JNCIS-ENT Switching Study Guide

Worldwide Education Services

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

JNCIS-ENT Switching Study Guide.

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 12.1R1.9. 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.

Contents

Overview

Welcome to the *JNCIS-ENT Switching Study Guide*. The purpose of this guide is to help you prepare for your JN0-343 exam and achieve your JNCIS-ENT Switching credential. The contents of this document are based on the *Junos Enterprise Switching* course. This study guide is designed to provide students with introductory switching knowledge and configuration examples. This study guide includes an overview of switching concepts and operations, virtual LANs (VLANs), spanning tree protocol, port and device security features, and high-availability features. This study guide is based on the Junos operating system Release 12.1R1.9.

Agenda

Document Conventions

CLI and GUI Text

Frequently throughout this 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 Student Guide.
`Courier New`	Console text: • Screen captures • Noncommand-related syntax	`commit complete` `Exiting configuration mode`
	GUI text elements: • Menu names • Text field entry	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 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 or text where the variable's value as shown in the lab guide might differ from the value the user must input according to the lab topology.	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-ENT Switching Study Guide* was developed and tested using software Release 12.1R1.9. 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).

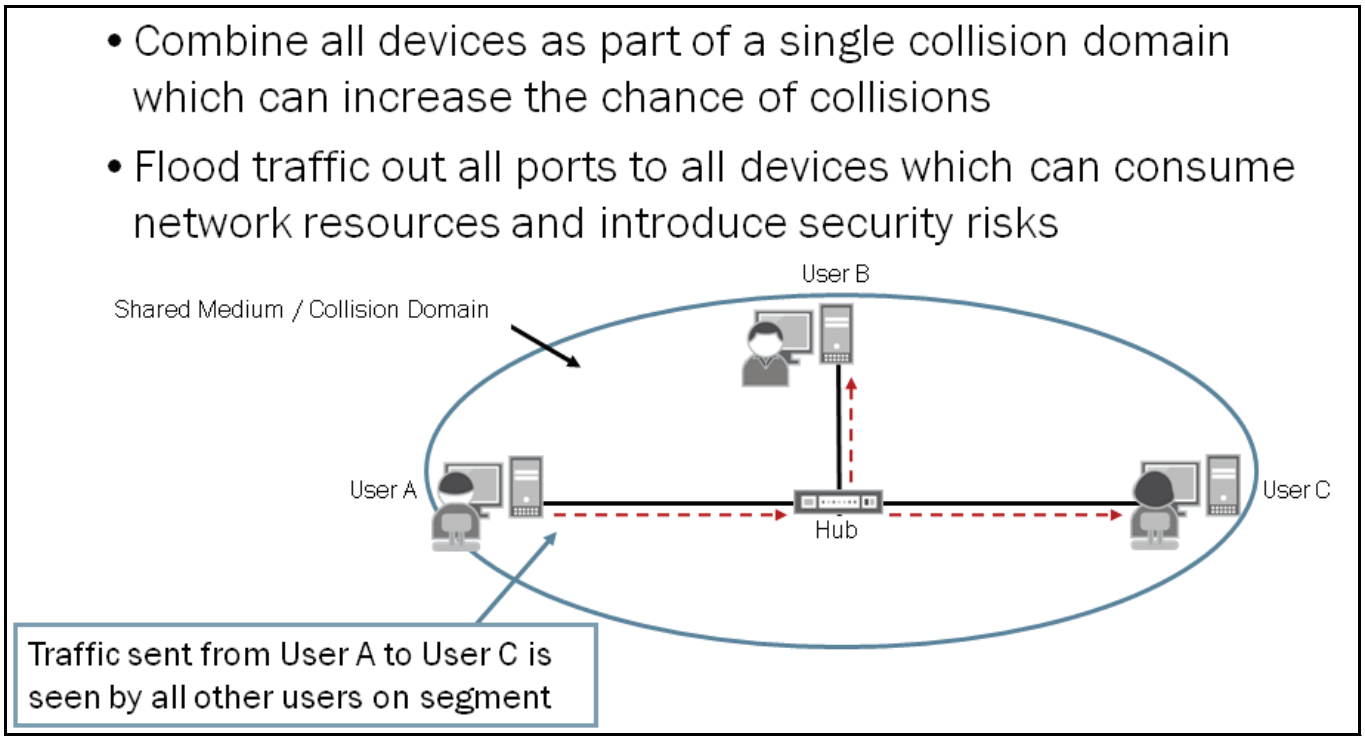
JNCIS-ENT Switching Study Guide

Chapter 1: Layer 2 Switching

This Chapter Discusses:

- The benefits of implementing switched LANs;
- Transparent bridging concepts and operations;
- Terminology and design considerations for switched LANs;
- Enterprise platforms that support Layer 2 switching;
- The configuration of interfaces for Layer 2 operations; and
- The display and interpretation of the Ethernet switching table.

Shared LANs



On a shared Ethernet LAN all devices share and communicate through a common medium. All devices participating on a shared medium are part of the same collision domain.

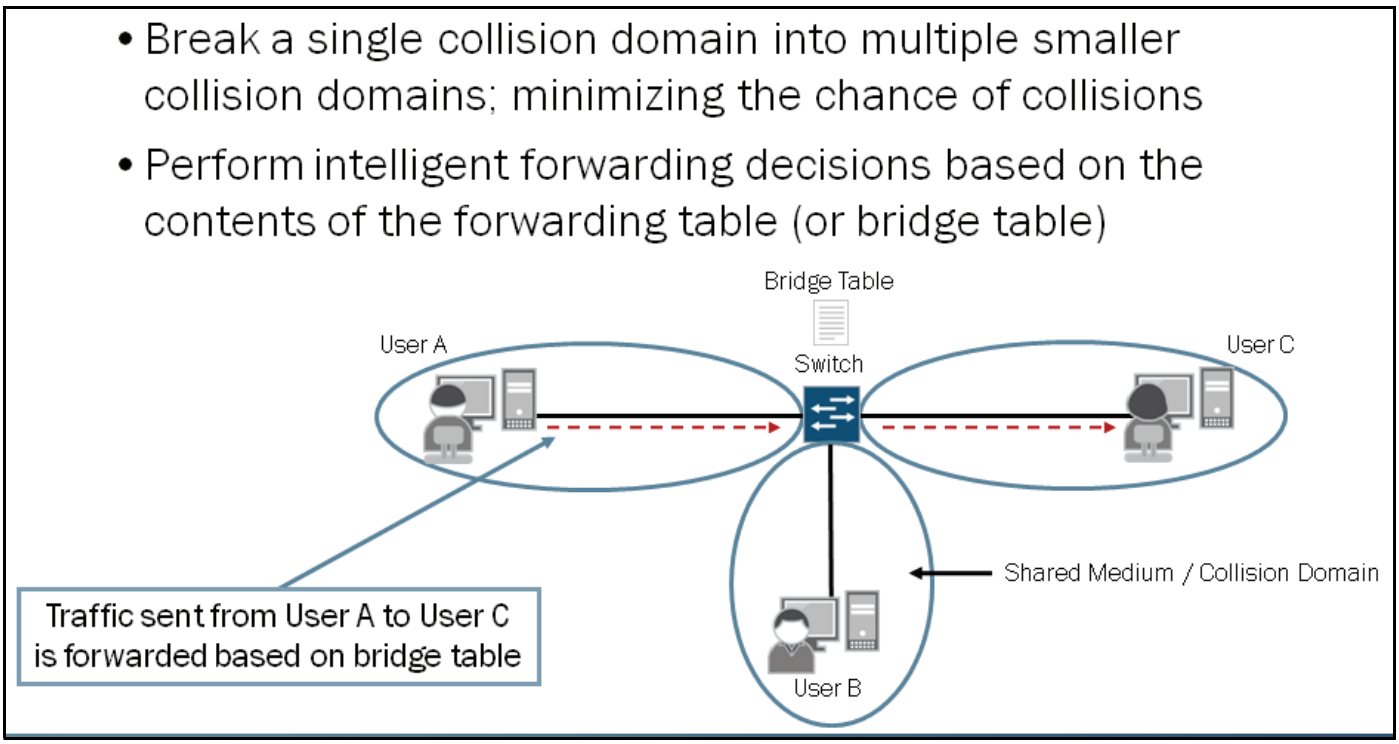
Ethernet uses the carrier-sense multiple access with collision detection (CSMA/CD) protocol to avoid and manage frame collisions. The sample topology on the graphic shows a series of nodes connected through a hub using a copper-based physical medium. This type of implementation only allows a single stream 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 a JAM signal is sent by the transmitting devices so all devices on the segment know a collision has occurred and that the line is in use. When node receive the JAM signal, they stop transmitting immediately and wait for a period of time before trying to send traffic. If the nodes continue to detect collisions, they progressively increase 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 is replicated out all ports on the hub and is seen by all other nodes on the shared Ethernet segment. This traffic-flooding approach, coupled with collisions, consumes network resources and can pose security risks.

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 are added to 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.

