control lists (ACLs), are cached into hardware. These tables are stored in CAM and ternary content addressable memory (TCAM).

Layer 2 Switch Forwarding

This subtopic describes how a switch processes a Layer 2 frame.



Layer 2 forwarding in hardware is based on the destination MAC address. The Layer 2 switch learns the address, based on the source MAC address. The MAC address table lists MAC and VLAN pairs with associated interfaces.

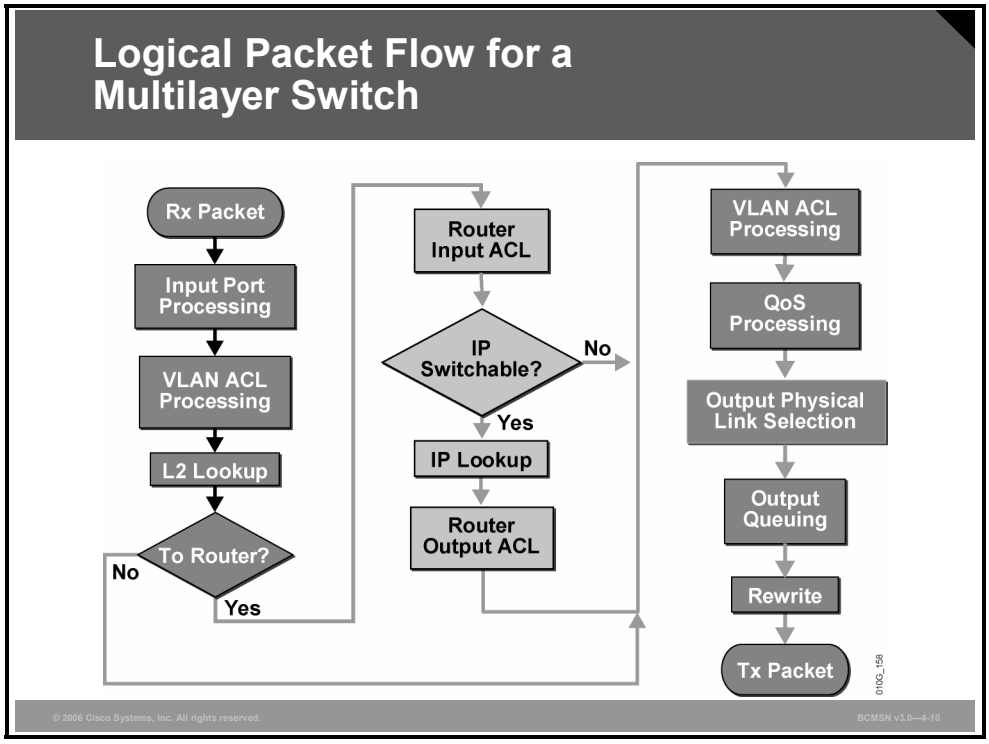
How a Layer 2 Switch Forwards Packets

The table describes how a Layer 2 switch forwards packets.

Step	Action
1.	The Layer 2 engine receives a frame.
2.	The Layer 2 engine performs the input ACL lookup.
3.	The Layer 2 lookup engine looks up the destination MAC address and determines if the frame is to be switched at Layer 2 or Layer 3.
4.	If the frame is to be switched at Layer 2, then the Layer 2 forwarding engine performs the outbound security ACL lookup.
5.	The Layer 2 forwarding engine performs the outbound quality of service (QoS) lookup.
6.	The Layer 2 forwarding engine forwards the packet.

Layer 3 Switch Forwarding

This subtopic describes how a Layer 3 packet is processed.



Layer 3 forwarding is based on the destination IP address. Layer 3 forwarding occurs when a packet is routed from a source in one subnet to a destination in another subnet. When a multilayer switch sees its own MAC address in the Layer 2 header, it recognizes that the packet is either destined for itself or has been sent to the default gateway. If the packet is not destined for the multilayer switch, then the destination IP address is compared against the Layer 3 forwarding table for the longest match. In addition, any router ACL checks are performed.

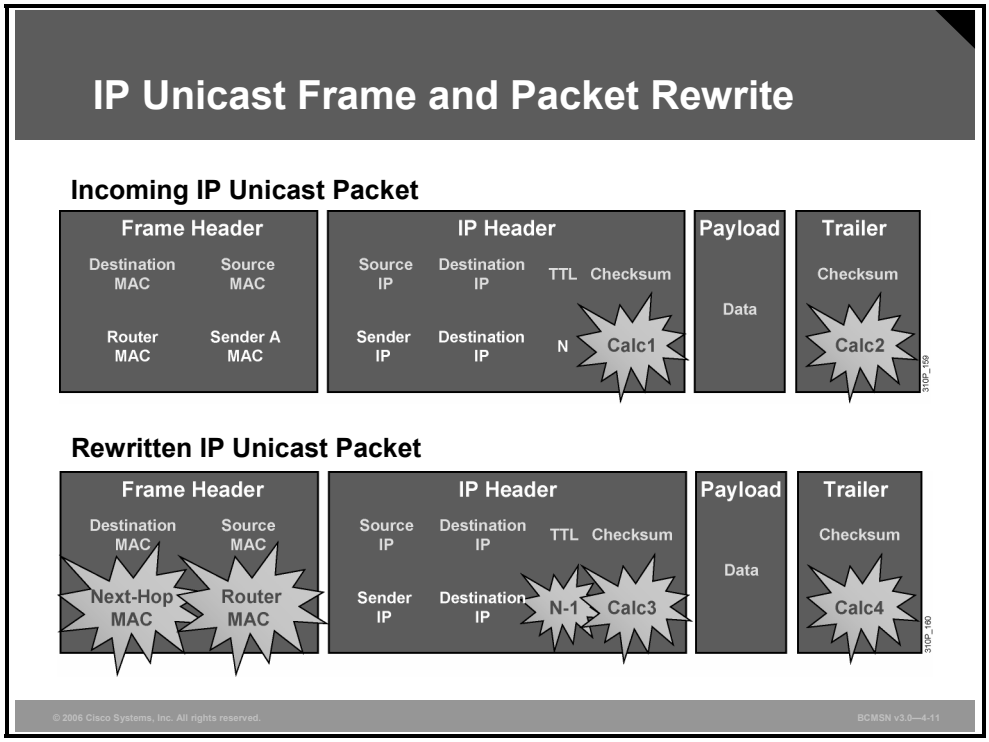
How a Layer 3 Switch Forwards Packets

The table describes how a Layer 3 switch forwards packets.

Step	Action
1.	The Layer 2 engine receives a frame.
2.	The Layer 2 engine performs the input ACL lookup.
3.	The Layer 2 lookup engine recognizes the MAC address of the multilayer switch and therefore determines that the packet is to be switched at Layer 3.
4.	If necessary, an input router ACL check is performed.
5.	The destination IP address is compared against the Layer 3 forwarding table for the longest match.
6.	If necessary, an output router ACL check is performed.
7.	The Layer 2 forwarding engine performs the outbound QoS lookup.
8.	The Layer 2 and Layer 3 header are rewritten.
9.	The Layer 2 forwarding engine forwards the packet.

Frame Rewrite

This topic describes the frame rewrite process.

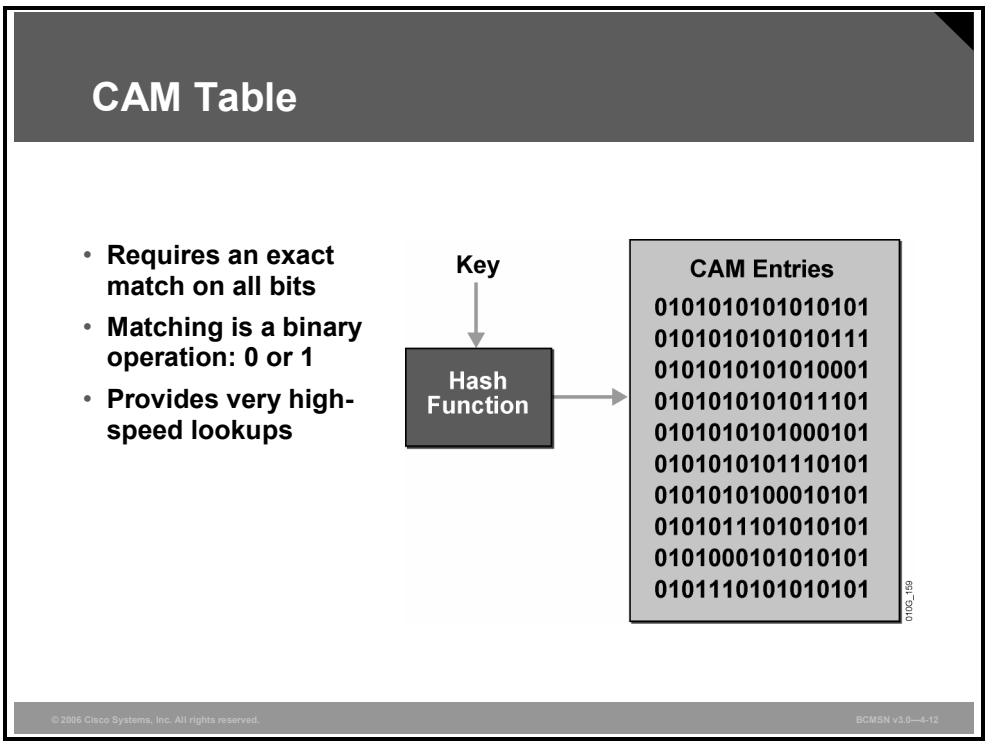


The figure shows how the frame and packet header would be altered when Cisco Express Forwarding (CEF) is used to forward frames. IP unicast packets are rewritten on the output interface as follows:

- The source MAC address changes from the sender MAC address to the router MAC address.
- The destination MAC address changes from the router MAC address to the next-hop MAC address.
- The TTL is decremented by one and, as a result, the IP header checksum is recalculated.
- The frame checksum must be recalculated.

Which Switching Tables Are Used?

This subtopic describes the multilayer switching table architectures.



Routing, switching, ACL, and QoS tables are stored in a high-speed table memory so that forwarding decisions and restrictions can be made in high-speed hardware. Cisco Catalyst switches have two primary table architectures:

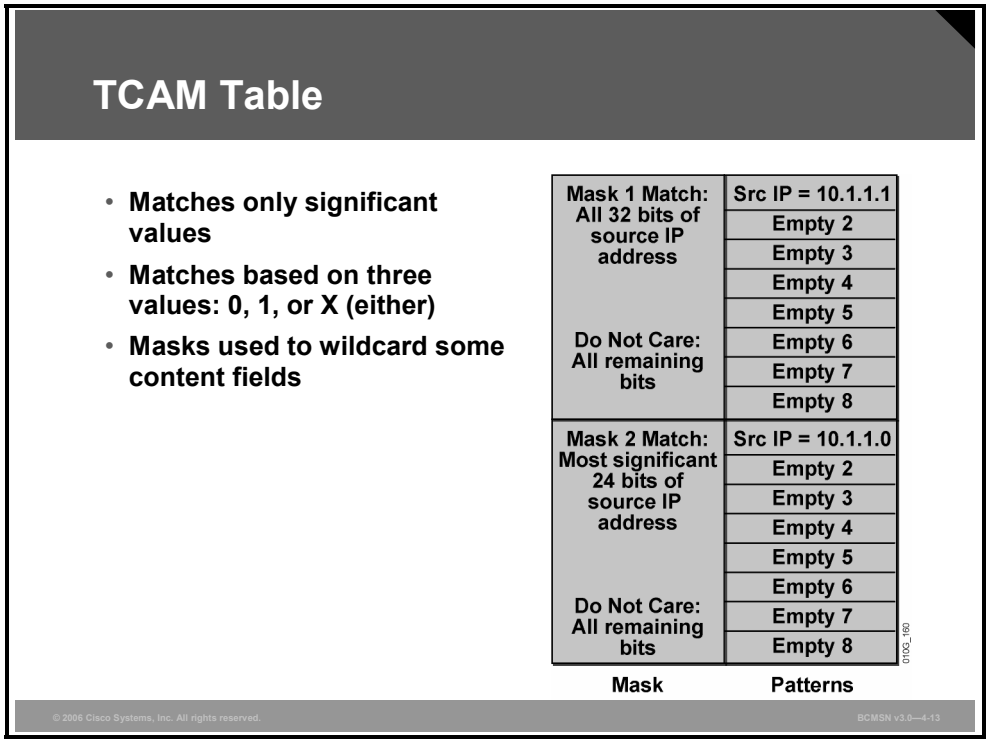
- **CAM table:** The CAM table is the primary table used to make Layer 2 forwarding decisions. The table is built by recording the source address and inbound port of all frames. When a frame arrives at the switch with a destination MAC address of an entry in the CAM table, the frame is forwarded out through only the port that is associated with that specific MAC address.
- **TCAM table:** The TCAM table stores ACL, QoS, and other information generally associated with upper-layer processing.

Table lookups are done with efficient search algorithms. A "key" is created to compare the frame to the table content. For example, the destination MAC address and VLAN ID (VID) of a frame would constitute the key for Layer 2 table lookup. This key is fed into a hashing algorithm, which produces a pointer into the table. The system uses the pointer to access a smaller specific area of the table without requiring a search of the entire table.

In a Layer 2 table, all bits of all information are significant for frame forwarding (for example, VLANs, destination MAC addresses, and destination protocol types). However, in more complicated tables associated with upper-layer forwarding criteria, some bits of information may be too inconsequential to analyze. For example, an ACL may require a match on the first 24 bits of an IP address, but the last 8 bits may be insignificant information.

TCAM Table

This subtopic describes how the TCAM is used.



In specific high-end switch platforms, the TCAM is a portion of memory designed for rapid, hardware-based table lookups of Layer 3 and Layer 4 information. In the TCAM, a single lookup provides all Layer 2 and Layer 3 forwarding information for frames, including CAM and ACL information.

The figure displays the ACL information stored in the TCAM table that would result in a packet being permitted or denied.

