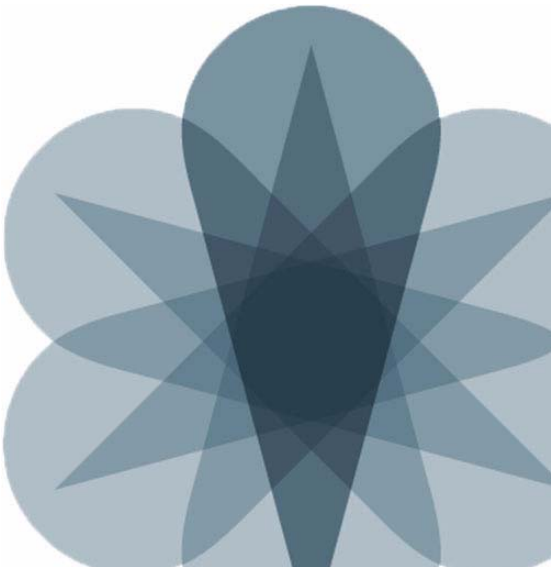
JNCIS-SP Study Guide—Part 2

Study Guide

Worldwide Education Services

1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net



JNCIS-SP Study GuideñPart 2

Printed in USA.

The information in this document is current as of the date listed above.

The information in this document has been carefully verified and is believed to be accurate for software Release 10.3R1.9. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental, or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

This study guide is aligned with the related Juniper Networks Certification track and is an approved resource for your exam preparation. Please go to the Juniper Networks Certification Program webpage (www.juniper.net/certification) to:

- **Learn more about the Juniper Networks certification program**
- **See the certification tracks**
- **Get detailed exam descriptions**
- **Find additional exam preparation materials**
- **Register for an exam**

Contents

Overview

Welcome to the JNCIS-SP Study Guide—Part 2. The purpose of this guide is to help you prepare for your JN0-360 exam and achieve your JNCIS-SP credential. The contents of this document are based on the *Junos Service Provider Switching* (JSPX) course. This study guide provides students with intermediate switching knowledge and configuration examples. The content includes an overview of switching concepts such as LANs, Layer 2 address learning, bridging, virtual LANs (VLANs), provider bridging, VLAN translation, spanning-tree protocols, and Ethernet Operation, Administration, and Maintenance (OAM). This guide also covers Junos operating system-specific implementations of integrated routing and bridging (IRB) interfaces, routing instances, virtual switches, load balancing, and port mirroring. This guide also covers the basics of Multiple VLAN Registration Protocol (MVRP), link aggregation groups (LAG), and multichassis LAG (MC-LAG). This content is based on the Junos OS Release 10.3R1.9.

Document Conventions

CLI and GUI Text

Frequently throughout this study guide, we refer to text that appears in a command-line interface (CLI) or a graphical user interface (GUI). To make the language of these documents easier to read, we distinguish GUI and CLI text from chapter text according to the following table.

Style	Description	Usage Example
Franklin Gothic	Normal text.	Most of what you read in the Lab Guide and Study Guide.

`Courier New`	Console text:
	• Screen captures
	• Noncommand-related syntax
	GUI text elements:
	• Menu names
	• Text field entry
		`commit complete`
		`Exiting configuration mode`
		Select `File > Open`, and then click `Configuration.conf` in the `Filename` text box.

Input Text Versus Output Text

You will also frequently see cases where you must enter input text yourself. Often these instances will be shown in the context of where you must enter them. We use bold style to distinguish text that is input versus text that is simply displayed.

Style	Description	Usage Example
`Normal CLI` `Normal GUI`	No distinguishing variant.	`Physical interface:fxp0, Enabled` View configuration history by clicking `Configuration > History`.
`CLI Input` **`GUI Input`**	Text that you must enter.	`lab@San_Jose>` **`show route`** Select `File > Save`, and type **`config.ini`** in the `Filename` field.

Defined and Undefined Syntax Variables

Finally, this study guide distinguishes between regular text and syntax variables, and it also distinguishes between syntax variables where the value is already assigned (defined variables) and syntax variables where you must assign the value (undefined variables). Note that these styles can be combined with the input style as well.

Style	Description	Usage Example
`CLI Variable` *`GUI Variable`*	Text where variable value is already assigned.	`policy` *`my-peers`* Click *`my-peers`* in the dialog.
`CLI Undefined` *`GUI Undefined`*	Text where the variable's value is the user's discretion and text where the variable's value as shown in the lab guide might differ from the value the user must input.	Type **`set policy`** ***`policy-name`***. **`ping 10.0.`*`x.y`* Select `File > Save`, and type ***`filename`*** in the `Filename` field.

Additional Information

Education Services Offerings

You can obtain information on the latest Education Services offerings, course dates, and class locations from the World Wide Web by pointing your Web browser to: http://www.juniper.net/training/education/.

About This Publication

The *JNCIS-SP Part 2 Study Guide* was developed and tested using software Release 10.3R1.9. Previous and later versions of software might behave differently so you should always consult the documentation and release notes for the version of code you are running before reporting errors.

This document is written and maintained by the Juniper Networks Education Services development team. Please send questions and suggestions for improvement to training@juniper.net.

Technical Publications

You can print technical manuals and release notes directly from the Internet in a variety of formats:

- Go to http://www.juniper.net/techpubs/.
- Locate the specific software or hardware release and title you need, and choose the format in which you want to view or print the document.

Documentation sets and CDs are available through your local Juniper Networks sales office or account representative.

Juniper Networks Support

For technical support, contact Juniper Networks at http://www.juniper.net/customers/support/, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

JNCIS-SP Study Guide—Part 2

Chapter 1: Carrier Ethernet

This Chapter Discusses:

- Carrier Ethernet;
- Different Ethernet standards organizations; and
- Layer 2 services that are available on the Juniper Networks MX Series 3D Universal Edge Routers.

Local Area Network

A LAN is usually a network of Ethernet switches and bridges that provides connectivity between end stations that in general are very close together. In most cases, the end stations and switches are within the same building.

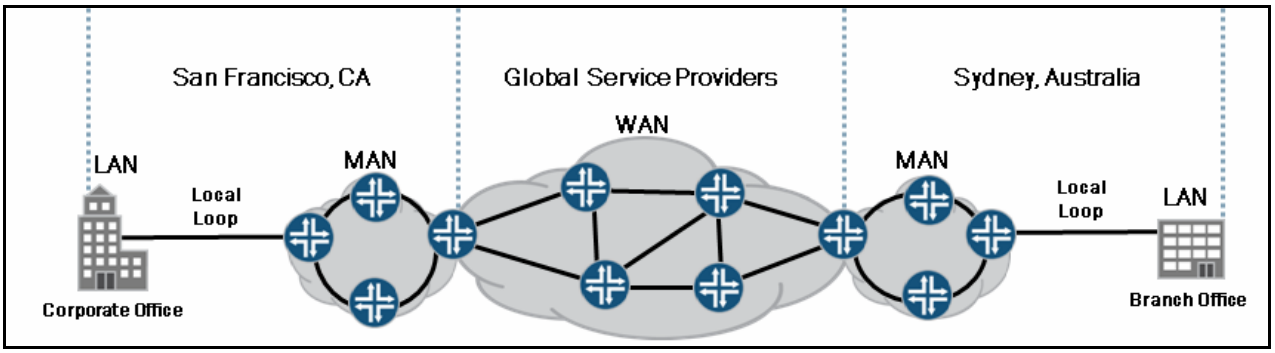
Metropolitan Area Network

A Metropolitan Area Network (MAN) is located within the confines of a city or town where a service provider might have a fiber infrastructure or a cable infrastructure. A MAN provides the ability to connect customer sites that are located near each other.

Wide Area Network

A WAN allows for connectivity that extends far beyond the MAN. A WAN typically connects devices that are hundreds and thousands of miles away from each other.

Service Providers



When a business decides to interconnect two or more sites that are not physically near each other, a service provider usually provides MAN or WAN connectivity between those sites at a price. A service provider (cable company or telco) has the facilities—such as the miles and miles of fiber—that are necessary to transfer data around the world. A customer of the service provider gains access to the MAN or WAN through a local loop or access circuit that the service provider delivers to each site.

Site-to-Site Connectivity Options

Customers have options when it comes to ordering service from the service provider. If the customer sites are relatively close to each other, the customer can purchase a private-line service from the service provider. A private line is a point-to-point circuit that customers can order at varying speeds (DS0, T1, E1, T3, and more). As the distance between sites grows, so does the price for the private-line service. Other options for site-to-site connectivity include Asynchronous Transfer Mode (ATM), Frame Relay, and now Ethernet.

Equipment and Expertise

To support the site-to-site connectivity, the customer must purchase the correct equipment and have the expertise to be able to support the new circuits. The customer will need Ethernet experts for the LAN, and, in the case of ATM WAN connectivity, they will need ATM experts as well.

Bandwidth Becomes a Factor

Over the last 10 years, the need for high-speed access to the Internet as well as for site-to-site connectivity has skyrocketed. With more and more video, voice, and other bandwidth-hogging applications being placed on the network, ATM and Frame Relay networks have not been able to keep up with the demand.

Ethernet Is the Solution

Ethernet interfaces as fast as 10 Gbps are available. Soon speeds will exceed that limit as well. An Ethernet solution in the WAN benefits both the service provider and the customer in many ways. Using Ethernet as the WAN solution, the customer no longer needs Ethernet and ATM experts to run the network. Service providers can offer multiple services using a single interface to the customer. The list of benefits can go on and on.

Scalability

Allowing an Ethernet WAN to scale has always posed a challenge to the service provider. For instance, for an Ethernet switch to forward Ethernet frames it must learn the MAC address of each of the end stations on the customer network. For a service provider serving thousands of customers, this need might mean that the service provider-owned switches must potentially learn millions of MAC addresses. Also, when redundant links exist between the service provider and its customers for resiliency purposes, the question arises, “How can you prevent a loop?” The spanning tree protocols of today simply cannot scale to prevent the loops of thousands of customer sites.

Service-Level Agreements

Usually when a customer purchases WAN service, service-level agreements (SLAs) are in place to ensure that the service provider provides a good service to the customer. Common SLAs would cover frame delay and frame loss.

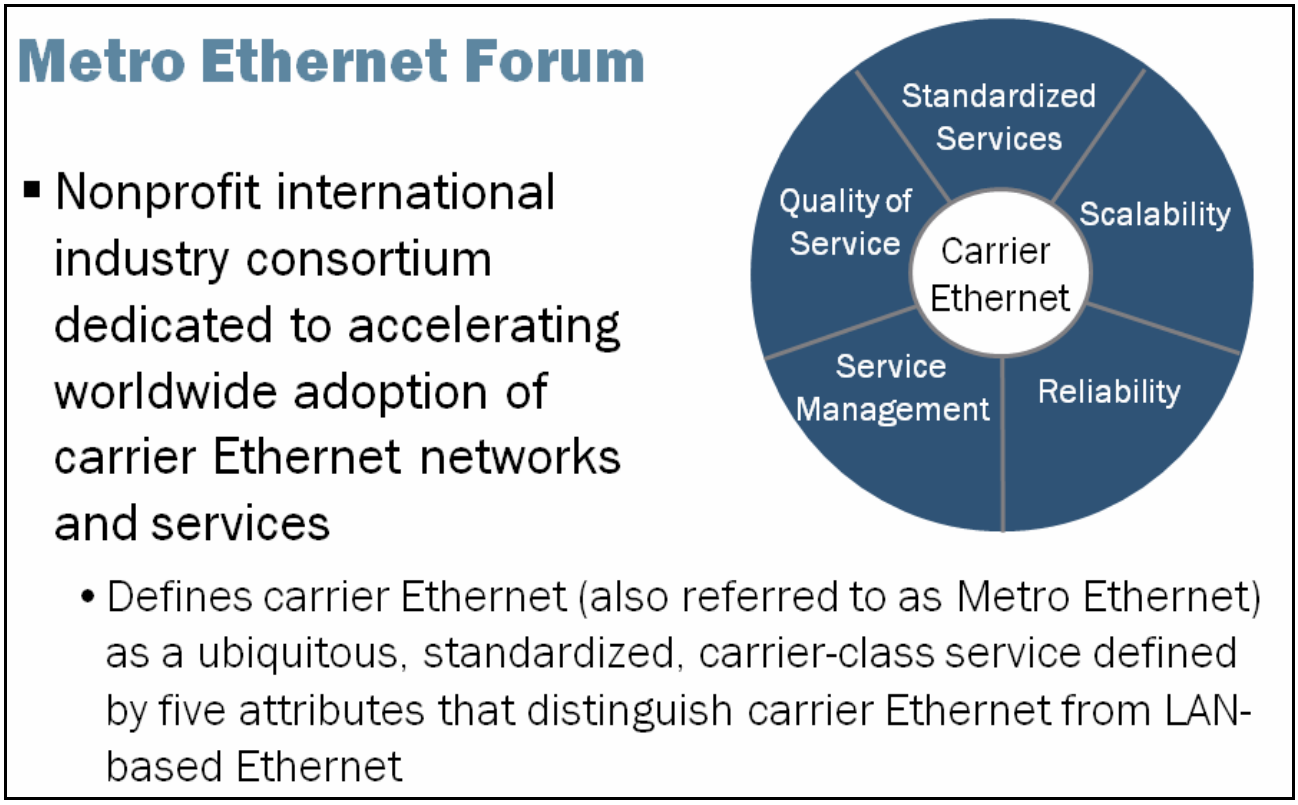
Operation, Administration, and Maintenance

The ability for a service provider to provide and prove the same level of service with Ethernet that a customer could get from ATM, Frame Relay, and private-line service needed to be developed. Ethernet was also lacking Operation, Administration, and Maintenance (OAM) features. For example, in the case of ATM, OAM features would allow administrators to verify the status of ATM permanent virtual circuits (PVCs). This same capability was necessary for Ethernet virtual connections (EVCs).

Ethernet Standards Organizations

Several organizations have been working to solve the problems that Ethernet poses in the WAN. The three primary organizations that are helping to enable Ethernet WAN services are the Metro Ethernet Forum (MEF), the Institute of Electrical and Electronics Engineers (IEEE), and the International Telecommunication Union (ITU).

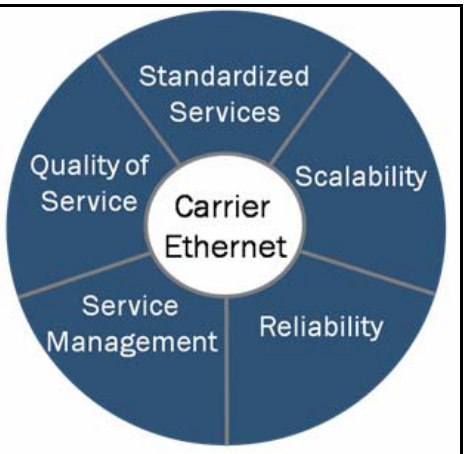
Metro Ethernet Forum



The MEF, as the defining body of carrier Ethernet, is a global alliance of over 150 organizations including service providers, cable multiple service operators (MSOs), network equipment manufacturers, software manufacturers, semiconductor vendors, and testing organizations. The goal of the MEF is to accelerate the worldwide adoption of carrier Ethernet networks and services. The MEF defines carrier Ethernet as a ubiquitous, standardized, carrier-class service defined by five attributes (illustrated on the slide) that distinguish carrier Ethernet from LAN-based Ethernet. An objective of the MEF is to build a consensus and unite service providers, equipment vendors, and customers on Ethernet service definitions, technical specifications, and interoperability.

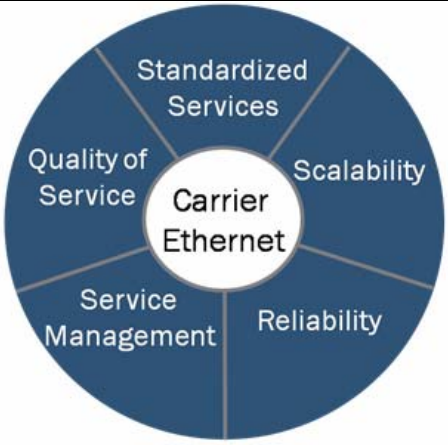
MEF Attributes

- MEF attributes:
  - Standardized services:
    - E-Line, E-LAN, and E-Tree
    - Require no change to customer LAN equipment
    - Suited for converged voice, video, and data networking
    - Wide choice of granularity of bandwidth and quality of service options
  - Scalability:
    - The ability for millions of customers to use the service
    - Spans access, metropolitan, national, and global networks with a wide variety of physical infrastructures and service providers



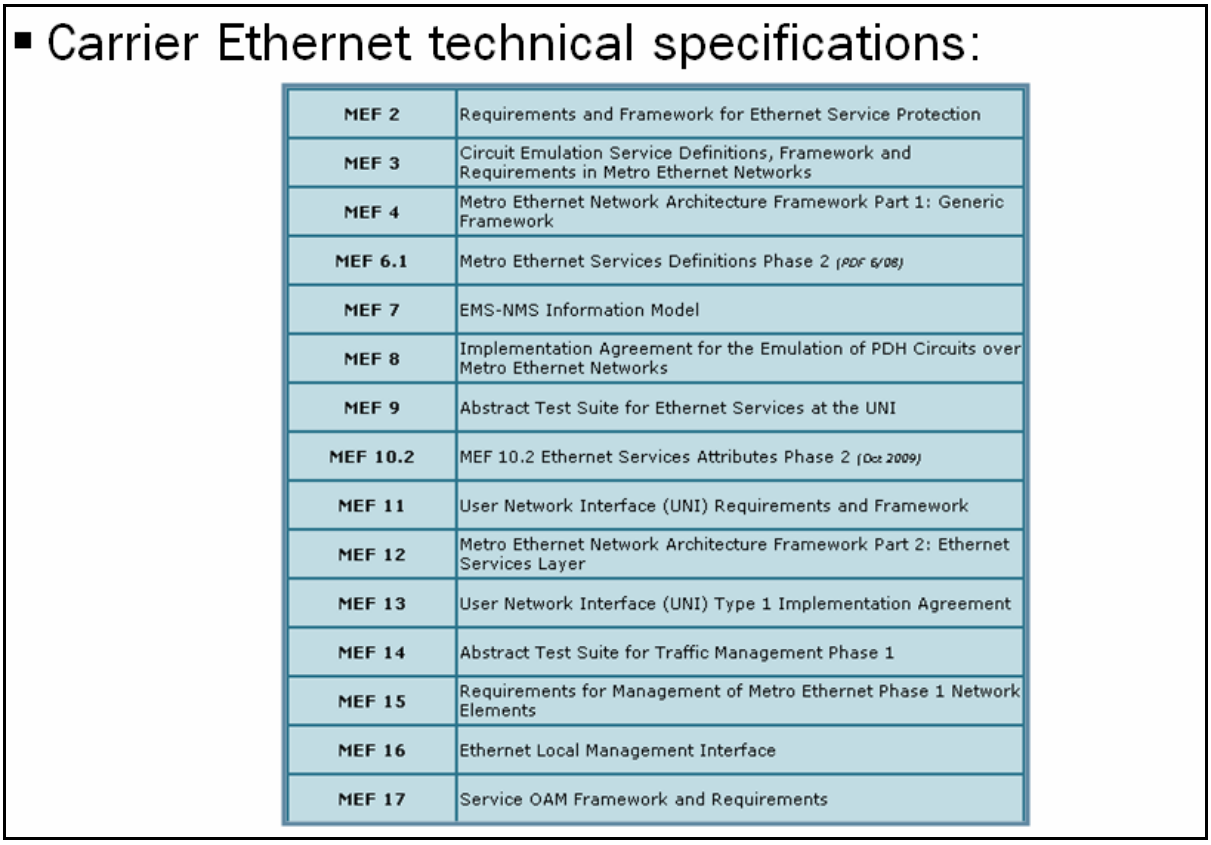
The graphic discusses the definitions of the Standardized Services and Scalability attributes.

- MEF attributes (contd.):
  - Reliability
    - Rapid recovery time
    - The network should be able to detect and recover from outages without impacting users
  - Quality of Service
    - Many bandwidth and quality of service options
    - SLAs for end-to-end performance based on committed information rate, frame loss, delay, and delay variation
  - Service Management
    - Ability to monitor, diagnose, and centrally manage the network using standards-based implementations
    - Carrier class OAM

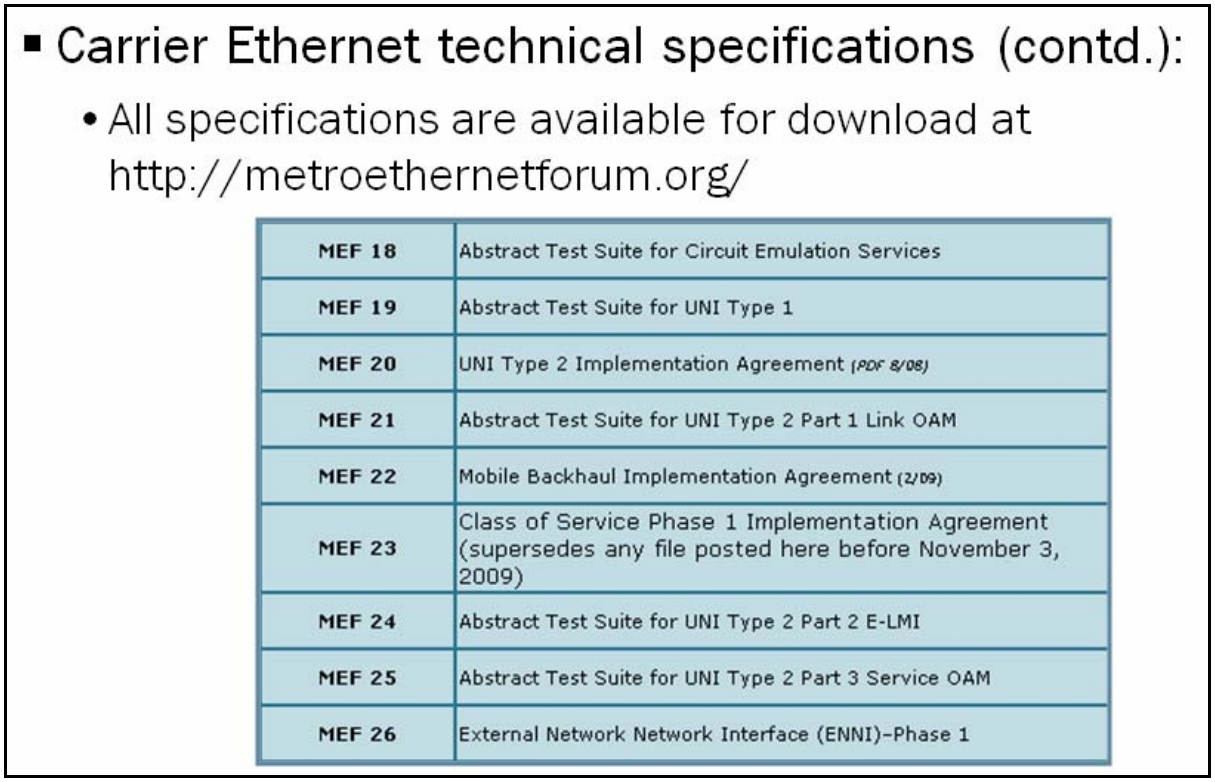


The graphic discusses the definitions of the Reliability, Quality of Service, and Service Management attributes.

Technical Specifications



The graphic shows the MEF-developed carrier Ethernet technical specifications.



The graphic shows the continuation of the MEF-developed carrier Ethernet technical specifications table.

MEF Certification Program

- MEF launched a certification program in 2005 to verify compliance of vendor equipment and service-provider services to MEF technical specifications
  - Eliminates the need for expensive and complex testing between equipment vendors
  - Establishes a solid foundation for carrier Ethernet interoperability
  - Provides for a single, universally recognized test and certification process
  - Accelerates carrier Ethernet deployment at reduced costs
  - Eases making informed decisions about equipment vendors

To help in its objective to promote interoperability between service providers and equipment vendors, the MEF introduced a new certification program in 2005. The certification applies to both service providers and equipment vendors. Having a standardized certification all but eliminates the need for expensive and complex interoperability tests.

MEF 9 Certification

The MEF 9 certification tests for compliance with MEF 6.1, 10, and 11. This test ensures the meeting of all requirements at the user-to-network interface (UNI). Some of the tests include:

- Non-looping frame delivery;
- Single copy broadcast and multicast delivery; and
- Customer VLAN (C-VLAN) ID preservation.

MEF 14 Certification

The MEF 14 certification tests for compliance with MEF 9 and 10. This test ensures the meeting of all requirements for traffic management. Some of the tests include:

- Frame delay service performance;
- Frame delay variation service performance; and
- Frame loss ration service performance.

Continued on next page.

MEF 18 Certification

The MEF 18 certification tests for compliance with MEF 8. This certification ensures the meeting of all requirements for reliable transport of time-division multiplexing (TDM) circuits. This certification includes some of the following tests:

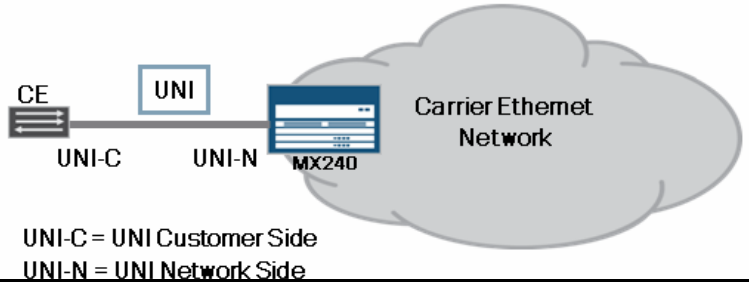
- Encapsulation layers;
- Payload format; and
- Defects.

MEF 21 Certification

The MEF 21 certification tests for compliance with MEF 20. This certification ensures the meeting of all requirements for UNI Type 2 and link OAM features.

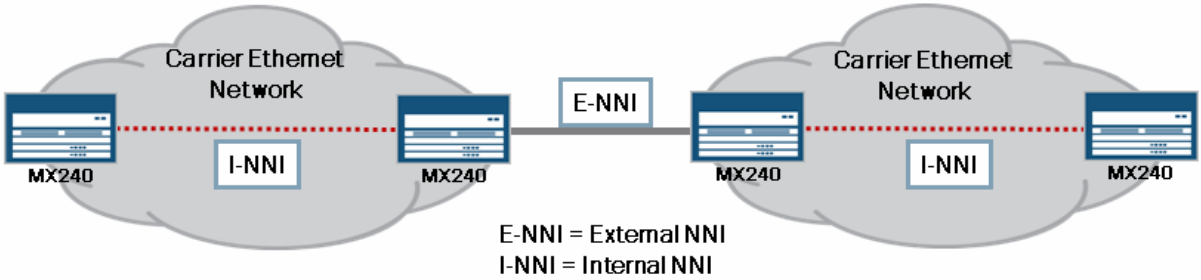
Carrier Ethernet Terms: Part 1

- UNI:
  - A physical interface or port that is the demarcation between the customer and the service provider
  - UNI Type 1: Compliant with MEF 13 and manually configurable
  - UNI Type 2: Automatic service discovery through Ethernet-Local Management Interface; supports OAM
  - UNI Type 3: Provides for dynamic EVC setup
  - An Ethernet interface operating at 10 Mbps, 100 Mbps, 1 Gbps, or 10 Gbps
- Customer equipment



The graphic lists some of the common terms found in a carrier Ethernet network.

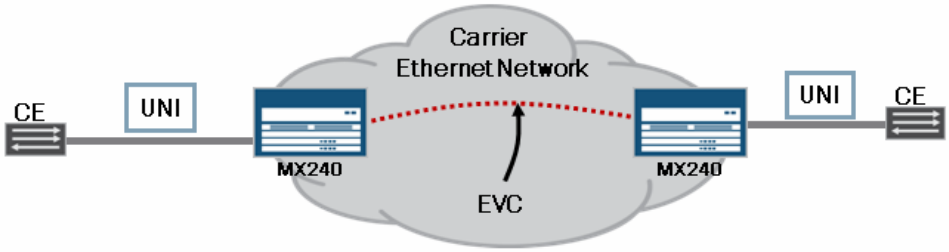
- Network-to-network interface
  - A physical interface or port that is the demarcation between distinct carrier Ethernet networks, operated by one or more service providers
- Carrier Ethernet network
  - An access, metropolitan, national, or global Ethernet transport network connecting user endpoints



The graphic discusses more of the common terms found in a carrier Ethernet network.

Ethernet Virtual Connection

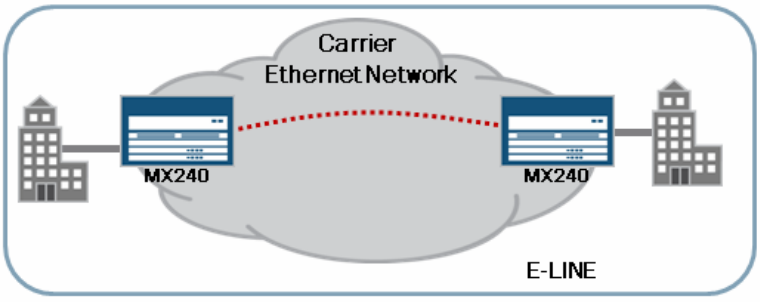
- Connects two or more customer sites or UNIs
- Prevents data transfer between sites that are not part of the same EVC
- Defined in MEF 6.1 and 10.2
  - Point-to-point
  - Multipoint-to-multipoint
  - Rooted multipoint



An (EVC) is a carrier Ethernet service offered by a service provider. It connects two or more sites. A requirement of an EVC is to prevent data transfer between UNIs that are not part of the same EVC. Three types of EVCs exist: point-to-point, multipoint-to-multipoint, and rooted multipoint.

E-Line Service EVC

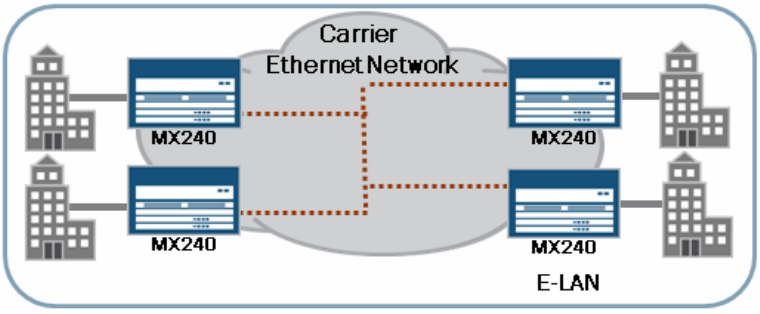
- Two types:
  - Ethernet Private Line (port-based)
  - Virtual Private Line (VLAN-based)
- Allow for communication between only two UNIs



A point-to-point EVC is referred to as an Ethernet Line (E-Line) EVC. It provides connectivity between only two UNIs. Two types of E-Line EVCs exist. An Ethernet Private Line EVC is port-based, where each of the UNIs is a dedicated port to a customer. All virtual LANs (VLANs) for the UNI can traverse the EVC. A Virtual Private Line EVC is VLAN-based, such that it allows for the mapping of individual VLANs to the EVC. This mapping allows the service provider to multiplex multiple customers using a single access port.

E-LAN Service EVCs

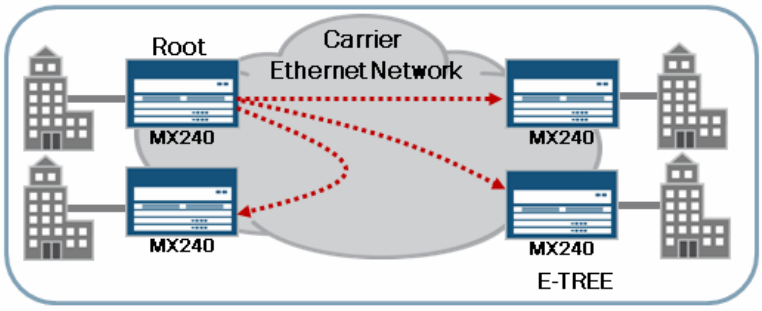
- Two types:
  - Ethernet Private LAN (port-based)
  - Virtual Private LAN (VLAN-based)
- Allows for communication between two or more UNIs
  - Ingress broadcast or multicast frames at one UNI are forwarded to all other UNIs



Multipoint-to-multipoint EVCs are referred to as Ethernet LAN (E-LAN) EVCs. Essentially, an E-LAN EVC makes the service provider network appear to be a single broadcast domain to the customer. E-LAN EVCs come in the form of either an Ethernet Private LAN or Virtual Private LAN, similar to the E-Line EVC.

E-Tree Service EVCs

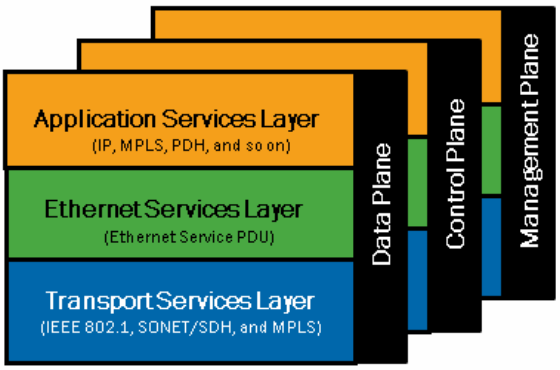
- Two types:
  - Ethernet Private Tree (Port-based)
  - Virtual Private Tree (VLAN-based)
- A root UNI can send ingress frames to one or all leaf UNIs
- A leaf UNI can exchange data only with the root UNI
- Useful for multicast video applications



Rooted multipoint EVCs are referred to as E-Tree EVCs. The slide describes the forwarding properties of an E-Tree EVC. E-Tree EVCs come in the form of either an Ethernet Private Tree or Virtual Private Tree, similar to the E-Line EVC.

MEF's Three-Layer Model

- Application Layer
  - End-user applications carried by the Ethernet Services Layer
- Ethernet Services Layer
  - EVCs
- Transport Services Layer
  - Various networking and media types that deliver the Ethernet services



The MEF has defined a three-layer model for carrier Ethernet networks. The Application Services Layer supports end-user applications. The Ethernet Services Layer carries the applications. This layer is the main focus of the MEF. Carrier Ethernet resides on the Ethernet Services Layer. To deliver the Ethernet services, the Transport Services Layer uses various networking and media types. This layer includes technologies like provider backbone bridging, virtual private LAN service (VPLS), and SONET. As shown on the slide, each layer of the MEF model has its own data, control, and management planes.

IEEE Standards

- The IEEE Ethernet standards fall into the 802 category:
  - IEEE 802.3—Physical Layer and Data Link MAC sublayer for wired Ethernet
  - IEEE 802.1—Bridging and management
    - 802.1D/802.1Q: Bridges and VLAN
    - 802.1ad: Provider bridging
    - 802.1ah: Provider backbone bridging
    - 802.1ag: Connectivity fault management
    - Many more

The graphic lists some of the important IEEE Ethernet standards.

ITU-T Recommendations

- G series—Transmission systems and media, digital systems, and networks
  - G.8010: Architecture of Ethernet Layer networks
  - G.8011.1: Ethernet Private Line Service
  - G.8011.2: Ethernet Virtual Private Line Service
  - G.8032: Ethernet Ring Protection
- Y series—Global information infrastructure, IP aspects, and next-generation networks
  - Y.1730: Ethernet OAM requirements
  - Y.1731: OAM mechanisms

Note: Information is current as of Feb 11, 2010

The graphic lists some of the International Telecommunication Union Telecommunication Standardization (ITU-T) Ethernet recommendations.

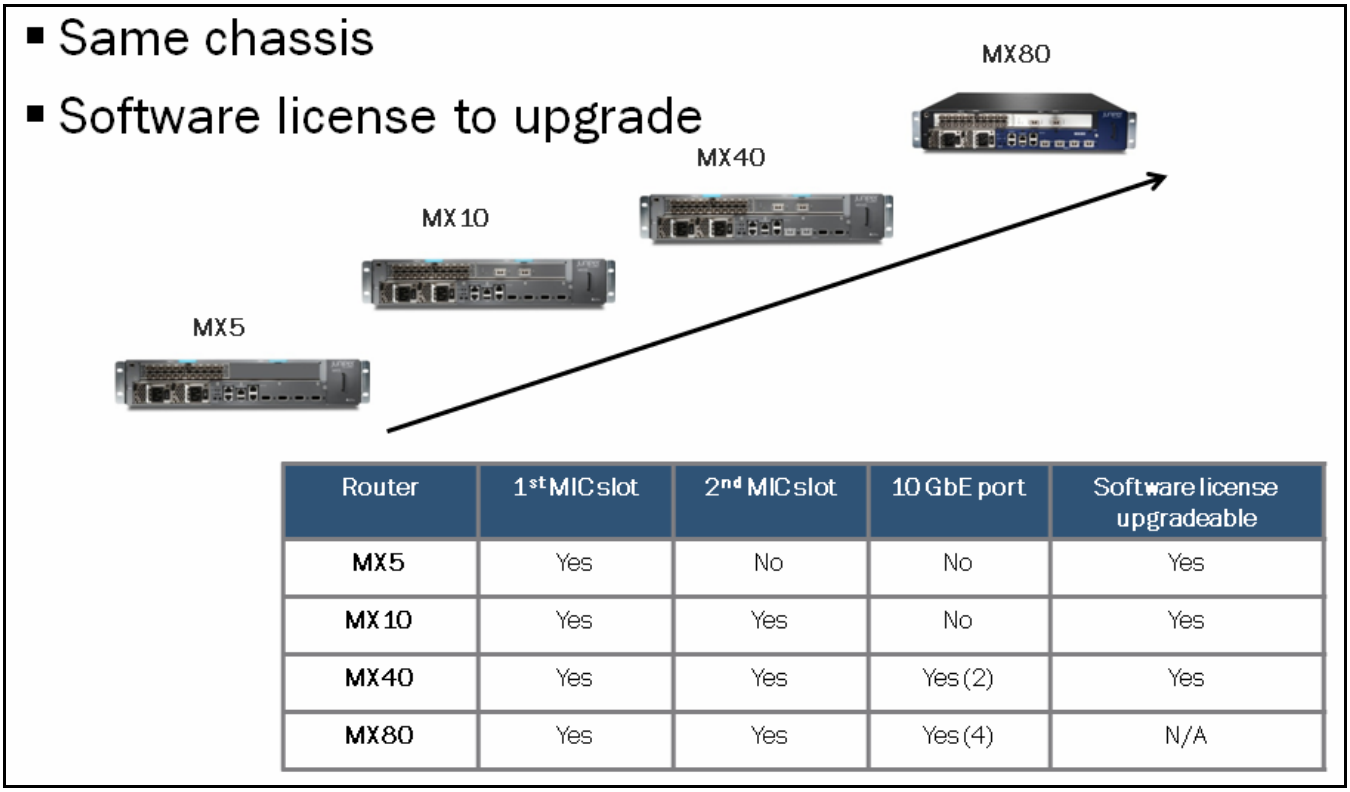
MX Series Highlights

- Designed for next-generation services at the Ethernet edge
  - Function as Layer 2 switches, Layer 3 routers, or both
  - Provider edge for Layer 3 VPN, Layer 2 VPN, VPLS, or the Internet
- Scalable and reliable
  - Fully redundant design
  - Distributed packet forwarding
  - Full set of Junos OS routing capabilities
- MX240 and MX480
  - Mid-range platforms
  - Optimized for sites with space and power restrictions

	MX80	MX240	MX480	MX960
Units per 7′ rack	24	9	6	3
Gbps Capacity	80 G	240 G	480 G	960 G
10 GE and GE ports	Varies	12 / 120	24 / 240	48 / 480
MAC Addresses	1 million			

The graphic shows some of the highlights of the MX Series devices.

MX Series Highlights



Router	1st MIC slot	2nd MIC slot	10 GbE port	Software license upgradeable
MX5	Yes	No	No	Yes
MX10	Yes	Yes	No	Yes
MX40	Yes	Yes	Yes (2)	Yes
MX80	Yes	Yes	Yes (4)	N/A

The graphic shows some of the highlights of the MX Series devices.

Layer 2 Features

- IEEE
  - 802.1D: Bridging
  - 802.1Q: VLAN tagging
  - 802.1ad: Provider bridging
  - 802.1ah: QinQ
  - 802.1ag: CFM
  - 802.3, clause 57: LFM
- ITU-T
  - Y.1731: CFM and Frame Delay Measurement
  - G.8032: Ethernet Ring Protection
- Internet Engineering Task Force
  - RFC 4761: VPLS using BGP
  - RFC 4762: VPLS using LDP

The graphic shows some of the Layer 2 features supported on MX Series devices.

Review Questions

1. List two properties that make carrier Ethernet more desirable than older WAN methods like Frame Relay and ATM.
2. List the three prominent Ethernet standards organizations.
3. List three Layer 2 services that an MX Series device can provide.

Answers

1.

Service providers can offer more that one service to an individual customer over a single access port with carrier Ethernet, and enterprise customers need only hire Ethernet experts to manage the entire network.

2.

The three prominent Ethernet standards organizations are the MEF, the IEEE, and the ITU.

3.

An MX Series device can provide provider bridging, provider backbone bridging, and OAM.

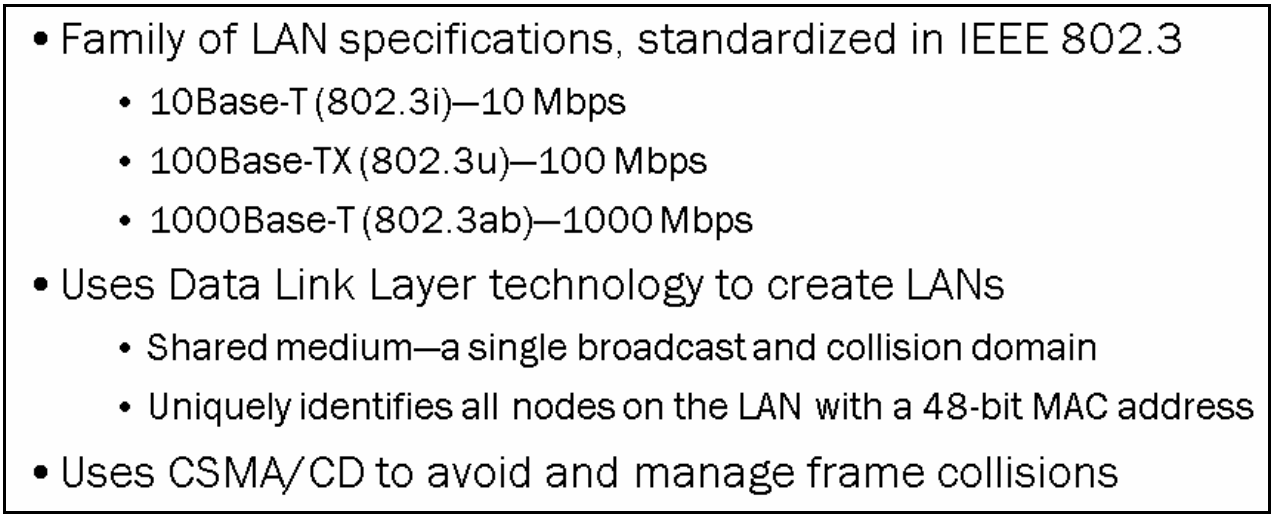
JNCIS-SP Study Guide—Part 2

Chapter 2: Ethernet Switching and Virtual LANs

This Chapter Discusses:

- The functions of an Ethernet LAN;
- Learning and forwarding in a bridging environment;
- Implementation of virtual LAN (VLAN) tagging;
- Automation of VLAN administration through Multiple VLAN Registration Protocol (MVRP);
- Implementation of integrated routing and bridging (IRB);
- Implementation of Layer 2 address learning and forwarding; and
- Implementation of Layer 2 firewall filters.

Ethernet Defined

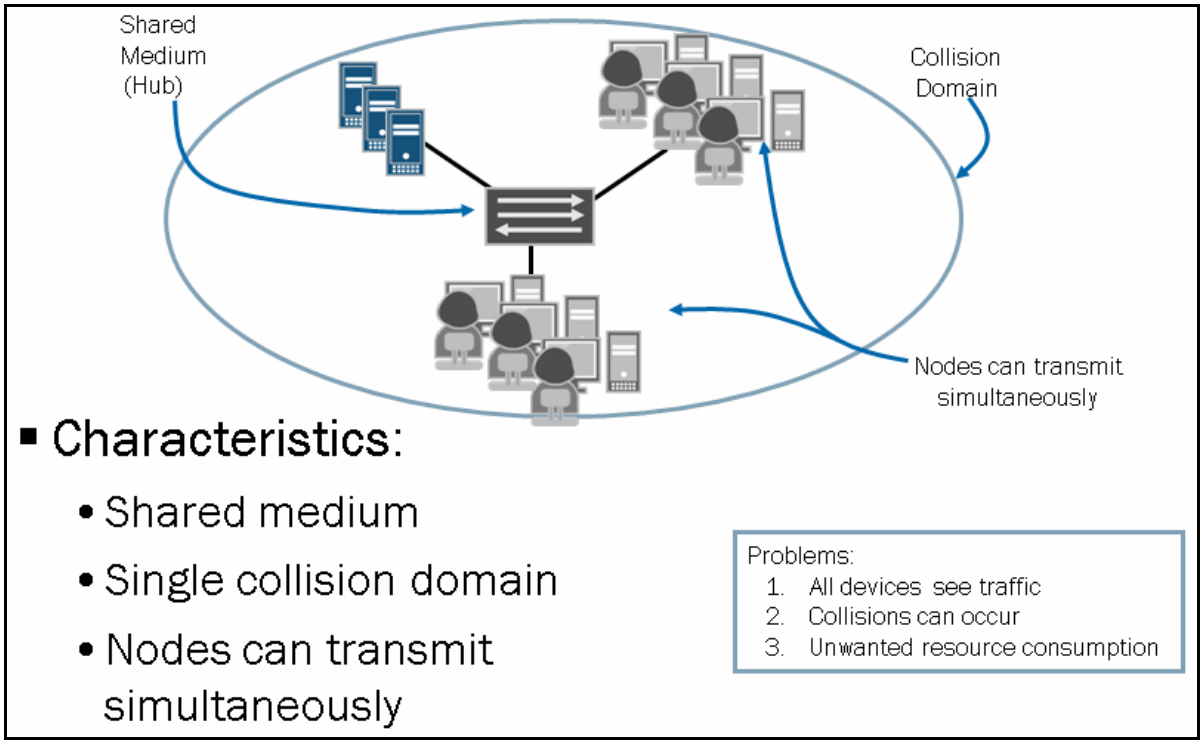


Ethernet is a family of LAN specifications defined in the Institute of Electrical and Electronics Engineers (IEEE) 802.3 standard. The graphic lists some common examples, including the 802.3i, 802.3u, and 802.3ab specifications. Each Ethernet implementation uses a unique wiring and signaling standard—typically a copper-based medium or fiber optics—for the Physical Layer. Although the various implementations of Ethernet can use various wiring and signaling standards, they all use a common addressing format.

Ethernet is a Data Link Layer technology, as defined by Layer 2 of the Open Systems Interconnection (OSI) model of communications. An Ethernet LAN consists of a shared medium that encompasses a single broadcast and collision domain. Network devices, referred to as nodes, on the Ethernet LAN transmit data in bundles that are generally referred to as frames. Each node on a LAN has a unique identifier so that it can be unambiguously located on the network. Ethernet uses the Layer 2 media access control (MAC) address for this purpose. MAC addresses are 48-bit hardware addresses programmed into the Ethernet processor of each node.

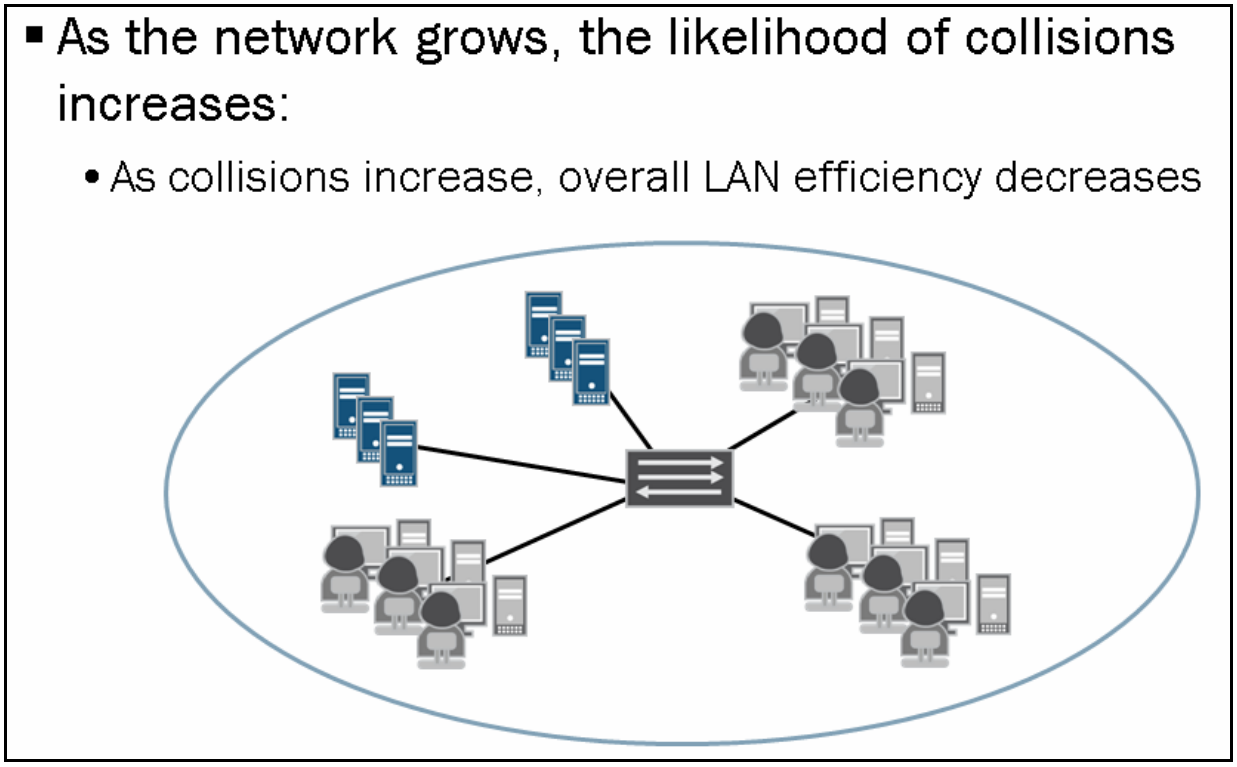
Ethernet uses the carrier-sense multiple access with collision detection (CSMA/CD) protocol to avoid and manage frame collisions.

Ethernet LANs: Part 1



Ethernet LANs consist of a shared medium that defines a single collision domain. As previously mentioned, Ethernet uses the CSMA/CD protocol to help avoid and manage frame collisions. The sample topology on the slide shows a series of nodes connected through a hub using a copper-based physical medium. This type of implementation allows only a single stream of data at a time. All nodes participating in this shared Ethernet LAN listen to verify that the line is idle before transmitting. If the line is idle, the nodes begin transmitting data frames. If multiple nodes listen and detect that the line is idle and then begin transmitting data frames simultaneously, a collision occurs. When collisions occur, an error is generated and travels back to the transmitting devices. When a node receives a collision error message, it stops transmitting immediately and waits for a period of time before trying to send the frame again. If the node continues to detect collisions, it progressively increases the time between retransmissions in an attempt to find a time when no other data is being transmitted on the LAN. The node uses a backoff algorithm to calculate the increasing retransmission time intervals. When a node does successfully transmit traffic, that traffic replicates out all ports on the hub and all other nodes on the shared Ethernet segment see it. This traffic-flooding approach, coupled with collisions, consumes network resources.

Ethernet LANs: Part 2



Ethernet LANs were originally implemented for small, simple networks. Over time, LANs have become larger and more complex. As an Ethernet LAN grows, the likelihood of collisions on that LAN also grows. As more users join a shared Ethernet segment, each participating node receives an increase of traffic from all other participating nodes for which it is not the actual destination. This unwanted consumption of network resources, along with an increase of collisions, inevitably decreases the overall efficiency on the LAN.

Bridging Defined

Defined in the IEEE 802.1D-2004 standard, bridging addresses some of the inherent problems of large, shared Ethernet LANs. Bridging uses microsegmentation to divide a single-collision domain into multiple, smaller, bridged collision domains. Reducing the size of a collision domain effectively reduces the likelihood that collisions might occur. This approach also enhances performance by allowing multiple streams of data to flow through the switch within a common LAN or broadcast domain.

Bridging allows a mixed collection of interface types and speeds to be logically grouped within the same bridged LAN. The ability to logically group dissimilar interfaces in a bridged LAN environment provides design flexibility not found in a shared Ethernet LAN environment.

Bridging builds and maintains a forwarding table, known as a bridge table, for all destinations within the bridged LAN. The bridge table is based on the source MAC addresses for all devices participating in the bridged LAN. The bridge table can aid in intelligent forwarding decisions. This approach reduces unnecessary traffic on the LAN.

Bridging Mechanics

- Transparent bridging builds and maintains bridge tables using the following mechanisms:
  - Learning:
    - Learns MAC addresses and associated ports
  - Forwarding:
    - Forwards packets out the proper egress interface toward the destination
  - Flooding:
    - Replicates packets out *other* ports for unknown destination MAC addresses; also used when passing multicast and broadcast traffic
  - Filtering:
    - Limits traffic to its associated network segment
  - Aging:
    - Ensures bridge table entries are current

The transparent bridging protocol allows a switch to learn information about all nodes on the LAN. The switch uses this information to create the address-lookup tables, referred to as bridge tables, that it consults when forwarding traffic to (or toward) a destination on the LAN.

When a switch first connects to an Ethernet LAN or VLAN, it has no information about other nodes on the network. *Learning* is a process the switch uses to obtain the MAC addresses of all the nodes on the network. It stores these addresses in a bridge table. To learn MAC addresses, the switch reads all frames that it detects on the LAN or on the local VLAN, looking for MAC addresses of sending nodes. It places these addresses into its bridge table, along with two other pieces of information—the interface (or port) on which the traffic was received and the time it learned the address.

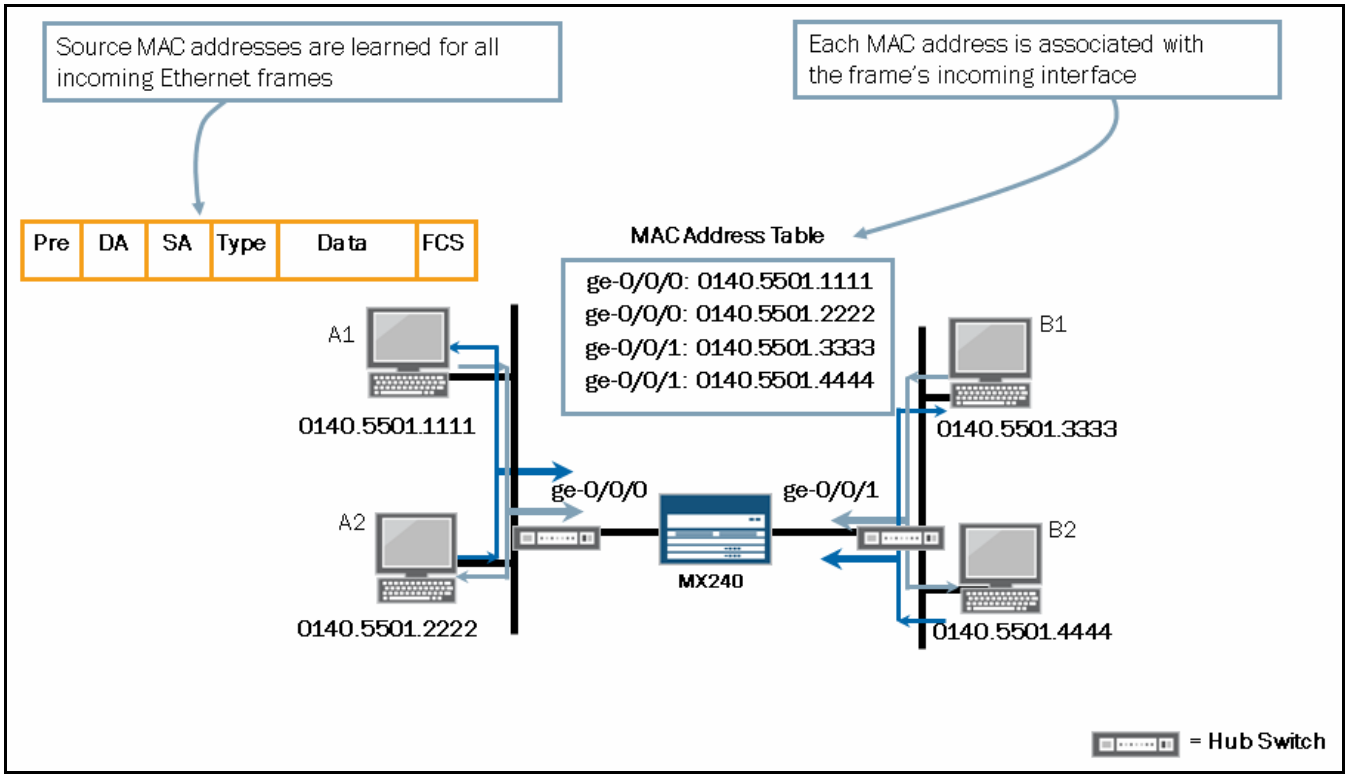
The switch uses the *forwarding* mechanism to deliver traffic, passing it from an incoming interface to an outgoing interface that leads to (or toward) the destination. To forward frames, the switch consults the bridge table to determine whether the table contains the MAC address corresponding to the destination of the frames. If the bridge table contains an entry for the desired destination address, the switch sends the traffic out the interface associated with the MAC address. The switch also consults the bridge table in the same way when transmitting frames that originate on devices connected directly to the switch.

Flooding is a transparent mechanism used to deliver packets to unknown MAC addresses. If the bridging table has no entry for a particular destination MAC address, or if the packet received is a broadcast or multicast packet, the switch floods the traffic out all interfaces except the interface on which it was received. (If traffic originates on the switch, the switch floods that traffic out all interfaces.) When the unknown destination host responds to traffic that has been flooded through a switch, the switch learns the MAC address of that node and updates its bridge table with the source MAC address of the host and ingress port.

The *filtering* mechanism limits traffic to its associated network segment or VLAN. As the number of entries in the bridge table grows, the switch pieces together an increasingly complete picture of the individual network segments—the picture clarifies which nodes belong to which network. The switch uses this information to filter traffic. Filtering prevents the switch from forwarding traffic from one network segment to another.

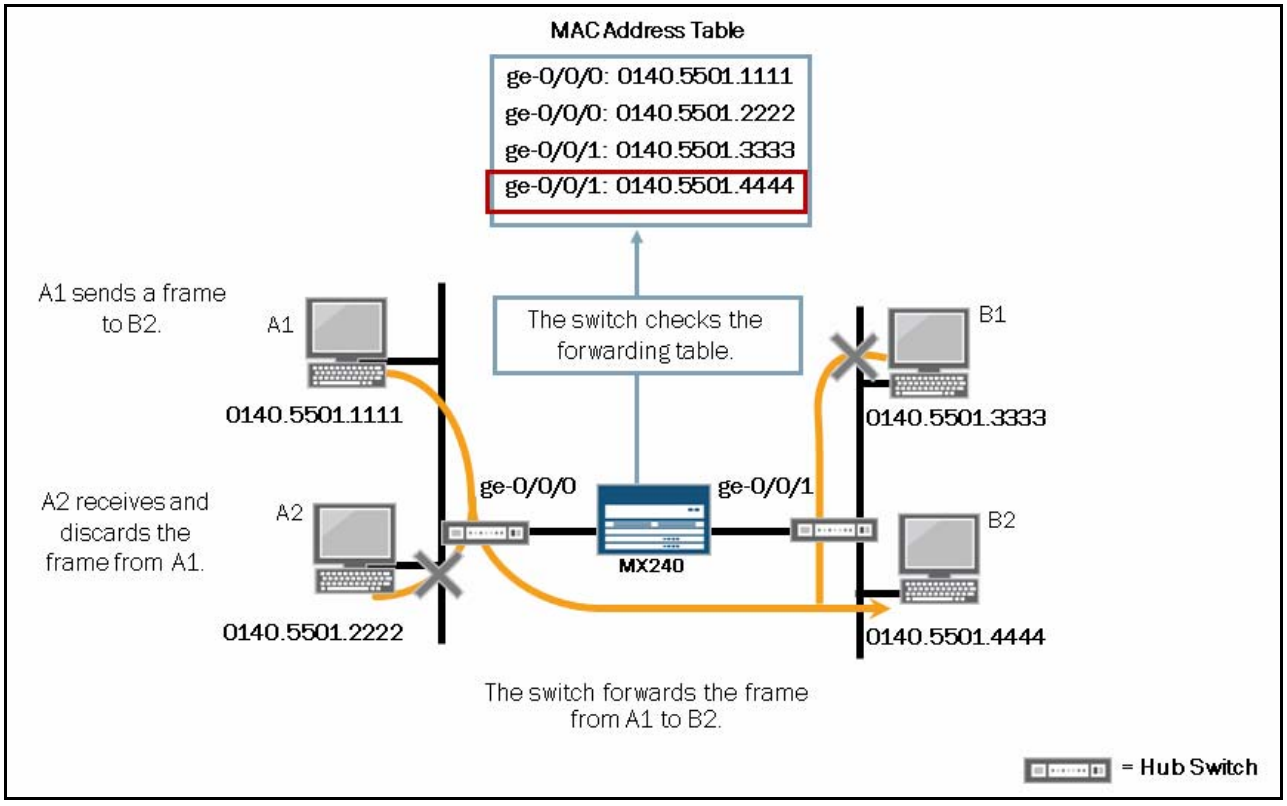
Finally, the switch uses *aging* to ensure that only active MAC address entries are in the bridge table. For each MAC address in the bridge table, the switch records a timestamp of when it learned the information about the network node. Each time the switch detects traffic from a MAC address, it updates the timestamp. A timer on the switch periodically checks the timestamp; if the timestamp is older than the `global-mac-table-aging-time` value (discussed later in this chapter), the switch removes the node's MAC address from the bridge table.

MAC Address Learning



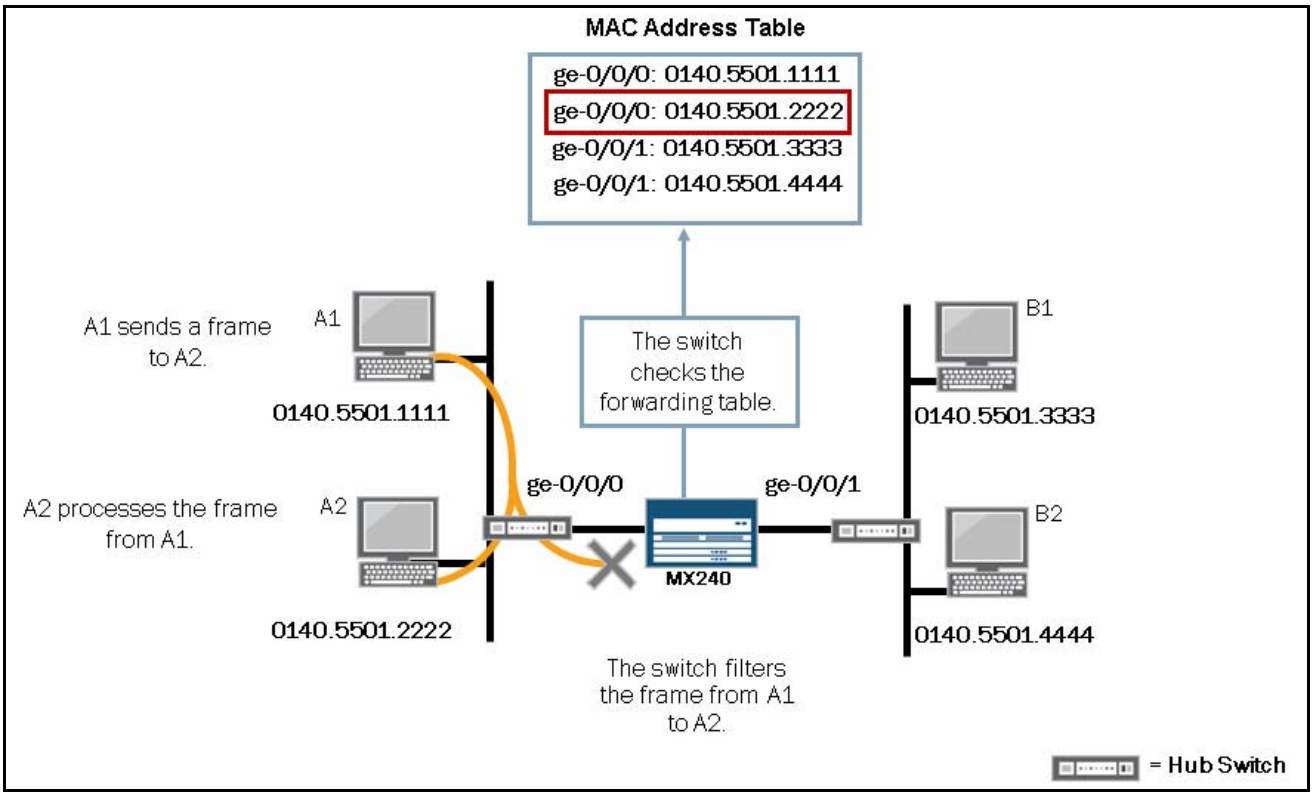
The graphic illustrates a basic view of the MAC address learning process. In this example, each switch port connects to a hub and the individual hubs have multiple connected nodes. As each node sends traffic toward the other nodes on the bridged LAN, the switch reviews that traffic and creates a MAC address table (a bridge table) based on the source address of the sender along with the switch port on which it received the traffic. In this example, we see that the MAC addresses for A1 and A2 are associated with port ge-0/0/0, whereas the MAC addresses for B1 and B2 are associated with port ge-0/0/1.

Forwarding Known Unicast Frames: Part 1



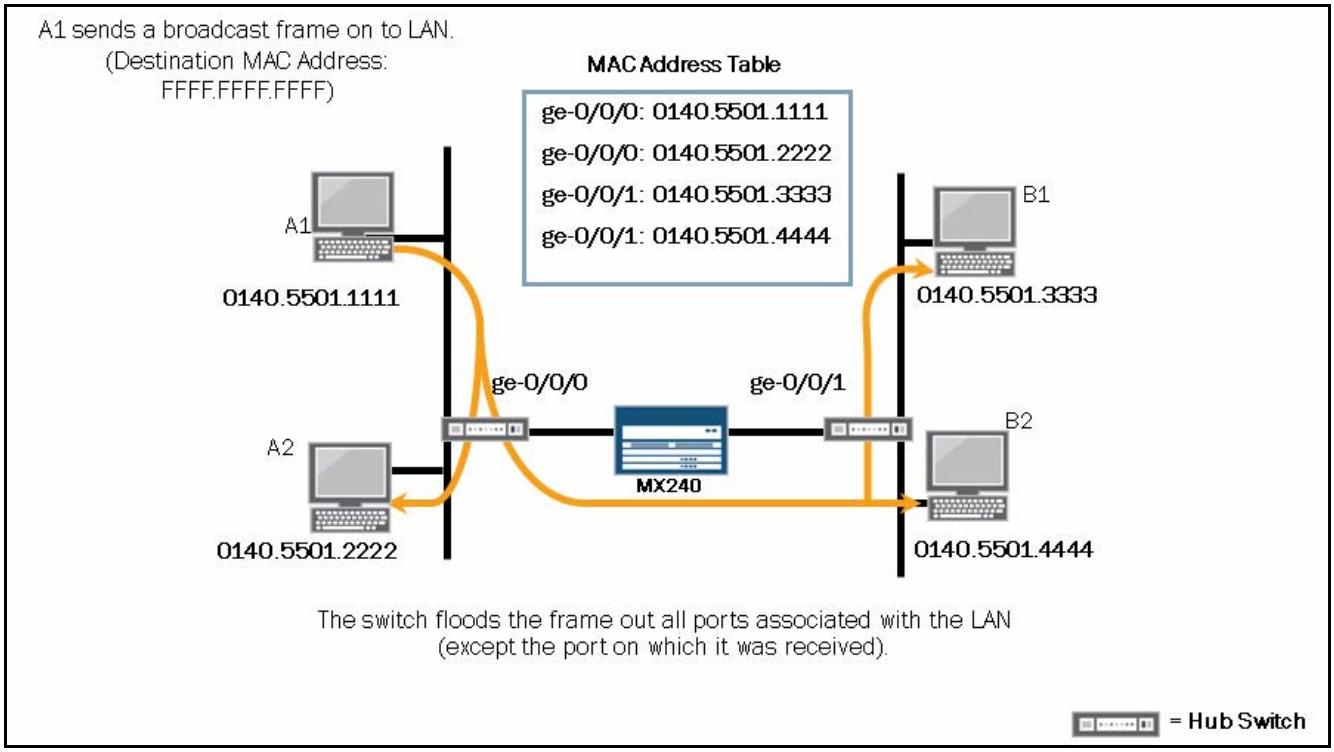
In the example on the slide, A1 sends a frame to B2. The frame is repeated out all ports on the attached hub, which results in frames traveling to both A2 as well as the switch shown in the middle of the illustration. A2 receives the frame and detects that the destination MAC address does not match its own MAC address, at which time A2 discards the frame. The switch receives the frame, checks the MAC address table for a matching entry, and forwards the frame out the associated port based on the lookup results. Ultimately, B2 receives and processes the frame while B1 receives and discards the frame.

Forwarding Known Unicast Frames: Part 2



In this example, A1 sends a frame to A2. The attached hub receives the frame and sends it out all ports, which results in duplicate frames sent to A2 as well as to the switch. A2 receives the frame and detects that the destination MAC address matches its own MAC address, at which time A2 processes the frame. The switch receives the frame and checks the MAC address table for a matching entry. The entry in the MAC address table shows the egress port, which, in this example, is the same port on which the switch received the frame. Because the egress port in the MAC address table is the same port on which the frame was received, the switch filters the frame.

Flooding Frames



Flooding is used to learn a MAC address not recorded in the bridge table. This mechanism is also used when sending broadcast and, in many cases, multicast frames. The example on the slide shows A1 sending a broadcast frame with a destination MAC address of FFFF.FFFF.FFFF to the LAN. The attached hub sends the frame out all ports. The switch floods the broadcast frame out all ports associated with the LAN, except for the port on which it received the frame. The slide shows that, ultimately, all nodes on the LAN receive the frame.

Viewing the MAC Address Table

- Use the **show bridge mac-table** command to view MAC address table entries

```
user@switch> show bridge mac-table

MAC flags (S -static MAC, D -dynamic MAC,
           SE -Statistics enabled, NM -Non configured MAC)

Routing instance : default-switch
 Bridging domain : vlan_100, VLAN : 100
   MAC                 MAC      Logical
   address             flags    interface
   00:21:59:ab:8a:95   D        ge-1/0/0.0
   00:21:59:ab:8a:96   D        ge-1/0/1.0
```

Entries are organized based on associated VLAN.

Use the **show bridge mac-table** command to view all entries within the MAC address table. This command generates a list of learned MAC addresses along with the corresponding VLANs and interfaces. All entries are organized based on their associated VLANs.

Clearing MAC Address Table Entries

- Use the `clear bridge mac-table` command to clear the MAC address table contents

```
user@switch> clear bridge mac-table ?
Possible completions:
  <[Enter]>            Execute this command
  <address>            MAC address
  bridge-domain        Name of bridging domain, or 'all'
  instance             Display information for a specified instance
  interface            Clear media access control table for specified interface
  logical-system       Name of logical system, or 'all'
  vlan-id              Clear MAC address learned on a specified VLAN (0..4095)
  |                    Pipe through a command
```

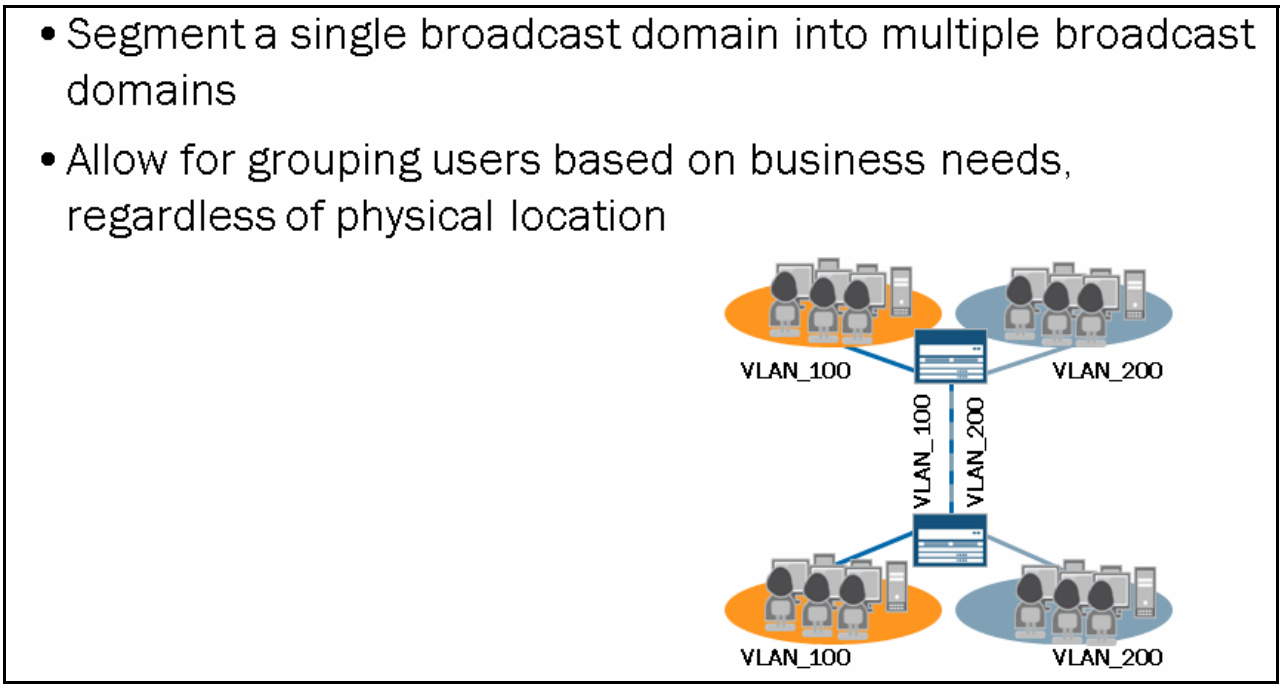
Clear all entries in the table or entries based on a specific property.

Use the **`clear bridge mac-table`** command to clear all entries within the MAC address table. Optionally, you can use the **`interface`** option to clear only those MAC table entries learned through the specified interface. The following example highlights the use of the **`interface`** option:

```
user@switch> show bridge mac-table
MAC flags (S -static MAC, D -dynamic MAC,
           SE -Statistics enabled, NM -Non configured MAC)

Routing instance : default-switch
 Bridging domain : vlan_100, VLAN : 100
   MAC                 MAC      Logical
   address             flags    interface
   00:21:59:ab:8a:95   D        ge-1/0/0.0
   00:21:59:ab:8a:99   D        ge-1/0/3.0
...
Routing instance : default-switch
 Bridging domain : vlan_200, VLAN : 200
   MAC                 MAC      Logical
   address             flags    interface
   00:21:59:ab:8a:97   D        ge-1/0/2.0
   00:21:59:ab:8a:99   D        ge-1/0/3.0
user@switch> clear bridge mac-table interface ge-1/0/3.0

user@switch> show bridge mac-table

MAC flags (S -static MAC, D -dynamic MAC,
           SE -Statistics enabled, NM -Non configured MAC)

Routing instance : default-switch
 Bridging domain : vlan_100, VLAN : 100
   MAC                 MAC      Logical
   address             flags    interface
   00:21:59:ab:8a:95   D        ge-1/0/0.0


MAC flags (S -static MAC, D -dynamic MAC,
           SE -Statistics enabled, NM -Non configured MAC)
```

```
Routing instance : default-switch
 Bridging domain : vlan_200, VLAN : 200
   MAC                 MAC      Logical
   address             flags    interface
   00:21:59:ab:8a:97   D        ge-1/0/2.0
```

VLANs Defined



A VLAN is a collection of network nodes that are logically grouped together to form separate broadcast domains. A VLAN has the same general attributes as a physical LAN, but it allows all nodes for a particular VLAN to be grouped together, regardless of physical location. One advantage of using VLANs is design flexibility. VLANs allow grouping of individual users based on business needs. You can establish and maintain connectivity within a VLAN through software configuration, which makes VLANs such a dynamic and flexible option in today's networking environments.