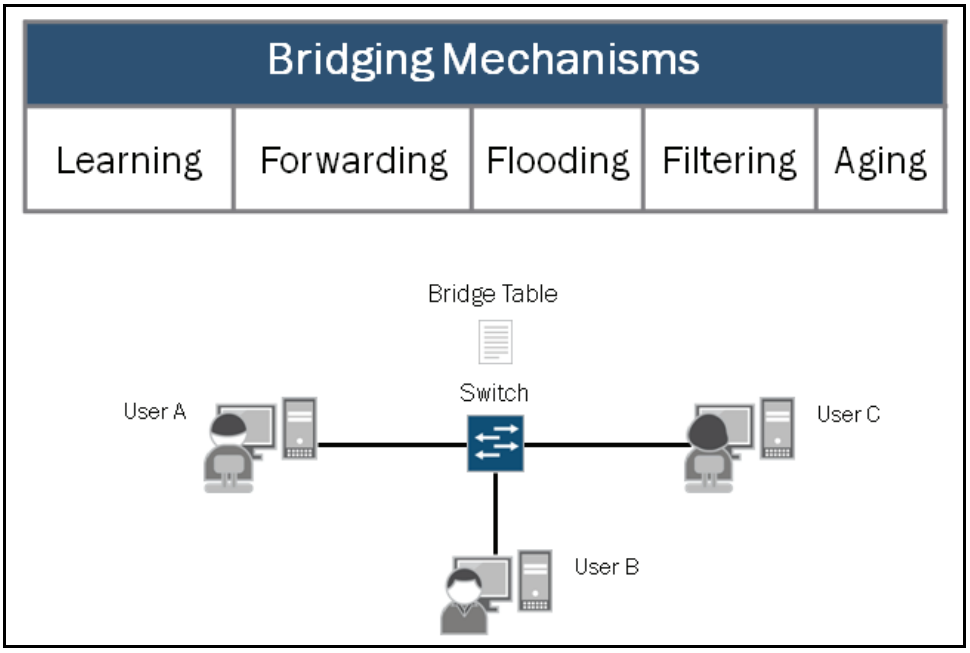
Switched LANs



Although similarities exist between shared and switched LANs, switched LANs do not have the same issues found in shared LANs and highlighted on the previous graphic. Switched LANs reduce the likelihood of collisions by breaking a single collision domain into multiple smaller collision domains. As shown in the sample diagram, switched LANs use switches rather than hubs. A collision domain in a switched LAN consists of the physical segment between a node and its connected switch port.

Using a switch increases network performance and minimizes some types of security risks by only forwarding traffic to its intended destination rather than always flooding traffic to all connected devices. Switches build and maintain a forwarding table, also known as a bridge table, to make forwarding decisions. We discuss the mechanisms switches use to build and maintain a bridge table on subsequent pages.

How Does Bridging Work?

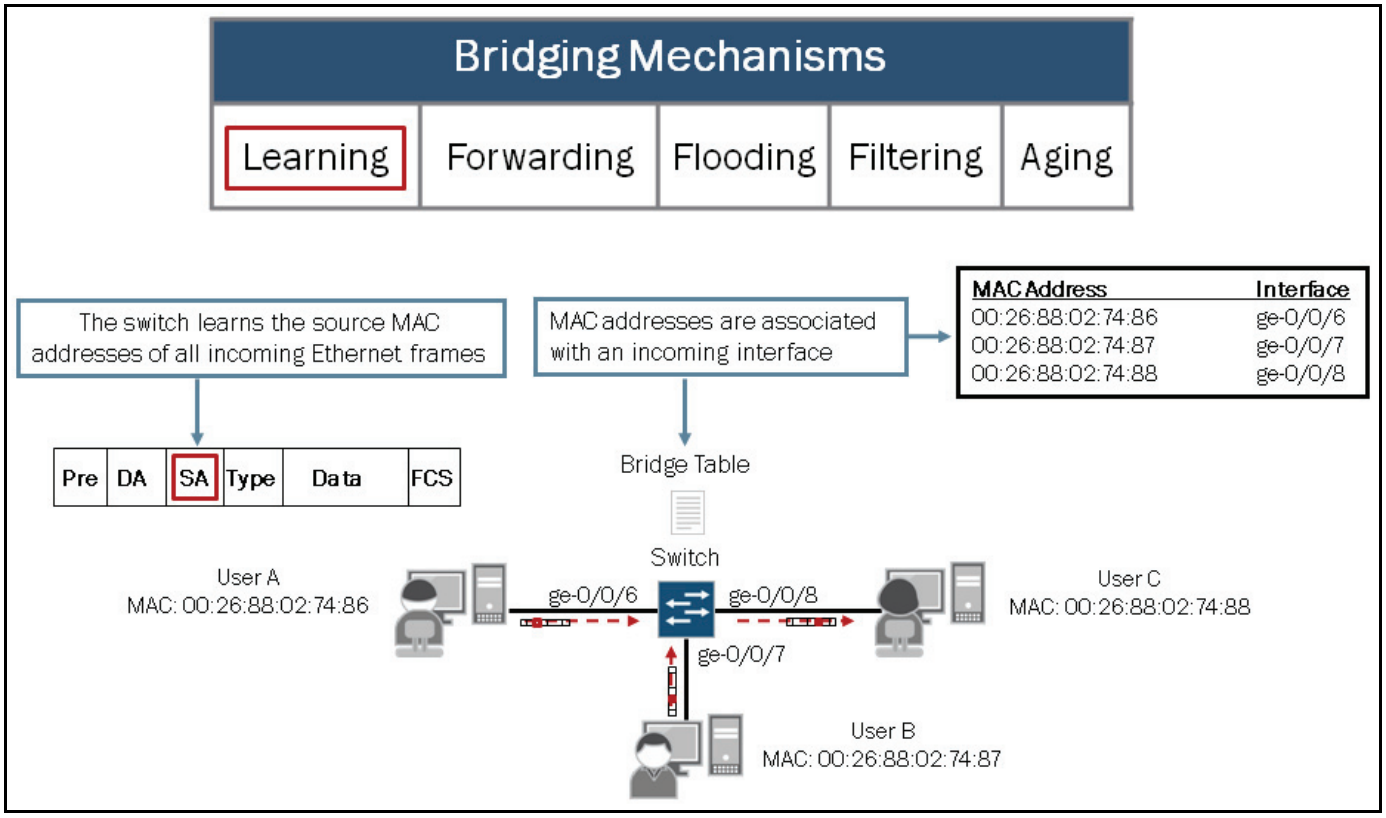


Defined in the IEEE 802.1D-2004 standard, bridging addresses some of the inherent problems of large shared Ethernet LANs. Bridging uses microsegmentation to divide a single collision domain into multiple, smaller bridged collision domains. Reducing the size of a collision domain effectively reduces the likelihood that collisions will 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 switch populates the bridge table based on the source MAC address of incoming frames received from devices participating in the bridged LAN. The switch makes an intelligent forwarding decision by comparing the destination MAC address of incoming frames to the contents of the bridge table. This approach reduces unnecessary traffic on the LAN. As shown on the graphic, several mechanisms contribute to the bridging process. We cover the listed bridging mechanisms in detail on subsequent graphics.