TCAM matching is based on three values: 0, 1, or *x* (where *x* is either number), hence the term "ternary." The memory structure is broken into a series of patterns and masks. Masks are shared among a specific number of patterns and are used as wildcards in some content fields.

These two ACL entries are referenced in the figure because it shows how their values would be stored in the TCAM:

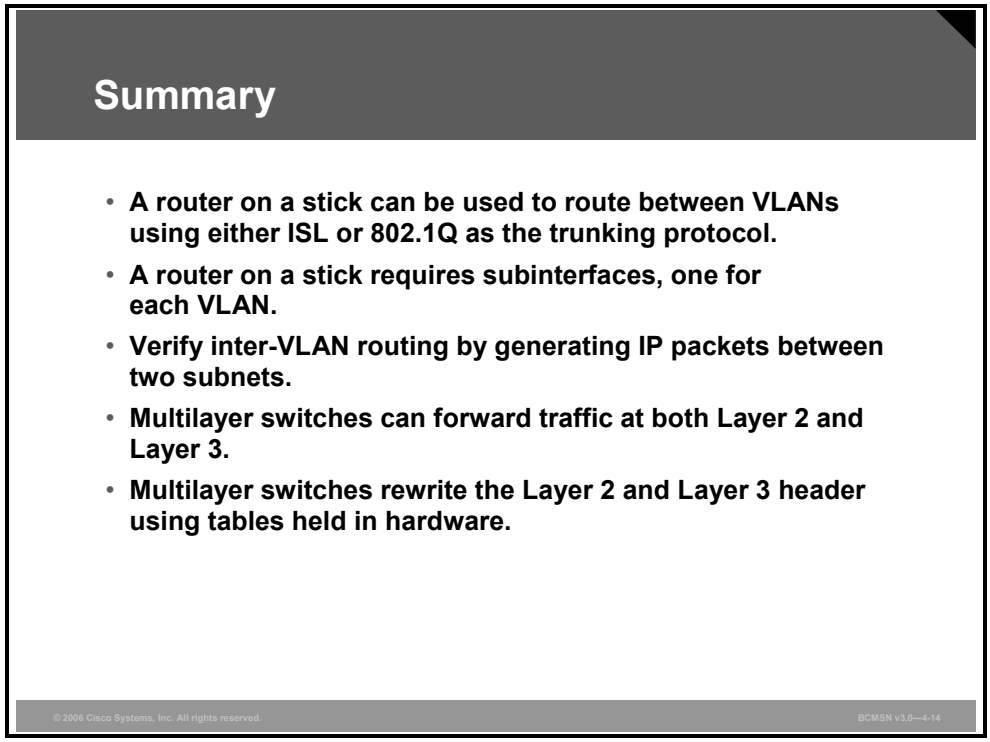
```
access-list 101 permit ip host 10.1.1.1 any
access-list 101 deny ip 10.1.1.0 0.0.0.255 any
```

The TCAM table entries in the figure consist of two types of regions:

- **Longest-match region:** Each longest-match region consists of groups of Layer 3 address entries ("buckets") organized in decreasing order by mask length. All entries within a bucket share the same mask value and key size. The buckets can change their size dynamically by borrowing address entries from neighboring buckets. Although the size of the whole protocol region is fixed, you can reconfigure it. The reconfigured size of the protocol region takes effect only after the next system reboot.
- **First-match region:** The first-match region consists of ACL entries. Lookup stops after first match of the entry.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 2

Enabling Routing Between VLANs on a Multilayer Switch

Overview

When multiple VLANs are configured on a multilayer switch, routing between those VLANs can occur on the switch itself through the configuration of Layer 3 switch virtual interfaces (SVIs). SVIs are configured and verified using Layer 3 Cisco IOS commands to facilitate inter-VLAN routing on a multilayer switch. It is also possible to convert Layer 2 switch ports to operate as Layer 3 interfaces.

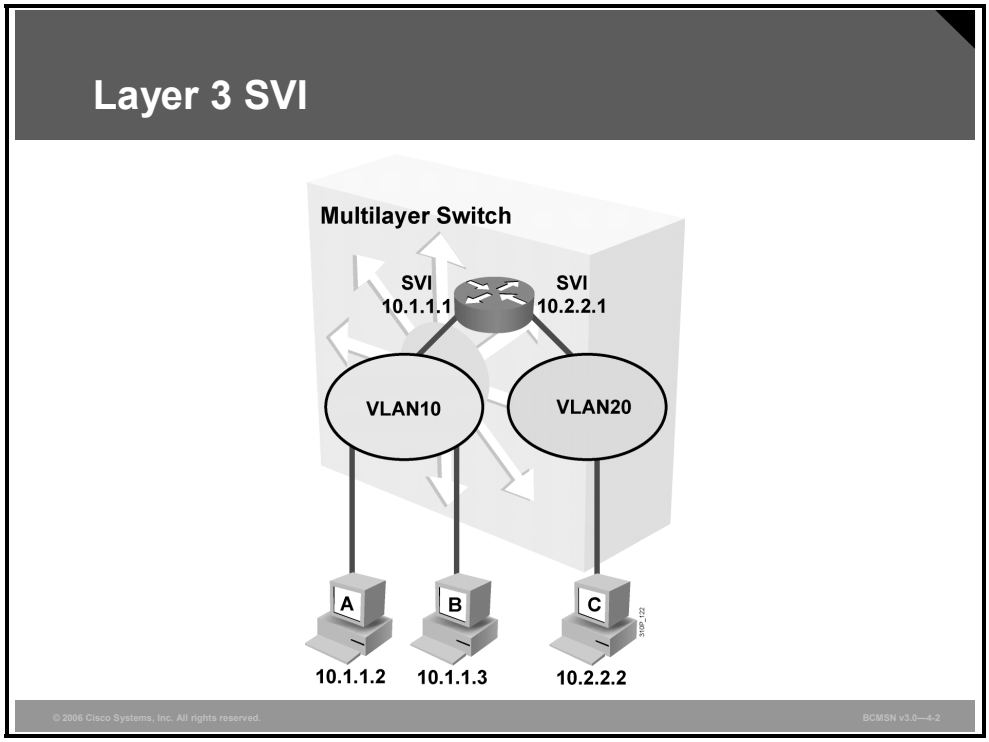
Objectives

Upon completing this lesson, you will be able to implement and verify SVI and routed ports on Cisco Catalyst switches. This ability includes being able to meet these objectives:

- Describe a Layer 3 SVI
- Describe commands used to configure inter-VLAN routing on a multilayer switch through an SVI
- Explain the procedure to configure inter-VLAN routing on a multilayer switch
- Describe a routed port on a multilayer switch
- Describe commands used to configure a routed port on a multilayer switch
- Explain the procedure to configure routed ports on a multilayer switch

Describing Layer 3 SVI

This topic describes a VLAN SVI.



An SVI is a virtual Layer 3 interface that can be configured for any VLAN that exists on a Layer 3 switch. It is "virtual" in that there is no physical interface for the VLAN, and yet it can accept configuration parameters applied to any Layer 3 router interface. The SVI for the VLAN provides Layer 3 processing for packets from all switch ports associated with that VLAN. Only one SVI can be associated with a VLAN. You configure an SVI for a VLAN for these reasons:

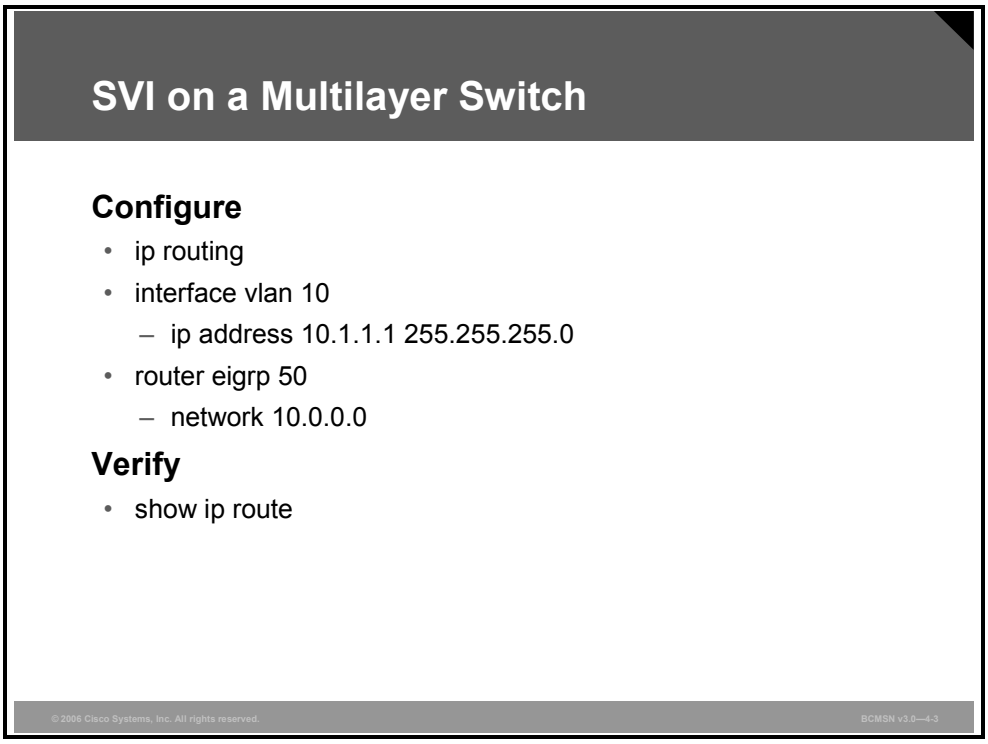
- To provide a default gateway for a VLAN so that traffic can be routed between VLANs
- To provide fallback bridging if it is required for nonroutable protocols
- To provide Layer 3 IP connectivity to the switch
- To support routing protocol and bridging configurations

By default, an SVI is created for the default VLAN (VLAN1) to permit remote switch administration. Additional SVIs must be explicitly created.

SVIs are created the first time a VLAN interface configuration mode is entered for a particular VLAN SVI. The VLAN corresponds to the VLAN tag associated with data frames on an Inter-Switch Link (ISL) or 802.1Q encapsulated trunk or to the VLAN ID (VID) configured for an access port. Configure and assign an IP address to each VLAN SVI that is to route traffic off and onto the local VLAN.

Describing Configuration Commands for Inter-VLAN Communication on a Multilayer Switch

This topic describes commands used to configure inter-VLAN routing on a multilayer switch through an SVI.



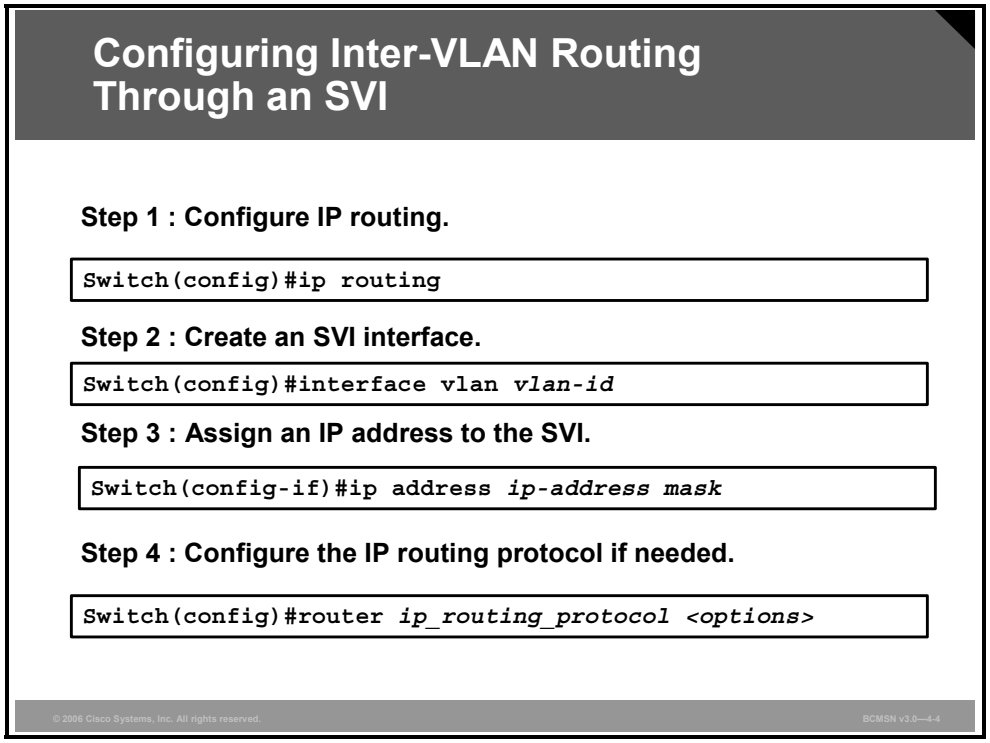
These commands are used to configure inter-VLAN routing on a multilayer switch through an SVI.

Inter-VLAN Routing Through SVI Commands

Command	Description
`Switch(config)#` **`ip routing`**	Enables Layer 3 routing on the switch. Enabled by default on some Cisco Catalyst platforms.
`Switch(config)#` **`interface vlan`** *`vlan-id`*	Creates an SVI for the VLAN and moves to interface configuration mode for that SVI.
`Switch(config-if)#` **`ip address`** *`ip-address mask`*	Assigns a Layer 3 address to the SVI. This will likely be the default gateway for host on that VLAN.
`Switch(config)#` **`router`** *`routing-protocol <options>`*	(Optional) Invokes a dynamic routing protocol, Enhanced Interior Gateway Routing Protocol (EIGRP), so that the routing services on this switch can exchange routing updates with other devices.
`Switch#` **`show ip route`**	Indicates if the networks associated with the SVIs appear in the routing table.

Configuring Inter-VLAN Routing on a Multilayer Switch

This topic explains the procedure to configure inter-VLAN routing on a multilayer switch.



To configure inter-VLAN routing on a Cisco Catalyst SVI, perform these steps.