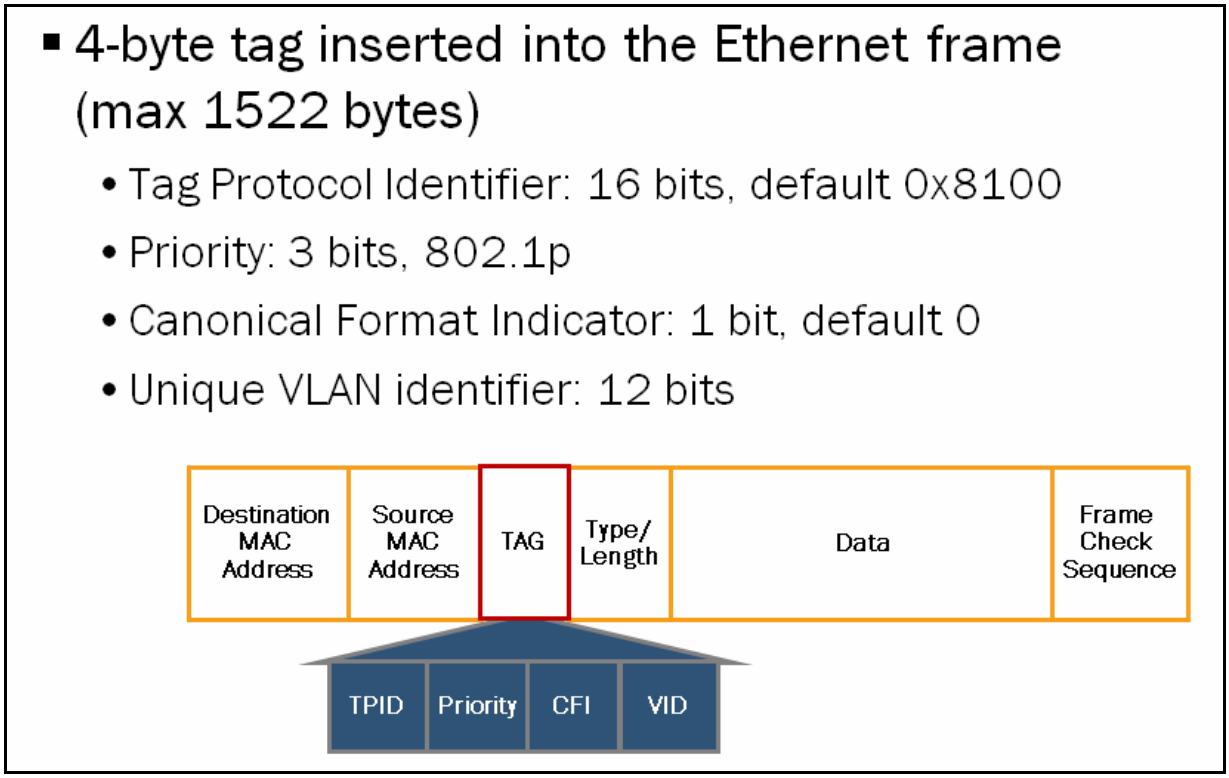
Switch Port Modes

Switch ports operate in either access mode or trunk mode.

An access port connects to network devices such as desktop computers, IP phones, printers, or file servers. Access ports typically belong to a single VLAN and transmit and receive untagged Ethernet frames.

A trunk port typically connects to another switch or to a customer edge router. Interfaces configured for trunk mode handle traffic for multiple VLANs, multiplexing the traffic for all configured VLANs over the same physical connection, and separating the traffic by tagging it with the appropriate VLAN ID. Trunk ports can also carry untagged traffic when configured with the **`native-vlan-id`** statement. Furthermore, trunk ports send control traffic untagged.

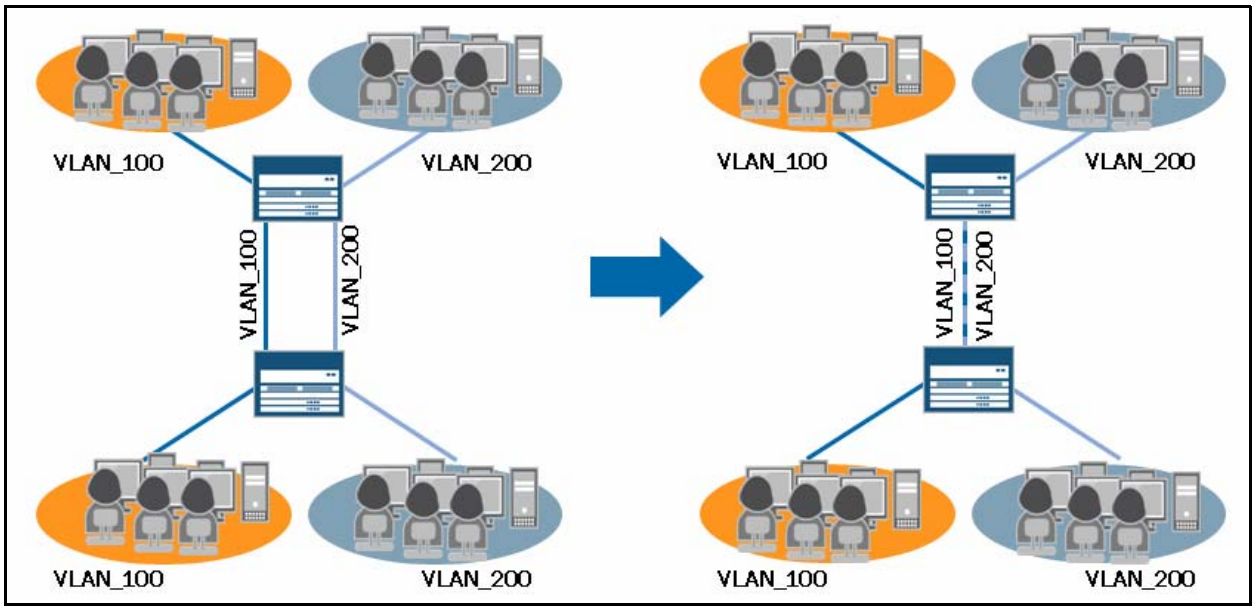
802.1Q—Ethernet Frame



To consistently associate traffic with a particular VLAN, the individual frames must be tagged as they pass throughout a network. The slide illustrates an 802.1Q-tagged Ethernet frame along with the key components of the tag:

- Tag Protocol Identifier (TPID);
- Priority;
- Canonical Format Indicator (CFI); and
- Unique VLAN identifier (VID).

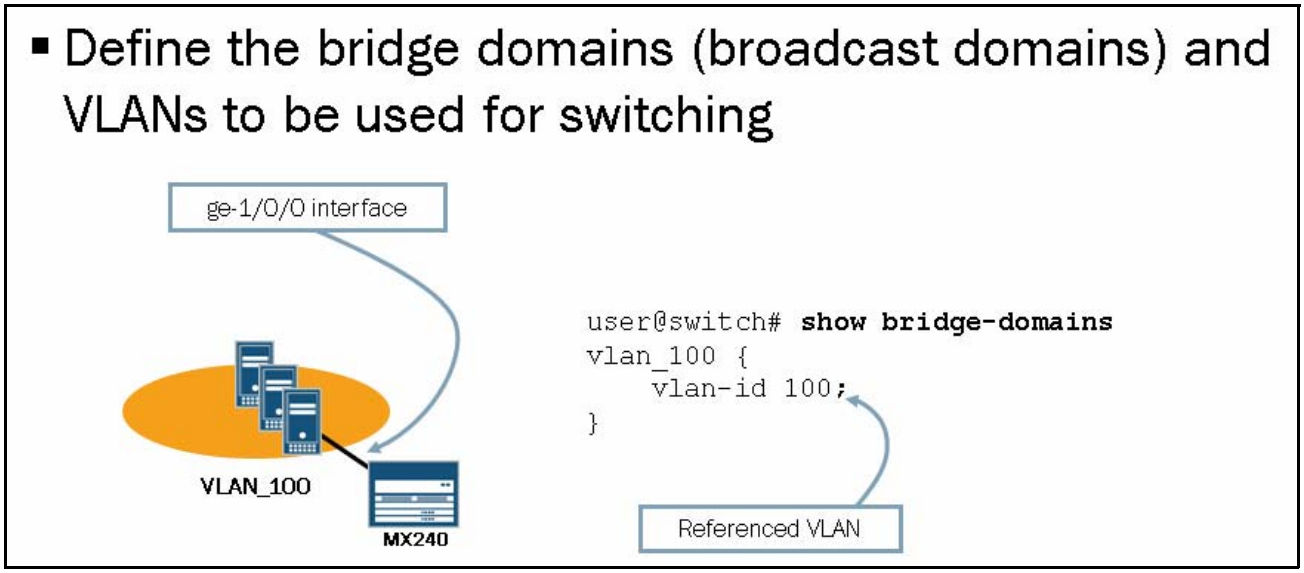
802.1Q Trunk Links



A trunk is a single Ethernet link used to carry traffic for multiple VLANs. A trunk link typically interconnects multiple switches or a switch with a customer edge router. As shown on the slide, interfaces configured as trunk ports handle traffic for multiple

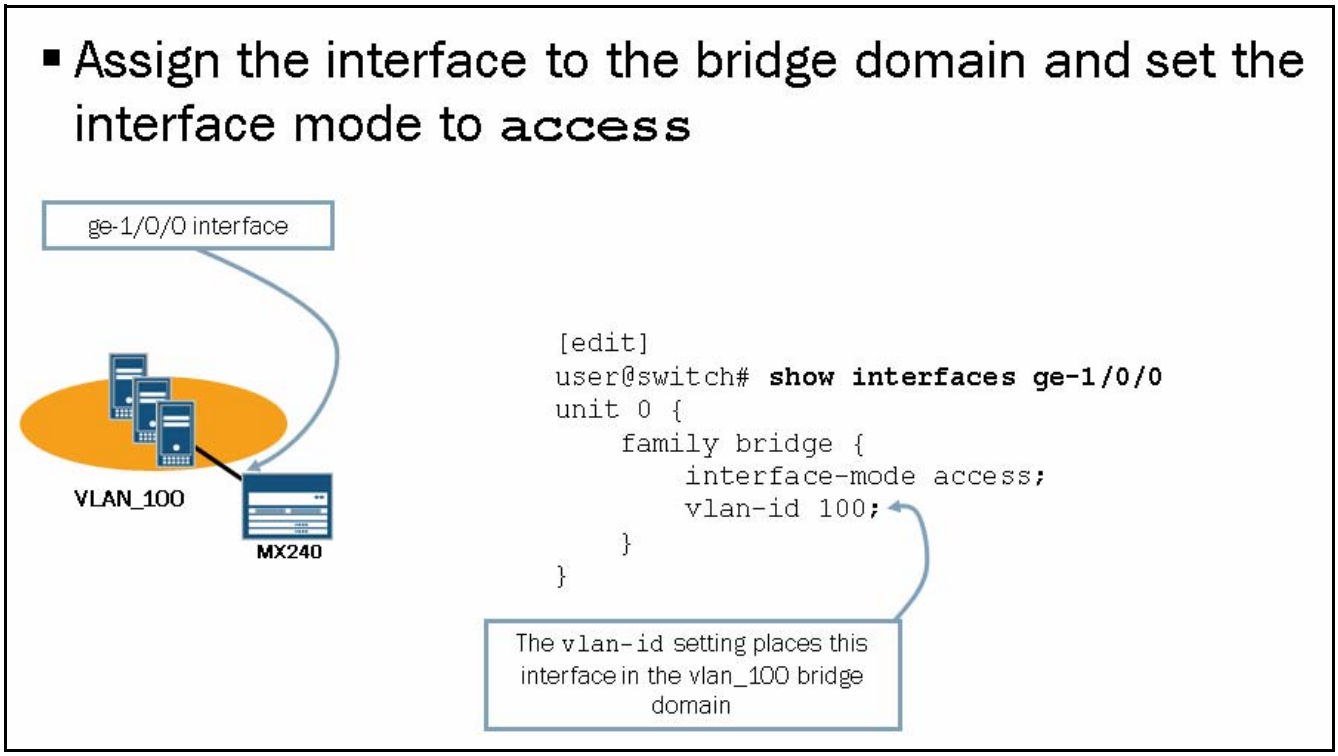
VLANs, multiplexing the traffic for all configured VLANs over a single physical connection rather than using separate physical links for each configured VLAN.

Define a Bridge Domain



To allow an MX Series 3D Ethernet Universal Edge Router to act as a switch and build a MAC address table, you must first specify the particular VLAN IDs that it will use for the purpose of switching. To do so, specify the appropriate VLAN ID as part of a named bridge domain. This method requires that you configure each VLAN as part of a single bridge domain. On a following slide, we cover how we can specify several VLANs within a single bridge domain using the `vlan-id-list` statement.

Assign an Interface to a Bridge Domain



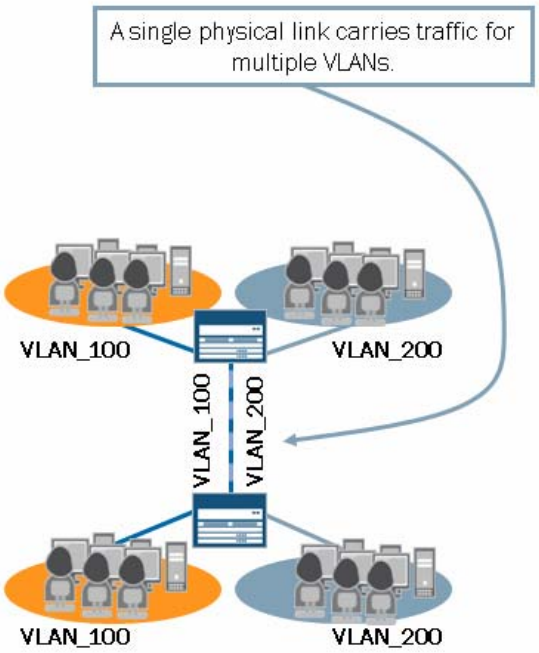
To allow an interface to act as an access port for a particular VLAN, you must specify its interface mode as **access** and you must specify the VLAN to which it belongs. For access ports, you must use **0** as the unit number.

To view the alternate, original style configuration method, please see Appendix A.

802.1Q Trunk Configuration Example

```
[edit]
user@switch# show interfaces
...
ge-1/0/3 {
    native-vlan-id 100;
    vlan-tagging;
    unit 0 {
        family bridge {
            interface-mode trunk;
            vlan-id-list [ 100 200 ];
        }
    }
}

[edit]
user@switch# show bridge-domains
vlan_100 {
    vlan-id 100;
}
vlan_200 {
    vlan-id 200;
}
```

Used for received untagged frames



The graphic illustrates an 802.1Q trunk configuration example. In this case, the interface is configured as a trunk port and is associate with the vlan_100 and vlan_200 bridge domains. The partnering switch would have a similar configuration for the interface functioning as a trunk.

The graphic also illustrates the usage of the `native-vlan-id` statement. This configuration statement does two things. First, if interface ge-1/0/3 receives any untagged frames, it associates those frames to VLAN 100. Second, if interface ge-1/0/3 transmits any outgoing frames that associate with VLAN 100, they transmit as untagged frames.

Notice the `vlan-id-list` statement. It specifies the VLANs to which the interface will be a member. The following statements are examples of how you can use the `vlan-id-list` statement:

- `vlan-id-list [100]`: VLAN 100 only;
- `vlan-id-list [100-200]`: All VLANs between 100 and 200, inclusive;
- `vlan-id-list [100-109 111-200]`: All VLANs between 100 and 200, except VLAN 110; or
- `vlan-id-list [100-109 111 113-200]`: All VLANs between 100 and 200, except VLAN 110 and 112.

Dealing with Many VLANs

- Service providers typically deal with thousands of bridge domains and VLANs for each switch:
  - Use a single statement to create multiple bridge domains
    - Bridge domain names take the form *prefix*-vlan-*number*

```
[edit]
user@switch# show bridge-domains
customer {
    vlan-id-list [ 2-400 405 409-650 ];
}
user@switch> show bridge domain
Routing instance        Bridge domain            VLAN ID     Interfaces
…
default-switch          customer-vlan-0099       99
default-switch          customer-vlan-0100       100
                                                            ge-1/0/0.0
                                                            ge-1/0/1.0
                                                            ge-1/0/3.0
default-switch          customer-vlan-0101       101
…
```

As opposed to configuring individual bridge domains for each VLAN used for switching, the Junos operating system allows for the configuration of many VLANs within a single bridge domain. The slide shows that instead of using the `vlan-id` statement, you would use the `vlan-id-list` statement. The usage of this statement is similar to the usage described on the previous page. When using the `vlan-id-list` statement, the switch automatically configures the appropriate bridge domains, which have names that take the form *`prefix`*`-vlan-`*`number`*, where the *`prefix`* is the configured bridge domain name.

Monitoring VLAN Assignments

```
user@switch> show bridge domain

Routing instance        Bridge domain            VLAN ID     Interfaces
default-switch          vlan_100                 100
                                                            ge-1/0/0.0
                                                            ge-1/0/1.0
                                                            ge-1/0/3.0
default-switch          vlan_200                 200
                                                            ge-1/0/2.0
                                                            ge-1/0/3.0
                                                            ge-1/1/4.0

user@switch> show bridge domain vlan_100 detail

Routing instance: default-switch
  Bridge domain: vlan_100                       State: Active
  Bridge VLAN ID: 100
  Interfaces:
    ge-1/0/0.0
    ge-1/0/1.0
    ge-1/0/3.0
  Total mac count: 2
```

The graphic shows some key commands used to monitor VLAN assignments. In this example, the ge-1/0/3 interface belongs to the bridge domain named `vlan_100`, which has an 802.1Q tag of 100. Because this interface is configured as an access port, it receives and transmits only untagged frames. If a trunk port were also configured to pass traffic for the vlan_100 bridge domain, it would add and remove the 802.1Q tag value of 100 for all traffic for the `vlan_100` bridge domain. We look at a trunk port configuration and monitoring example next.

Monitoring 802.1Q Trunks: Part 1

- Use the **show interfaces** command to determine the interface mode

```
user@switch> show interfaces ge-1/0/3
Physical interface: ge-1/0/3, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 175
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation:
Enabled, Remote fault: Online,
  Speed-negotiation: Disabled, Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  …

  Logical interface ge-1/0/3.0 (Index 75) (SNMP ifIndex 226)
    Flags: SNMP-Traps 0x24020000 Encapsulation: Ethernet-Bridge
    Input packets : 335
    Output packets: 345
    Protocol bridge, MTU: 1514
      Flags: Trunk-Mode
```

The interface is in trunk mode.

The **show interfaces** command shows that the ge-1/0/3 interface is configured for trunk mode, meaning it will transmit VLAN tags.

Monitoring 802.1Q Trunks: Part 2

- Use the **show bridge domain** command to determine the interface VLAN assignments

```
user@switch> show bridge domain

Routing instance        Bridge domain            VLAN ID     Interfaces
default-switch          vlan_100                 100
                                                             ge-1/0/0.0
                                                             ge-1/0/1.0
                                                             ge-1/0/3.0
default-switch          vlan_200                 200
                                                             ge-1/0/2.0
                                                             ge-1/0/3.0
                                                             ge-1/1/4.0
```

The interface is in trunk mode using vlan-id 100 and 200.

The **show bridge domain** command shows the interfaces and their VLAN assignments.

Monitor Bridge Statistics

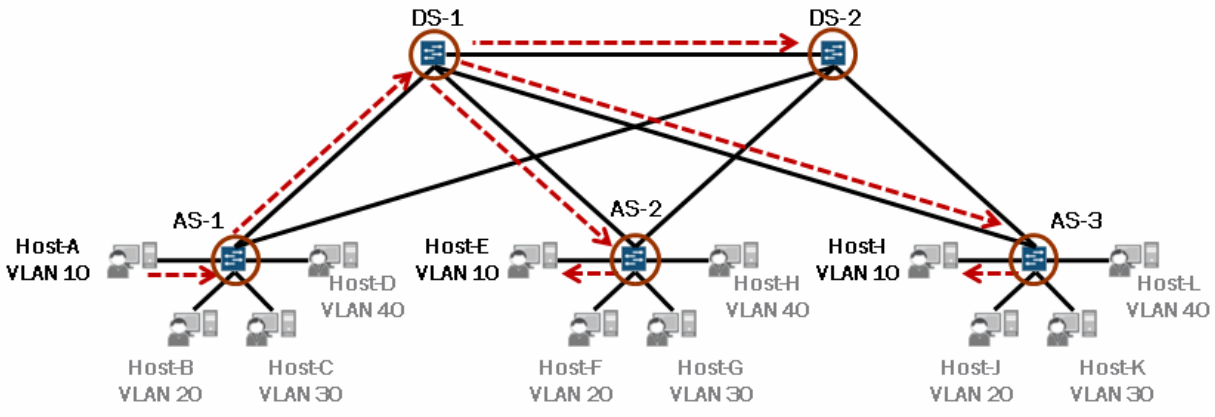
```
user@switch> show bridge statistics
   Local interface: ge-1/0/0.111, Index: 81
     Broadcast packets:                    31
     Broadcast bytes  :                  1984
     Multicast packets:                     0
     Multicast bytes  :                     0
     Flooded packets  :                    31
     Flooded bytes    :                  2362
     Unicast packets  :                 24093
     Unicast bytes    :               2553438
     Current MAC count:                     1 (Limit 1024)
   Local interface: ge-1/0/0.112, Index: 80
     Broadcast packets:                     0
     Broadcast bytes  :                     0
     Multicast packets:                     0
     Multicast bytes  :                     0
     Flooded packets  :                     0
     Flooded bytes    :                     0
     Unicast packets  :                     0
     Unicast bytes    :                     0
     Current MAC count:                     0 (Limit 1024)
…
```

Current number of MACs learned by the interface

The **`show bridge statistics`** command displays traffic statistics and MAC count information related to each logical interface of the switch.

Test Your Knowledge: Part 1

- On which switches would you expect to see traffic sourced from Host-A and destined to an unknown destination in the same VLAN (assume all inter-switch connections are trunks supporting all defined VLANs)?

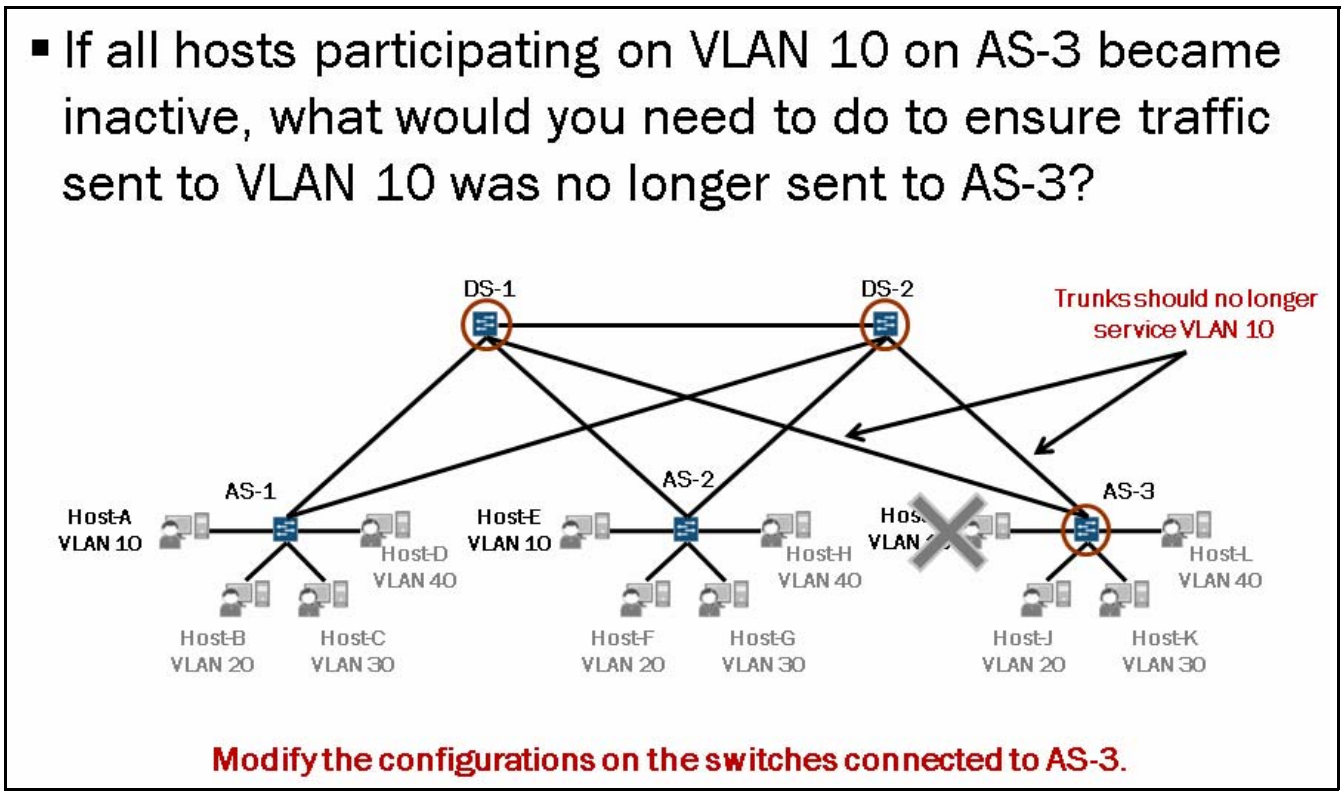


All access and distribution switches will receive this traffic.

This graphic and the next are designed to test your understanding of basic bridging operations in an environment with multiple VLANs. As the slide indicates, all switches are configured to support all VLANs on their respective trunk ports (the ports interconnecting the switches). Because of this configuration, all broadcast and unknown unicast traffic sourced and destined

within a given VLAN should be flooded throughout the entire Layer 2 network passing through all access and distribution switches.

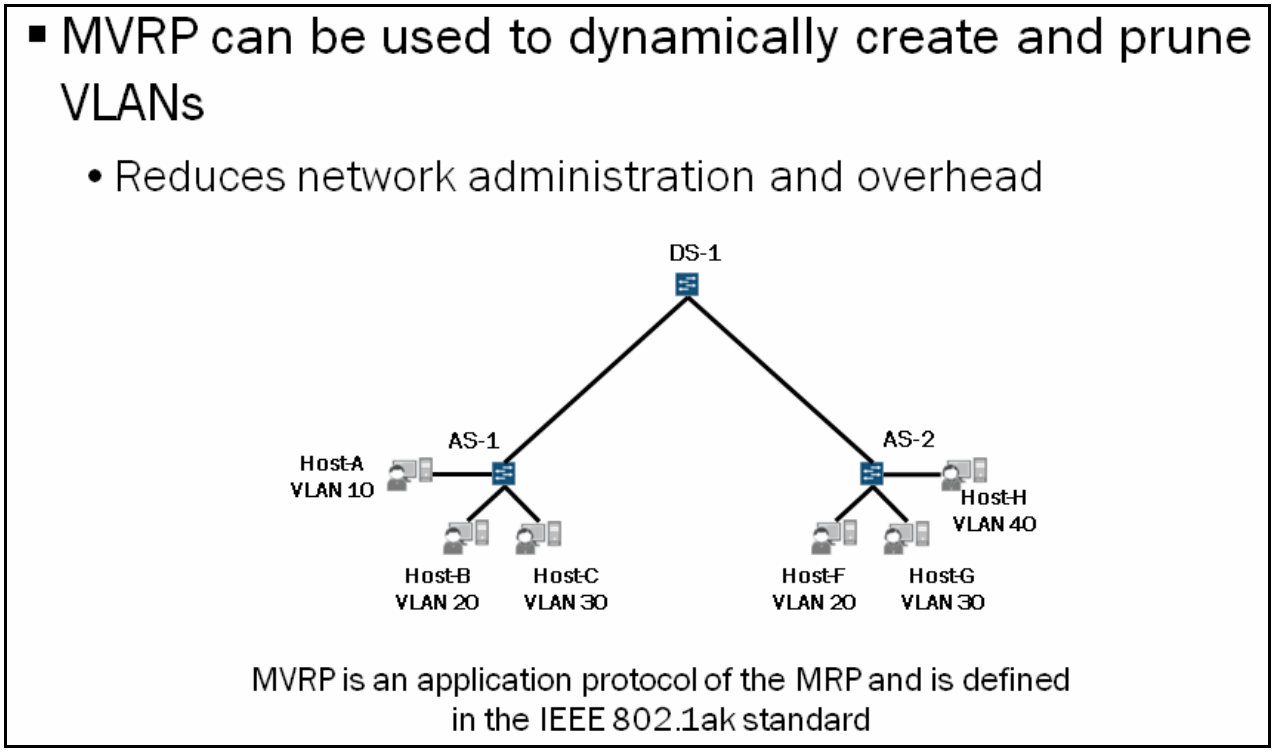
Test Your Knowledge: Part 2



The scenario illustrated in this graphic builds on the details covered on the previous slide. In this example, the end-user device named Host-I, which is connected to the AS-3 switch, is no longer active (meaning that AS-3 no longer has any active access ports for VLAN 10). Even though AS-3 no longer has active end-user devices participating in VLAN 10, it will still receive all broadcast and unknown unicast traffic associated with VLAN 10 because of the current configurations on the connected switches.

To stop this unwanted traffic from being flooded on to AS-3, you must modify the configurations on the connected distribution switches (DS-1 and DS-2) so that their trunk ports, which connect to AS-3, no longer service VLAN 10.

Introducing MVRP

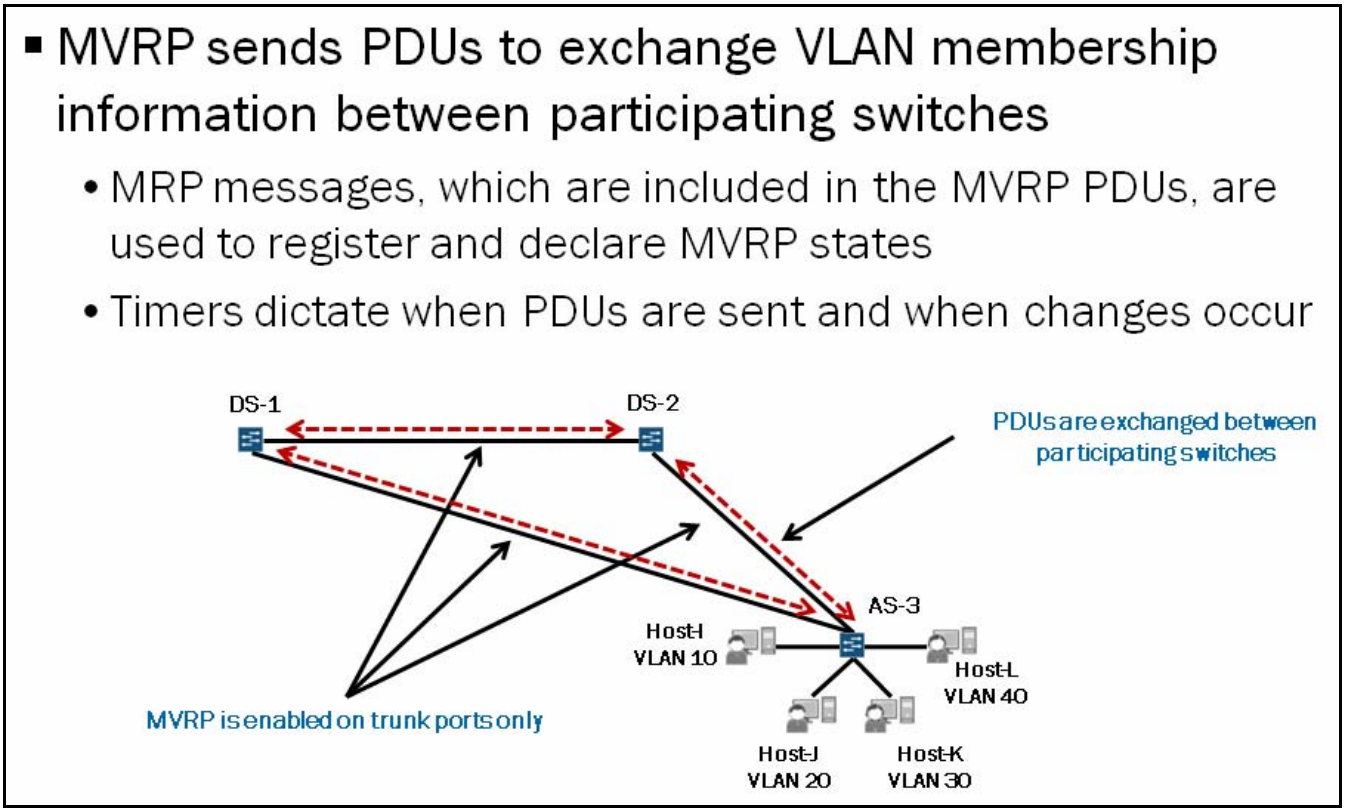


To simplify VLAN management, you can enable MVRP on your EX Series Ethernet Switches. MVRP dynamically manages VLAN registration in a LAN. MVRP helps reduce administration and network overhead by dynamically pruning VLAN information when a switch no longer has active access ports for a configured VLAN. In addition to the pruning functionality, MVRP can also be used to dynamically create VLANs in switching networks.

MVRP is an application protocol of the Multiple Registration Protocol (MRP) and is defined in the IEEE 802.1ak standard. MRP and MVRP were designed by IEEE to perform the same functions as Generic Attribute Registration Protocol (GARP) and GARP VLAN Registration Protocol (GVRP). MRP and MVRP overcome some GARP and GVRP limitations, in particular limitations involving bandwidth usage and convergence time in large networks with large numbers of VLANs.

MVRP was created by IEEE as a replacement application for GVRP. MX Series switches support MVRP. We do not cover GVRP in this study guide.

Exchanging VLAN Membership Information



MVRP uses protocol data units (PDUs) to send VLAN registration information which includes the current VLAN membership details of the sending switch. The VLAN membership information is used to communicate which switches are members of which VLANs and which switch interfaces are in which VLAN. MVRP shares all information in the PDU with all switches participating in MVRP in the switching network.

MVRP stays synchronized using these PDUs. The MVRP PDUs are sent to other switches on the network only when an MVRP state change occurs. Switches participating in MVRP receive these PDUs during state changes and update their MVRP states accordingly. MVRP timers dictate when PDUs can be sent and when switches receiving MVRP PDUs can update their MVRP information.

MVRP registration and updates are controlled by timers that are part of the MRP protocol. These timers are set on a per-interface basis and define when MVRP PDUs can be sent and when MVRP information can be updated on a switch. The following timers are used to control MVRP operations:

- `Join`: Controls the interval for the next MVRP PDU transmit opportunity.
- `Leave`: Controls the period of time that an interface on the switch waits in the Leave state before changing to the unregistered state.
- `LeaveAll`: Controls the frequency with which the interface generates LeaveAll messages.

VLAN information is distributed as part of the MVRP message exchange process and can be used to dynamically create VLANs, which are VLANs created on one switch and propagated to other switches as part of the MVRP message exchange process. Dynamic VLAN creation using MVRP is enabled by default but can be disabled.

MVRP uses MRP messages to register and declare MVRP states for a switch and to inform the switching network of state changes. These messages are included in the PDUs and communicate state information to the other switches in the network. The following messages are communicated for MVRP:

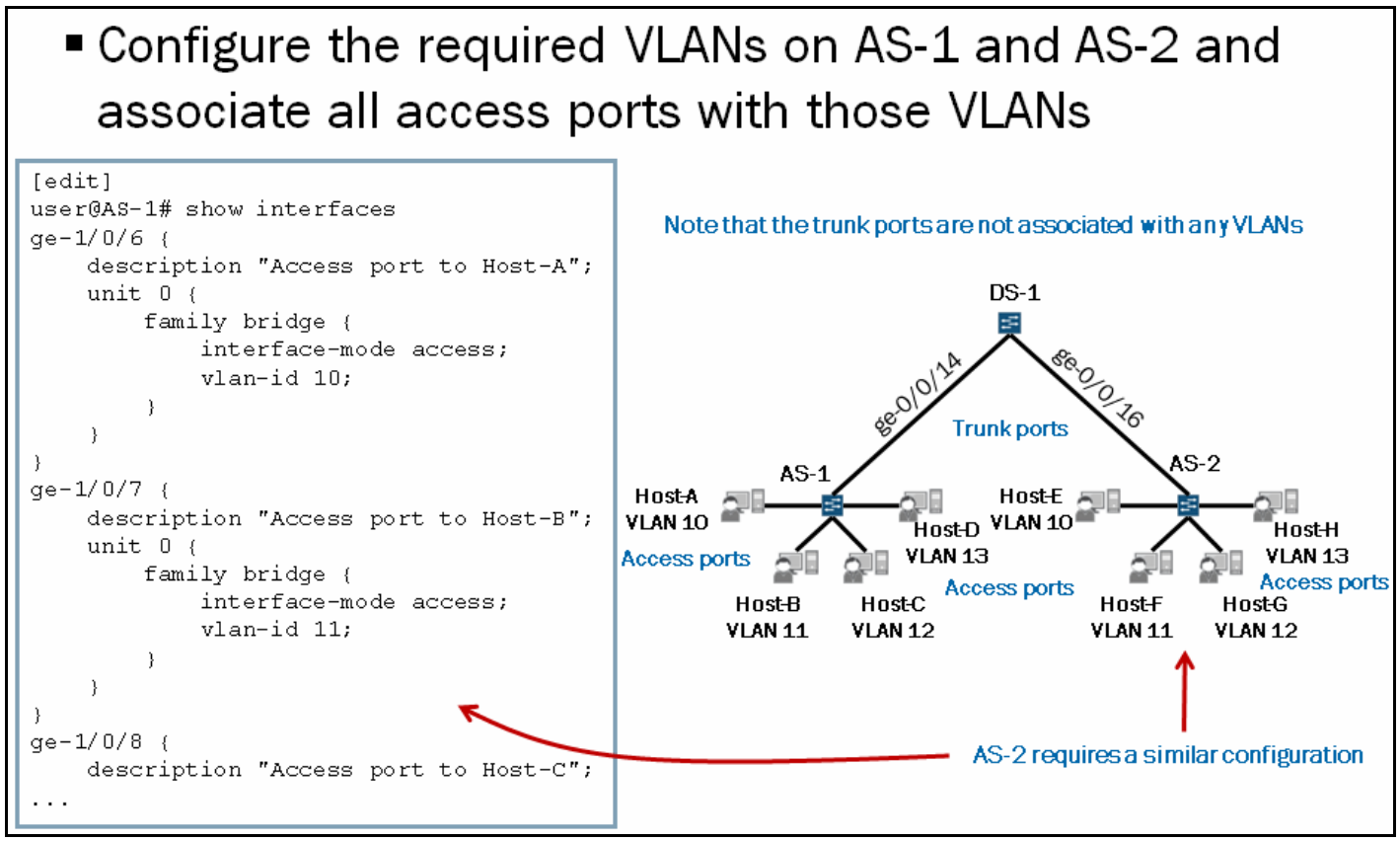
- `Empty`: VLAN information is not being declared and is not registered.
- `In`: VLAN information is not being declared but is registered.
- `JoinEmpty`: VLAN information is being declared but not registered.
- `JoinIn`: VLAN information is being declared and is registered.
- `Leave`: VLAN information that was previously registered is being withdrawn.

- `LeaveAll`: All registrations will be de-registered. Participants that want to participate in MVRP must re-register.
- New: VLAN information is new and possibly not previously registered.

To ensure VLAN membership information is current, MVRP uses the MRP messages to remove switches and interfaces that are no longer available from the VLAN information. Pruning VLAN information limits the network VLAN configuration to active participants only, reducing network overhead. Pruning VLAN information also targets the scope of broadcast, unicast with unknown destination, and multicast (BUM) traffic to interested devices only.

MVRP is disabled by default on all MX Series devices. You can configure MVRP on MX Series device interfaces to participate in MVRP for the switching network. MVRP can only be enabled on trunk interfaces, and dynamic VLAN configuration through MVRP is enabled by default when MVRP is enabled. We cover MVRP configuration on a subsequent slide. Note that MVRP does not support all spanning tree protocols. Currently, MVRP does not support the VLAN Spanning Tree Protocol (VSTP).

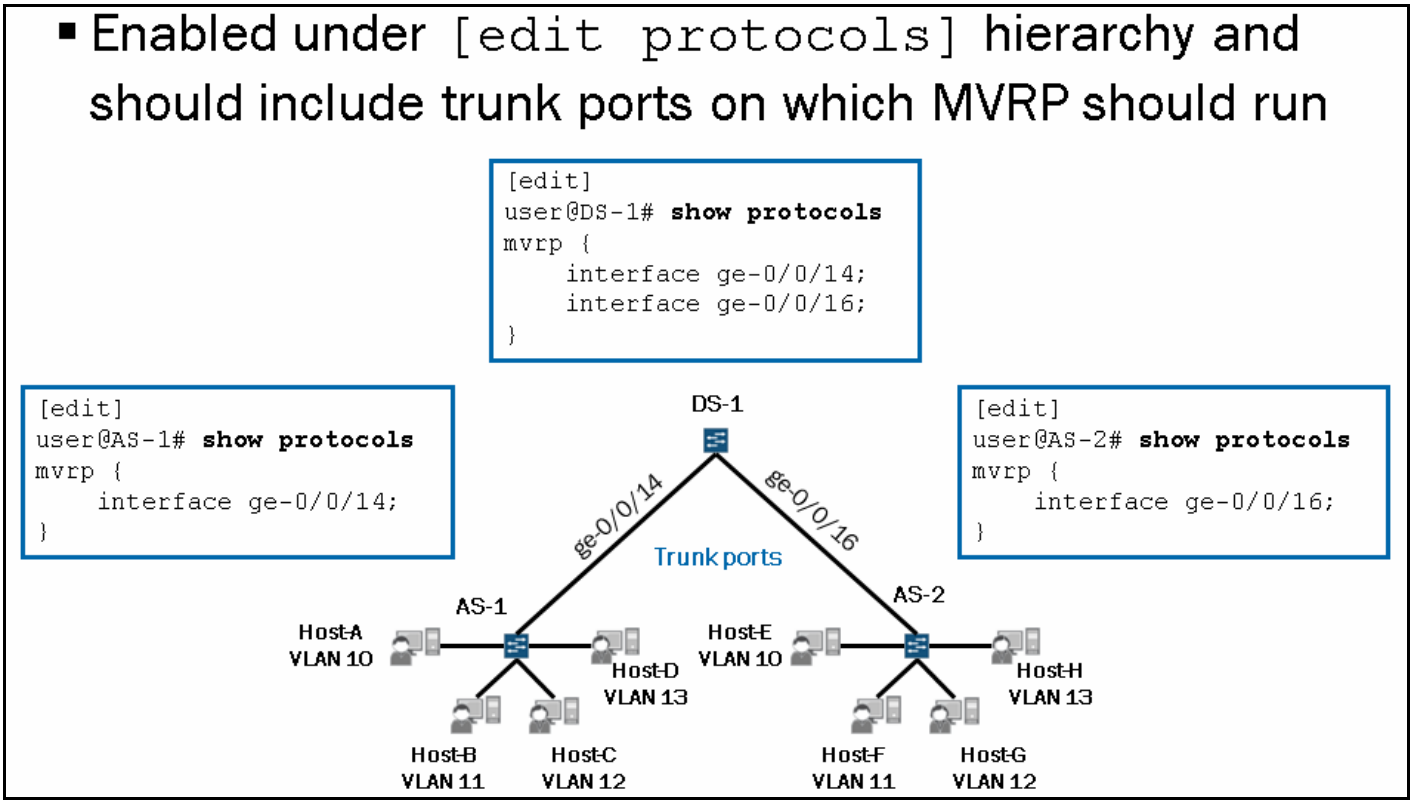
A Starting Point



When implementing MVRP, you should ensure that all required VLANs are configured on the access switches and that the access ports are associated with their respective VLANs. We illustrate a basic starting point configuration for the AS-1 switch on the slide. Note that the sample configuration is trimmed for brevity and that the AS-2 switch requires a similar configuration.

Also worth noting is that none of the trunk ports, on any of the participating switches, should be associated with the configured VLANs. The trunk ports must still be configured under the `[edit interfaces]` hierarchy level as trunk ports, but they will not be manually associated with VLANs. MVRP will make the needed associations once it is enabled.

Enabling MVRP



This graphic illustrates the required configuration used to enable MVRP. Note that MVRP is only enabled on the trunk ports of all participating switches. Once MVRP is enabled, dynamic VLAN configuration information will be shared and created on participating switches. You can disable dynamic VLAN configuration using the following **no-dynamic-vlan** statement:

```
[edit protocols]
user@AS-1# show
mvrp {
    no-dynamic-vlan;
    interface ge-0/0/14.0;
}
```

Remember that MVRP registration and updates are controlled by timers, which are part of MRP. These timers are set on a per-interface basis and define when MVRP PDUs can be sent and when MVRP information can be updated. If needed, you can adjust the timers as shown here:

```
[edit protocols]
user@AS-1# set mvrp interface ge-0/0/14 ?
Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  join-timer           Join timer interval (100..500 milliseconds)
  leave-timer          Leave timer interval (300..1000 milliseconds)
  leaveall-timer       Leaveall timer interval (10..60 seconds)
  point-to-point       Port is point to point
  registration         Registration mode
  |                    Pipe through a command
```

The default MVRP timer values are 200 ms for the join timer, 800 ms for the leave timer, and 10,000 ms for the leaveall timer. Unless there is a compelling reason to make a change, we recommend you use the default timer settings. Modifying timers to inappropriate values might cause an imbalance in MVRP operations.

Monitoring MVRP

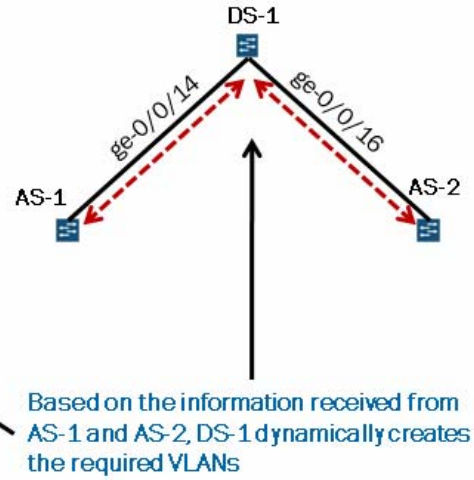
- Use **show mvrp** to monitor MVRP status and message and timer information for each interface

```
user@DS-1> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-1/1/14        200     800     10000
  ge-1/1/16        200     800     10000
```

This graphic and the two that follow highlight some key monitoring commands used when verifying MVRP operations. This slide illustrates the use of the **show mvrp** command, which is used to monitor MVRP status along with message and timer information on a per-interface basis.

- Use **show mvrp dynamic-vlan-memberships** to view dynamic VLAN membership

```
user@DS-1> show mvrp dynamic-vlan-memberships
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

    VLAN Id          Interfaces
         10          ge-0/0/14
                     ge-0/0/16
         11          ge-0/0/14
                     ge-0/0/16
         12          ge-0/0/14
                     ge-0/0/16
         13          ge-0/0/14
                     ge-0/0/16
```



This graphic illustrates the **show mvrp dynamic-vlan-memberships** command, which is used to view dynamic VLAN membership information.

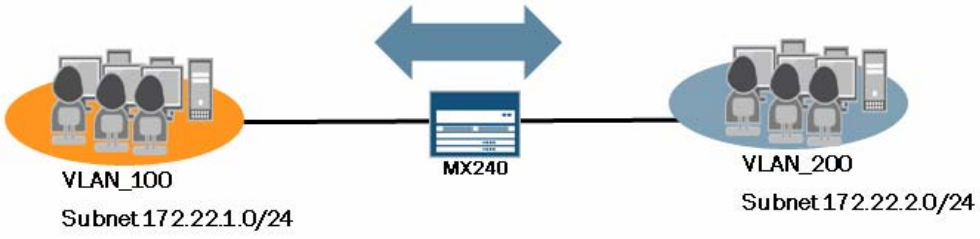
- Use **show mvrp statistics** to view MVRP statistics on a per-interface basis

```
user@DS-1> show mvrp statistics
MVRP statistics

Interface name                  : ge-1/1/14
VLAN IDs registered             : 1
Sent MVRP PDUs                  : 69
Received MVRP PDUs without error: 36
Received MVRP PDUs with error   : 0

Interface name                  : ge-1/1/16
VLAN IDs registered             : 2
Sent MVRP PDUs                  : 35
Received MVRP PDUs without error: 35
Received MVRP PDUs with error   : 0
...
```

This graphic illustrates the **show mvrp statistics** command, which is used to view MVRP statistics on a per-interface basis.

IRB Allows for Bridging and Routing

- IRB allows for both Layer 2 bridging and Layer 3 routing in a bridge domain
  - An IRB interface is an IP gateway for the hosts of a bridge domain



If you use a Layer 2-only Ethernet switch (no Layer 3 functionality), then you must add a separate router to your environment to provide routing between the subnets shown on the slide. However, an MX Series router can act as both a Layer 2 Ethernet switch and a router at the same time. An IRB interface is a logical Layer 3 interface used as an IP gateway for a VLAN. The following sections provide configuration and monitoring examples for an IRB interface.

IRB Configuration Example

```
[edit]
user@switch# show interfaces
ge-1/0/0 {
    unit 0 {
        family bridge {
            interface-mode access;
            vlan-id 100;
        }
    }
}
…
irb {
    unit 0 {
        description "GW for VLAN_100";
        family inet {
            address 172.22.1.254/24;
        }
    }
    unit 1 {
        description "GW for VLAN_200";
        family inet {
            address 172.22.2.254/24;
        }
    }
} …
```

```
[edit]
user@switch# show bridge-domains
vlan_100 {
    vlan-id 100;
    routing-interface irb.0;
}
vlan_200 {
    vlan-id 200;
    routing-interface irb.1;
}
```

This example facilitates routing between the vlan_100 and vlan_200 bridge domains.

The graphic provides a configuration example for an IRB interface. In this example, the switch performs a Layer 3 lookup when it receives traffic with a destination MAC address that matches its own MAC address. For the switch to perform this routing operation, the attached devices must have configured gateway addresses that match the IP address associated with the corresponding IRB interface.

Monitoring IRB

- Use the **show interfaces** command to verify the status of an IRB interface

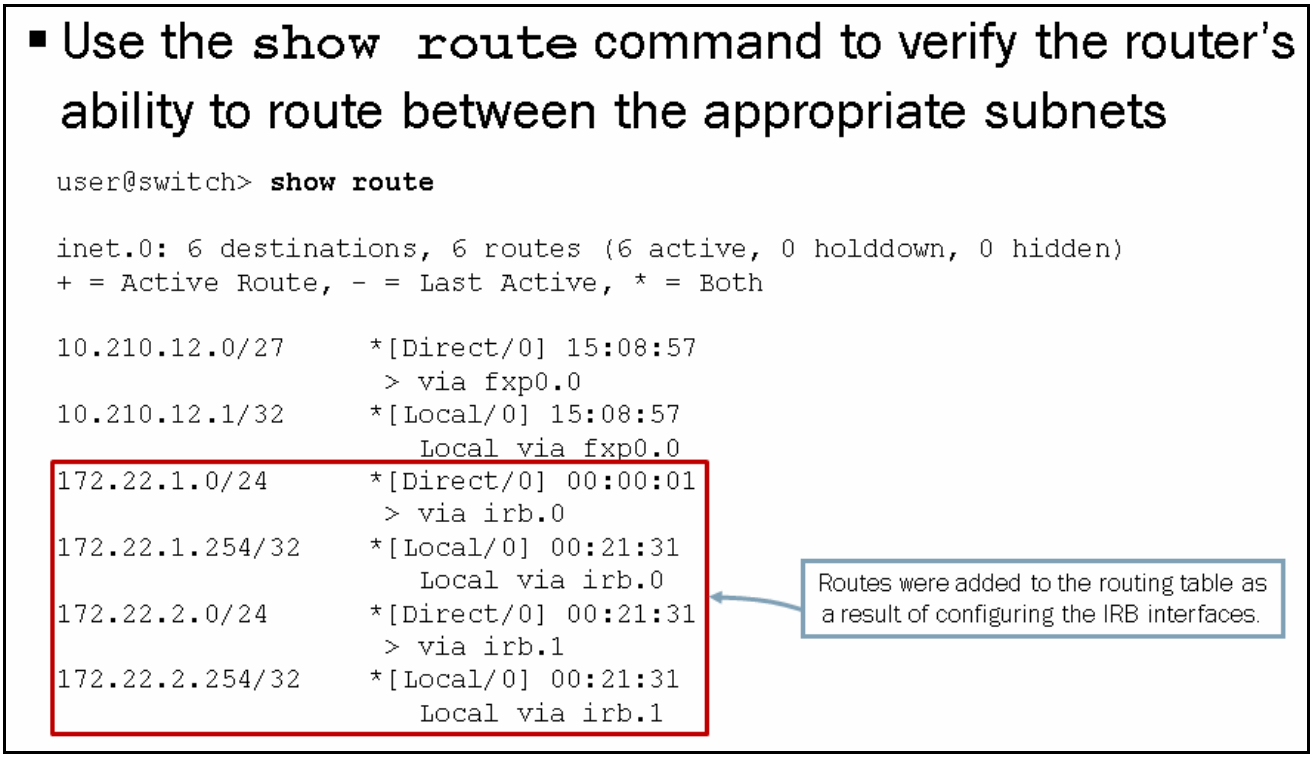
```
user@switch> show interfaces terse irb*
Interface               Admin Link Proto    Local                 Remote
irb                     up    up
irb.0                   up    up   inet     172.22.1.254/24

irb.1                   up    up   inet     172.22.2.254/24
```

At least one port must be active for IRB state to be up.

IRB state and IP address details

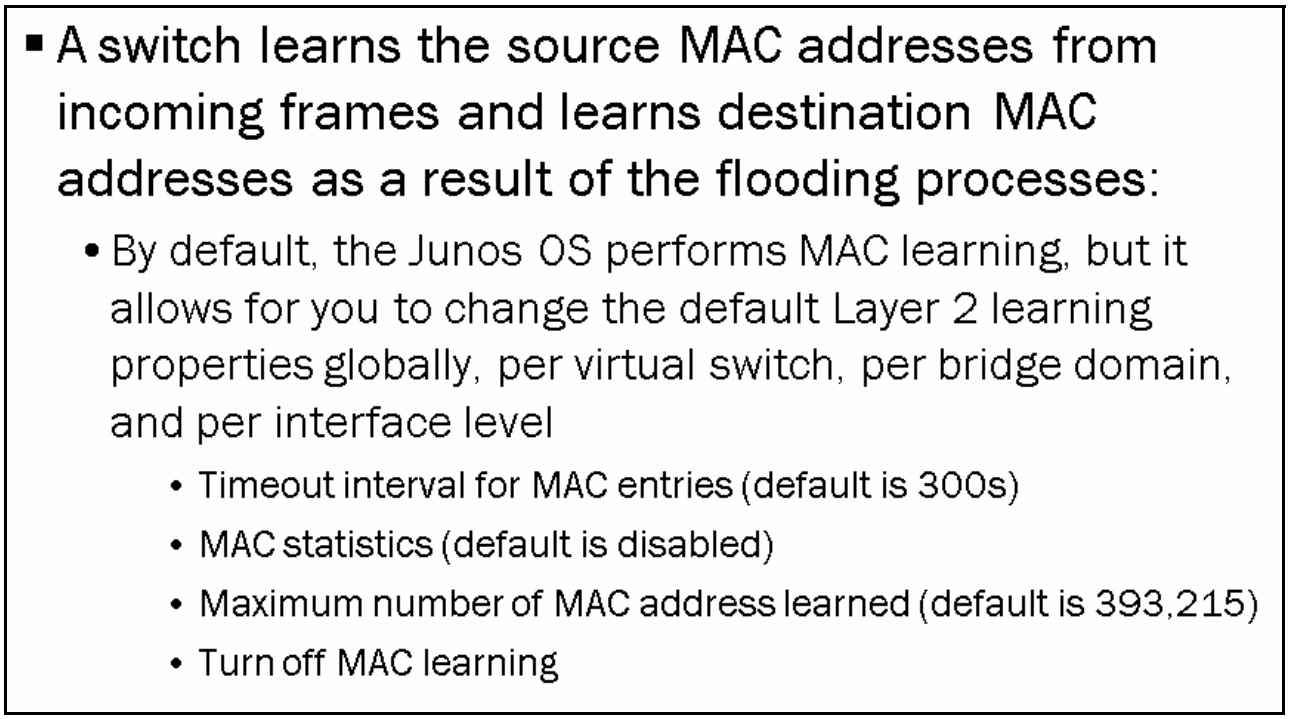
The graphic lists a key command used to monitor an IRB interface, and shows the output from the **show interfaces terse** command. This command shows the state and IP address information for an IRB interface. As indicated on the graphic, at least one active port must associate with the bridge domain for the IRB interface to be administratively up.

Verifying Routing



As with any router, when you configure an IP address for an interface on that router, routes are automatically added to the routing table. In the Junos OS, for each configured IP interface, two routes are added to the routing table. One route is a host route (32-bit mask) used to forward traffic to the Routing Engine (RE) when locally destined packets arrive. The other route is a route to the network subnet to which that interface belongs. This route allows the router to route packets to other hosts on that same subnet. The graphic shows that four routes were added to the inet.0 table as a result of configuring two IRB interfaces.

MAC Address Learning and Forwarding



As discussed previously, a switch learns MAC addresses from incoming frames as well as from the flooding process. The Junos OS allows you to override the default MAC learning behavior. The graphic lists the items you can change as well as where the changes can apply to a switch.

The following list provides configurable values for each of the MAC learning properties:

- MAC timeout interval: 10s–1,000,000s (300s is the default);
- MAC statistics: Can be enabled (disabled by default);
- Global MAC limit: 20–1,048,575 (393215 is the default);
- Switch MAC limit: 16–1,048,575 (5120 is the default);
- Bridge domain MAC limit: 16–1,048,575 (5120 is the default); and
- Interface MAC limit: 1–131,071 (1024 is the default).

To view MAC statistics once you enable the feature, issue the **show bridge mac-table extensive** command.

Global Level and Switch Level Settings

```
[edit protocols l2-learning]
user@switch# set ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> global-mac-limit     System level MAC limit options
  global-mac-statistics  Enable MAC address statistics at system level
  global-mac-table-aging-time  System level MAC table aging time (10..1000000
seconds)
  global-no-mac-learning  Disable dynamic MAC address learning at system
level
```

Global level settings apply to all virtual switches (discussed in a later chapter) and all bridge domains.

Switch Level Settings

```
[edit switch-options]
user@switch# set ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> interface            Interface for configuring bridge-options
> interface-mac-limit  Maximum MAC address learned per interface
  mac-statistics       Enable MAC address statistics
> mac-table-size       Size of MAC address forwarding table
  no-mac-learning      Disable dynamic MAC address learning
```

Switch level settings apply to all bridge domains associated with a virtual switch.

Bridge Domain Level Settings

```
[edit bridge-domains bd]
user@switch# set bridge-options ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> interface            Interface that connect this site to the VPN
> interface-mac-limit  Maximum MAC address learned per interface
  mac-statistics       Enable MAC address statistics
> mac-table-size       Size of MAC address forwarding table
  no-mac-learning      Disable dynamic MAC address learning
```

Settings at this level affect all interfaces associated with the bridge domain.

Interface Level

```
[edit bridge-domains bd]
user@switch# set bridge-options interface ge-1/0/0.0 ?
Possible completions:
  <[Enter]>            Execute this command
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> interface-mac-limit  Maximum number of MAC addresses learned on the
interface
  no-mac-learning      Disable dynamic MAC address learning
> static-mac           Static MAC addresses assigned to this interface
```

Settings at this level affect only the interface specified in the configuration.

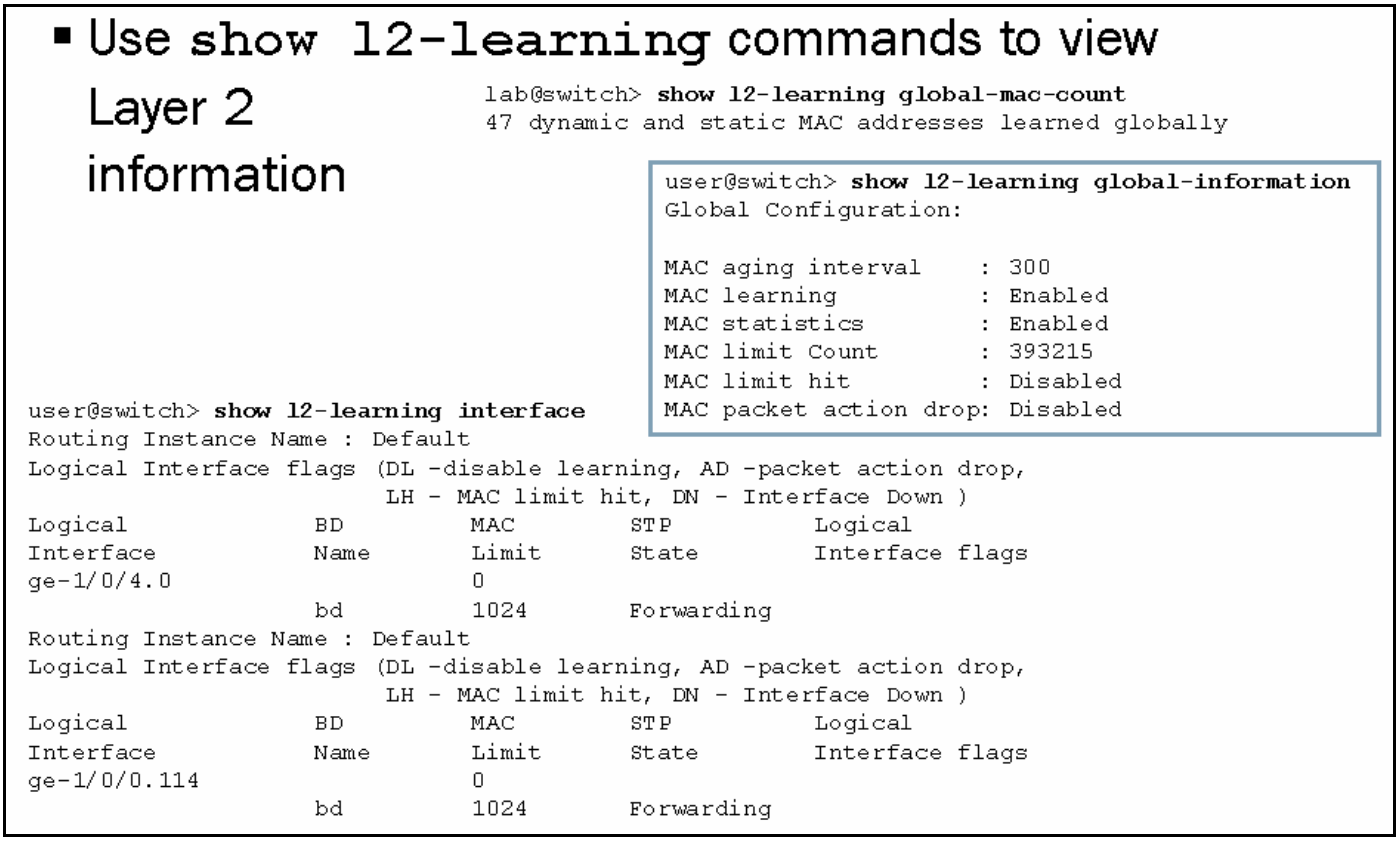
Layer 2 Learning Example

- Specify the **mac-table-size** ***number*** option to limit the number of learned MAC addresses for the bridge domain:
  - The default action when the limit is reached is to flood any frames with unknown MAC addresses
    - Optionally, you can specify **packet-action drop** to discard frames with unknown MAC addresses when the MAC table is full

```
[edit bridge-domains bd]
user@switch# show
vlan-id 100;
bridge-options {
    mac-table-size {
        4000;
        packet-action drop;
    }
}
```

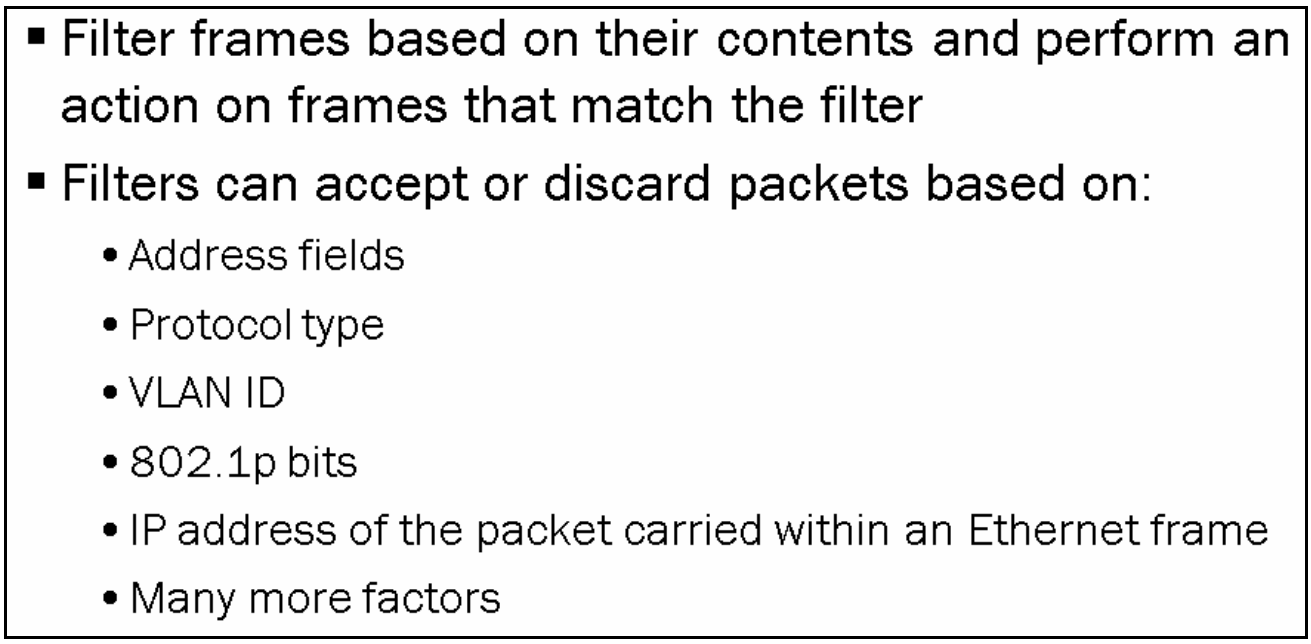
The example in the graphic shows that the MAC table size limit for the bridge domain changed from the default of 5120 to 4000. By default, when the bridge domain MAC learning limit is reached, the device does not learn any more MAC addresses but still forwards or floods traffic in the case of unknown destinations. The graphic shows that this default behavior was overridden so that Ethernet frames with unknown destinations will drop when the configured limit is reached.

Layer 2 Learning and Forwarding Status



The graphic shows some of the commands that you can use to view the Layer 2 learning and forwarding status for the switch.

Layer 2 Firewall Filters



You can use filters to control the frames destined to the RE as well as control frames passing through the router.

You can define input filters that affect only inbound traffic and output filters that affect only outbound traffic. Filters can accept or discard frames based on the contents of the frame's address fields, protocol type, VLAN ID, or even the 802.1p bit field in the frame header.

Syntax

A Junos OS filter consists of one or more named terms, similar to a policy statement. Each `term` has a set of match conditions preceded by the keyword `from`, and a set of actions or action modifiers preceded by the keyword `then`.

```
[edit firewall family bridge]
filter filter-name {
  term term-name {
      from {
         match-conditions;
      }
     then {
       action;
       action-modifiers;
  term implicit-rule {
     then discard;
     }
   }
}
```

Hierarchy Level

Layer 2 firewall filters are defined under the `[edit firewall family bridge]` section of the configuration hierarchy.

One or More Terms

Firewall filter terms (at least one term is necessary) are processed sequentially. If no `from` condition is present, then all frames match. If no frames match any term, the default action is to discard the frame silently! Take care to ensure that wanted frames are not discarded. Use the command-line interface (CLI) **`insert`**, **`copy`**, and **`rename`** functions to assist in the management of your multiterm firewall filters.

Actions and Modifiers

A filter can accept a frame for normal forwarding or discard a frame silently. You can modify these actions by applying a modifier. For example, you can apply the **`count`** modifier to increment a counter. We discuss other modifiers on the following graphics.

Applying Layer 2 Firewall Filters

- You can apply Layer 2 firewall filters to either an individual interface, a bridge domain, or to both
  - Interface level
    - You can apply a single filter for each interface (input, output, or both)
    - Apply a chain of filters using the **`input-list`** or **`output-list`** statements
  - Bridge domain level
    - You can apply a single filter for each bridge domain (input only)
  - Interface and bridge domain at the same time (input only)
    - The interface filter is processed first, followed by the bridge domain filter

```
[edit]
user@switch# show interfaces ge-1/0/0
unit 0 {
    family bridge {
        filter {
            input example;
        }
        interface-mode access;
        …
```

```
[edit]
user@switch# show bridge-domains
vlan_100
vlan-id 100;
routing-interface irb.0;
forwarding-options {
    filter {
        input example;
…
```

Once you configure a firewall filter, you must apply it to one or more interfaces. You can accomplish this task in several different ways. The easiest way to apply a firewall filter to an individual Layer 2 interface is to specify the filter as an input or output filter at the `[edit interface interface interface-name unit number family bridge filter]` level of the configuration hierarchy. To apply a filter to all interfaces that belong to a particular bridge domain, you can apply a firewall filter at the `[edit bridge-domain name forwarding-options filter]` level of the configuration hierarchy. If firewall

filters are applied as input filters to both the interface and bridge-domain levels, the Junos OS logically concatenates the bridge-domain-level filter to the end of the interface-level filter.

Note that you cannot use bridge-domain-level filters when the **vlan-id-list** statement was used to create the bridge domain.

Single Terms

When a firewall filter consists of a single term, the filter is evaluated as follows: if the frame matches all the conditions, the device takes the action in the `then` statement; if the frame does not match all the conditions, the device discards it.

Multiple Terms

When a firewall filter consists of more than one term, the filter is evaluated sequentially. First, the frame is evaluated against the conditions in the `from` statement in the first term. If the frame matches, the device takes the action in the `then` statement. If it does not match, it is evaluated against the conditions in the `from` statement in the second term. This process continues until either the frame matches the `from` condition in one of the subsequent terms or until no more terms remain.

- If a frame passes through all the terms in the filter without matching any of them, the device discards it.
- If a term does not contain a `from` statement, the frame is considered to match, and the device takes the action in the term's `then` statement.
- If a term does not contain a `then` statement, or if you do not configure an action in the `then` statement (that is, the frame is just counted), and if the frame matches the conditions in the term's `from` statement, the device accepts the frame.

Filter Lists

Instead of applying a single filter to an interface using filter input or filter output, you can apply a list of up to 16 filters. You perform this action with the **input-list** and **output-list** keywords.

Match Conditions: Part 1

```
[edit]
user@switch# set firewall family bridge filter example term 10 from ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> destination-mac-address  Destination MAC address
+ destination-port     Match TCP/UDP destination port
+ destination-port-except  Do not match TCP/UDP destination port
+ dscp                 Match Differentiated Services (DiffServ) code point
+ dscp-except          Do not match Differentiated Services (DiffServ) code point
+ ether-type           Match Ethernet type
+ ether-type-except    Do not match Ethernet type
+ forwarding-class     Match forwarding class
+ forwarding-class-except  Do not match forwarding class
```



```
+ icmp-code              Match ICMP message code
+ icmp-code-except       Do not match ICMP message code
+ icmp-type              Match ICMP message type
+ icmp-type-except       Do not match ICMP message type
+ interface-group        Match interface group
+ interface-group-except  Do not match interface group
> ip-address            Match IP source or destination address
> ip-destination-address  Match IP destination address
+ ip-precedence          Match IP precedence value
+ ip-precedence-except   Do not match IP precedence value
+ ip-protocol            Match IP protocol type
+ ip-protocol-except     Do not match IP protocol type
> ip-source-address      Match IP source address
```

```
…
+ isid                  Match Internet Service ID
+ isid-dei              Match Internet Service ID DEI bit
+ isid-dei-except       Do not match Internet Service ID DEI bit
+ isid-except           Do not match Internet Service ID
+ isid-priority-code-point  Match Internet Service ID Priority Code Point
+ isid-priority-code-point-except  Do not match Internet Service ID Priority Code Point
+ learn-vlan-1p-priority  Match Learned 802.1p VLAN Priority
+ learn-vlan-1p-priority-except  Do not match Learned 802.1p VLAN Priority
+ learn-vlan-dei        Match User VLAN ID DEI bit
+ learn-vlan-dei-except  Do not match User VLAN ID DEI bit
+ learn-vlan-id         Match Learnt VLAN ID
+ learn-vlan-id-except  Do not match Learnt VLAN ID
```

```
+ loss-priority         Match Loss Priority
+ loss-priority-except   Do not match Loss Priority
+ port                  Match TCP/UDP source or destination port
+ port-except           Do not match TCP/UDP source or destination port
> source-mac-address    Source MAC address
+ source-port           Match TCP/UDP source port
+ source-port-except    Do not match TCP/UDP source port
  tcp-flags             Match TCP flags
+ traffic-type          Match Match traffic type
+ traffic-type-except   Do not match Match traffic type
+ user-vlan-1p-priority  Match User 802.1p VLAN Priority
+ user-vlan-1p-priority-except  Do not match User 802.1p VLAN Priority
+ user-vlan-id          Match User VLAN ID
+ user-vlan-id-except   Do not match User VLAN ID
+ vlan-ether-type       Match VLAN Ethernet type
+ vlan-ether-type-except  Do not match VLAN Ethernet type
```

The graphics show some of the many match conditions that you can use in a Layer 2 firewall filter.

Match Actions

```
user@switch# set firewall family bridge filter example term 10 then ?
Possible completions:
  accept               Accept the packet
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  count                Count the packet in the named counter
  discard              Discard the packet
  forwarding-class     Classify packet to forwarding class
  loss-priority        Packet's loss priority
  next                 Continue to next term in a filter
  next-hop-group       Use specified next-hop group
  policer              Name of policer to use to rate-limit traffic
  port-mirror          Port-mirror the packet
  port-mirror-instance  Port-mirror the packet to specified instance
> three-color-policer  Police the packet using a three-color-policer
```

You can apply the actions `accept` and `discard` to a frame. However, you can apply modifiers to the frames as well:

- `count`: This modifier counts the number of matches that occur to a named counter. See the current totals by issuing the **`show firewall`** command.
- `forwarding-class`: This modifier is used for multifield classification for class of service (CoS). Essentially, this setting specifies the queue in which this frame should be placed.
- `loss-priority`: This modifier allows you to change the packet loss priority bit of the IP packet in the payload of the Ethernet frame.
- `next`: This modifier allows the frame to be evaluated by the next term in the filter.
- `next-hop-group`: This modifier specifies which next-hop group will be applied.
- `policer`: This modifier applies a rate-limiting policer to the matching frames.
- `port-mirror`: This modifier allows copies of the frame to be sent to an outbound interface for analysis. The original frame forwards as normal.

Example Filter

```
[edit]
user@switch# show firewall
family bridge {
    filter example {
        term 10 {
            from {
                learn-vlan-id 100;
            }
            then {
                count learn-vlan-100;
                accept;
            }
        }
        term 1000 {
            then {
                count all-others;
                accept;
            }
        }
    }
}
```

```
user@switch# show bridge-domains
vlan_100 {
    vlan-id 100;
    routing-interface irb.0;
    forwarding-options {
        filter {
            input example;
        }
    }
}
```

```
user@switch> show firewall

Filter: __default_bpdu_filter__

Filter: example
Counters:
Name                        Bytes              Packets
all-others                      0                    0
learn-vlan-100              56300                  533
```

The graphic shows an example of configuring, applying, and viewing the effects of a firewall filter. To clear the counters, use the **`clear firewall`** command.

Review Questions

1. What is the purpose of a bridge domain on an MX Series router?
2. How does a bridge handle multicast Ethernet frames?
3. What is the purpose of an IRB interface?
4. Which match condition is used in a Layer 2 firewall filter to match on 802.1p priority bits?

Answers

1.

A bridge domain allows you to specify which VLANs will be used for Layer 2 switching.

2.

A bridge generally forwards multicast frames out of every interface except for the one from which they were received.

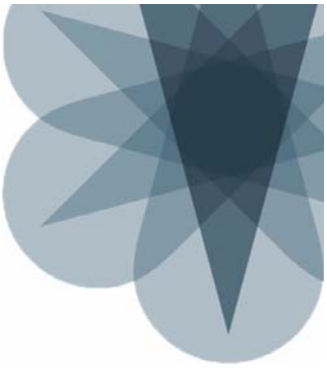
3.