Learning

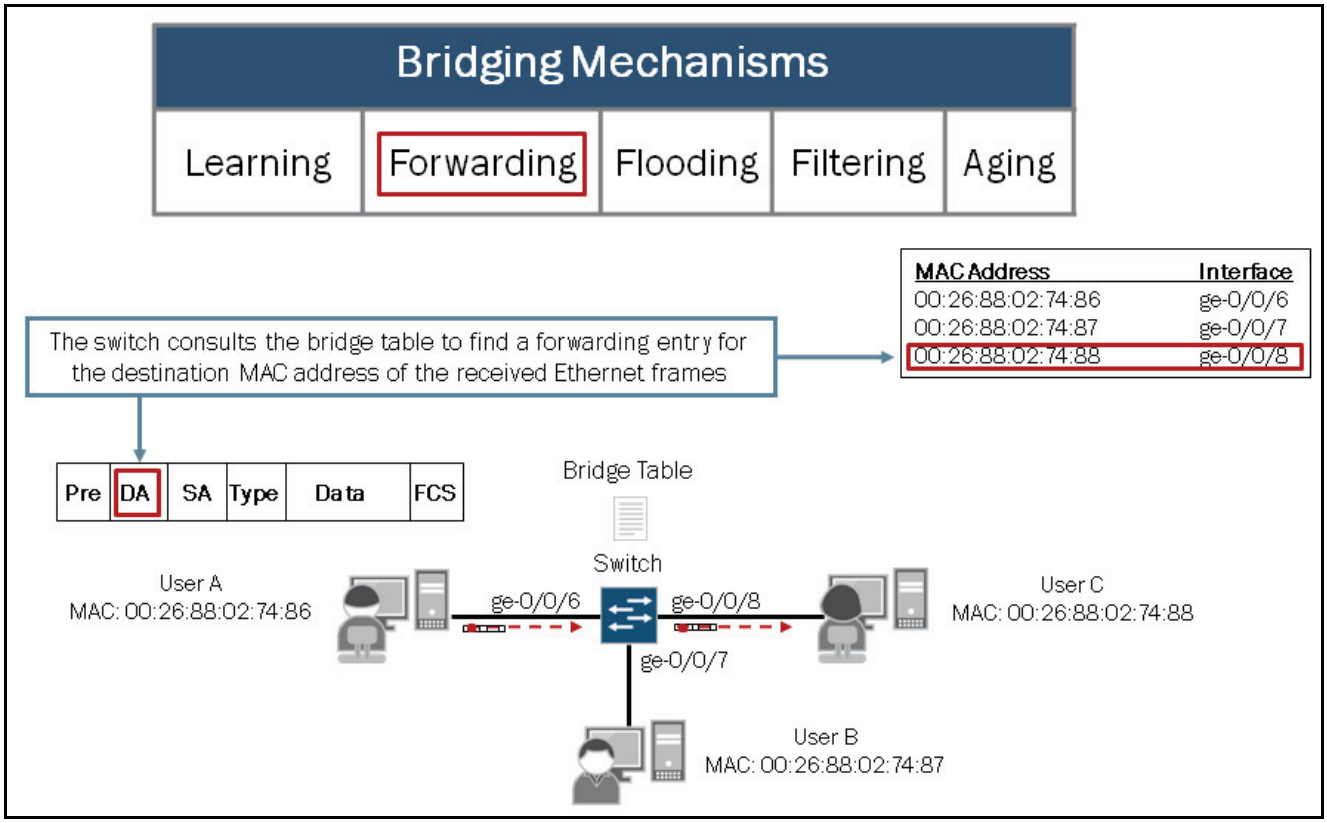


When a switch is first connected to an Ethernet LAN, it has no information about the devices connected to the network. *Learning* is the process a switch uses to obtain the MAC addresses of nodes on the network. The switch stores all learned MAC address in the bridge table. To learn MAC addresses, the switch examines the Ethernet header information of all received frames from the LAN, looking for source MAC addresses of sending nodes. The switch places learned MAC 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 when the MAC address was learned. The port information is used to forward traffic to its intended destination (*forwarding* mechanism) while the timestamp information is used to keep the bridge table up-to-date (*aging* mechanism). We discuss the *forwarding* and *aging* mechanisms in detail on subsequent pages in this section.

Note that MAC learning can be disabled on individual interfaces on EX Series switches. The command used to disable MAC learning follows:

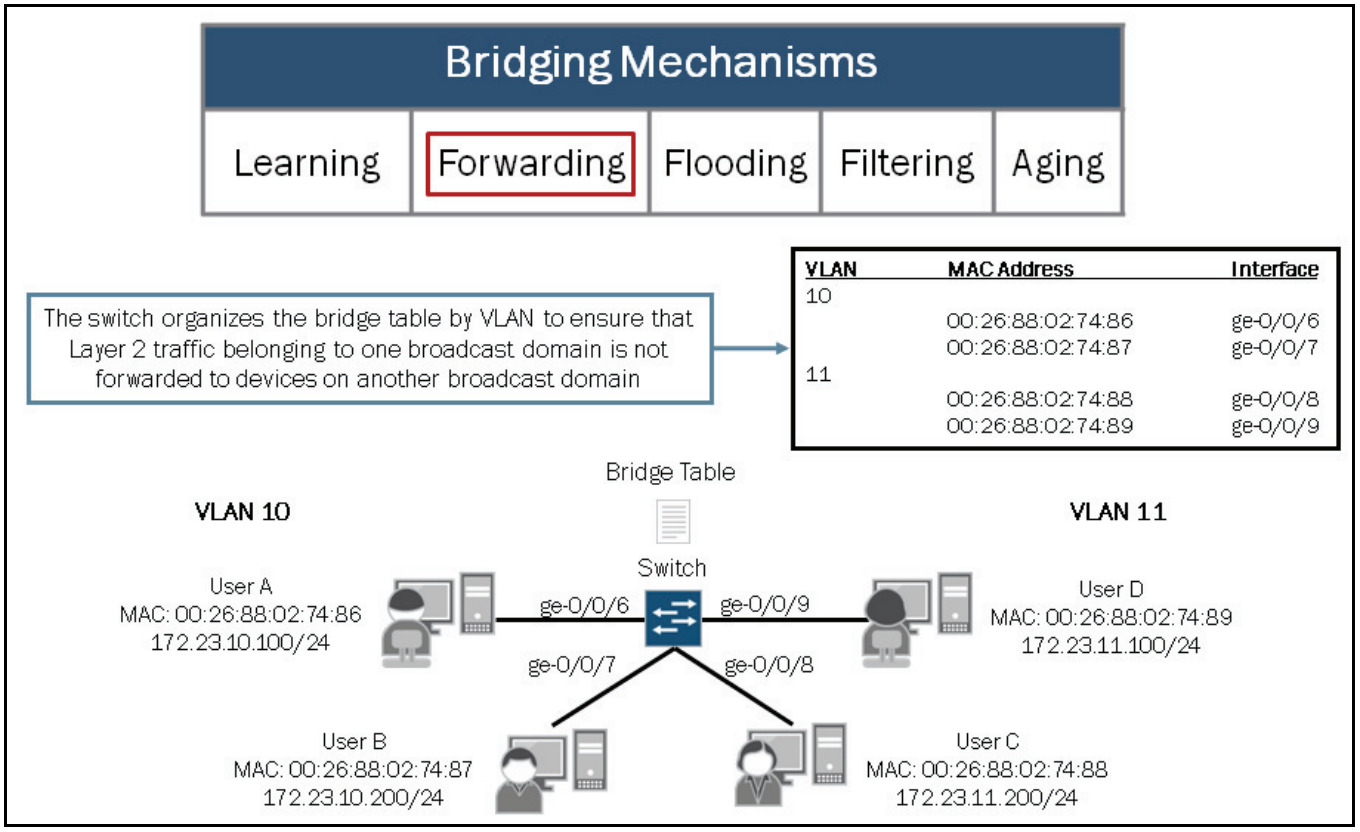
```
{master:0}[edit]
user@Switch# set ethernet-switching-options interfaces ge-0/0/0.0 no?
Possible completions:
  no-mac-learning      Disable mac learning for this interface
```

Forwarding: Part 1



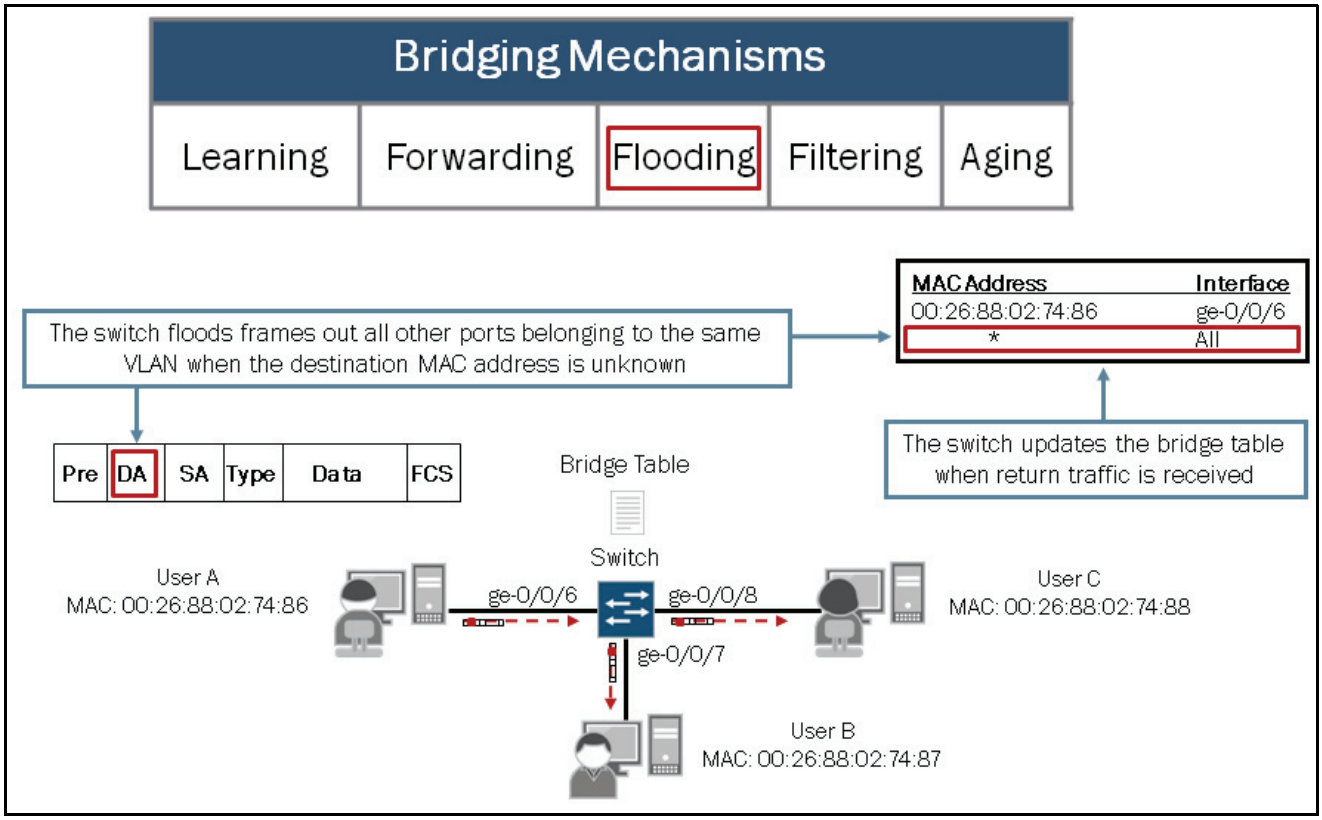
The *forwarding* mechanism is used by the switch to deliver traffic, passing it from an incoming interface to an outgoing interface that leads to (or toward) the destination. To forward frames, the switch consults the bridge table to see whether the table contains the MAC address corresponding to the frames' destination. 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. If the switch does not have a MAC entry in its bridge table, it floods the frame out all other interfaces belonging to the same broadcast domain (VLAN) as the interface on which the frame was received. The frame is not sent back out the ingress interface.