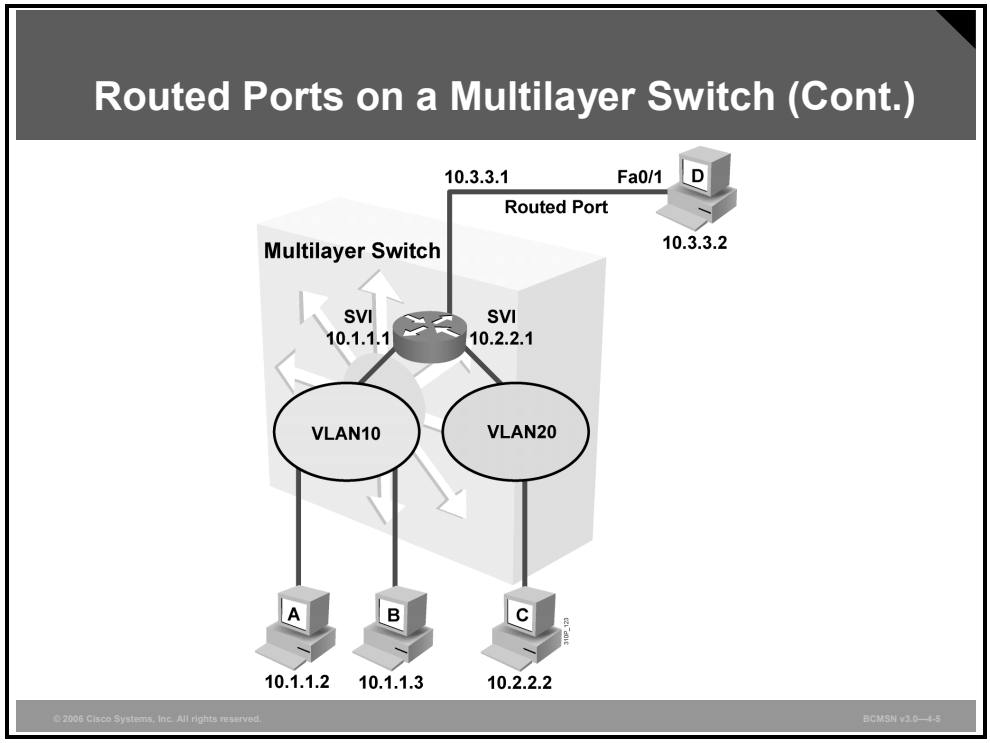
Steps for Inter-VLAN Routing Configuration

The table describes the steps needed to configure inter-VLAN routing.

Step	Description	Notes and Comments
1.	`Switch(config)#`**`ip routing`**	Enable IP routing on the router.
2.	`Switch(config)#`**`interface VLAN`** *`vlan-id`*	Create the SVI interface or navigate to configuration mode for the interface.
3.	`Switch(config-if)#`**`ip address`** *`n.n.n.n subnet-mask`*	Assign an IP address to the SVI for the VLAN.
4.	(Optional) Specify an IP routing protocol. `Switch(config)#`**`router`** *`ip_routing_protocol <options>`*	This step is necessary for the switch to exchange dynamic routing updates with other routing devices. The routing protocol specified may require additional options. Refer to the documentation for the routing protocol for further details.

Describing Commands for Routed Ports on a Multilayer Switch

This topic describes commands used to configure a routed port on a multilayer switch.



A routed switch port is a physical switch port on a multilayer switch that is capable of Layer 3 packet processing. A routed port is not associated with a particular VLAN, as is an access port or SVI.

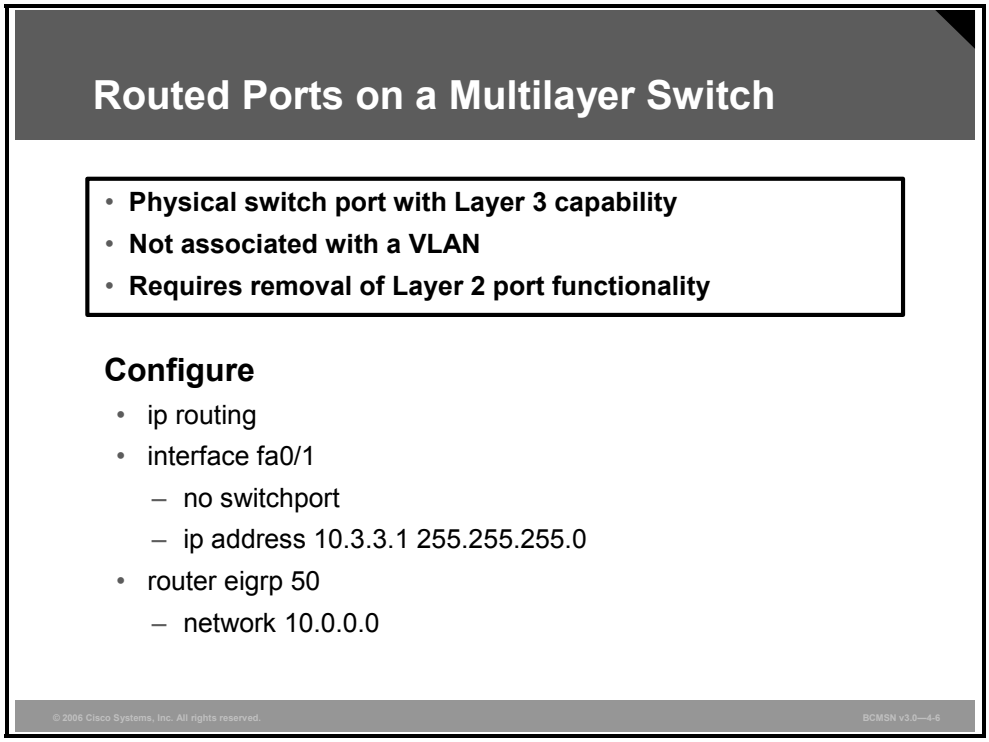
A routed port behaves like a regular router interface, except that it does not support VLAN subinterfaces. Routed switch ports can be configured using most commands applied to a physical router interface, including the assignment of an IP address and the configuration of Layer 3 routing protocols.

A routed switch port is similar to an SVI in that it is a switch port that provides Layer 3 packet processing. SVIs generally provide Layer 3 services for devices connected to the ports of the switch where the SVI is configured. Routed switch ports can provide a Layer 3 path into the switch for a number of devices on a specific subnet, all of which are accessible from a single physical switch port.

The number of routed ports and SVIs that can be configured on a switch is not limited by software. However, the interrelationship between these interfaces and other features configured on the switch may overload the CPU due to hardware limitations.

Describing Routed Ports on a Multilayer Switch

This topic describes a routed port on a multilayer switch.



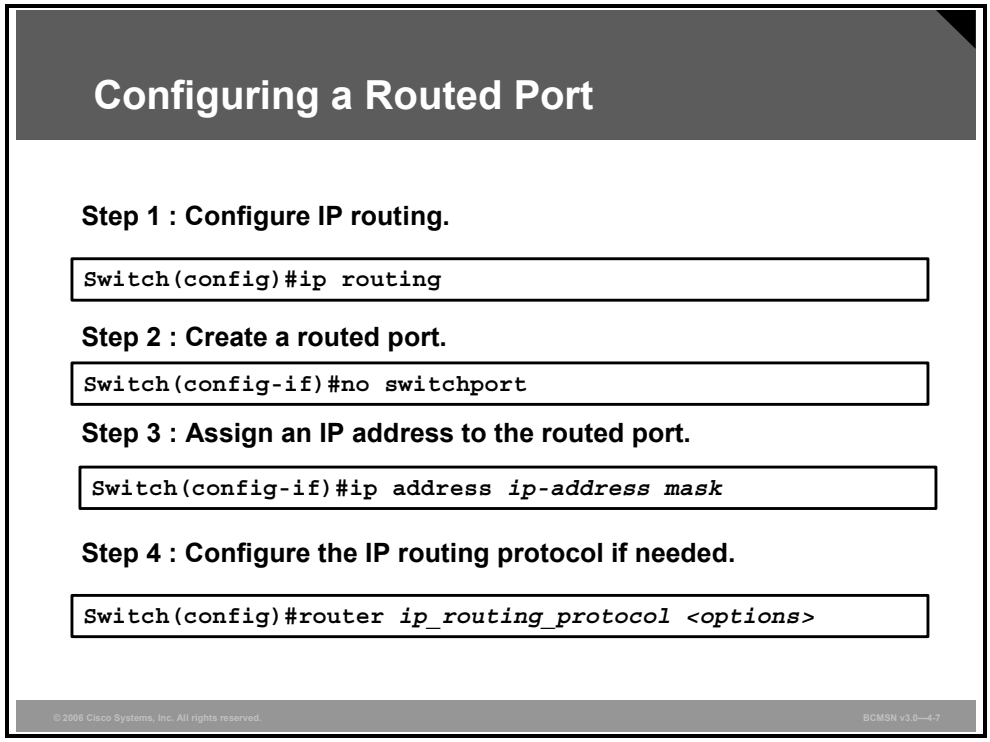
Routed switch ports are typically configured by removing the Layer 2 switch port capability of the switch port. On most switches, the ports are Layer 2 ports by default. On some switches, the ports are Layer 3 ports by default. The layer at which the port functions determines the commands that can be configured on the port.

Routed ports have these characteristics and functions:

- The port is a physical switch port with Layer 3 capability.
- The port is not associated with any VLAN.
- The port serves as the default gateway for devices out that switch port.
- Layer 2 port functionality must be removed before a routed port can be configured.

Configuring Routed Ports on a Multilayer Switch

This topic explains the procedure to configure routed ports on a multilayer switch.



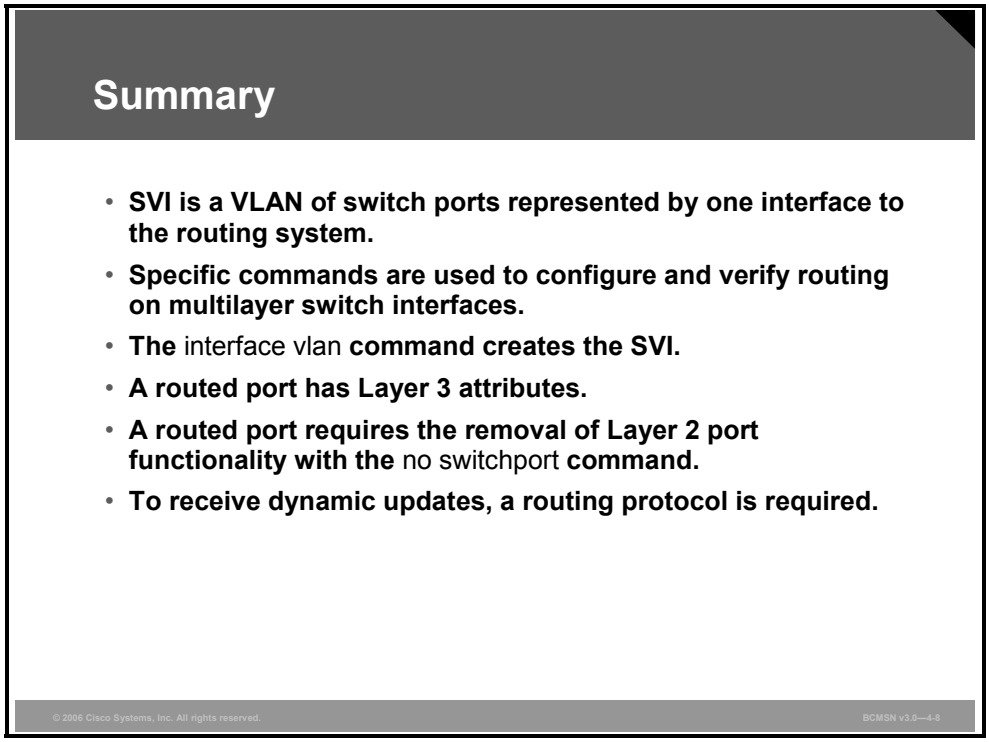
To configure a routed port, perform these steps.

Steps for Inter-VLAN Routing Configuration

Step	Description	Notes and Comments
1.	`Switch(config)#`**`ip routing`**	Enable IP routing on the router.
2.	`Switch(config)#`**`no switchport`**	Create the SVI interface or navigate to configuration mode for the interface.
3.	`Switch(config-if)#`**`ip address`** *`n.n.n.n subnet-mask`*	Assign an IP address to the SVI for the VLAN.
4.	(Optional) Specify an IP routing protocol. `Switch(config)#`**`router`** *`ip_routing_protocol <options>`*	This step is necessary for the switch to exchange dynamic routing updates with other routing devices. The routing protocol specified may require additional options. Refer to the documentation for the routing protocol for further details.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 3

Deploying CEF-Based Multilayer Switching

Overview

Layer 3 switching provides a wire-speed mechanism by which to route packets between VLANs using tables that store Layer 2 and Layer 3 forwarding information in hardware.

Cisco Express Forwarding (CEF) is the most efficient means of providing Layer 3 switching on a multilayer switch. CEF uses a very specific process to build forwarding tables in hardware and then uses that table information to forward packets at line speed.