A IRB interface eliminates the need for an external router to route between VLANs. It acts as an IP gateway for the hosts attached to a VLAN.

4.

The match condition used in a Layer 2 firewall filter to match on 802.1p priority bits is `learn-vlan-1p-priority`.

JUNIPER NETWORKS



JNCIS-SP Study Guide—Part 2

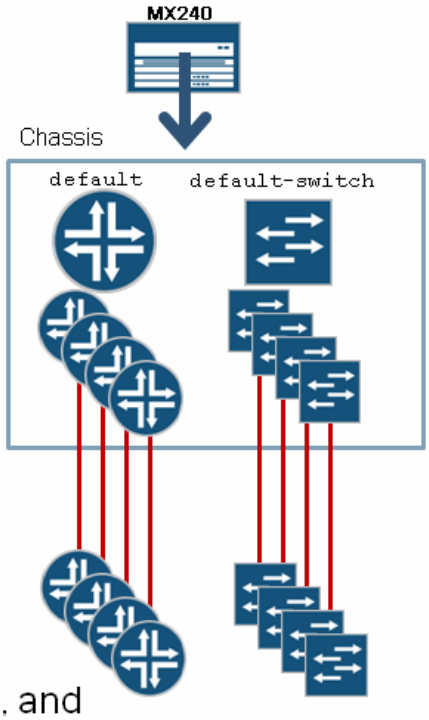
Chapter 3: Virtual Switches

This Chapter Discusses:

- The use of a routing instance;
- The function of a virtual router;
- The function of a virtual switch;
- Implementation of a virtual switch; and
- Interconnection of local routing instances.

Routing Instance Types

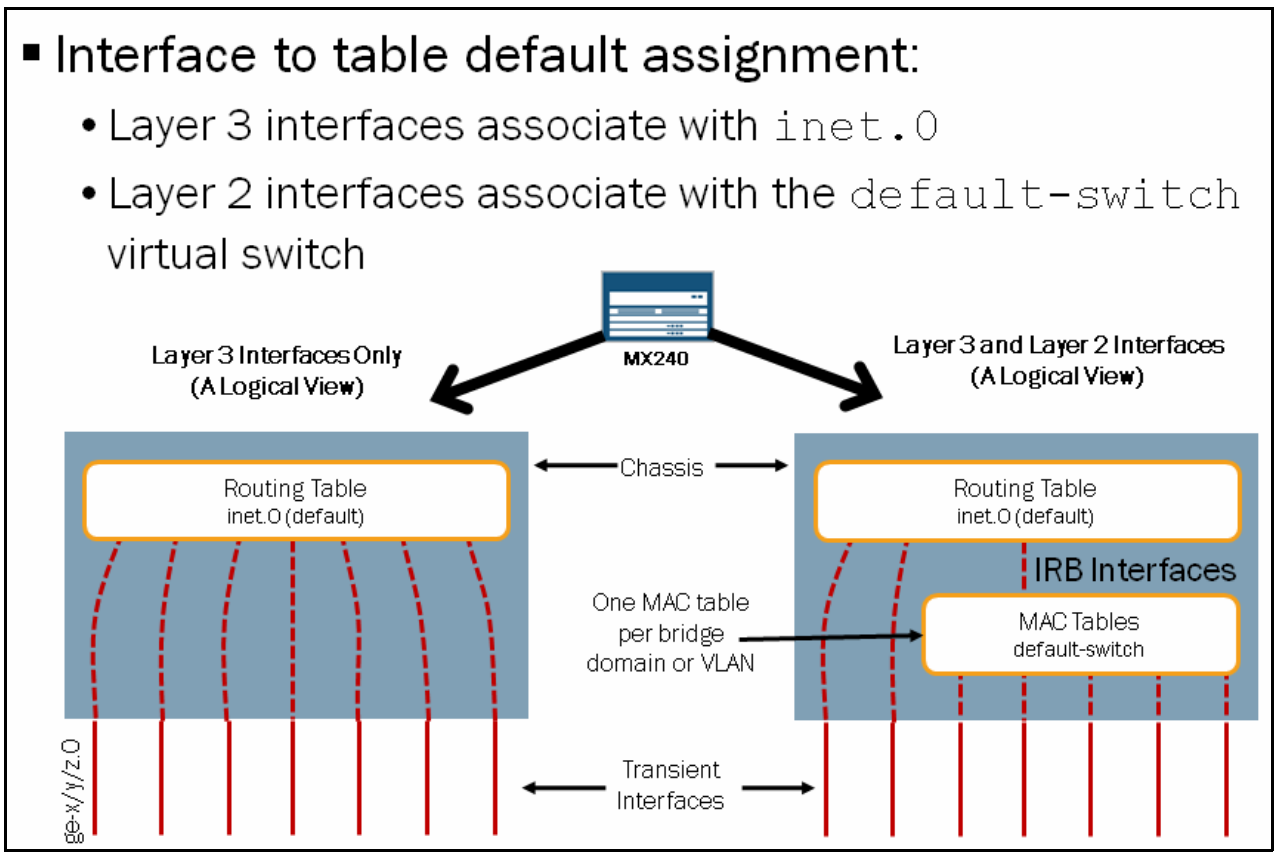
- Several different types of routing instances exist:
  - Virtual-router routing instances allow for your single chassis to appear as multiple routers to the outside world
    - Each with their own separate routing tables, protocols, link-state databases, and so on
    - Default instance is named `default`
  - Virtual-switch routing instances allow for your single chassis to appear as multiple switches to the outside world
    - Each has its own MAC tables, VLAN ID space, and spanning tree domains
    - The default instance is named `default-switch`



The Junos operating system provides several different routing-instance types with which to work. In this guide, we work with two types of routing-instances: `virtual-router` and `virtual-switch`. Essentially, these two routing-instance types allow your single chassis to appear as either more than one router or more than one switch, respectively. Each virtual router acts as a standalone router. For example, each virtual router has its own routing table, routing protocols, interfaces, and just

about everything that encompasses the typical things that comprise a router. Similarly, each configured virtual switch has its own MAC tables, virtual LAN (VLAN) ID space, bridge domains, spanning-tree domains, and so forth. A Juniper Networks MX Series 3D Universal Edge Router uses two default routing instances. For routing, it uses the `default` virtual router (inet.0 is its routing table). For switching, it uses the `default-switch` virtual switch.

Routing Instance and Interface Default Relationship

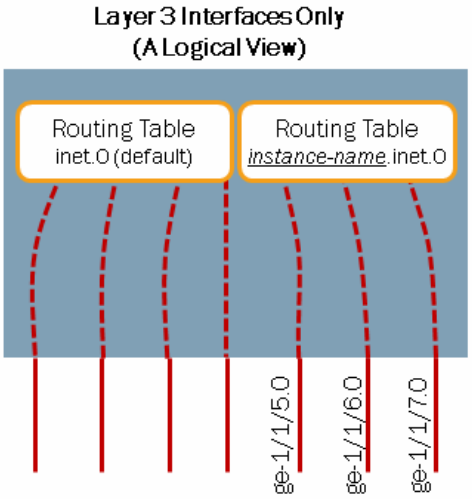


The graphic shows a very simplistic view of the default relationship of interfaces to the routing and MAC tables of an MX Series router. Keep in mind that we have left out discussion of the Packet Forwarding Engine (PFE) and the associated forwarding tables. When troubleshooting virtual routers and switches, you generally can spend your time focused on the Routing Engine's (RE's) copy of the routing and MAC tables, while trusting that equivalent copies appear as forwarding tables in the PFEs of your switch. To view the PFE forwarding tables, both for routing and switching, use the **`show route forwarding-table`** command.

In a routing-only environment, configured interfaces and their associated local and direct routes appear in the default virtual router's routing table, `inet.0`. In a mixed Layer 2 and Layer 3 environment, Layer 3 interfaces continue to work as described, whereas Layer 2 interfaces, having been associated with a bridge domain at the `[edit bridge-domains]` hierarchy, associate with the default virtual switch's MAC tables. Because IRB interfaces are Layer 3 interfaces, their associated local and direct routes appear in `inet.0` as well.

Assign Interfaces to a Virtual Router

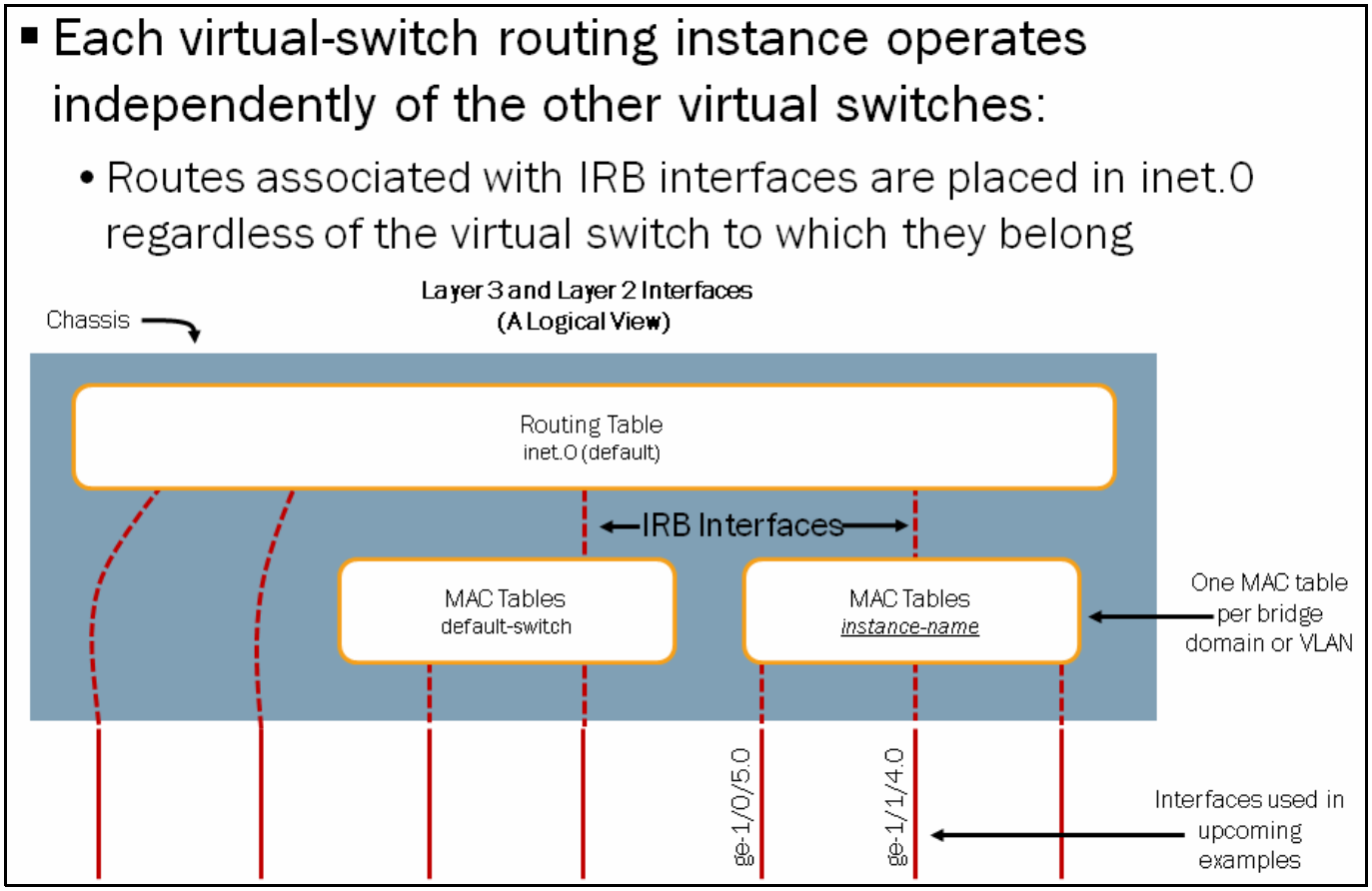
- You must assign interfaces to a virtual router
  - Place routes associated with those interfaces only in that virtual router's routing table




```
[edit]
user@switch# show routing-instances
vr1 {
    instance-type virtual-router;
    interface ge-1/1/5.0;
    interface ge-1/1/6.0;
    interface ge-1/1/7.0;
    protocols {
        ospf {
            area 0.0.0.0 {
                interface ge-1/1/5.0;
                interface ge-1/1/6.0;
                interface ge-1/1/7.0;
            }
…

user@switch> show route table vr1.inet.0

vr1.inet.0: 2 destinations, 2 routes (2 active, …
+ = Active Route, - = Last Active, * = Both

172.22.2.0/24       *[Direct/0] 20:59:24
                     > via ge-1/1/5.0
172.22.2.3/32       *[Local/0] 20:59:24
                       Local via ge-1/1/5.0
…
```

By default, once you configure an interface with properties at the `[edit interfaces` *`interface-name`* `unit` *`number`* `family inet]` level of the hierarchy, that interface's local and direct routes are placed in the inet.0 routing table. To override that behavior, you simply list the interface at the `[edit routing-instances` *`instance-name`*`]` level of the hierarchy. The local and direct routes now appear in the *`instance-name`*`.inet.0` routing table (the virtual router's routing table.)

Virtual Switch



The graphic shows the routing and MAC table relationships when using virtual switches. Each virtual switch, including the default switch, has interfaces assigned for bridging. Also, you can configure integrated routing and bridging (IRB) interfaces for each virtual switch. The local and direct routes for all IRB interfaces in all virtual switches are placed in inet.0, by default. However, you can also place them in a virtual router's routing table by listing the IRB interfaces at the `[edit routing-instances` *`instance-name`*`]` level of the hierarchy. The following sections cover the process of configuring a virtual switch.

Virtual-Switch Routing Instance

- Configure a base virtual-switch routing instance:
  - Define the bridge domains and VLAN IDs that the switch will use

```
[edit]
user@switch# show routing-instances
virtual-sw-1 {
    instance-type virtual-switch;
    bridge-domains {
        vlan_100 {
            vlan-id 100;
        }
        vlan_200 {
            vlan-id 200;
        }
    }
}
```

The configuration on the slide creates a `virtual-sw-1` routing instance and allows for VLAN IDs 100 and 200 to be used for the purpose of Layer 2 switching. MAC tables for these new bridge domains will not be used for learning and forwarding until you assign at least one interface to the virtual switch.

Virtual-Switch Access Port

- Configure an interface that acts as an access port for the virtual switch:
  - Specify the correct `vlan-id` so that this interface is associated with the correct bridge domain

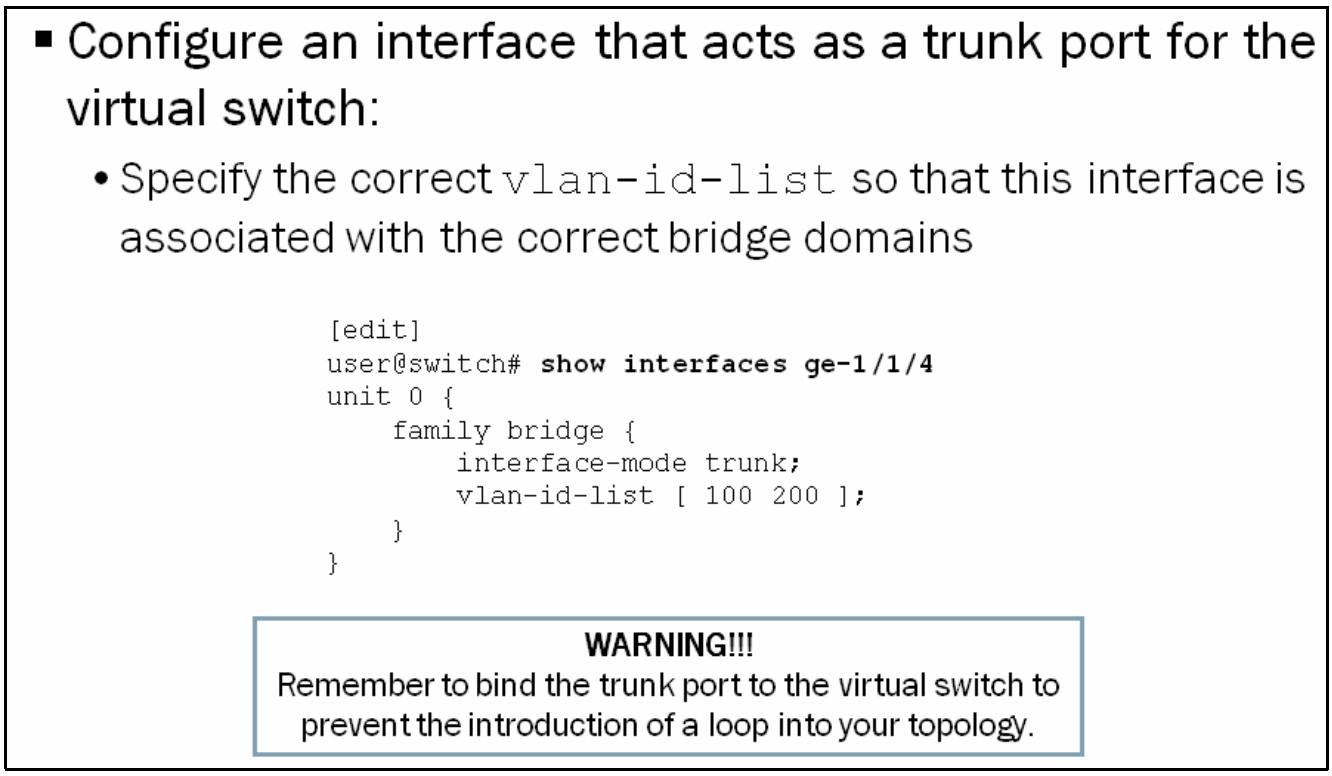
```
[edit]
user@switch# show interfaces ge-1/0/5
unit 0 {
    family bridge {
        interface-mode access;
        vlan-id 200;
    }
}
```

WARNING!!!
Committing at this point can cause this interface to be added to the `default-switch` routing instance, which could introduce a loop into your topology.

You configure the interface properties for an access port using the exact same process as when defining it for the default switch. In fact, if you were to commit the configuration, the ge-1/0/5.0 interface would be placed in the default switch. Be careful not to commit the configuration as it stands, because you might introduce a loop into your switched network. One of the

following sections shows how to place the interface in the virtual switch. We highly recommend that you perform that step before committing the configuration.

Configure a Trunk Port



You configure the interface properties for a trunk port using the exact same process as when defining it for the default switch. In fact, if you were to commit the configuration on the slide, the ge-1/1/4.0 interface would be placed in the default switch. Be careful not to commit the configuration as it stands, because you might introduce a loop into your switched network. One of the following sections shows how to place the interface in the virtual switch. We highly recommended that you perform that step before committing the configuration.

Configure an IRB Interface

- Configure an IRB interface that acts as the IP gateway for a bridge domain within the virtual switch

```
[edit]
user@switch# show interfaces
…
irb {
    unit 1 {
        description "GW for VLAN_200";
        family inet {
            address 172.22.2.254/24;
        }
    }
}
…
```

You configure the interface properties for an IRB interface using the exact same process as when defining it for the default switch. In fact, if you were to commit the configuration on the slide, the irb.1 interface would be placed in the default switch. Be careful not to commit the configuration as it stands, because you might introduce a loop into your switched network. The following section shows how to place the interface in the virtual switch. We recommended that you perform that step before committing the configuration.

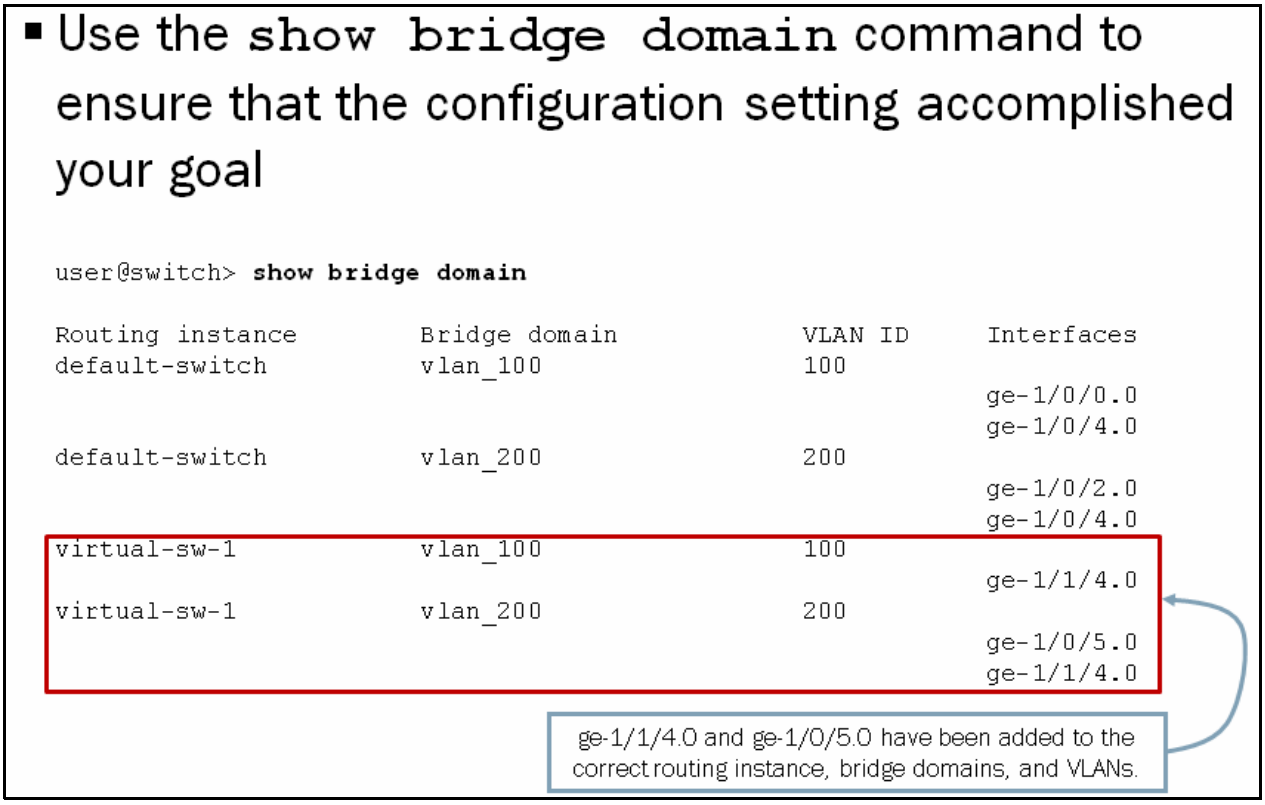
Add the Interfaces to the Virtual Switch

- Specify the interfaces that belong to the virtual switch:
  - List the trunk and access ports as member interfaces of the virtual switch
  - List the IRB as the routing interface for the appropriate bridge domain within the virtual switch

```
[edit]
user@switch# show routing-instances
virtual-sw-1 {
    instance-type virtual-switch;
    interface ge-1/0/5.0;
    interface ge-1/1/4.0;
    bridge-domains {
        vlan_100 {
            vlan-id 100;
        }
        vlan_200 {
            vlan-id 200;
            routing-interface irb.1;
        }
    }
}
```

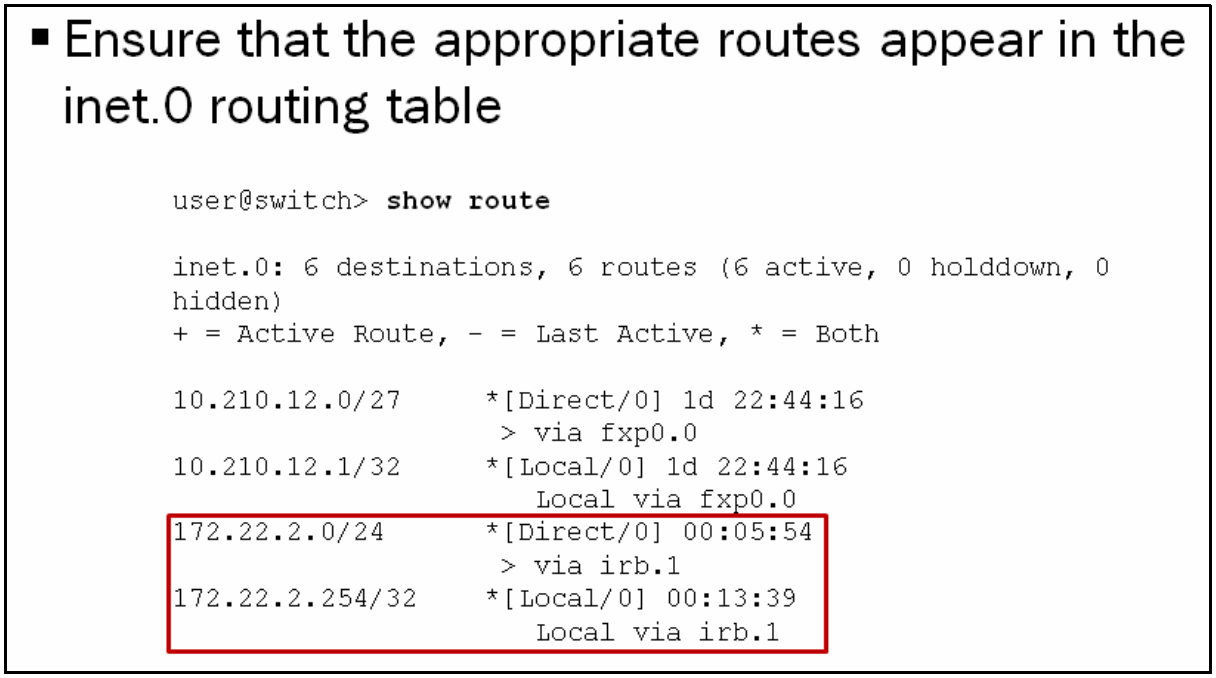
After configuring the access and trunk ports as shown on the previous sections, you simply need to list the interface at the `[edit routing-instances` *`instance-name`*`]` level of the hierarchy. The `irb.1` interface should be listed as the `routing-interface` for the appropriate bridge domain.

Verify Settings



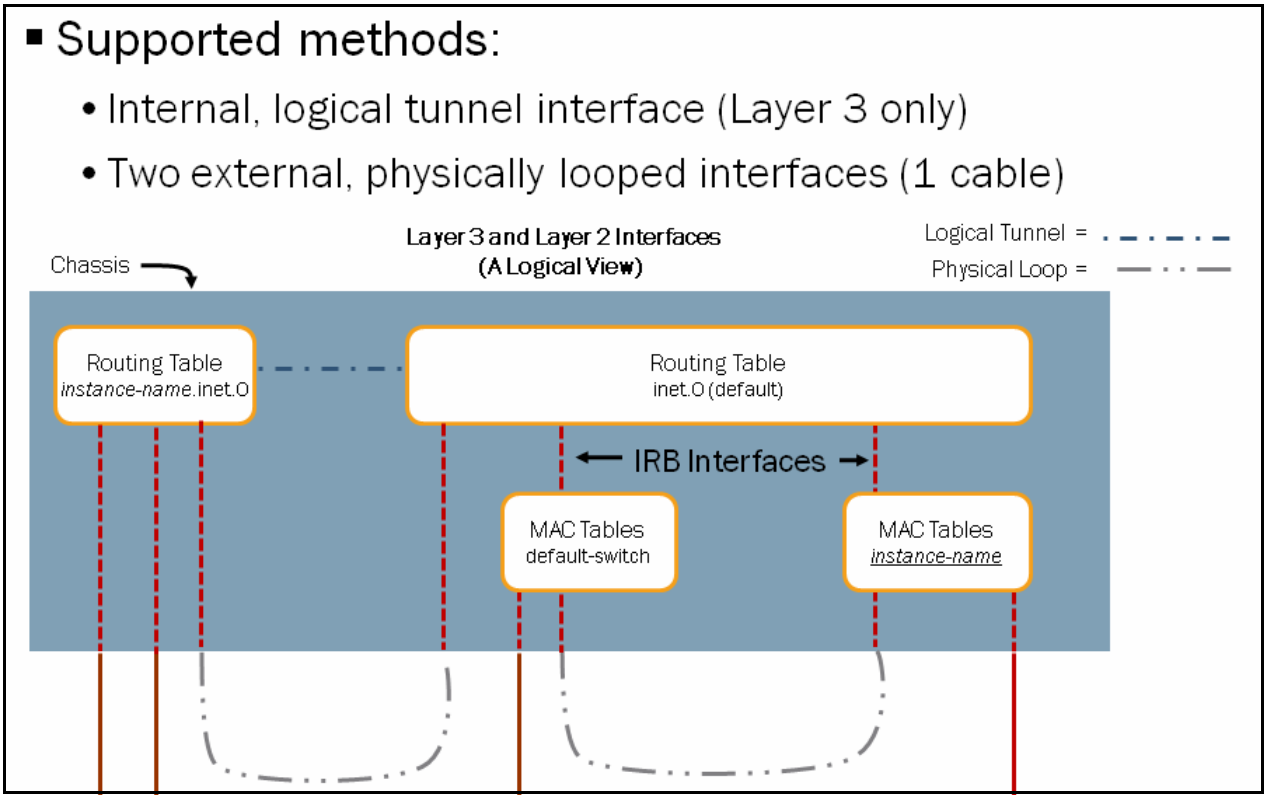
Looking at the output on the slide, you can see that the `ge-1/1/4.0` interface is now bound to the `virtual-sw-1` routing instance and the bridge domains `vlan_100` and `vlan_200`. Also, `ge-1/0/5.0` is bound to the appropriate routing instance and bridge domain.

IRB Routes



The local and direct routes that associate with the IRB interface should be in the inet.0 table. Use the **show route** command to verify that the routes were added properly.

Supported Methods of Interconnecting Routing Instances



As mentioned previously, to the outside world virtual routers and virtual switches appear as individual routers and switches. At some point you might want to interconnect the virtual routers and virtual switches that are local to a single chassis. For virtual routers, you can accomplish this task using either a logical tunnel interface or by looping two interfaces together with a single cable. For virtual switches, this process works only using the external cable method. The reason why spanning tree protocols do not function properly between virtual switches is because all virtual switches use the same MAC address as part of their bridge ID in the bridge protocol data units (BPDUs). Unfortunately, you cannot change a virtual switch's MAC address.

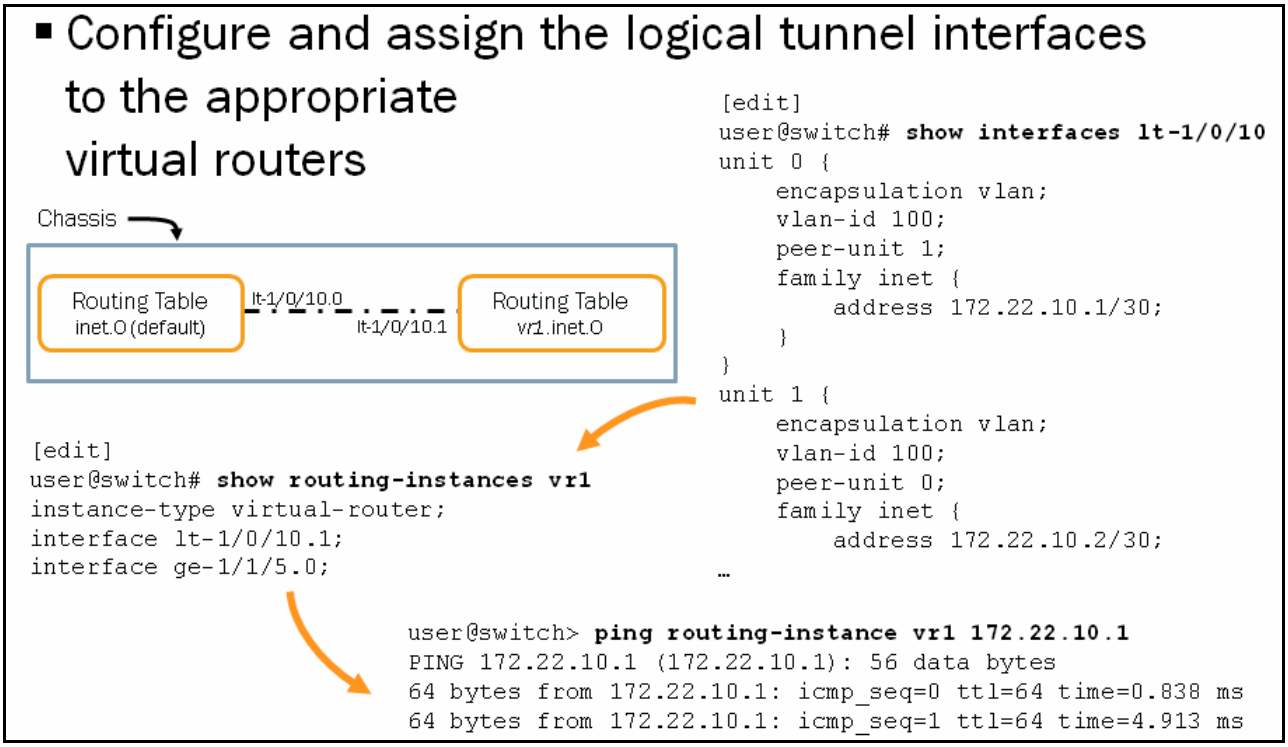
Tunnel Services

```
user@switch> show interfaces terse
Interface               Admin Link Proto Local  Remote
…
ge-1/0/8                up    down
ge-1/0/9                up    down
ge-1/1/0                up    down

user@switch> show interfaces terse
Interface               Admin Link Proto Local  Remote
…
ge-1/0/8                up    down
ge-1/0/9                up    down
gr-1/0/10               up    up
ip-1/0/10               up    up
lt-1/0/10               up    up
mt-1/0/10               up    up
pd-1/0/10               up    up
pe-1/0/10               up    up
vt-1/0/10               up    up
ge-1/1/0                up    down
```

```
[edit]
user@switch# show chassis
fpc 1 {
    pic 0 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}
```

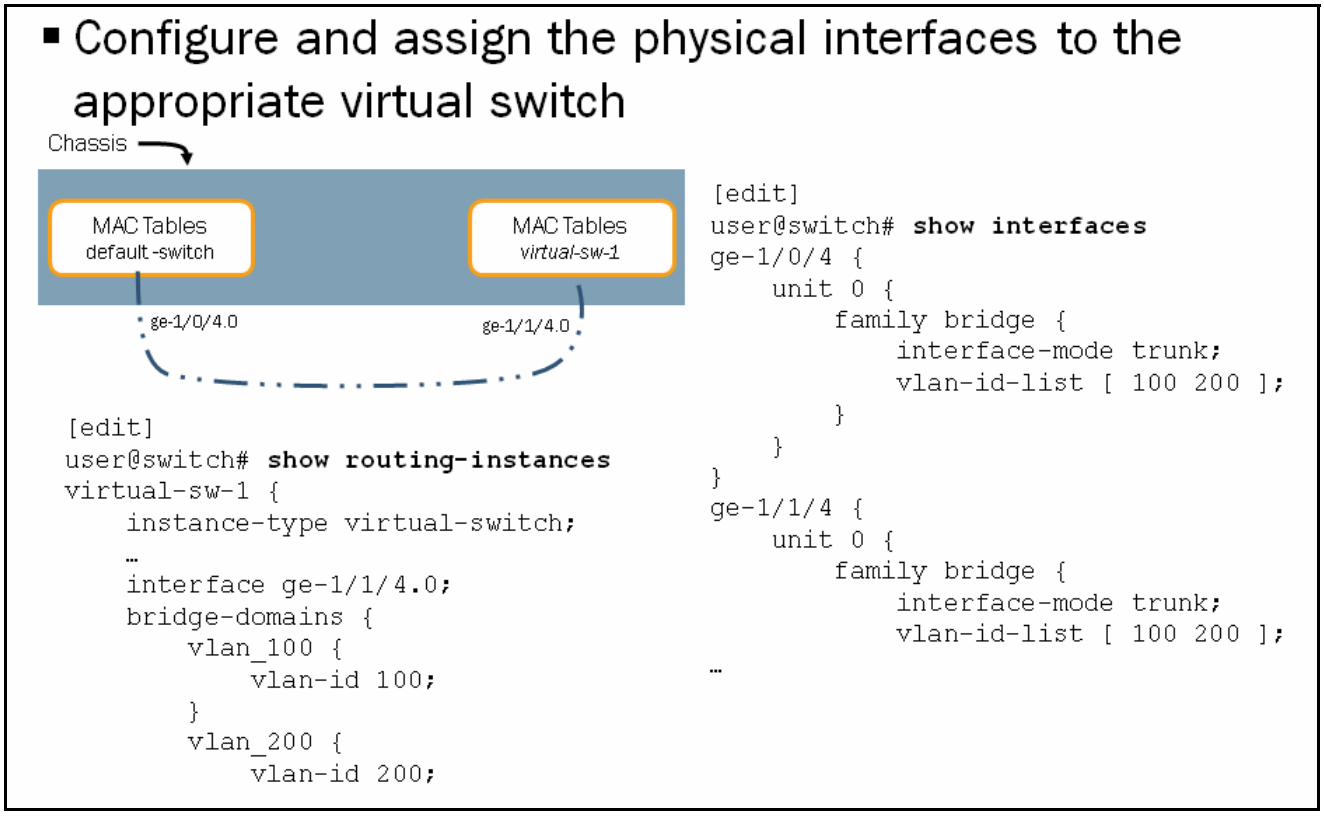
Anytime you need to use layer tunneling, you must enable tunnel services on the MX Series router. For example, you must enable tunnel services for a generic routing encapsulation (GRE) tunnel, an IP over IP (IP-IP) tunnel, Physical Interface Module (PIM) encapsulation or decapsulation of register messages, and for our case, using logical tunnel interfaces. Each Dense Port Concentrator (DPC) on a switch has either 40 Gigabit Ethernet ports (10 ports per PFE) or 4 10-Gigabit Ethernet ports (1 port per PFE.) Each PFE on an MX Series DPC can provide tunneling services but you must enable it. The slide shows how to enable tunnel services on the first PFE (serving ge-1/0/0 through ge-1/0/9) on the 40 1-Gigabit Ethernet DPC in slot number 1. Once you enable this feature, you will notice that you have several tunnel type interfaces that become available for your use. Notice that the tunnel interfaces use the logical PIC port number of 10 (normally PIC port numbers stop at 9.) When enabling tunnel services on a PFE of a 4-port 10 Gigabit Ethernet DPC, the Ethernet interface for that PFE is removed from service and is no longer visible in the command-line interface (CLI).

Configure and Assign Logical Tunnel Interfaces



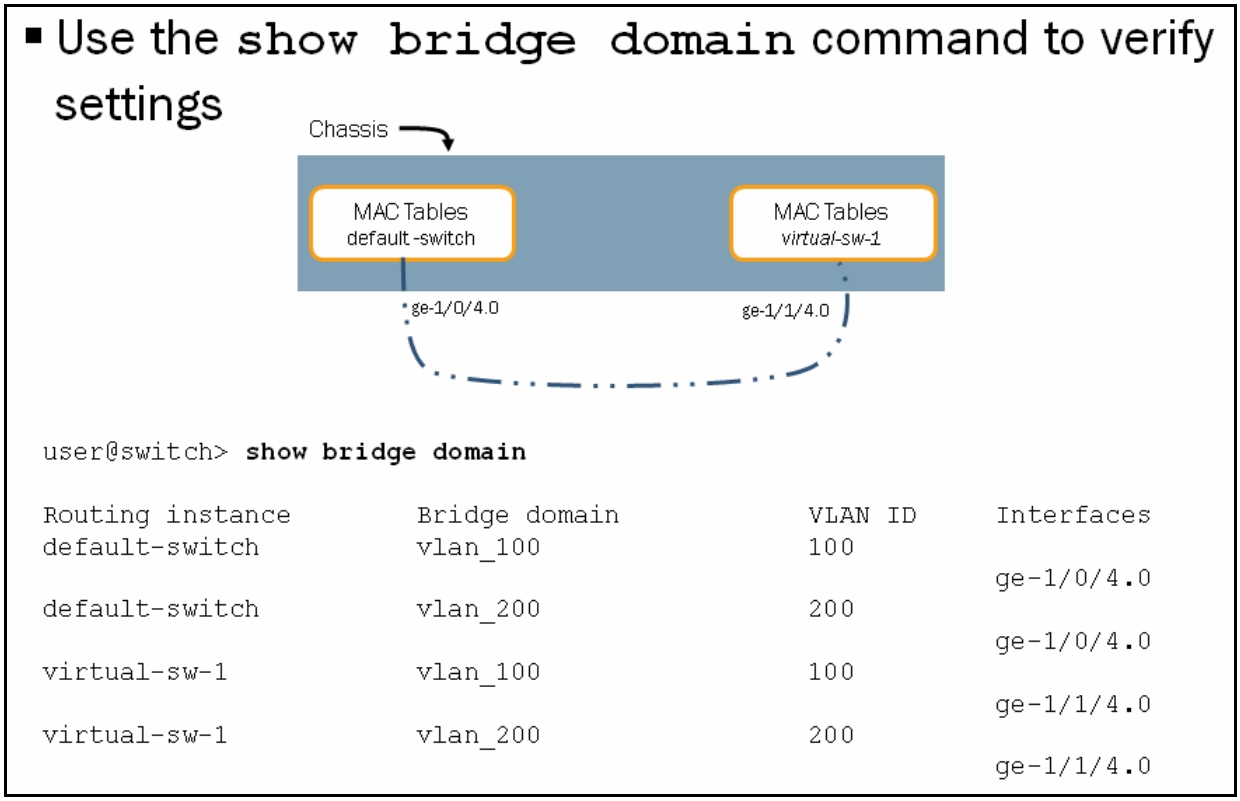
You configure the logical tunnel interfaces similar to how you would for any other Layer 3 interface. You configure each logical tunnel Layer 3 interface as a logical unit. To map one logical unit to another, use the **`peer-unit`** statement. By default, logical tunnel interfaces are placed in the default virtual router. To place a logical tunnel interface in a virtual router, specify the logical tunnel interface at the `[edit routing instance` *`instance-name`*`]` level of the hierarchy.

Configure and Assign Physical Interfaces



The graphic shows how to configure and assign Layer 2 interfaces to virtual switches.

Verify Settings



Looking at the output on the slide, you can see that the `ge-1/1/4.0` interface is now bound to the `virtual-sw-1` routing instance and the bridge domains `vlan_100` and *`vlan_200`*, whereas `ge-1/0/4.0` belongs to the default switch.

Review Questions

1. How can you make your MX Series router appear as multiple routers to other devices? How can you make it appear as multiple switches?
2. After configuring an interface under `[edit interfaces]`, which step do you perform next to ensure that the interface appears as part of the `vs1` virtual switch?
3. After configuring an IRB interface as part of the `vs1` virtual switch, in which routing table will you find its associated routes?

Answers

1.

For multiple routers, you can configure virtual-router routing instances. For multiple switches, you can configure virtual-switch routing instances.

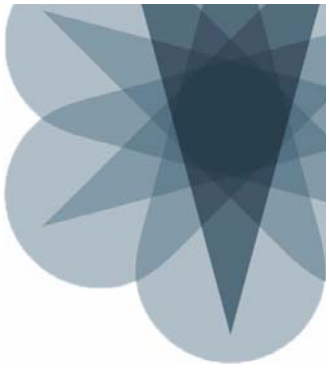
2.

You must list the interface at the `[edit routing-instances vs1]` level of the hierarchy to ensure that it appears as part of the `vs1` virtual switch.

3.

By default, you can find the routes associated with IRB interfaces in the inet.0 routing table.

JUNIPER NETWORKS



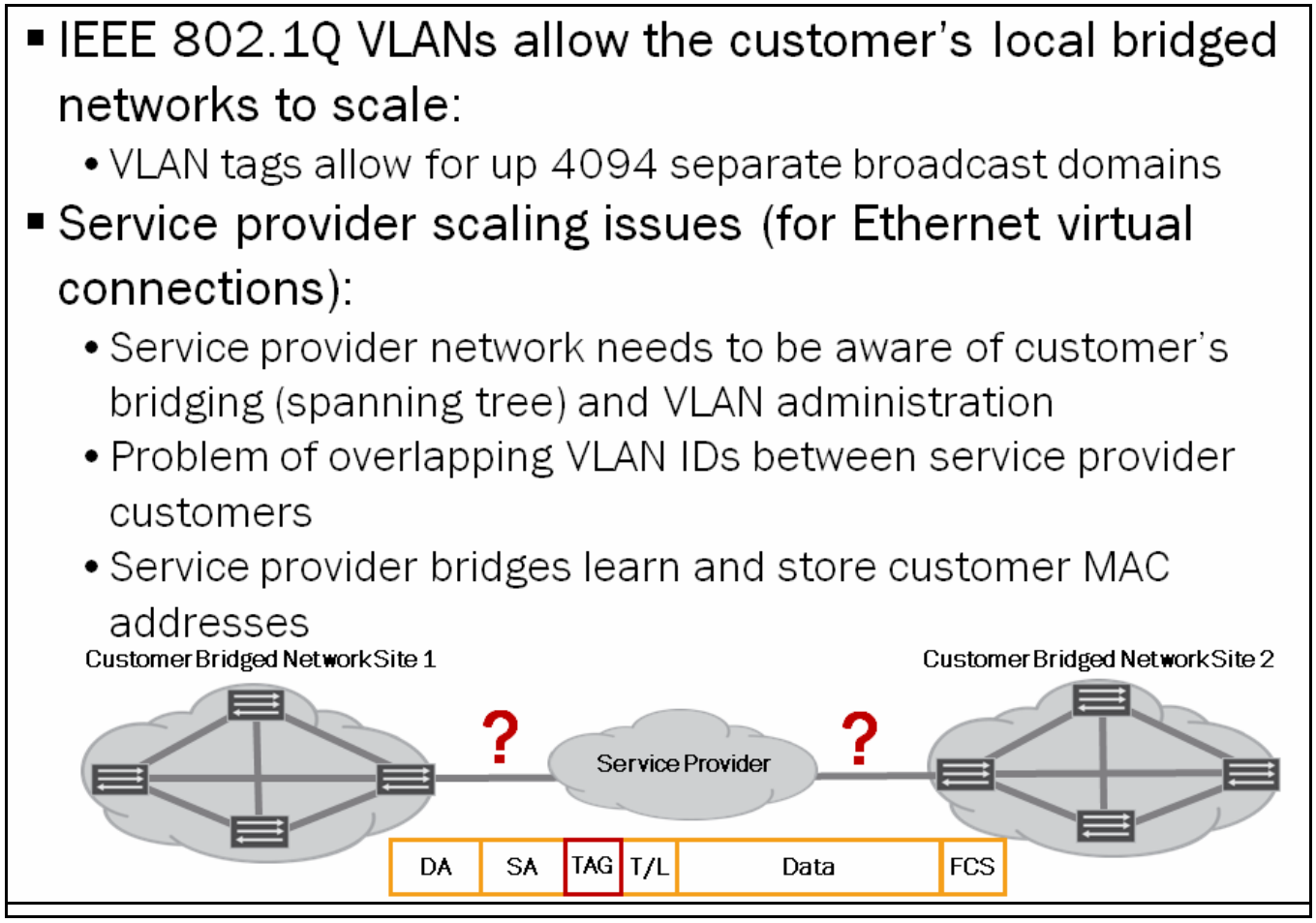
JNCIS-SP Study Guide—Part 2

Chapter 4: Provider Bridging

This Chapter Discusses:

- Institute of Electrical and Electronics Engineers (IEEE) virtual LAN (VLAN) stacking models;
- The components of provider bridging; and
- Configuration of provider bridging.

Scaling Customer Bridged Networks



IEEE 802.1Q VLAN tagging makes it possible for a customer's bridged network to scale. Instead of needing to add more bridging equipment to a growing network, VLAN tagging allows for the logical separation of a bridged network into many broadcast domains (or VLANs). With a 12-bit length VLAN ID, 4094 VLANs are available for use on a single physical Ethernet network.

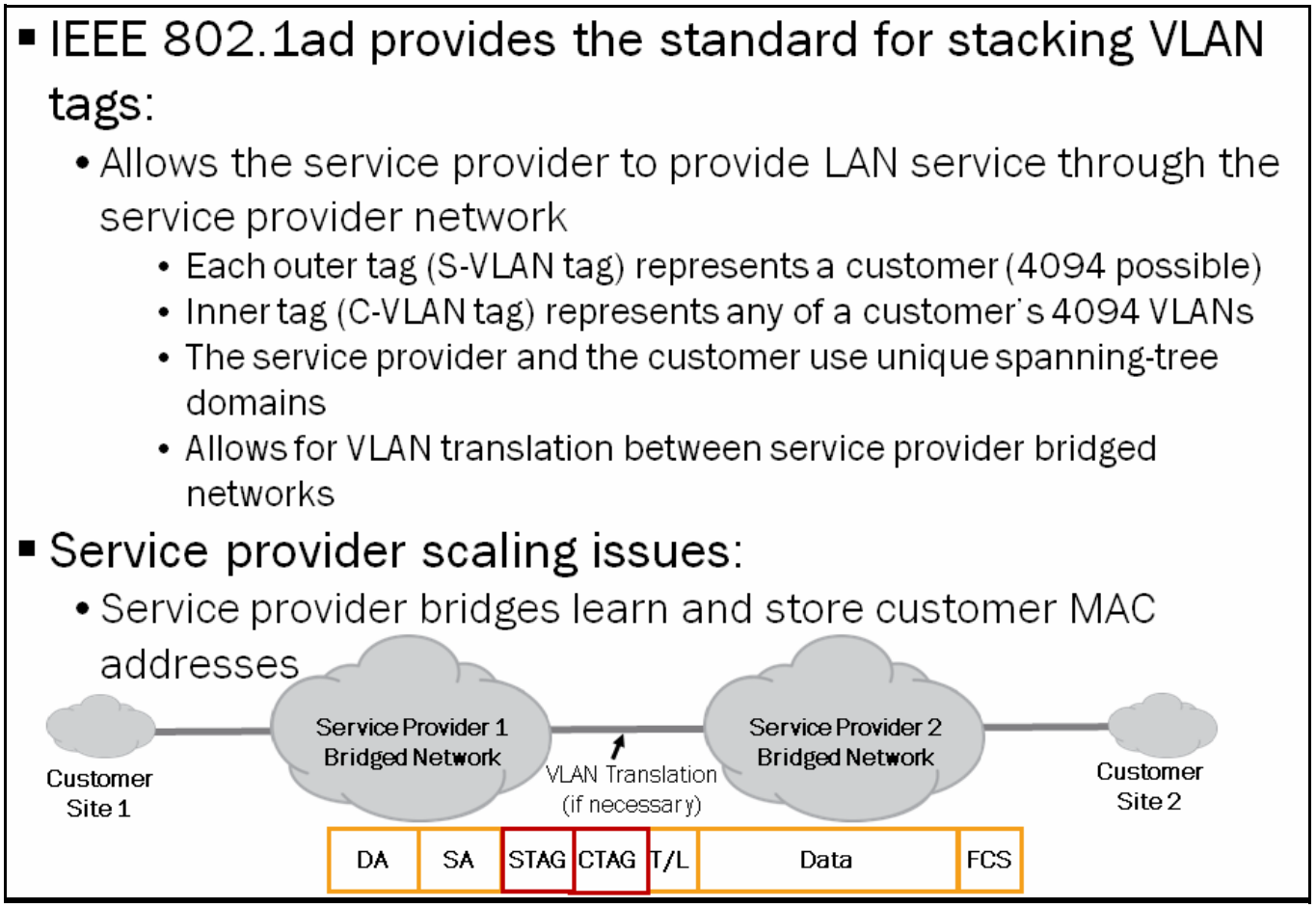
Ethernet from Service Providers

Because of its simple nature, service provider customers generally understand Ethernet. For a long time, service providers have searched for ways to deliver Ethernet Virtual Circuits (EVCs) to the customer premises. To a customer, an EVC between two sites should appear as a simple Ethernet link or VLAN through the service provider's network. IEEE 802.1Q VLAN tagging does not provide the scalability (in the service provider network) for a service provider to deliver that type of service.

From the service provider's point of view, the following is a list of some of the scaling issues that might arise:

- Because only one VLAN tag field exists in an 802.1Q frame, customers and the service provider need to coordinate the use of VLAN ID space. Considering that a service provider might have thousands of customers, this coordination would be an overly extreme effort.
- To pass Ethernet frames between customer sites, the service provider bridges must learn customer MAC addresses.
- To provide redundant links between customers and the service provider, running a form of the Spanning Tree Protocol (STP), which is generally not a viable solution, might be necessary. The STPs of today cannot scale to support all service provider and customer bridges of the world in a single spanning-tree domain.

IEEE 802.1ad



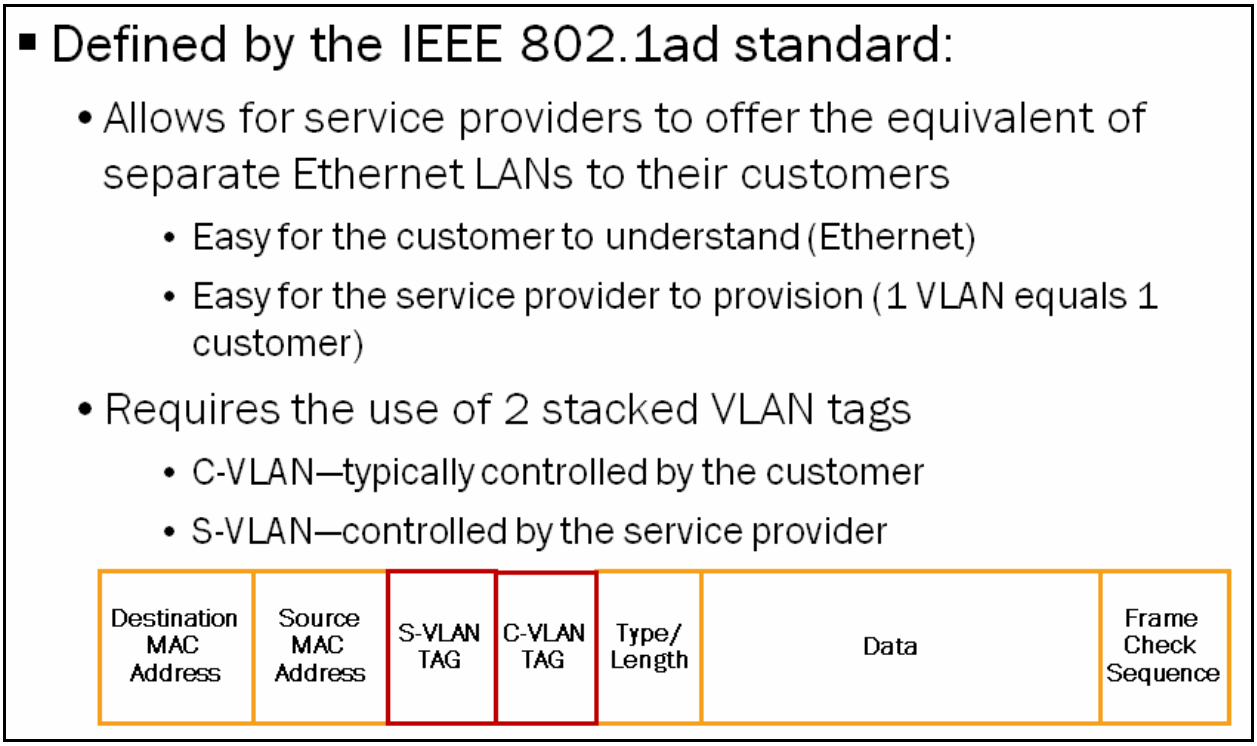
IEEE 802.1ad, also known as Q-in-Q tunneling, has standardized the methodology of stacking VLAN tags. The slide shows the frame format that the standard introduced. The standard gives a new name to the 802.1Q VLAN tag: the Customer VLAN (C-VLAN) tag (C-TAG). It also introduces a new tag named the Service VLAN (S-VLAN) tag (S-TAG). By adding the S-TAG to the frame, much less coordination is necessary between the customer and the service provider. At the customer site, the customer can continue to use 802.1Q tagging using C-VLAN IDs that are relevant only to their network (not the service provider's network). As 802.1Q-tagged frames arrive at the edge of the service provider's bridged network, the provider edge bridge (PEB) adds an S-TAG to the frame. The S-TAG, using a single S-VLAN ID, can carry any or all of the 4094 C-VLANs that are possibly in use by the customer. In the simplest case, a service provider can allocate a single S-VLAN ID to represent each of its individual customers, which allows the service provider to potentially support up to 4094 customers. IEEE 802.1ad also allows for the translating of

S-VLAN IDs at the edge of a service provider's bridged network, which helps in the coordination of VLAN ID usage between service providers.

Scaling Issues

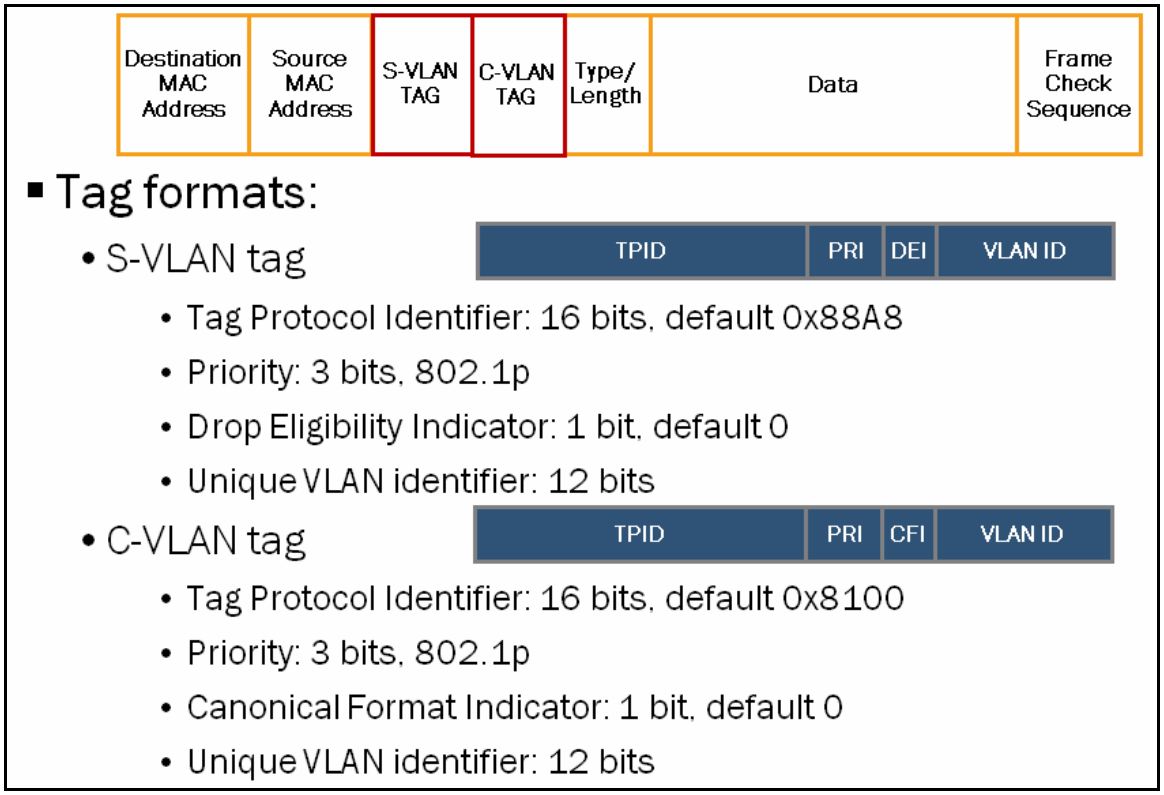
Although IEEE 802.1ad helps to solve the issue of the limited VLAN ID space that we discussed in relation to IEEE 802.1Q tagging, it does not solve the MAC learning problem. That is, for frames to be forwarded between bridges in the service provider's network, the bridges each must learn and store MAC addresses learned from the customer networks. A service provider can help alleviate this problem by limiting the number of learned MAC addresses or charging the customer more for the EVC service if they exceed the MAC address limit.

Provider Bridging



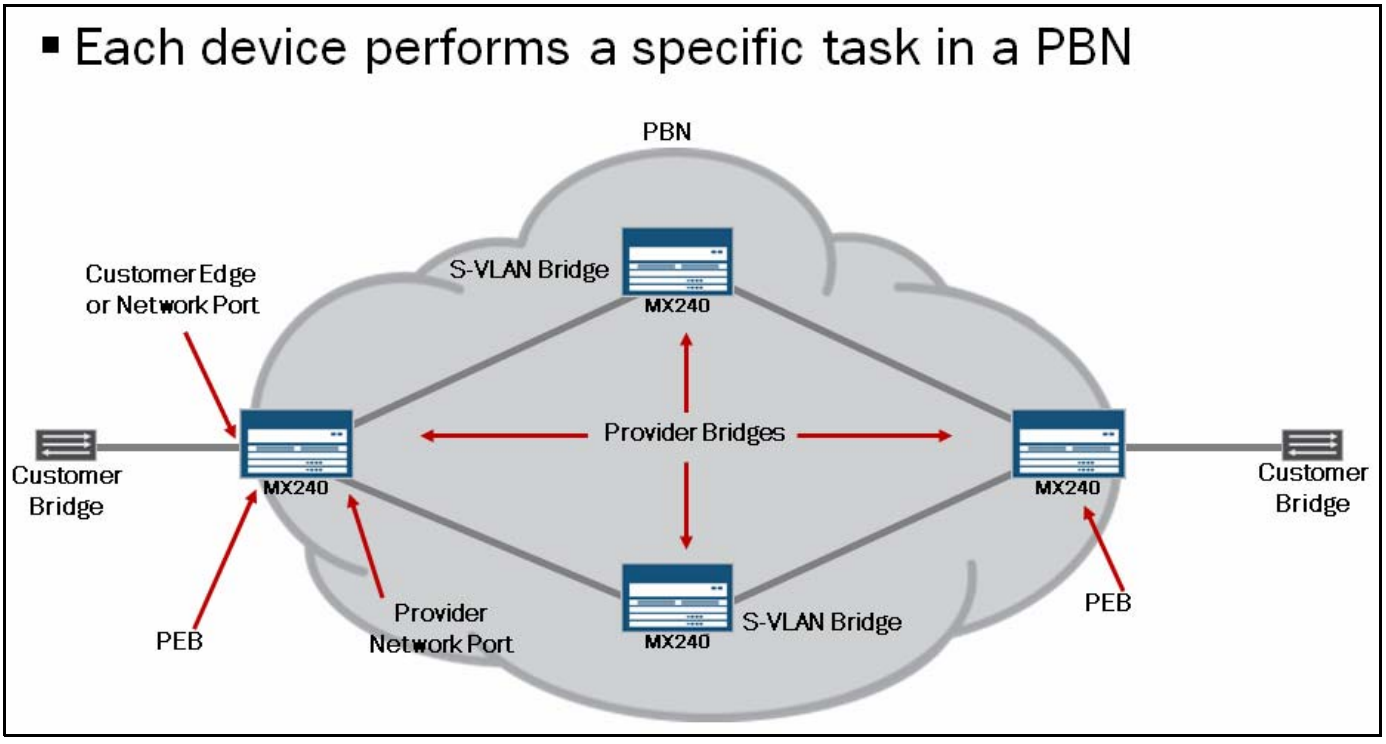
Provider bridging is defined under IEEE 802.1ad. It was developed to allow a service provider to provide a more scalable EVC service to its customers. A typical provider bridged network (PBN) provides for C-VLAN tagging and forwarding at the edge of the network using the ports that face the customer. For all ports that face the core of the PBN, the provider bridges forward based only on the S-VLAN tag.

IEEE 802.1ad TAG Formats



The graphic shows the S-TAG and C-TAG formats defined under IEEE 802.1ad. Note that the C-TAG remains identical to the IEEE 802.1Q VLAN tag. The S-TAG is similar but a few fields have been redefined. For example, because the canonical format indicator (CFI) field in the C-TAG is rarely used (for use in token ring networks), it has been redefined in the S-TAG to represent a frame's eligibility to be dropped. The Drop Eligibility Indicator (DEI) is used for class of service, which we do not discuss in this guide. Also, IEEE 802.1ad has reserved a Tag Protocol Identifier (TPID) of 0x88A8 for the S-TAG, however, the Junos operating system default behavior is to set the TPID equal to 0x8100.

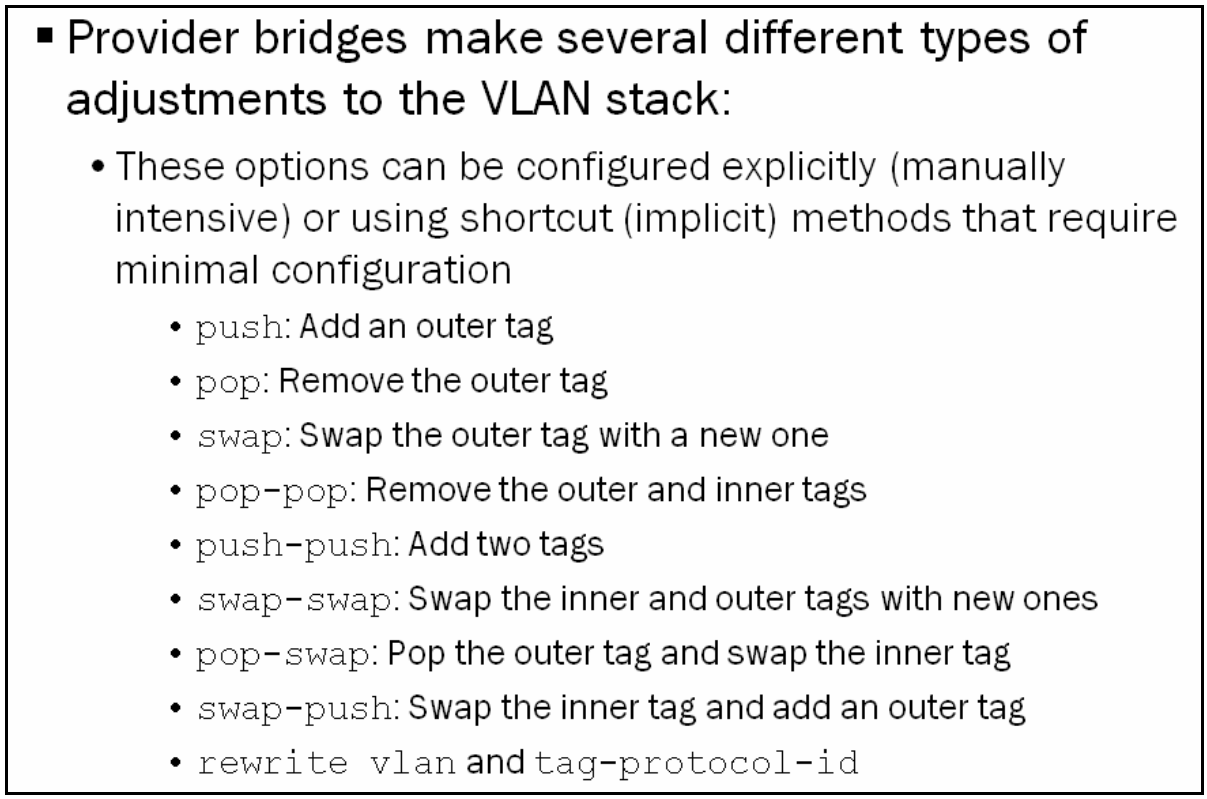
PBN Terms



The following terms are used in a PBN network:

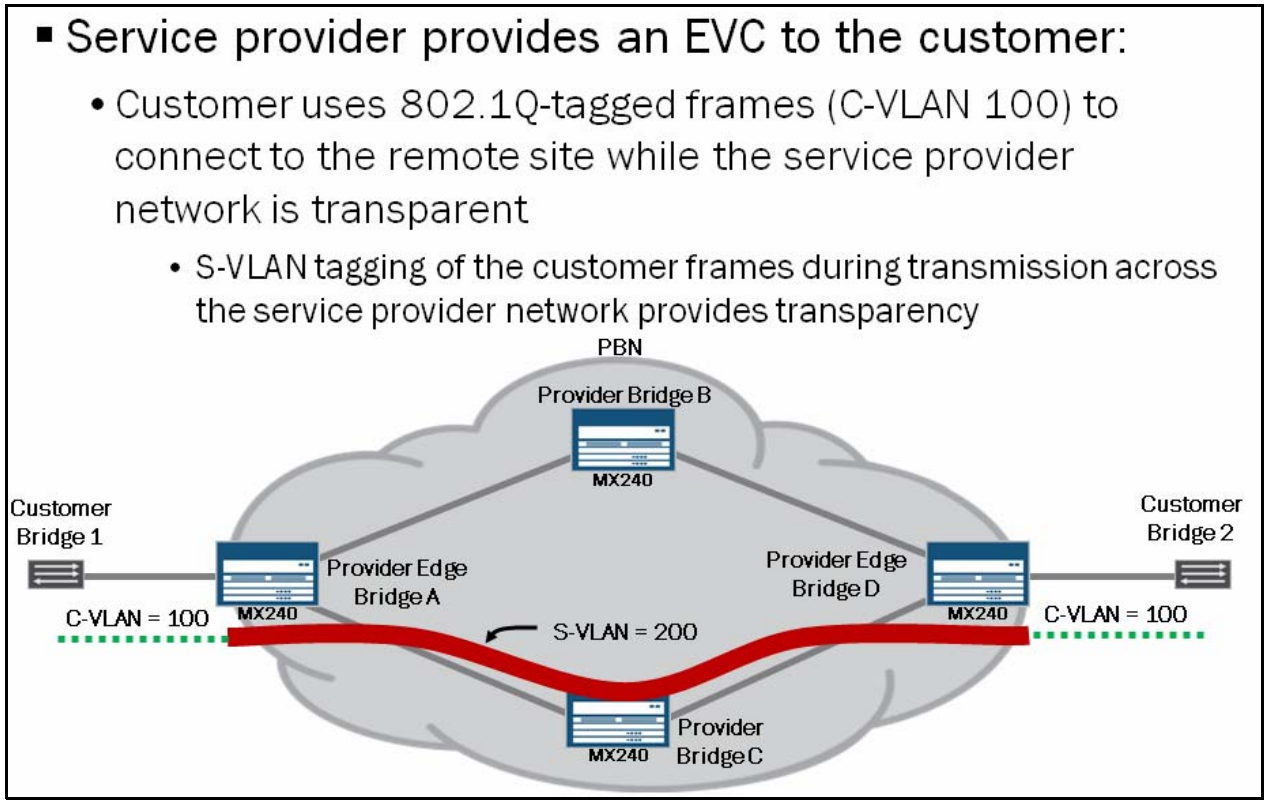
- *PBN*: A network of provider bridges that provide for transparent EVC service to the service provider's customers.
- *Provider Bridge*: A bridge in the service provider's network that performs IEEE 802.1ad VLAN tagging and forwarding. These bridges learn and store the MAC addresses of the service provider's customers.
- *Provider Edge Bridge (PEB)*: Accepts and forwards IEEE 802.1Q frames to and from customers. PEBs also encapsulate the received customer frames using the IEEE 802.1ad format to forward customer frames across the PBN.
- *S-VLAN Bridge*: A nonedge provider bridge that forwards frames based only on the S-VLAN tag.
- *Provider Network Port*: A port on a provider bridge that forwards frames based on the S-VLAN tag.
- *Customer Edge Port*: A port on a PEB that connects to customer equipment that receives and transmits C-VLAN tagged frames.
- *Customer Network Port*: A port on a PEB that receives and transmits S-VLAN tagged frames.

VLAN Tag Operations



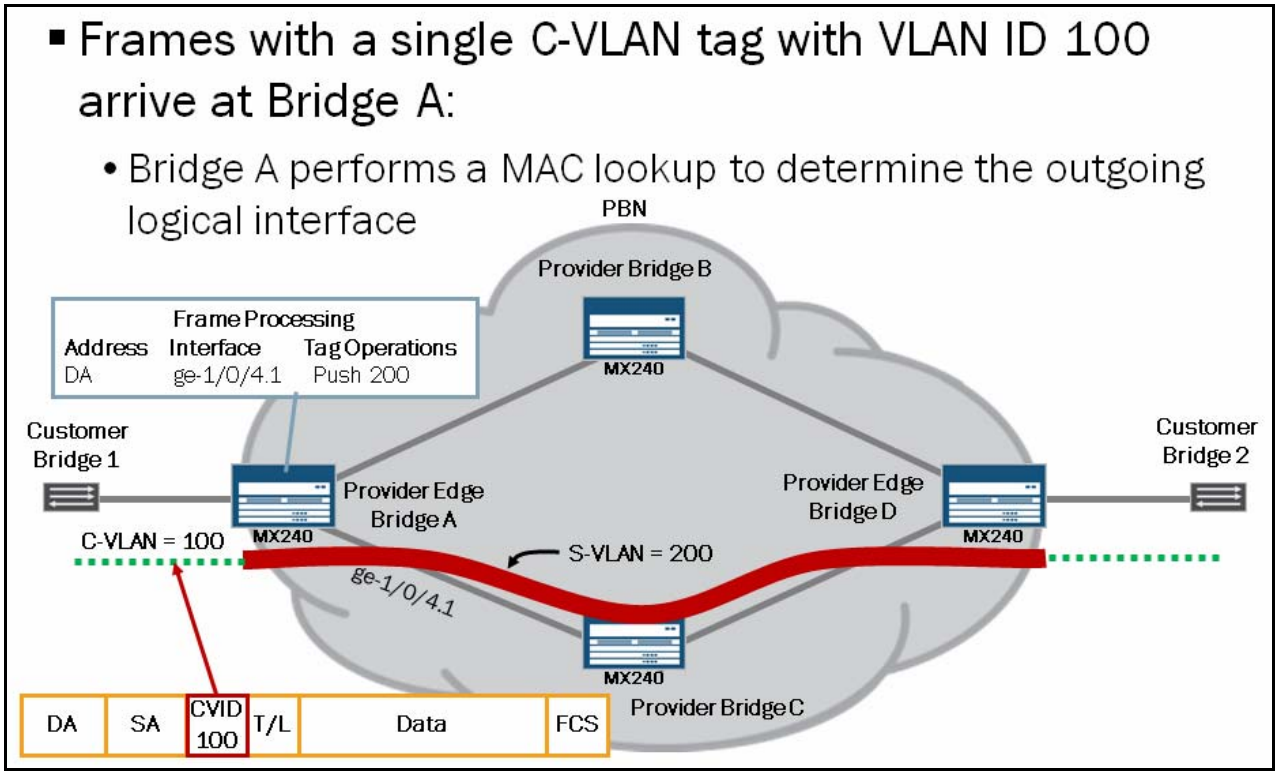
The graphic shows all of the possible operations that a provider bridge can perform on C-tagged frames and S-tagged frames that a port receives and transmits.

Service Provider Provides EVC Service to the Customer



In the example, the service provider delivers an Ethernet circuit to each of the customer premises. To provide connectivity between Customer Bridge 1 and Customer Bridge 2, the customer must enable an IEEE 802.1Q VLAN using VLAN ID 100 on the service provider-facing ports. The service provider has allocated an S-VLAN tag of 200 to transparently forward the customer's frames across the PBN. This allocation is performed by configuring a bridge domain on each provider bridge specifically for the customer specifying an S-VLAN ID of 200, and by configuring all possible inbound and outbound interfaces to support the appropriate VLAN tagging for the customer's bridge domain. For example, on Bridge A, the service provider would need to configure a Bridge Domain that accepts C-tagged frames on the customer-facing interface and S-tagged frames (VLAN ID 200) on the core-facing interfaces. Over the next several slides we look at the frame processing steps for traffic traversing a Q-in-Q tunnel.