Forwarding: Part 2



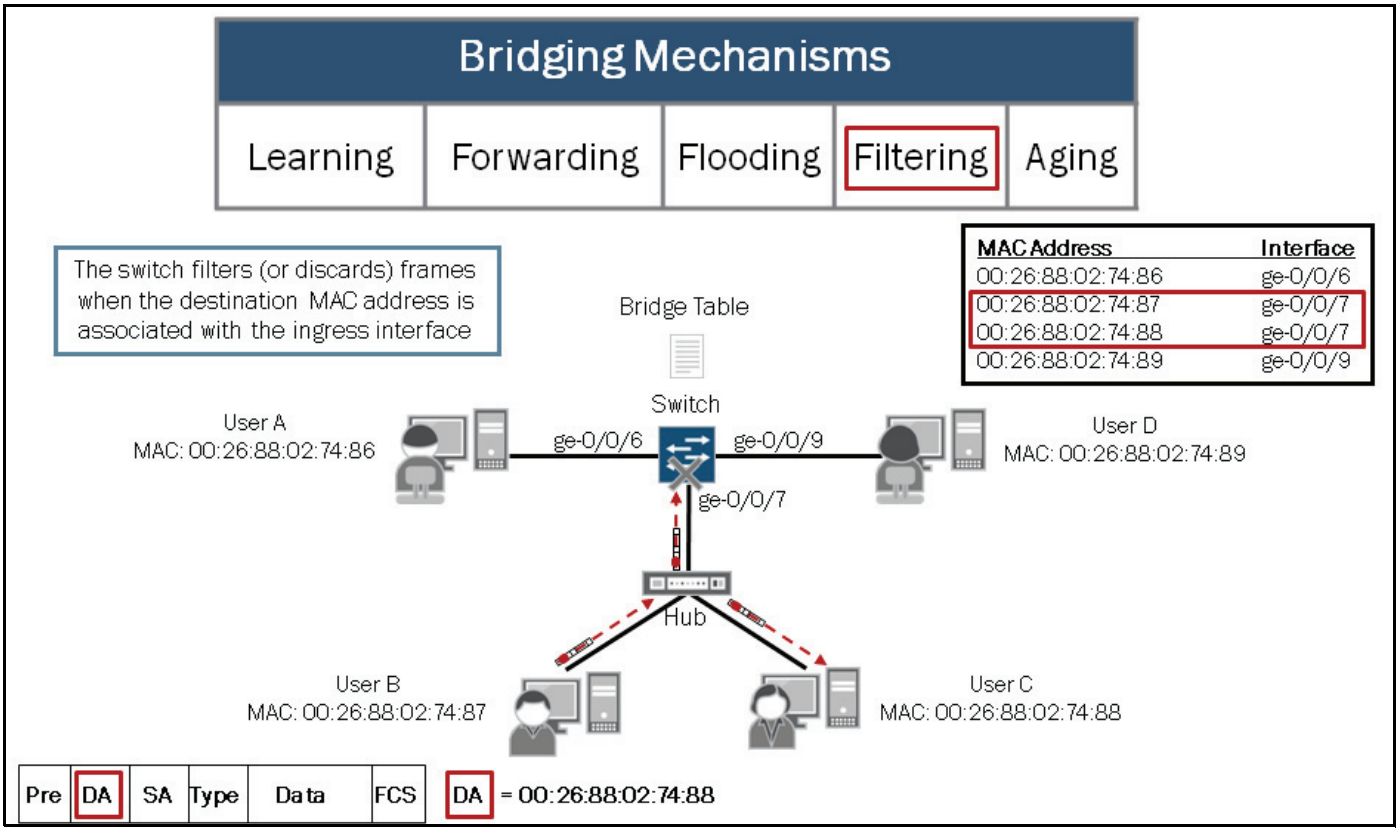
To forward frames, the switch consults the bridge table to see whether the table contains the MAC address corresponding to the frames' destination. The bridge table is organized by VLAN to ensure Layer 2 traffic is only forwarded out switch ports belonging to the same broadcast domain (VLAN) as the interface on which the frame was received.

Flooding



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 an unknown destination 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 and ingress port.

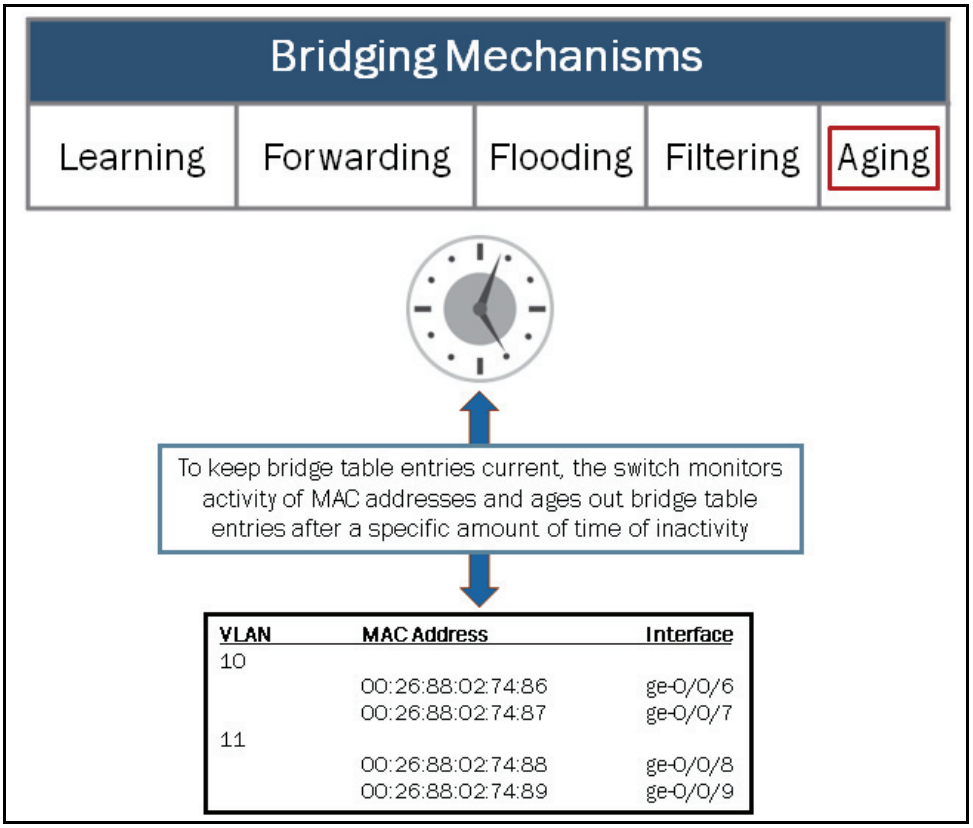
Filtering



The *filtering* mechanism is used to limit traffic to its associated segment or switch port. 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 switch ports are used to forward traffic to a specific node. The switch uses this information to filter traffic.

The graphic illustrates how a switch filters traffic. In this example the device associated with User B sends traffic destined to the device associated with User C (MAC address 00:26:88:02:74:88). Because the destination MAC address 00:26:88:02:74:88 is also associated with ge-0/0/7, the switch filters or discards the traffic.