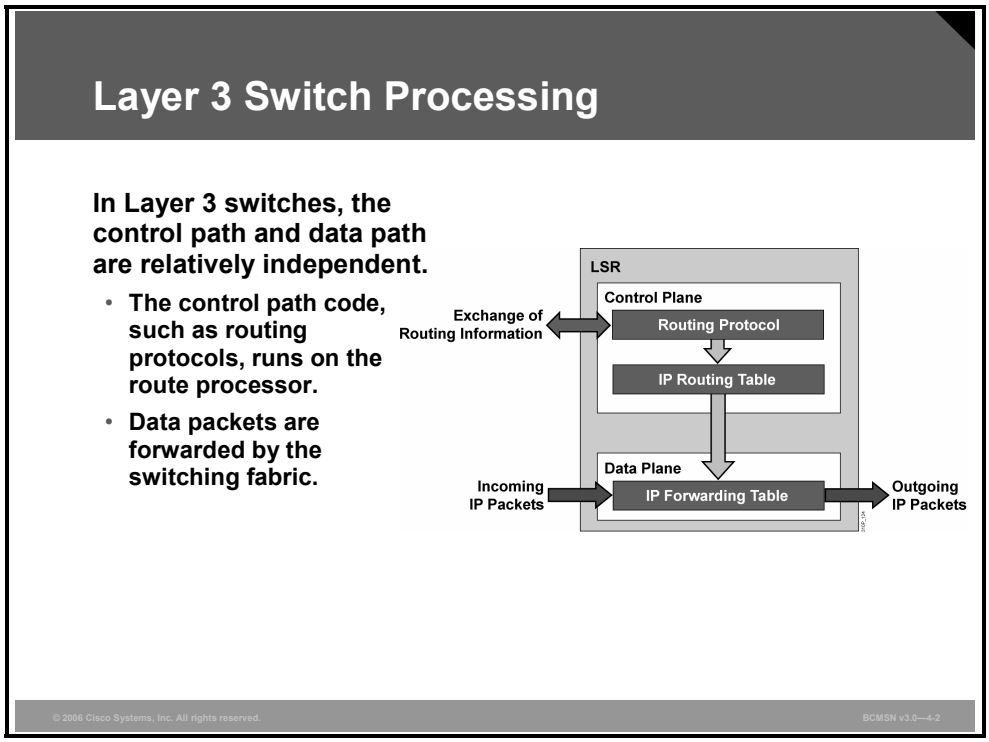
Objectives

Upon completing this lesson, you will be able to describe the operation of CEF in a multilayer switch environment. This ability includes being able to meet these objectives:

- Explain Layer 3 switch processing
- Explain a CEF-based multilayer switch
- Describe the process that a multilayer switch uses to forward packets
- Describe the commands used to configure CEF on Cisco Catalyst multilayer switches
- Explain the procedure to enable CEF-based MLS
- Describe common problems that can occur with CEF and solutions
- Describe the commands used to troubleshoot CEF on multilayer switches
- Explain the procedure to troubleshoot problems with CEF-based MLS

Explaining Layer 3 Switch Processing

This topic explains Layer 3 switching.



Layer 3 switching refers to a class of high-performance routers optimized for the campus LAN or intranet, providing both wire-speed Ethernet routing and switching services.

A Layer 3 switch router performs these three major functions:

- Packet switching
- Route processing
- Intelligent network services

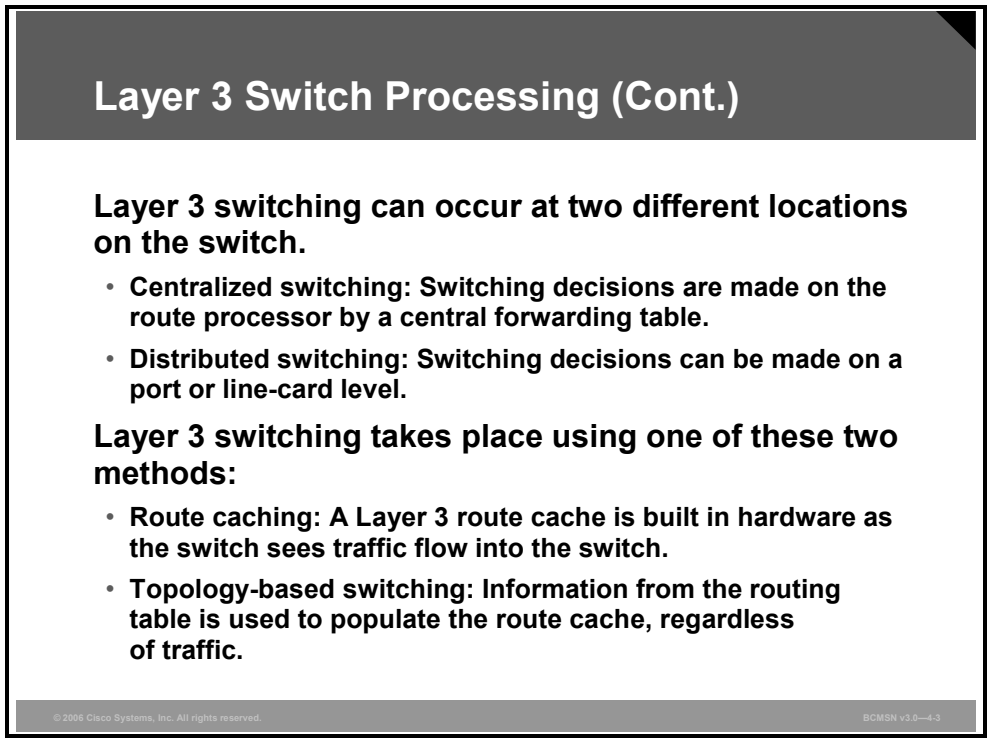
Compared to other routers, Layer 3 switch routers process more packets faster by using ASIC hardware instead of microprocessor-based engines. Layer 3 switch routers also improve network performance with two software functions: route processing and intelligent network services.

Distributed Hardware Forwarding

Layer 3 switching software employs a distributed architecture in which the control path and data path are relatively independent. The control path code, such as routing protocols, runs on the route processor, whereas most of the data packets are forwarded by the Ethernet interface module and the switching fabric.

Each interface module includes a microcoded processor that handles all packet forwarding. These are the main functions of the control layer between the routing protocol and the firmware datapath microcode:

- Managing the internal data and control circuits for the packet-forwarding and control functions
- Extracting the other routing and packet-forwarding-related control information from the Layer 2 and Layer 3 bridging and routing protocols and the configuration data, and then conveying the information to the interface module for control of the data path
- Collecting the data path information, such as traffic statistics, from the interface module to the route processor
- Handling certain data packets sent from the Ethernet interface modules to the route processor



Layer 3 switching can occur at two different locations on the switch.

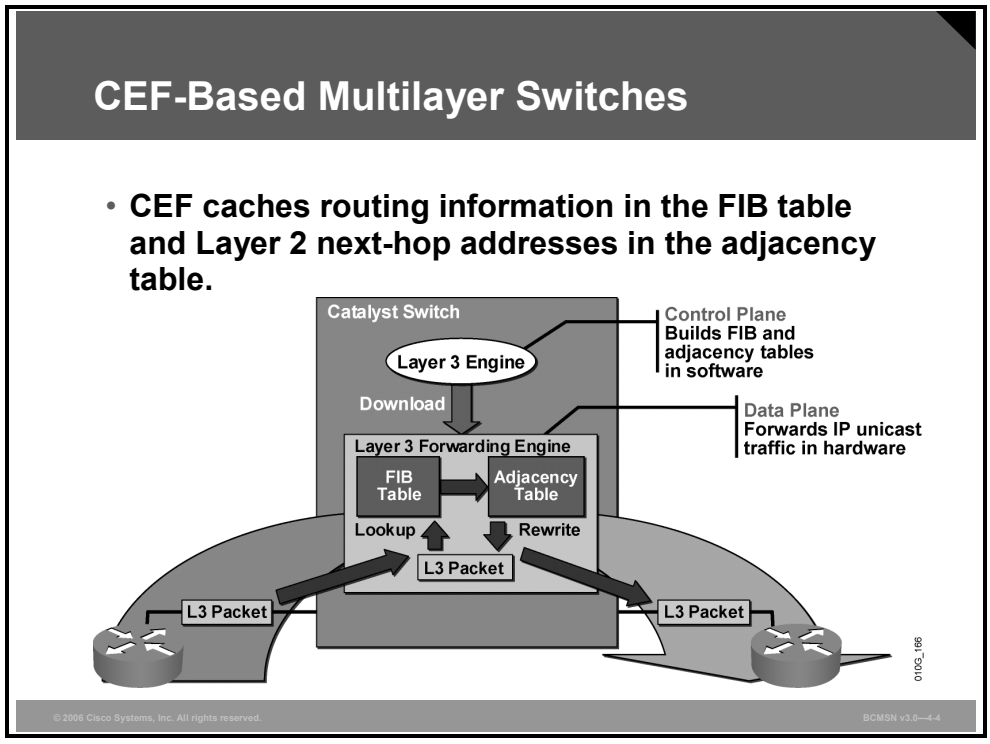
- **Centralized switching:** Switching decisions are made on the route processor by a central forwarding table, typically controlled by an ASIC.
- **Distributed switching:** Switching decisions can be made on a port or line-card level rather than on a central route processor. Cached tables are distributed and synchronized to various hardware components so that processing can be distributed throughout the switch chassis.

Layer 3 switching takes place using one of these two methods, which are platform dependent:

- **Route caching:** Also known as flow-based or demand-based switching, a Layer 3 route cache is built within hardware functions as the switch sees traffic flow into the switch.
- **Topology-based switching:** Information from the routing table is used to populate the route cache, regardless of traffic flow. The populated route cache is called the Forwarding Information Base (FIB). CEF is the facility that builds the FIB.

Explaining CEF-Based Multilayer Switches

This topic explains a CEF -based multilayer switch.



Cisco Systems Layer 3 devices can use a variety of methods to switch packets from one port to another. The most basic method of switching packets between interfaces is called process switching.

Process switching moves packets between interfaces, based on information in the routing table and the Address Resolution Protocol (ARP) cache, on a scheduled basis. As packets arrive, they will be moved into a queue to wait for further processing. When the scheduler runs, the outbound interface will be determined and the packet will be switched. Waiting for the scheduler introduces latency.

To speed the switching process, strategies exist to switch packets on demand as they arrive on an interface and to cache information that is needed to make packet-forwarding decisions.

CEF uses these strategies to expediently switch data packets to their destinations. It caches information generated by the Layer 3 routing engine. CEF caches routing information in one table (the FIB) and caches Layer 2 next-hop addresses for all FIB entries in an adjacency table. Because CEF maintains multiple tables for forwarding information, parallel paths can exist and enable CEF to load balance per packet.

CEF operates in one of two modes.

- **Central CEF mode:** The CEF FIB and adjacency tables reside on the route processor, and the route processor performs the express forwarding. Use this CEF mode when line cards are not available for CEF switching, or when features are not compatible with distributed CEF.

- **Distributed Cisco Express Forwarding (dCEF) mode:** dCEF is supported on only Cisco Catalyst 6500 switches. When dCEF is enabled, line cards maintain identical copies of the FIB and adjacency tables. The line cards can perform the express forwarding by themselves, relieving the main processor of involvement in the switching operation. dCEF uses an interprocess communications (IPC) mechanism to ensure synchronization of FIBs and adjacency tables on the route processor and line cards.

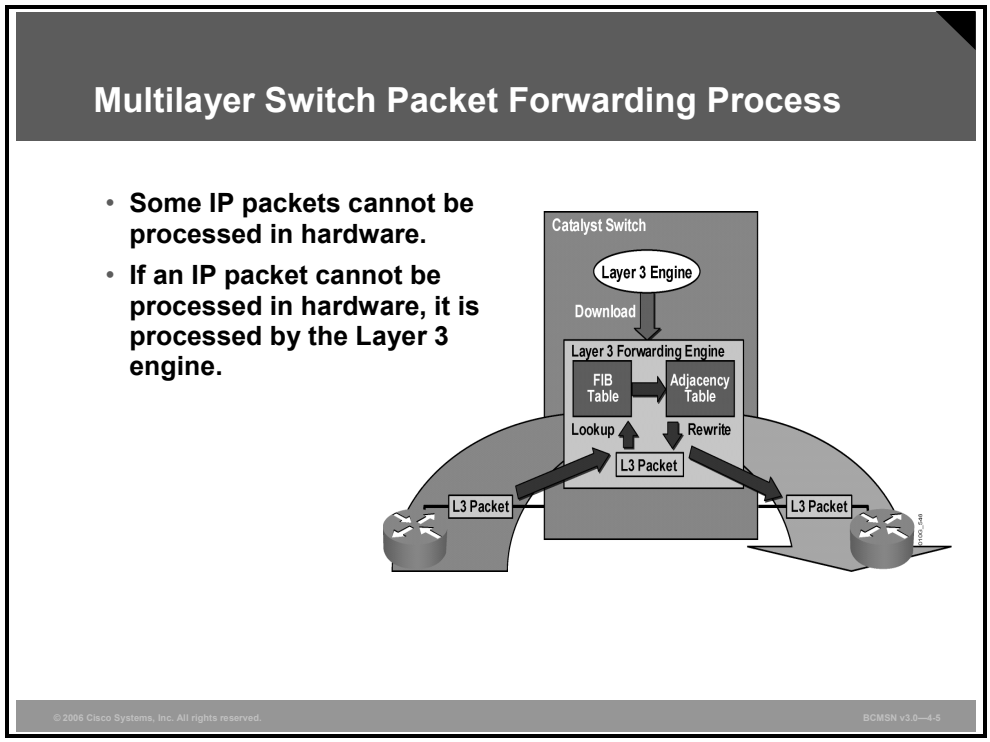
There is a wide range of CEF-based Cisco multilayer switches:

- Catalyst 2970
- Catalyst 3550
- Catalyst 3560
- Catalyst 3750
- Catalyst 4500
- Catalyst 4948
- Catalyst 6500

The Cisco Catalyst 6500 is a modular switch whereby the Multilayer Switch Feature Card (MSFC) is responsible for control-plane operations, and the supervisor Policy Feature Card (PFC) is responsible for the data-plane operations.

Identifying the Multilayer Switch Packet Forwarding Process

This topic describes the process used by a multilayer switch to forward packets.