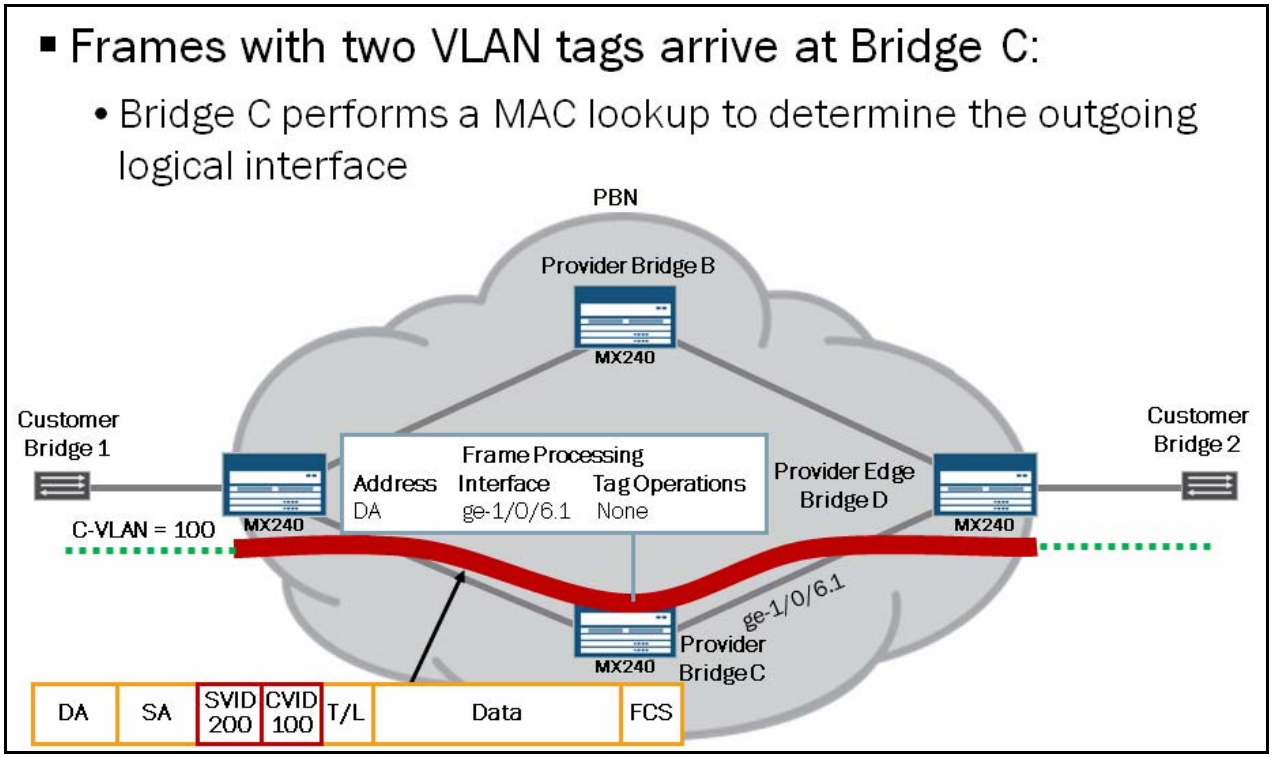
PEB Processing of Incoming Frames



When C-VLAN-tagged frames arrive at Bridge A (a PEB), Bridge A performs a MAC-table lookup based on the customer's bridge domain. If Bridge A has previously learned the destination MAC address of the frame, it forwards the frame to the appropriate outbound interface (ge-1/0/4.1 in this case) and the interface adds an S-VLAN of 200 on to the frame before sending the frame to the next bridge. The act of adding an outer tag to the frame is known as a push operation.

Note that if Bridge A did not previously learn the destination MAC address of the frames, it floods the frame out of every other interface associated with the customer's bridge domain except for the one that originally received the frame.

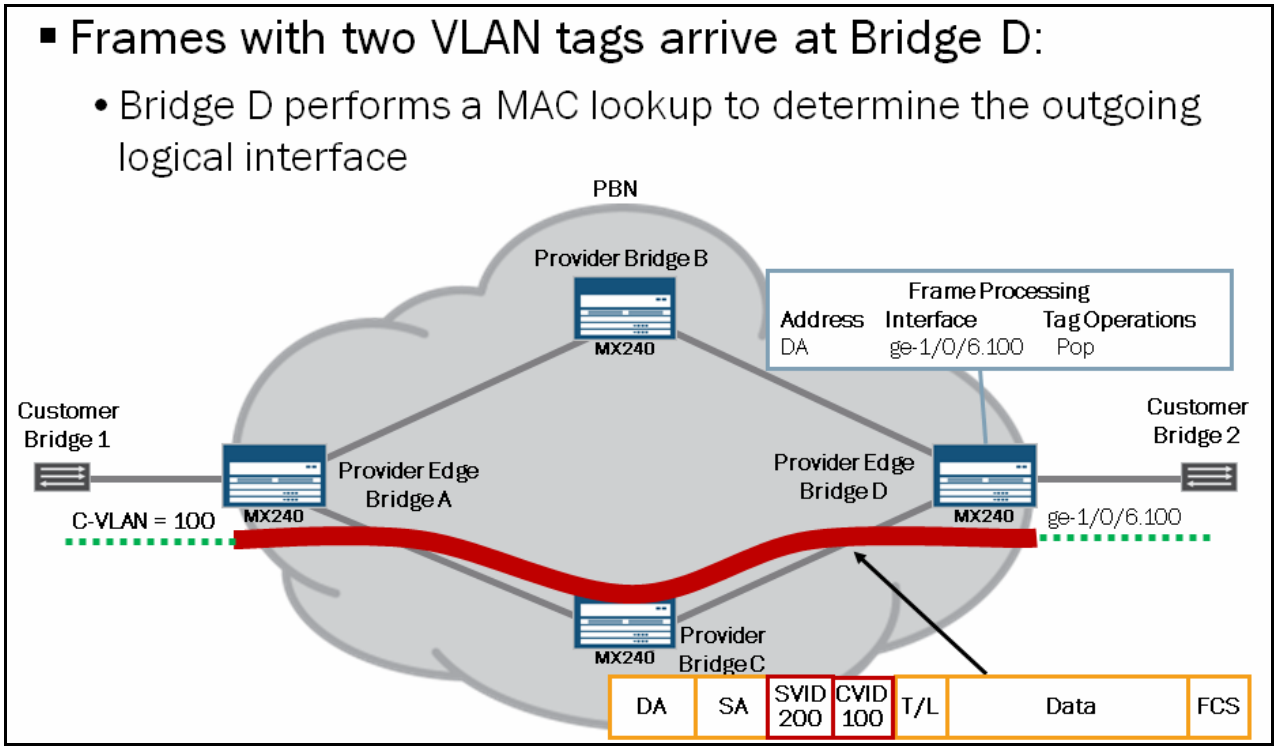
Bridge C Processes the Frame



When S-VLAN-tagged frames arrive at Bridge C (an S-VLAN bridge), it performs a MAC-table lookup based on the customer's bridge domain. If Bridge C has previously learned the destination MAC address of the frame, it forwards the frame to the appropriate outbound interface (ge-1/0/6.1 in this case) and the interface sends the frame unchanged to the next bridge.

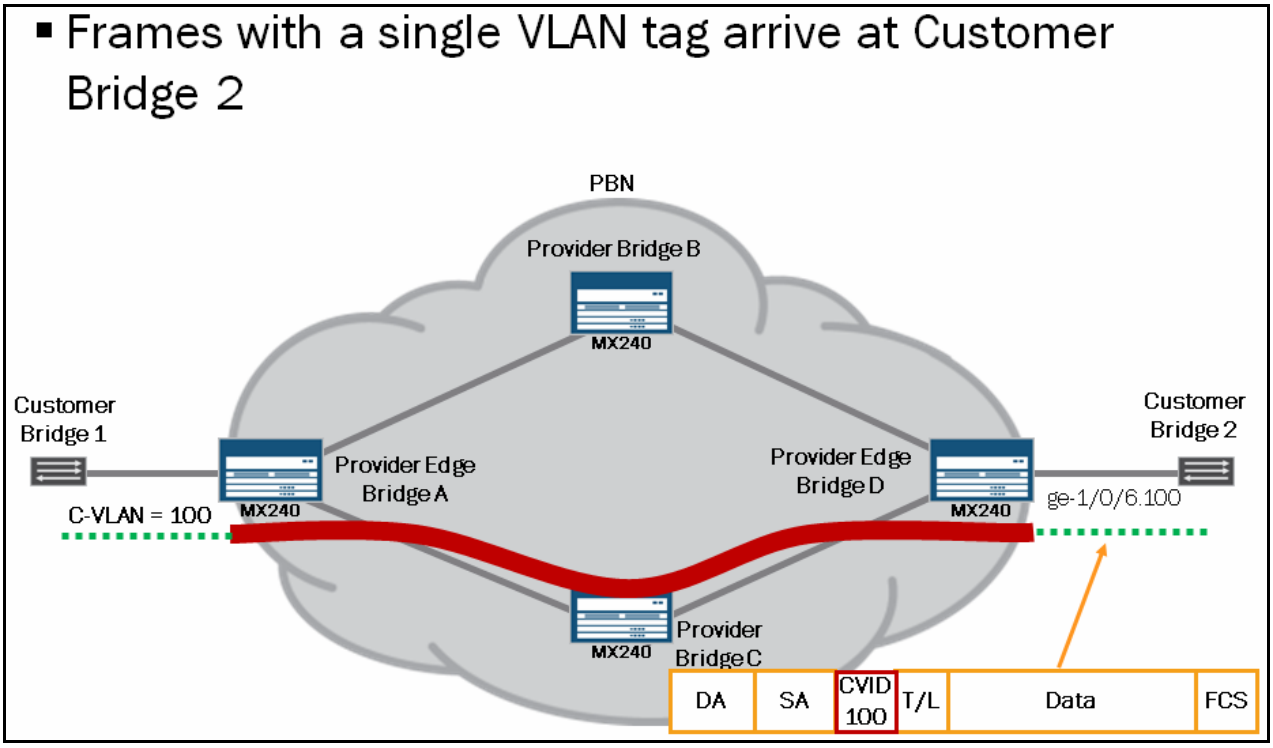
Note that a few ways exist to configure the VLAN operations on an S-VLAN bridge. The inbound interface on Bridge C can possibly also pop the S-VLAN tag on reception and then the outbound interface can push S-VLAN of 200 on transmittal.

Bridge D Processes the Frame



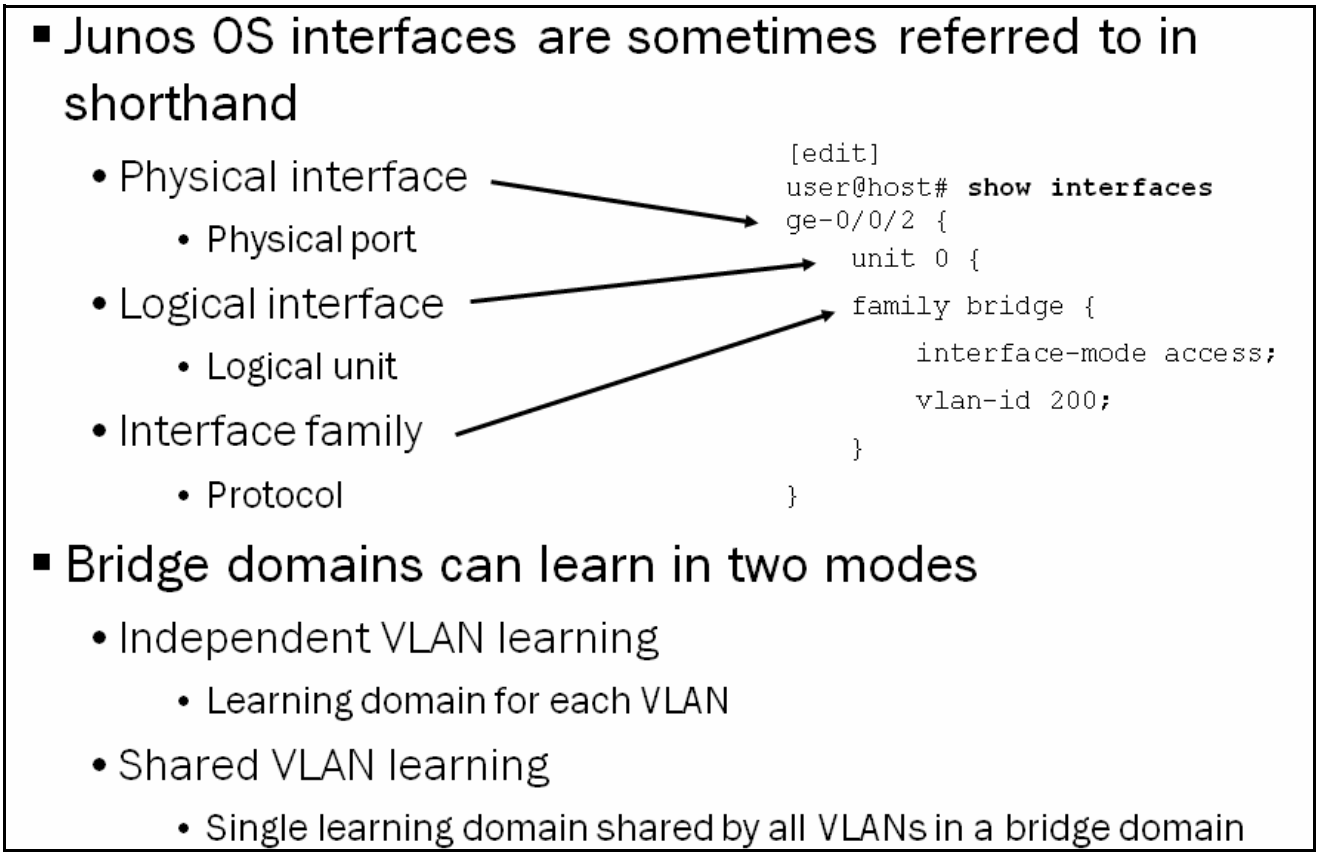
When S-VLAN-tagged frames arrive at Bridge D (PEB), the inbound interface pops the S-VLAN tag and Bridge D performs a MAC-table lookup based on the customer's bridge domain. If Bridge D has previously learned the destination MAC address of the frame, it forwards the frame to the appropriate outbound interface (ge-1/0/6.100 in this case) and the interface sends the C-tagged frame to the customer bridge.

Remote Customer Site



The slide shows the frame format of the Ethernet frame as it arrives at Customer Bridge 2. Note that the frame looks exactly as it did when Customer Bridge 1 transmitted it. At this point, Customer Bridge 2 will perform its own MAC-table lookup and forward the frame on to their intended destination, if known. If the destination MAC address is unknown, Customer Bridge 2 will flood frame out all other interfaces associated with VLAN-ID 100.

Junos OS Interface Terminology



Shorthand methods of describing the Junos OS interfaces are common. A physical interface refers to a physical port. A logical interface refers to an individual logical unit. An interface family refers to an individual protocol family. Multiple logical interfaces can be configured for each physical interface. Multiple interface families can be configured for each logical interface. In regards to bridging, understanding how a configuration affects the number of logical interfaces on an MX Series 3D Ethernet Universal Edge Router (64 K maximum) is important.

Bridge Domain Modes

So far, we discussed configuring bridge domains in independent VLAN learning mode (IVL). In this mode, MAC learning occurs on a per VLAN basis. That means, broadcast, unicast with unknown destination, and multicast (BUM) traffic flooded on interfaces is associated with a single VLAN. However, another bridge domain mode exists named shared VLAN learning mode (SVL). This allows for VLANs to share MAC learning. That means, the BUM traffic floods on all interfaces and all VLANs associated with a bridge domain. The following slides show examples of each mode of operation.

New Style of Configuration

The example in the graphic shows the "new" method of configuration to create dual-stacked VLAN subinterfaces. To configure the outer VLAN, specify a `vlan-id` at the unit level. To specify one or more inner VLAN IDs, use the **`inner-vlan-id-list`** command at the `family bridge` level of the hierarchy. To view the "older" method of configuring dual-stacked VLANs, refer to Appendix A.

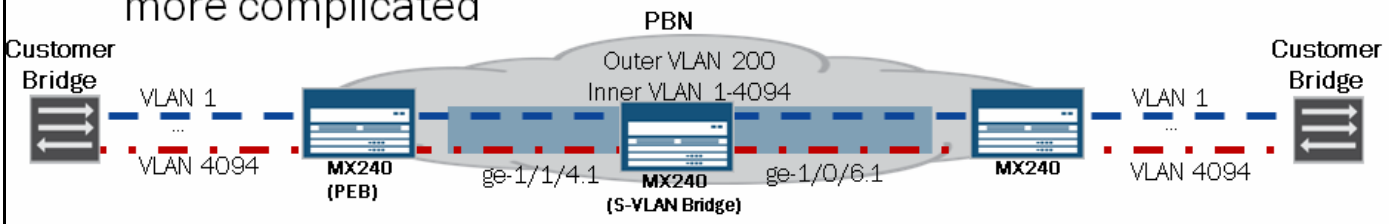
```
[edit]
user@peb# show interfaces ge-1/0/4
flexible-vlan-tagging;
unit 0 {
    vlan-id 200;
    family bridge {
        interface-mode trunk;
        inner-vlan-id-list 111-114;
…
```

Outer VLAN

Inner VLANs

S-VLAN Bridge Configuration

- An S-VLAN bridge forwards frames using only the S-TAG:
  - If C-TAG operations are necessary, configuration becomes more complicated

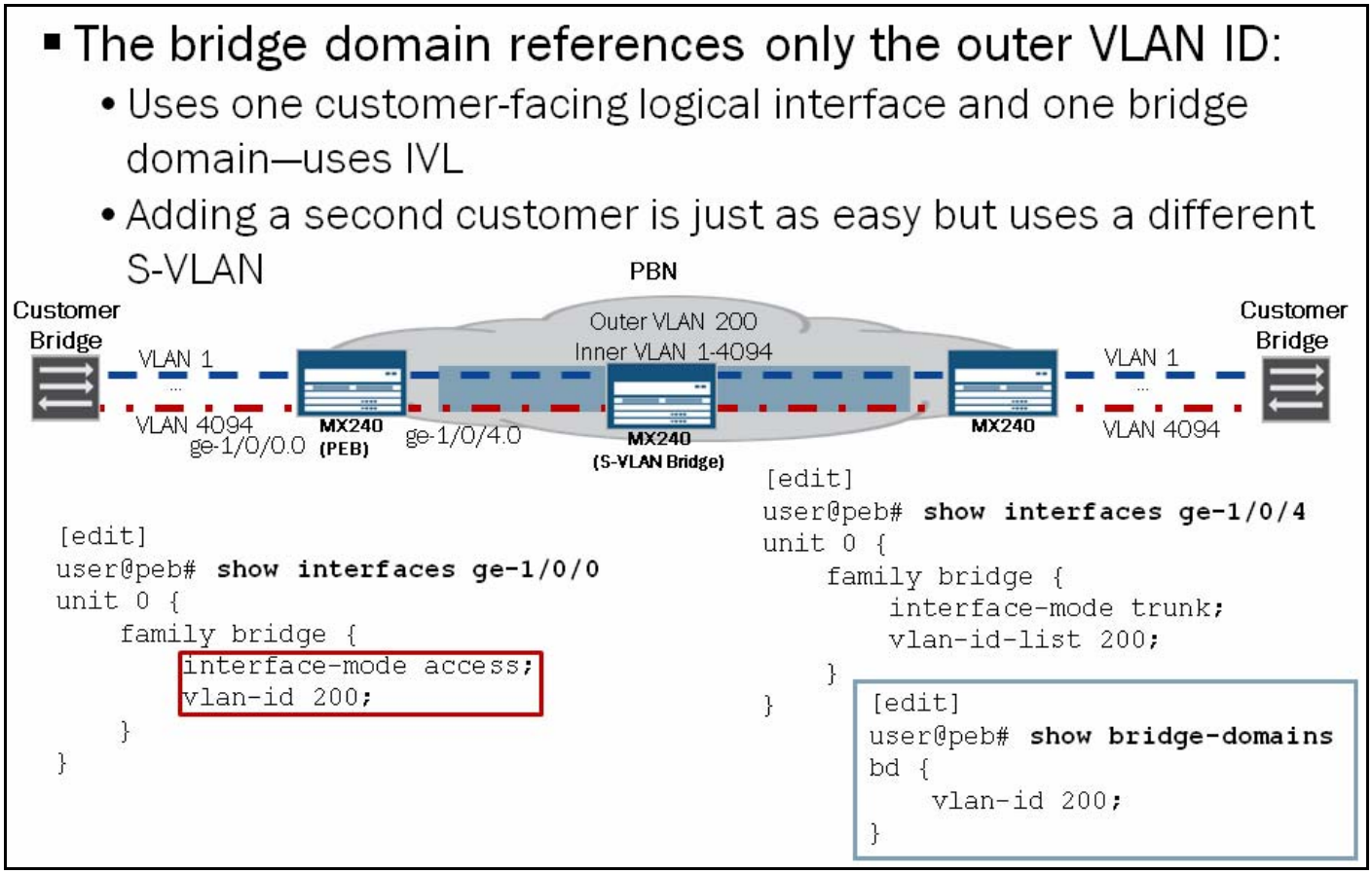



```
[edit]
user@S-VLAN-Bridge# show interfaces
ge-1/0/6 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 200;
…
ge-1/1/4 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 1 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 200;
        …
```

```
[edit]
user@S-VLAN-Bridge# show bridge-domains
customer1 {
    vlan-id 200;
}
```

The easiest configuration for supporting provider bridging is on an S-VLAN bridge similar to the core (middle) switch on the slide. Because the switch processes only S-VLAN tags, you can configure the bridge domain using the **`vlan-id`** ***`number`*** statement. We expect only S-tagged frames to arrive on each trunk interface, so you can configure them for a single **`vlan-id-list`** statement as well. To allow the interfaces to support two VLAN tags, include the **`stacked-vlan-tagging`** statement or the **`flexible-vlan-tagging`** statement.

Tunnel All C-VLANs



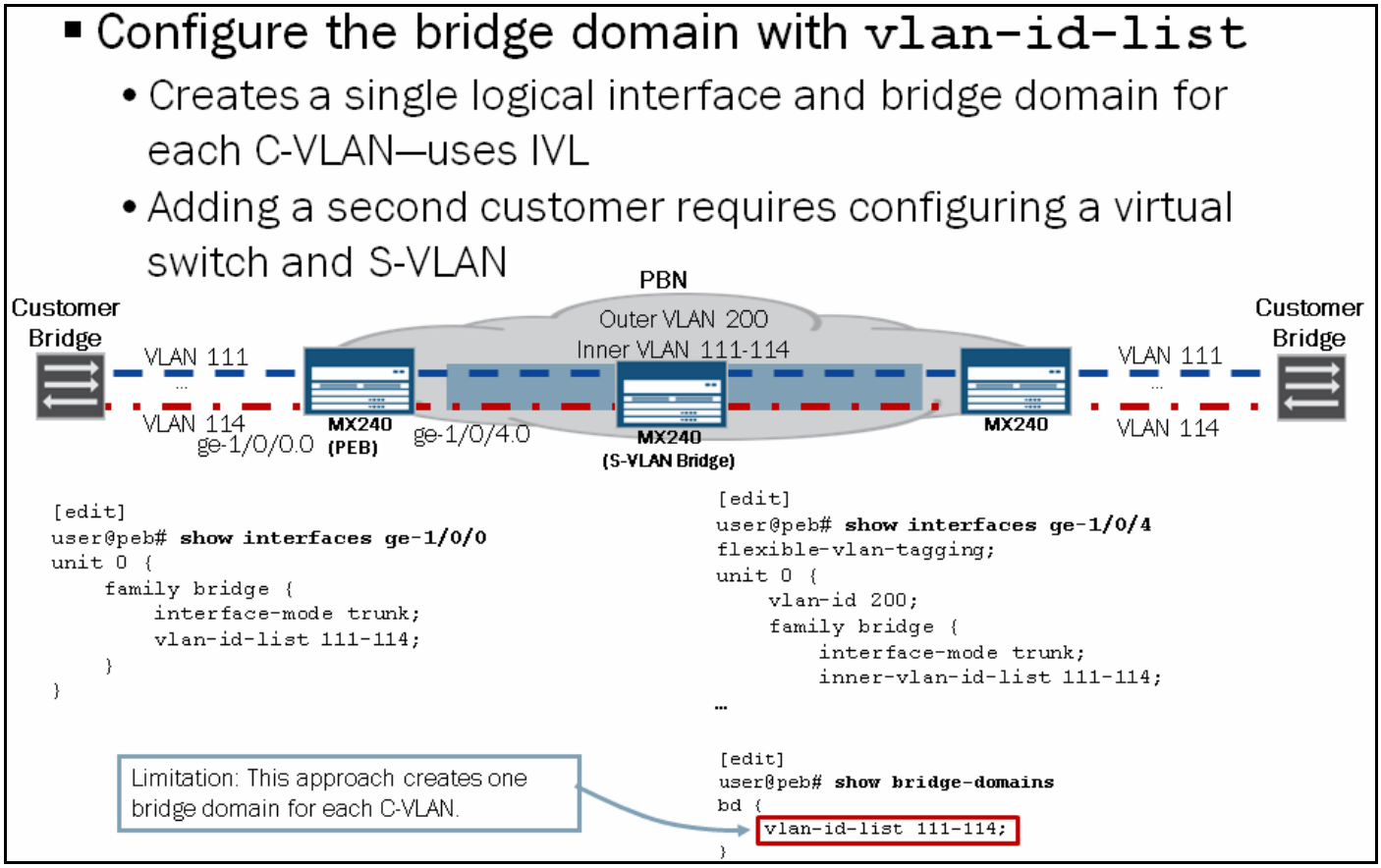
The method shown on the slide is the easiest and most elegant method of tunneling all customer C-VLANs across the core of a PBN. The interface and bridge domain configuration require only that you specify the outer S-VLAN ID. To allow single-tagged frames to enter the customer-facing interface, you must specify the `interface-mode` access statement.

You will see on the next few slides that each configuration method results in some combination of one of the following:

1. A bridge domain mode (IVL or SVL).
2. Customer-facing logical interface usage.
3. Bridge domain usage.
4. Virtual switch usage.

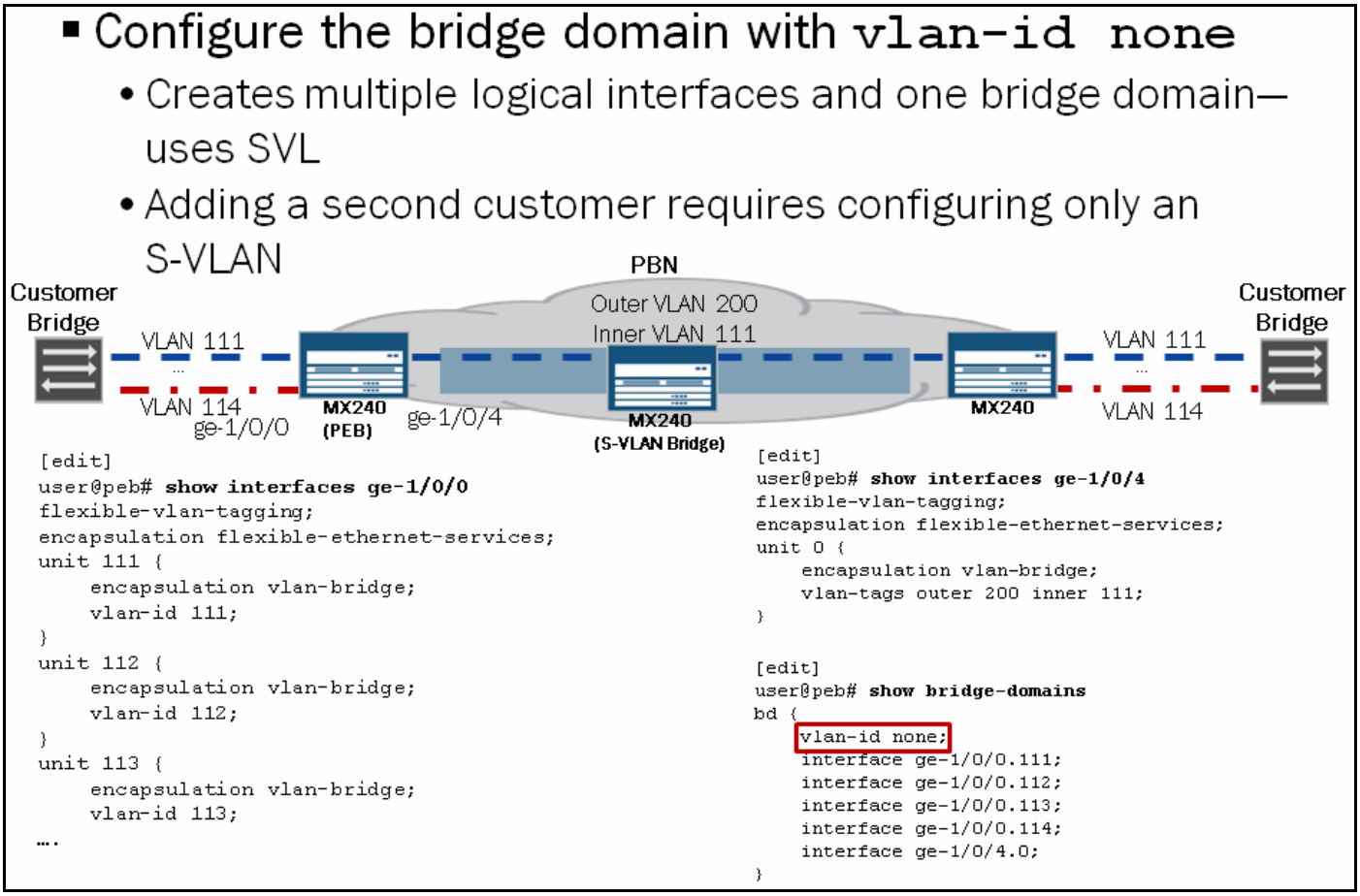
The solution on this slide is so elegant because to support each customer it requires the use of one logical interface, one bridge domain, and also, you can place each customer in the same virtual switch.

Range of C-VLANs: Part 1



Allowing only certain C-VLANs to be tunneled across the core might be necessary. Few solutions will allow this tactic. In this solution, the bridge domain references the C-VLAN IDs to be tunneled. Because of this reference, you must add each customer to its own virtual switch (in the case of overlapping C-VLAN space).

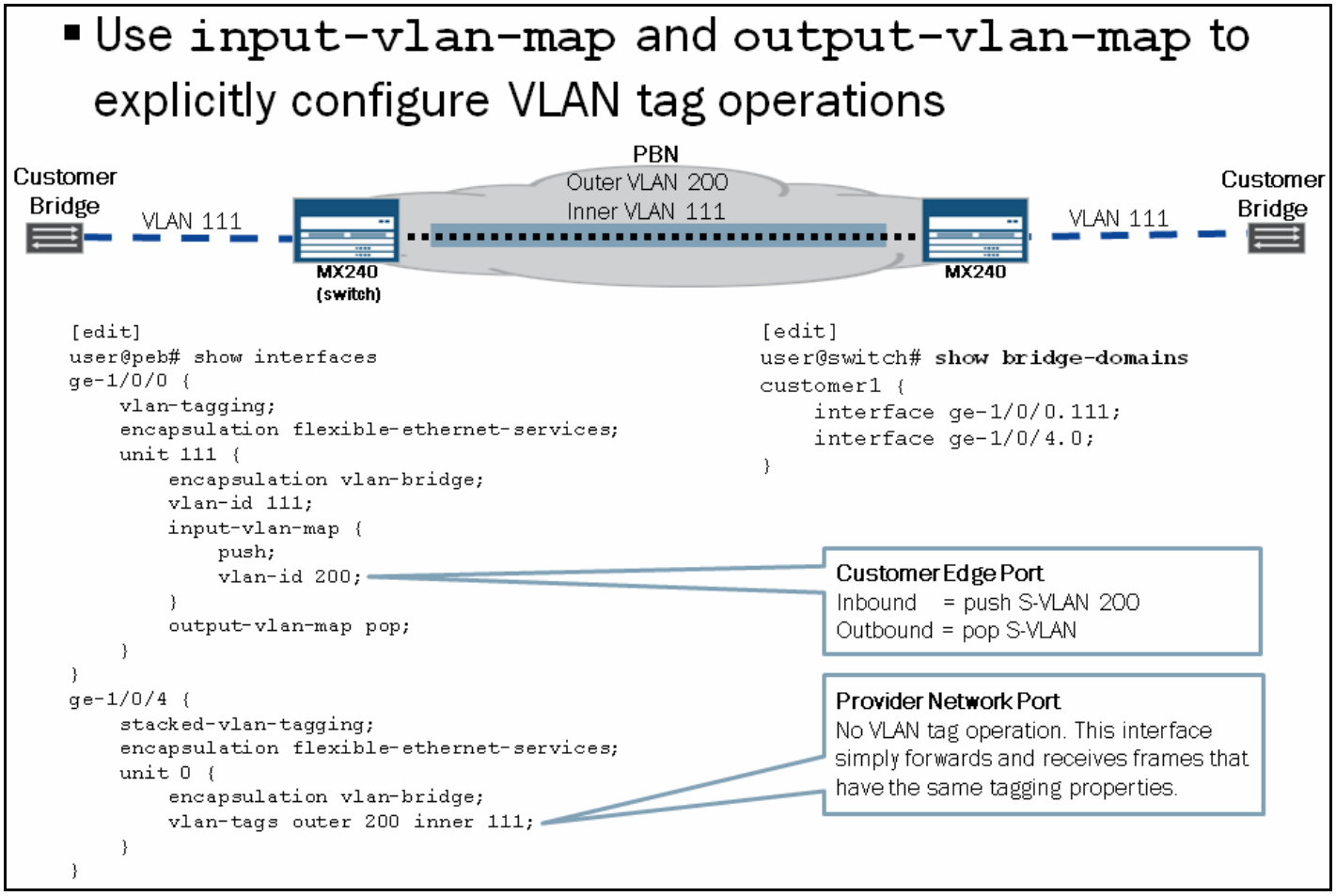
Range of C-VLANs: Part 2



The graphic shows the first example of SVL as well as an example of VLAN normalization (translation). The best way to describe how this solution works is to discuss what happens to a customer frame as it traverses the PBN:

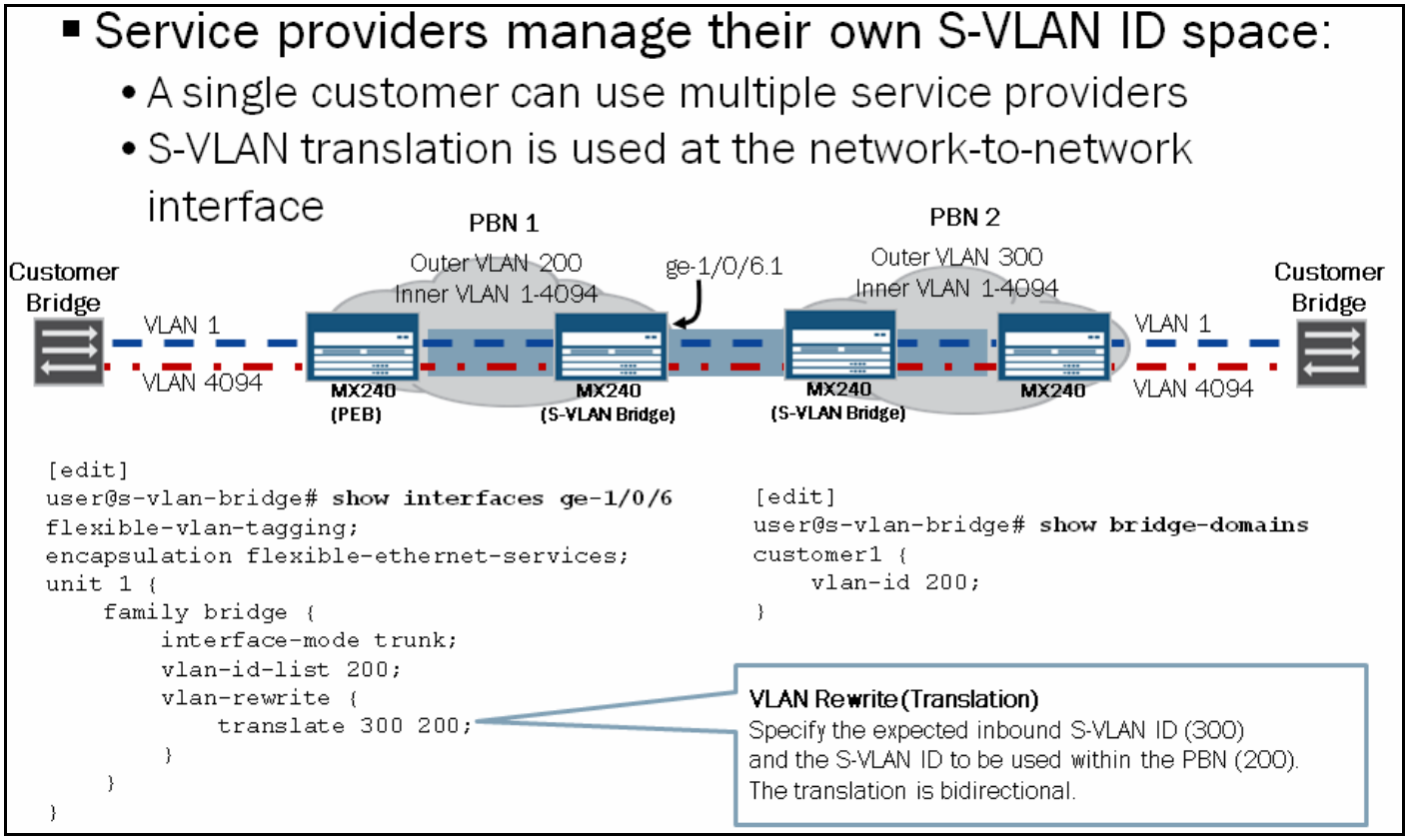
1. A frame with C-VLAN ID 112 arrives on ge-1/0/0.112 destined for a MAC address that exists on the remote side of the network.
2. Because the bridge domain is configured for **`vlan-id none`**, the C-VLAN tag pops before the MAC-table lookup.
3. If the destination MAC address is unknown, then the frame is flooded out of all interfaces that associate with the bridge domain, including the subinterfaces of ge-1/0/0 (because of SVL). If the destination MAC is known, the frame is forwarded out of the ge-1/0/4.0 interface with a C-VLAN of 111 (normalization) and an S-VLAN of 200.
4. Upon arriving at the remote PEB, assuming the bridge domain is configured for **`vlan-id none`**, the S-VLAN and the C-VLAN tags are popped before the MAC-table lookup.
5. If the destination MAC address is unknown, then the frame is flooded out of all interfaces that associate with the bridge domain, including the subinterfaces of customer-facing interfaces (because of SVL). If the destination MAC address is known, the frame is forwarded out of the appropriate subinterface using the encapsulation specified on the interface.

Explicit Configuration of Tag Operations



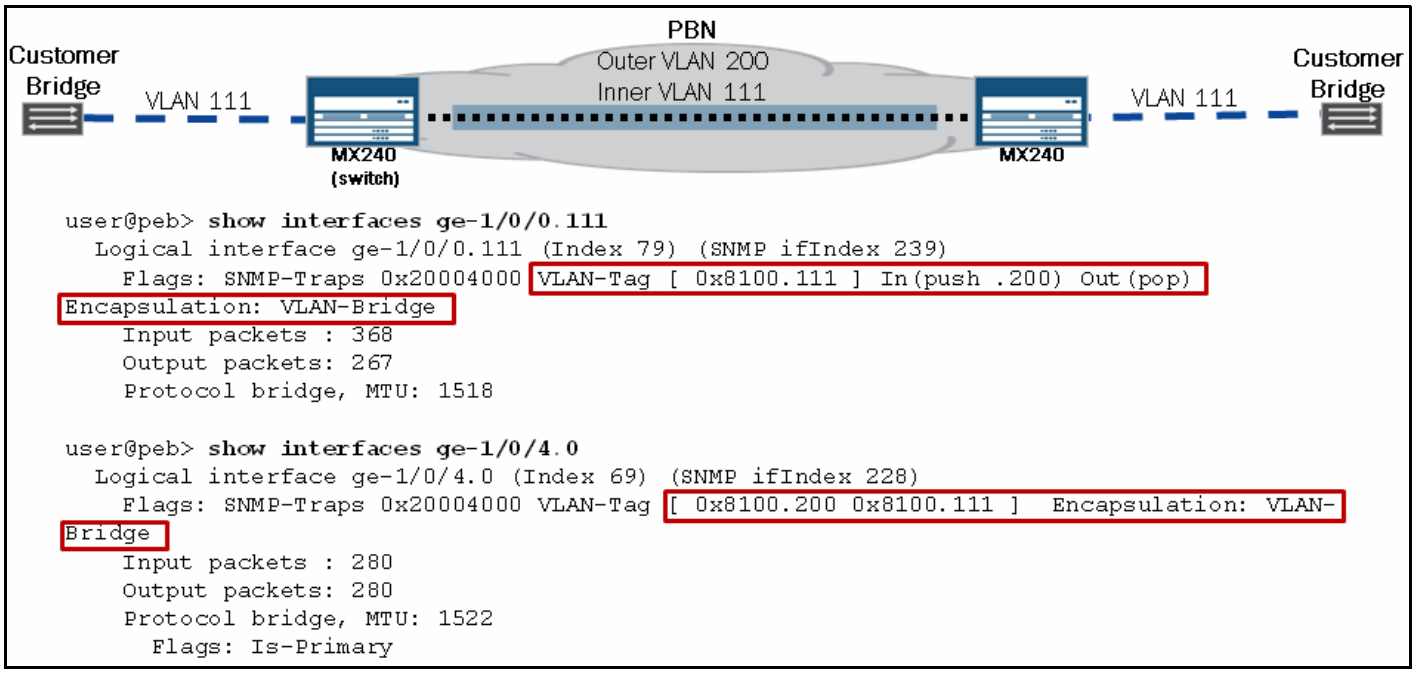
The graphic shows an example of explicitly configuring the VLAN tag operations to be performed on an interface.

PBN Network-to-Network Interface



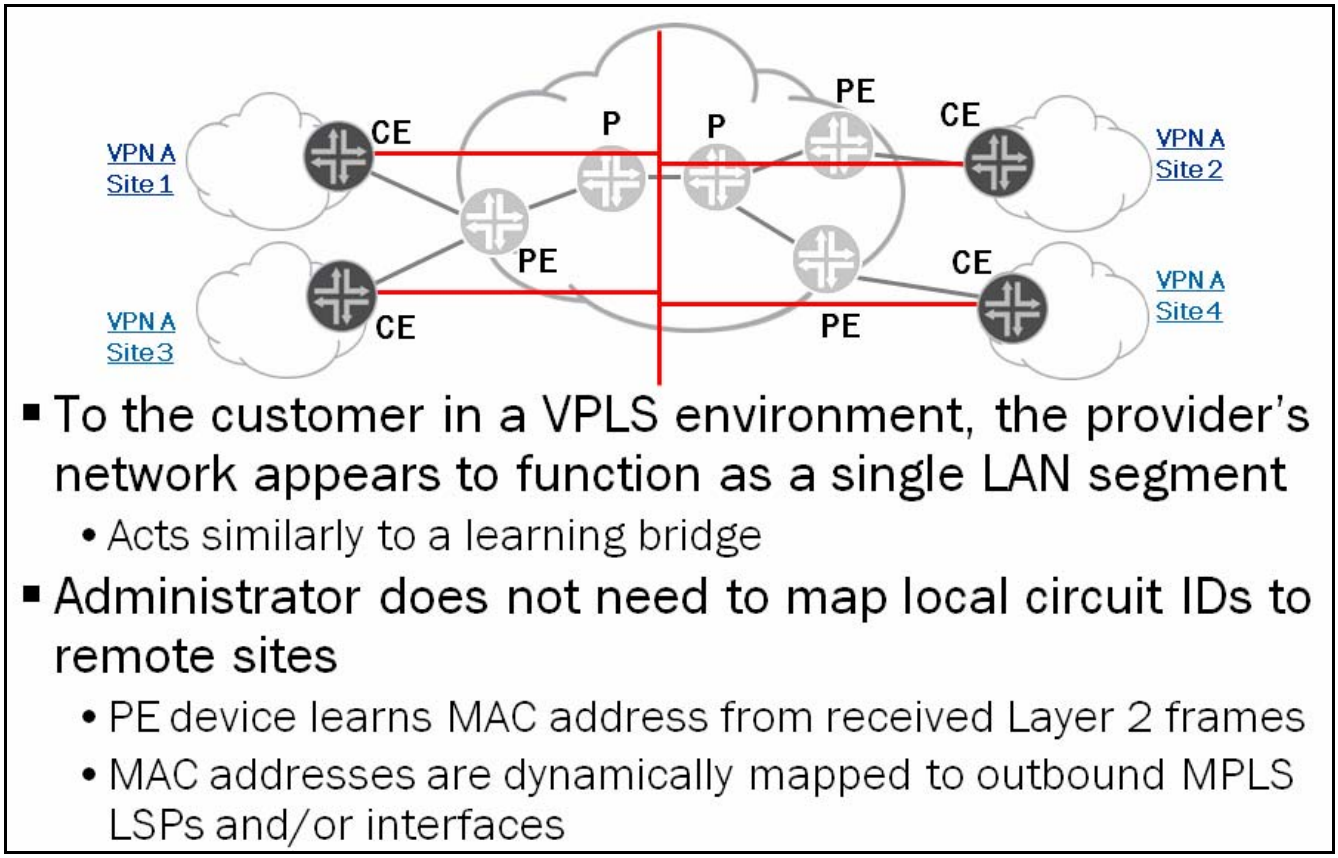
In the graphic, two service providers provide an EVC to a single customer. To allow for the interconnection of the two customer sites, the two service providers must exchange S-VLAN-tagged frames between one another. However, the case might be that each service provider is using a different S-TAG to provide the EVC. In the example, PBN 1 uses S-VLAN 200 and PBN 2 uses S-VLAN 300. IEEE 802.1ad provides the ability to perform S-VLAN translation between service providers. The slide shows the configuration necessary for the S-VLAN bridge in PBN 1 to perform VLAN translation.

View Tag Operation Settings



To view the expected VLAN tag operations that an interface will perform, issue the **`show interfaces`** command. The `VLAN-tag` field shows the VLAN IDs for which the interface was specifically configured. The `In` and `Out` fields show the VLAN operations that the interface will perform.

An Alternative to Q-in-Q Tunneling



Q-in-Q tunneling has some drawbacks:

- Because there are 4096 unique VLANs, the number of customers can be severely limited.
- If there is a network failure, Ethernet's STP can take tens of seconds to find an alternate path. Even the new Rapid Spanning Tree Protocol (RSTP) can take multiple seconds in most situations, and convergence time increases as the network grows.

An alternative to Q-in-Q tunneling that can be provided to the customer is virtual private LAN service (VPLS). VPLS delivers an Ethernet service that can span one or more metro areas and that provides connectivity between multiple sites as if these sites were attached to the same Ethernet LAN. To the customer, a VPLS appears to be a single LAN segment. In fact, it appears to act similarly to a learning bridge. That is, when the destination media access control (MAC) address is not known, an Ethernet frame is sent to all remote sites. If the destination MAC address is known, it is sent directly to the site that owns it. VPLS requires a strong background in MPLS as well as other routing protocols. A full discussion on VPLS is outside the scope of this guide.

Review Questions

1. What are some of the scaling issues that can occur if a service provider were to use IEEE 802.1Q VLANs to provide LAN service to its customers?
2. List three VLAN tag operations that a switch can perform on a frame.
3. Bridge domains can learn in two modes. What are the two modes?

Answers

1.

The service provider and potentially thousands of customers must share a limited number of VLAN IDs when a service provider uses IEEE 802.1Q VLANs to provide LAN service. Also, each service provider switch must learn the MAC addresses of its customers.

2.

Three VLAN tag operations that a switch can perform on a frame are pop, push, and swap.

3.

The two modes are independent VLAN learning mode (IVL) and shared VLAN learning mode (SVL).

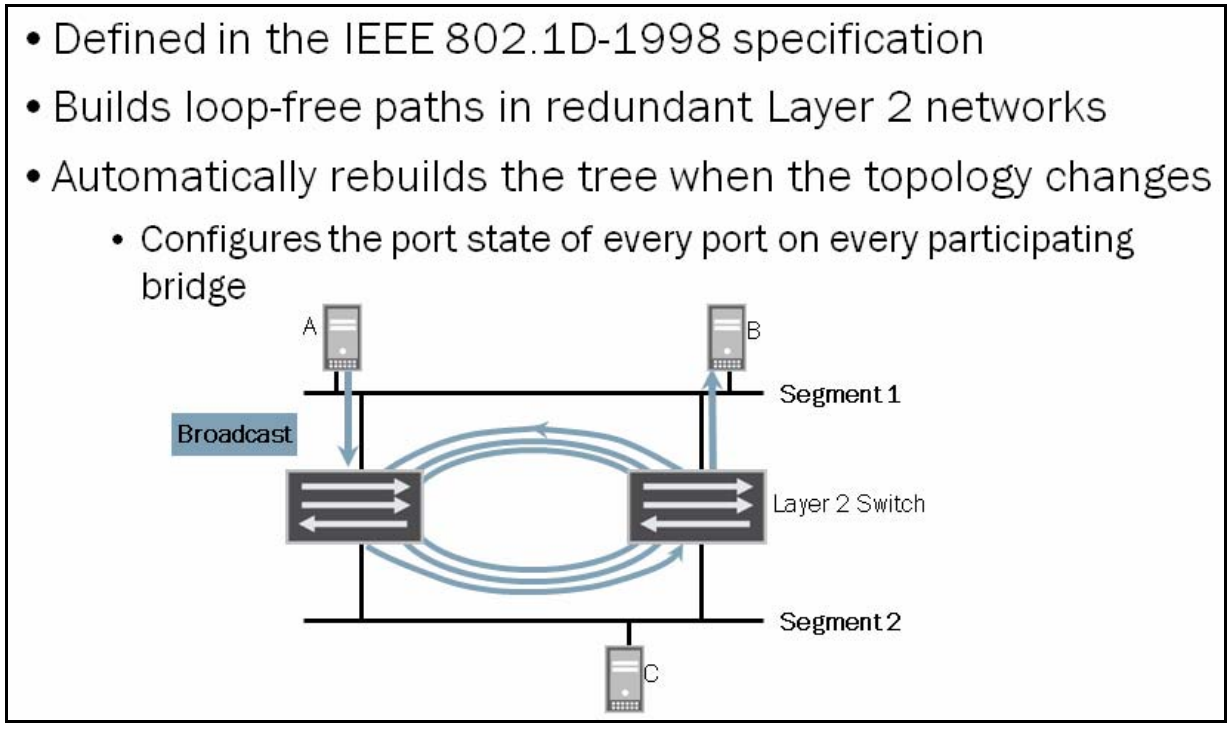
JNCIS-SP Study Guide—Part 2

Chapter 5: Spanning Tree Protocols

This Chapter Discusses:

- The purpose of spanning-tree protocols;
- The basic operation of the Spanning Tree Protocol (STP), the Rapid Spanning Tree Protocol (RSTP), the Multiple Spanning Tree Protocol (MSTP), and the virtual LAN (VLAN) Spanning Tree Protocol (VSTP);
- Configuration and monitoring of STP, RSTP, MSTP, and VSTP; and
- Implementation bridge protocol data unit (BPDU), loop, and root protection.

STP



STP is defined in the Institute of Electrical and Electronics Engineers (IEEE) 802.1D 1998 specification. STP is a simple Layer 2 protocol that prevents loops and calculates the best path through a switched network that contains redundant paths. STP is necessary only when redundant paths exist within a Layer 2 network. STP automatically rebuilds the tree when a topology change occurs.

STP Terms and Concepts

- *Bridge ID*: Unique identifier for each switch
- *Root bridge*: Switch with the lowest bridge ID
- *Root port*: The port on each bridge closest to the root bridge
- *Root path cost*: A bridge's calculated cost to get from itself to the root bridge
  - Equal to the received root path cost from configuration BPDUs plus the port cost of the root port on the bridge
- *Port cost*: Every interface on a bridge has an assigned port cost value
  - Used in the calculation of the root path cost for the local bridge
  - Configurable value (1–200,000,000)
  - The default value is 20,000 for 1 Gigabit Ethernet

All switches participating in STP have a unique bridge ID. The bridge ID is a combination of the system MAC address and a configurable priority value. The lowest bridge ID determines the *root bridge*.

Once the root bridge is determined, each nonroot switch determines the least-cost path from itself to the root bridge. The port associated with the least-cost path, referred to as the *root path cost*, becomes the *root port* for the switch. Every port on a switch has a configurable *port cost* associated with it. A nonroot switch receives periodic STP BPDUs—described on next slide—that contain a root path cost as determined by the neighboring switch. The local switch adds the received root path cost to each of the port costs for its interfaces. Whichever interface is associated with the lowest value (root path cost + port cost) becomes the root port for the switch.

- *Designated bridge*: A switch representing the LAN segment
- *Port ID*: A unique identifier for each port on each switch
- *Designated port*: The designated bridge's forwarding port on a LAN segment
  - The port used by a designated bridge to send traffic from the direction of the root to the LAN or from the LAN toward the root
- *Bridge protocol data unit*: Packets used to exchange information between switches
  - Configuration BPDU
  - Topology change notification BPDU

All switches participating on a common network segment must determine which switch offers the least-cost path from the network segment to the root bridge. The switch with the best path becomes the *designated bridge* for the LAN segment, and the port connecting this switch to the network segment becomes the *designated port* for the LAN segment. If equal-cost paths to the root bridge exist between two or more switches for a given LAN segment, the *bridge ID* acts as a tiebreaker. If the bridge ID is used to help determine the designated bridge, the lowest bridge ID is selected. If two equal-cost paths exist between two ports

on a single switch, then *port ID* acts as the tiebreaker (lower is preferable). The designated port transmits BPDUs on the segment.

Port States

- Blocking
  - The port drops all data packets and listens to BPDUs
  - The port is not used in active topology
- Listening
  - The port drops all data packets and listens to BPDUs
  - The port is transitioning and will be used in active topology
- Learning
  - The port drops all data packets and listens to BPDUs
  - The port is transitioning and the switch is learning MAC addresses
- Forwarding
  - The port receives and forwards data packets and sends and receives BPDUs
  - The port has transitioned and the switch continues to learn MAC addresses

The graphic highlights the STP port states along with a brief description of each state. In addition to the states listed on the slide, an interface can have STP administratively disabled (default behavior). An administratively disabled port does not participate in the spanning tree but does flood any BPDUs it receives to other ports associated with the same VLAN. Administratively disabled ports continue to perform basic bridging operations and forward data traffic based on the MAC address table.

BPDU Ethernet Frame

- Ethernet frame:

  - Source address—The outgoing port of the originating switch
  - Destination address—The bridge group address (01:30:C2:00:00:00)
  - Length
  - LLC header
    - DSAP and SSAP = 0x42 (Bridge Spanning Tree Protocol)

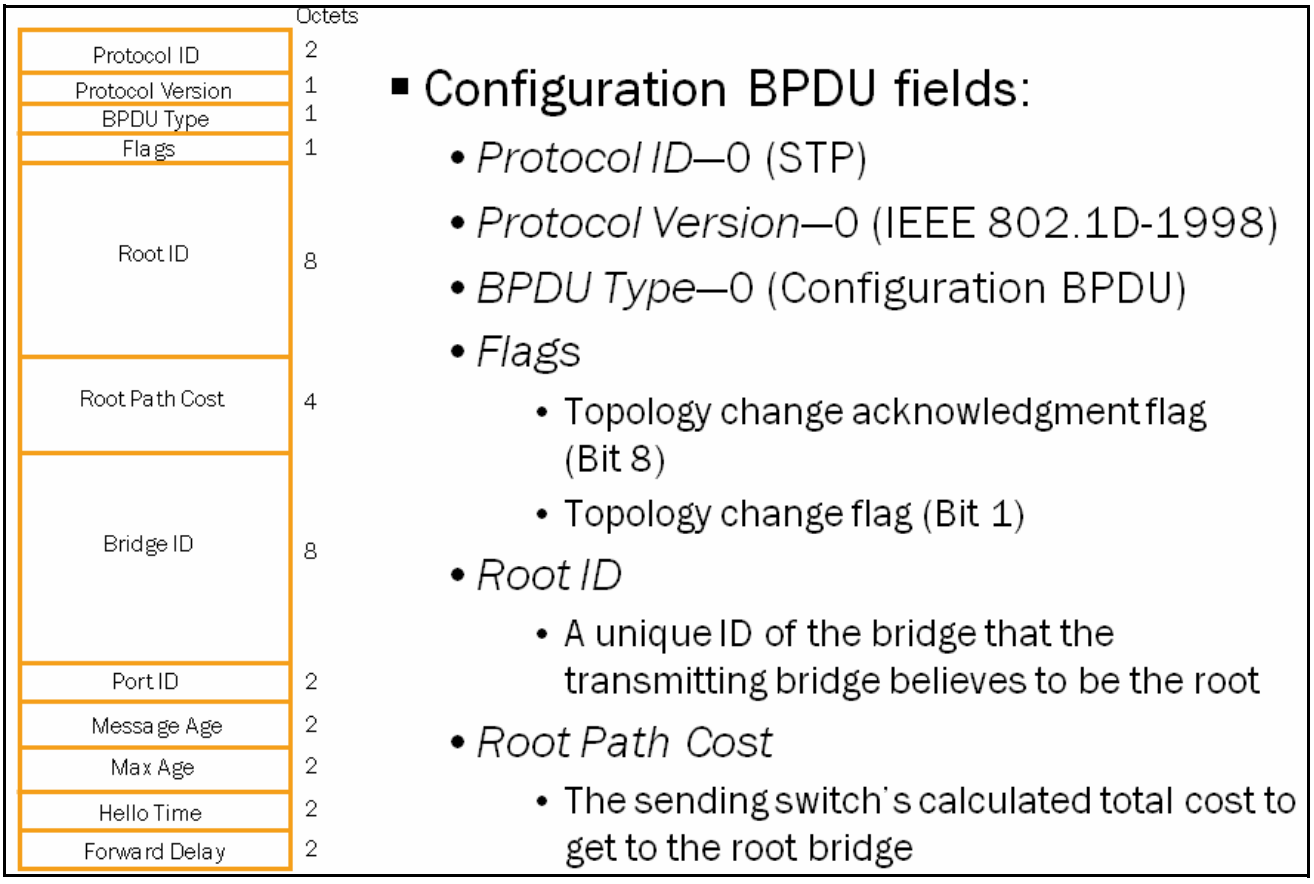
The graphic shows the Ethernet frame format of an STP BPDU. Notice that the Ethernet frame does not contain any 802.1-type VLAN tagging. The source address of the frame is the MAC address of the outgoing port of the sending switch. The destination address is the multicast MAC address that is reserved for STP. The frame also contains an LLC header that uses a destination service access point (DSAP) of 0x42, which refers to the bridge STP.

BPDU Types

- BPDU types:
  - Configuration BPDUs
    - Used to build the spanning-tree topology
  - Topology change notification BPDUs
    - Reports topology changes

STP uses BPDU packets to exchange information between switches. Two types of BPDUs exist: configuration BPDUs and topology change notification (TCN) BPDUs. Configuration BPDUs determine the tree topology of a LAN. STP uses the information that the BPDUs provide to elect a root bridge, to identify root ports for each switch, to identify designated ports for each physical LAN segment, and to prune specific redundant links to create a loop-free tree topology. TCN BPDUs report topology changes within a switched network.

Configuration BPDU Format

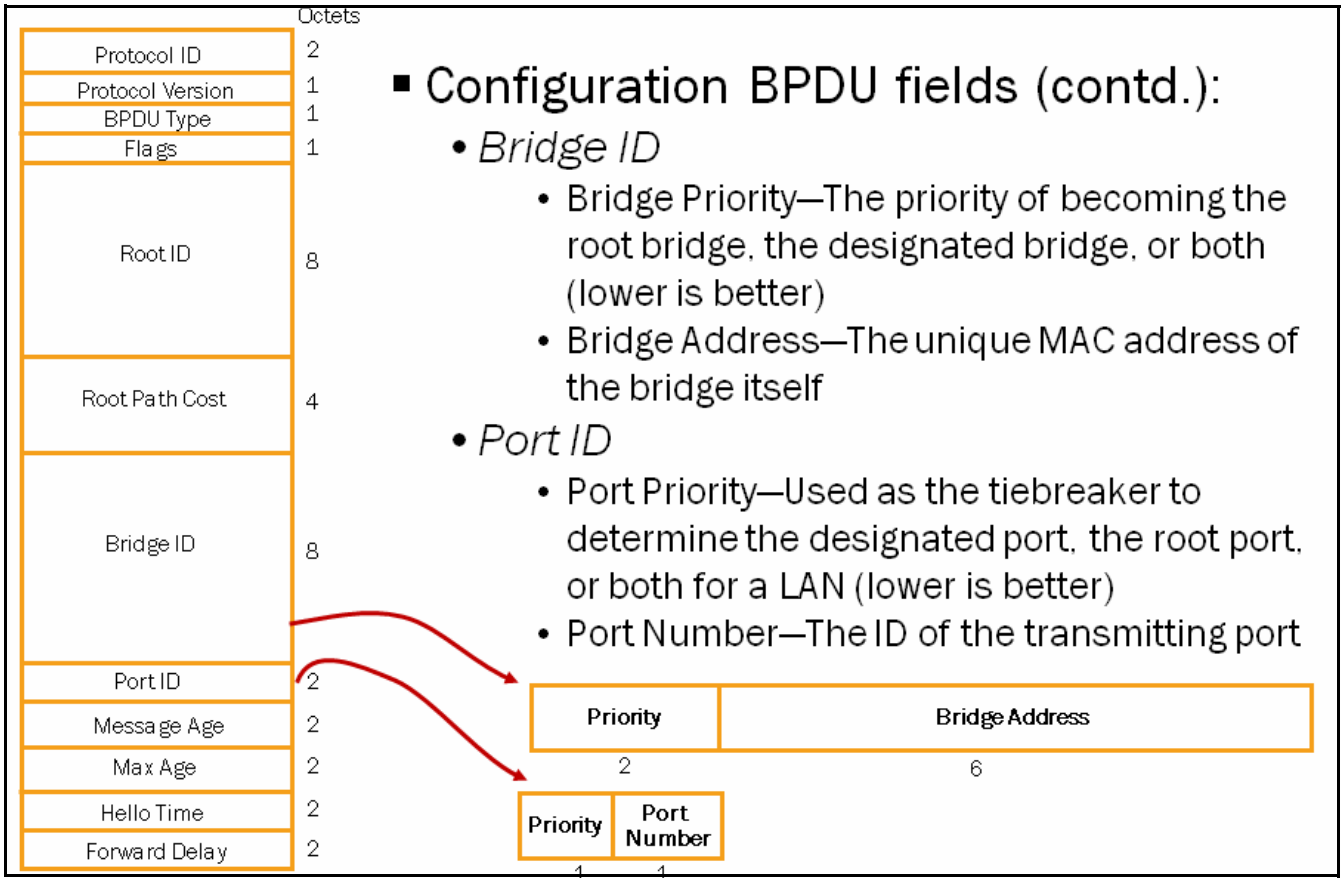


When an STP network is first turned up, all participating bridges send out configuration BPDUs to advertise themselves as candidates for the root bridge. Each bridge uses the received BPDUs to help build the spanning tree and elect the root bridge, root ports, and designated ports for the network. Once the STP network converges and is stable, the root bridge sends a configuration BPDU once every few seconds (the hello time default is 2 seconds).

The following list provides a brief explanation of each of the BPDU fields:

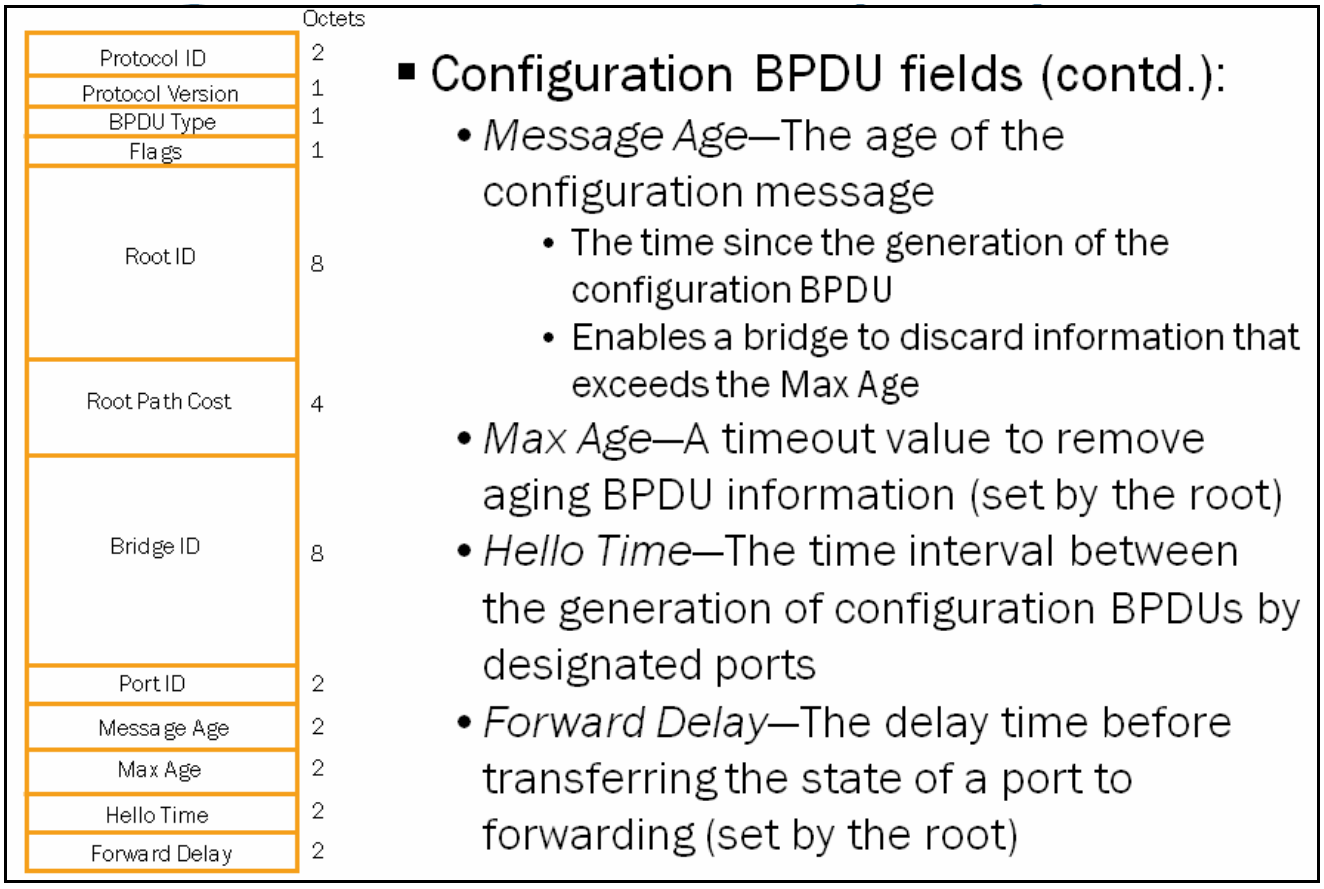
- *Protocol ID*: This value is always 0.
- *Protocol Version*: This value is always 0.

- *BPDU Type*: This field determines which of the two BPDU formats this frame contains—configuration BPDU or TCN BPDU.
- *Flags*: This field is used to handle changes in the active topology; we discuss this field later.
- *Root ID*: This field contains the bridge ID (BID) of the root bridge. After convergence, all configuration BPDUs in the bridged network should contain the same value for this field (for a single VLAN). Some network sniffers break out the two BID subfields: bridge priority and bridge MAC address.
- *Root Path Cost*: This value is the cumulative cost of all links leading to the root bridge.



The following list is a continuation of the explanation of BPDU fields:

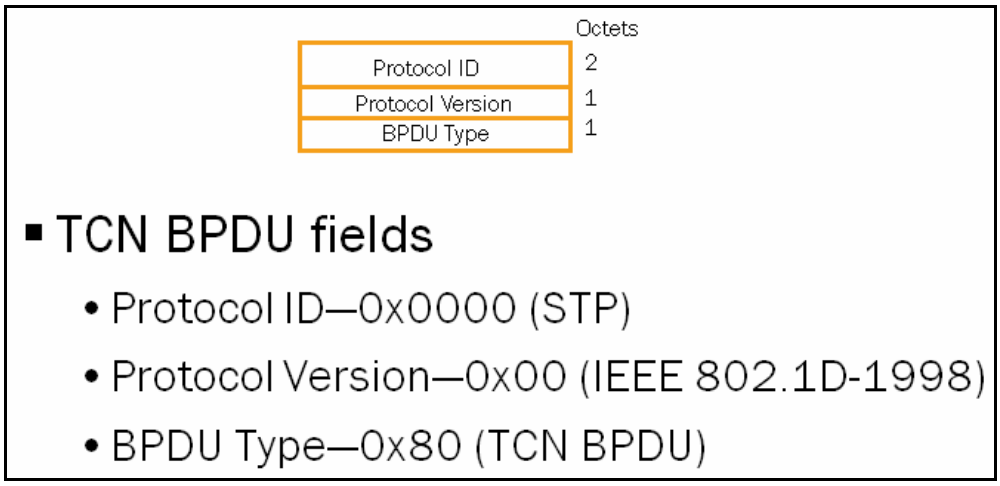
- *BID*: This value is the BID of the bridge that created the current BPDU. This field is the same for all BPDUs sent by a single switch (for a single VLAN), but it differs between switches. The BID is a combination of the sender bridge's priority to become root or designated bridge and the bridge address (a unique MAC address for the bridge.)
- *Port ID*: This field contains a unique value for every port. This value is a combination of the outbound port's priority and a unique value to represent the port. The default port priority is 128 for every interface on an MX Series 3D Universal Edge Router. The switch automatically generates the port number and you cannot configure it. For example, ge-1/0/0 contains the value 128:41, whereas ge-1/0/1 contains the value 128:42.



The following list is a continuation of the explanation of BPDU fields:

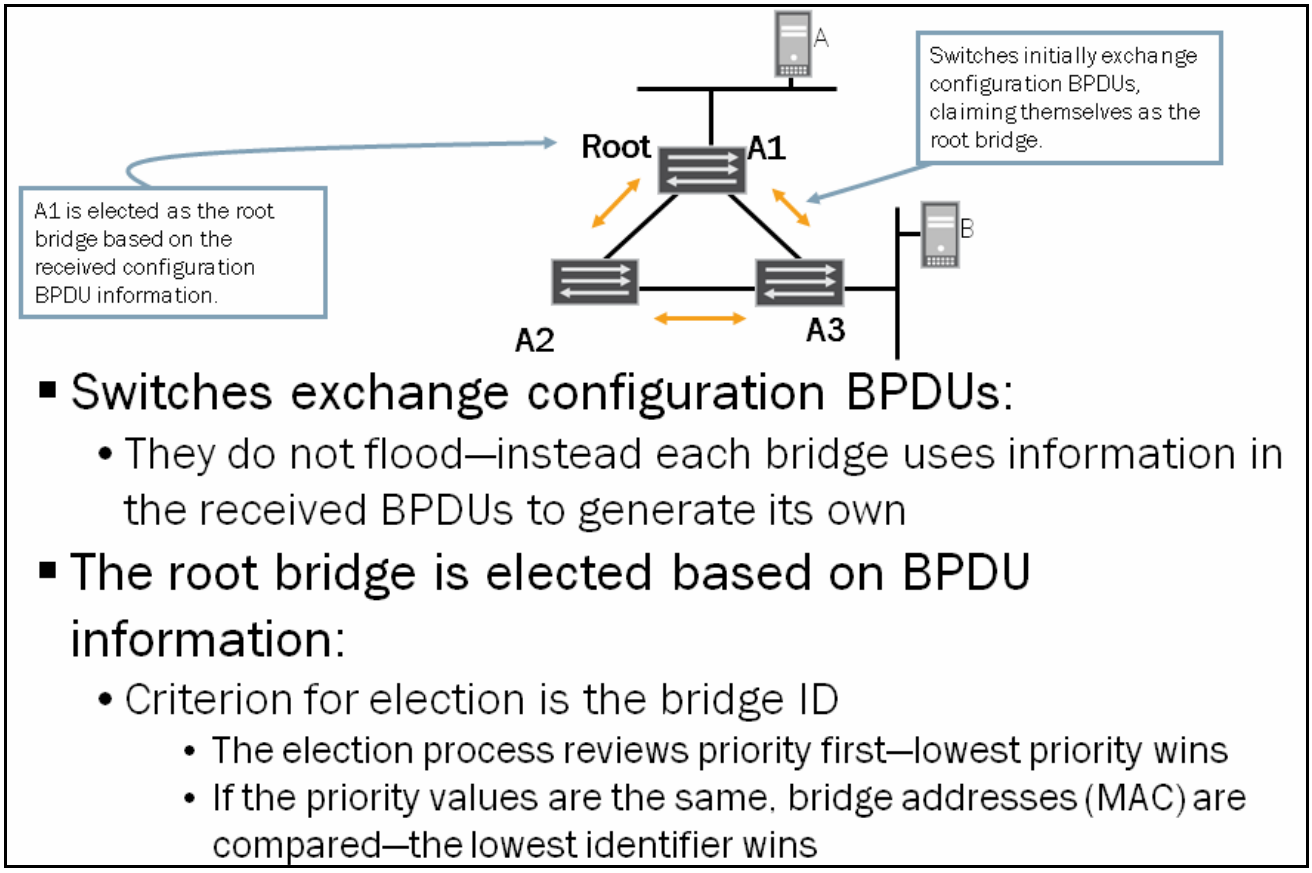
- *Message Age*: This field records the time since the root bridge originally generated the information from which the current BPDU is derived.
- *Max Age*: This value is the maximum time that a BPDU is saved. It also influences the bridge table aging timer during the topology change notification process.
- *Hello Time*: This value is the time between periodic configuration BPDUs.
- *Forward Delay*: This value is the time a bridge spends in the listening and learning states. It also influences timers during the topology change notification process.

TCN BPDU



The graphic shows the format of the TCN BPDU. We describe its usage later in this content.

Exchange of BPDUs



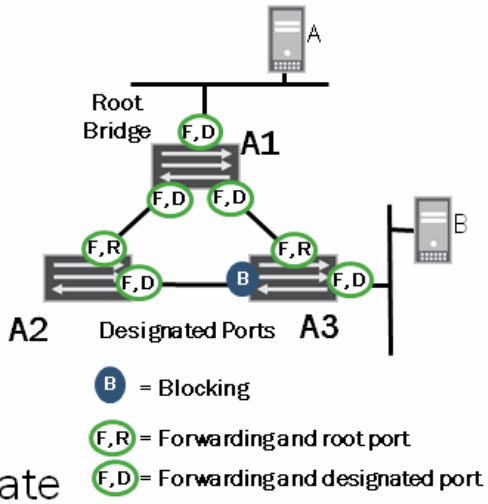
All switches participating in a switched network exchange BPDUs with each other. Through the exchanged BPDUs, neighboring switches become familiar with each other and learn the information necessary to select a root bridge. Each bridge creates its own configuration BPDUs based upon the BPDUs that it receives from neighboring routers. Non-STP bridges simply flood BPDUs as they would any multicast Ethernet frames.

Root Bridge Election

STP elects the root bridge device based on the BID, which actually consists of two distinct elements: a configurable priority value and a unique device identifier, which is the system MAC address. Each bridge reviews the priority values first and determines the root bridge. If the priority value of one device is lower than the priority value of all other devices, that device receives the root bridge election. If the priority values are equal for all devices, STP evaluates the bridge addresses (MAC), and each bridge elects the device with the lowest MAC address as the root bridge.

Port Role and State Determination

- The least-cost path calculation to the root bridge determines the port role; the port role determines the port state:
  - Ports on the root bridge assume the designated port role and forwarding state
  - Root ports on switches are placed in the forwarding state
    - The root bridge will have no root ports
  - Designated ports on designated bridges are placed in the forwarding state



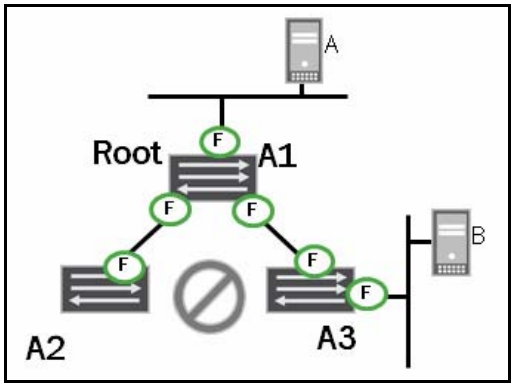
Once root bridge election occurs, all nonroot devices perform a least-cost path calculation to the root bridge. The results of these calculations determine the role of the switch ports. The role of the individual switch ports determines the port state.

All switch ports belonging to the root bridge assume the designated port role and forwarding state. Each nonroot switch determines a root port, which is the port closest to the root bridge, based on its least-cost path calculation to the root bridge. Each interface has an associated cost that is based on the configured speed. An interface operating at 10 Mbps assumes a cost of 2,000,000, an interface operating at 100 Mbps assumes a cost of 200,000, an interface operating at 1 Gbps assumes a cost of 20,000, and an interface operating at 10 Gbps assumes a cost of 2000. If a switch has two equal-cost paths to the root bridge, the switch port with the lower port ID is selected as the root port. The root port for each nonroot switch is placed in the forwarding state.