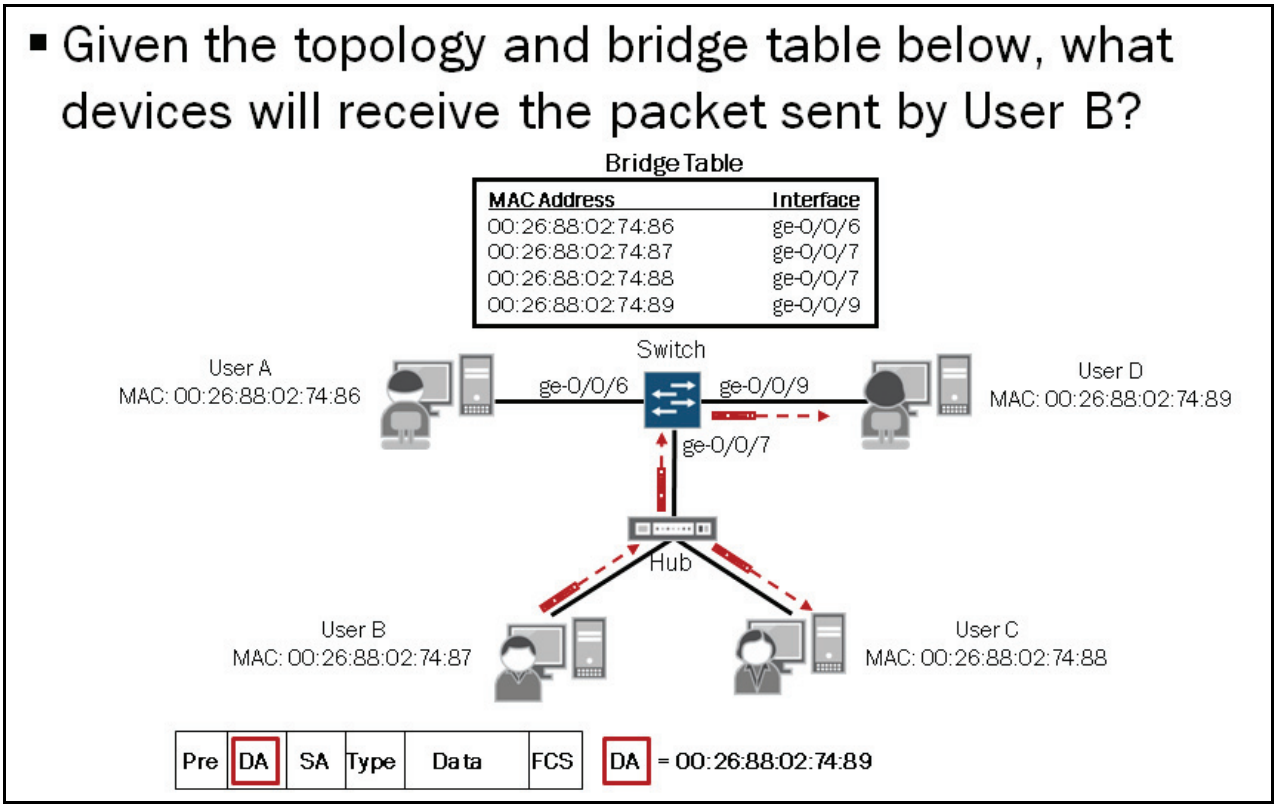
Aging



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 the information about the network node was learned. 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 a user-configured value, the switch removes the node's MAC address from the bridge table. The default aging timer interval is 300 seconds and can be configured for all VLANs or on a per-VLAN basis as shown here:

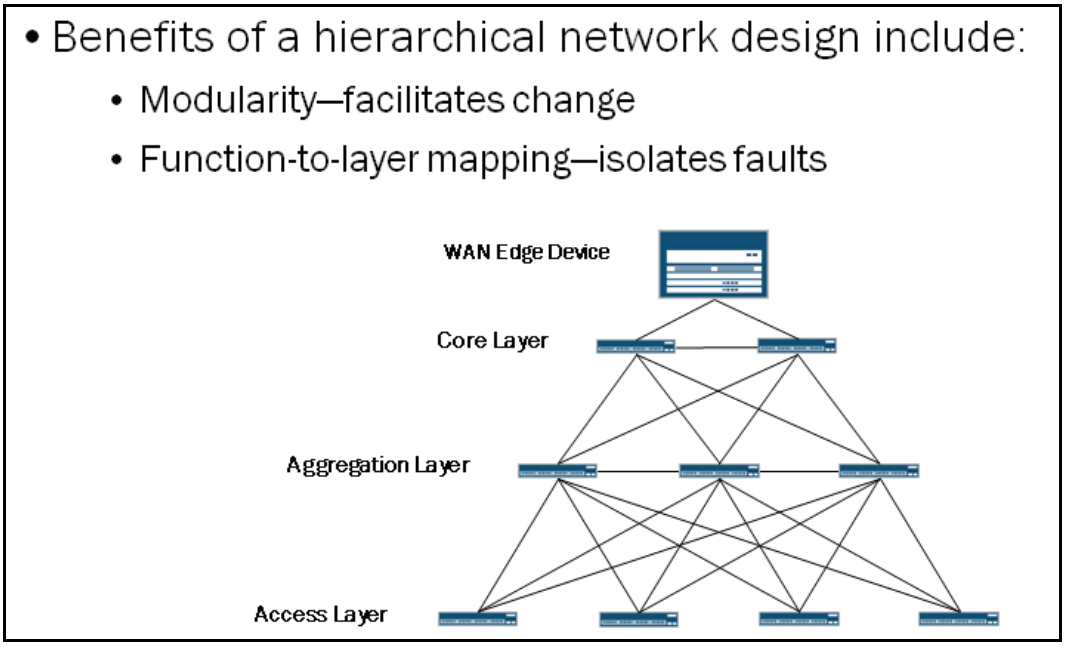
```
{master:0}[edit]
user@switch# set ethernet-switching-options mac-table-aging-time ?
Possible completions:
  <mac-table-aging-time>  MAC aging time (60..1000000 seconds)

{master:0}[edit]
user@switch# set vlans vlan-name mac-table-aging-time ?
Possible completions:
  <mac-table-aging-time>  MAC aging time (60..1000000 seconds)
```

Think About It



This graphic is designed to get you to think about the recently described concepts and mechanisms. This graphic illustrates a network topology where shared and switched LANs are merged. When User B sends traffic, the hub to which User B is connected floods the traffic out all ports. Based on this knowledge we know that the traffic will be received by User D and User C even though the traffic is intended for User D.

Multiple Layers



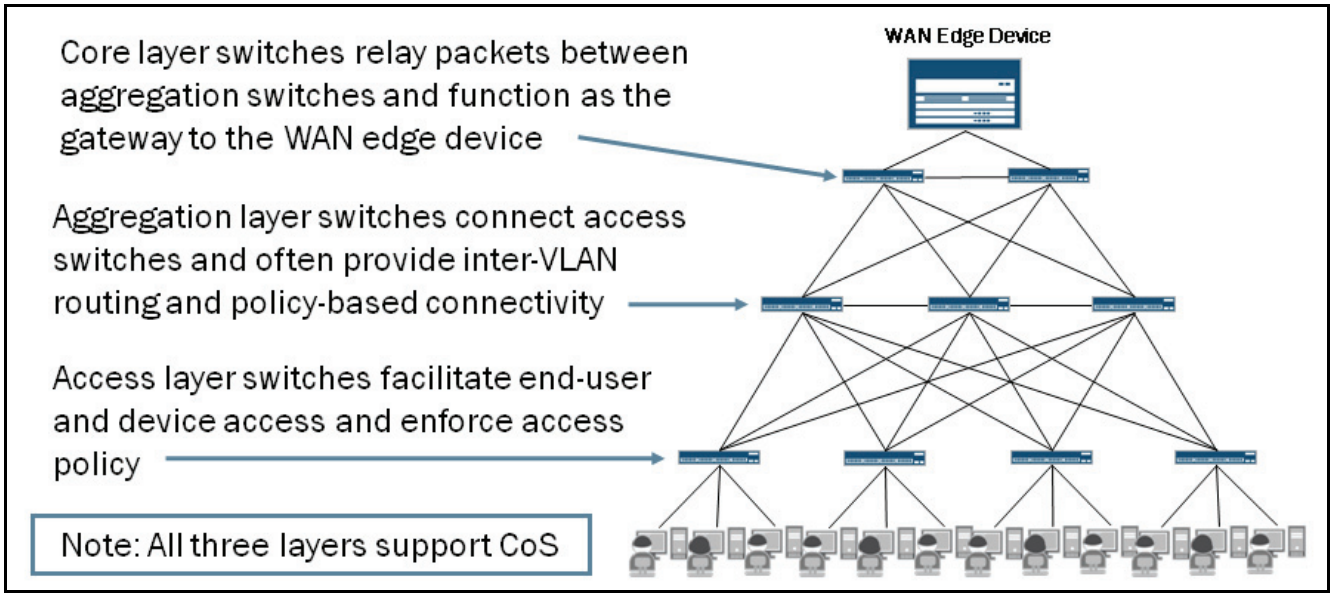
Switched networks are often hierarchical and consist of multiple layers. The diagram on the graphic illustrates the typical layers, which include access, aggregation (or distribution), and core. Each of these layers performs unique responsibilities.

Hierarchical networks are designed in a modular fashion. This inherent modularity facilitates change and makes this design option quite scalable. When working with a hierarchical network, the individual elements can be replicated as the network

grows. The cost and complexity of network changes is generally confined to a specific portion (or layer) of the network rather than to the entire network.

Because functions are mapped to individual layers, faults relating to a specific function can be isolated to that function's corresponding layer. The ability to isolate faults to a specific layer can greatly simplify troubleshooting efforts.