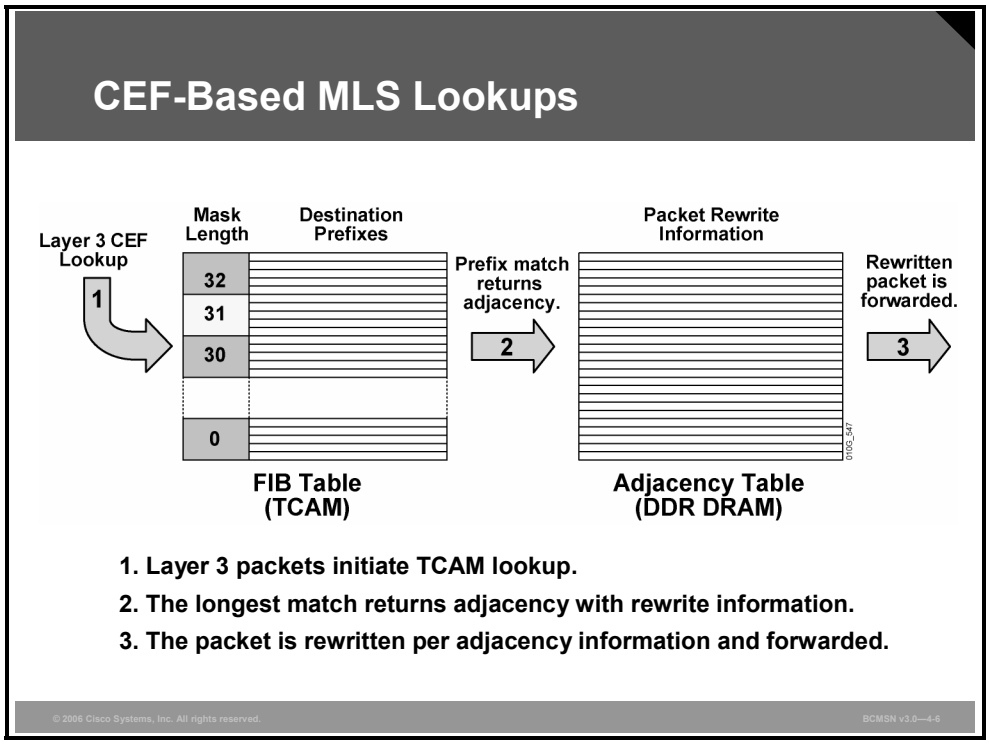
CEF separates the control plane hardware from the data plane hardware and switching. ASICs in switches are used to separate the control plane and data plane, thereby achieving higher data throughput. The control plane is responsible for building the FIB table and adjacency tables in software. The data plane is responsible for forwarding IP unicast traffic using hardware.

When traffic cannot be processed in hardware, it must receive processing in software by the Layer 3 engine. This traffic does not receive the benefit of expedited hardware-based forwarding. A number of different packet types may force the Layer 3 engine to process them. Some examples of IP exception packets are packets that:

- Use IP header options (Packets that use TCP header options are switched in hardware because they do not affect the forwarding decision.);
- Have an expiring IP Time to Live (TTL) counter;
- Are forwarded to a tunnel interface;
- Arrive with nonsupported encapsulation types;
- Are routed to an interface with nonsupported encapsulation types;
- Exceed the maximum transmission unit (MTU) of an output interface and must be fragmented.

CEF-Based Tables and MLS Lookups

This subtopic describes the CEF process as it relates to Multilayer Switching (MLS).



CEF-based tables are initially populated and used as follows:

- The FIB is derived from the IP routing table and is arranged for maximum lookup throughput.
- The adjacency table is derived from the ARP table, and it contains Layer 2 rewrite (MAC) information for the next hop.
- CEF IP destination prefixes are stored in the ternary content addressable memory (TCAM) table, from the most specific to the least specific entry.
- When the CEF TCAM table is full, a wildcard entry redirects frames to the Layer 3 engine.
- When the adjacency table is full, a CEF TCAM table entry points to the Layer 3 engine to redirect the adjacency.
- The FIB lookup is based on the Layer 3 destination address prefix (longest match).

FIB Table Updates

The FIB table is updated when these events occur:

- An ARP entry for the destination next hop changes, ages out, or is removed.
- The routing table entry for a prefix changes.
- The routing table entry for the next hop changes.

These are the basic steps for initially populating the adjacency table:

Step 1 The Layer 3 engine queries the switch for a physical MAC address.

Step 2 The switch selects a MAC address from the chassis MAC range and assigns it to the Layer 3 engine. This MAC address is assigned by the Layer 3 engine as a burned-in address for all VLANs and is used by the switch to initiate Layer 3 packet lookups.

Step 3 The switch installs wildcard CEF entries, which point to drop adjacencies (for handling CEF table lookup misses).

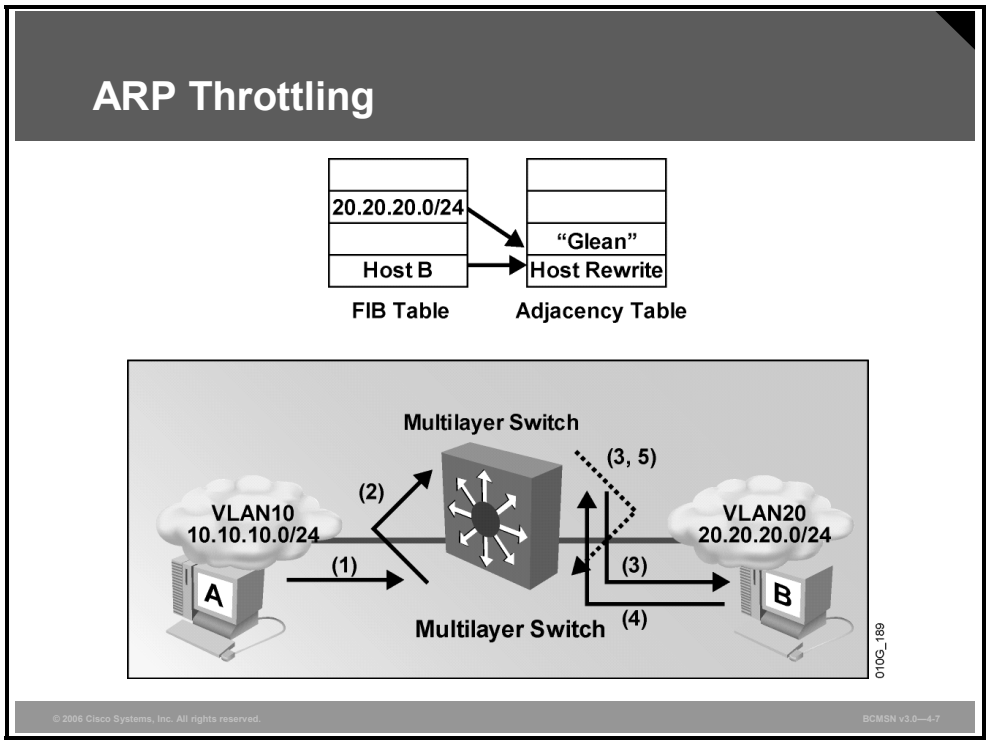
Step 4 The Layer 3 engine informs the switch of its interfaces participating in MLS (MAC address and associated VLAN). The switch creates the (MAC, VLAN) Layer 2 CAM entry for the Layer 3 engine.

Step 5 The Layer 3 engine informs the switch about features for interfaces participating in MLS.

Step 6 The Layer 3 engine informs the switch about all CEF entries related to its interfaces and connected networks. The switch populates the CEF entries and points them to Layer 3 engine redirect adjacencies.

ARP Throttling

This subtopic describes ARP throttling.



Only the first few packets for a connected destination reach the Layer 3 engine so that the Layer 3 engine can use ARP to locate the host. Throttling adjacency is installed so that subsequent packets to that host are dropped in hardware until an ARP response is received.

The throttling adjacency is removed when an ARP reply is received (and a complete rewrite adjacency is installed for the host). The switch removes throttling adjacency if no ARP reply is seen within 2 seconds, to allow more packets through to reinitiate ARP. This relieves the Layer 3 engine from excessive ARP processing or from ARP-based denial of service attacks.

The figure provides an example of ARP throttling, which consists of these steps:

Step 1 Host A sends a packet to host B.

Step 2 The switch forwards the packet to the Layer 3 engine based on the "glean" entry in the FIB. A glean adjacency entry indicates that a particular next hop should be directly connected, but there is no MAC header rewrite information available.

Step 3 The Layer 3 engine sends an ARP request for host B and installs the drop adjacency for host B.

Step 4 Host B responds to the ARP request.

The Layer 3 engine installs adjacency for host B and removes the drop adjacency. The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created (such as through the ARP protocol), a link-layer header for that adjacent node is precomputed and stored in the adjacency table.

After a route is determined, it points to a next hop and corresponding adjacency entry. The route is subsequently used for encapsulation during CEF switching of packets.

A route might have several paths to a destination prefix, as when a router is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.

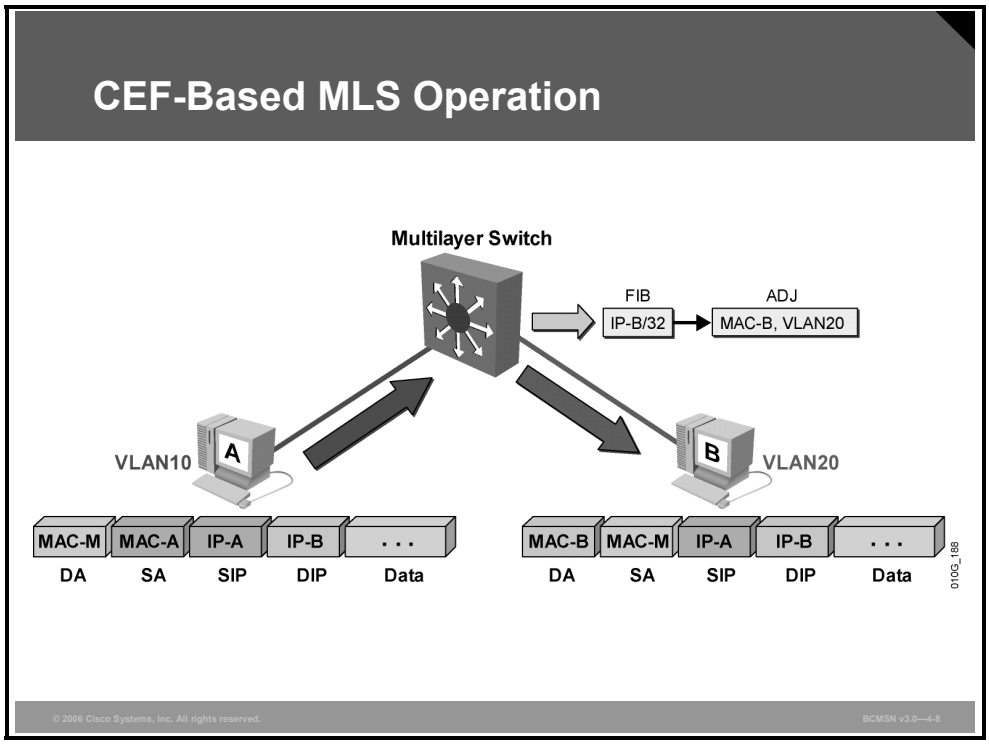
In addition to adjacencies associated with next-hop interfaces (host-route adjacencies), other types of adjacencies are used to expedite switching when certain exception conditions exist. When the prefix is defined, prefixes requiring exception processing are cached with one of these special adjacencies:

- **Null adjacency:** Packets destined for a "Null0" interface are dropped. This can be used as an effective form of access filtering.
- **Glean adjacency:** When a router is connected directly to several hosts, the FIB table on the router maintains a prefix for the subnet rather than for the individual host prefixes. The subnet prefix points to a glean adjacency. When packets need to be forwarded to a specific host, the adjacency database is gleaned for the specific prefix.
- **Punt adjacency:** Features that require special handling, or features that are not yet supported in conjunction with CEF switching paths, are forwarded to the next switching layer for handling. For example, the packet may require CPU processing. Features that are not supported are forwarded to the next-higher switching level.
- **Discard adjacency:** Packets are discarded.
- **Drop adjacency:** Packets are dropped, but the prefix is checked.

When a link-layer header is appended to packets, FIB requires the appended header to point to an adjacency corresponding to the next hop. If an adjacency was created by FIB and not discovered through a mechanism such as ARP, the Layer 2 addressing information is not known, and the adjacency is considered incomplete. After the Layer 2 information is known, the packet is forwarded to the route processor, and the adjacency is determined through ARP.

CEF-Based MLS Operation

This subtopic describes CEF-based Multilayer Switching (MLS) operation.



These are the steps that would occur when you use CEF to forward frames between host A and host B on different VLANs.

Step 1 Host A sends a packet to host B. The switch recognizes the frame as a Layer 3 packet because the destination MAC (MAC-M) matches the Layer 3 engine MAC.

Step 2 The switch performs a CEF lookup based on the destination IP address (IP-B). The packet hits the CEF entry for the connected (VLAN20) network and is redirected to the Layer 3 engine using a "glean" adjacency.

Step 3 The Layer 3 engine installs an ARP throttling adjacency in the switch for the host B IP address.

Step 4 The Layer 3 engine sends ARP requests for host B on VLAN20.

Step 5 Host B sends an ARP response to the Layer 3 engine.

Step 6 The Layer 3 engine installs the resolved adjacency in the switch (removing the ARP throttling adjacency).

Step 7 The switch forwards the packet to host B.

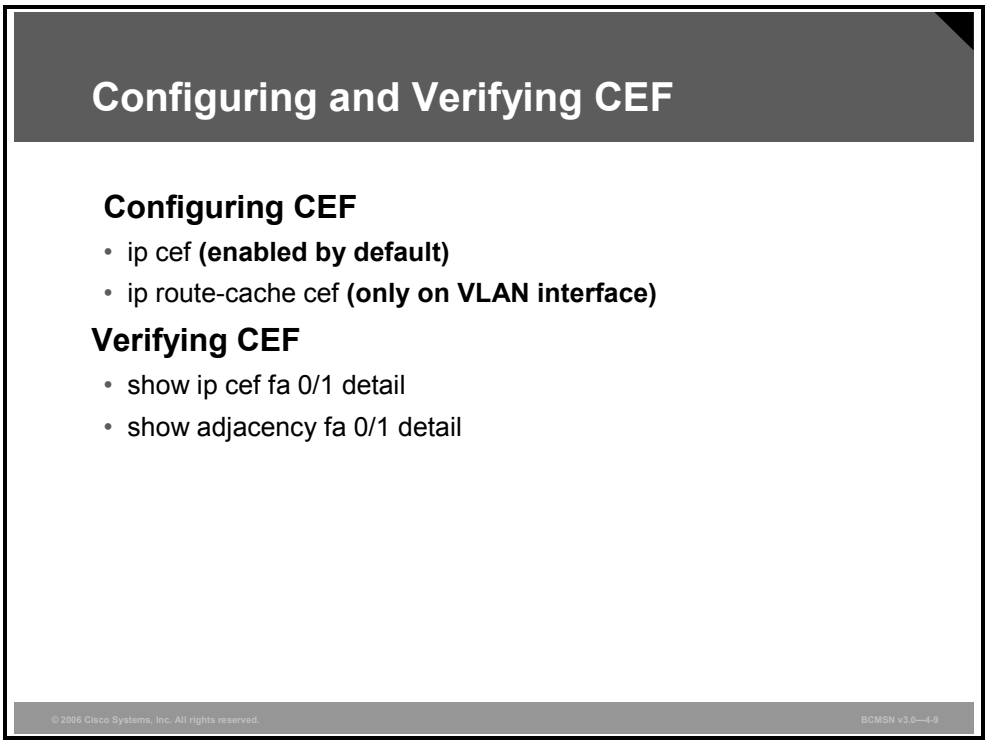
Step 8 The switch receives a subsequent packet for host B (IP-B).