STP selects a designated bridge on each LAN segment. This selection process is also based on the least-cost path calculation from each switch to the root bridge. Once the designated bridge selection occurs, its port, which connects to the LAN segment, is chosen as the designated port. If the designated bridge has multiple ports connected to the LAN segment, the port with the lowest ID participating on that LAN segment is selected as the designated port. All designated ports assume the forwarding state. All ports not selected as a root port or as a designated port assume the blocking state. While in blocked state, the ports do not send any BPDUs. However, they listen for BPDUs.

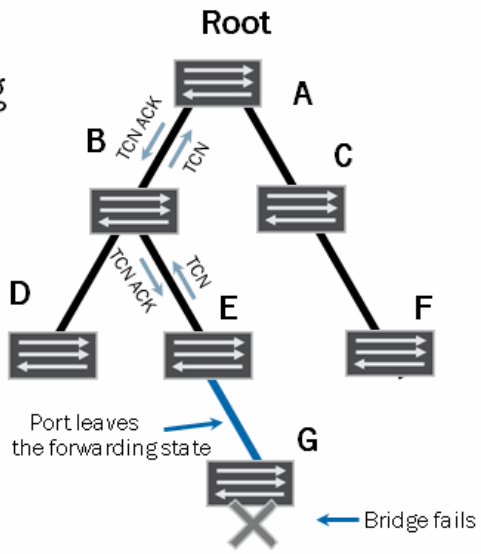
Full Tree Convergence

Once each bridge determines the role and state for all switch ports, the tree is considered fully converged. The convergence delay can take up to 50 seconds when the default forwarding delay (15 seconds) and max age timer (20 seconds) values are in effect. The formula to calculate the convergence delay for STP is 2x the forwarding delay + the maximum age. In the example shown on the slide, all traffic passing between Host A and Host B transits the root bridge (Switch A1).

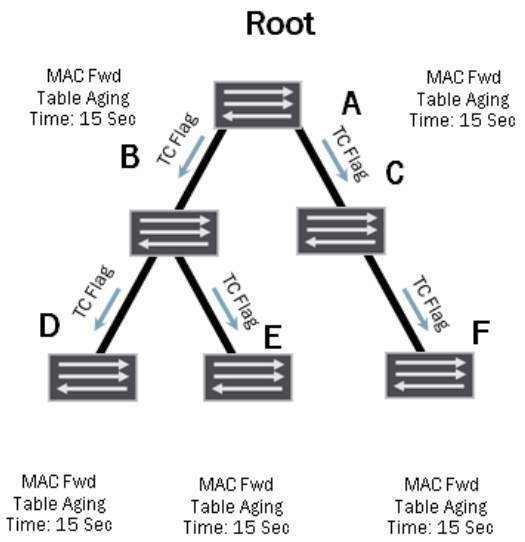


Reconvergence Example

1. Bridge G fails
2. Bridge E's port leaves the forwarding state
3. Bridge E sends a TCN
   a. The TCN always travels out the root port;
   It continues every 2 seconds until the root port from B receives the TCN ACK in the form of a configuration BPDU
4. Bridge B sends a TCN ACK
5. Bridge B sends a TCN out of the root port
6. Bridge A sends a TCN ACK



7. The root bridge sets the topology change flag and sends an updated configuration BPDU
8. Bridges B and C relay the topology change flag to downstream switches
9. All nonroot bridges change the MAC address forwarding table aging timer to equal the forwarding delay time (default: 15 seconds)



The graphics show the steps involved in a failure and reconvergence scenario. Once the nonroot bridges change their MAC address forwarding table aging timer to the shortened interval and wait that period of time (15 seconds by default), they then delete all entries from the MAC table that were not refreshed within that time frame. All deleted entries must then be learned once again through the normal learning process.

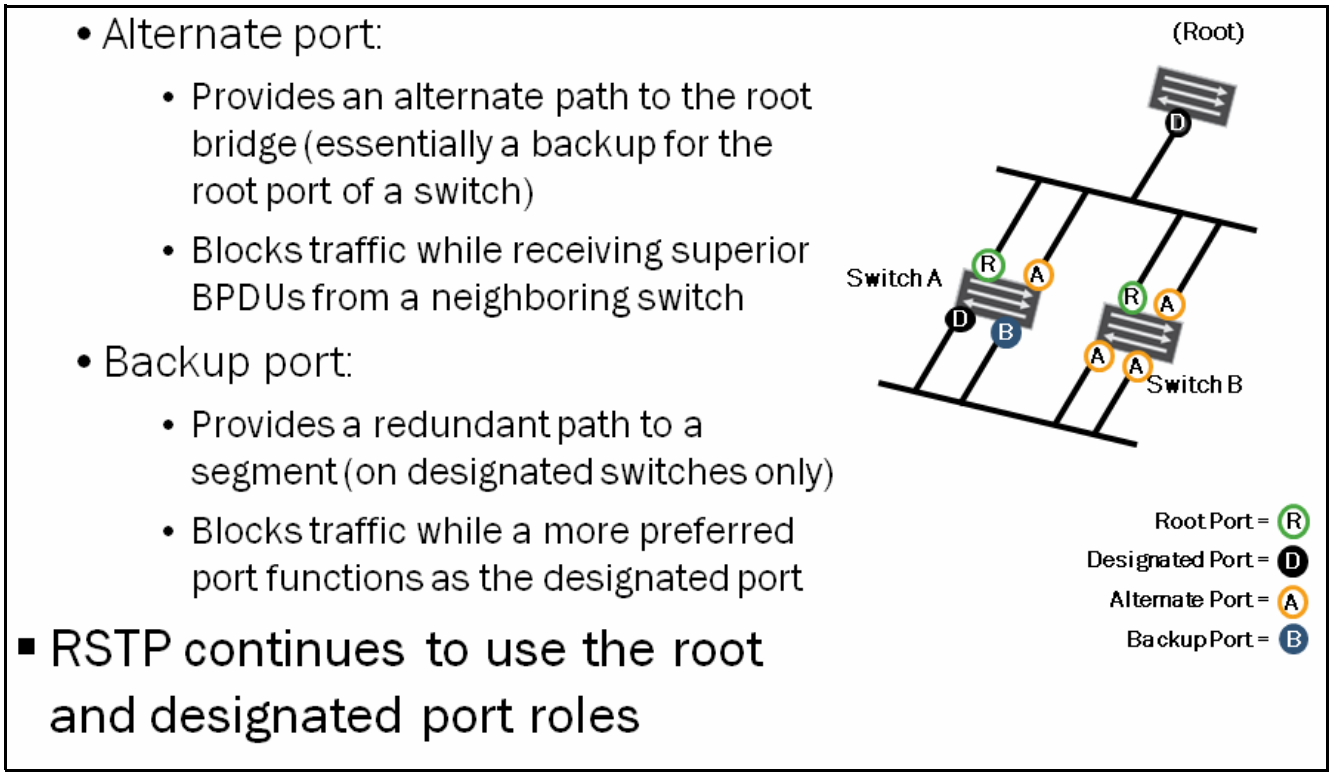
RSTP Defined

RSTP was originally defined in the IEEE 802.1w draft and was later incorporated into the IEEE 802.1D-2004 specification. RSTP introduces a number of improvements to STP while performing the same basic function.

RSTP Convergence Improvements

RSTP provides better reconvergence time than the original STP. RSTP identifies certain links as point-to-point. When a point-to-point link fails, the alternate link can transition to the forwarding state without waiting for any protocol timers to expire. RSTP provides fast network convergence when a topology change occurs and it greatly decreases the state transition time compared to STP. To aid in the improved convergence, RSTP uses additional features and functionality, such as edge port definitions and rapid direct and indirect link failure detection and recovery. We examine these features in more detail later in this chapter.

RSTP Introduces New Port Roles



RSTP introduces the alternate and backup port roles. An alternate port is a switch port that has an alternate—generally higher-cost—path to the root bridge. In the event that the root port fails, the alternate port assumes the role of the root port and is placed in the forwarding state. Alternate ports are placed in the discarding state but receive superior BPDUs from neighboring switches. Alternate ports are found on switches participating in a shared LAN segment for which they are not functioning as the designated bridge.

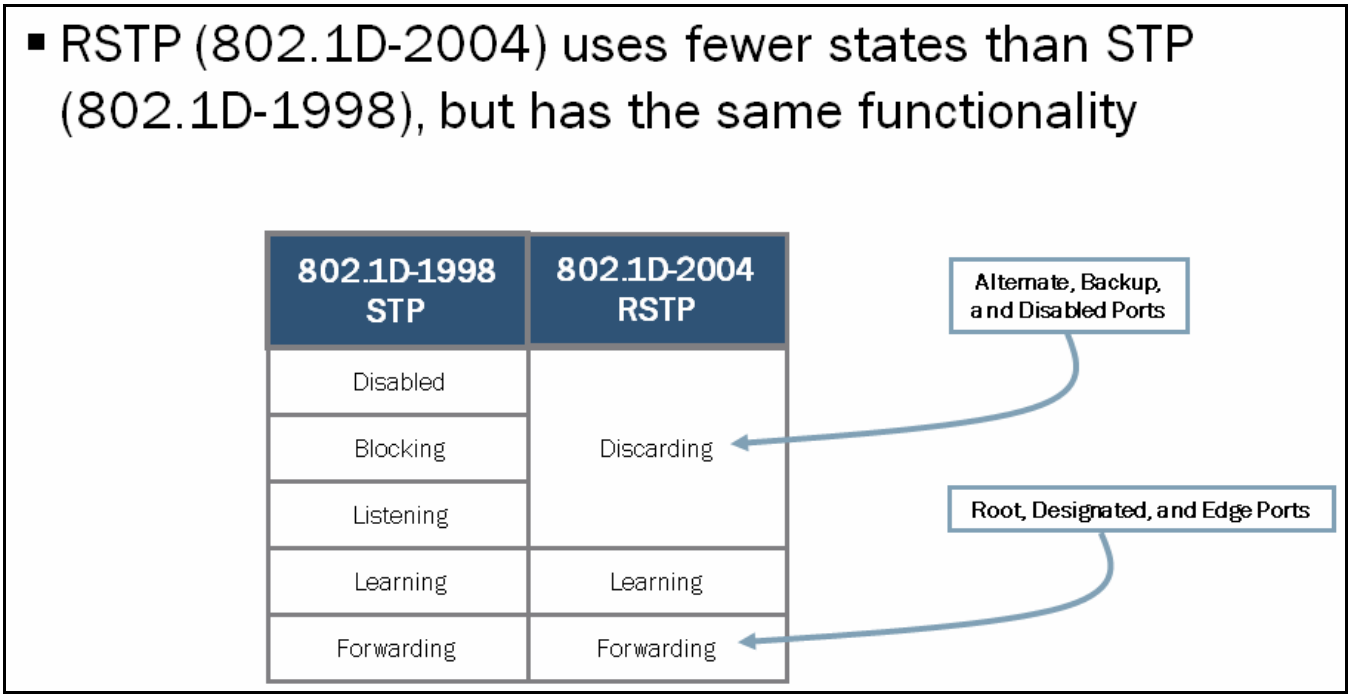
When a designated bridge has multiple ports connected to a shared LAN segment, it selects one of those ports as the designated port. The designated port is typically the port with the lower port ID. RSTP considers all other ports on the designated switch that connects to that same shared LAN segment as backup ports. In the event that the designated port is unable to perform its role, one of the backup ports assumes the designated port role upon successful negotiation and it is placed in the forwarding state.

Backup ports are placed in the discarding state. While in the discarding state, backup ports receive superior BPDUs from the designated port.

Continued Use of Root and Designated Ports

RSTP continues to use the root and designated port roles. Only ports selected for the root port or designated port role participate in the active topology. We described the purpose of the root port and designated ports previously in this chapter.

RSTP Port States



RSTP uses fewer port states than STP. The three possible port states found in RSTP are *discarding*, *learning*, and *forwarding*. Any administratively disabled port excluded from the active topology through configuration, or dynamically excluded from forwarding and learning, is placed in the discarding state. Ports that are actively learning but not currently forwarding are in the learning state, whereas ports that are both learning and forwarding frames simultaneously are in the forwarding state. As the slide indicates, only those ports selected as root ports and designated ports use the forwarding state.

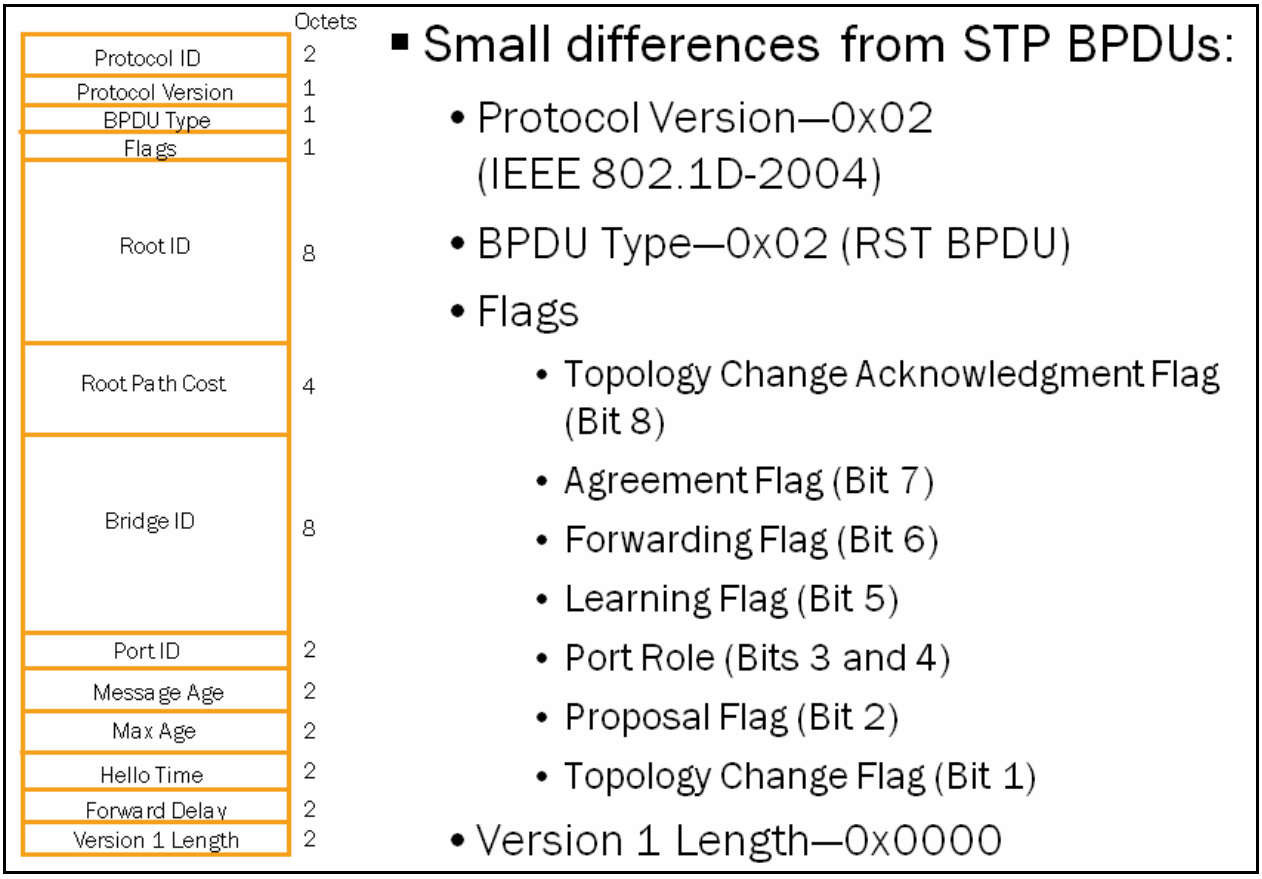
RSTP BPDUs

As previously mentioned, STP uses BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, prune specific redundant links to create a loop-free tree topology, and report and acknowledge topology changes. RSTP configuration BPDUs also function as keepalives. All RSTP bridges send configuration BPDUs every 2 seconds by default. You can alter this value, if necessary.

By monitoring neighboring switches through the use of BPDUs, RSTP can detect failures of network components much more quickly than STP can. If a neighboring bridge receives no BPDU within three times the hello interval, it assumes connectivity is faulty and updates the tree. By default, it detects failures within 6 seconds when using RSTP, whereas it might take up to 50 seconds when using STP.

On MX Series devices, Ethernet interfaces operating in full-duplex mode are considered point-to-point links. When a failure occurs, a switch port operating as a point-to-point link can become a new root port or designated port and transition to the forwarding state without waiting for the timer to expire. Switch ports operating in half-duplex mode are considered to be shared (or LAN) links and must wait for the timer to expire before transitioning to the forwarding state.

Configuration BPDU Differences

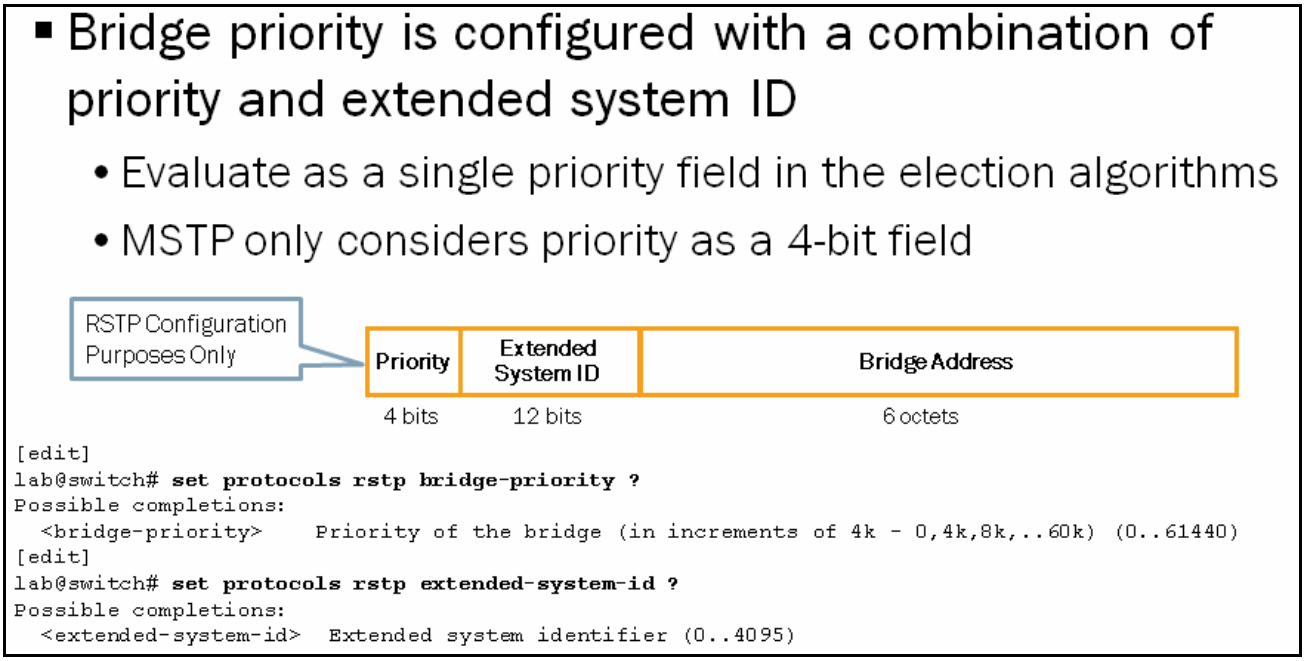


RSTP is backward compatible with STP. If a device configured for RSTP receives STP BPDUs, it reverts to STP. In a pure RSTP environment, a single type of the BPDU exists named Rapid Spanning Tree BPDU (RST BPDU). RST BPDUs use a similar format to the STP configuration BPDUs. RSTP devices detect the type of BPDU by looking at the protocol version and BPDU type fields. The BPDUs contain several new flags, as shown on the slide. The following is a brief description of the flags:

- TCN Acknowledgment: This flag is used when acknowledging STP TCNs;
- Agreement and Proposal: These flags are used to help quickly transition a new designated port to the forwarding state;
- Forwarding and Learning: These flags are used to advertise the state of the sending port;
- Port Role: This flag specifies the role of the sending port: 0 = Unknown, 1 = Alternate or Backup, 2 = Root, and 3= Designated; and
- Topology Change: RSTP uses configuration BPDUs with this bit set to notify other switches that the topology has changed.

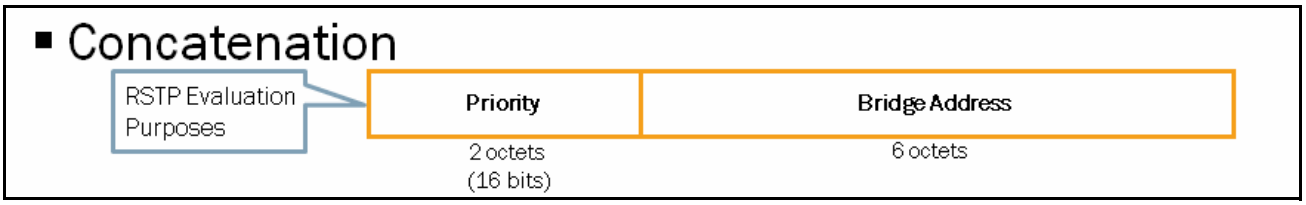
RST BPDUs contain a Version 1 Length field that is always set to 0x0000. This field allows for future extensions to RSTP.

Bridge Priority Configuration



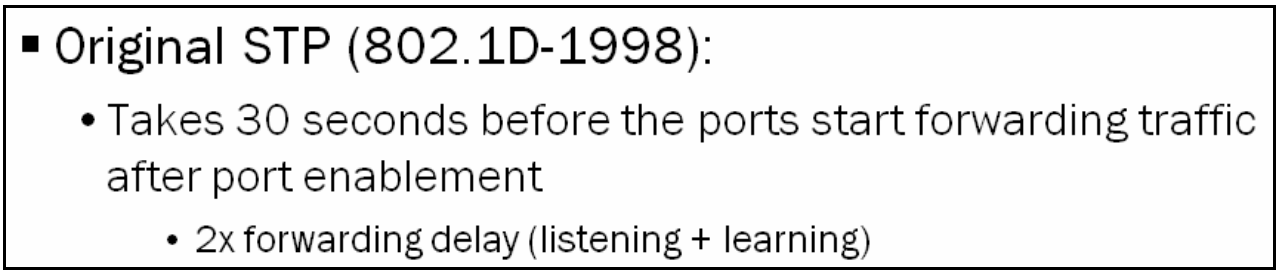
Over time, 16 bits was determined to be too big to represent a bridge's priority for becoming the root or designated bridge. With the advent of MSTP (covered later in this chapter), the older 16-bit priority field was broken into two separate fields: a 4-bit priority field and a 12-bit Extended System ID field. RSTP allows for the configuration of both values. MSTP automatically populates the extended system ID field with a VLAN ID.

RSTP Bridge Priority Evaluation



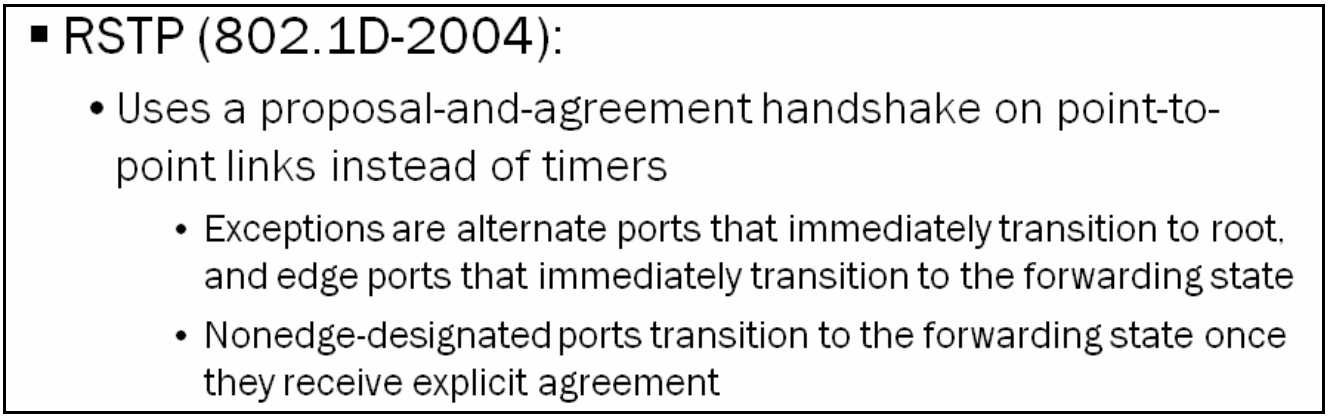
Although priority and extended system ID are configured separately, RSTP evaluates a bridge's priority for the root and designated bridge election process by concatenating the two fields together into a single value. That is, it continues to be a 16-bit field during the election process.

STP Forwarding State Transition



With the original STP, as defined in 802.1D-1998, a port can take more than 30 seconds before it forwards user traffic. As a port is enabled, it must transition through the listening and learning states before graduating to the forwarding state. STP allows two times the forwarding delay (15 seconds by default) for this transition to occur.

RSTP Forwarding State Transition

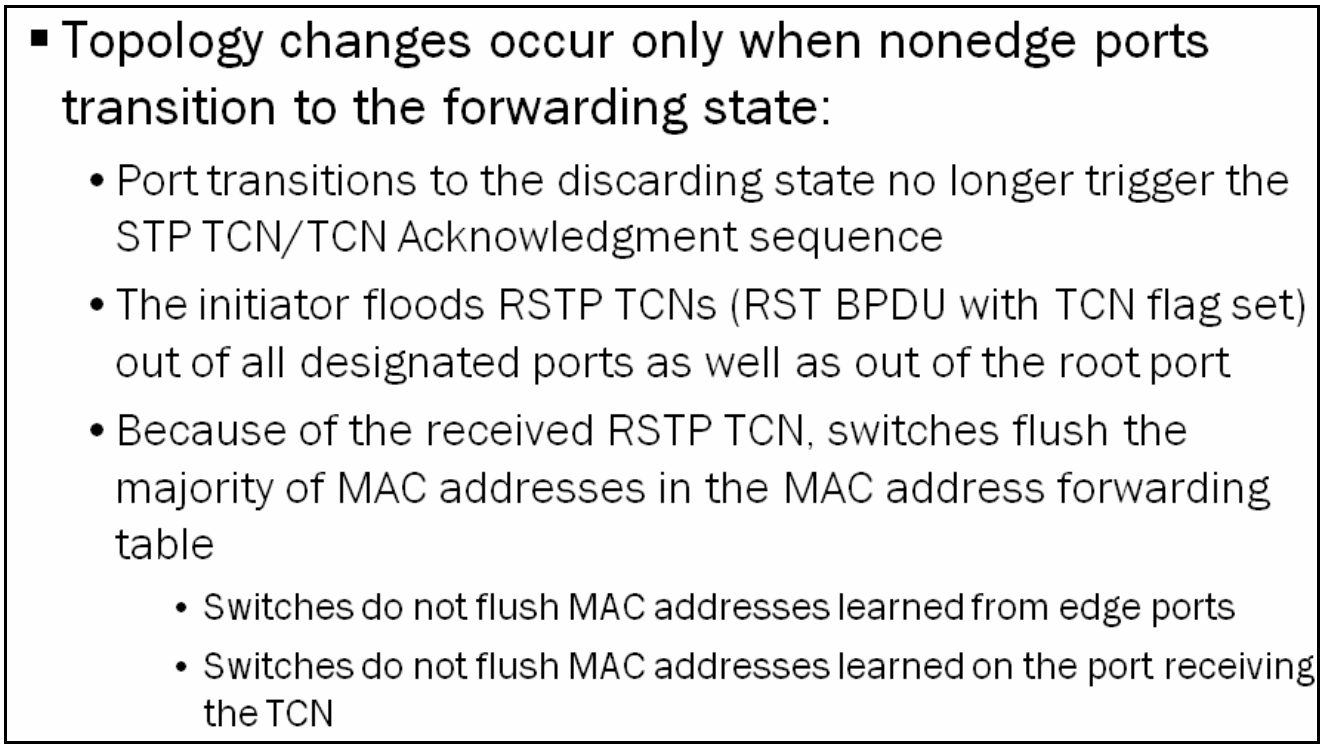


RSTP offers considerable improvements when transitioning to the forwarding state. RSTP converges faster because it uses a proposal-and-agreement handshake mechanism on point-to-point links instead of the timer-based process used by STP. On MX Series devices, network ports operating in full-duplex mode are considered point-to-point links, whereas network ports operating in half-duplex mode are considered shared (LAN) links.

Root ports and edge ports transition to the forwarding state immediately without exchanging messages with other switches. Edge ports are ports that have direct connections to end stations. Because these connections cannot create loops, they are placed in the forwarding state without any delay. If a switch port does not receive BPDUs from the connecting device, it automatically assumes the role of an edge port. When a switch receives configuration messages on a switch port that is configured to be an edge port, it immediately changes the port to a normal spanning-tree port (nonedge port).

Nonedge-designated ports transition to the forwarding state only after receipt of an explicit agreement from the attached switch.

Topology Changes

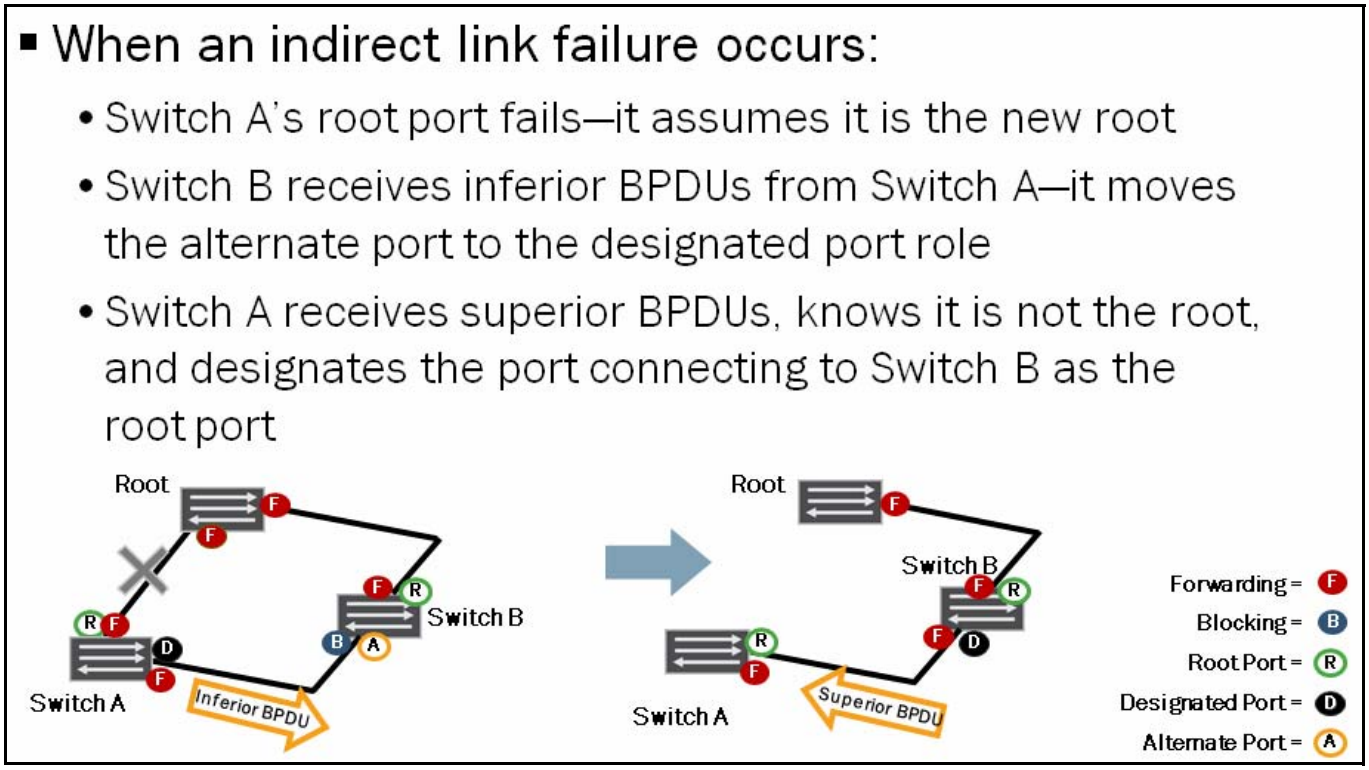


When using STP, state transitions on any participating switch port cause a topology change to occur. RSTP reduces the number of topology changes and improves overall stability within the network by generating TCNs only when nonedge ports transition to the forwarding state. Nonedge ports are typically defined as ports that interconnect switches. Edge ports are typically defined as ports that connect a switch to end stations.

RSTP also provides improved network stability because it does not generate a TCN when a port transitions to the discarding state. With RSTP, TCNs are not generated when a port is administratively disabled, excluded from the active topology through configuration, or dynamically excluded from forwarding and learning.

When a TCN is necessary and is generated, the initiating device floods all designated ports as well as the root port. Unlike traditional STP, neighboring switches that are not in the path of the initiator to the root bridge do not need to wait for this information from the root bridge. As the changes propagate throughout the network, the switches flush the majority of the MAC addresses located in their MAC address forwarding tables. The individual switches do not, however, flush MAC addresses learned from their locally configured edge ports.

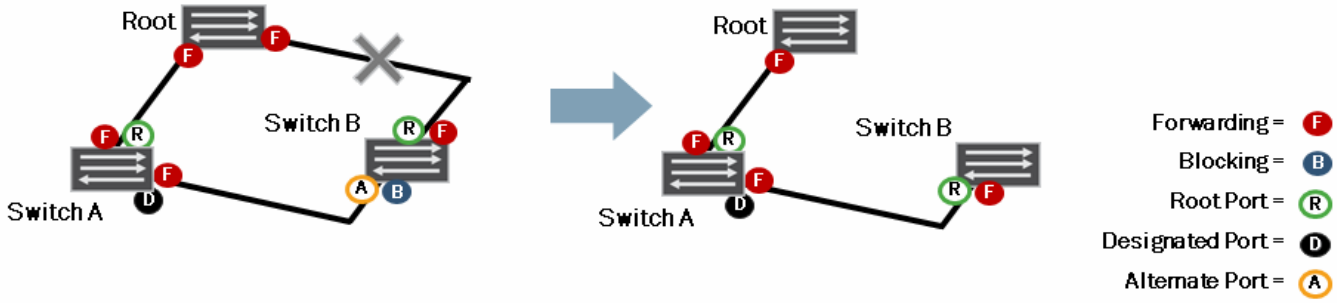
Indirect Link Failure



RSTP performs rapid recovery for link failures. The graphic illustrates a typical scenario for an indirect link failure.

Direct Link Failure

- When a direct link failure occurs:
  - The alternate port transitions to the forwarding state and assumes the new root port role following the failure of the old root port
  - Switch B signals upstream switches to flush their MAC tables by sending RSTP TCNs out of the new root port
    - Upstream switches only flush MAC entries that they learned on active ports that did not receive the RSTP TCNs (except edge ports)



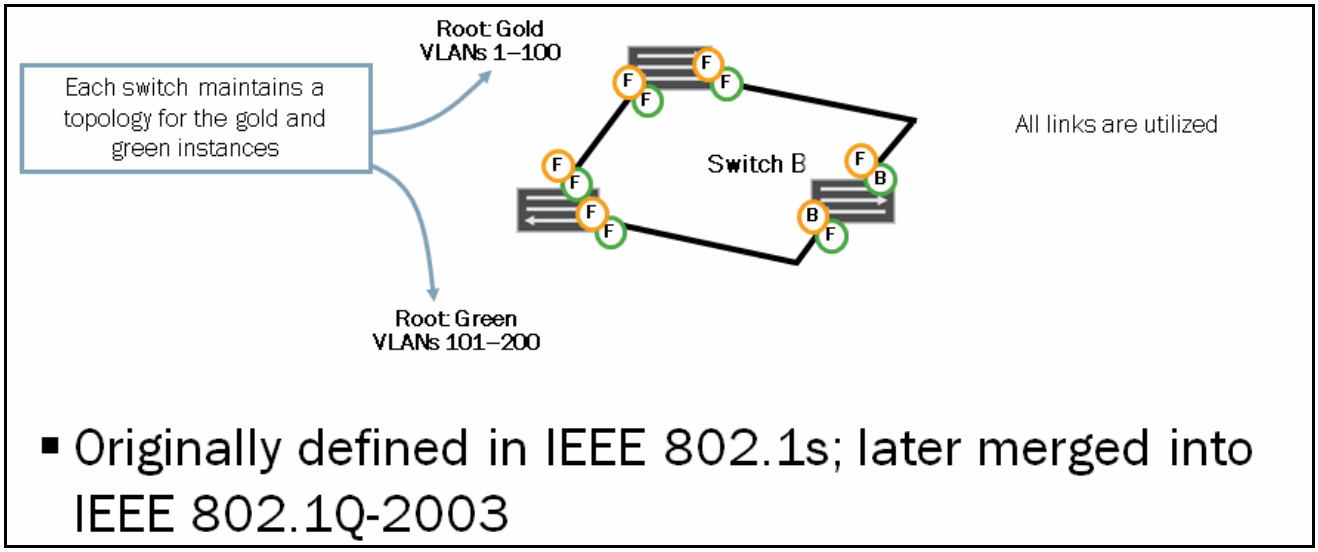
The graphic illustrates a typical scenario in which a direct link failure occurs.

Interoperability Considerations

- STP and RSTP interoperability considerations:
  - If a switch supports only the 802.1D-1998 STP protocol, it discards any RSTP BPDUs it receives
  - If an RSTP-capable switch receives 802.1D-1998 BPDUs, it reverts to 802.1D-1998 STP mode on the receiving interface only
    - Uses STP BPDUs

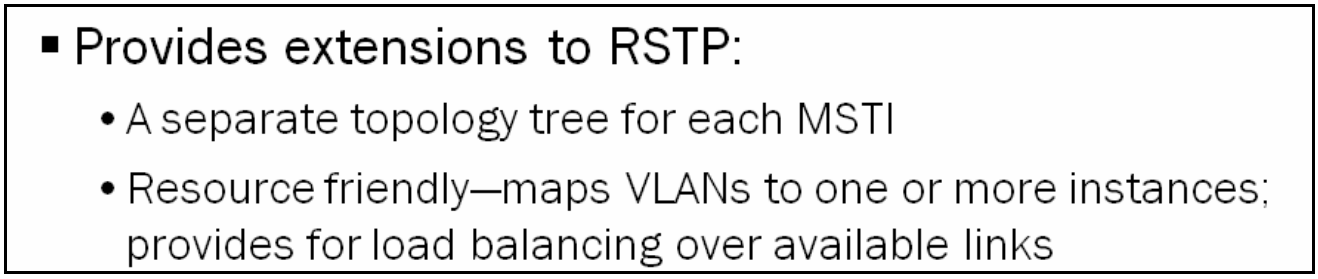
Switches configured for STP and RSTP interoperate with one another. However, you should keep a few basic considerations in mind. If a switch supports only STP and interconnects with a switch running RSTP, it discards the RSTP BPDUs. The RSTP-capable switch, upon receiving STP BPDUs, reverts to STP mode, thus allowing interoperability between the two devices.

MSTP Defined



MSTP was originally defined in the IEEE 802.1s draft and later incorporated into the IEEE 802.1Q-2003 specification.

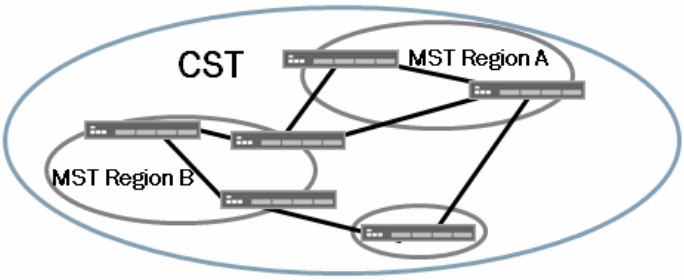
MSTP Enhancements over RSTP



Although RSTP provides faster convergence than STP, it still does not make good use of all available paths within a redundant Layer 2 network. With RSTP, all traffic from all VLANs follows the same path as determined by the spanning tree; therefore, redundant paths are not utilized. MSTP overcomes this limitation through the use of multiple spanning-tree instances (MSTIs). Each MSTI creates a separate topology tree and you can administratively map it to one or more VLANs. Allowing users to administratively map VLANs to MSTIs facilitates better load sharing across redundant links within a Layer 2 switching environment.

Multiple Spanning Tree Region

- An MST region is a group of switches with the same region name, revision level, and VLAN-to-instance mapping
  - Max of 64 MSTIs per region
  - One regional root bridge per instance
- Backward compatible with STP and RSTP through a CST



MSTP allows switches to be logically grouped into manageable clusters, known as multiple spanning tree (MST) regions. An MST region is a group of switches that share the same region name, revision level, and VLAN-to-instance mapping parameters.

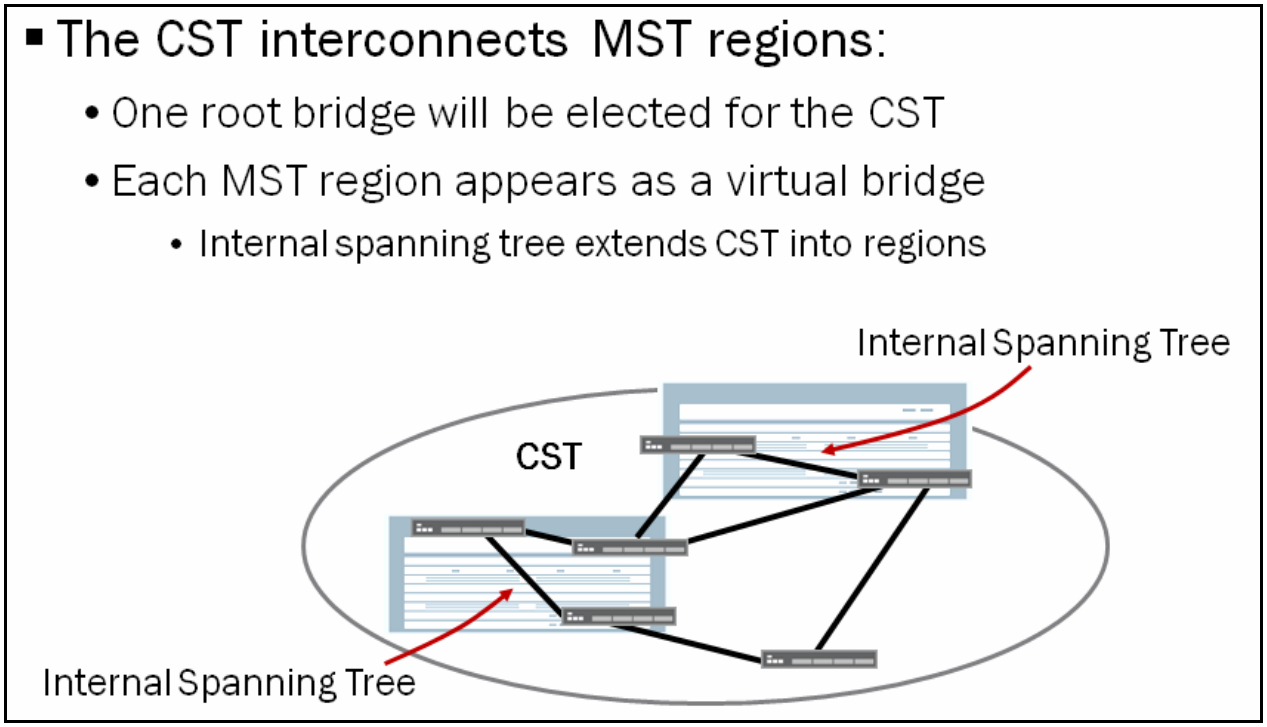
Each MST region supports up to 64 MSTIs. MSTP greatly reduces the number of BPDUs on a LAN by including the spanning tree information for all MSTIs in a single BPDU. MSTP encodes region information after the standard RSTP BPDU along with individual MSTI messages. The MSTI configuration messages convey spanning tree information for each instance.

MSTP elects a regional root bridge for each MSTI. The regional root bridge is elected based on the configured bridge priority and calculates the spanning tree within its designated instance.

MSTP Compatibility with STP and RSTP

Because MSTP encodes region information after the standard RSTP BPDU, a switch running RSTP interprets MSTP BPDUs as RSTP BPDUs. This behavior facilitates full compatibility between devices running MSTP and devices running STP or RSTP. All RSTP switches outside of an MST region view the MST region as a single RSTP switch. The common spanning tree (CST), which interconnects all MST regions as well as STP devices not bound to a particular region, facilitates end-to-end paths within an MSTP environment.

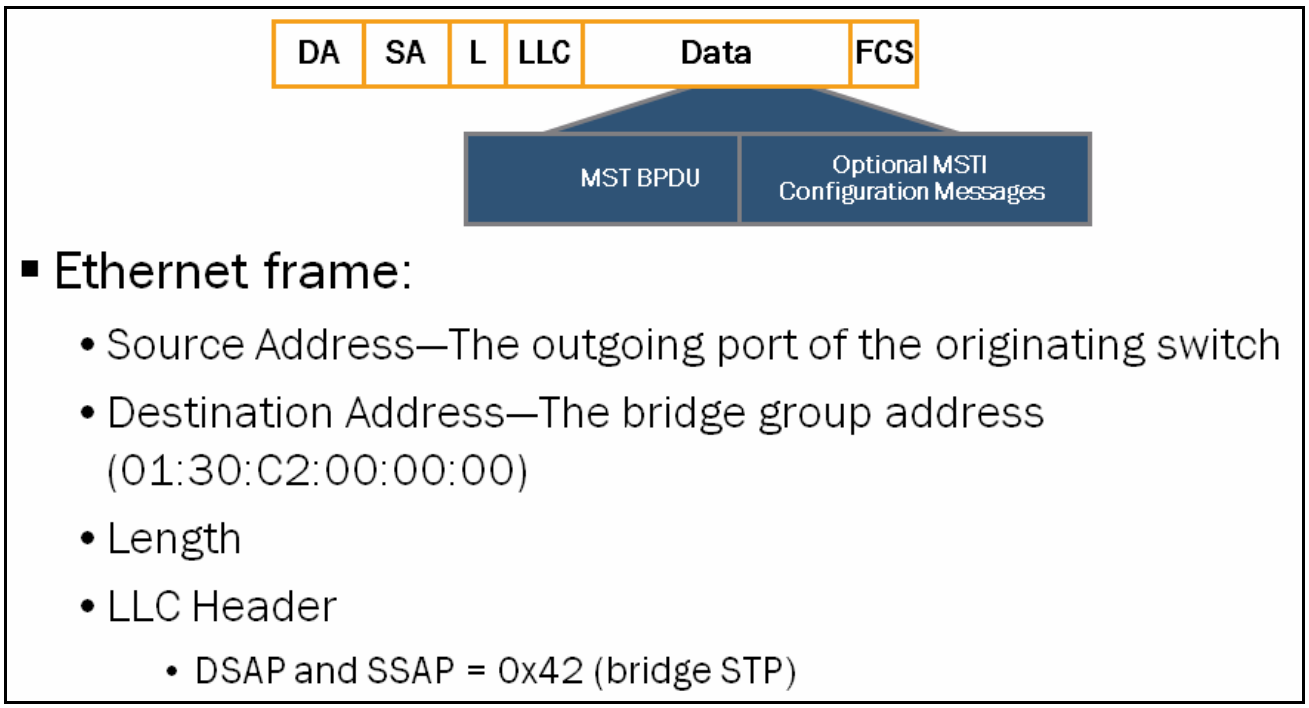
Common and Internal Spanning Tree



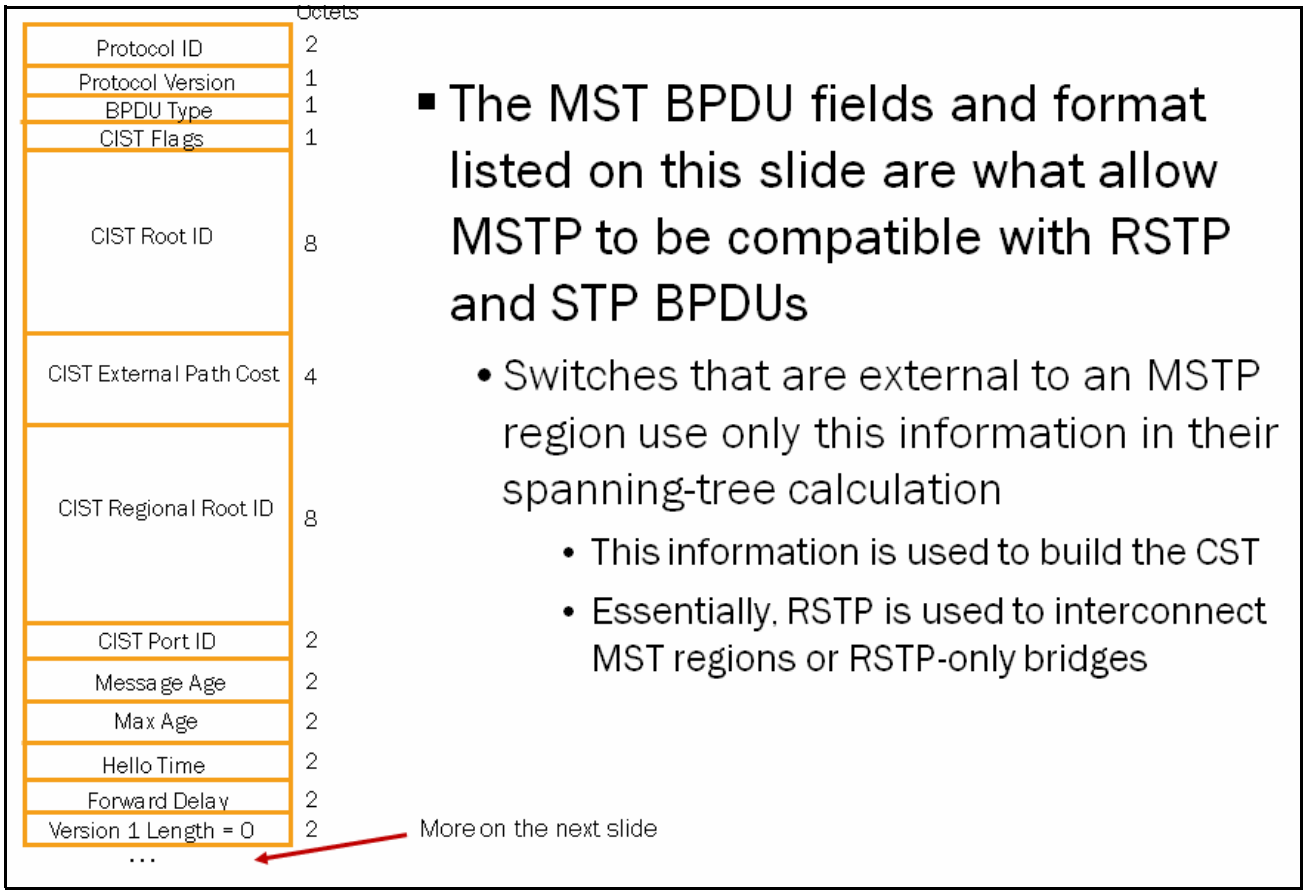
All MSTP environments contain a CST, which is used to interconnect individual MST regions and independent STP devices. All bridges in the CST elect a single root bridge. The root bridge is responsible for the path calculation for the CST. As illustrated on the slide, bridges outside of the MST region treat each MST region as a virtual bridge, regardless of the actual number of devices participating in each MST region.

The common and internal spanning tree (CIST) is a single topology that connects all switches (RSTP and MSTP devices) through an active topology. The CIST includes a single spanning tree as calculated by RSTP together with the logical continuation of connectivity through MST regions. MSTP calculates the CIST and the CIST ensures connectivity between LANs and devices within a bridged network.

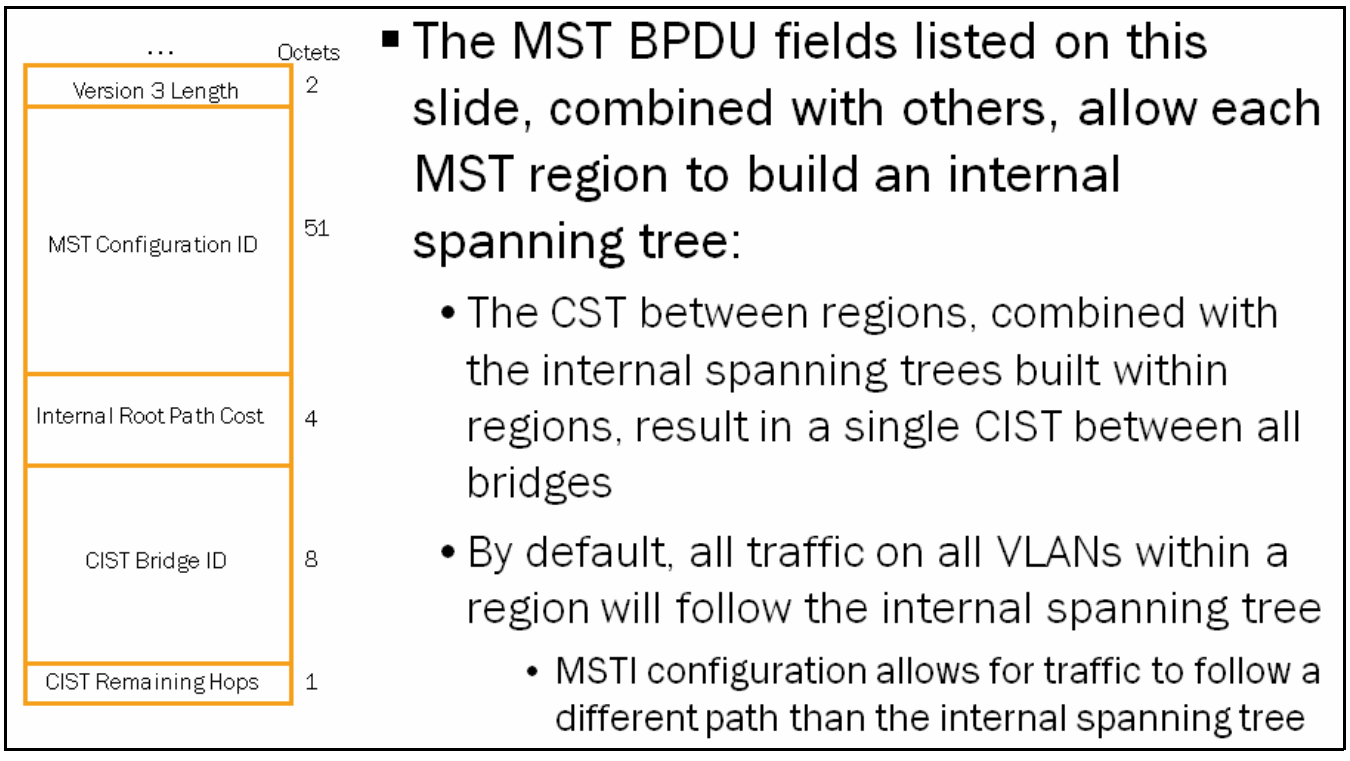
MST BPDU Format



The graphic shows that MSTP uses the same Ethernet frame as STP and RSTP. However, the BPDU information in the data field is different. The next few slides discuss the MST BPDU information and the optional MSTI configuration messages.

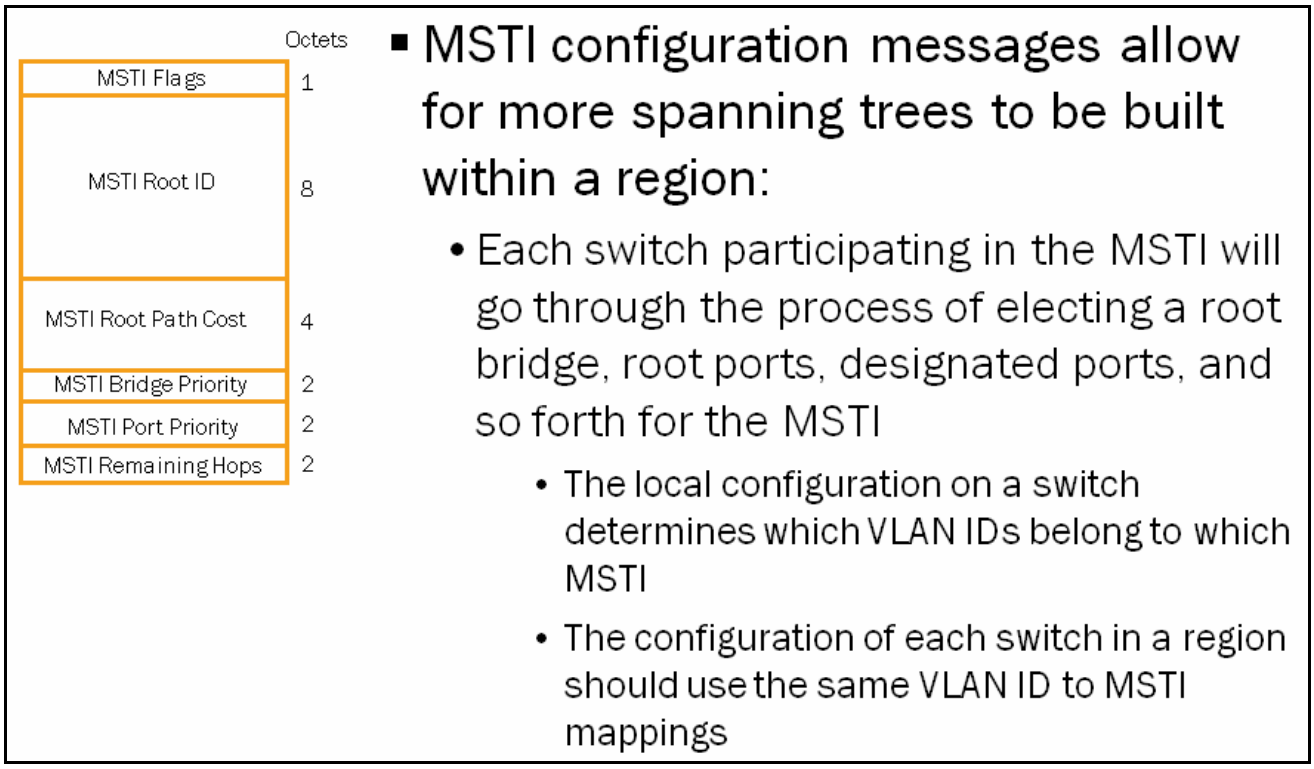


The first 13 fields in the MST BPDU contain similar information to what you would find in an RSTP BPDU. In fact, an RSTP-speaking switch evaluates these fields in the same manner as it would any other RST PDU. To the outside world (other MSTI regions or standalone RSTP speakers), these fields are a representation of the virtual bridge that is an individual MSTP region. This information is used to build the CST.



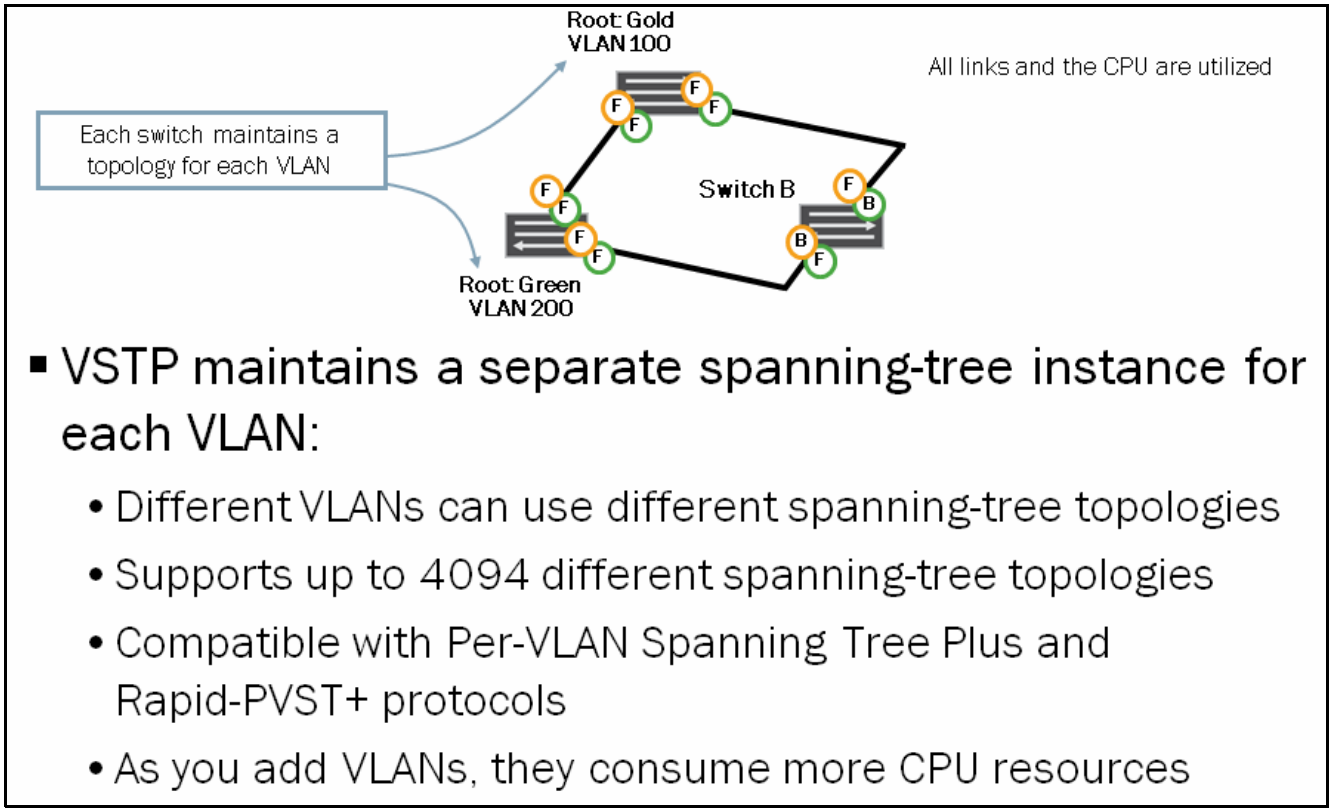
Each MSTP region builds a spanning tree for the region, referred to as an internal spanning tree, based upon the BPDU fields on this slide as well as some of the fields on the previous slide (CIST Port ID, CIST Regional Root ID, and so forth). For a switch to participate in a region's internal spanning tree and use the information in this portion of the BPDU, it must be configured with the same configuration ID. Therefore, all switches in the same region must be configured with the same configuration ID. This approach to configuration ensures that when MSTP switches outside of the local MSTP region receive MSTP BPDUs, those switches will evaluate only the CST-related information (previous slide). Once the internal spanning tree is built, by default, all traffic on all VLANs will follow it.

MSTI Configuration Methods



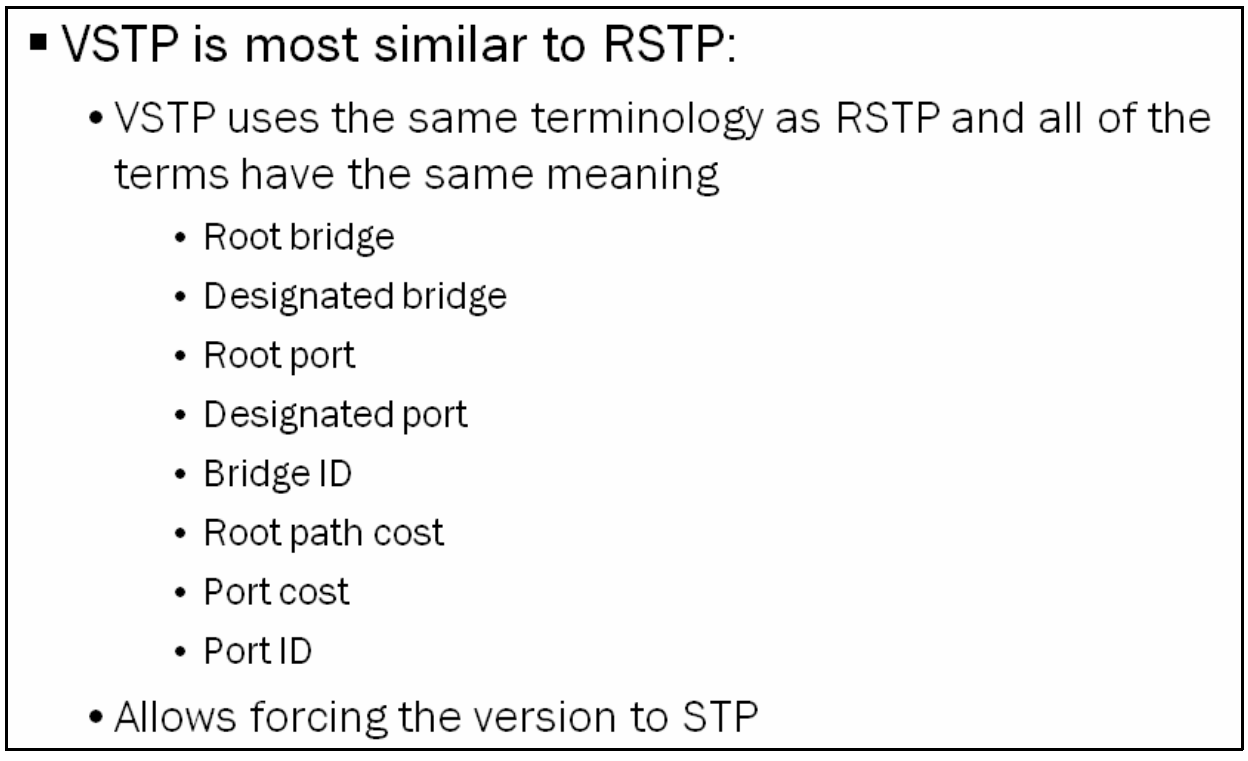
Without the use of MSTI configuration methods, traffic for all VLANs within a region flows along the path of the internal spanning tree. To override this behavior and allow some VLANs to take one path through the region and let others take other paths (64 paths are possible for each region), you must configure MSTIs as part of the router MSTI configuration. The information carried in the MSTI configuration messages allows each switch to elect root bridges, root ports, designated ports, designated bridges, and so forth for each MSTI. Each MSTI will have one or more VLANs associated with them. One VLAN cannot be in more than one MSTI. Notice that the MSTI messages do not carry VLAN ID information. The VLAN-to-MSTI mappings are configured locally on each switch and each switch configuration should use the same mappings.

VSTP



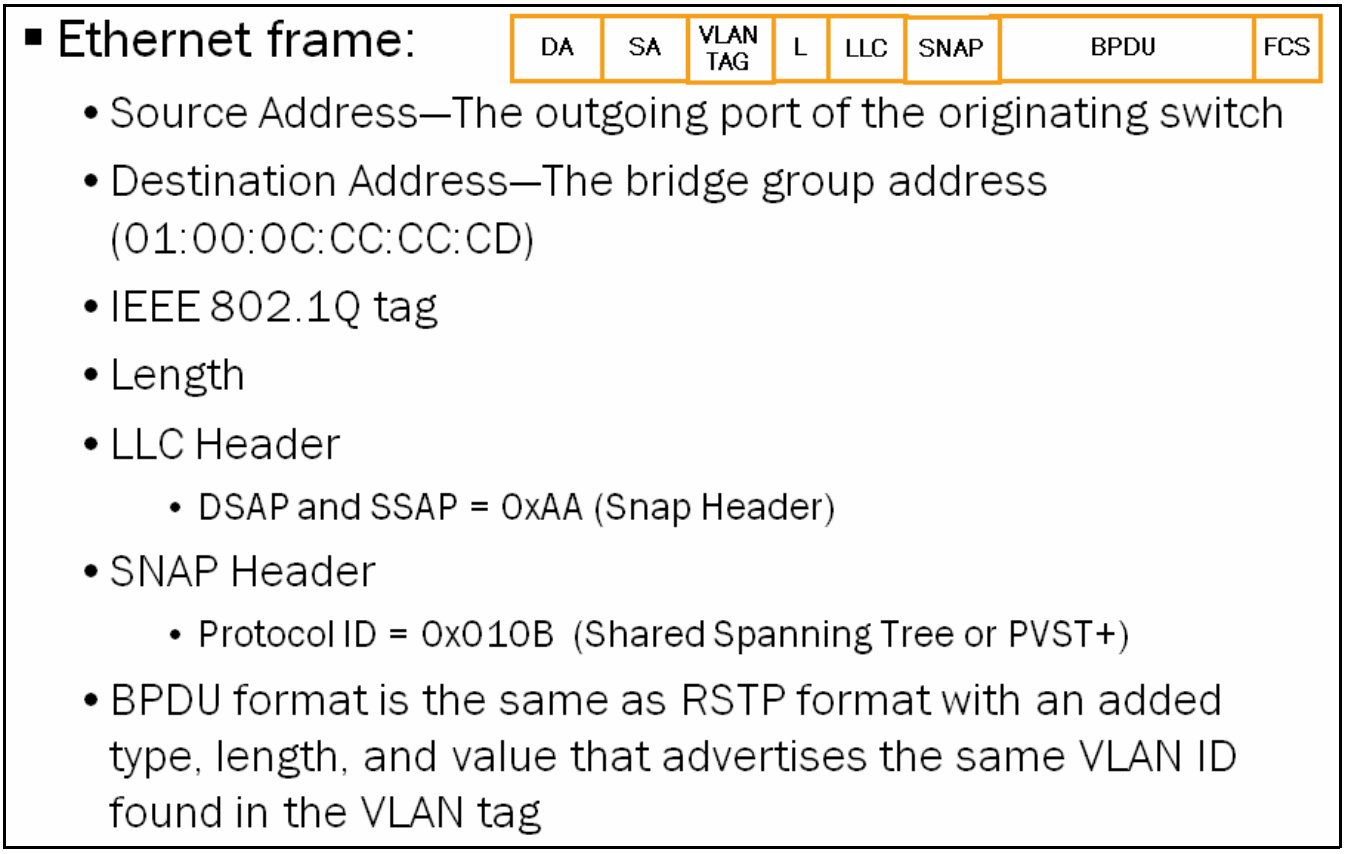
VSTP allows for spanning trees to be calculated for each VLAN. VSTP supports up to 4094 separate paths through the network. VSTP is a nonstandard protocol, yet it is compatible with Cisco's Per-VLAN Spanning Tree Plus (PVST+) and Rapid Per-VLAN Spanning Tree Plus (Rapid-PVST+) protocols. As you add more VLANs to the network, they consume more CPU resources. For example, imagine a network that is configured for 4000 VLANs. If VSTP is in use, each switch must participate in the election of 4000 root bridges, 4000 root ports, and so forth.

VSTP Versus RSTP



VSTP uses the same terminology as the other spanning-tree protocols. It is most similar to RSTP. VSTP also provides for the ability to force the version to STP.

VSTP Frame Format



The graphic shows the format of the VSTP BPDU. Notice that it uses a special destination MAC address and it is also carried in an IEEE 802.1Q tag.

Spanning-Tree Protocols Summary

- STP (802.1D-1998) is used in Layer 2 networks to prevent logical loops
  - Automated—user selects root switch and STP does the rest
  - STP is slow to converge and can be difficult to troubleshoot
- RSTP (802.1D-2004) reduces link-convergence time to subseconds on point-to-point links
- STP and RSTP support a single STP instance
  - Lacks load-balancing mechanism; creates underutilized links
- MSTP (802.1Q-2005) supports up to 64 instances
  - Overcomes the shortcomings of a single spanning tree
- VSTP (proprietary) supports up to 4094 instances
  - Compatible with proprietary protocols from other vendors

The graphic provides a quick overview along with the highlights of STP, RSTP, MSTP, and VSTP.

Configuring STP

```
[edit protocols rstp]
user@switch# set ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  backup-bridge-priority  Priority of the bridge (in increments of 4k - 4k,8k,..60k) (4096..61440)
  bpdu-block-on-edge   Block BPDU on all interfaces configured as edge (BPDU Protect)
  bpdu-destination-mac-address  Destination MAC address in the spanning tree BPDUs
  bridge-priority      Priority of the bridge (in increments of 4k - 0,4k,8k,..60k) (0..61440)
  disable              Disable STP
  extended-system-id   Extended system identifier (0..4095)
  force-version        Force protocol version
  forward-delay        Time spent in listening or learning state (4..30 seconds)
  hello-time           Time interval between configuration BPDUs (1..10 seconds)
> interface            Interface options
  max-age              Maximum age of received protocol bpdu (6..40 seconds)
  priority-hold-time   Hold time before switching to primary priority when core domain becomes up (1..255 seconds)
> system-id            System ID to IP mapping
> traceoptions         Tracing options for debugging protocol operation
  vpls-flush-on-topology-change  Enable VPLS MAC flush on root protected CE interface receiving topology change
```

Configuration example illustrates default STP settings

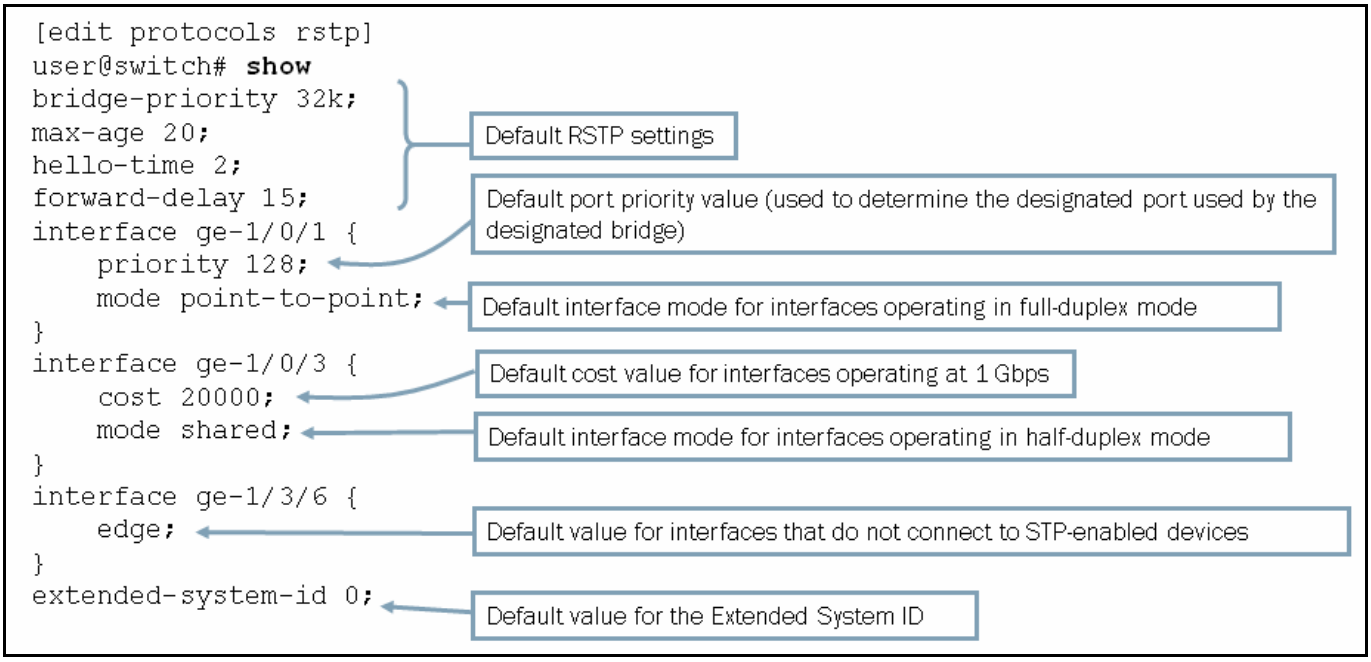
```
[edit protocols rstp]
user@switch# show
bridge-priority 32k;
max-age 20;
hello-time 2;
forward-delay 15;
interface ge-1/0/0;
force-version stp;
```

You must enable STP on at least one interface

The graphic shows some STP configuration options along with a basic STP configuration. MX Series devices use a version of STP based on IEEE 802.1D-2004, with a forced protocol version of 0, running RSTP in STP mode. Because of this implementation,

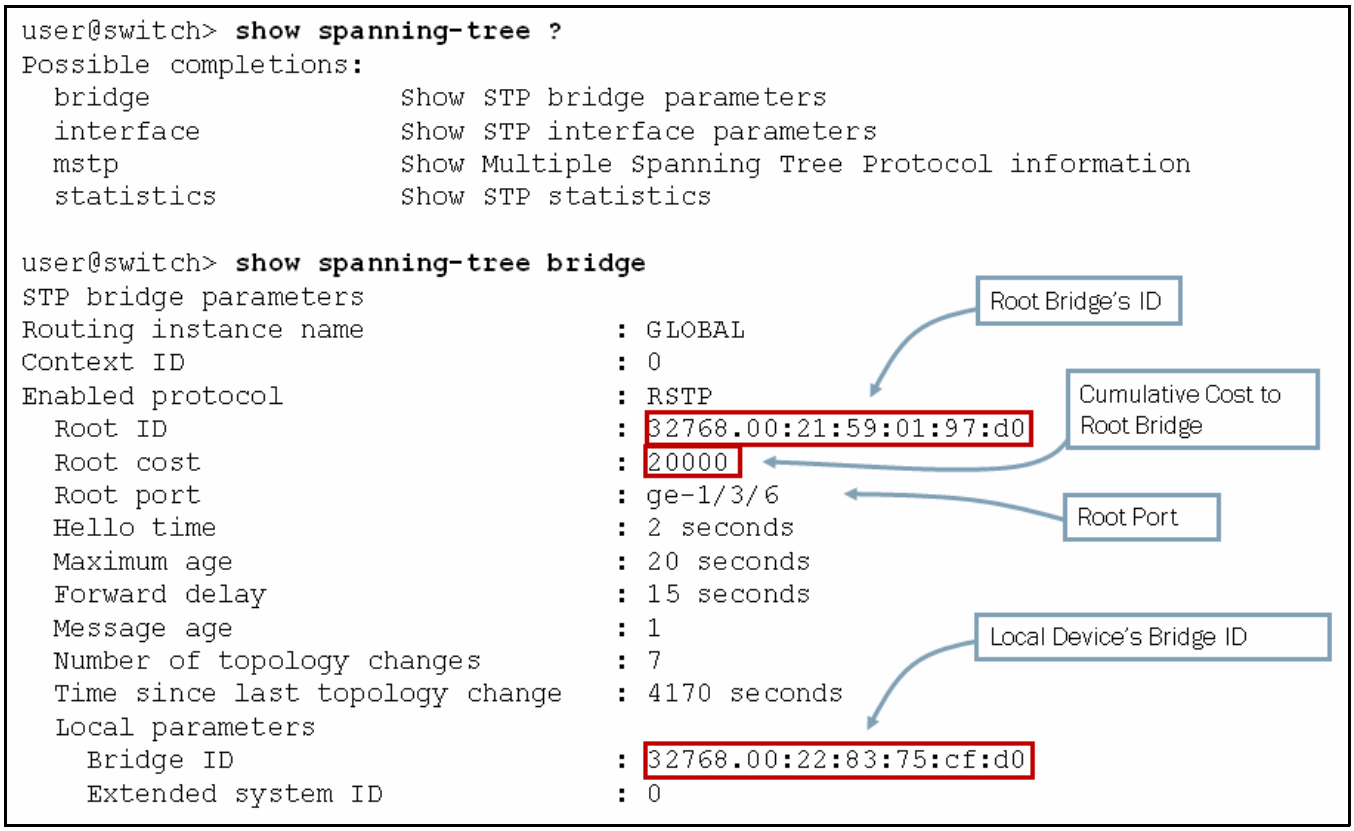
you can define RSTP configuration options, such as `hello-time`, under the `[edit protocols rstp]` configuration hierarchy. To specify running RSTP in STP mode, you simply need to specify **`force-version stp`**.