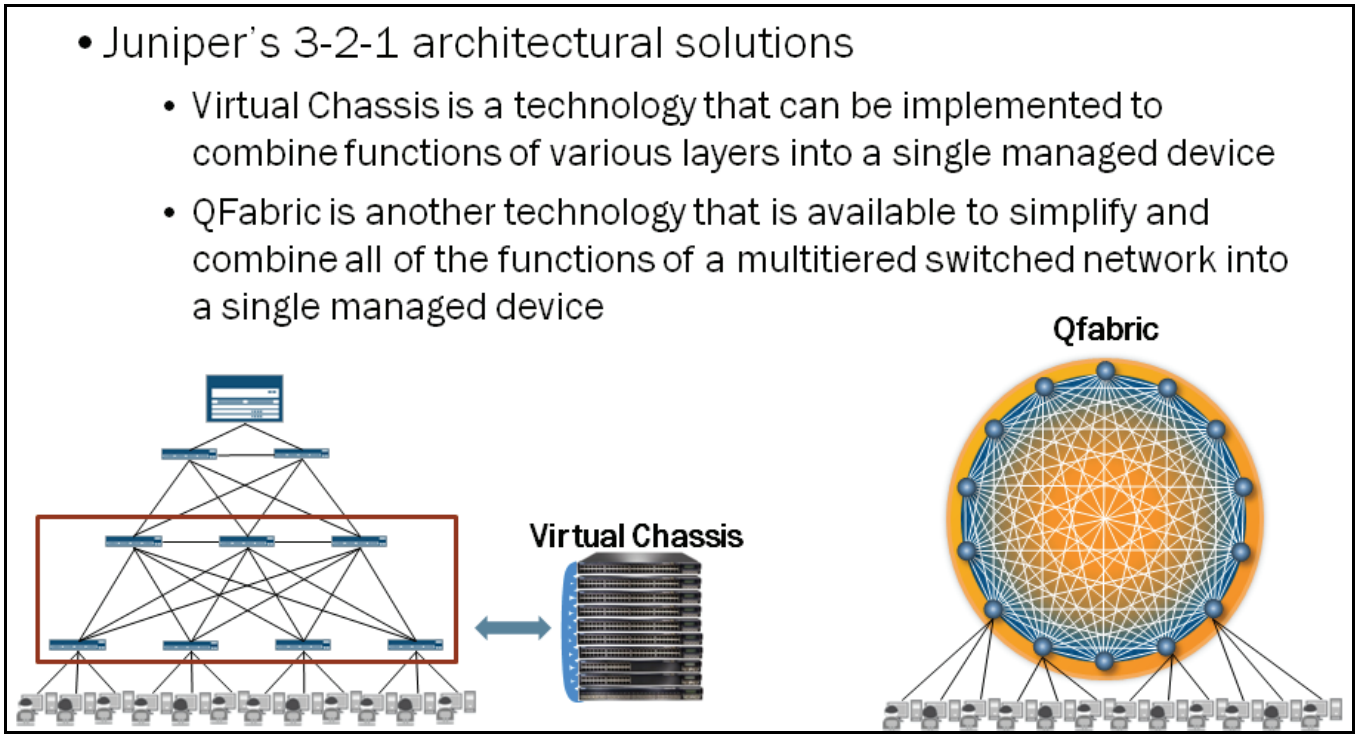
Functions of Layers



When designing a hierarchical switched network, individual layers are defined and represent specific functions found within a network. It is often mistakenly thought that the access, aggregation (or distribution), and core layers must exist in clear and distinct physical devices, but this is not a requirement, nor does it make sense in some cases. The layers are defined to aid successful network design and to represent functionality that exists in many networks.

The graphic highlights the access, aggregation, and core layers and provides a brief description of the functions commonly implemented in those layers. If CoS is used in a network, it should be incorporated consistently in all three layers.

Consolidation of Layers



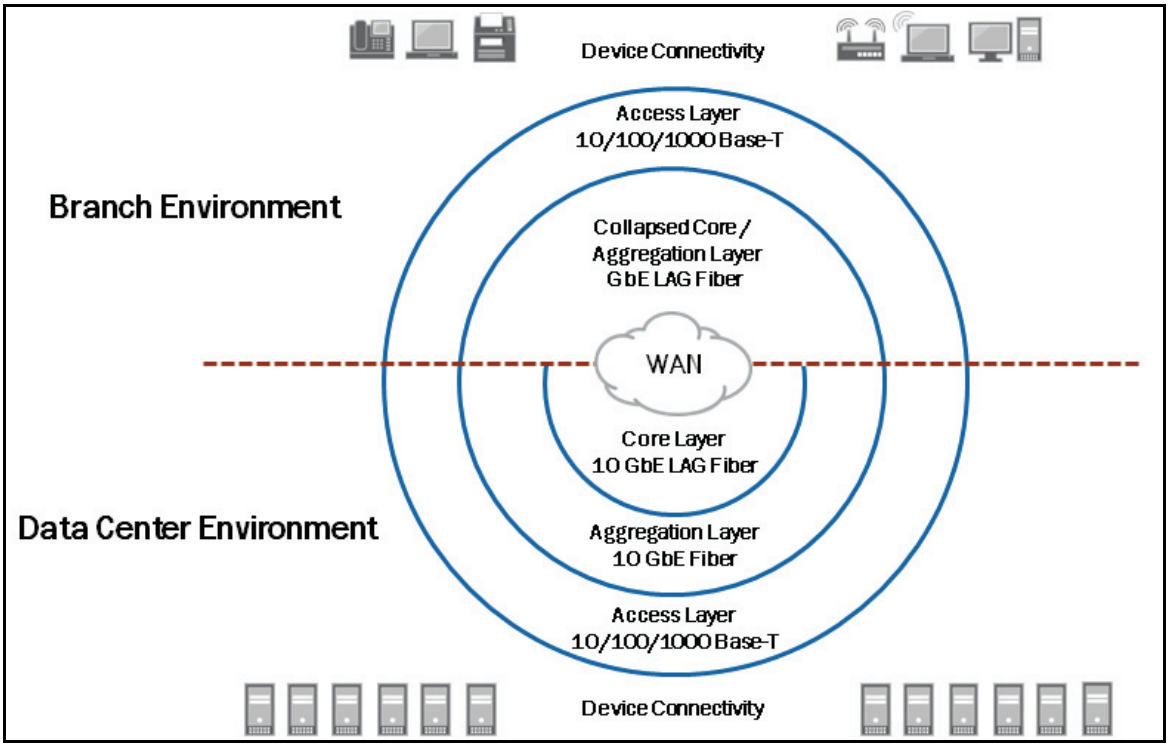
As networks and data centers continue grow, so does the complexity and overall number of devices that must be managed. We deliver a strategy for simplifying the data center network called the 3-2-1 data center network architecture. The 3-2-1 architecture eliminates layers of switching to flatten and collapse the network from today's three-tier tree structure to two layers, and ultimately, just one layer. This simplification is achieved by interconnecting multiple physical switches, creating a single, logical device that combines the performance and simplicity of a switch with the connectivity and resiliency of a network.

The key to the 3-2-1 architecture is fabric technology, which is the ability to make multiple devices appear, behave, and operate as one. This capability is available today with Virtual Chassis technology on select Juniper Networks EX Series switches. Virtual Chassis technology allows multiple interconnected switches to operate as a single, logical device.

For some small IT data centers featuring 1 Gigabit Ethernet (GbE) servers, the Virtual Chassis technology can allow customers to collapse their network into a single switching layer and manage the configuration as a single device. We discuss the Virtual Chassis technology in greater detail in a subsequent chapter in this study guide.

For larger organizations, Juniper's new Quantum Fabric (QFabric) technology enables the entire data center network to be managed as a single switch running the Junos OS, delivering the simplicity and efficiency businesses are looking for. Running a single instance of the Junos OS, QFabric will bring drastic change to the data center and will continue to allow growth for years to come. QFabric is outside the scope of this study guide and will not be covered in detail.

Comparing Environments



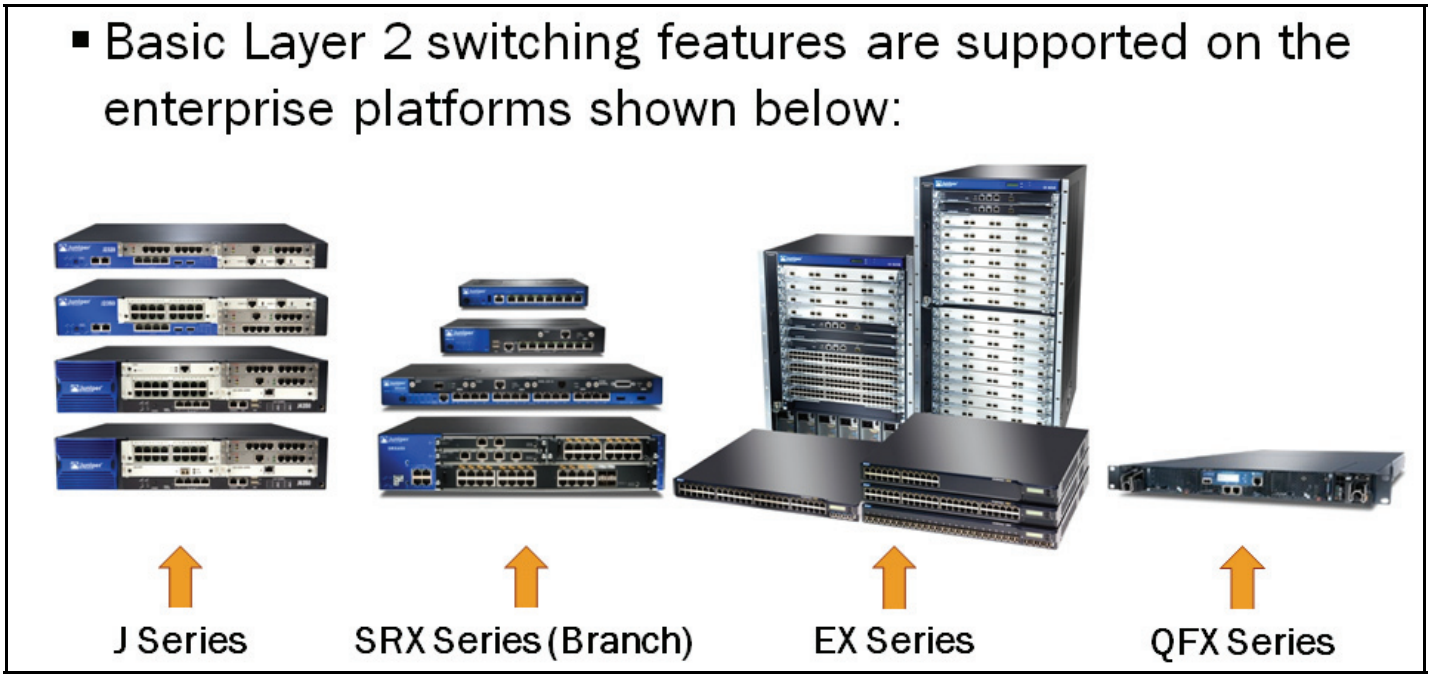
This graphic illustrates some points of comparisons between branch and data center environments. As shown on the graphic, branch environments typically do not have the three distinct hierarchical layers while data center (and many campus) environments do. In many branch environments, the core and aggregation layers are combined and the related functions are performed on the same physical device.