Step 9 The switch performs a Layer 3 lookup and finds a CEF entry for host B. The entry points to the adjacency with rewrite information for host B.

The switch rewrites packets per the adjacency information and forwards the packet to host B on VLAN20.

Describing CEF Configuration Commands

This topic describes the commands used to configure CEF on Cisco Catalyst multilayer switches.



Use these commands to configure CEF when possible and verify its operation.

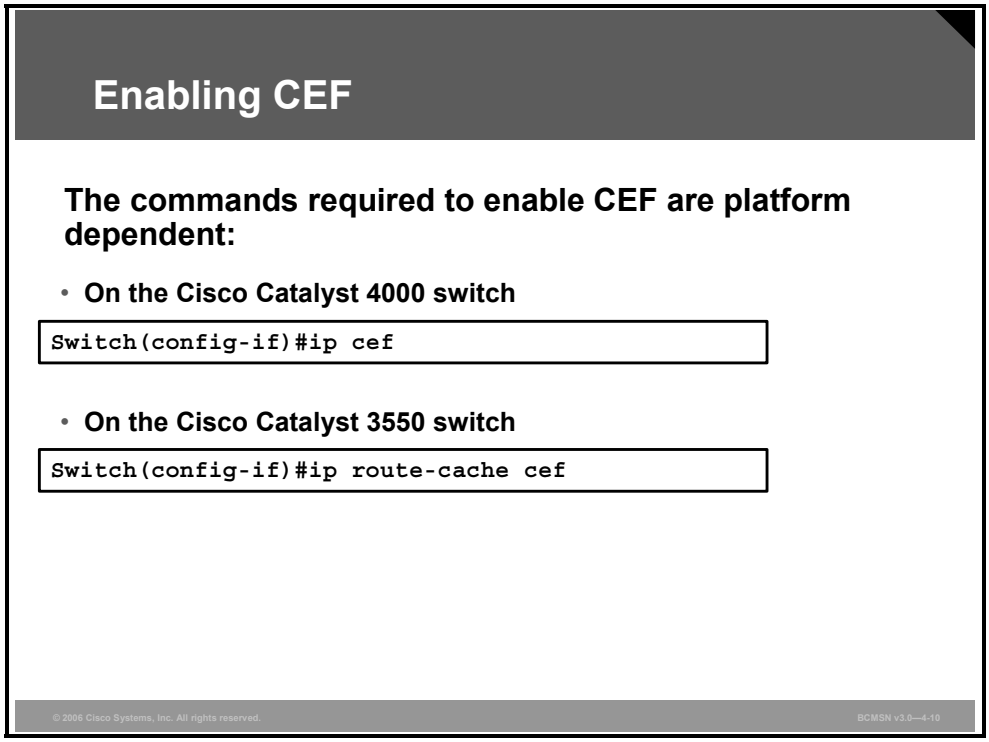
CEF Configuration Commands

The table describes CEF configuration commands.

Command	Description
`Switch(config-if)#` **`ip cef`**	On a Cisco Catalyst 4000 Series switch, enables CEF if it has been previously disabled. CEF is on by default.
`Switch(config-if)#` **`no ip cef`**	Disables CEF on a Cisco Catalyst 4000 Series switch.
`Switch(config-if)#` **`ip route-cache cef`**	On a Cisco Catalyst 3550 Series switch, enables CEF if it has been previously disabled on an interface. CEF is on by default.
`Switch(config-if)#` **`no ip route-cache cef`**	Disables CEF on a Cisco Catalyst 3550 Series switch.
`Switch#` **`show ip cef`** [*type mod/port*] [**detail**]	Verifies CEF operation.
`Switch#` **`show interface`** *type mod/port* | **`begin L3`**	Displays information about Layer 3 switched traffic for the interface.
`Switch#` **`show interface`** *type mod/port* | **`include switched`**	Displays counts on packets switched at Layer 2 and Layer 3.

Enabling CEF-Based MLS

This topic explains the procedure to enable CEF-based MLS.



Hardware Layer 3 switching is permanently enabled on Cisco Catalyst 6500 Series Supervisor Engine 720 with Policy Feature Card 2 (PFC2) or PFC3, Multilayer Switch Feature Card 3 (MSFC3), and Distributed Forwarding Card (DFC). No configuration is required, and CEF cannot be disabled.

To disable CEF, the **no ip cef** command can be used on the Cisco Catalyst 4000, or the **no ip route-cache cef** command can be used on a Cisco Catalyst 3550 interface.

If CEF is enabled globally, it is automatically enabled on all interfaces, provided that IP routing is enabled on the device. It can then be enabled or disabled on an interface basis. Cisco recommends that CEF be enabled on all Layer 3 interfaces. If CEF is disabled on an interface, you can enable CEF as follows:

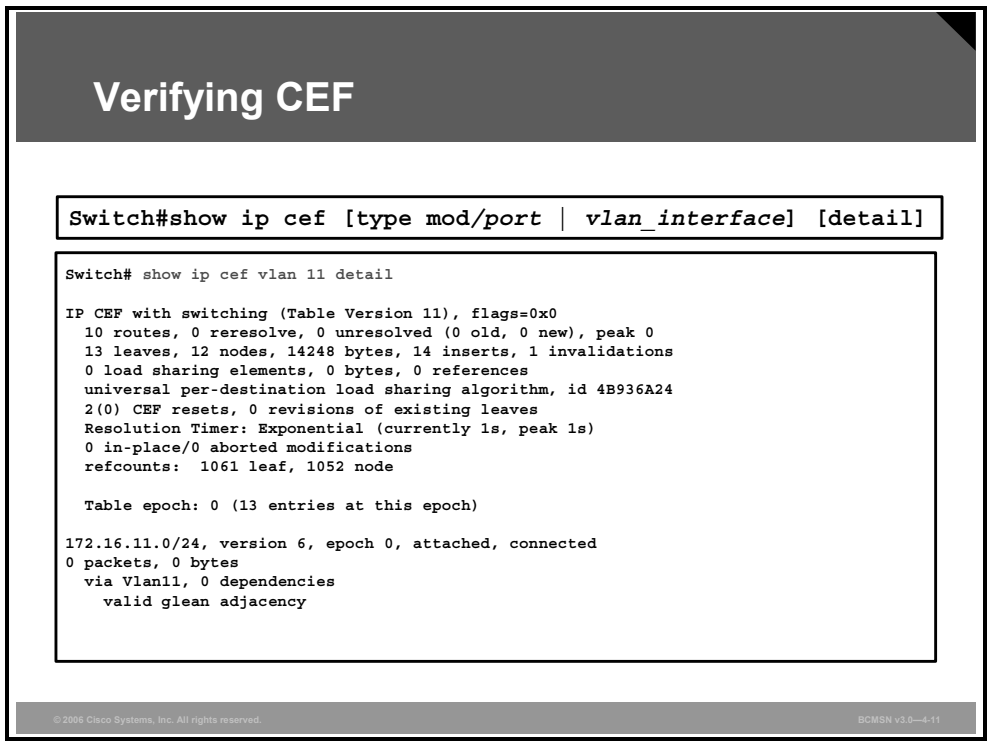
- On the Cisco Catalyst 3550 switch, use the **ip route-cache cef** interface configuration command to enable CEF on an interface.
- On the Cisco Catalyst 4000 switch, use the **ip cef** interface configuration command to enable CEF on an interface after it has been disabled.

Per-destination load balancing allows the router to use multiple paths to achieve load sharing. Packets for a given source-destination host pair are guaranteed to take the same path, even if multiple paths are available. This ensures that packets for a given host pair arrive in order. Per-destination load balancing is enabled by default when you enable CEF, and it is the load balancing method of choice for most situations.

Because per-destination load balancing depends on the statistical distribution of traffic, load sharing becomes more effective as the number of source-destination pairs increases.

Verifying CEF

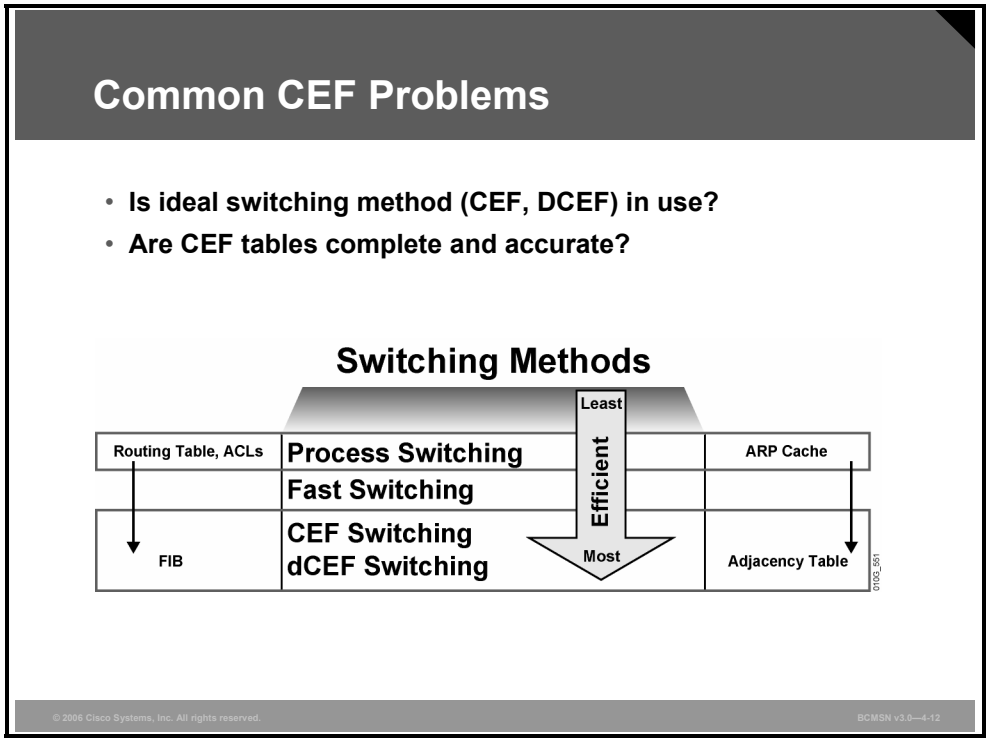
This subtopic lists commands used to verify the operation of CEF.



The **show ip cef** command displays entries in the FIB.

Describing Common CEF Problems and Solutions

This topic describes common problems that can occur with CEF and solutions to those problems.



CEF is the fastest means of switching Layer 3 packets in hardware. The CEF tables stored in hardware are populated from information that is gathered by the route processor. Troubleshooting CEF operations therefore has two primary steps.

- Ensure that the normal Layer 3 operations on the route processor are functioning properly so that the switch tables will be populated with accurate and complete information.
- Verify that information from the route processor has properly populated the FIB and adjacency table, and is being used by CEF to switch Layer 3 packets in hardware.

Troubleshooting CEF is, in essence, verifying that packets are indeed receiving the full benefit of CEF switching and not being “punted” to a slower packet-switching or processing method. The Cisco term "punt" describes the action of sending a packet "down" to the next-fastest switching level. This list defines the order of preferred Cisco IOS switching methods, from fastest to slowest.

1. Distributed CEF
2. CEF
3. Fast switching
4. Process switching

A punt occurs when the preferred switching method did not produce a valid path or, in CEF, did not produce a valid adjacency. If the CEF lookup process fails to find a valid entry in the FIB, CEF will install a punt adjacency to the less-preferred system. CEF will punt all packets with that adjacency to the next-best switching mode to forward all the packets by some means, even if that means is less efficient.

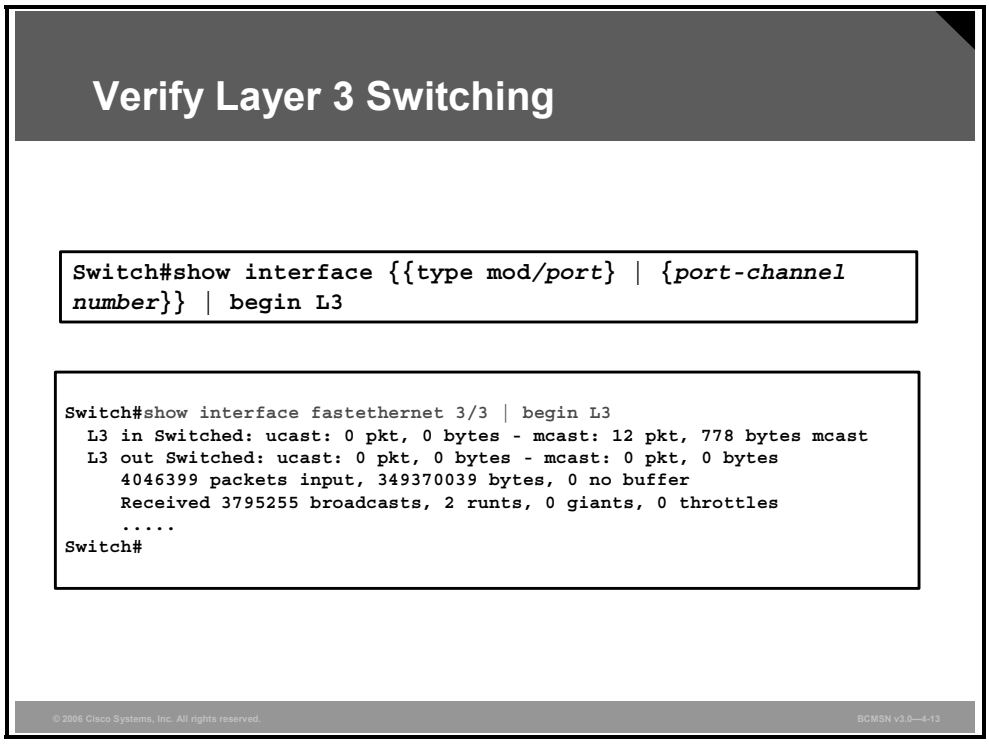
CEF Problems and Solutions

The table describes some basic CEF problems and associated solutions.