Configuring RSTP



The sample RSTP configuration provided on the graphic shows the typical configuration structure along with various settings.

Monitoring Spanning Tree Operation

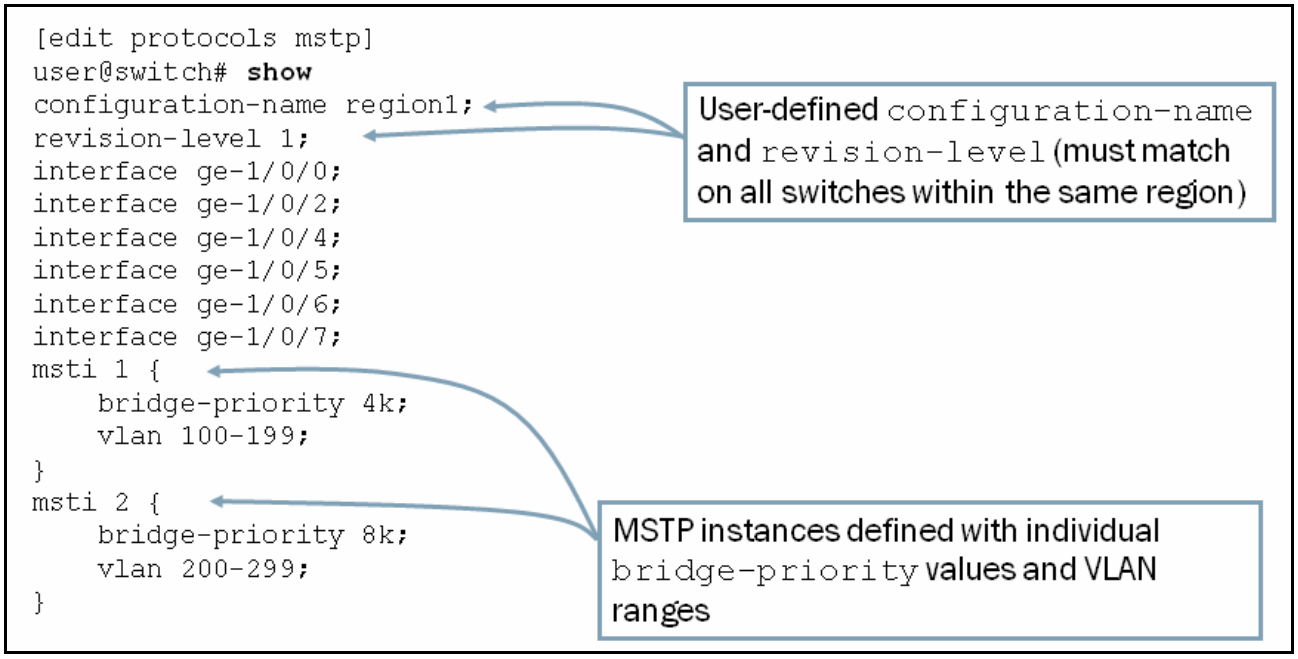


This graphic and the next illustrate some common operational mode commands used to monitor the operation of STP and RSTP.

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ge-1/3/8         128:1        128:1  32768.00228375cfd0     20000  FWD    DESG
ge-1/3/9         128:2        128:2  32768.00228375cfd0     20000  FWD    DESG
ge-1/3/6         128:6       128:77  32768.0021590197d0     20000  FWD    ROOT
ge-1/3/7         128:7       128:78  32768.0021590197d0     20000  BLK    ALT

user@switch> show spanning-tree statistics interface


Interface   BPDUs sent   BPDUs received      Next BPDU
                                            transmission
ge-1/3/8          5915          1418                1
ge-1/3/9          5179          2041                1
ge-1/3/6          4103          4267                0
ge-1/3/7          3979          4393                0
```

The graphic shows typical output for the **`show spanning-tree interface`** and **`show spanning-tree statistics interface`** commands.

Configuring MSTP



The sample MSTP configuration provided on the graphic shows the typical configuration structure along with various settings.

Monitoring MSTP Operation

```
user@switch> show spanning-tree ?
Possible completions:
  bridge               Show STP bridge parameters
  interface            Show STP interface parameters
  mstp                 Show Multiple Spanning Tree Protocol information
  statistics           Show STP statistics


user@switch> show spanning-tree mstp configuration
MSTP configuration information
Context identifier      : 0
Region name             : region1
Revision                : 1
Configuration digest    : 0xb0c8664d7263bcd9aea9f884439230e4


MSTI      Member VLANs
    0       0-99,300-4094
    1       100-199
    2       200-299
```

Values must match for all switches within a common MST region

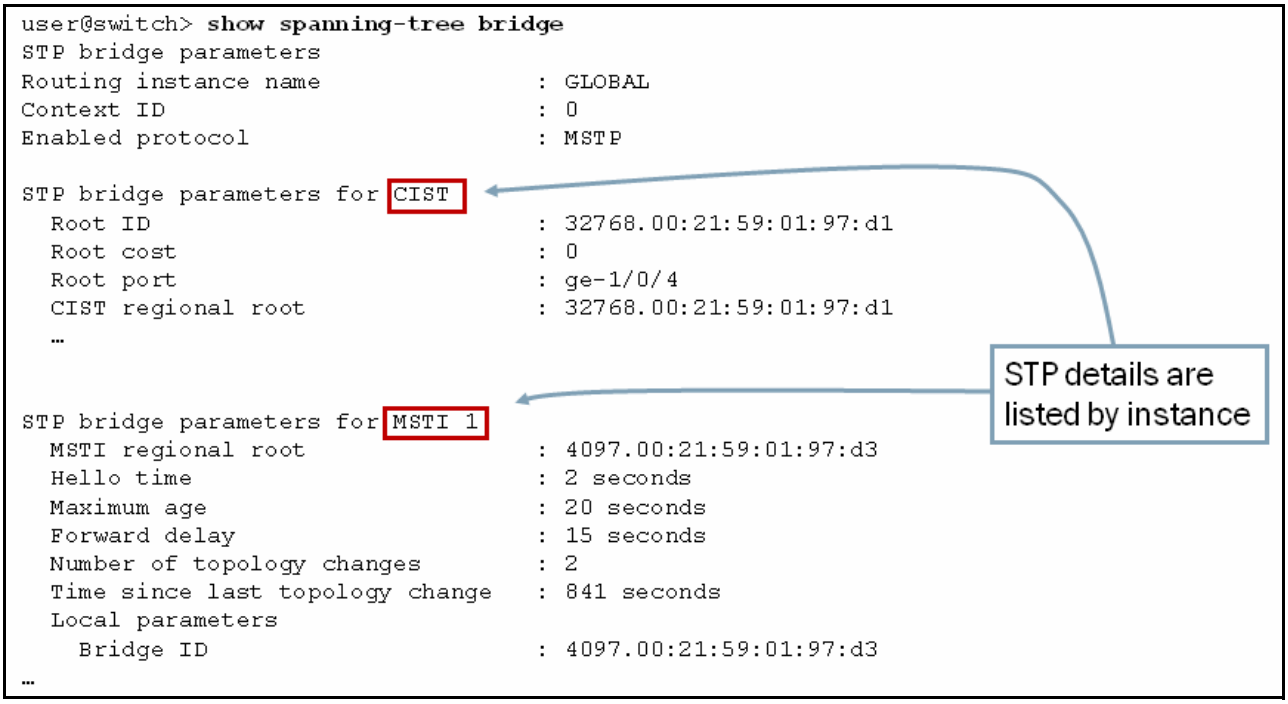
Configuration digest is determined by the contents of MSTI to VID table

This graphic and the next two slides illustrate some common operational mode commands used to monitor MSTP. This graphic highlights the **show spanning-tree mstp configuration** command, which you can use to verify MSTP configuration parameters including region, revision, and assigned MSTI parameters.

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
...
ge-1/0/7        128:48       128:58  32768.0021590197d2     20000  BLK    ALT

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
...
ge-1/0/7        128:48       128:48   4097.0021590197d3     20000  FWD    DESG

Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
...
ge-1/0/7        128:48       128:48   8194.0021590197d3     20000  FWD    DESG
```

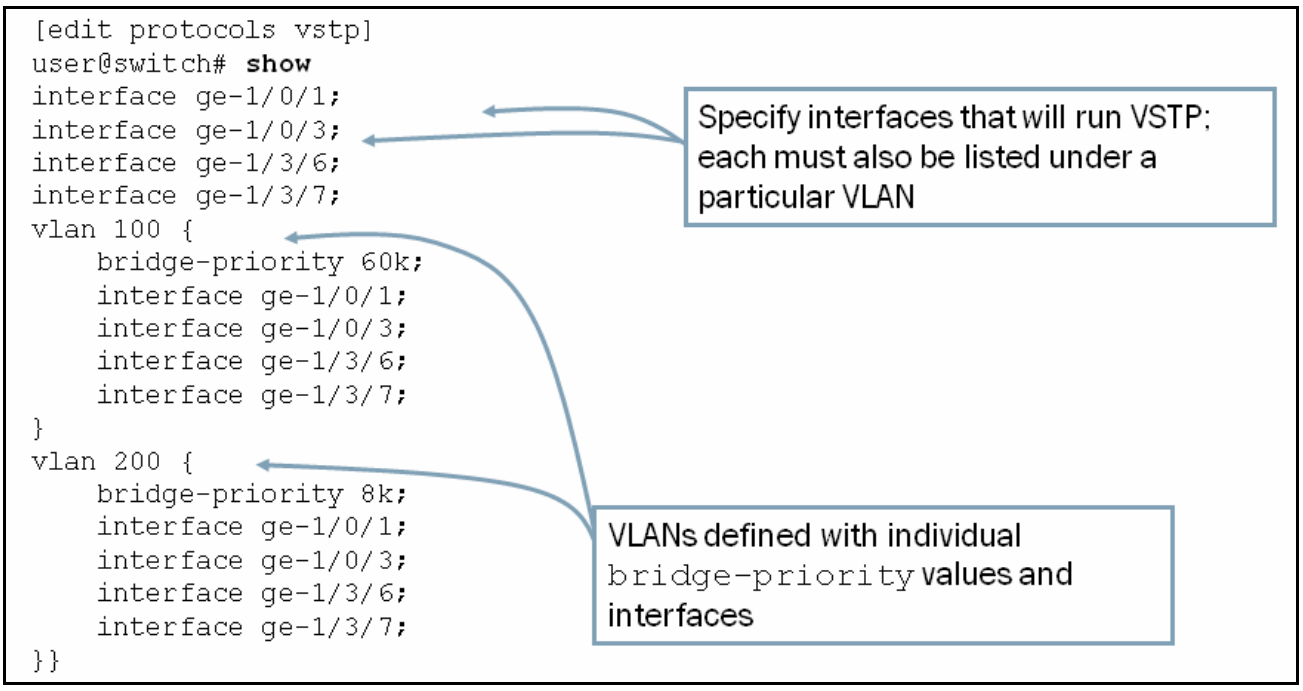
Interfaces and associated details are listed by instance

The graphic highlights the use of the **show spanning-tree interface** command, which you use to verify the MSTP interface status and role assignment along with various other details.



The graphic highlights the **`show spanning-tree bridge`** command, which you use to display STP bridge parameters for the CIST and individual MSTIs.

Configuring VSTP



The sample VSTP configuration provided on the graphic shows the typical configuration structure along with various settings.

Monitoring VSTP:

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for VLAN 100

Interface    Port ID    Designated      Designated          Port    State  Role
                         port ID         bridge ID           Cost
ge-1/0/1        128:42       128:42   8292.002159ab8fd0     20000  FWD    ROOT
ge-1/0/3        128:44       128:44   8292.002159ab8fd0     20000  BLK    ALT
ge-1/3/6        128:77       128:46   8292.002159ab8fd0     20000  BLK    ALT
ge-1/3/7        128:78       128:46   8292.002159ab8fd0     20000  BLK    ALT

Spanning tree interface parameters for VLAN 200

Interface    Port ID    Designated      Designated          Port    State  Role
                         port ID         bridge ID           Cost
ge-1/0/1        128:42       128:42   8392.0021590197d0     20000  BLK    DESG
ge-1/0/3        128:44       128:44   8392.0021590197d0     20000  BLK    DESG
ge-1/3/6        128:77       128:42   8392.0021590197d0     20000  BLK    BKUP
ge-1/3/7        128:78       128:42   8392.0021590197d0     20000  BLK    BKUP
```

Interfaces and associated details are listed by VLAN ID

The graphic highlights the use of the **show spanning-tree interface** command, which you use to verify the VSTP interface status and role assignment along with various other details.

```
user@switch> show spanning-tree bridge
STP bridge parameters
Routing instance name              : GLOBAL
Enabled protocol                   : RSTP

STP bridge parameters for VLAN 100
  Root ID                          : 8292.00:21:59:ab:8f:d0
  Root cost                        : 20000
  Root port                        : ge-1/0/1
  Hello time                       : 2 seconds
  Maximum age                      : 20 seconds
  Forward delay                    : 15 seconds
  Message age                      : 1
  Number of topology changes       : 3
  Time since last topology change  : 141 seconds
  Local parameters
    Bridge ID                      : 61540.00:21:59:01:97:d0
    Extended system ID             : 100

STP bridge parameters for VLAN 200
  Root ID                          : 8392.00:21:59:01:97:d0
  Hello time                       : 2 seconds
  Maximum age                      : 20 seconds
  Forward delay                    : 15 seconds
…
```

STP details are listed by VLAN ID

The graphic highlights the **show spanning-tree bridge** command, which you use to display STP bridge parameters for the individual VLANs.

Purpose for BPDU Protection—Problem

- Bridge applications running on PCs or "personal" switches can generate BPDUs
- STP, RSTP, or MSTP running on a switch could detect those BPDUs and trigger spanning-tree miscalculations leading to network outages

As mentioned previously in this chapter, the purpose for STP, RSTP, and MSTP is to prevent Layer 2 loops in the network. The exchange of BPDUs achieves a loop-free Layer 2 network. A user bridge application running on a PC or a user's "personal" switch for connecting multiple end devices can also generate BPDUs. If these BPDUs are picked up by STP, RSTP, or MSTP applications running on the switch, they can trigger spanning-tree miscalculations, which in turn could cause loops, leading to network outages.

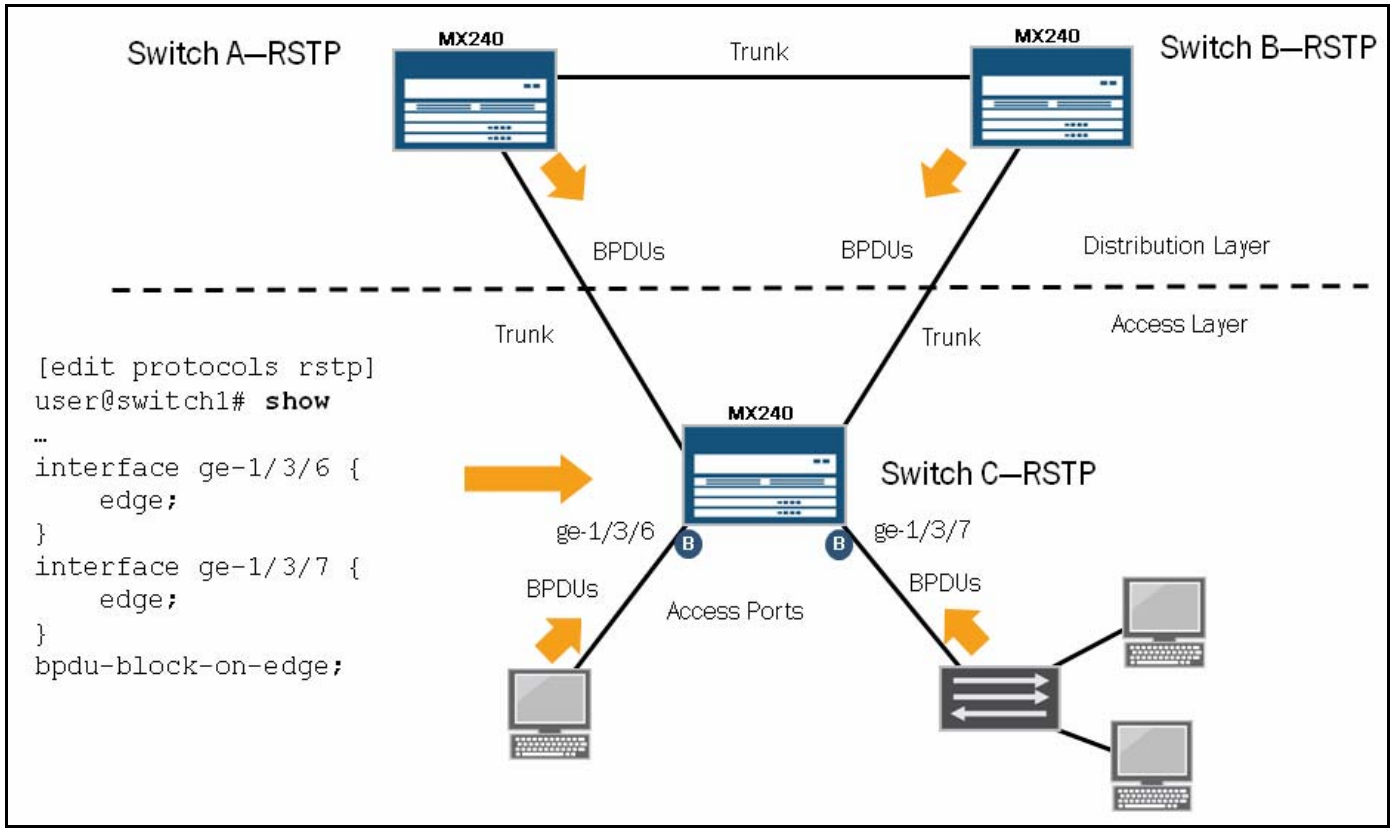
Purpose for BPDU Protection—Solution

- Enable BPDU protection on Layer 2 interfaces connected to user devices or on interfaces on which no BPDUs are expected
- If a protected interface receives a BPDU, the bridge disables the interface and stops forwarding frames by transitioning to a blocking state

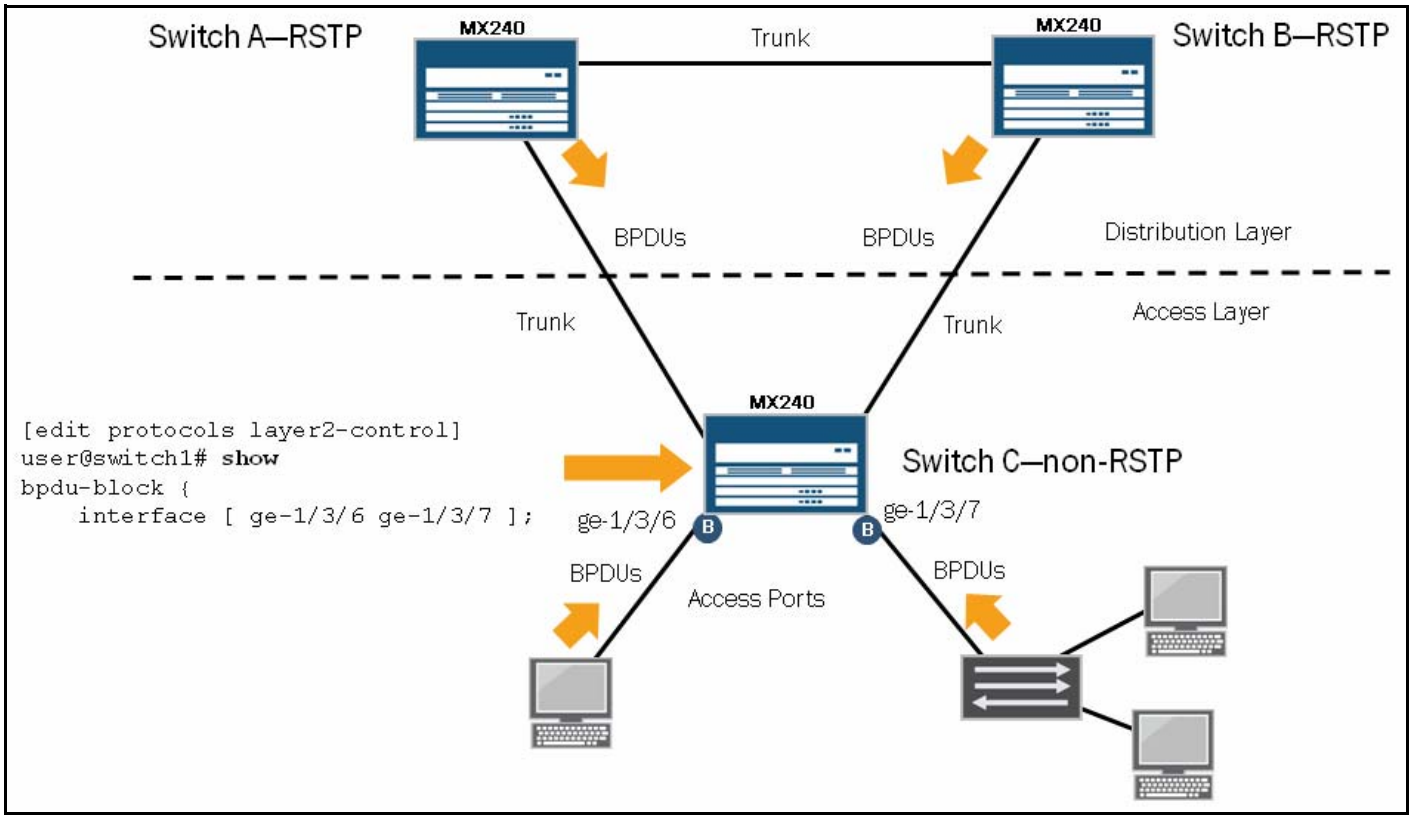
You can enable BPDU protection on switch interfaces on which no BPDUs are expected. If a protected interface receives BPDUs, the switch disables the interface and stops forwarding frames by transitioning to a blocking state.

You can configure BPDU protection on a switch with a spanning tree as well as on a switch that is not running STP.

Configuring BPDU Protection: Part 1



Consider a network in which two switches, Switch A and Switch B, are at the Distribution Layer. A third switch, Switch C, is at the Access Layer. Switch C connects to a PC and a user's unauthorized switch by means of the access ports, ge-1/3/6 and ge-1/3/7, respectively. You have configured Switch A, Switch B, and Switch C to use STP. You should configure BPDU protection on Switch C access ports. The graphic illustrates the required configuration. When BPDU-protected interfaces receive BPDUs, the interfaces transition to a blocking state and stop forwarding frames.



Now consider a network in which two switches, Switch A and Switch B, are at the Distribution Layer. A third switch, Switch C, is at the Access Layer. Switch C connects to a PC and a user's unauthorized switch by means of the access ports—ge-1/3/6 and ge-1/3/7, respectively. Switch C is not configured to participate in STP. However, you should protect its access ports—ge-1/3/6 and ge-1/3/7. The slide illustrates the required configuration. When BPDU-protected interfaces receive BPDUs, the interfaces transition to a blocking state and stop forwarding frames.

Verifying of BPDU Protection Functionality

- Use the **`show spanning-tree interface`** command before and after enabling the BPDU protection feature on an STP-running switch
- Use the **`show l2-learning interface`** command on a non-STP switch
- Watch for state changes, role changes, or both in the output:
  - `FWD` state transitions to `BLK`
  - `DESG` role transitions to `DIS` (loop inconsistent)
  - `unblocked` transitions to `blocked`
- To unblock the interface:
  - Use the **`clear error bpdu interface`** operational mode command

To confirm that the configuration is working properly on the STP-running switch, use the **`show spanning-tree interface`** operational mode command. To confirm that the configuration is working properly on the switch that is not running STP, you should observe the interfaces using the **`show l2-learning interface`** operational mode command.

These commands provide the information on the state and role changes on the protected interfaces. For example, if the PCs send BPDUs and the protected interfaces receive them, the interfaces transition to the `DIS` role. The BPDU inconsistent state changes the interface state to blocking (BLK), preventing it from forwarding traffic.

To unblock the interfaces, you must use the **`clear error bpdu interface`** operational mode command.

The Purpose of Loop Protection—Problem

- Switch hardware and configuration errors could cause an STP loop
- A nondesignated port might stop receiving superior BPDUs from the designated bridge, causing an interface to go into the forwarding state (causing a loop)

Although the purpose of STP, RSTP, and MSTP is to provide Layer 2 loop prevention, switch hardware or software errors could result in an erroneous interface-state transition from the blocking state to the forwarding state. Such behavior could lead to Layer 2 loops and consequent network outages.

The Purpose of Loop Protection—Solution

- Enable loop protection on all root and alternate ports
- Once enabled, selected ports do not interpret the lack of BPDUs as a false positive for making the interface a designated port
  - Ports that detect the loss of BPDUs transition to the "loop inconsistent state" (essentially the same as blocking)
- The interface recovers and transitions back to the blocking state when it receives a BPDU

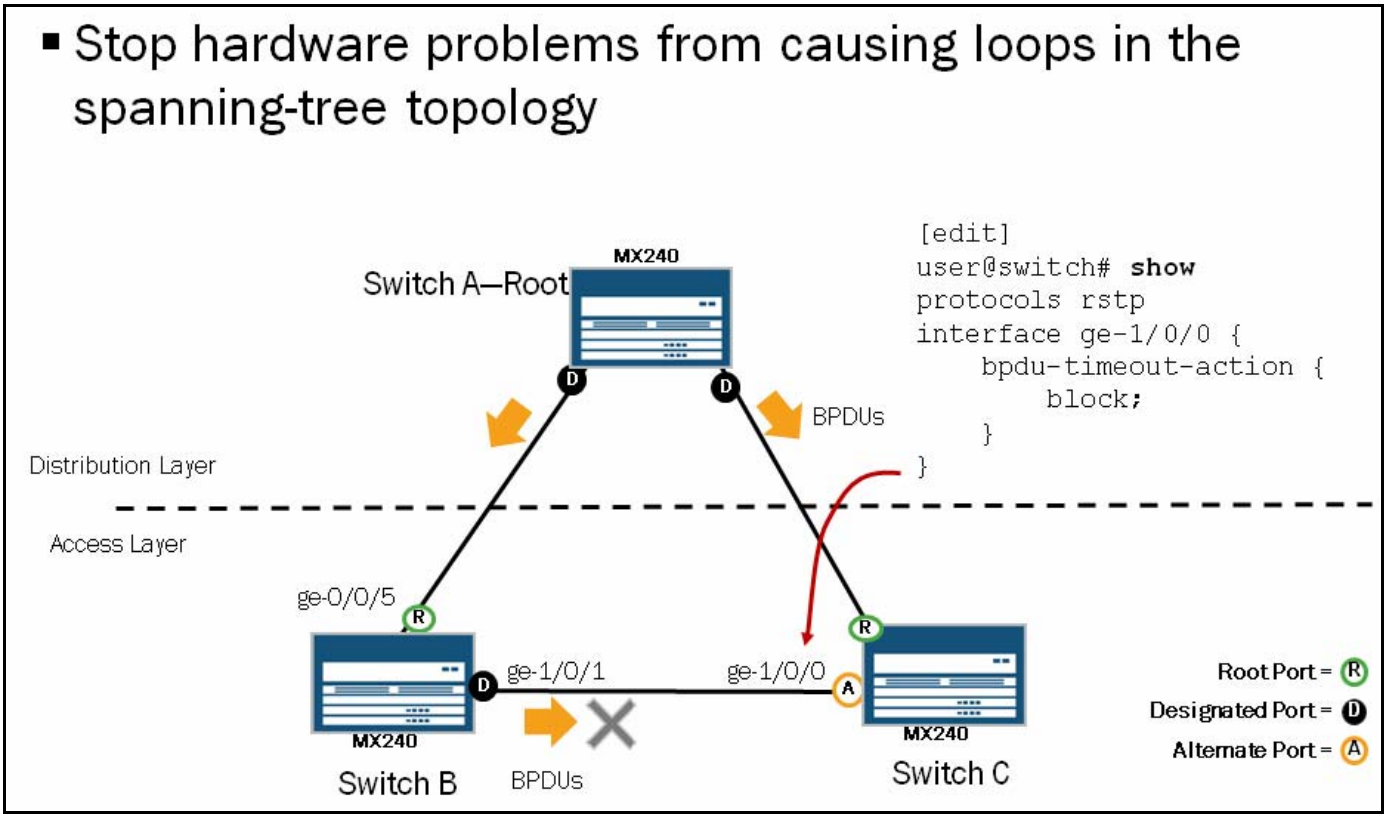
When loop protection is enabled, the spanning-tree topology detects root ports and alternate ports, and ensures that both are receiving BPDUs. If a loop-protection-enabled interface stops receiving BPDUs from its designated port, it reacts as it would react to a problem with the physical connection on the interface. It does not transition the interface to a forwarding state. Instead, it transitions the interface to a loop-inconsistent state. The interface recovers and then it transitions back to the spanning-tree blocking state when it receives a BPDU.

We recommend that if you enable loop protection, you should enable it on all switch interfaces that have a chance of becoming root or designated ports. Loop protection is most effective when it is enabled on all switches within a network.

You can configure an interface for either loop protection or root protection, but not for both.

Configuring Loop Protection

- Stop hardware problems from causing loops in the spanning-tree topology



Now consider a network in which one switch, Switch A, is at the Distribution Layer, and two switches, Switch B and Switch C, are at the Access Layer. Being in an alternate state, interface ge-1/0/0 of Switch C is blocking traffic between Switch B and Switch C; therefore, the traffic forwards through Switch A from Switch C. BPDUs are traveling from the root bridge on Switch A to both its interfaces. Switch C, during normal operation, receives BPDUs from Switch B. In the example on the slide, assume that a hardware problem exists on Switch B's ge-1/0/1 interface such that both Switch B and Switch C believe the interface is up,

but Switch C cannot receive BPDUs from Switch B. Without loop protection, Switch C might place its ge-1/0/0 interface into the forwarding state, causing a loop. The slide illustrates the required configuration, and shows how to configure loop protection on interface ge-1/0/0 to prevent it from transitioning from a blocking state to a forwarding state and causing a loop in the spanning-tree topology.

You can configure an interface for either loop protection or root protection, but not both.

Verification of Loop Protection Functionality

- Use the **`show spanning-tree interface`** command before and after enabling the loop protection feature
- Watch for state changes, role changes, or both in the interface output
  - `BLK` state remains `BLK`
  - `ALT` role transitions to `DIS` (loop inconsistent)
- The interface recovers and transitions back to its original state when it receives BPDUs

To confirm that the configuration is working properly on the STP-running switch, use the **`show spanning-tree interface`** operational mode command prior to configuring loop protection. This command provides information for the interface's spanning-tree state, which should be blocking (BLK).

Once BPDUs stop arriving at the protected interface, the loop protection is triggered on that interface. You can use the **`show spanning-tree interface`** command to observe the state of the interface. This command displays the loop-inconsistent state for the protected interface, which prevents the interface from transitioning to the forwarding state. The interface recovers and transitions back to its original state when it receives BPDUs.

The Purpose of Root Protection—Problem

- Bridge applications running on PCs can generate BPDUs and interfere with root port election
- Erroneous root port election on a switch

Although the purpose of STP, RSTP, and MSTP is to provide Layer 2 loop prevention, a root port elected through the spanning-tree algorithm has the possibility of having been wrongly elected. In addition, user bridge applications running on PCs can generate BPDUs, interfering with root port election.

The Purpose of Root Protection—Solution

- Enable root protection on the switch interfaces that should not receive superior BPDUs from the root bridge and should not be elected as the root port
- The interfaces become designated ports
- Once a superior BPDU arrives on a port with root protection enabled, the port transitions to an inconsistency state, blocking the interface
- The interface recovers and transitions back to the forwarding state when it stops receiving superior BPDUs

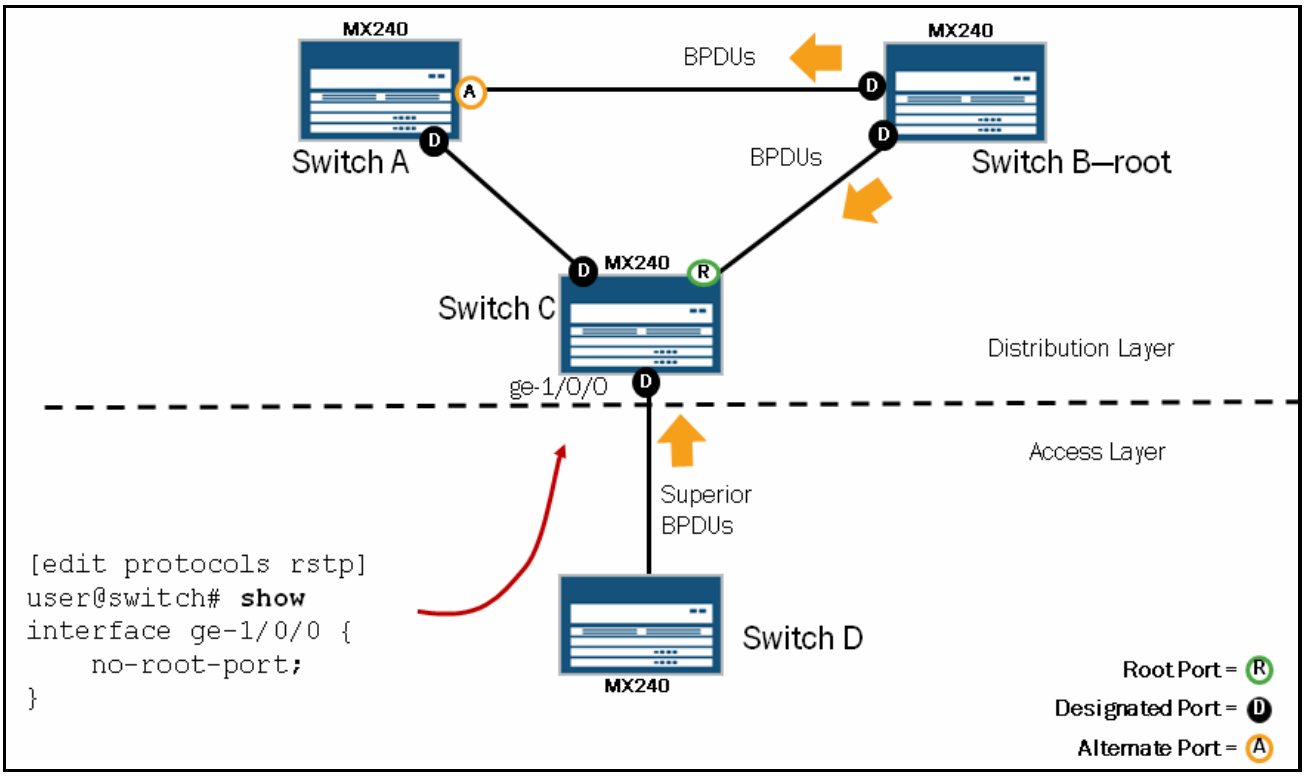
Enable root protection on interfaces that should not receive superior BPDUs and should not be elected as the root port. These interfaces become designated ports. If the bridge receives superior BPDUs on a port that has root protection enabled, that port transitions to an inconsistency state, blocking the interface. This blocking prevents a bridge that should not be the root bridge from being elected.

After the bridge stops receiving superior BPDUs on the interface with root protection, the interface returns to a listening state, followed by a learning state, and ultimately returns to a forwarding state. Recovery back to the forwarding state is automatic.

When root protection is enabled on an interface, it is enabled for all the STP instances on that interface. The interface is blocked only for instances for which it receives superior BPDUs. Otherwise, it participates in the spanning-tree topology.

You can configure an interface for either loop protection or root protection, but not for both.

Configuring Root Protection



Now consider a network in which three switches, Switch A, Switch B, and Switch C, are at the Distribution Layer, and one switch, Switch D, is at the Access Layer.

Interface ge-1/0/0 of Switch C is configured with root protection. If Switch D sends superior BPDUs, they trigger root protection on interface ge-1/0/0, blocking it. The slide illustrates the required configuration.

You can configure an interface for either loop protection or root protection, but not both.

Verification of Root Protection Functionality

- Use the **`show spanning-tree interface`** command before and after you enable the root protection feature
- Receipt of superior BPDUs on the watched interface triggers root protection
  - `FWD` state changes to `BLK`
  - `DESG` role transitions to `DIS` (loop inconsistent)
- The interface recovers and transitions back to its original state when it no longer receives superior BPDUs

To confirm that the configuration is working properly on the STP-running switch, use the **`show spanning-tree interface`** operational mode command prior to configuring loop protection. This command provides information for the interface's spanning-tree state.

Once you configure root protection on an interface and that interface starts receiving superior BPDUs, root protection is triggered. You can use the **`show spanning-tree interface`** command to observe the state of the impacted interface. This command displays the loop-inconsistent state for the protected interface, which prevents the interface from becoming a candidate for the root port. When the root bridge no longer receives superior BPDUs from the interface, the interface recovers and transitions back to a forwarding state. Recovery is automatic.

Review Questions

1. What is the purpose of STP?
2. Describe the operation of the STP port states.
3. Describe how to build a spanning tree.
4. How are STP, RSTP, MSTP, and VSTP different?

Answers

1.

STP is a simple Layer 2 protocol that prevents loops and calculates the best path through a switched network that contains redundant paths. STP automatically rebuilds the tree when a topology change occurs.

2.

Blocking drops all data packets and receives BPDUs. *Listening* drops all data packets and listens to BPDUs. *Learning* does not forward data traffic but builds the MAC address table. *Forwarding* forwards all data traffic and transmits and receives BPDUs. *Disabled* does not participate in STP (administratively disabled).

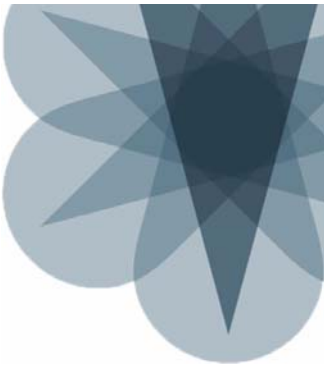
3.

The basic steps involved in building a spanning tree are that switches exchange BPDUs, each individual bridge elects a single root bridge based on the received BPDUs, and the bridges determine the role and state of individual ports, at which time the tree is considered fully converged.

4.

RSTP improves link-convergence time significantly over STP. MSTP supports up to 64 regions and allows load balancing over redundant links—STP and RSTP do not. VSTP allows for per-VLAN spanning trees.

JUNIPER NETWORKS



JNCIS-SP Study Guide—Part 2

Chapter 6: Ethernet OAM

This Chapter Discusses:

- Typical Operation, Administration, and Maintenance (OAM) features;
- The basic operation of link fault management (LFM);
- The basic operation of connectivity fault management (CFM); and
- Configuration and monitoring of Ethernet OAM.

OAM Monitors the Health of a Network

- OAM is a set of functions that allows network operators to monitor the health of the network:
  - Determines the location of faulty links or faulty conditions
  - Measures performance of the network
  - Allows for diagnosis testing (loopback and so forth)

For years, SONET and Asynchronous Transfer Mode (ATM) devices have been able to perform the functions of OAM. Only recently have standards been introduced to bring these same types of features to Ethernet. The purpose of OAM is to help network operators determine the location of faulty conditions, measure the performance of the network, and allow for diagnosis testing. One key to Ethernet's success in becoming *carrier-class* is support for OAM features.

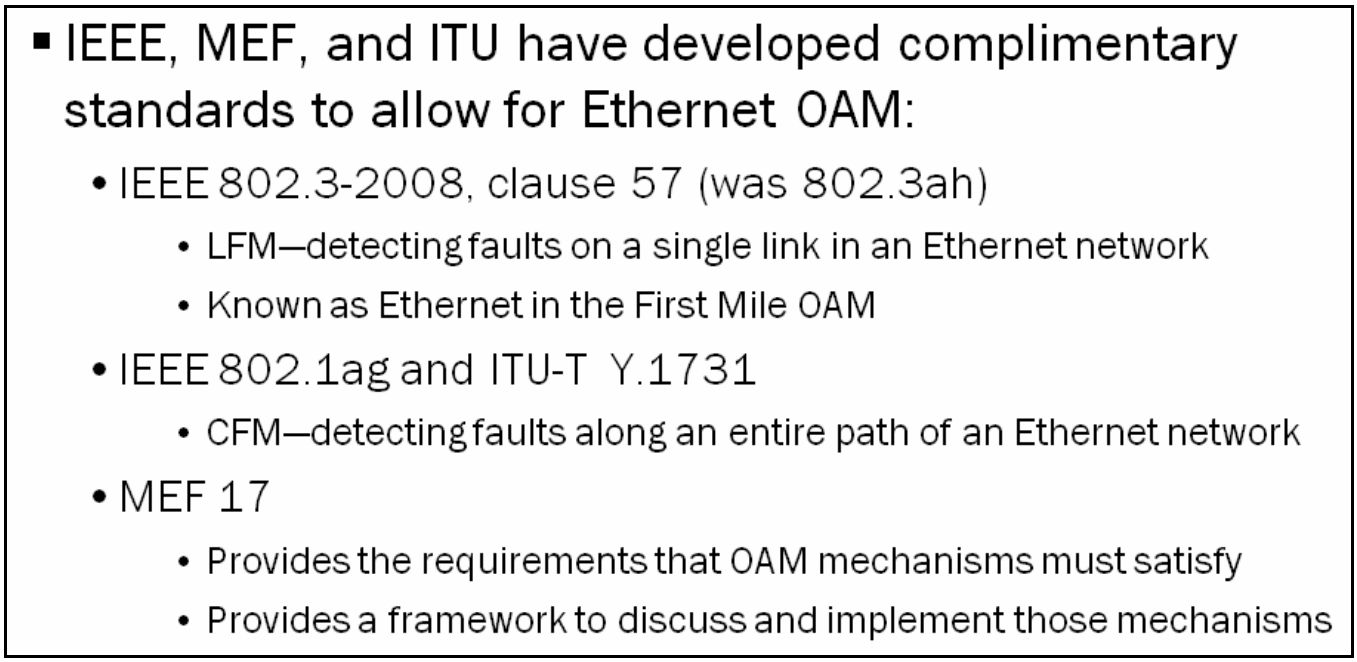
OAM Measurements

- Measurements that can be taken:
  - Availability
  - Frame delay
  - Frame delay variation
  - Frame loss

A device supporting OAM should determine the following measurements:

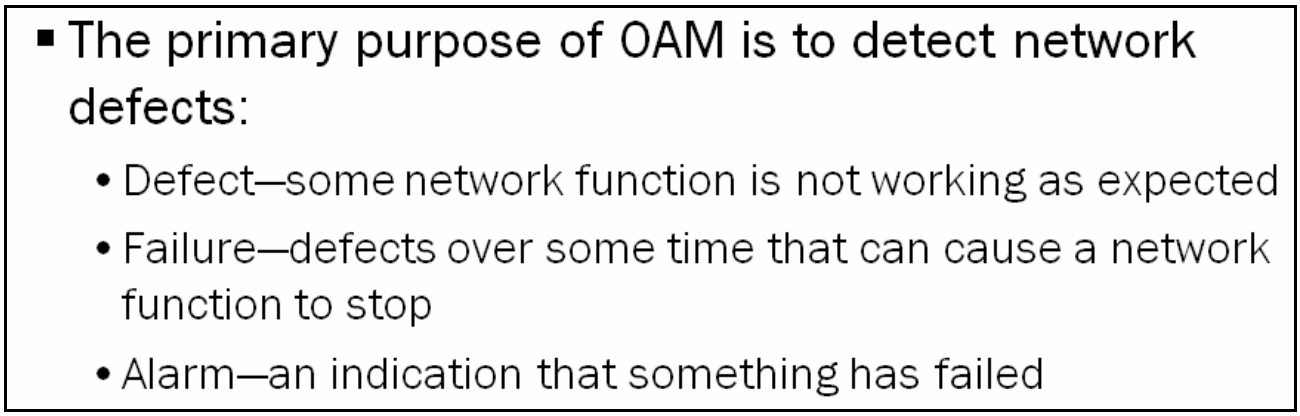
- *Availability*: The ratio of uptime over total time the measure takes;
- *Frame delay*: The time required to transmit a frame from one device to another;
- *Frame delay variation*: The variation in frame delay measurements between consecutive test frames; and
- *Frame loss*: The number of frames lost over time.

Ethernet OAM Standards



The Institute of Electrical and Electronics Engineers (IEEE), Metro Ethernet Forum (MEF), and International Telecommunication Union Telecommunication Standardization (ITU-T) organizations have developed complementary standards to allow for Ethernet OAM. IEEE 802.3-2008, clause 57, defines a method of OAM to monitor link performance, detect faults, and perform loopback testing over a single link. It is usually used on the customer's access link to the provider. The standard is referred to as Ethernet in the First Mile OAM (EFM OAM). The IEEE 802.1ag standard specifies the requirements to detect faults along an end-to-end path of an Ethernet network using CFM. CFM provides for fault monitoring, path discovery, fault isolation, and frame delay measurement (ITU-T Y.1731). The MEF 17 technical specification defines the requirements that must be satisfied by both an equipment vendor and service provider in the area of fault management, performance monitoring, autodiscovery, and intraprovider and interprovider service OAM.

OAM Basics



In general, OAM has the main purpose of detecting network defects. A defect is a network function that is not working as expected. If a defect continues to occur over time, a device considers the recurring defect a failure. The device signals the

failure as an alarm. An alarm is a notification that alerts a human and potentially other devices that something has gone wrong in the network.

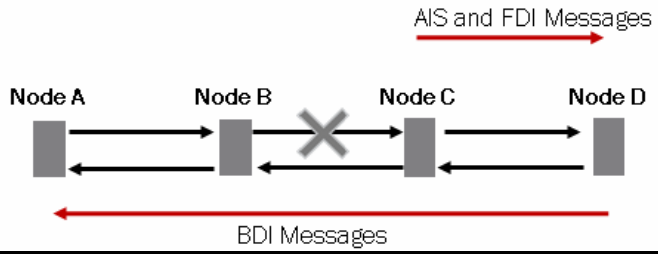
Continuity Check Messages

- Unidirectional messages
  - Sent at regular intervals by one endpoint
  - If the remote end does not receive the message within a certain interval, a fault is detected, potentially causing an alarm

A common feature of OAM is the usage of continuity check (CC) messages. These messages are unidirectional (no acknowledgments), and they are sent between devices. These messages notify a remote device that the local device is still reachable along the path of the CC message. If for some reason a failure occurs that prevents the CC messages from being delivered, the remote device might consider that network path down and generate an alarm.

Indications

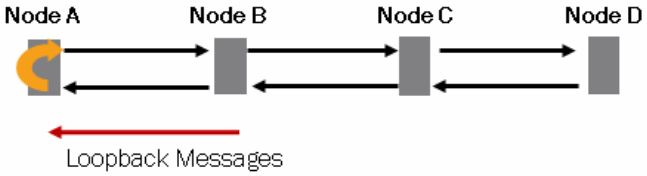
- Alarm indication signal and Forward Defect Indicator
  - Notify downstream network nodes when a failure or defect occurs
- Backward Defect Indicator
  - Notifies upstream network nodes when a failure occurs in the reverse direction



Another feature of OAM protocols is the use of indicators to signal failures in the network. The diagram on the slide shows several devices that are interconnected by some network media (SONET for example). The diagram also shows that the network media has a transmit path and a receive path. The example shows that a failure has occurred on Node B's transmit path and Node C's receive path on the link between Node B and Node C. OAM capabilities allow the devices in the data path to *indicate* that a failure has occurred. Node C uses an alarm indication signal (AIS) and Forward Defect Indicators (FDIs) to inform downstream devices (Node D) that a failure has occurred along the upstream path. Once Node D receives the AIS and FDI messages, it might send a Backward Defect Indicator (BDI) along its transmit path to inform nodes in the reverse direction of the failure (Node A and Node B) that a problem exists downstream from those devices.

Loopback Messages

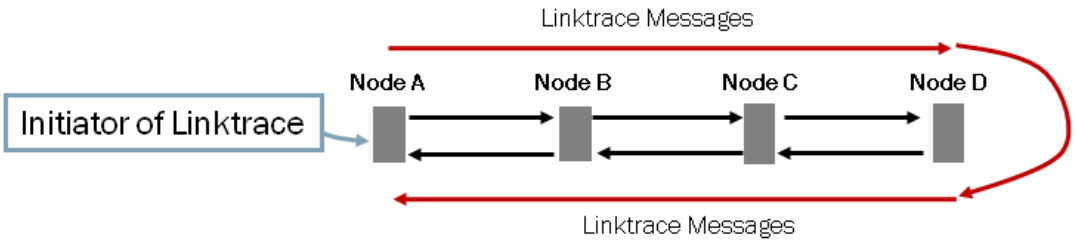
- Help narrow the scope of the issue if a problem exists on a network with multiple nodes
- Allow for detection of a defect between nodes
- Comprise two different types:
  - Nonintrusive loopback messages—do not cause disruption to service (like the ping facility)
  - Intrusive loopback messages—signal a remote node to go into a special test mode (normal transit traffic cannot flow)



Loopback messages are a feature of OAM that help an administrator to find faults in the network. Two types of loopback messages exist and both types are initiated by an administrator of a device. Non-intrusive loopback messages (like ping for IP) allow an administrator to direct a device to send a loopback message downstream to another device with the expectation that the device will respond with a loopback response message. If the response is not received, the administrator might need to perform further testing. Another type of loopback message is an intrusive type. This type of message also is initiated by the administrator of a device, but it signals a downstream device to place a loop on its interface. Usually, when the remote device's interface is in a loop, it cannot pass normal traffic. Instead of receiving any traffic arriving on the receive path of the remote device's looped interface, the remote device loops the traffic around and sends it back out its transmit path. By setting a loop on a remote interface, an administrator can send traffic toward the remote device, and if it returns to the initiating device, then the administrator can eliminate the data path from this list of potentially bad data paths.

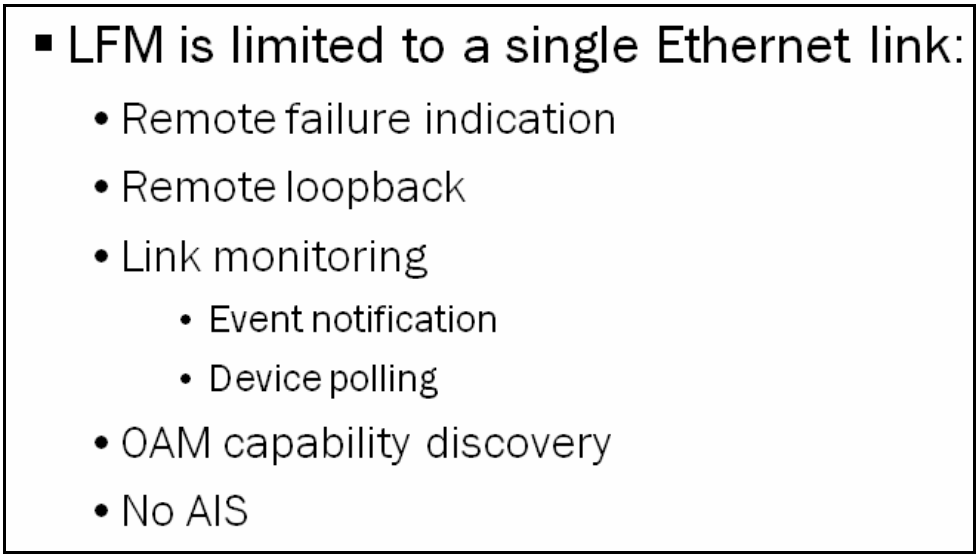
Linktrace Messages

- Linktrace messages:
  - Bidirectional continuity check
    - Similar to traceroute for IP
  - Identifies nodes along the path of the messages



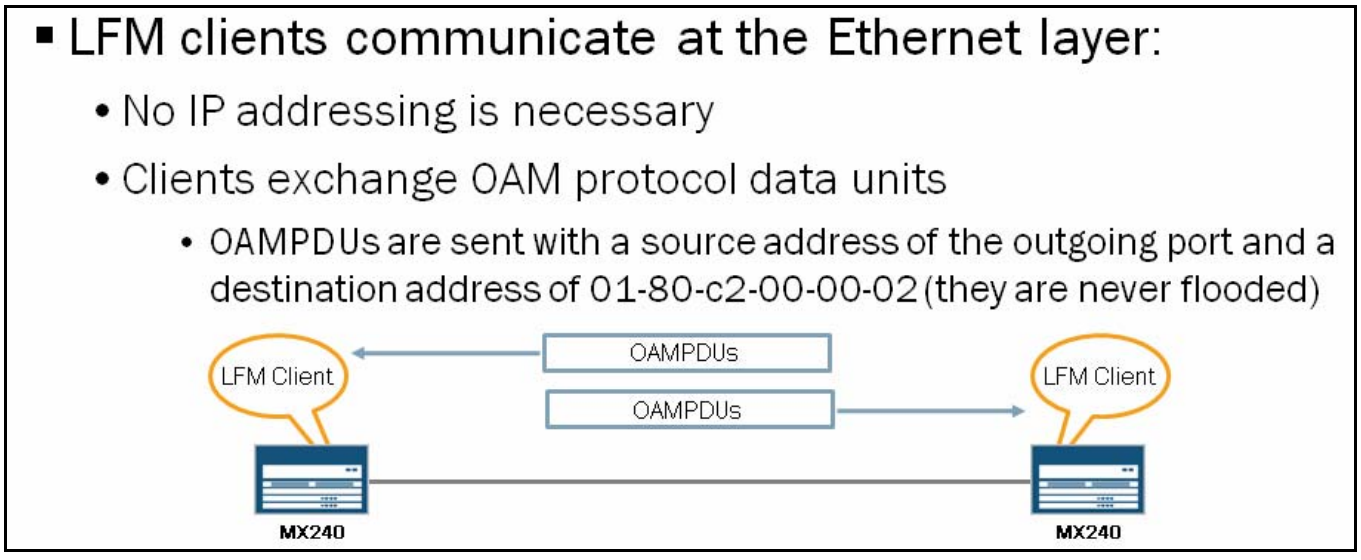
Linktrace messages are a feature of OAM that is similar to the nonintrusive loopback messages that we discussed previously. The difference is that each device along the path of the linktrace message will identify itself by sending a response to the initiator of the linktrace message. After gathering the responses from all of the devices along the path of the message, the administrator knows the identity of all of the devices along the data path.

LFM Capabilities



LFM is defined in IEEE 802.3, Clause 57. It specifies a method of OAM to be used on a single link. Usually, LFM is used on user-to-network interface (UNI) links between a customer and the service provider. On the single Ethernet link, LFM can provide for remote failure indication, remote loopback (intrusive), link monitoring, event notification, and OAM capability discovery. Because LFM is used on a single link, no AIS capabilities are available.

LFM Clients



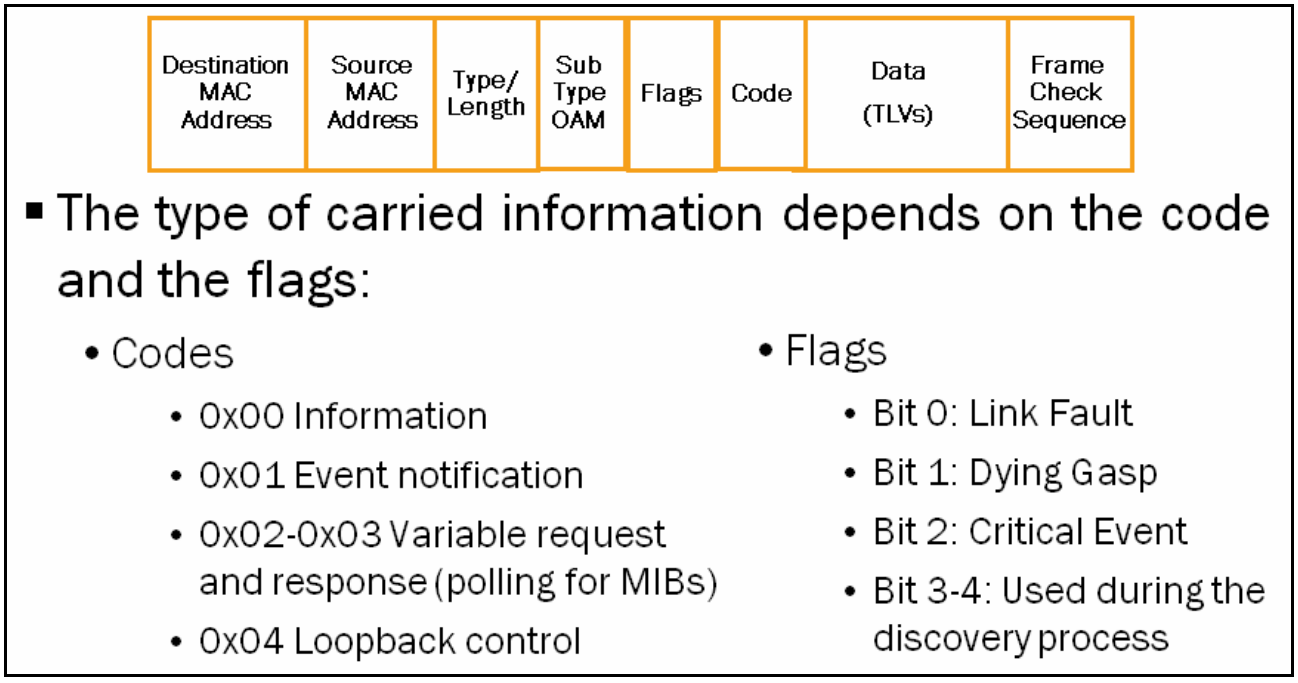
A switch must have LFM client capabilities to support LFM. LFM clients exchange OAM protocol data units (OAMPDUs) that are addressed to 01-80-c2-00-00-02—a multicast MAC address. These messages are sent only across a single link and never flooded. To determine whether an LFM client is on the opposite side of an Ethernet link, an LFM client goes through a discovery process. The discovery process is where LFM clients discover their peers and determine each other's supported capabilities.

Active and Passive

- LFM clients can be either active or passive:
  - Passive clients cannot initiate the discovery process or loopback control messages
  - At least one client must be in active mode (both can be)

A client can be either active or passive (both clients can also be active). Only active LFM clients can initiate the discovery process. Also, only active clients can send loopback messages.

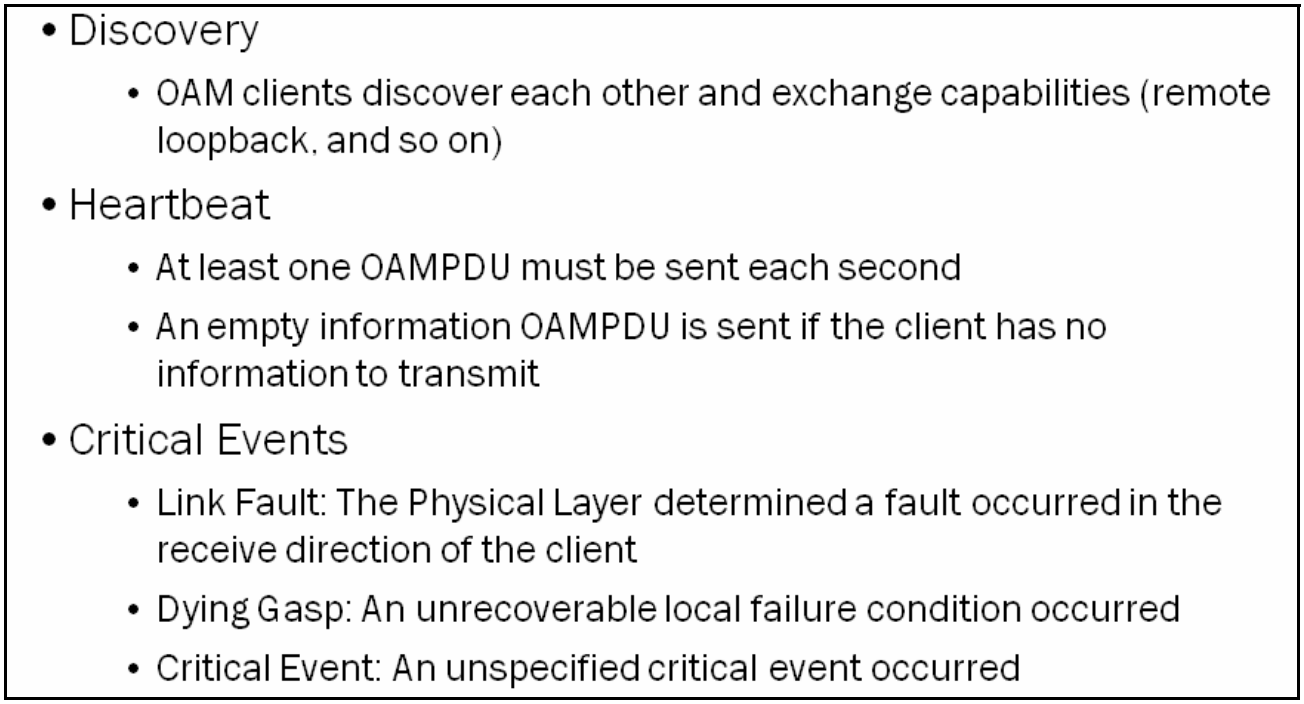
OAMPDUs



Several different types of OAMPDUs exist. We discuss each over the next few slides. This slide shows the format of an OAMPDU. The type of OAMPDU is determined by the code and the flag settings. In general, four types of OAMPDUs exist: information, event notification, variable request and response, and loopback control. Except in the case of the discovery process, the flags are used as BDIs, as described in the previous slides, which notify the remote LFM client of a failure. The following events result in the setting of flags:

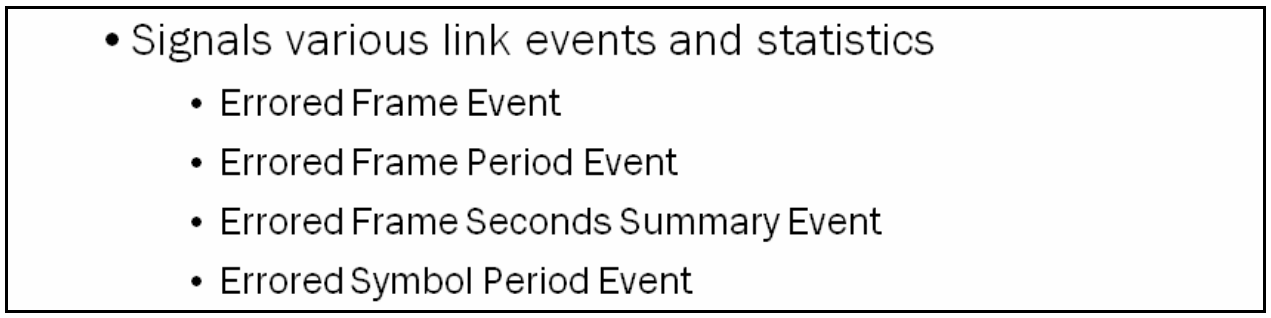
- *Link Fault*: Signal loss is detected on the receive path.
- *Dying Gasp*: An external failure condition occurred. A power failure is a good example of a Dying Gasp event.
- *Critical Event*: An unspecified failure event. By configuring an `action-profile` (described on the following slides), an administrator can specify which events can cause the local switch to send OAMPDUs with the critical event bit set.

Information OAMPDUs



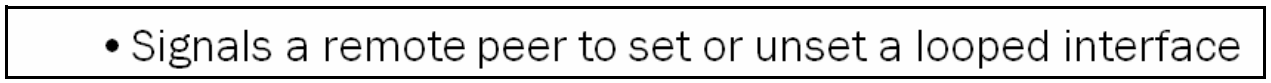
Information OAMPDUs serve several purposes. They are used during the discovery process to discover neighboring clients and exchange capabilities. They also are used to perform continuity checks between clients. When the clients have no information to transmit, they exchange empty information OAMPDUs on a regular, configured interval. In this case, they provide a heartbeat between the two clients. If the peer stops receiving so many messages, it determines that a failure occurred on the link between the two peers. Also, the information OAMPDU signals critical events, as described previously.

Event Notification OAMPDUs



Event notification OAMPDUs are used as BDIs. That is, they inform the upstream client that errors have occurred on the local receive path. The slide lists the four different types of event notifications.

Loopback Control OAMPDU

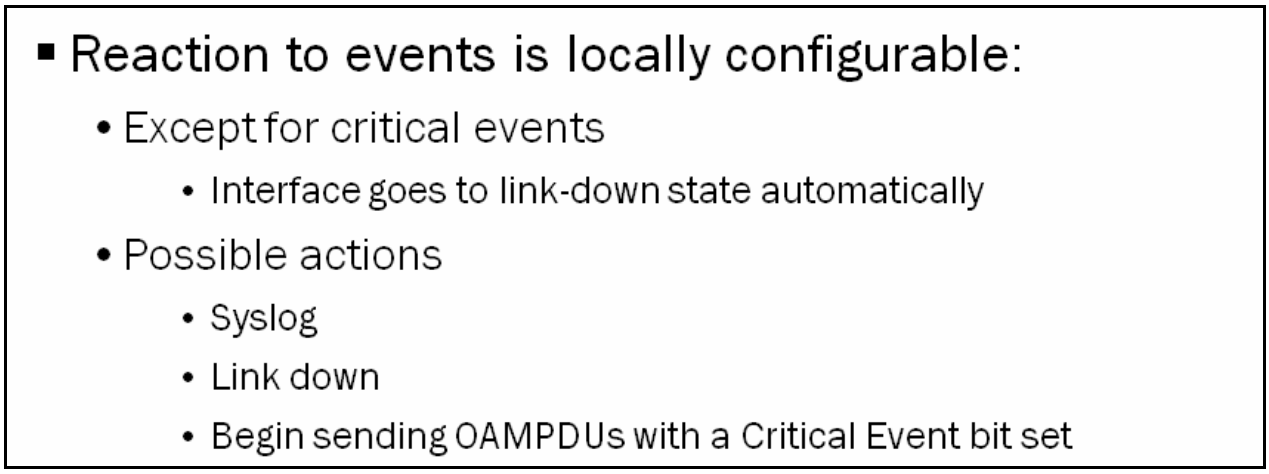


The loopback control OAMPDU allows an LFM client to direct the remote clients to set or unset a loop on its interface.

Variable Request and Response OAMPDUs

Variable request and response OAMPDUs are not currently supported in the Junos operating system. They are used to allow for one client to gather information about the remote clients by polling it for IEEE 802.3, clause 30 MIBs (read only).

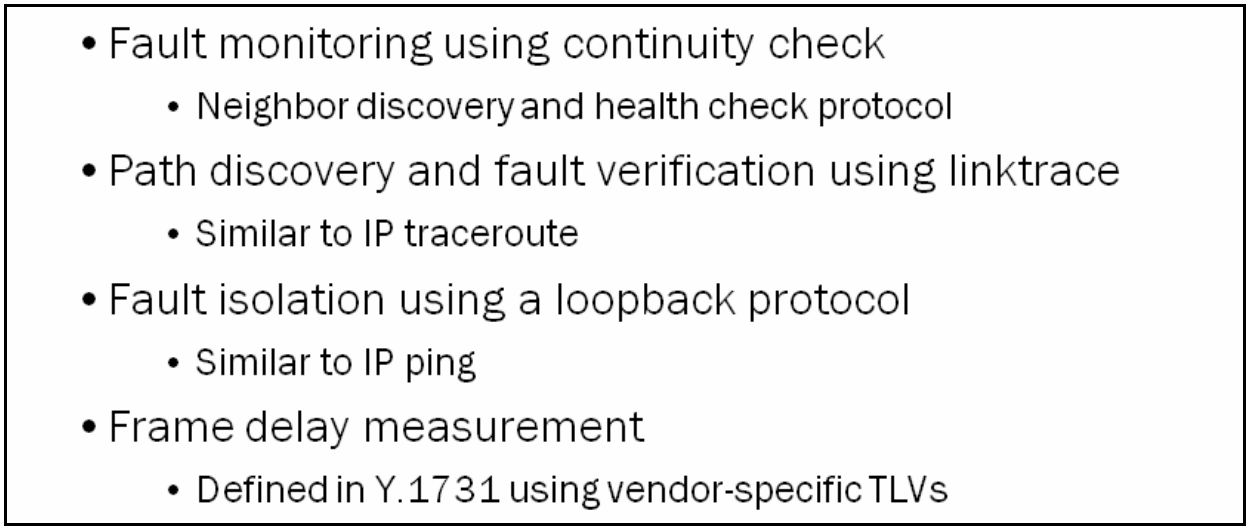
Reacting to Events



When an LFM client receives a critical event, it automatically places the interface in a down state (it removes routes and causes spanning-tree recalculation). It does, however, continue to monitor the interface for LFM messages in the event that the link becomes stable again.

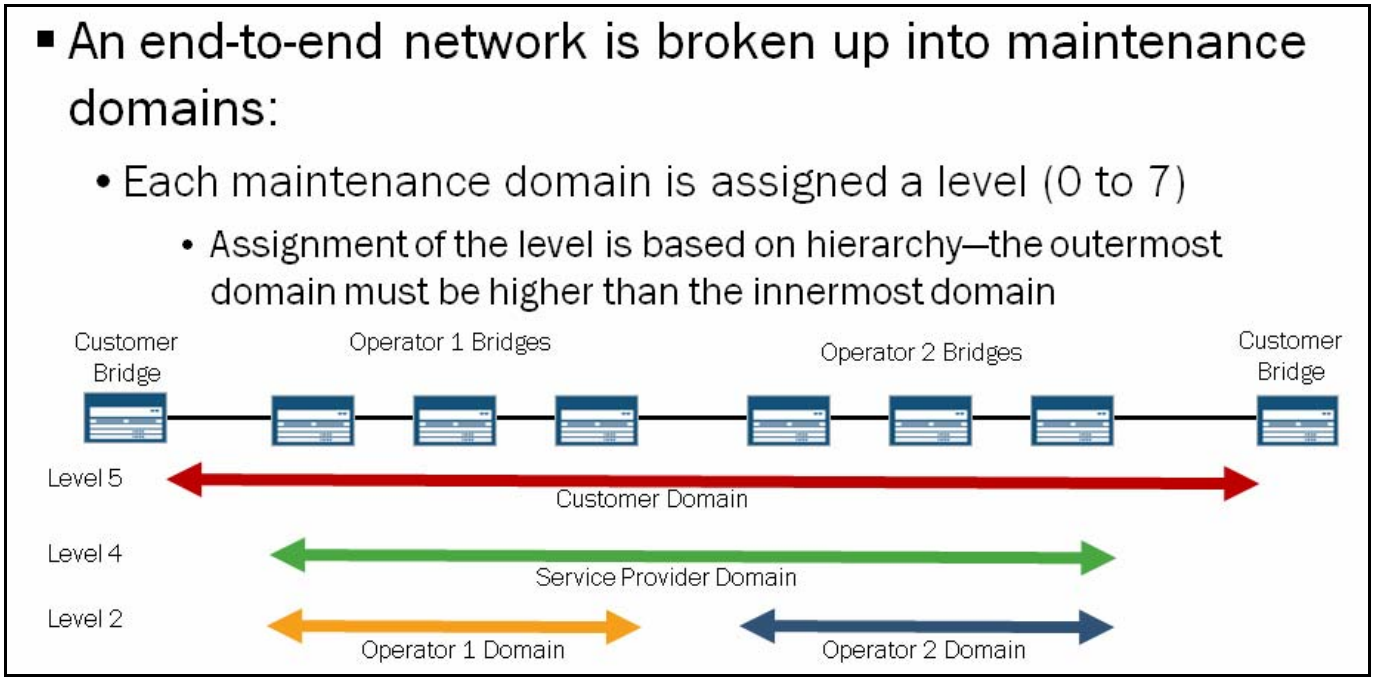
For other types of events, the action an LFM client performs is configurable. To specify the action to be performed for a particular event, you must create an `action-profile` and apply it to an interface. Actions you can configure are generation of a syslog message, placement of the interface in a down state, or the sending of an OAMPDU to the remote peers with the critical event bit set.

CFM Features



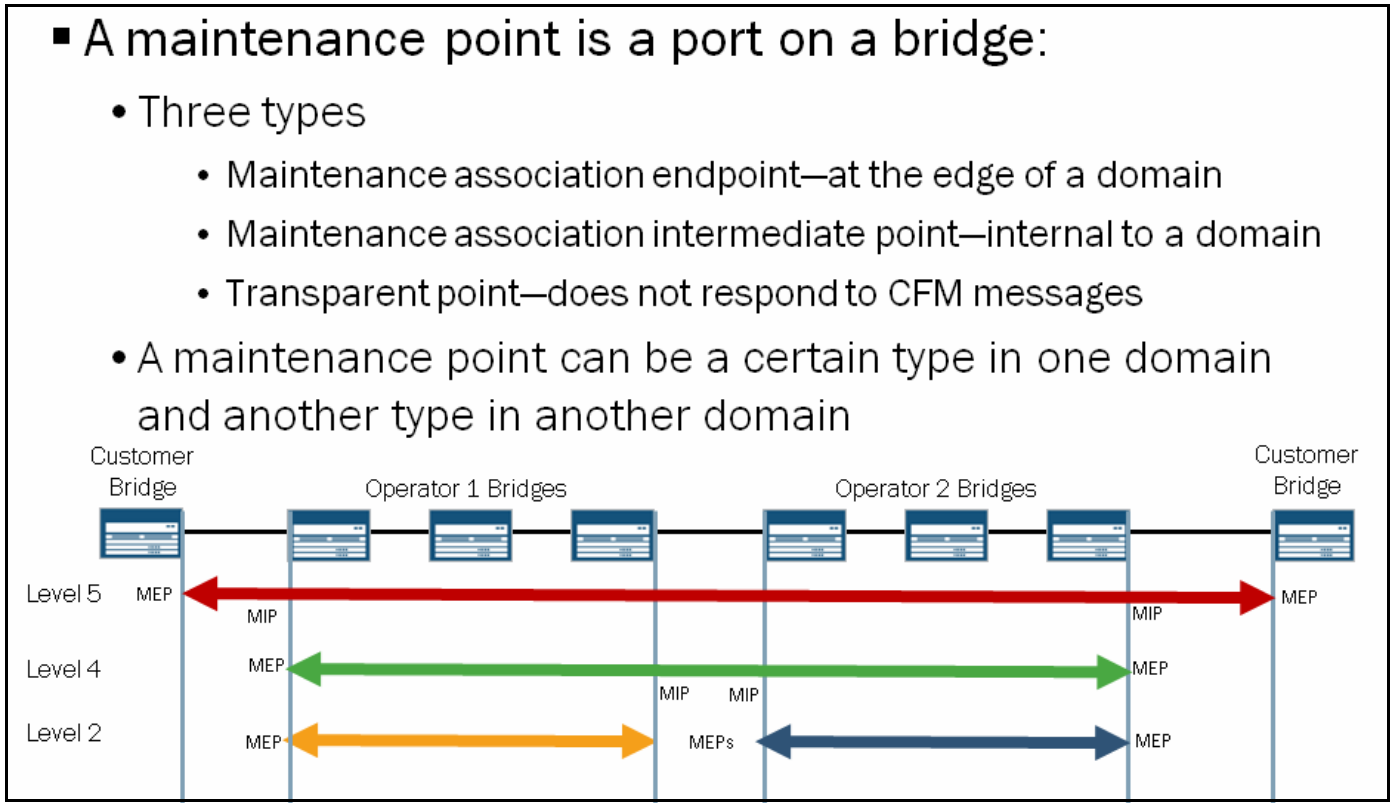
The graphic lists the CFM features that an MX Series 3D Universal Edge Router supports. Other features defined in Y.1731 are not yet supported (such as AIS).

Maintenance Domains



CFM operates in a layered environment. As you will see on the next few slides, this layering allows end-to-end OAM functionality without having to expose all of the details of the service provider network to the customer. Each layer of the CFM network is assigned a maintenance domain ID and a level. A level can be in the range of 0 to 7. Level 5 through Level 7 are reserved for customers, Level 3 and Level 4 are reserved for providers, and Level 0 through Level 2 are reserved for operators. An operator level maintenance domain represents a subset of the provider network. Besides hiding the details of the network from the upper-level maintenance domains, this layering allows for quicker fault detection in the network because a fault detected in the Operator 1 maintenance domain actually eliminates the need to troubleshoot devices in the Operator 2 maintenance domain.

Maintenance Points



A maintenance point is a port or interface on a switch. Three possible types of maintenance points might be present in a maintenance domain. A maintenance domain has at least two maintenance endpoints (MEPs). MEPs are interfaces found at the edge of the maintenance domain. A MEP forms a relationship with a single MEP or several MEPs that are in the same maintenance domain, and level, and that protect the same Ethernet virtual circuit (EVC) (also called a maintenance association). A MEP forms a relationship with a single MEP when protecting an Ethernet Line (E-Line) EVC and forms relationships with multiple MEPs when protecting an Ethernet LAN (E-LAN) EVC. Among other things, MEPs exchange CC messages with each other to ensure that the path between them is up and available.

Another type of maintenance point is a maintenance intermediate point (MIP). MIPs are completely optional. MIPs are used to expose some of the network at a lower maintenance domain level to an upper level. For example, consider the diagram where MIP functionality was configured on the Level 4 MEPs. A linktrace from the customer bridge on the left side of the diagram at Level 5 shows three hops to the customer bridge on the right side. The two MIPs that were configured at Level 4 and the final MEP at Level 5 respond to the linktrace message. MIPs respond only to CFM messages that were received from a MEP at one higher level than their own. The final type of maintenance point is a transparent point. A transparent point is not configured for CFM messages and simply forwards them as regular data traffic.

Maintenance Point Roles

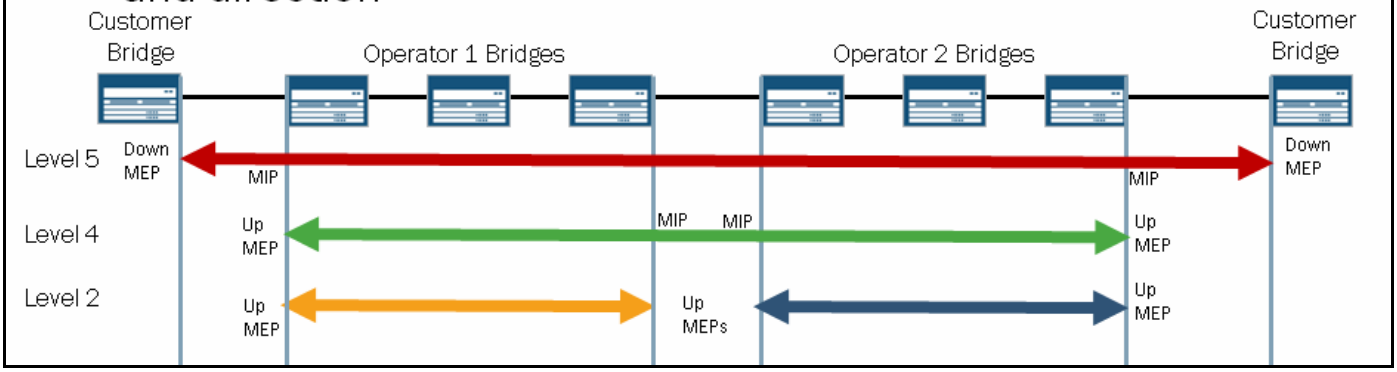
- Each maintenance point has a role to perform:
  - The goal is to maintain the integrity of each domain while still providing each domain with enough information to isolate faults

Tasks	MEP	MIP	Transparent
Initiate CFM messages	Yes	No	No
Respond to loopback and linktrace messages	Yes	Yes	No
Track CC messages	Yes	Yes	No

The graphic shows the roles that each type of maintenance point plays in a CFM network.

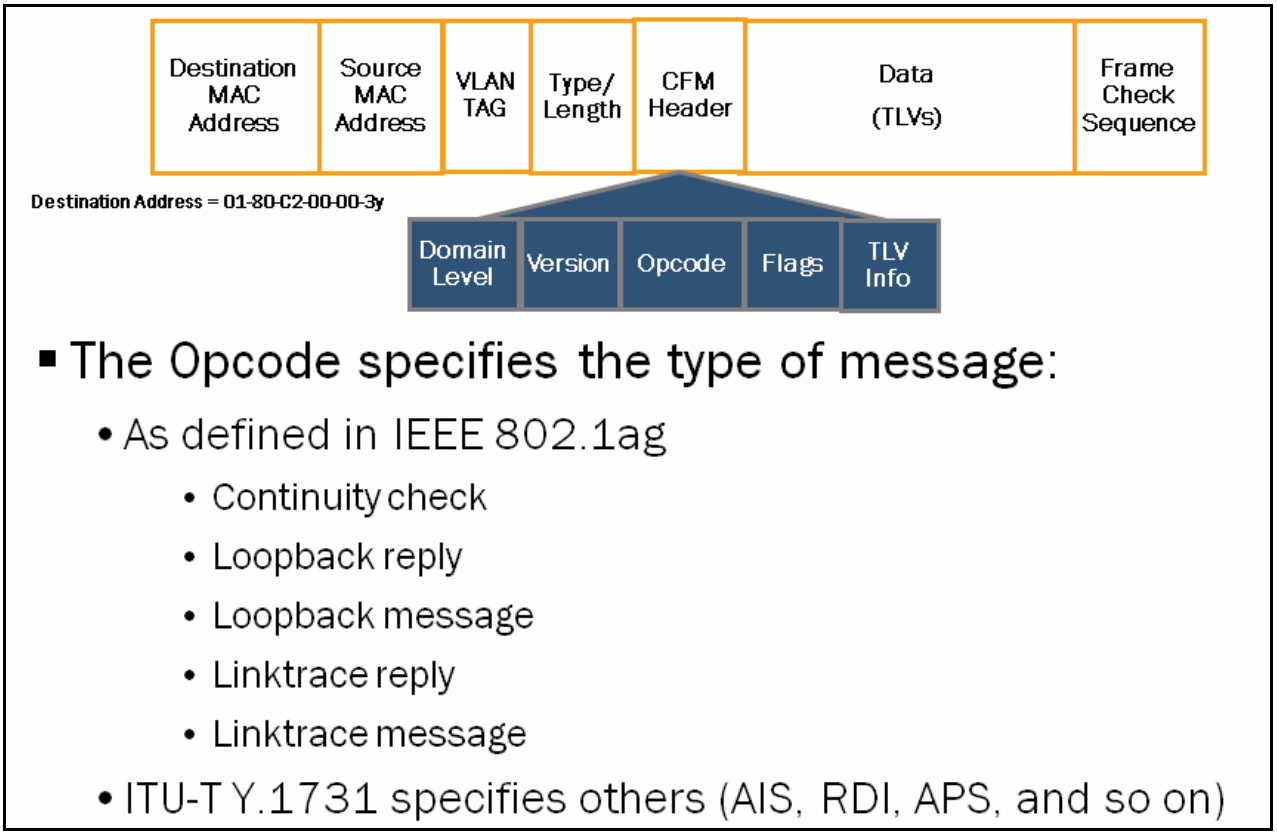
MEP-to-MEP Relationship

- A MEP forms a neighbor relationship with other MEPs in the same domain with the exchange of CC messages
  - Two types of MEPs:
    - Down MEP—A MEP (interface) that faces a neighboring down MEP
    - Up MEP—A MEP (interface) that faces away from a neighboring up MEP
  - To become neighbors, two MEPs must be configured with the same maintenance domain, maintenance association, level, and direction



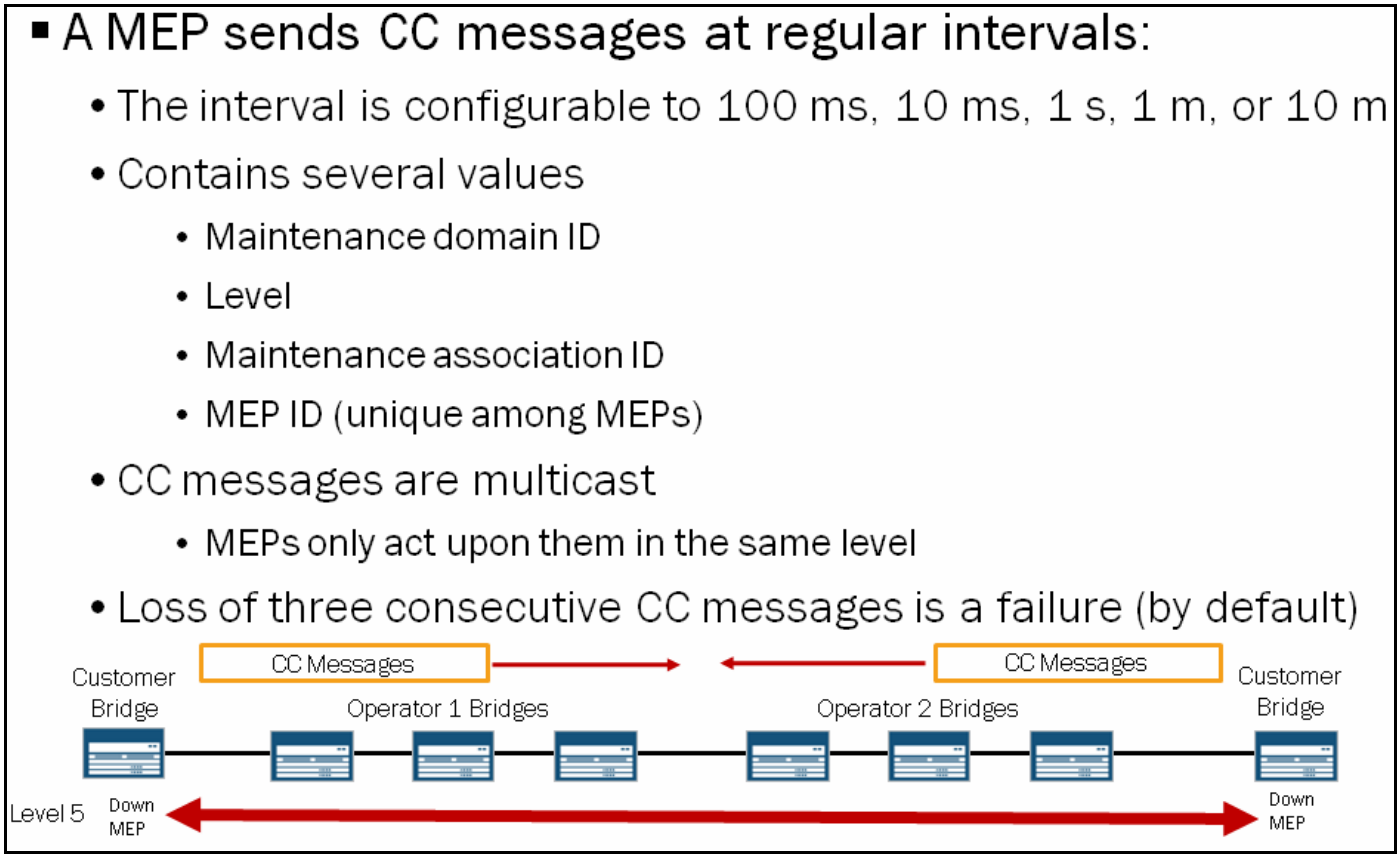
Once a MEP is configured, it attempts to form a neighbor relationship with other MEPs that have are similarly configured. The relationship establishes by means of the exchange of CC messages. Each MEP is configured with a MEP ID (a number). The MEP ID must be unique among all MEPs in the network. Each MEP also is configured with a direction—either up or down. A down MEP expects to find neighboring MEPs downstream. An up MEP expects to find neighboring MEPs upstream. To become neighbors, two MEPs must be configured with the same maintenance domain, maintenance association, level, and direction. This data is carried in each CFM message.

CFM Messages



The graphic shows the format of a CFM message. IEEE 802.1ag defines five types of messages, as shown on the slide. CC and linktrace messages are sent to a multicast destination address. The same specification has reserved a set of type, length, and values (TLVs) to allow for the future expansion of the CFM protocol. The ITU-T uses some of these extended TLVs to allow for Ethernet frame delay measurement, remote defect indications, and more. Note that the last four bits of the destination address represent the level of the sending MEP as well as the type of message. If y equals 0–7, then the message is a CC message destined to the appropriate level. If y equals 8–F, then the message is a linktrace message destined for Levels 0–7, respectively. CFM messages are also encapsulated with the virtual LAN (VLAN) tag of the EVC that is being protected.

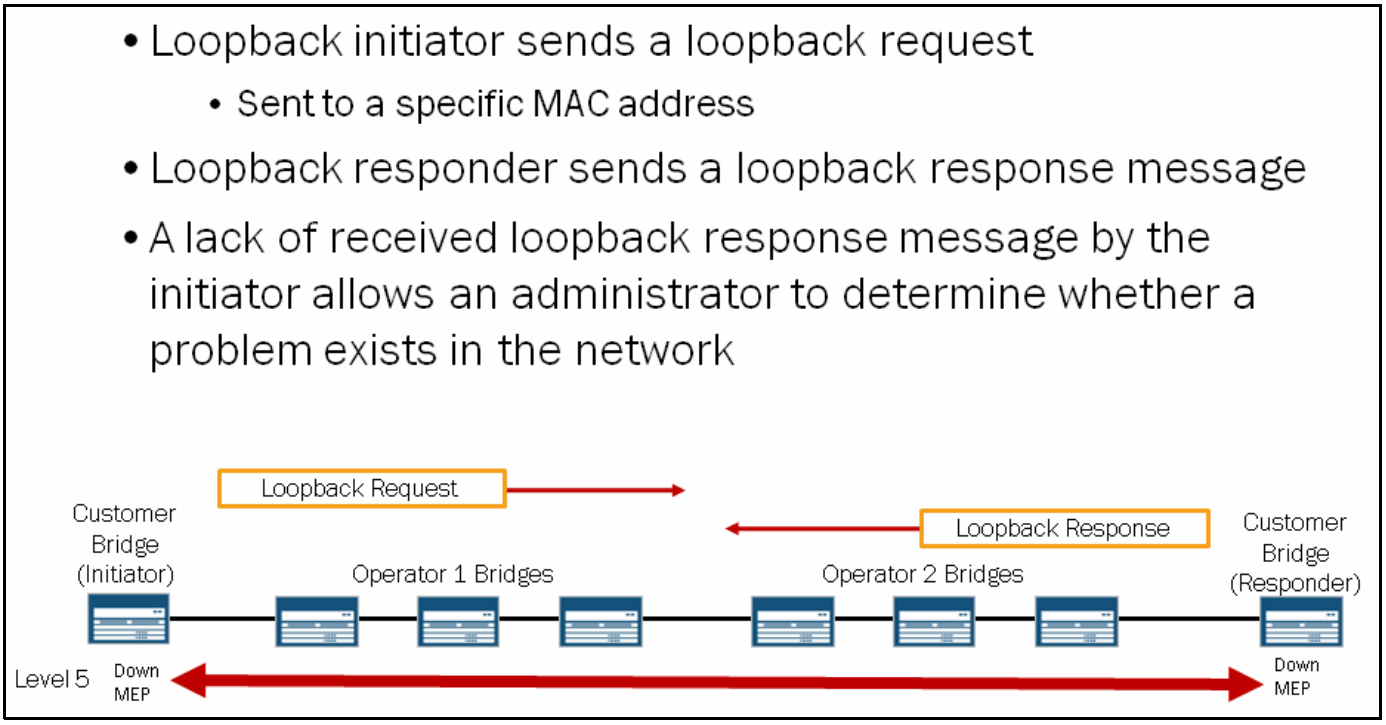
Continuity Check Messages



The graphic shows the details of CC messages. Continuity Check protocol is used to detect connectivity failures and unintended connectivity. Periodically, each MEP transmits a multicast continuity check message (CCM) embedded with identity of the MEP and maintenance association. MEPs use CCM group destination address, 01-80-C2-00-00-3y, for the destination MAC address in CCM frames. The maintenance domain level of the CCM is used for the “y” address bits. For example, if the maintenance domain level of the CCM is 0, then the CCM destination address is 01-80-C2-00-00-30. If the maintenance domain level of the CCM is 7, then the destination address is 01-80-C2-00-00-37. The frames are sent with sequence numbers and multicast frames reduce bandwidth requirements in a full mesh. In addition, they allow detection of accidentally cross-connected MEPs belonging to different service instances. The transmission rate for CCMs is configurable.

As a MEP receives CCMs from other MEPs, it determines any discrepancies between the information received and waited for. The MEP processes CCMs at maintenance domain level equal or lower than its maintenance domain level and uses the information to update its CCM database. Continuity check function on MIPs and the MIP CCM database are optional. A MIP processes CCM at it maintenance domain level and updates the MIP CCM database. The remote MEP considers the loss of three CCMs a failure by default. This loss threshold is configurable.

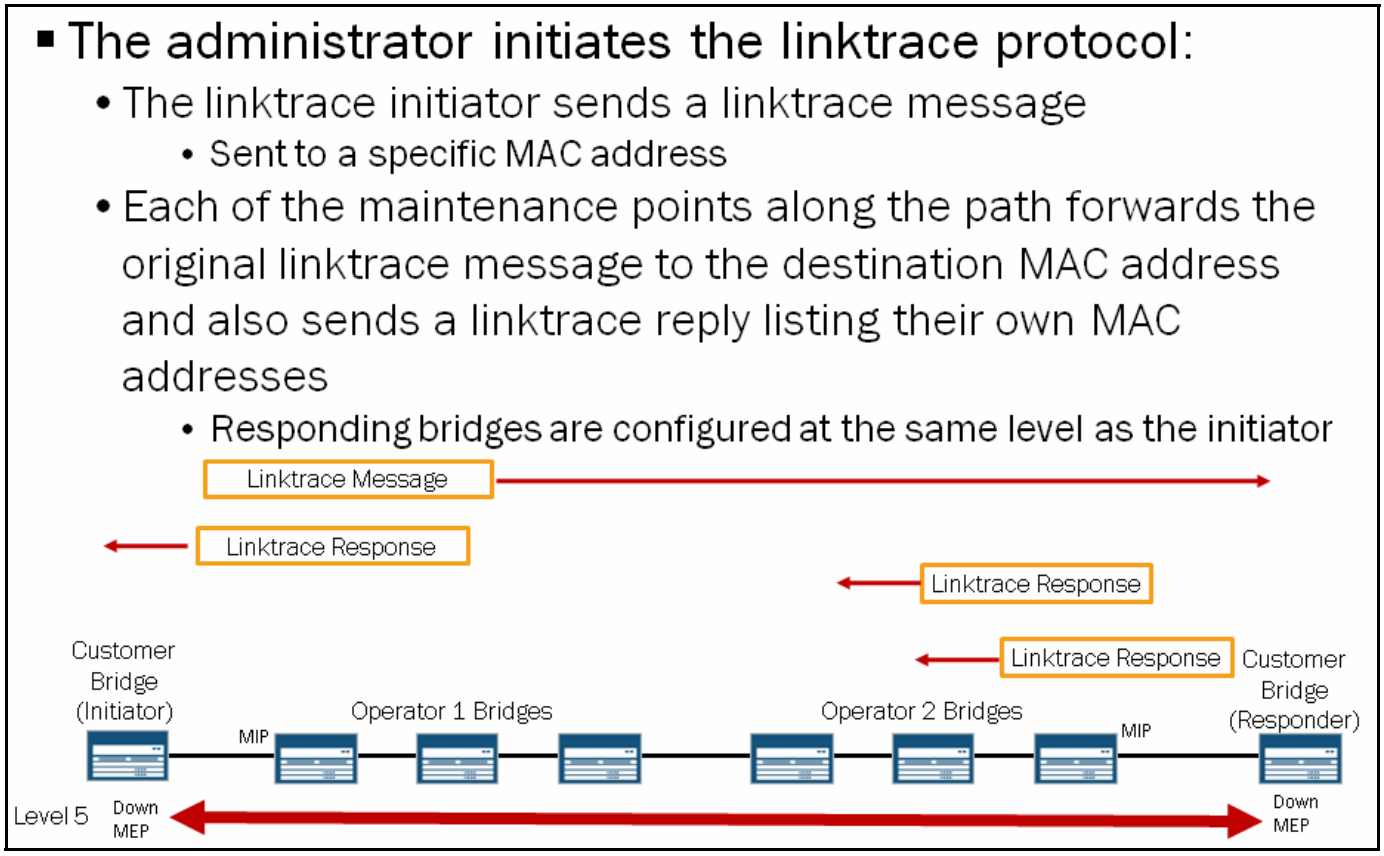
Loopback Protocol



The loopback protocol is similar to ping in IP and is used to verify and isolate connectivity faults. An administrator can trigger a MEP to send one or more loopback messages with an arbitrary amount of data. If the MEP does not receive a valid linktrace reply corresponding to the loopback message, the administrator knows a connectivity fault exists. The receiving MP turns the loopback message, at its maintenance domain level only, into a loopback reply (LBR) back toward the originating MEP. In response to a multicast loopback message frame, the receiving MP waits a random delay between zero and one second before sending an LBR. The source address from the loopback message is used as the destination address for the LBR. The source address of the LBR is the MAC address of the receiving MP. The receiving MP changes the opcode in the frame from loopback message to LBR.

The originating MEP keeps count of the LBRs from other MPs by incrementing in-sequence counter and out-of-sequence counter. It correlates received LBRs with transmitted loopback messages using loopback transaction identifier in the loopback frames. A LBR is valid if it has an expected transaction identifier and is received by the originating MEP within five seconds after transmitting the initial loopback message.

Linktrace Protocol



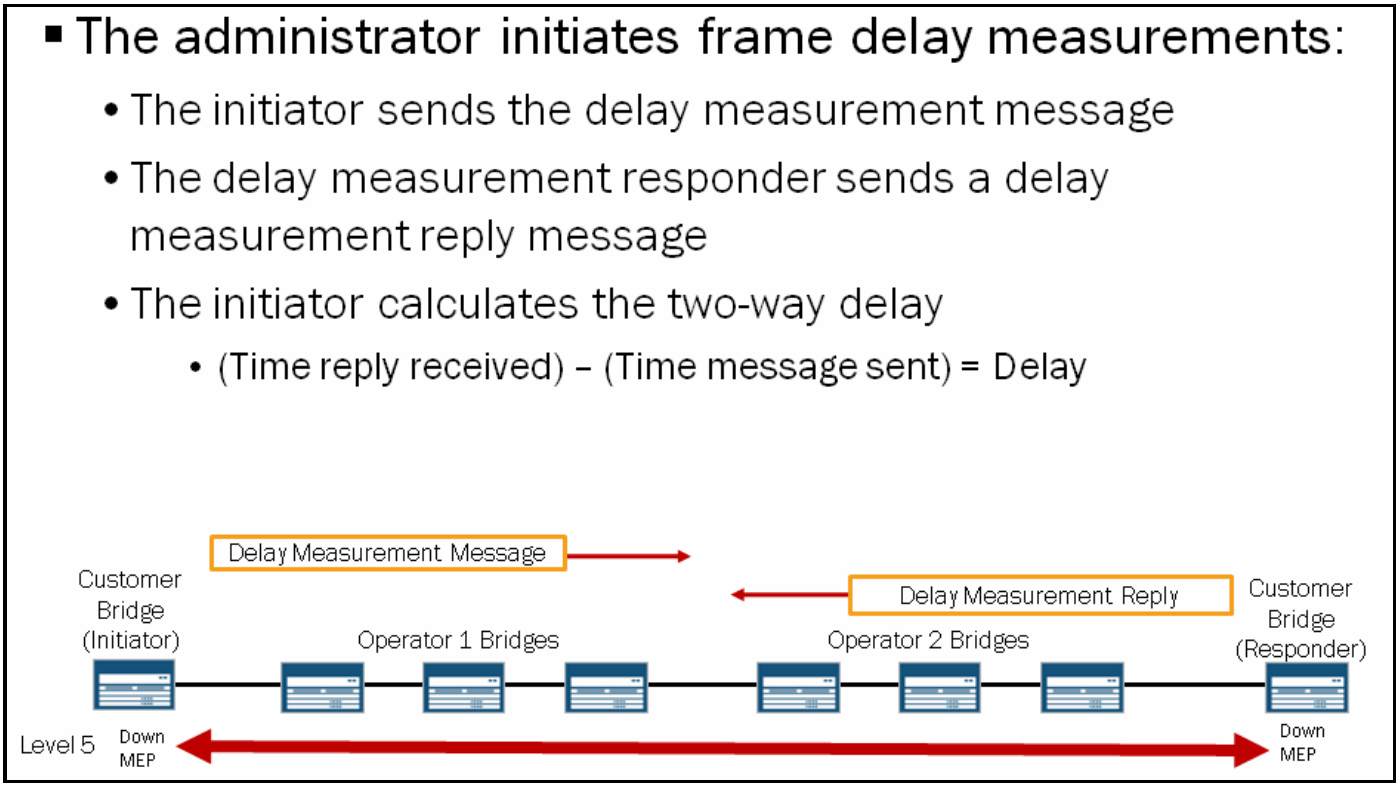
The linktrace protocol is similar to the traceroute function in IP and is used to perform path discovery and fault isolation in a network.

As part of the linktrace protocol, a MEP multicasts a linktrace message using a linktrace message group destination MAC address, 01-80-C2-00-00-3y. The maintenance domain level of the linktrace message plus eight is used for the “y” address bits. For example, a linktrace message at level 0 has destination address 01-80-C2-00-00-38, and a linktrace message at level 7 has destination address 01-80-C2-00-00-3F. The destination address of the replying MP is embedded in the payload of the linktrace message.

A MEP transmits a linktrace message over a maintenance association to neighboring MIPs, from MIP to MIP, to the terminating MP at the end of the path. Only one egress port on a bridge sends linktrace messages. The linktrace message traverses through the bridged network until it reaches a MEP at an equal or higher maintenance domain level or a MIP at an equal maintenance domain level. A MEP at a higher maintenance domain level discards the linktrace message. The linktrace message at an equal maintenance domain level is sent to the linktrace responder.

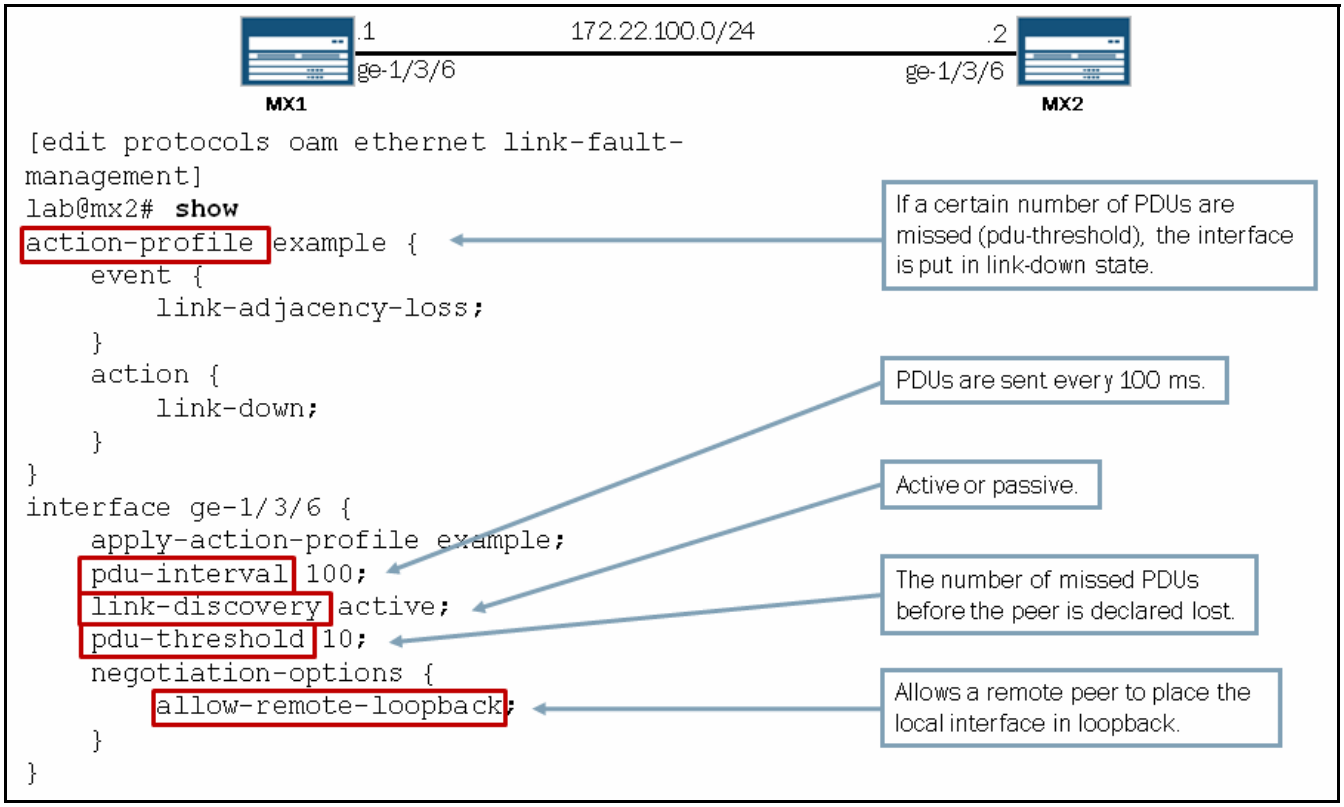
Each linktrace message has a linktrace message transaction identifier. Linktrace message transaction identifiers that are transmitted inside linktrace messages are unique for a MEP for at least five seconds so that linktrace replies from slow MPs can be matched with the corresponding linktrace messages. Using the linktrace replies collected, the originating MEP builds the sequence of MPs traversed by the initial linktrace message. The administrator can then determine the path taken from the MEP to the destination MAC address by examining the sequence of MPs. The difference between the path taken by the linktrace message and the expected sequence helps pinpoint the location of a fault.

Frame Delay Measurement



Two types of delay tests exist: one-way and two-way. An MX Series router uses hardware-assisted timestamping. When an administrator initiates a one-way frame delay test, a delay measurement message is sent to the remote MEP. The delay measurement message contains a timestamp. The remote MEP then calculates the delay from the time the frame was sent to the time it arrived. For the measurement to be accurate, both devices must have their clocks synchronized. A two-way test does not require the two devices to have their clocks synchronized. The slide shows the details of a two-way frame delay test.

LFM Settings



The graphic shows some of the typical LFM configuration settings.

Action Profile

- Use action profiles to specify how a switch should react to certain events:
  - Critical events cause the interface to go into link-down state automatically

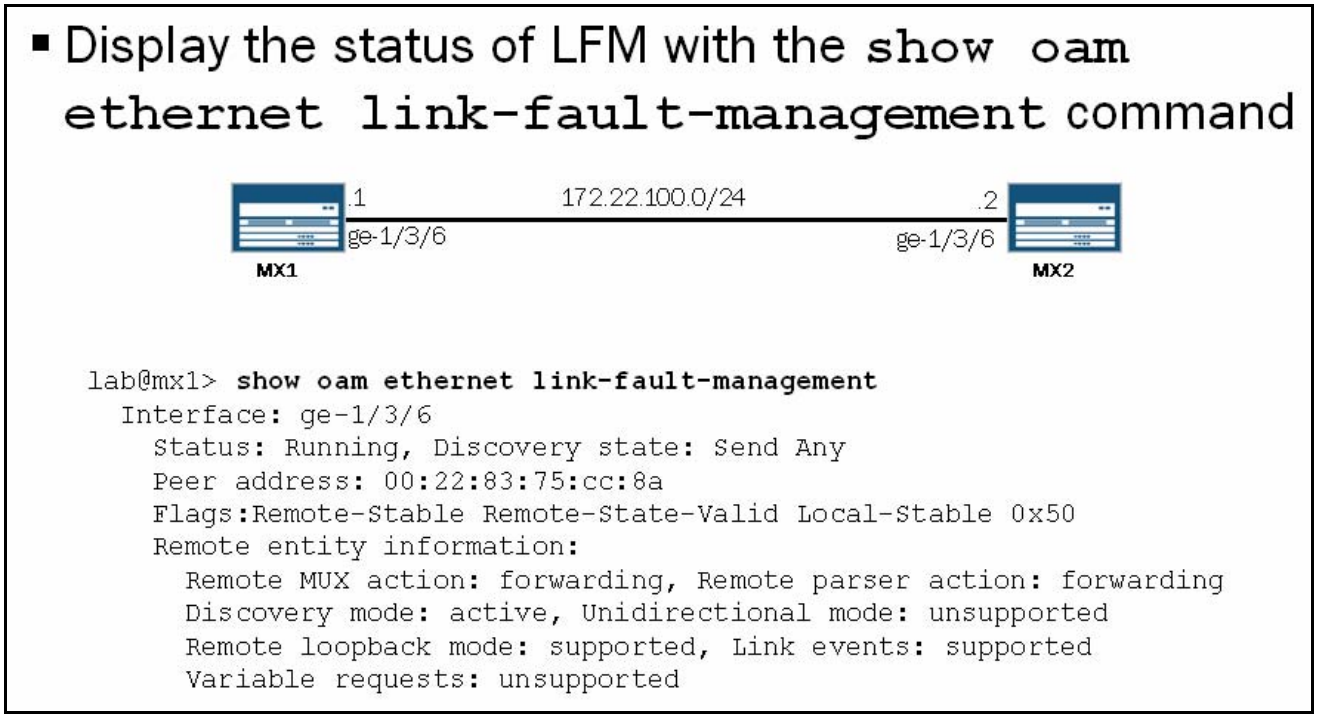
```
[edit protocols oam ethernet link-fault-management]
lab@mx1# set action-profile example event ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  link-adjacency-loss  Loss of adjacency with OAM peer
> link-event-rate
  protocol-down        Upper layer indication on protocol down