You can see that the types of devices found within the different environments can vary. In a branch or campus environment you will typically see a wide range of devices connected to the access layer such as end-user PCs, VoIP phones, printers, and wireless access points. In a data center environment, you will typically only see servers.

You can also see that the types of connections used within the different environments can vary. You will often use fiber connections between the access and aggregation or collapsed core layers to account for distance between the switches. Also, depending on your implementation, it might make sense to increase the throughput capacity of the links connecting the access and aggregation or collapsed core layers. You can increase the capacity by using a high-speed link, such as a 10 GbE interface, or by combining multiple lower-speed links in a link aggregation group (LAG). We discuss link aggregation in a subsequent chapter.

Our intent is to show some common design considerations. Your environment and design implementation may vary from that shown on the graphic.

Enterprise Devices and Layer 2 Switching



This graphic illustrates the enterprise platform families that run the Junos OS and that support Layer 2 switching operations. Note that the J Series and branch SRX Series do not support all of the Layer 2 switching features supported on the EX Series. The primary function of J Series and branch SRX Series is security while the primary function of the EX Series is switching. For this reason, this study guide focuses on the EX Series switches.

The QFX3500 is the first product available in the QFabric family. As a standalone switch, the QFX3500 is a high performance, low latency, top-of-rack switch. With a simple configuration change, the QFX3500 functions as QF/Node, which is the access component of QFabric architecture. Many of the topics discussed throughout this study guide do relate to the QFX3500 switch, but the QFX3500 can do many additional functions that are not covered by this study guide. The focus of this study guide is EX Series switches.

For Layer 2 switching support details for J Series and branch SRX Series, as well as additional information regarding the QFX3500 switch, refer to the technical publications at http://www.juniper.net/techpubs/.

Fixed Chassis Switches

Chassis	# Ports	POE	Uplink	Virtual Chassis
EX2200	12 - 48	Yes	2-4 x SFP	Yes
EX2500	24	No	0	No
EX3200	24 - 48	Yes	4 x SFP 2 x XFP	No
EX3300	24 - 48	Yes	4 x SFP 4 x XFP	Yes
EX4200	24 - 48	Yes	4 x SFP 2 x XFP	Yes
EX4500	40	No	8 x SFP+	Yes

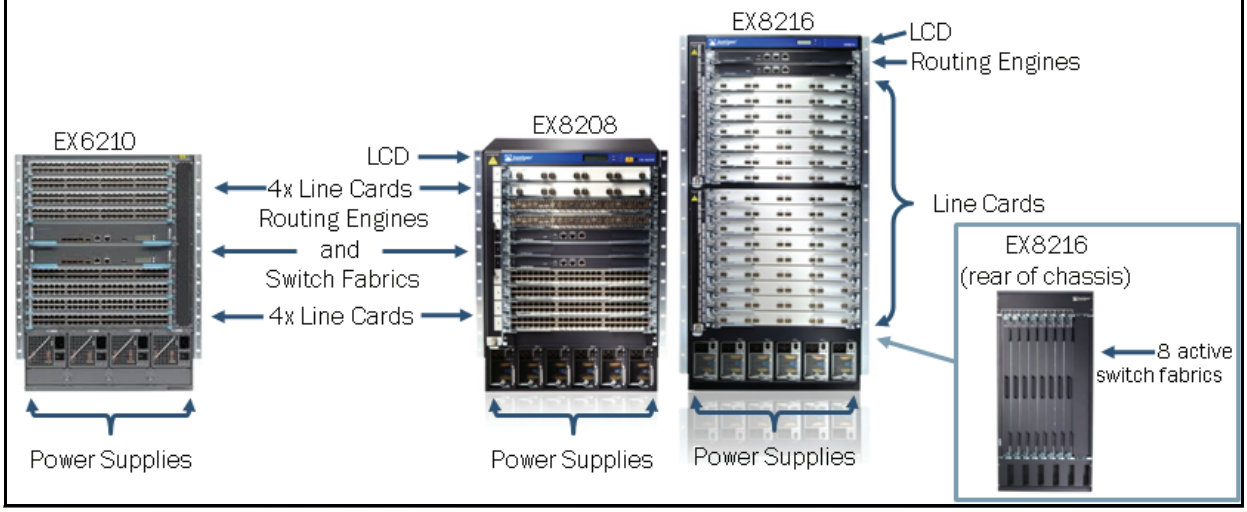
A brief description of the fixed chassis EX Series switches that run the Junos OS follows:

- The EX2200 line of fixed-configuration switches is ideal for access-layer deployments in branch and remote offices, as well as campus networks. Four platform configurations are available offering 24 and 48 10/100/1000BASE-T ports with or without Power over Ethernet (PoE).
- The EX2500 line of fixed-configuration switches is ideal for high-density 10-Gigabit Ethernet data center top-of-rack applications.
- The EX3200 line of fixed-configuration switches is ideal for access-layer deployments in branch and remote offices, as well as campus networks. Four platform configurations are available offering 24 and 48 10/100/1000BASE-T ports with either full or partial PoE.
- The EX3300 line of fixed-configuration switches with Virtual Chassis technology is ideal for access-layer deployments in branch and remote offices, as well as campus networks. Four platform configurations are available offering 24 and 48 10/100/1000BASE-T ports with either full or partial PoE.
- The EX4200 line of Ethernet switches with Virtual Chassis technology is ideal for data center, campus, and branch office environments. Eight platform configurations are available offering 24 and 48 10/100/1000BASE-T ports with either full or partial PoE, or 24 100/1000 BASE-X ports with no PoE. We discuss Virtual Chassis implementations in a subsequent chapter.
- The EX4500 line of Ethernet switches is ideal for high-density 10 gigabit per second (Gbps) data center top-of-rack as well as data center, campus, and service provider aggregation deployments. The EX4500 is designed to support Virtual Chassis technology and can be combined with the EX4200 switches within a single Virtual Chassis configuration to support environments where both GbE and 10 GbE servers are present.

Modular EX Series Chassis

Model	Routing Engines	Switch Fabrics	Power Supplies	Line Cards	Virtual Chassis
EX6210	2[1]	2[1]	4	8	No
EX8208	2[1]	3[1]	6	8	Yes[2]
EX8216	2	8	6	16	Yes[2]

[1] Switch Fabric and Routing Engine are on the same module (SRE)
[2] Virtual Chassis requires additional hardware (External Routing Engine [XRE])



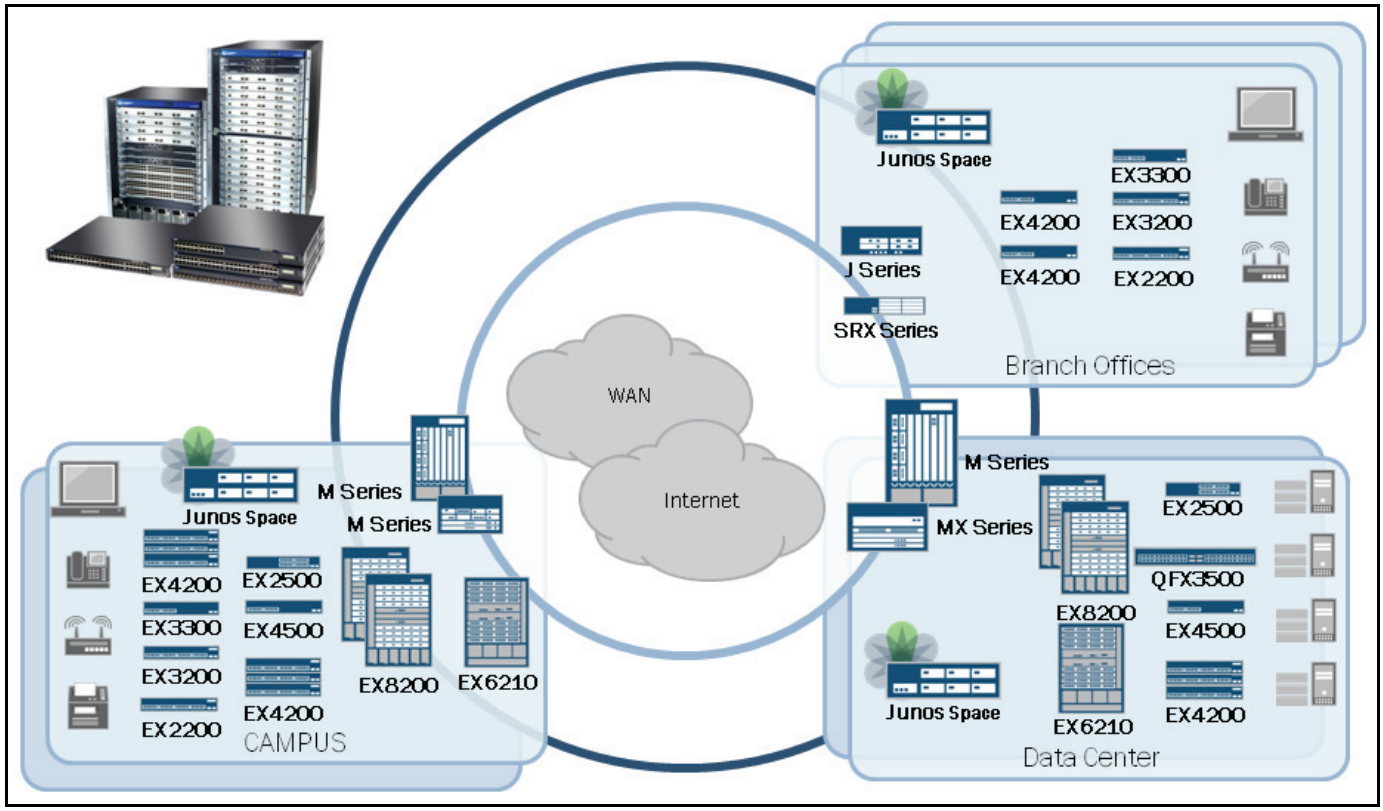
The EX6200 line of Ethernet switches is ideal for large enterprise campus and data center environments. The EX6210 is a 10-slot chassis featuring eight dedicated line-card slots that can accommodate a combination of the two available 48-port 10/1000/1000BASE-T line-card options. One option comes with support for POE and the other does not.