	Problem	Solution
1.	CEF or dCEF has been disabled on an interface or line card.	Verify that IP routing is enabled; disabling IP routing will disable CEF globally. Determine if lack of memory resource is disabling CEF or dCEF. This is possible if an inadequate amount of memory is available to store a large number of routing table entries in the FIB. Review the switch configuration for features that may not be compatible with CEF. Specifically look for commands and features related to switching services. MALLOCFAIL and FIBDISABLE messages will indicate if CEF is not functional.
2.	Routes are not appearing in routing table as designed.	Troubleshoot routing protocols to ensure that routing table is being properly populated.
3.	ARP is not resolving MAC addresses.	Verify that the IP host exists on a media that is accessible to the host that is trying to ARP resolve. Use a packet analyzer to determine if ARP is not functioning or if the sending host is not caching the ARP resolutions. Check that subsystems of ARP are functioning correctly: IP addresses may be invalid, local addresses may resolve, but remote addresses may not resolve to a default gateway, and so forth.
4.	Existing routing table entries do not appear in the CEF FIB table, or next hop is not appropriate.	This is a Cisco IOS software CEF issue.
5.	Resolved ARP addresses do not appear in the adjacency table.	This is a Cisco IOS software CEF issue.
6.	Part of an adjacency is an “incomplete” or “drop” adjacency.	Check that unsupported software features have not been enabled.
7.	CEF operations are suspect or their existence needs to be verified.	Debug CEF to observe all messages regarding CEF operations. Messages are indicative of successful or failed CEF operations.

Describing CEF Troubleshooting Commands

This topic describes the commands used to troubleshoot CEF on multilayer switches.



The commands available to troubleshoot CEF are platform dependent.

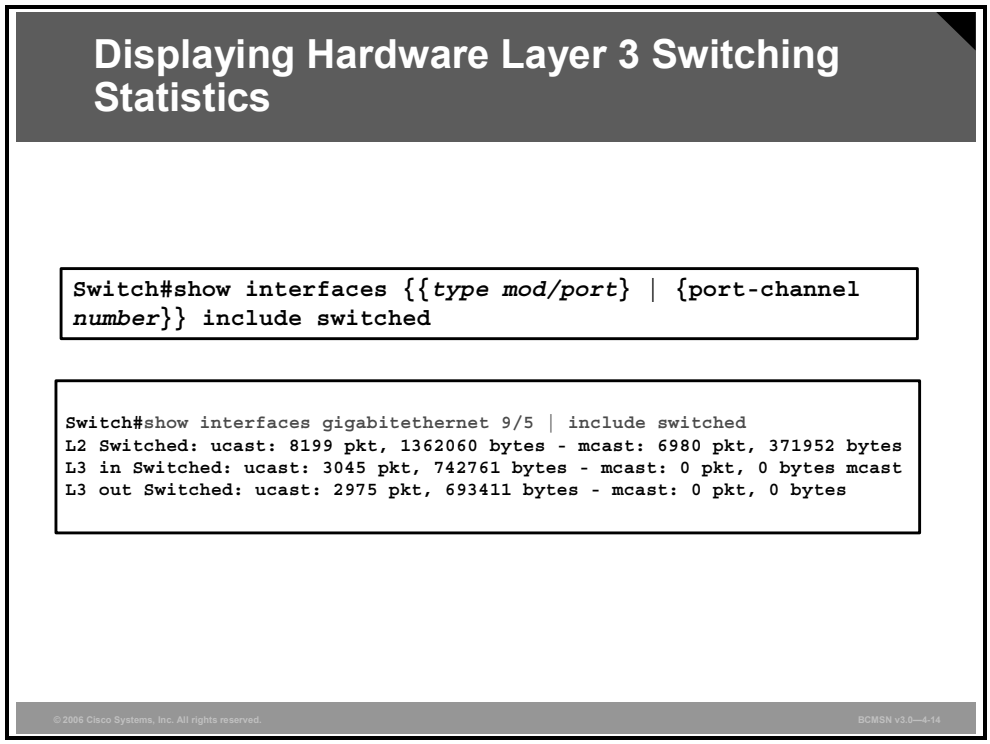
These commands can be used to troubleshoot CEF on the Cisco Catalyst 4500 series switch.

Troubleshooting Commands

Command	Description
`Switch(config)# ip cef`	Enables standard CEF operation.
`Switch (config)# [no] ip cef load-sharing algorithm [include-ports source destination]`	Enables load sharing hash function to use source and destination ports. Use the **no** keyword to set the switch to use the default Cisco IOS load-sharing algorithm.
`Switch# show ip cef`	Displays the collected CEF information.
`Switch# show interface type slot/interface | begin L3`	Displays a summary of IP unicast traffic.
`Switch# show interface type number counters detail`	Displays IP statistics.
`Switch# show adjacency [interface] [detail | internal | summary]`	Displays detailed adjacency information, including Layer 2 information, when the optional detail keyword is used.

Display CEF Statistics

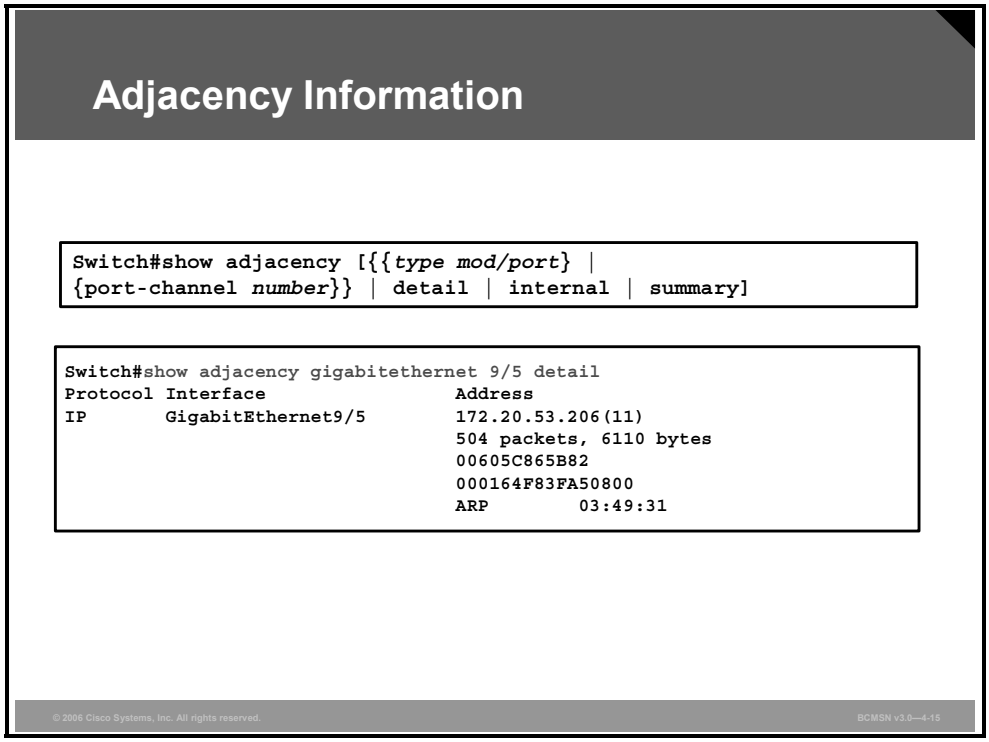
This subtopic discusses the **show interface** command with the | *begin L3* argument to verify that Layer 3 traffic is being switched, thereby utilizing CEF.



Use the **show interfaces** command with the | *include switch* argument to show switching statistics at each layer for the interface. Verify that Layer 3 packets are being switched.

Displaying Detailed Adjacency Information

Here is the command used to display detailed information about the adjacency table.



Each time an adjacency entry is created, a Layer 2 data link layer header for that adjacent node is precomputed and stored in the adjacency table. This information is subsequently used for encapsulation during CEF switching of packets.

Output from the command **show adjacency detail** displays the content of the information to be used during this Layer 2 encapsulation. Verify that the header information is displayed as would be expected during Layer 2 operations, not using precomputed encapsulation from the adjacency table. Adjacency statistics are updated approximately every 60 seconds.

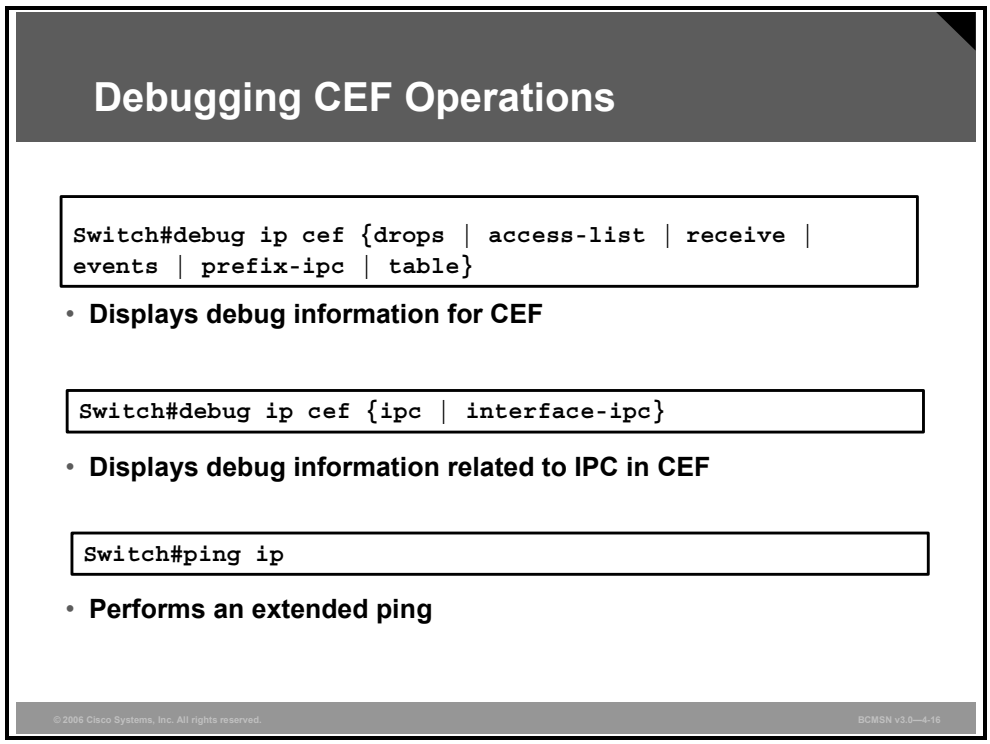
Also, the **show cef drops** command will display an indication of packets that are being dropped due to adjacencies that are either incomplete or nonexistent. There are two known reasons for incomplete or nonexistent adjacencies.

- The router cannot use ARP successfully for the next-hop interface.
- After a **clear ip arp** or a **clear adjacency** command, the router marks the adjacency as incomplete, and then it fails to clear the entry.

The symptoms of an incomplete adjacency include random packet drops during a ping test. Use the **debug ip cef** command to view CEF drops caused by an incomplete adjacency.

Debugging CEF Operations

The debug facility can be used to display detailed information on CEF operations.



Use the **debug ip cef** arguments to limit the debug output, thereby reducing the overhead of the debug command and providing focus on a specific CEF operation.

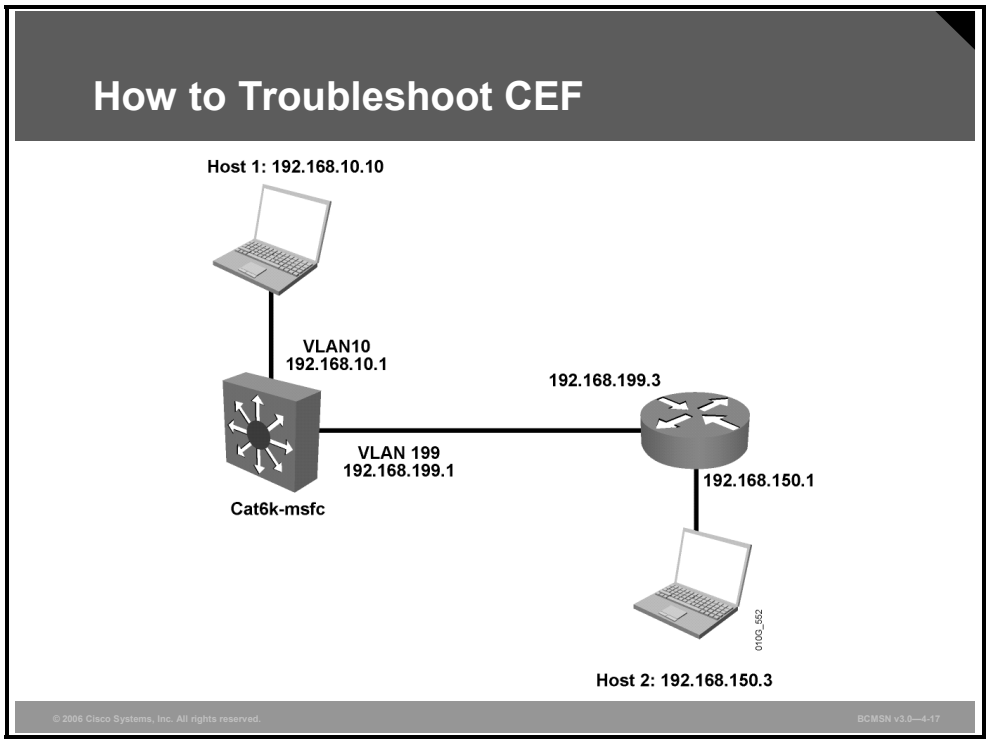
debug ip cef {drops [*access-list*] | receive [*access-list*] | events [*access-list*] | prefix-ipc [*access-list*] | table [*access-list*]}

Adding an argument to the **debug** command limits the debug output as follows:

- ***drops*:** Records dropped packets
- ***access-list* (optional):** Controls collection of debugging information from specified lists
- ***receive*:** Records packets that are not switched using information from the FIB table but that are received and sent to the next switching layer
- ***events*:** Records general CEF events
- ***prefix-ipc*:** Records updates related to IP prefix information, including these updates:
  - Debugging of IP routing updates in a line card
  - Reloading of a line card with a new table
  - Notification that adding a route update from the route processor to the line card exceeds the maximum number of routes
  - Control messages related to FIB table prefixes
- ***table*:** Produces a table showing events related to the FIB table. Possible types of events include these types:
  - Routing updates that populate the FIB table
  - Flushing of the FIB table
  - Adding or removing of entries to the FIB table
  - Table reloading process

Troubleshooting Layer 3 CEF-Based MLS

This topic describes the procedure to troubleshoot problems with CEF-based MLS.