lab@mx1# set action-profile example action ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  link-down            Mark the interface down for transit traffic
  send-critical-event  Start sending OAM PDUs with critical event bit set
  syslog               Generate syslog message
```

An action profile allows you to configure how the switch should react to certain events. You configure a profile for the following events:

1. `link-adjacency-loss`: Occurs when CC messages are no longer being received from the remote peer.
2. `link-event-rate`: Allows you to specify a rate of receiving different types of event messages that cause an action to take place.
3. `protocol-down`: Allows the MEP to monitor when maintenance associations at higher levels go down.

The graphic shows the actions you can take when the events specified in the action profile occur.

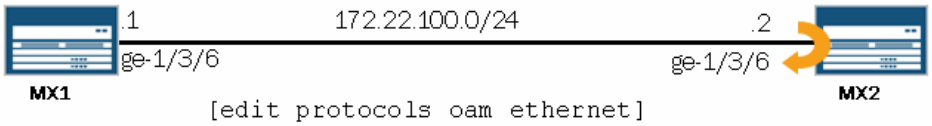
LFM Status



Use the **`show oam ethernet link-fault-management`** command to determine the status of the interfaces running LFM. The output shows the peer's MAC address, the state of the relationship with that peer, and the capabilities of the peer.

Setting a Remote Loop

- Set a loop on the remote peer by adding the `remote-loopback` statement to the configuration



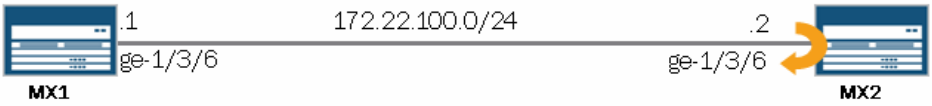

```
[edit protocols oam ethernet]
lab@mx1# show
link-fault-management {
    interface ge-1/3/6 {
        pdu-interval 100;
        link-discovery active;
        pdu-threshold 10;
        remote-loopback;
    }
}
```

```
lab@mx2> show oam ethernet link-fault-management
  Interface: ge-1/3/6
    Status: Running, Discovery state: Send Any
    Peer address: 00:21:59:01:94:8a
    Flags: Remote-Stable Remote-State-Valid Local-Stable 0x50
    Remote loopback status: Enabled on local port, Disabled on peer port
```

In the example on the graphic, MX1 is configured to send a loopback message to MX2 so that MX2 externally loops its ge-1/3/6 interface. Once MX1's configuration is committed, the loopback message travels to MX2. Looking at MX2's command output, you can see that the local interface was placed in a loop.

Testing the Looped Circuit

- Send pings from MX1 to MX2:
  - Success appears at TTL expirations
  - A static Address Resolution Protocol entry is necessary



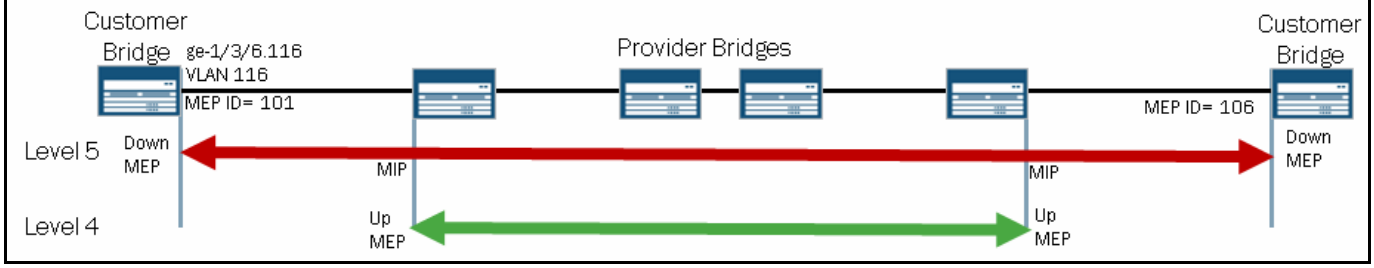

```
lab@mx1> show interfaces ge-1/3/6 | match hardware
  Current address: 00:21:59:01:94:8a, Hardware address: 00:21:59:01:94:8a

[edit interfaces ge-1/3/6 unit 0 family inet]
lab@mx1# set address 172.22.100.1/24 arp 172.22.100.2 mac 00:21:59:01:94:8a

lab@mx1> ping 172.22.100.2
PING 172.22.100.2 (172.22.100.2): 56 data bytes
36 bytes from 172.22.100.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 f7e7   0 0000  01  01 a191 172.22.100.1  172.22.100.2

36 bytes from 172.22.100.1: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 f7eb   0 0000  01  01 a18d 172.22.100.1  172.22.100.2
--- 172.22.100.2 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
```

The graphic shows the steps necessary to generate ping traffic across the looped circuit. A successful test of the link comes in the form of TTL expiration messages. The time-to-live (TTL) expirations show that the pings were successfully transmitted, received, and looped through the network until the TTL finally expired on the Internet Control Message Protocol (ICMP) packets.

Down MEP Configuration

- Remote MEP must be configured similarly to the local MEP:
  - Remote MEP must have the same maintenance domain, maintenance association, interval, and level

```
[edit protocols oam ethernet connectivity-fault-
management]
lab@switch1# show
action-profile evc1-profile {
    event {
        adjacency-loss;
    }
    action {
        interface-down;
    }
}
maintenance-domain customer {
    level 5;
    maintenance-association evc1 {
        continuity-check {
            interval 100ms;
        }
        mep 101 {
            interface ge-1/3/6.116 vlan 116;
            direction down;
            auto-discovery;
            remote-mep 106 {
                action-profile evc1-profile;
…
```

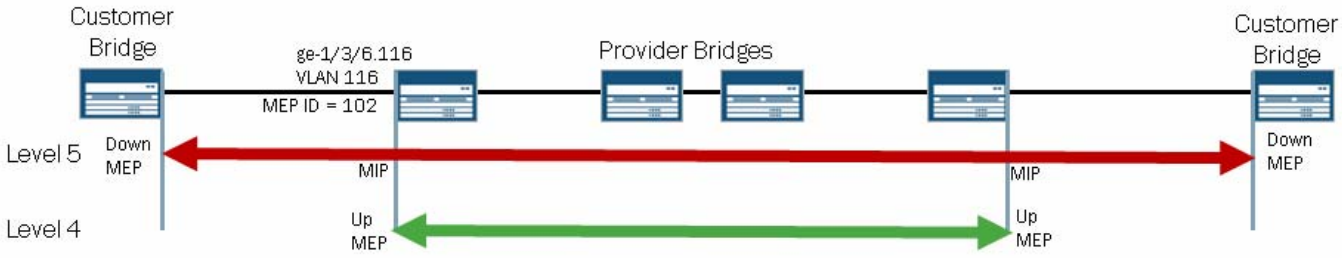


Any given MEP must be configured with a named maintenance domain, a level, a named maintenance association, a unique MEP ID, a direction, and a remote MIP ID (autodiscovery can be used as well). Also, a CC message interval must be specified to begin the neighbor discovery process and monitoring of the end-to-end EVC. The graphic also shows how to apply an action profile to a remote MEP. You cannot apply an action profile when using autodiscovery.

Up MEP Configuration

- You configure a provider edge bridge as a MEP and also as a MIP:
  - It acts as a MIP only for level 5 (level 4 + 1)

```
[edit protocols oam ethernet connectivity-fault-management]
lab@switch2# show
maintenance-domain provider {
    level 4;
    maintenance-association evc1 {
        continuity-check {
            interval 100ms;
        }
        mip-half-function default;
        mep 102 {
            interface ge-1/3/6.116 vlan 116;
            direction up;
            auto-discovery;
        }
    }
}
```



The graphic shows a typical configuration of an up MEP. To allow for an up MEP to also act as a MIP for a higher level, simply add the **`mip-half-function default`** statement to the configuration.

Status of CFM

```
lab@switch1> show oam ethernet connectivity-fault-management ?
Possible completions:
  delay-statistics     Show Ethernet OAM maintenance endpoint delay statistics information
  forwarding-state     Show Ethernet OAM forwarding state for received packets
  interfaces           Show Ethernet OAM information for interface
  mep-database         Show Ethernet OAM maintenance endpoint database information
  mep-statistics       Show Ethernet OAM maintenance endpoint statistics information
  mip                  Display MIP information
  path-database        Display the linktrace path-database for a remote host
  policer              Show Ethernet OAM policer information
```

The graphic shows all of the possible troubleshooting commands that you can use for CFM.

CC Status

- View the status of the switch's MEP-neighbor relationships

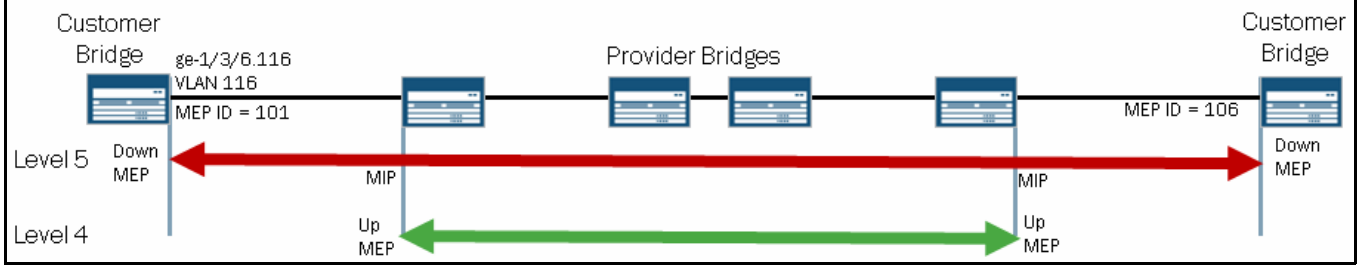
```
lab@switch2> show oam ethernet connectivity-fault-management interfaces ge-1/3/6.116 vlan 116
Interface        Link       Status           Level       MEP          Neighbors
                                                         Identifier
ge-1/3/6.116     Up         Active           4           102          1

lab@switch2> show oam ethernet connectivity-fault-management interfaces ge-1/3/6.116 vlan 116 extensive
Interface name: ge-1/3/6.116, vlan 116, Interface status: Active, Link status: Up
  Maintenance domain name: provider, Format: string, Level: 4
    Maintenance association name: evc1, Format: string
    Continuity-check status: enabled, Interval: 100ms, Loss-threshold: 3 frames
    Interface status TLV: none, Port status TLV: none
    MEP identifier: 102, Direction: up, MAC address: 00:22:83:75:cc:8a
    MEP status: running
    Defects:
      Remote MEP not receiving CCM                 : no
      Erroneous CCM received                       : no
      Cross-connect CCM received                   : no
      RDI sent by some MEP                         : no
      Some remote MEP's MAC in error state         : no
    Statistics:
      …
    Remote MEP count: 1
      Identifier    MAC address        State    Interface
        105     00:21:59:ad:f4:8a       ok  ge-1/3/5.116
```

Use the **`show oam ethernet connectivity-fault-management interface`** command to determine the status of a MEP's relationship with a remote MEP.

CFM Loopback

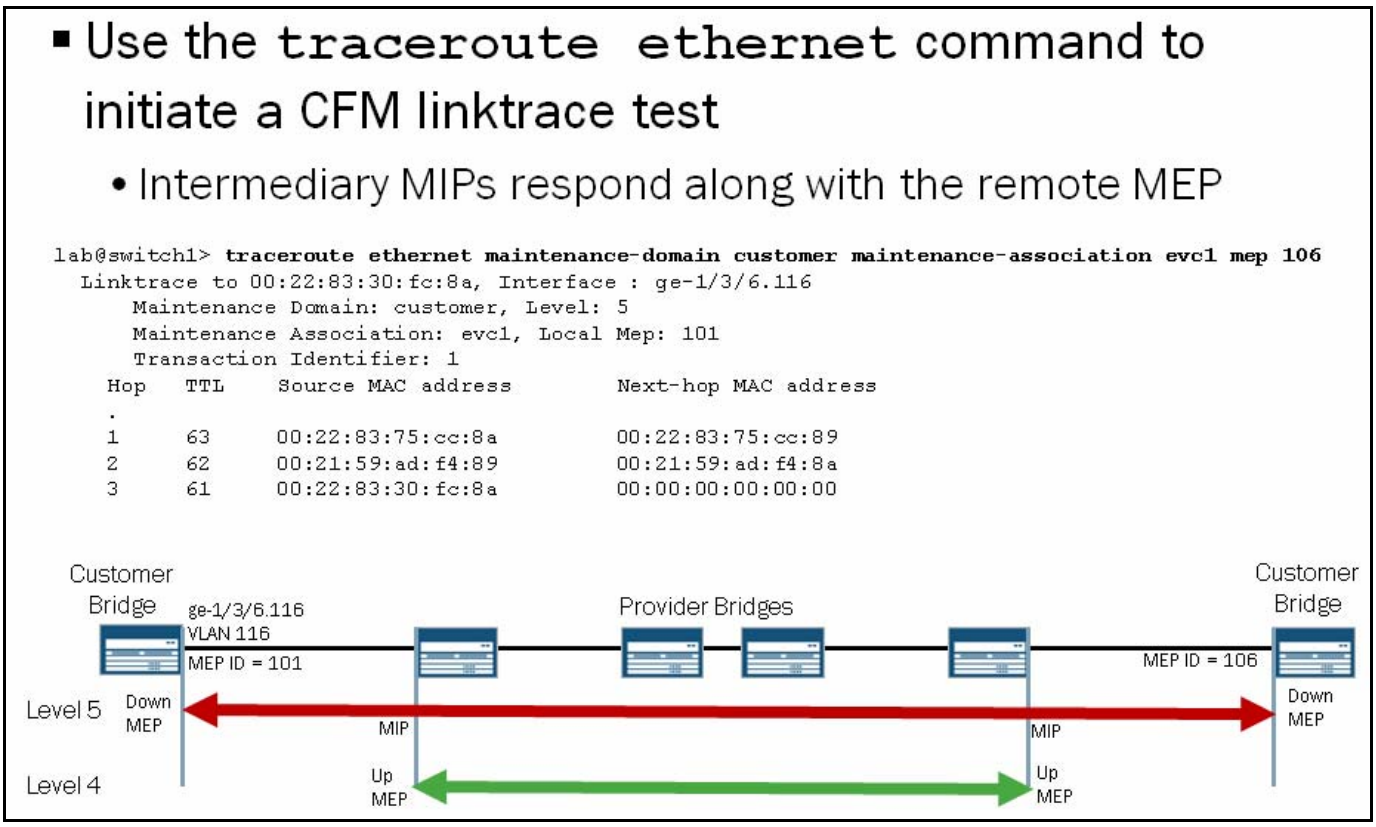
- Use the `ping ethernet` command to initiate a CFM loopback test
  - Specify the remote MEP by MEP ID or MAC address

```
lab@switch1> ping ethernet maintenance-domain customer maintenance-association evc1 mep 106
PING to 00:22:83:30:fc:8a, Interface ge-1/3/6.116
68 bytes from 00:22:83:30:fc:8a: lbm_seq=0
68 bytes from 00:22:83:30:fc:8a: lbm_seq=1
68 bytes from 00:22:83:30:fc:8a: lbm_seq=2
68 bytes from 00:22:83:30:fc:8a: lbm_seq=3
--- ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss

lab@switch1> ping ethernet maintenance-domain customer maintenance-association evc1 00:22:83:30:fc:8a
PING to 00:22:83:30:fc:8a, Interface ge-1/3/6.116
68 bytes from 00:22:83:30:fc:8a: lbm_seq=4
68 bytes from 00:22:83:30:fc:8a: lbm_seq=5
68 bytes from 00:22:83:30:fc:8a: lbm_seq=6
68 bytes from 00:22:83:30:fc:8a: lbm_seq=7
--- ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```



The graphic shows how to perform a loopback test with the **`ping ethernet`** command.

Linktrace

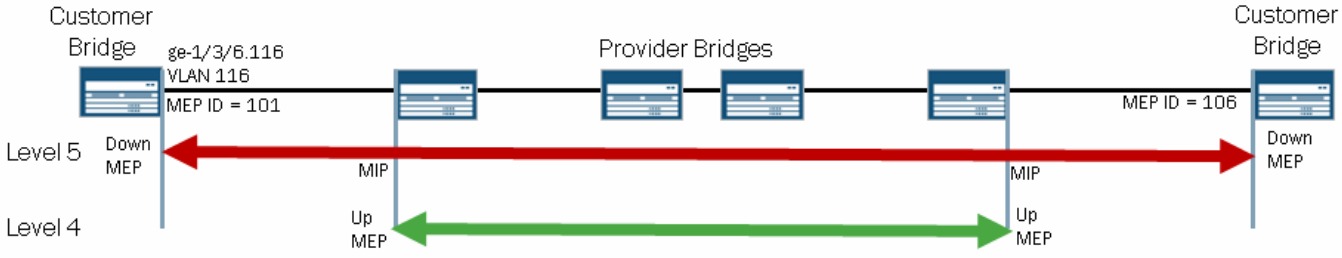


To perform a CFM linktrace to a remote MEP, issue the **`traceroute ethernet`** command. Note that any MIPs configured at Level 4 also respond to the linktrace message initiated by a Level 5 MEP.

Frame Delay Measurement

- Use the `monitor ethernet delay-measurement` command to initiate a CFM frame delay test

```
lab@switch1> monitor ethernet delay-measurement maintenance-domain customer maintenance-association
evc1 mep 106 two-way
Two-way ETH-DM request to 00:22:83:30:fc:8a, Interface ge-1/3/6.116
DMR received from 00:22:83:30:fc:8a Delay: 232 usec Delay variation: 0 usec
DMR received from 00:22:83:30:fc:8a Delay: 201 usec Delay variation: 31 usec
DMR received from 00:22:83:30:fc:8a Delay: 187 usec Delay variation: 14 usec
DMR received from 00:22:83:30:fc:8a Delay: 180 usec Delay variation: 7 usec
DMR received from 00:22:83:30:fc:8a Delay: 197 usec Delay variation: 17 usec
DMR received from 00:22:83:30:fc:8a Delay: 178 usec Delay variation: 19 usec
DMR received from 00:22:83:30:fc:8a Delay: 199 usec Delay variation: 21 usec
DMR received from 00:22:83:30:fc:8a Delay: 163 usec Delay variation: 36 usec
DMR received from 00:22:83:30:fc:8a Delay: 193 usec Delay variation: 30 usec
DMR received from 00:22:83:30:fc:8a Delay: 179 usec Delay variation: 14 usec

--- Delay measurement statistics ---
Packets transmitted: 10, Valid packets received: 10
Average delay: 190 usec, Average delay variation: 18 usec
Best case delay: 163 usec, Worst case delay: 232 usec
```



The graphic shows how to use the **`monitor ethernet delay-measurement`** command to test the frame delay between MEPs.

Save Frame Delay Measurements

- The switch maintains a record of previous frame delay measurements

```
lab@switch1> show oam ethernet connectivity-fault-management mep-statistics maintenance-
domain customer maintenance-association evc1
…
Delay measurement statistics:
    Index  One-way delay  Two-way delay
              (usec)         (usec)
      1                      191
      2                      189
      3                      182
      4                      185
      5                      173
      6                      160
      7                      189
      8                      184
      9                      182
      10                     205
      11                     232
      12                     201
      13                     187
      14                     180
      15                     197
      16                     178
      17                     199
      18                     163
      19                     193
      20                     179
    Average two-way delay          : 187 usec
    Average two-way delay variation: 15 usec
    Best case two-way delay        : 160 usec
    Worst case two-way delay       : 232 usec
```

To go back and look at historical frame delay measurements, issue the **`show oam ethernet connectivity-fault-management mep-statistics`** command.

Review Questions

1. Which Ethernet OAM protocol allows for setting a loop on a remote switch's interface?
2. What is the difference between an up MEP and a down MEP?
3. What must be true for a MIP to respond to a linktrace message?

Answers

1.

LFM allows for setting a remote loop.

2.

A down MEP expects to find neighboring MEPs downstream. An up MEP expects to find neighboring MEPs upstream.

3.

The MIP must be configured at one level below the MEP that initiated the linktrace message.

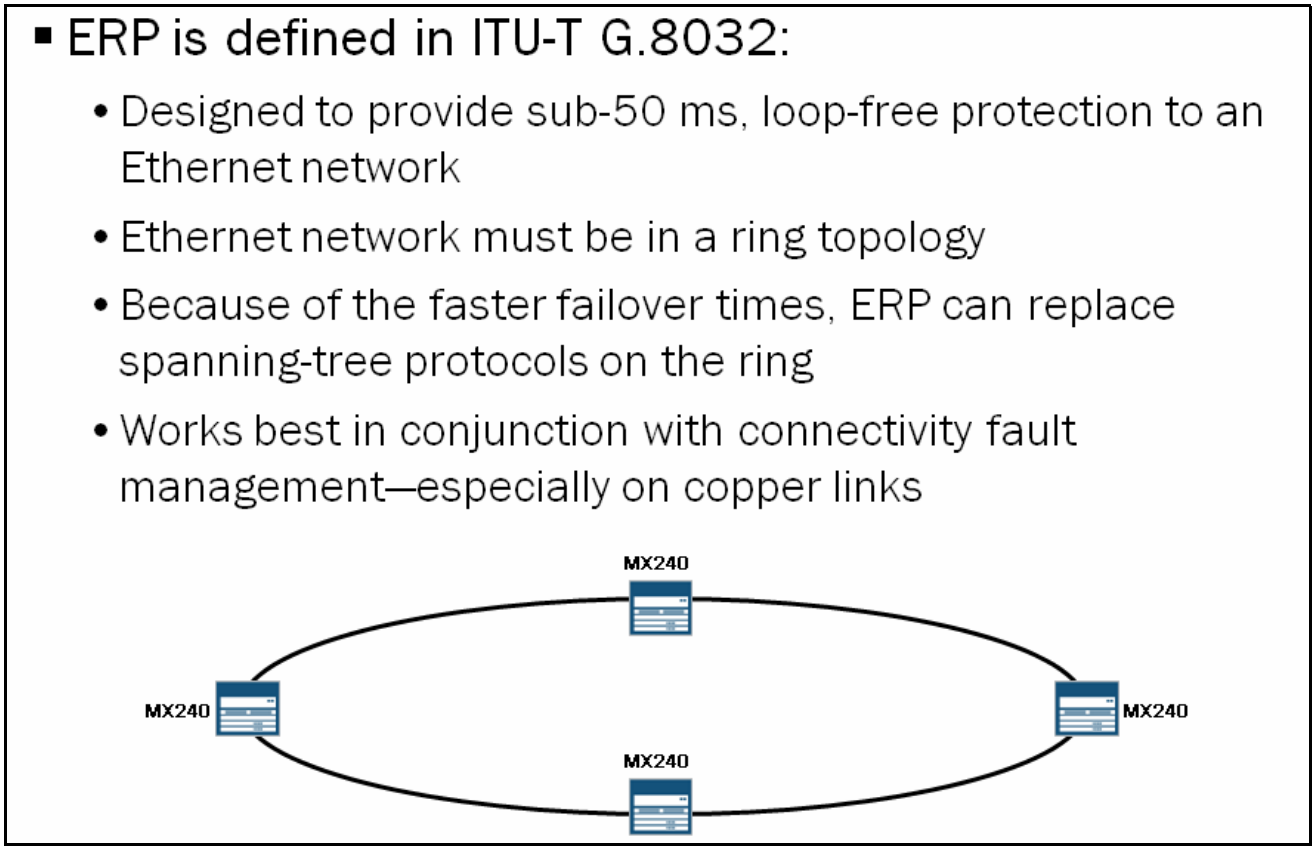
JNCIS-SP Study Guide—Part 2

Chapter 7: High Availability and Network Optimization

This Chapter Discusses:

- Ethernet Ring Protection (ERP);
- Configuration and monitoring of ERP;
- Link aggregation groups (LAGs); and
- Configuration and monitoring of LAGs.

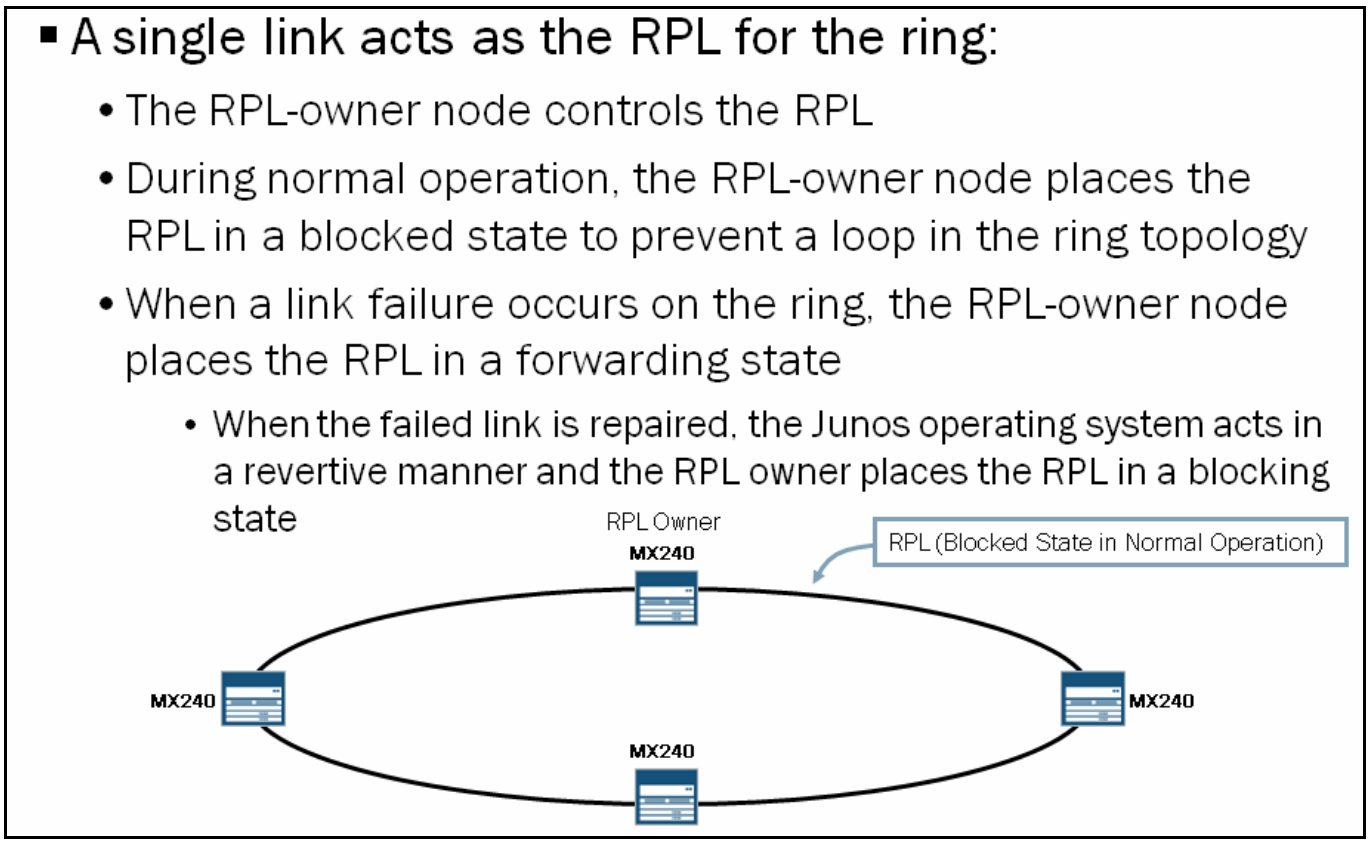
ERP



Defined in the International Telecommunication Union Telecommunication Standardization (ITU-T) G.8032 recommendation, ERP provides highly reliable, stable, and loop-free protection for Ethernet ring topologies. ERP is a solution for an Ethernet ring where each ring node (switch) connects to two adjacent nodes, participating in the same ring, using two independent links. The minimum number of nodes on a ring is two. Because ERP can provide sub-50 ms, loop-free protection for a ring topology, it can viably replace any spanning-tree protocol on the ring. Using an Ethernet fiber ring of less the 1200 km and less than 16 nodes,

the switch completion time at the time of failure should be less than 50 ms. Copper links can also be used, but we recommend that you use connectivity fault management (CFM) to help detect failures between nodes.

Ring Protection Link

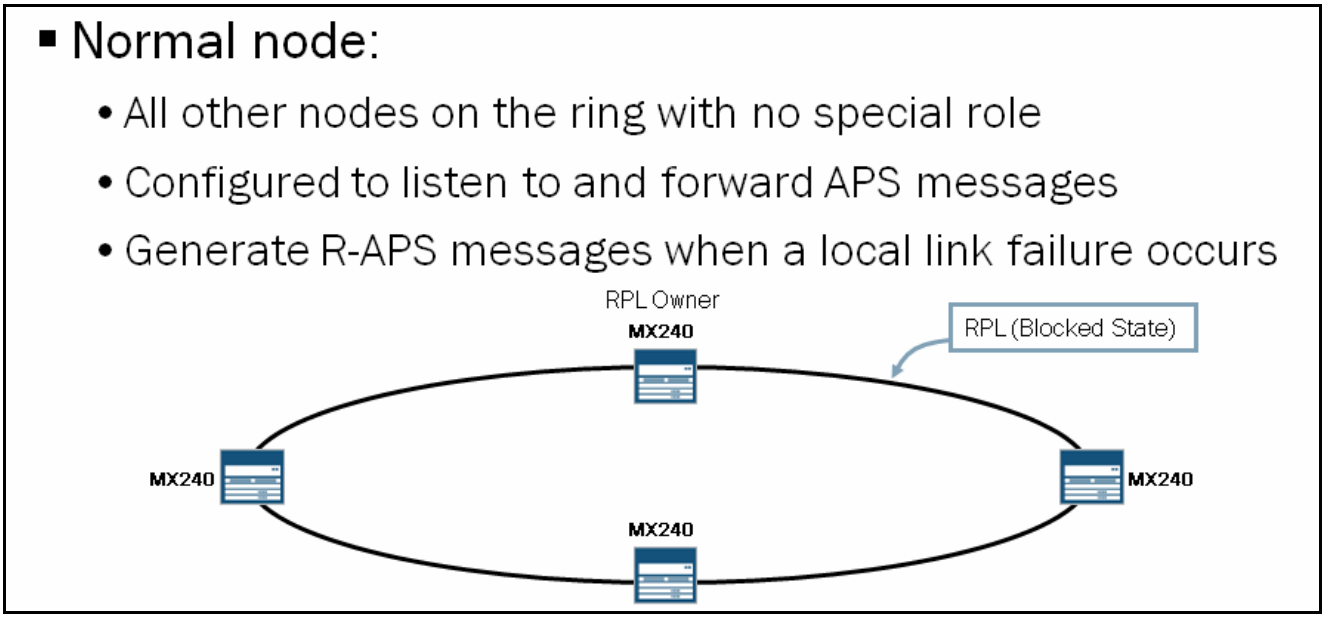


To protect the Ethernet ring, a single link between two nodes acts as the ring protection link (RPL) on the ring. One of the adjacent nodes, which is referred to as the RPL owner, controls the state of the RPL. During normal operation with no failures (idle state), the RPL owner places the RPL in the blocking state, which results in a loop-free topology. If a link failure occurs somewhere on the ring, the RPL owner places the RPL in a forwarding state until the failed link is repaired. Once the failed link is repaired, the Junos operating system acts in a revertive manner, returning the RPL to the blocking state.

RPL-Owner Node

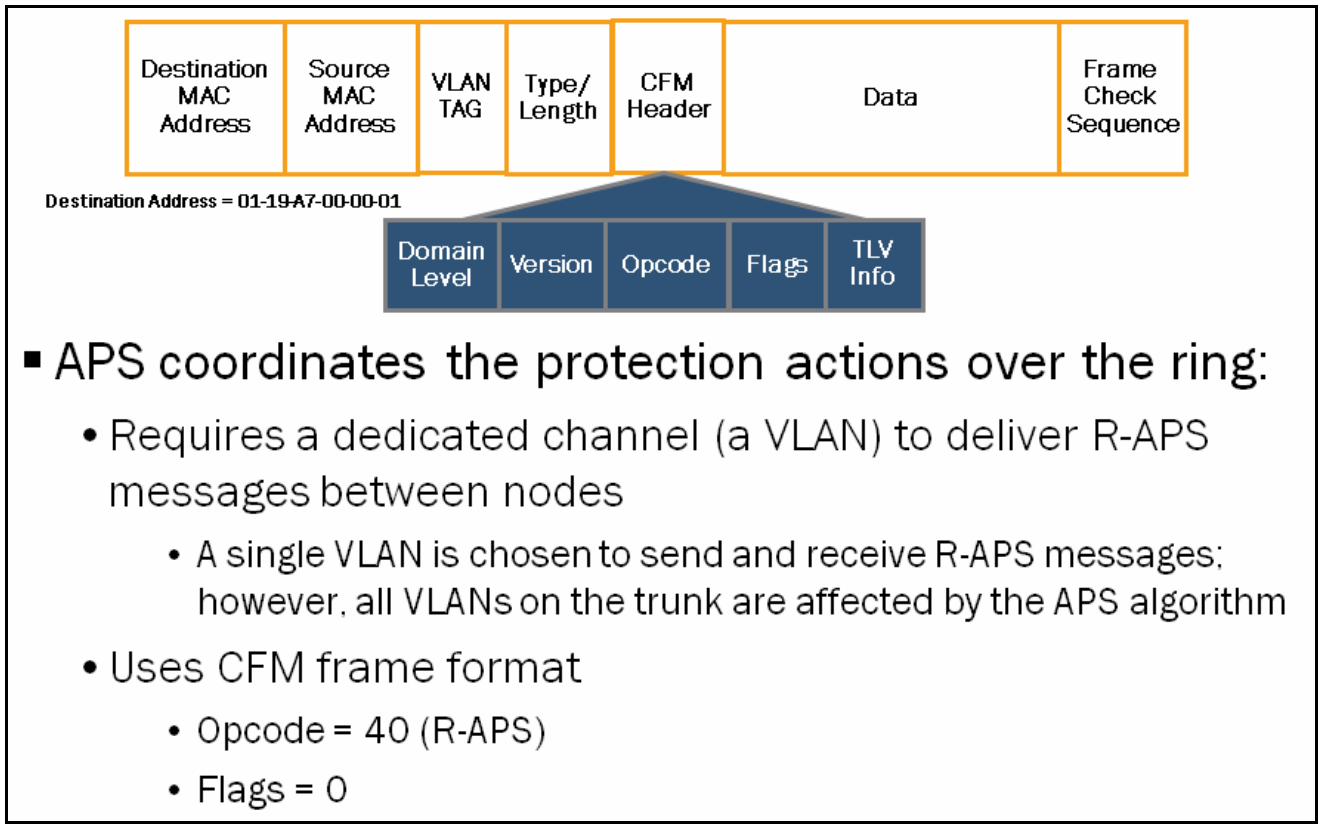
The RPL owner controls the state of the RPL. During the idle state, it is the only node that sends periodic Ring Automatic Protection Switching (R-APS) messages to notify the other nodes about the state of the RPL. The next few slides discuss the details of the Automatic Protection Switching (APS) protocol and R-APS messages.

Normal Node



A normal node is any other node on the ring besides the RPL owner. It listens to and forwards R-APS messages. Also, if a local ring link failure occurs, a normal node signals all other nodes that the failure has occurred using R-APS messages.

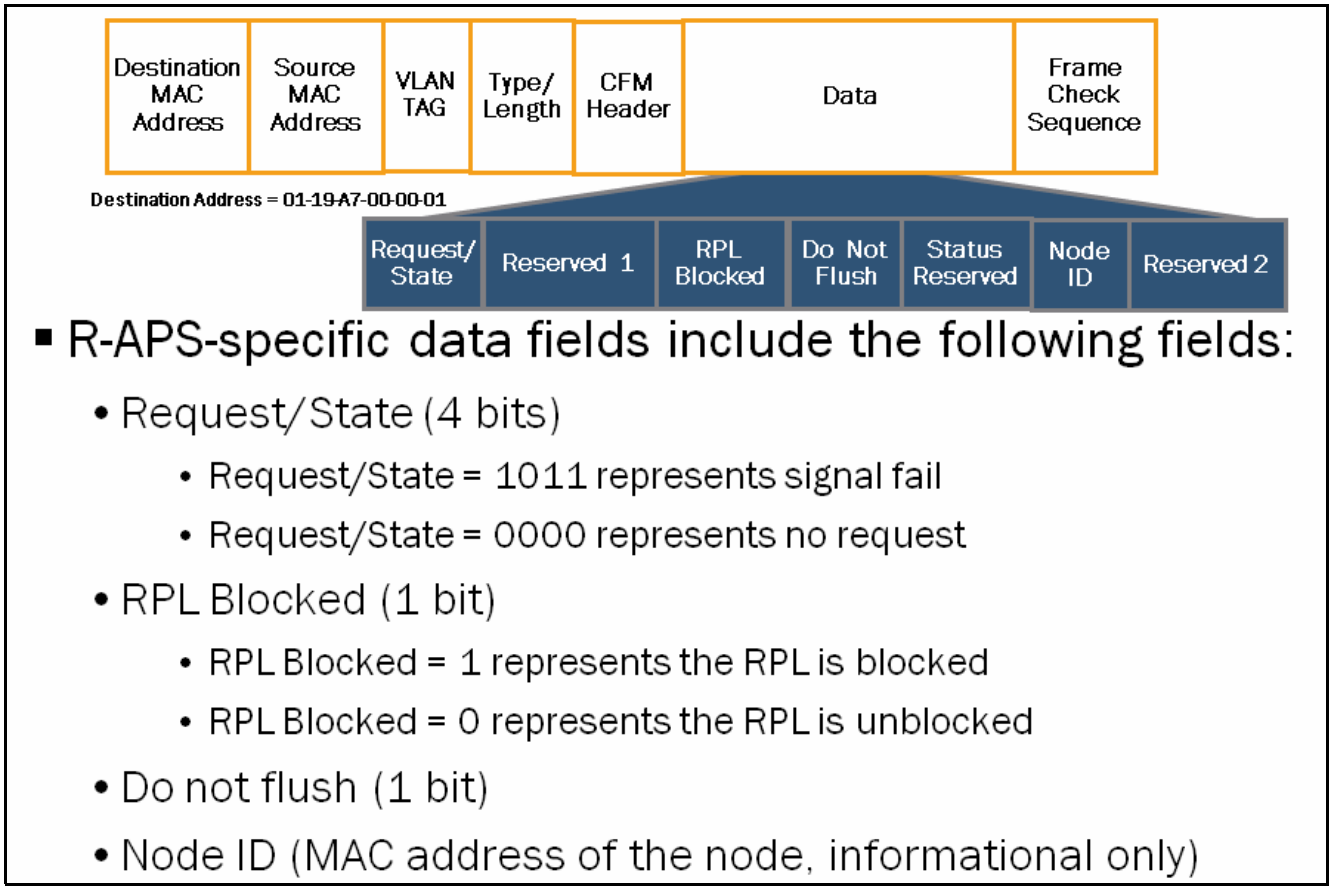
APS



To coordinate the effort of protecting the Ethernet ring, each node participates in the APS. Each of the two ports on each node must be configured for a dedicated channel—a virtual LAN (VLAN) or a bridge domain—to communicate using the APS protocol. Although the APS protocol uses a single VLAN to communicate, the changes in the forwarding state of interfaces that occur as a result of the exchange of R-APS messages affect the entire port of a node (all VLANs). ITU-T G.8032 specifies the use of the CFM frame format as described in the Operation, Administration, and Maintenance (OAM) chapter of this study guide. To allow

differentiation between an R-APS message from a CFM message, an R-APS message uses a destination address of 01-19-A7-00-00-01, as well as an opcode of 40.

R-APS Data Fields

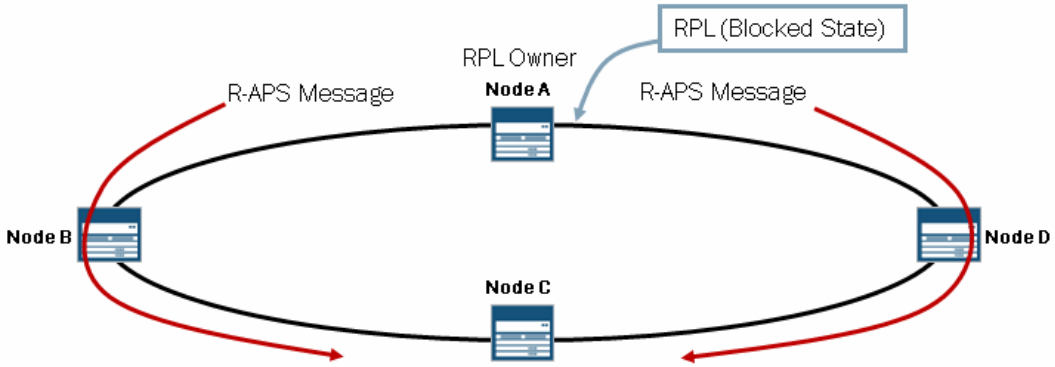


Currently, APS has no specified type, length, and values (TLVs). The slide shows the data fields found in an R-APS message. The following list describes each data field:

- *Request/State* (4 bits): Currently only two values are defined. A value of 0000 is used when a node wants to signal that it detects no failure on the ring (No request). A value of 1011 is used when a node wants to signal that an interface has failed (Signal Fail state).
- *Reserved 1* (4 bits): This value is always 0000. This field is reserved for future use.
- *RPL Blocked* (1 bit): Usage for this field is shown on the slide. Only the RPL owner can signal RPL Blocked.
- *Status Reserved* (6 bits): This value is always 000000. This field is reserved for future use.
- *Node ID* (6 octets): This field is a MAC address unique to the ring node.
- *Reserved 2* (24 octets): This value is all zeros. This field is reserved for future use.

Idle State

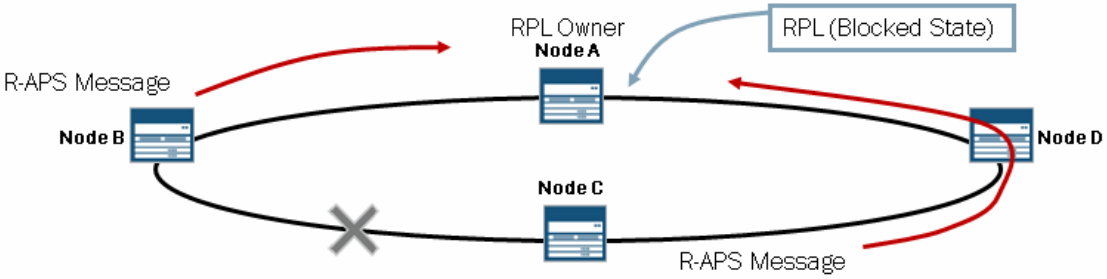
- RPL-owner node places RPL in the blocked state:
  - RPL owner (Node A) sends R-APS messages out of all ports every 5 seconds
    - Request/State = No request
    - Do not flush = 0 (Flush)
    - RPL Blocked = 1 (RPL is blocked)
  - All other switches place ring ports in the unblocked state



When no failures occur on the Ethernet ring, all nodes are in the idle state. During the idle state, the RPL owner places the RPL in a blocking state. Also, the RPL owner sends periodic (every 5 seconds) R-APS messages that signal that no failure is present on the ring (Request/State =no request), that all switches should flush their MAC tables (Do not flush = 0), and that the RPL is currently blocked (RPL Blocked = 1). All other switches flush their MAC tables once (on the first received R-APS message) while unblocking both of their ring ports.

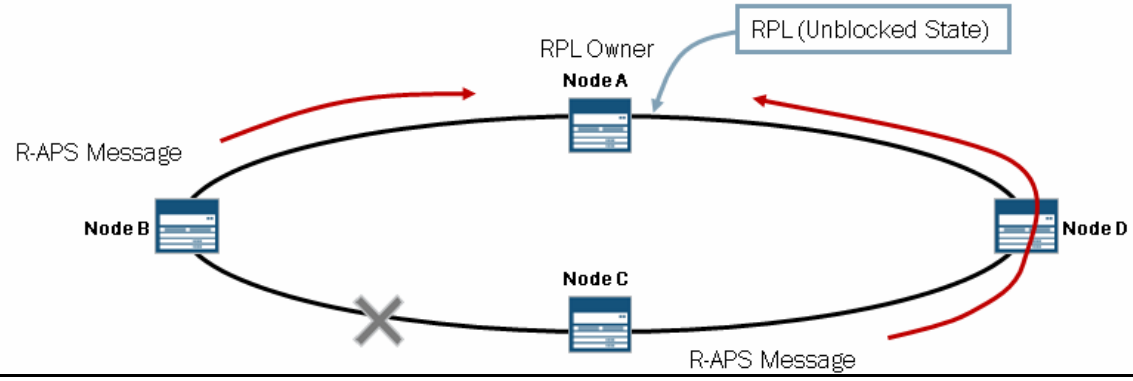
Signal Failure

- Occurs when a failure is detected on an unblocked ring link (CFM failure detection, and so on)
- Node B and Node C:
  - Wait for hold interval to expire (default 0)
  - Switch from idle state to protection state
  - Block failed ports and flush MAC tables
  - Send three R-APS messages in the first 10 ms followed by one every 5 seconds (Request/State = signal fail: Do not flush = 0) until signal failed condition clears



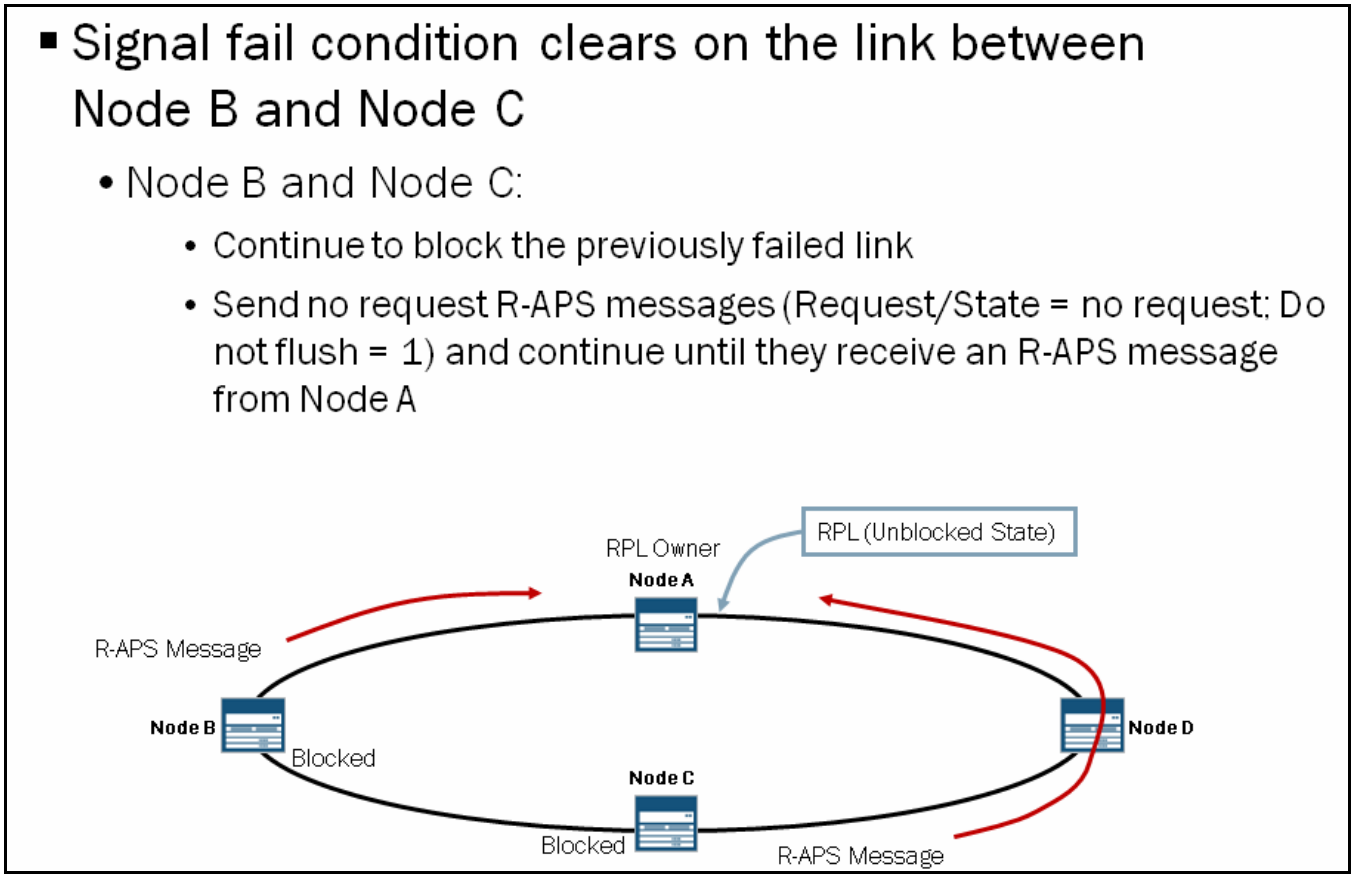
A signal failure occurs when a node detects a failure on a ring port. In the example, Node B and Node C detect a failure on the link between them. The Junos OS does not currently support hold interval. In other words, Node B and Node C react immediately to the failed link. The nodes switch from the idle state to the protection state, block the failed ports, flush their MAC table, and signal to all the other nodes that a signal failure has occurred using R-APS messages. The R-APS messages tell the other nodes that a failure has occurred (Request/State = signal fail) and that the nodes should flush their MAC tables (Do not flush= 0). Node B and Node C continually send R-APS messages every 5 seconds until the signal failure condition clears.

- All switches except Node B and Node C:
  - Switch from the idle state to the protection state
  - Flush MAC tables and stop sending R-APS messages
- RPL owner (Node A):
  - Unblocks RPL
  - Listens for subsequent R-APS messages from Node C and Node D (subsequent signal fail R-APS messages do not re-trigger flushes)



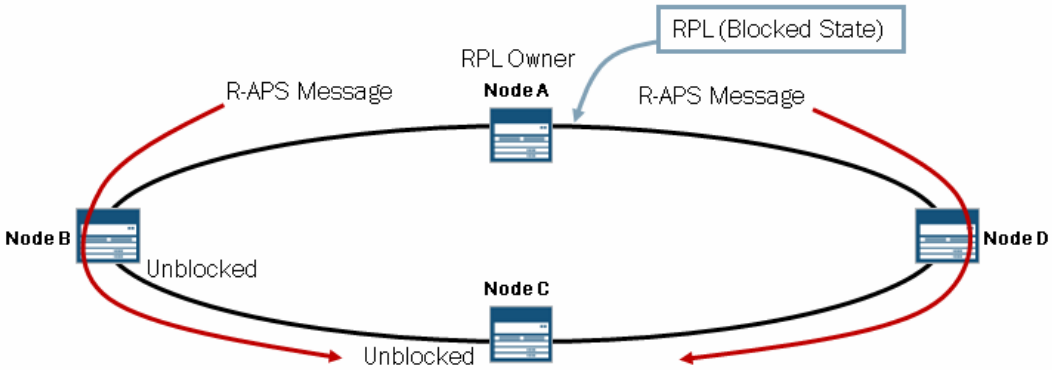
Upon receiving the signal fail R-APS messages from Node B and Node C, all other nodes (including the RPL owner) switch to the protection state, flush their MAC tables, and stop sending R-APS messages. The RPL owner unblocks the RPL and listens for subsequent R-APS message from Node A and Node B.

Restoration of a Failed Link:



When the failure is repaired between Node B and Node C, they begin sending out new R-APS messages. The R-APS messages tell the other nodes that the failure (Request/State = no request) is no longer present and that they should not flush their MAC tables (Do not flush = 1). Node B and Node C keep the previously failed ports in the blocked state (preventing a loop) until they receive R-APS messages from Node A as described in the following graphic.

- Node A (after receiving no request R-APS messages):
  - Waits for the restore timer to expire (default is 5 minutes)
  - Blocks RPL and transmits a R-APS message (Request/State = no request; RPL Blocked = 1, Do not flush = 0)
  - All other switches flush their MAC tables and unblock any blocked ring ports
  - All switches change from the protection state to the idle state



Upon receiving the no request R-APS messages from Node B and Node C, Node A starts a restore timer. The default is 5 minutes. You can configure the restore timer in 1-minute steps between 5 and 12 minutes. Once the restore timer expires, Node A blocks the RPL and transmits RPS messages that signal to the other nodes that no failure is present on the ring (Request/State = no request), that the RPL has been blocked (RPL Blocked = 1), and that the other nodes should flush their MAC tables (Do not flush = 0). Once they receive the R-APS messages from Node A, the other nodes flush their MAC tables and unblock any ring ports that had been blocked. At this point, all switches will be in the idle state.

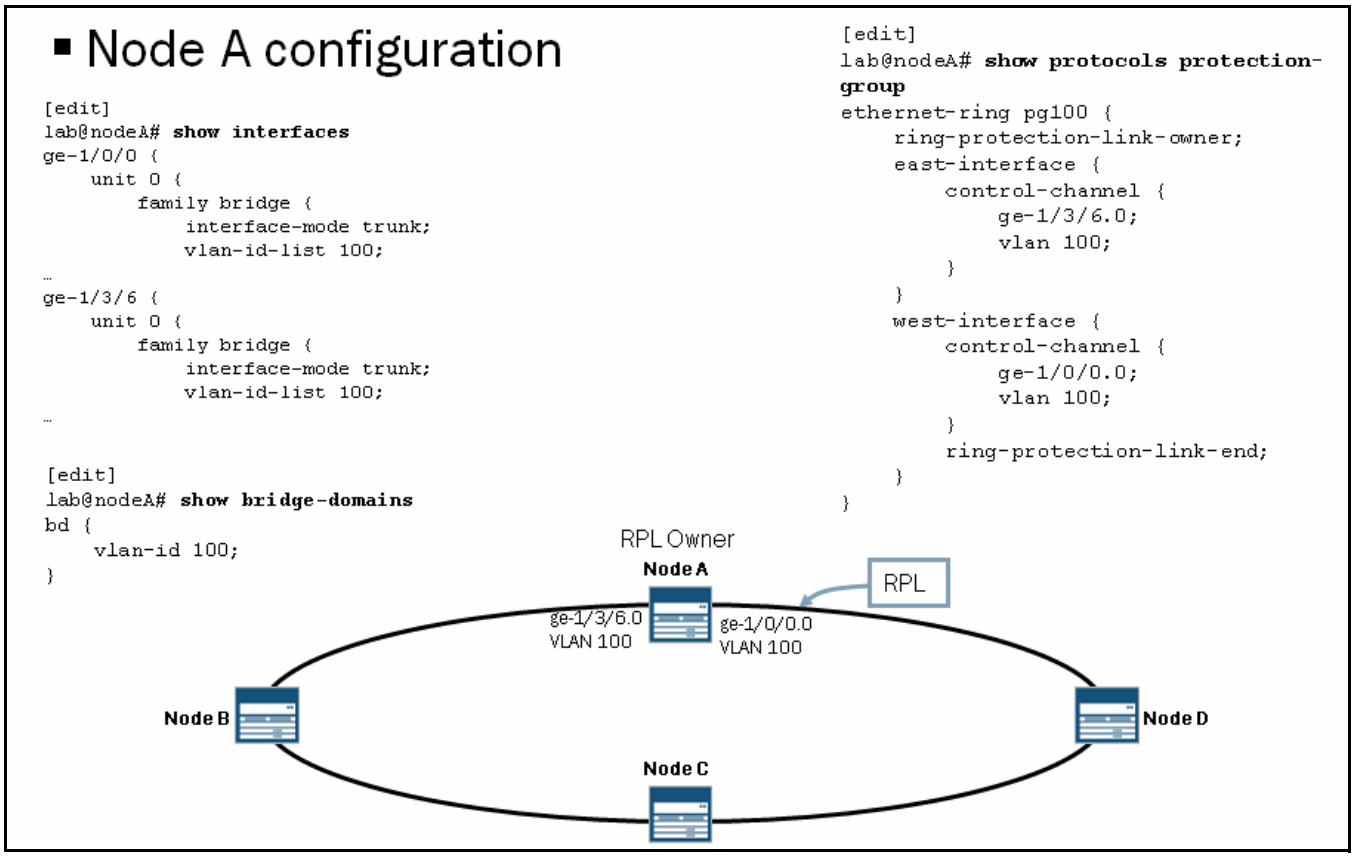
ERP Configuration Options

```
protection-group {
     guard-interval number;
     hold-interval number;
     restore-interval number;
     ethernet-ring ring-name {
              east-interface {
                       ring-protection-link-end;
                       control-channel channel-name {
                                vlan number;
                       }
              }
              guard-interval number;
              hold-interval number;
              node-id mac-address;
              restore-interval number;
              ring-protection-link-owner;
              west-interface {
                       control-channel channel-name {
                                vlan number;
                       }
              }
     }
}
```


The graphic shows all of the options available when configuring ERP. You must configure an `east-interface` and a `west-interface`. You need not configure the two interfaces in any specific order. You can specify global or ring-specific versions of the three intervals (timers) for ERP:

- `guard-interval` (disabled by default): Configurable in 10 ms intervals from 10 ms to 2000 ms. It is used to prevent a node from receiving outdated R-APS messages. Once an R-APS message is received, the guard timer starts. Any R-APS messages that arrive before the expiration of the guard timer drop.
- `hold-interval`: We described this interval on the previous slides.
- `restore-interval`: We described this interval on the previous slides.

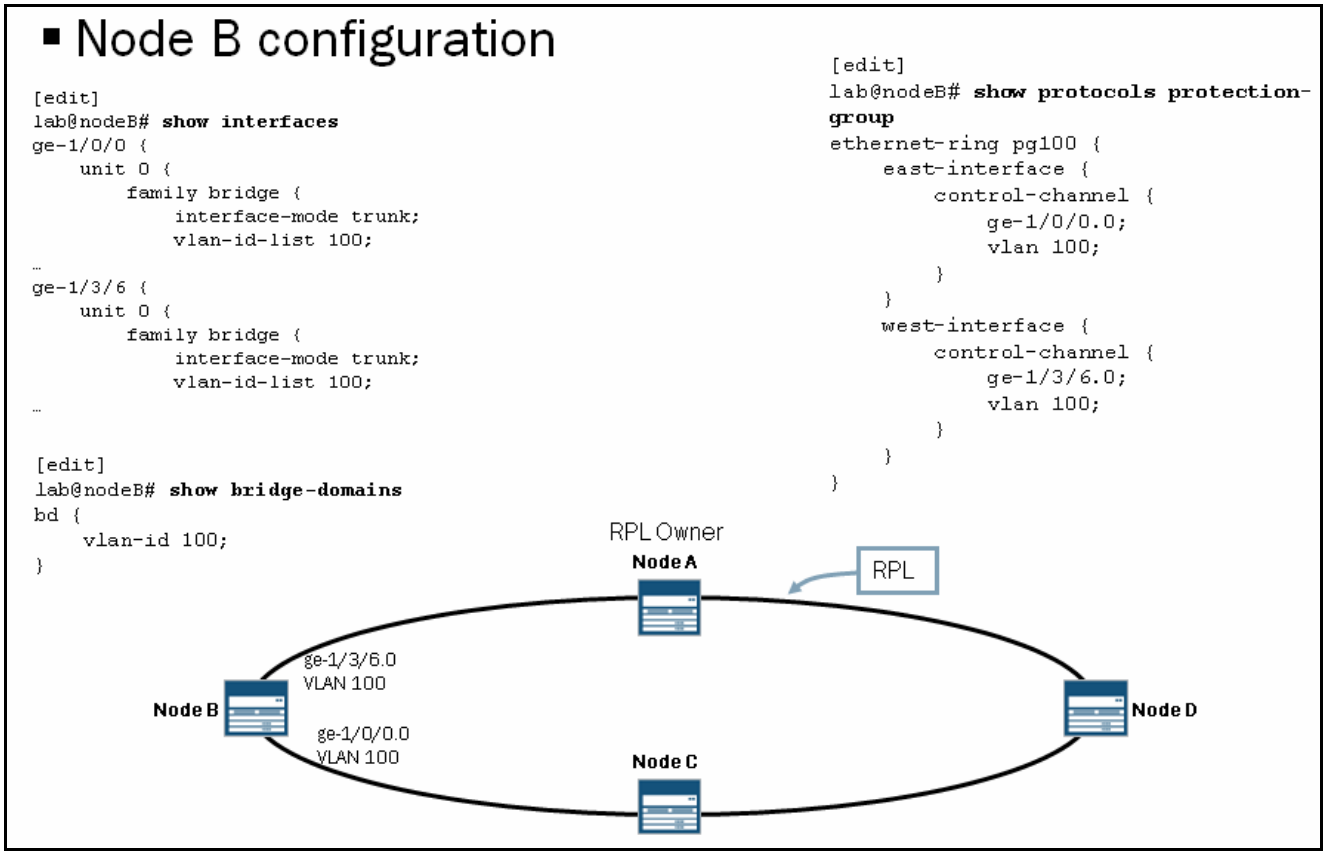
RPL Owner Configuration



The graphic shows a typical configuration for the RPL owner node. First, you must configure the two interfaces that participate in the Ethernet ring for the APS channel (VLAN and bridge domain). In this case, VLAN 100 is used as the communication channel between nodes. Configure ERP under `[edit protocols protection-group]`. The following are a few things to note about the ERP configuration for the RPL owner:

- You must configure the RPL owner node specifically as the `ring-protection-link-owner`;
- The interfaces are interchangeable with regard to selecting them to act as the `west-interface` and `east-interface` as long as you specify one of them as being the `ring-protection-link-end`; and
- For trunk-mode interfaces, you must also specify the VLAN.

Normal Node Configuration



The graphic shows a typical configuration for a normal node.

ERP Status

```
lab@nodeA> show protection-group ethernet-ring ?
Possible completions:
  aps                  Show RAPS PDU information for ethernet ring
  interface            Show interface information for ethernet ring
  node-state           Show RAPS state machine information for ethernet ring
  statistics           Show statistics for ethernet ring
```

The graphic shows all of the possible commands to monitor ERP. We discuss each one on the next few sections.

R-APS Information

```
lab@nodeA> show protection-group ethernet-ring aps
Ethernet Ring Name  Request/state  No Flush  Ring Protection Link Blocked  Originator
Remote Node ID
pg100               NR             No        Yes                           Yes

lab@nodeA> show protection-group ethernet-ring aps detail
Ethernet-Ring name              : pg100
Request/State                   : NR
No Flush Flag                   : No
Ring Protection Link blocked    : Yes
Originator                      : Yes
```

The command on the graphic shows the details of the R-APS messages to which the local node is currently listening or which it is forwarding. Based on the output, you can tell that the local node (Node A) is the RPL owner because the R-APS message originates from it and it is advertising that the RPL is currently blocked.

Interface Status

```
lab@nodeA> show protection-group ethernet-ring interface

Ethernet ring port parameters for protection group pg100

Interface    Control Channel  Forward State  Ring Protection Link End  Signal Failure  Admin State
ge-1/0/0     ge-1/0/0.0       discarding     Yes                       Clear           IFF ready
ge-1/3/6     ge-1/3/6.0       forwarding     No                        Clear           IFF ready

lab@nodeA> show protection-group ethernet-ring interface detail

Ethernet ring port parameters for protection group pg100

Interface name                 : ge-1/0/0
Control channel name           : ge-1/0/0.0
Ring Protection Link End       : Yes
Signal Failure                 : Clear
Forward State                  : discarding
Interface Admin State          : IFF ready

Interface name                 : ge-1/3/6
Control channel name           : ge-1/3/6.0
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready
```

The command in the graphic shows the state of the local node interfaces in relation to ERP. Note that the `Admin State` shows that it is `IFF ready`. This state means that the Ethernet flow forwarding function (the control channel) is available to forward R-APS traffic.

Local Node Details

```
lab@nodeA> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         Ring Protection Link Owner  Restore Timer  Guard Timer  Operation state
pg100            idle         NR-RB         Yes                         disabled        disabled    operational

lab@nodeA> show protection-group ethernet-ring node-state detail
Ethernet-Ring name             : pg100
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : Yes
Restore Timer                  : disabled
Guard Timer                    : disabled
Operation state                : operational
```

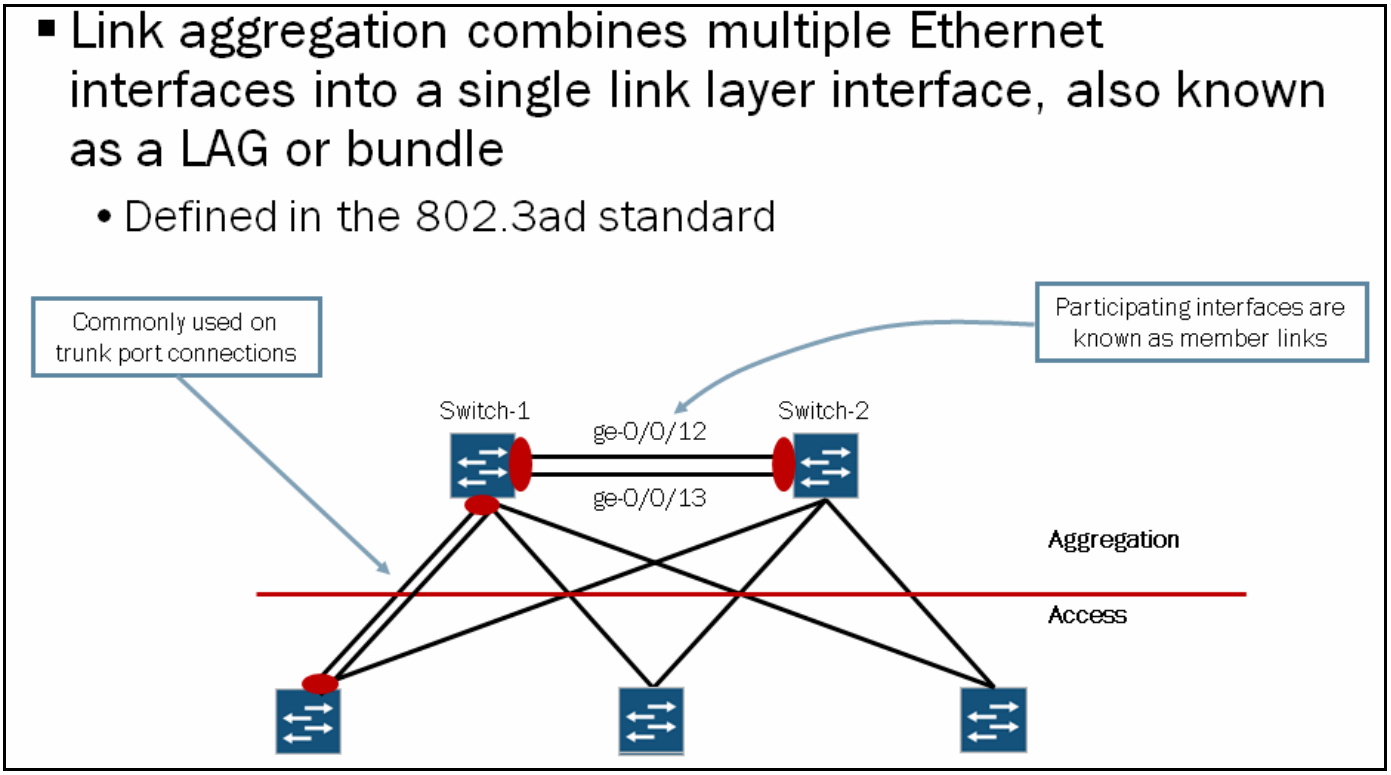
The command in the graphic shows the APS `State` of the local node, as well as some of the locally configured timer values.

ERP Statistics

```
lab@nodeA> show protection-group ethernet-ring statistics

Ethernet Ring statistics for PG pg100
RAPS event sent                  : 2
RAPS event received              : 2
Local SF happened:               : 0
Remote SF happened:              : 1
NR event happened:               : 1
NR-RB event happened:            : 2
```

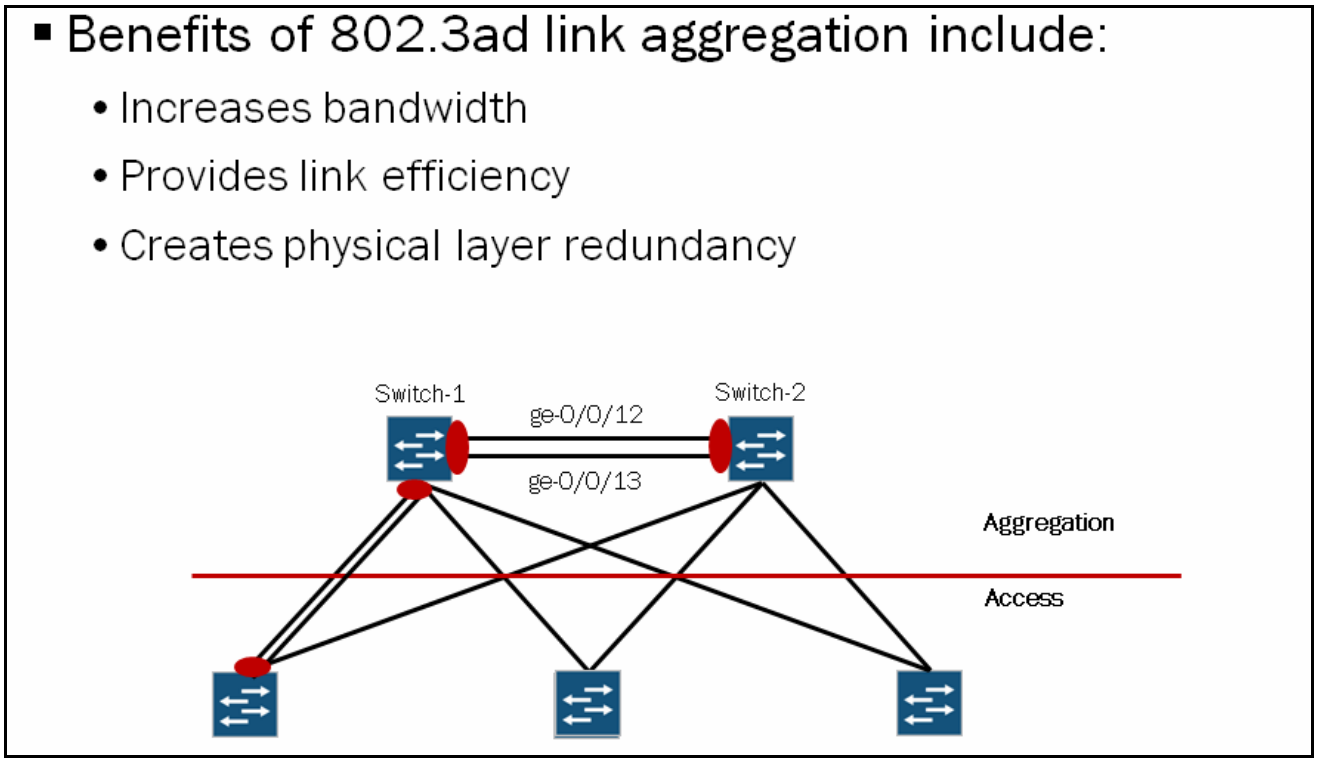
The command in the graphic shows the quantities of specific events that have occurred. You can reset these values to 0 by issuing the **`clear protection-group ethernet-ring statistics group-name`** ***`name`*** command.

Link Aggregation Groups



The Institute of Electrical and Electronics Engineers (IEEE) 802.3ad link aggregation specification enables multiple Ethernet interfaces to be grouped together and form a single link layer interface, also known as a link aggregation group (LAG) or bundle. The physical links participating in a LAG are known as member links. As illustrated in the graphic, LAGs are commonly used to aggregate trunk links between an access and aggregation switches.

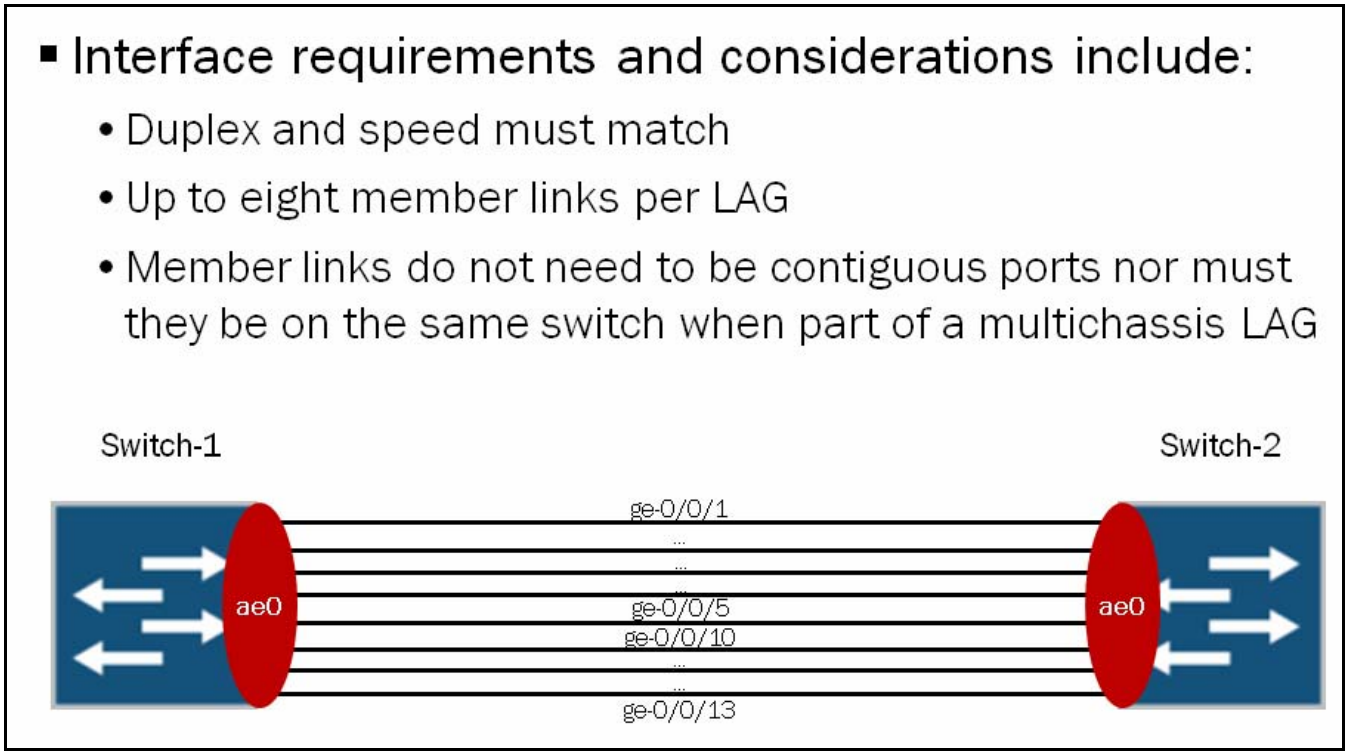
Benefits of Link Aggregation



You implement link aggregation using point-to-point connections between two devices. Based on the number of member links participating in the LAG, the bandwidth increases proportionately. The participating switches balance traffic across the member

links within an aggregated Ethernet bundle and effectively increase the uplink bandwidth. Another advantage of link aggregation is increased availability because the LAG is composed of multiple member links. If one member link fails, the LAG continues to carry traffic over the remaining links.

Link Requirements and Considerations

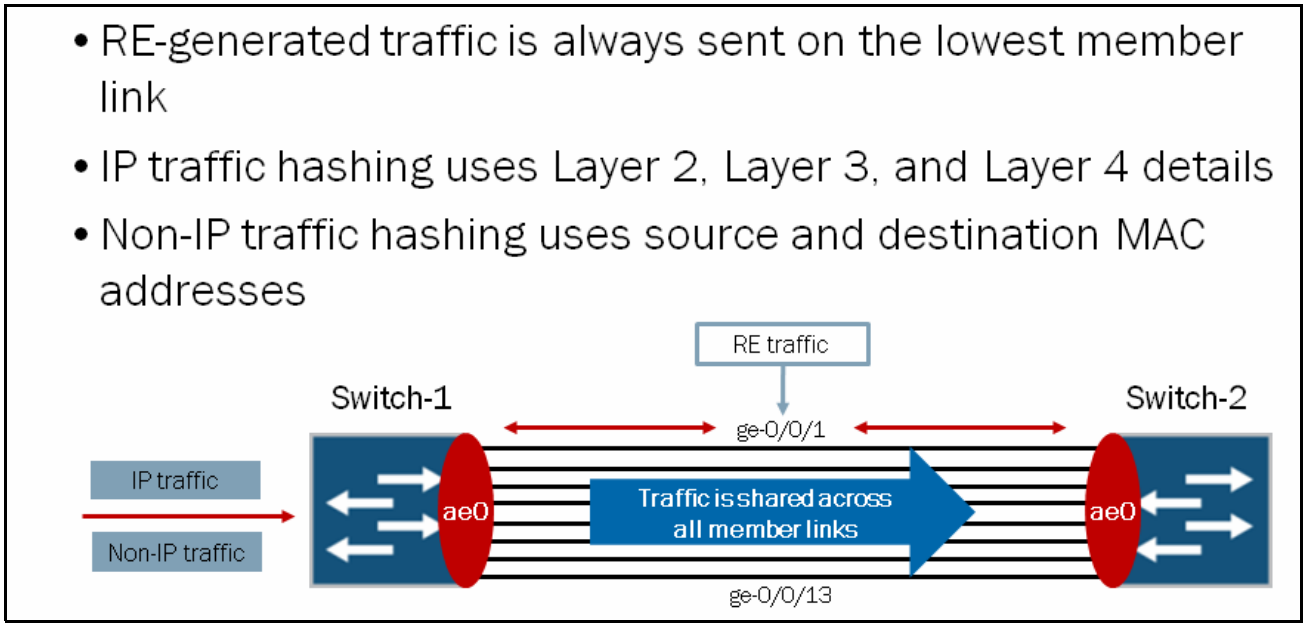


A number of hardware requirements and considerations exist when working with link aggregation. The following list highlights these details:

- Duplex and speed settings must match on both participating devices.
- Up to eight member links can belong to a single LAG.
- Member links are not required to be contiguous ports and can reside on different members within a multichassis LAG (MC-LAG).

Note that the number of member links allowed for a given LAG is platform dependent. Refer to the documentation for your specific product for support information.

Processing and Forwarding Considerations

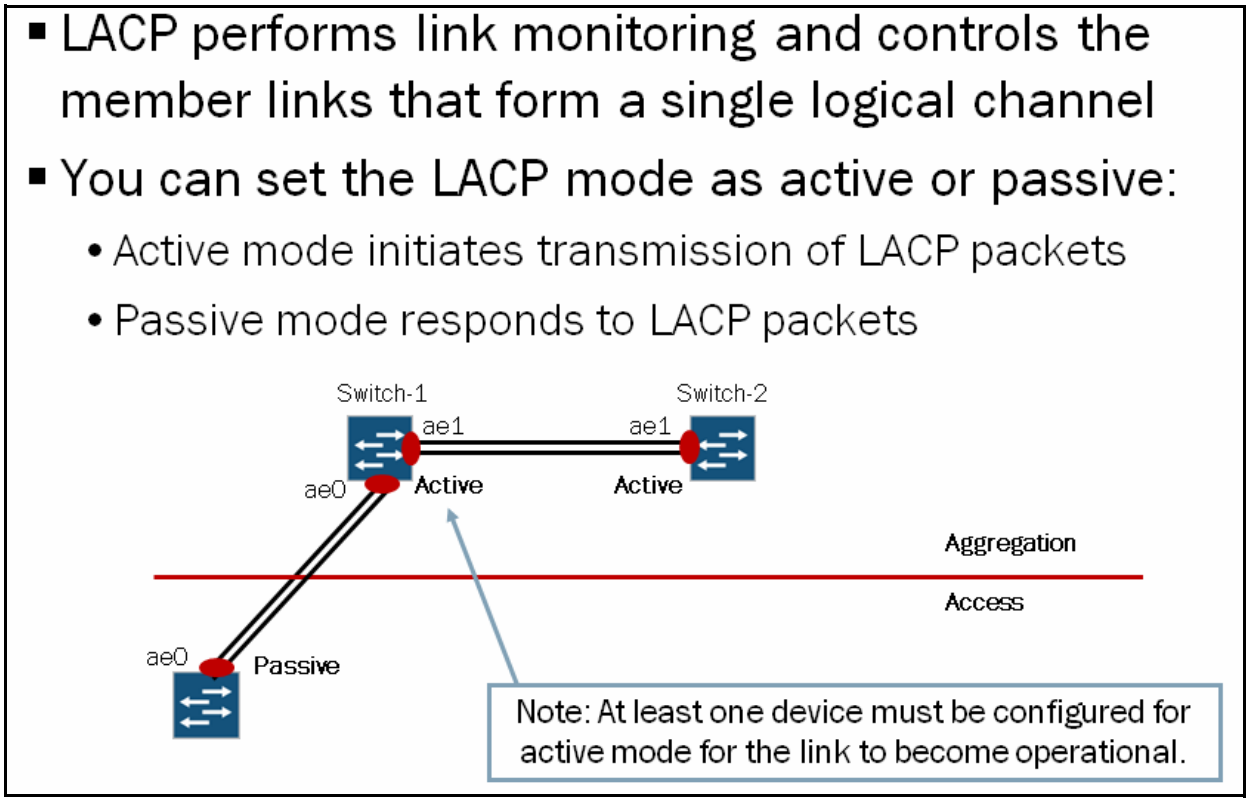


Some traffic processing and forwarding considerations exist when working with link aggregation. The following list highlights these details:

- All RE generated packets that traverse the LAG, such as protocol control traffic, will use the lowest member link.
- The load-balancing hash algorithm for IP traffic uses criteria at Layer 2, Layer 3, and Layer 4. No configuration is necessary to enable load balancing.

The load-balancing hash algorithm for non-IP traffic uses source and destination MAC addresses.

Link Aggregation Control Protocol



You can enable Link Aggregation Control Protocol (LACP) for aggregated Ethernet interfaces. LACP is one method of bundling several physical interfaces to form one logical interface. You can configure both VLAN-tagged and untagged aggregated Ethernet with or without LACP enabled.

LACP exchanges are made between actors and partners. An actor is the local interface in an LACP exchange. A partner is the remote interface in an LACP exchange. LACP is defined in IEEE 802.3ad, Aggregation of Multiple Link Segments and was designed to achieve the following:

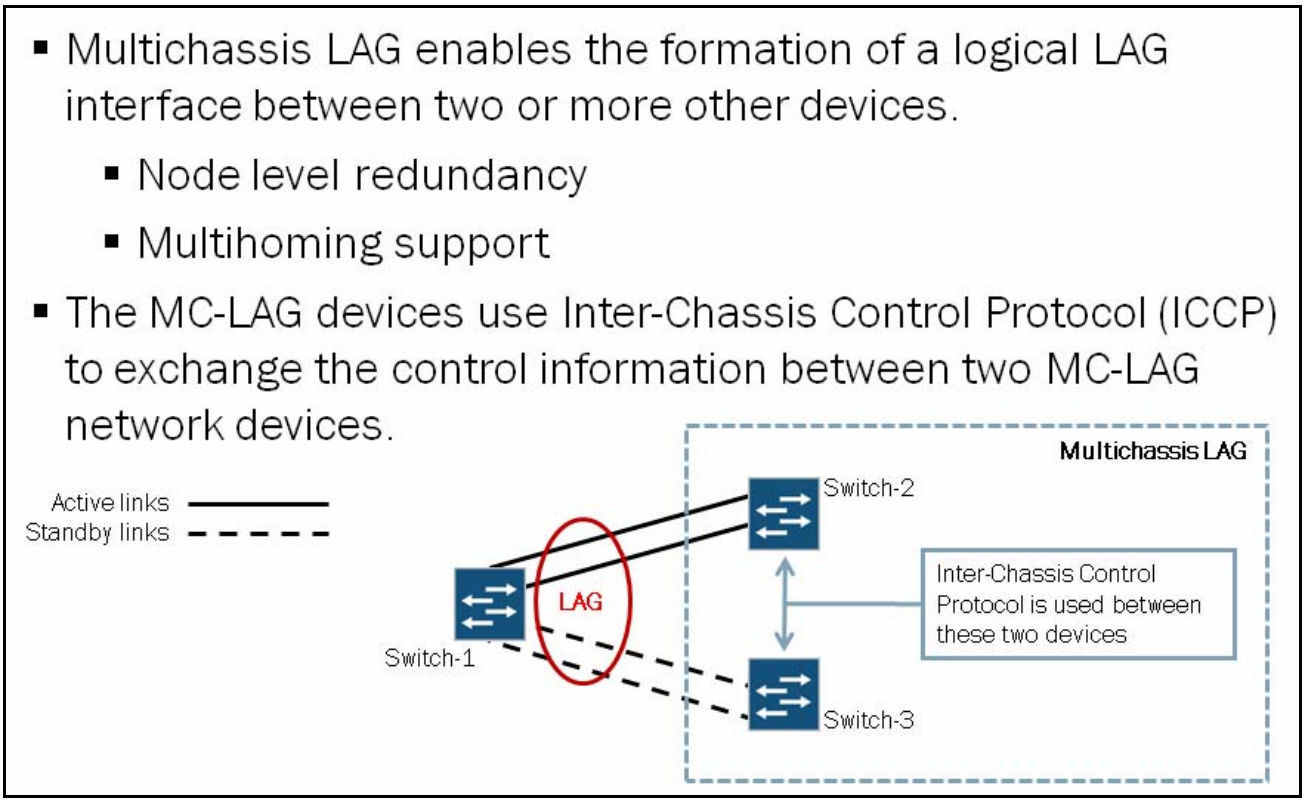
- Automatic addition and deletion of individual links to the aggregate bundle without user intervention
- Link monitoring to check whether both ends of the bundle are connected to the correct group

Note that the Junos OS implementation of LACP provides link monitoring but not automatic addition and deletion of links.

The LACP mode can be active or passive. If the actor and partner are both in passive mode, they do not exchange LACP packets, which results in the aggregated Ethernet links not coming up. If either the actor or partner is active, they do exchange LACP packets. By default, when LACP is configured its mode defaults to the passive mode on aggregated Ethernet interfaces. To initiate transmission of LACP packets and response to LACP packets, you must enable LACP active mode.

Note that LACP exchanges protocol data units (PDUs) across all member links to ensure each physical interfaces is configured and functioning properly.

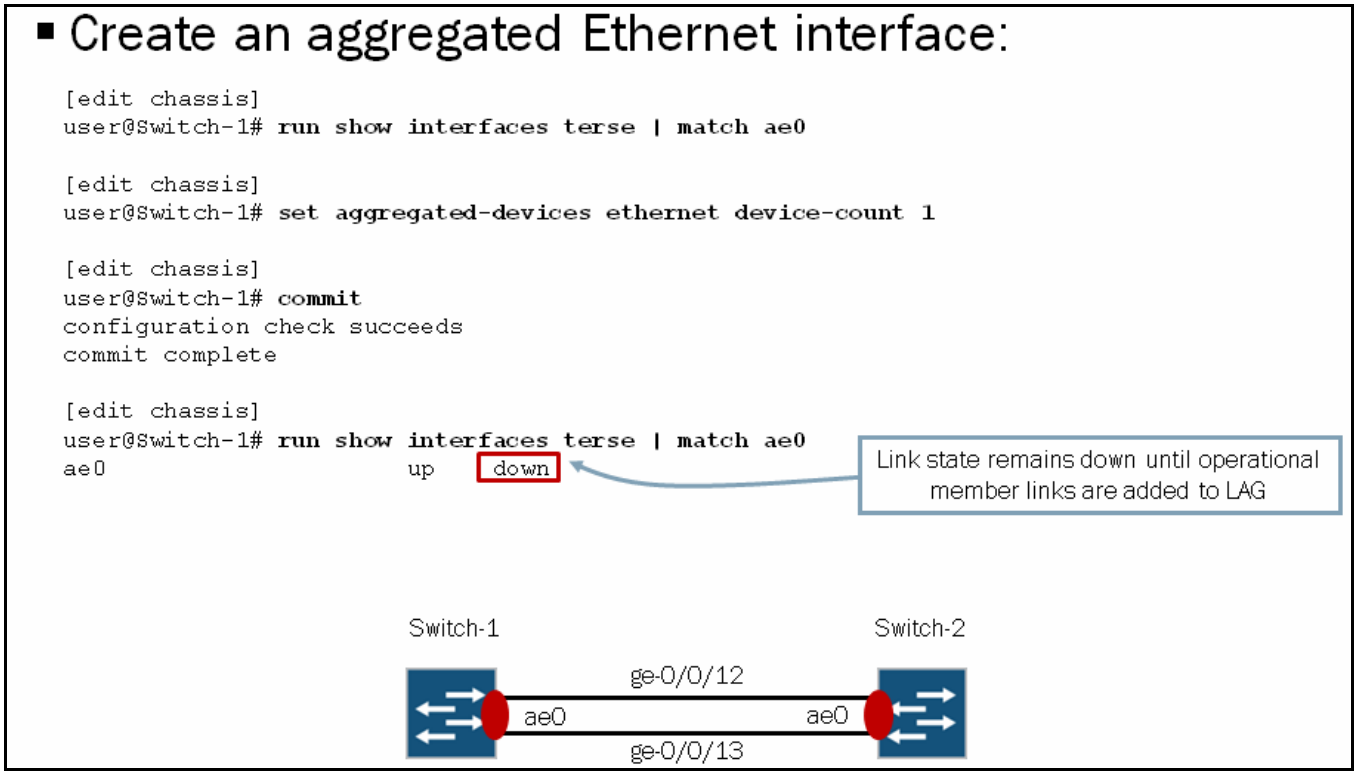
Multichassis Link Aggregation



On MX Series routers, MC-LAG enables a device to form a logical LAG interface with two or more other devices. MC-LAG provides additional benefits over traditional LAG in terms of node level redundancy, multihoming support, and loop-free Layer 2 network without running Spanning Tree Protocol (STP).

The MC-LAG devices use Inter-Chassis Control Protocol (ICCP) to exchange the control information between two MC-LAG network devices. In the diagram on the slide, Switch-1 is an MC-LAG client device that has four physical links in a link LAG. This client device does not need to be aware of the MC-LAG. On the other side of MC-LAG are two MC-LAG network devices, Switch-2 and Switch-3. These two network devices coordinate with each other to ensure that data traffic is forwarded properly.

Implementing LAGs



By default, no aggregated interfaces exist. To create an aggregated interface, simply add an aggregated device under the `[edit chassis]` hierarchy, as shown in the example on the slide. In this example, the aggregated Ethernet interface (ae0) interface has been created. Note that the **`device-count`** statement determines the number of aggregated Ethernet interfaces that the system creates. The number of supported aggregated Ethernet interface varies between platforms. For support information, refer to your product-specific documentation.

Aggregated interfaces are always created in numerical order starting with ae0. For example, a device count of two would create ae0 and ae1, as shown in the following output:

```
[edit]
user@Switch-1# show chassis
aggregated-devices {
    ethernet {
        device-count 2;
    }
}

[edit]
user@Switch-1# run show interfaces terse | match ae
ae0                     up    down
ae1                     up    down
```

- Configure the aggregated Ethernet interface and associate desired member links with the LAG:

```
[edit interfaces]
user@Switch-1# set ae0 unit 0 family bridge

[edit interfaces]
user@Switch-1# set ae0 aggregated-ether-options lacp active

[edit interfaces]
user@Switch-1# set ge-0/0/12 gigether-options 802.3ad ae0

[edit interfaces]
user@Switch-1# set ge-0/0/13 gigether-options 802.3ad ae0

[edit interfaces]
user@Switch-1# commit
configuration check succeeds
commit complete

[edit interfaces]
user@Switch-1# run show interfaces terse | match ae0
ge-1/1/2.0              up    up   aenet    --> ae0.0
ge-1/1/3.0              up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   bridge
```

This graphic illustrates the remainder of the configuration required to implement a LAG for Layer 2 operations. On this slide, you see that the (ae0 in this case) must be configured for Layer 2 operations. You also see that the physical links participating in this LAG (also known as member links) are configured and associated with the ae0 interface. Note that these member links must be operational for the aggregated Ethernet interface to become operational.

Once the illustrated configuration is activated, the aggregated Ethernet interface is up and can begin to process and forward user traffic. Note that in this example, we used LACP. LACP must be enabled on the remote device (Switch-2) for the aggregated Ethernet interface to come up and function properly. Given Switch-1's configuration, Switch-2 can be configured for LACP active or passive mode.

By default, the actor and partner send LACP packets every second. You can configure the interval at which the interfaces send LACP packets by including the **`periodic`** option at the `[edit interfaces` *`interface`* `aggregated-ether-options lacp]` hierarchy level. The interval can be **`fast`** (every second) or **`slow`** (every 30 seconds). You can configure different periodic rates on active and passive interfaces. When you configure the active and passive interfaces at different rates, the transmitter honors the receiver's rate.

```
[edit interfaces ae0 aggregated-ether-options lacp]
user@Switch-1# set periodic ?
Possible completions:
  fast                 Transmit packets every second
  slow                 Transmit packets every 30 seconds
```

Monitoring LAGs

- Use the `show interfaces` output to determine state information for aggregated interfaces:

```
user@Switch-1> show interfaces terse | match ae0
ge-1/1/2.0              up    up   aenet    --> ae0.0
ge-1/1/3.0              up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   bridge
```

- Use the `show lacp statistics interfaces` output to determine if LACP packets are passing

```
user@Switch-1> show lacp statistics interfaces
Aggregated interface: ae0
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      ge-1/1/2               3982        3983            0            0
      ge-1/1/3               3982        3983            0            0
```

This graphic illustrates one method of monitoring LAGs. Using the output from the **`show interfaces`** commands, you can determine state information along with other key information such as error conditions and statistics. The highlighted output shows state information for the aggregated Ethernet and member link interfaces. You can also see LACP statistics for the ae0 interface using the **`extensive`** option. Note that when LACP is used, you can find similar state and statistical information using the **`show lacp interfaces`** and **`show lacp statistics`** commands.

If a problem related to LACP occurs, you can configure traceoptions for LACP under the `[edit protocols lacp]` hierarchy:

```
[edit]
user@Switch-1# set protocols lacp traceoptions flag ?
Possible completions:
  all                  All events and packets
  configuration        Configuration events
  mc-ae                Multi-chassis AE messages
  packet               LACP packets
  ppm                  LACP PPM messages
  process              Process events
  protocol             Protocol events
  routing-socket       Routing socket events
  startup              Process startup events
```

Review Questions

1. When copper links exist between ring nodes, which other protocol should you use in conjunction with ERP?
2. What can cause the RPL owner to stop sending periodic R-APS messages?
3. Once a signal fail condition clears, how long does the RPL owner wait until it begins sending periodic R-APS messages? What is the name of the timer?

Answers

1.

You should use CFM with ERP for faster protection times.

2.

An RPL owner stops sending its own R-APS messages when it receives an R-APS message for another node that specifies signal failure.

3.

The RPL owner waits until the `Restore Timer` has expired. The default is 5 minutes.

JNCIS-SP Study Guide—Part 2

Appendix 8: Deprecated Syntaxes

This Appendix Discusses:

- Differences between “old” and “new” configuration syntaxes.

Define a Bridge Domain and Assign an Interface

- Define the bridge domains (broadcast domains), VLANs, and access ports to be used for switching

New syntax

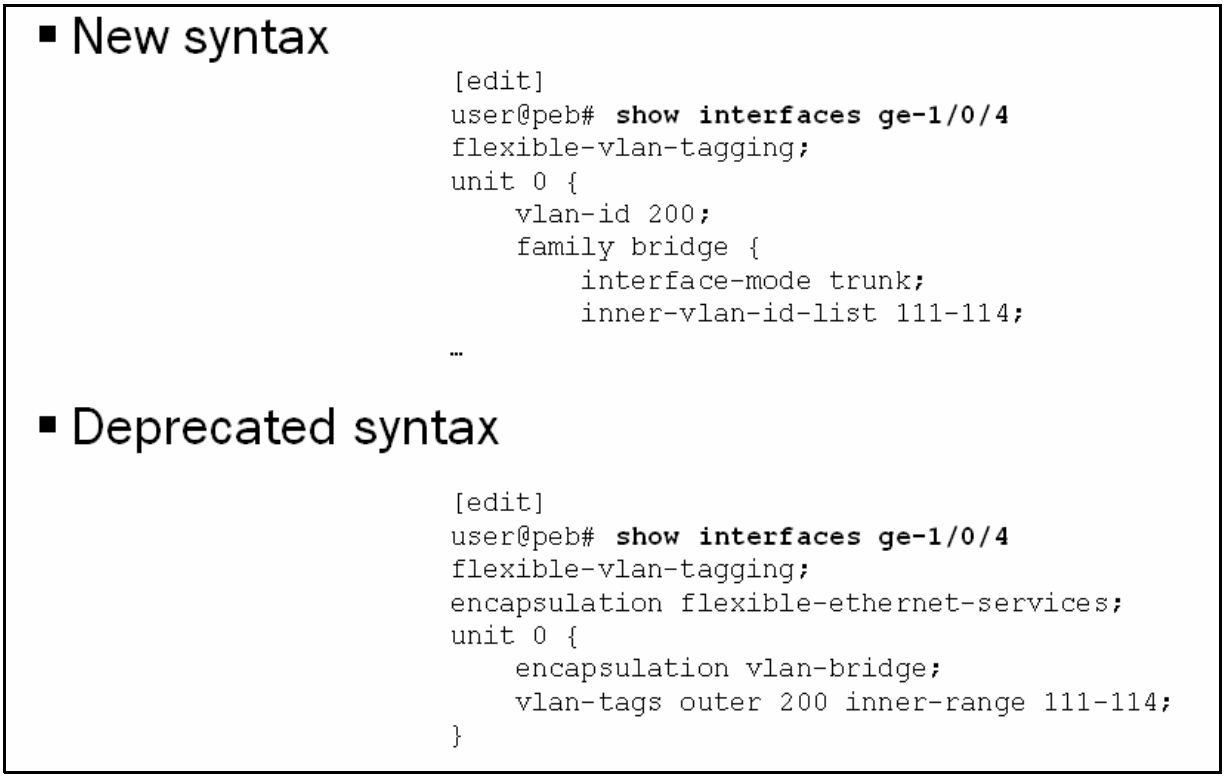
```
[edit]
user@switch# show
…
interfaces {
    ge-1/0/0 {
        unit 0 {
            family bridge {
                interface-mode access;
                vlan-id 100;
            }
        }
    }
}
…
bridge-domains {
    vlan_100 {
        vlan-id 100;
    }
}
```

Old syntax

```
[edit]
user@switch# show
…
interfaces {
    ge-1/0/0 {
        encapsulation ethernet-bridge;
        unit 0;
    }
}
…
bridge-domains {
        vlan_100 {
                vlan-id 100;
                interface ge-1/0/0.0;
        }
}
```

The graphic shows both methods that can be used to accomplish adding the ge-1/0/0 interface as an access port to VLAN 100.

Creating Dual-Stacked VLAN Subinterfaces and Bridge Domains



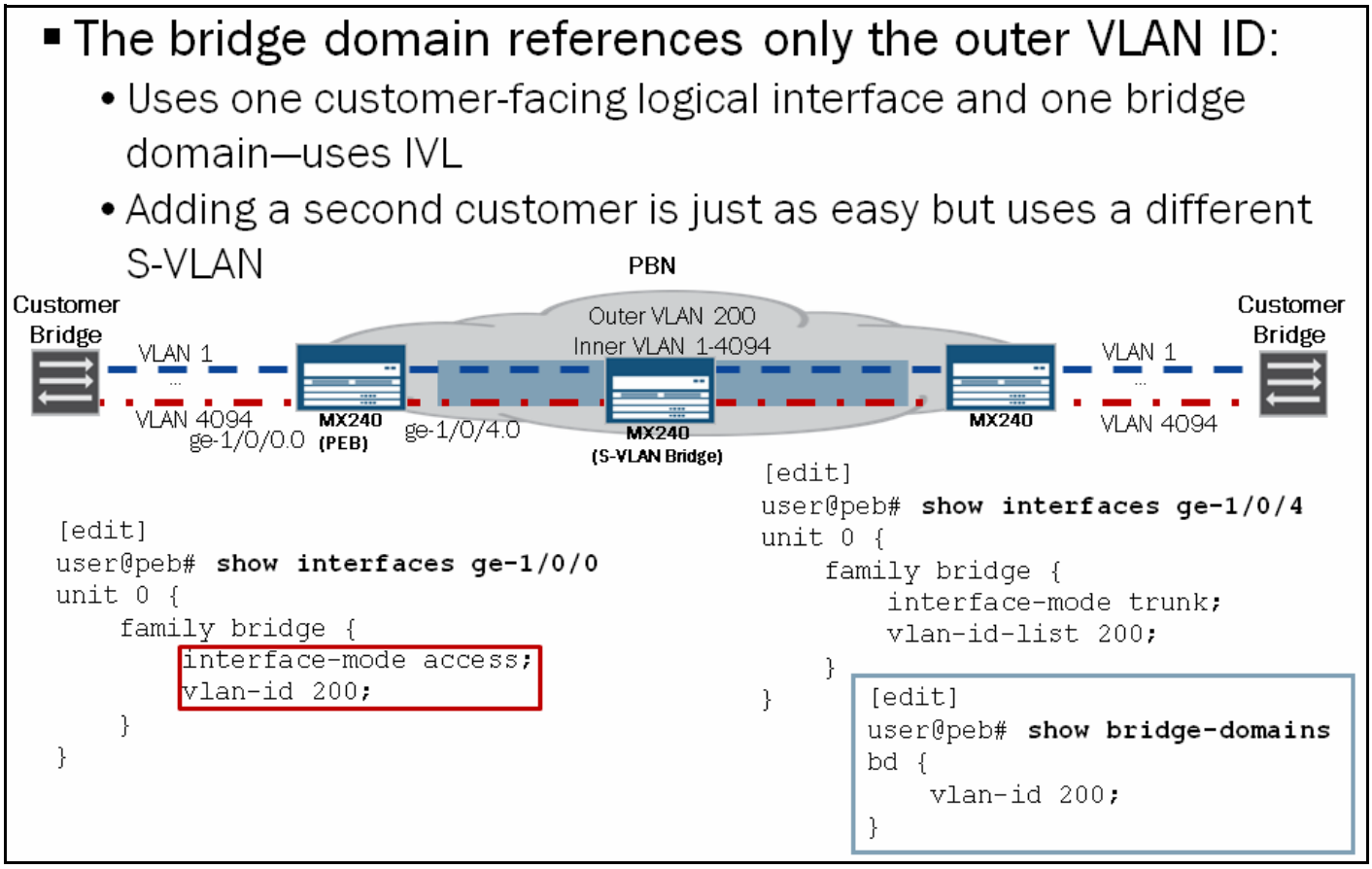
This graphic shows both methods that can be used to create dual-stacked virtual LAN (VLAN) subinterfaces. When using the original style of configuring a dual-stacked interface, you still must apply the interface to a bridge domain. The way in which you specify the VLAN IDs for the bridge domain determines the bridge domain's mode of operation. The following list briefly explains the different ways of configuring a bridge domain:

- *Default*: You do not specify a VLAN ID for the bridge domain. The bridge domain is a single learning domain. You configure an input and output VLAN map to explicitly configure push, pop, swap, and other VLAN operations.
- *None*: You specify `vlan-id none` for the bridge domain. The bridge domain is a single learning domain. In this case, all inbound frames have all VLAN IDs popped. All outbound frames take on the VLAN settings of the outbound interfaces.
- *Single*: You specify `vlan-id` *`number`* for the bridge domain. The bridge domain is a single learning domain. In this case, all inbound frames have all service VLAN (S-VLAN) IDs popped. All inbound customer VLAN (C-VLAN) IDs are normalized (translated) to the VLAN ID of the bridge domain. All outbound frames take on the VLAN settings of the outbound interface.
- *Double*: You specify `vlan-tags outer` *`number`* `inner` *`number`* for the bridge domain. The bridge domain is a single learning domain. All incoming frames have their VLANs normalized (translated) to the outer and inner VLAN ID that is specified for the bridge domain. All outbound frames take on the VLAN settings of the outbound interface.
- *All*: You specify `vlan-id all` for the bridge domain. The bridge domain has multiple learning domains. One learning domain exists for each C-VLAN configured on interfaces that are associated with the bridge domain. This type of configuration always results in independent VLAN learning mode (IVL). Inbound frames retain their VLAN tags. All outbound frames take on the VLAN settings of the outbound interface.

Most of these options listed cause a bridge domain to have a single learning domain. If the interfaces assigned to a bridge domain are configured for a unique C-VLAN ID, then the learning mode for the bridge domain will be IVL. If the interfaces assigned to a bridge domain are configured for multiple C-VLANs, then the learning mode for the bridge domain will be shared VLAN learning mode (SVL).

When using the new style of configuration, IVL is the usual mode of operation. SVL can occur only with the new style of configuration when mixing both old style and new style configurations in a bridge domain.

Tunnel All C-VLANs: New Style



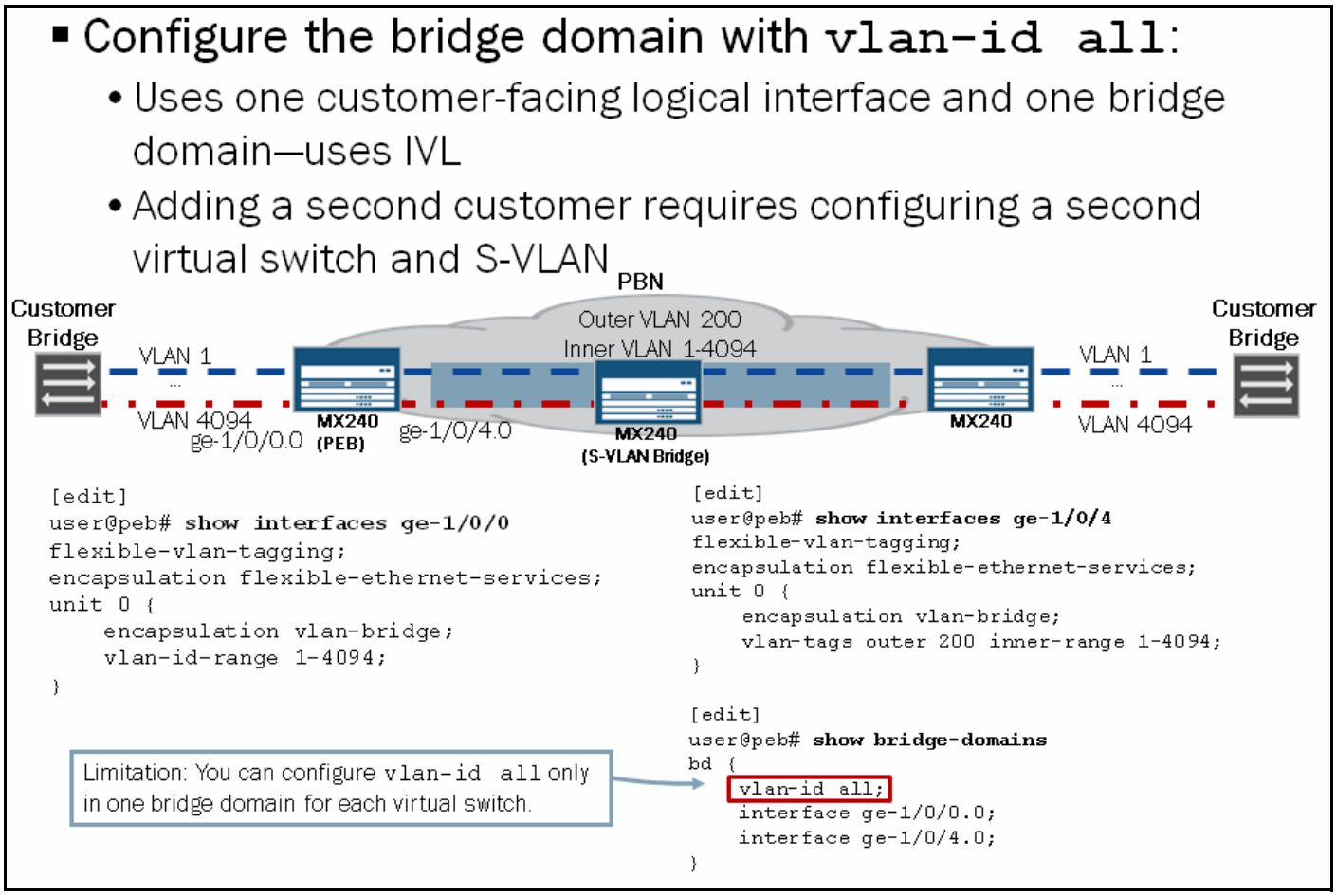
The method shown on the slide is the easiest and most elegant method of tunneling all customer C-VLANs across the core of a provider bridged network (PBN). The interface and bridge domain configuration require only that you specify the outer S-VLAN ID. To allow single-tagged frames to enter the customer-facing interface, you must specify the `interface-mode` access statement.

You will see on the next few slides that each configuration method results in some combination of one of the following:

1. A bridge domain mode (IVL or SVL).
2. Customer-facing logical interface usage.
3. Bridge domain usage.
4. Virtual switch usage.

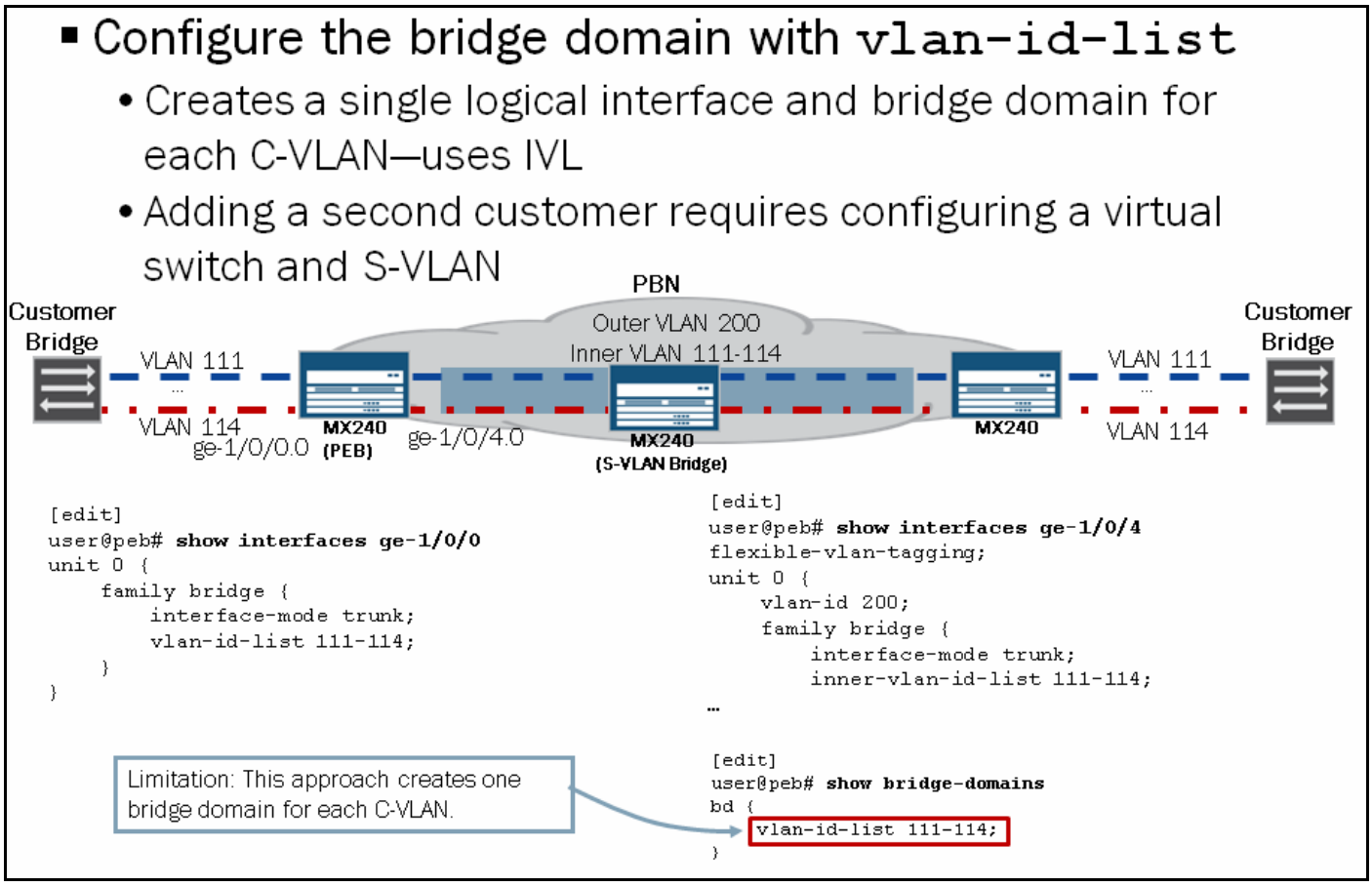
The solution on this slide is elegant because to support each customer, it requires the use of only one logical interface and one bridge domain. In addition, you can place each customer in the same virtual switch.

Tunnel All C-VLANs: Old Style



The graphic shows another method of tunneling all customer C-VLANs across the core of a PBN using the original style of configuration. For each customer, this solution requires the use of one logical interface and one bridge domain. Furthermore, you must place each customer in the same virtual switch. You must do it this way because only one bridge domain is allowed in a virtual switch that uses the `vlan-id all` statement.

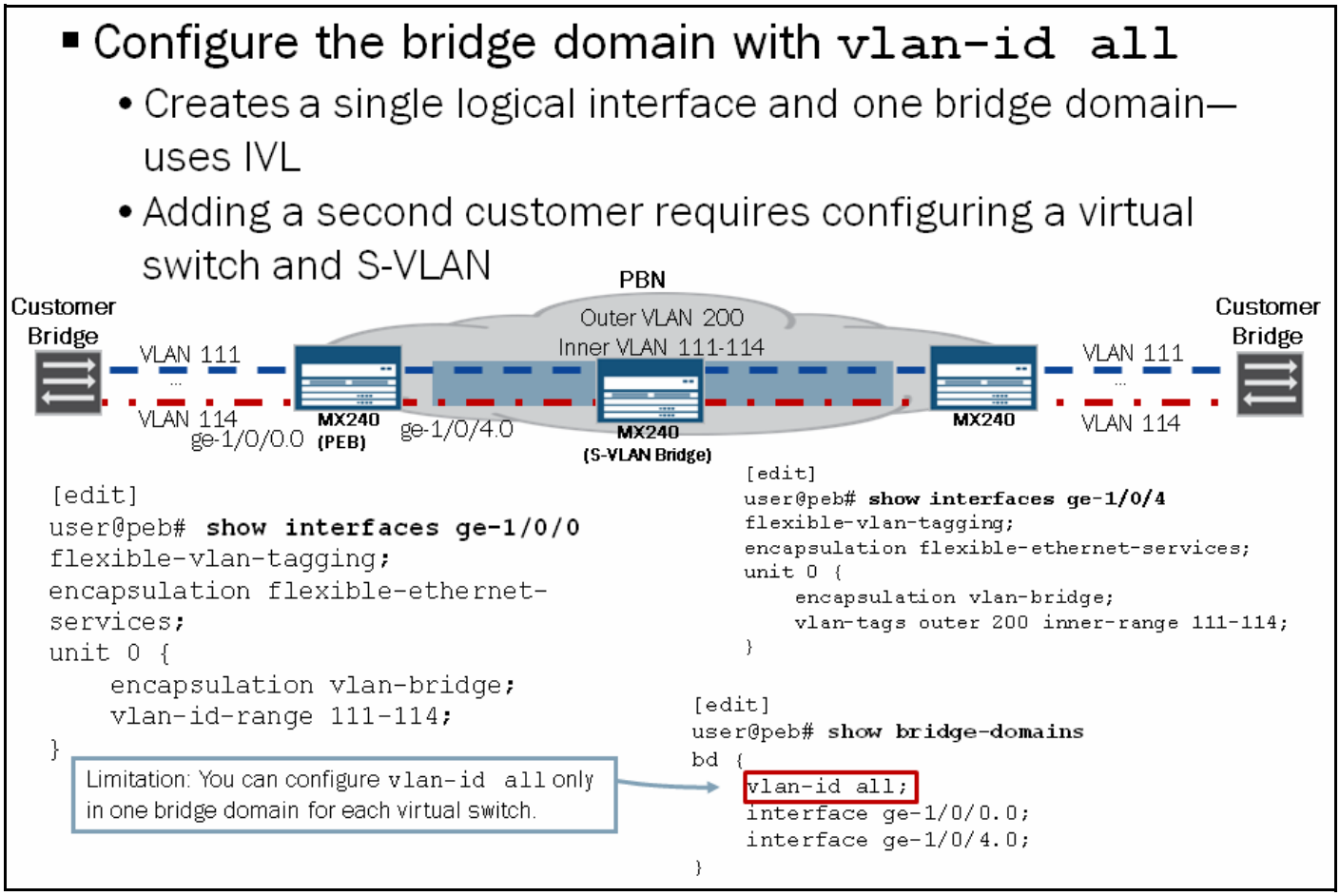
Range of C-VLANs: New Style



When using the original style of configuration, allowing only certain C-VLANs to be tunneled across the core might be necessary. However, few solutions allow this tactic. With the new style of configuration, the bridge domain references the C-VLAN IDs to be tunneled.

Note that because of this referencing—that is, in the case of overlapping C-VLAN space—you must add each customer to its own virtual switch.

Range of C-VLANs: Old Style



The graphic shows the deprecated method of tunneling a range of C-VLANs across the core.