The CEF tables stored in hardware are populated from information that is gathered by the route processor. To properly troubleshoot CEF operations, first ensure that the normal Layer 3 operations on the route processor are functioning properly so that the CEF tables will be populated with accurate and complete information. Next, verify that information from the route processor has properly populated the FIB and adjacency table used by CEF to perform Layer 3 switching of packets.

The steps that follow will verify if packet transfer between these hosts is occurring using CEF:

- Host 1 in VLAN10 with an IP address of 192.168.10.10
- Host 2 in VLAN150 with an IP address of 192.168.150.3

Step 1 Verify CEF.

Verify that CEF is operational at global or interface level using these commands:

```
show ip cef summary
show ip cef vlan 10
```

Note	On most Cisco Catalyst platforms, CEF cannot be turned off. If CEF is not operational, it is likely that the Cisco Catalyst has disabled the feature. This may be due to a software, feature, or hardware incompatibility or due to inadequate memory to support a large FIB and adjacency table.

Step 2 Verify the configuration.

If CEF is not operational, display the running configuration to determine if any switching functions have been configured that might disable CEF operations.

If CEF is operational, display the running configuration to verify the IP configuration of the Layer 3 interfaces used for the hosts to communicate. The IP addresses should be appropriate for the subnet, and the interfaces should not be shut down. This is a sample of the configuration output expected for the VLANs associated with the host communication. On this router, VLAN199 is the transit path that will be traversed to arrive at subnet 192.168.150.0:

```
Switch# show running-config
interface VLAN 10
  description Source VLAN
  ip address 192.168.10.1 255.255.255.0
!
interface VLAN 199
  description Transit VLAN
  ip address 192.168.199.1 255.255.255.0
```

Step 3 Verify population of the routing table on the route processor.

The routing protocols and route processor must populate the routing table accurately before those routing table entries can be of use as they are transferred to the FIB to facilitate Layer 3 switching. Verify the routing table by referring to a network diagram, knowing what routes should appear in the routing table, and then execute the **show ip route** command. In the case of troubleshooting connectivity to the specific network of the destination host (192.168.150.3/24), use this command:

```
Switch# show ip route | include 192.168.150.0
O 192.168.150.0/24  [110/2] via 192.168.199.3, 00:13:00, VLAN 199
```

Step 4 The network is accessible via next-hop address 192.168.199.3; therefore, the ARP entry by which to access 192.168.150.3 should be the MAC address resolved for 192.168.199.3.

Verify an ARP entry on the route processor.

Verify that there is an ARP entry for the next-hop IP address before checking if that entry is represented in the adjacency table.

```
Switch# show ip arp 192.168.199.3
Protocol   Address         Age   Hardware Addr   Type   Interface
Internet   192.168.199.3   176   0030.7150.6800  ARPA   VLAN 199
```

Step 5 Verify the CEF FIB table entry for the route.

Step 3 verified that a route to network 192.168.150.0 existed in the routing table. Now, verify that a CEF FIB entry exists to that same destination to ensure that packets will be CEF switched using the FIB rather than process switched using the routing table.

```
Switch# show ip cef 192.168.150.0
192.168.150.0/24, version 298, cached adjacency 192.168.199.3
0 packets, 0 bytes
via 192.168.199.3, VLAN 199, 0 dependencies
next-hop 192.168.199.3, VLAN 199
valid cached adjacency
```

This output verifies that there is a valid CEF entry for the destination network; packets can be CEF switched to the destination host.

Step 6 Verify an adjacency table entry for the destination.

Now, verify that the FIB entry shown in Step 5 has an associated adjacency table entry by using this command:

```
Switch# show adjacency detail | begin 192.168.199.3
IP VLAN 199 192.168.199.3(7)
0 packets, 0 bytes
003071506800
.....
...
.
```

The preceding output indicates that there is an adjacency for the next-hop IP address. The destination MAC address (003071506800) is the MAC address in the ARP table, as displayed in Step 4.

The counters (0 packets, 0 bytes) are almost always 0 because packets are switched in hardware; therefore, they never reach the route processor, which is required to increment counters.

Step 7 Verify CEF from the supervisor engine.

The CEF FIB and adjacency table entries shown in the example can also be verified from the supervisor engine on modular switch platforms, such as the 6500 series switches. This step is not necessary on fixed configuration switches, such as the 3550.

To display an FIB entry for the specific network from the supervisor engine:

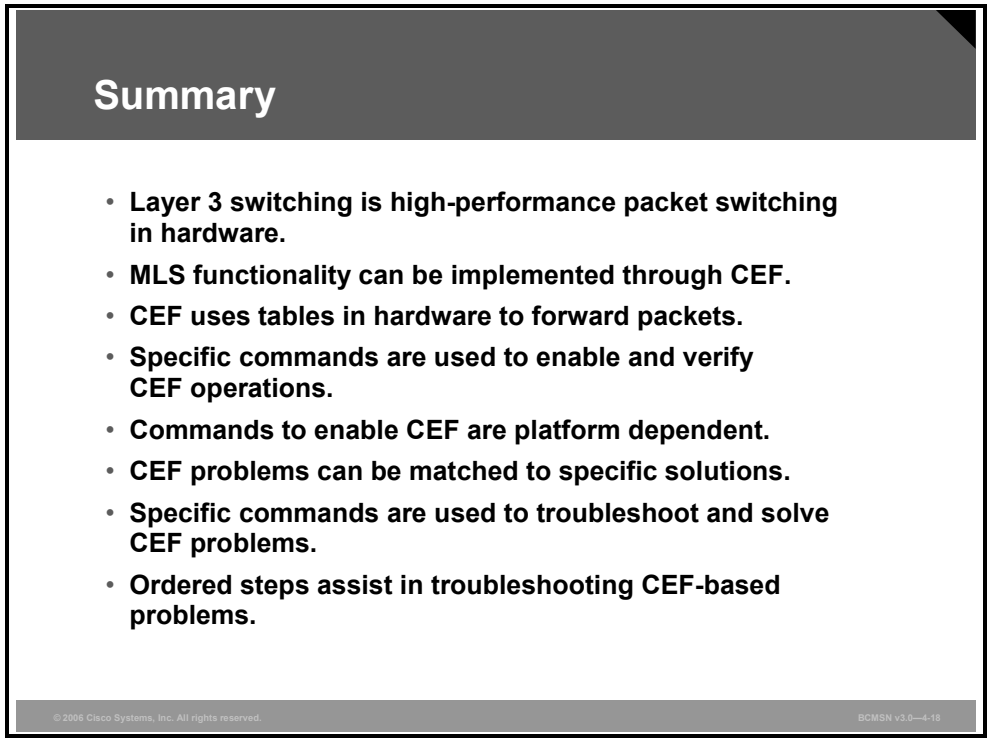
```
Console> (enable) show mls entry cef ip 192.168.150.0/24
Mod FIB-Type  Destination-IP   Destination-Mask  NextHop-IP    Weight
--- --------- --------------  ----------------  ------------  ---
15  resolved  192.168.150.0   255.255.255.255   192.168.199.3 1
```

To display an FIB entry for the specific network from the supervisor engine:

```
Console> (enable) show mls entry cef ip 192.168.150.0/24 adjacency
Mod:15
Destination-IP : 192.168.199.3 Destination-Mask : 255.255.255.255
FIB-Type : resolved

AdjType NextHop-IP    NextHop-Mac       VLAN Encp TX-Packets
------- ----------    ----------------- ---- ---- ------------
connect 192.168.199.3 00-30-71-50-68-00 199  ARPA 0
```

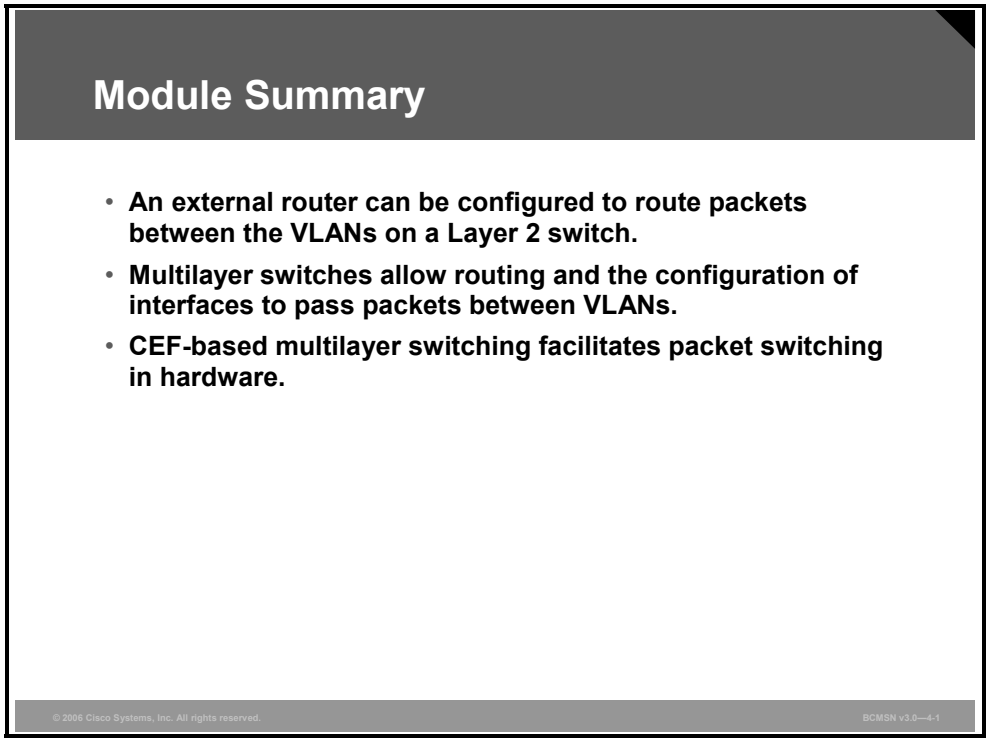

Summary

This topic summarizes the key points discussed in this lesson.



Module Summary

This topic summarizes the key points discussed in this module.



The configuration of multiple Layer 2 VLANs requires that Layer 3 routing occur between those VLANs. This inter-VLAN routing can be provided external to a Layer 2 switch or within a multilayer switch through the configuration of switch virtual interfaces (SVIs) and IP routing.

When routing occurs within a Cisco Catalyst multilayer switch, Cisco Express Forwarding (CEF) is deployed to facilitate Layer 3 switching through hardware-based tables, providing an optimal packet-forwarding process.

References

For additional information, refer to these resources:

- Cisco Systems, Inc., *How to Choose the Best Router Switching Path for Your Network:* http://www.cisco.com/en/US/partner/tech/tk827/tk831/technologies_white_paper09186a00800a62d9.shtml
- Cisco Systems, Inc., *Cisco Express Forwarding:* http://www.cisco.com/en/US/partner/tech/tk827/tk831/tk102/tsd_technology_support_sub-protocol_home.html

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) To perform inter-VLAN routing on a single connection between a switch and an external router, which two settings must be configured on the router? (Choose two.) (Source: Describing Routing Between VLANs)

A) subinterface
B) unique MAC addresses for each participating interface
C) a trunking encapsulation
D) STP

Q2) Which command is used to enable inter-VLAN routing for VLAN10 on an external router? (Source: Describing Routing Between VLANs)

A) Router(config-if)#**encapsulation dot1q 10**
B) Router(config-subif)#**encapsulation 10 dot1q**
C) Router(config-subif)#**encapsulation dot1q 10**
D) Router(config-if)#**switchport encapsulation dot1Q 10**

Q3) On a multilayer switch, which command must be used to change a switch port into a Layer 3-capable interface? (Source: Enabling Routing Between VLANs on a Multilayer Switch)

A) **IP address**
B) **IP routing**
C) **Layer 3 enable**
D) **no switchport**

Q4) How is an SVI created on a multilayer switch? (Source: Enabling Routing Between VLANs on a Multilayer Switch)

A) Switch(config)**#create svi vlan10**
B) Switch(config-if)**#interface vlan 10**
C) Switch(config-if)**#no switchport**
D) Switch(config-if)**#svi enable**

Q5) Which technology is required to forward packets between VLANs at wire speed? (Source: Deploying CEF-Based Multilayer Switching)

A) fast switching
B) CEF-based multilayer switching
C) an external router
D) STP

Q6) When you are employing CEF, where is the FIB derived from? (Source: Deploying CEF-Based Multilayer Switching)

A) the CAM
B) the TCAM
C) the IP routing table
D) the traffic that flows into the switch

Module Self-Check Answer Key

Q1) A, C

Q2) C

Q3) D

Q4) B

Q5) B

Q6) C