The EX8200 line of Ethernet switches is ideal for large campus and data center environments. The EX8200 Series switches also support the Virtual Chassis technology. The EX8200 Virtual Chassis can currently only support two chassis. It is also important to note that with the EX8200 Virtual Chassis, you must include an external Routing Engine (XRE) to manage both EX8200 switches.

Two chassis options exist for the EX8200 Series: an eight-slot option, EX8208, and a 16-slot option, EX8216. The EX8208 switch features eight dedicated line-card slots that can accommodate a variety of Ethernet interfaces. Options include a 48-port 10/100/1000BASE-T RJ-45 unshielded twisted pair (UTP) line card, a 48-port 100BASE-FX/1000BASE-X SFP fiber line card, and an eight-port 10GBASE-X SFP+ fiber line card. The EX8216 switch can accommodate any combination of EX8200 line Ethernet line cards. The EX8216 leverages the same EX8200 wire-speed line cards and power supplies used by the EX8208. The EX8200 Series switches deliver among the highest line-rate 10-Gigabit Ethernet port densities in the industry.

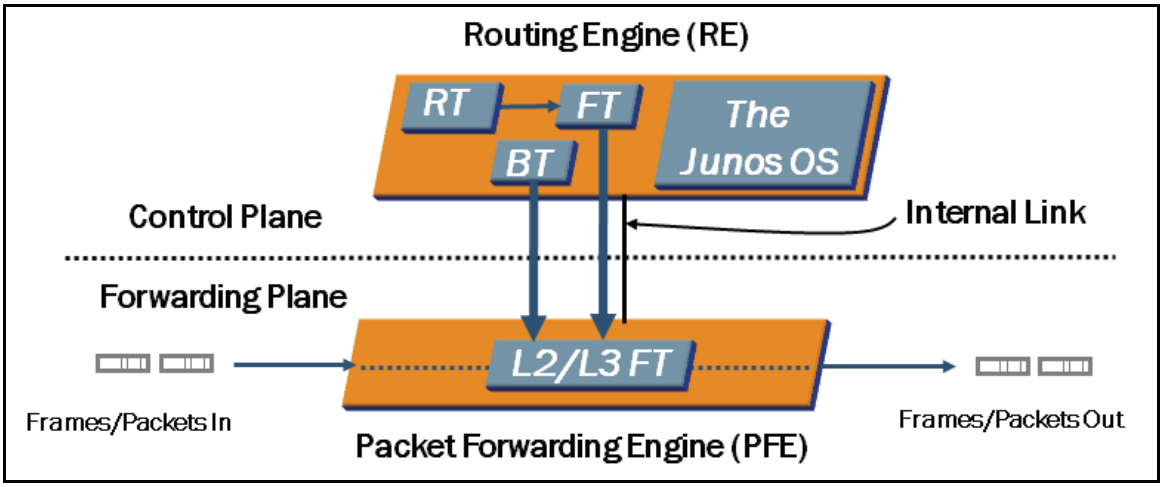
Support of the various Layer 2 switching features varies between platforms. For support information or more details on a specific EX Series platform, refer to the technical publications or the product-specific datasheets and literature found at: http://www.juniper.net/techpubs/ and http://www.juniper.net/us/en/products-services/switching/ex-series/, respectively.

EX Series Placement



This graphic illustrates the positioning of the various EX Series switches in data center, campus, and branch office environments. This graphic also shows the placement of Juniper's Routing and Security platforms as they might relate to the different environments. You can also note that all devices within each environment can all be managed through Junos Space.

Control and Forwarding Functions



EX Series switches, along with all other Junos-based devices, have a common design that separates the control and forwarding planes. To this end, all EX Series switches have two major components:

- *The Routing Engine (RE)*: The RE is the brains of the platform; it is responsible for performing protocol updates and system management. The RE runs various protocol and management software processes that reside inside a protected memory environment. The RE maintains the routing tables, bridging table, and primary forwarding table, and is connected to the PFE through an internal link.
- *The Packet Forwarding Engine (PFE)*: The PFE is responsible for forwarding transit frames, packets, or both through the switch. The PFE is implemented using ASICs on the EX Series platforms. Because this architecture separates control operations—such as protocol updates and system management—from frame and packet forwarding, the

switch can deliver superior performance and highly reliable deterministic operation. Note that the number of PFEs in each EX Series switch varies. Refer to the product-specific documentation for hardware architecture details.

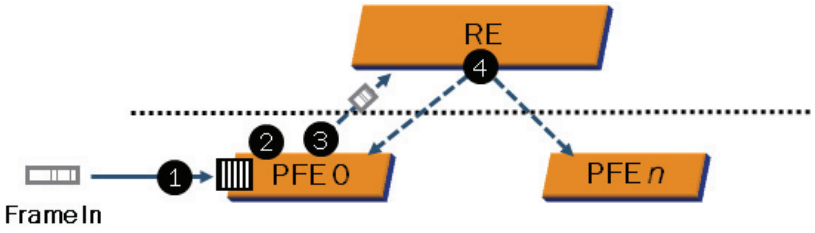
The PFE receives the Layer 2 and Layer 3 forwarding table from the RE by means of an internal link. Forwarding table updates are a high priority for the Junos OS kernel and are performed incrementally. The internal link that connects the RE and PFE is rate-limited to protect the RE from DoS attacks. The rate-limiting settings for this link are hard-coded and cannot be changed.

Because the RE provides the intelligence side of the equation, the PFE can simply do what it is told to do—that is, it forwards frames, packets, or both with a high degree of stability and deterministic performance.

Frame Processing: Unknown Source MAC Address

Processing steps for transit frames with an unknown source MAC address:

1. Frame enters ingress port and attached ingress PFE.
2. Ingress PFE performs a MAC address lookup and determines source MAC is unknown.
3. Ingress PFE sends header information to RE, where MAC is added or discarded (MAC limiting).
4. If RE adds new source MAC address to bridge table, newly added MAC entry is sent to and programmed into all PFEs.

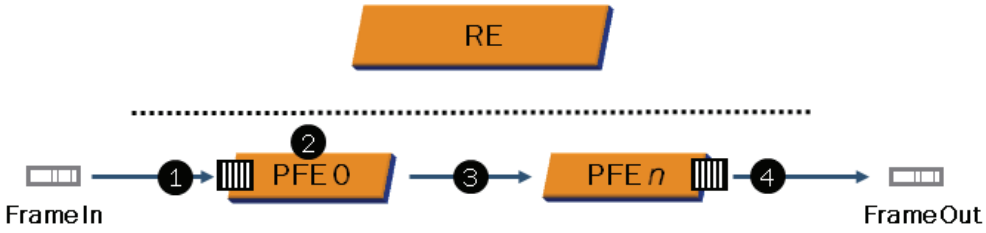


When frames enter a switch port, they are processed by the ingress PFE associated with that port. The ingress PFE determines how transit frames are processed and which lookup table is used when determining next-hop information. The PFE performs a lookup on the source and destination MAC address. In the example illustrated on the graphic, the source MAC address does not exist in the current bridging table.

In this example, the frame enters an ingress port and PFE. The ingress PFE performs a MAC address lookup and determines that the source MAC is unknown. The ingress PFE then sends the frame's header information to the RE through the internal link. The RE then either adds or discards the newly learned MAC address based on the configuration. If MAC limiting is enabled and a violation occurs, the MAC address is discarded or in other words is not added to the bridge table. If the configuration allows the newly learned MAC address to be added to the bridge table, the RE updates the bridge table with the relevant information and sends the update to all PFEs at which point the forwarding table on each PFE is updated accordingly.

Frame Processing: Known Destination MAC Address

Processing steps for transit frames with a known destination MAC address:
1. Frame enters ingress port and attached ingress PFE.
2. Ingress PFE performs a MAC address lookup and determines the egress PFE and port.
3. Ingress PFE forwards frame to egress PFE.
4. Egress PFE forwards frame out egress port toward destination. No additional lookup is needed.

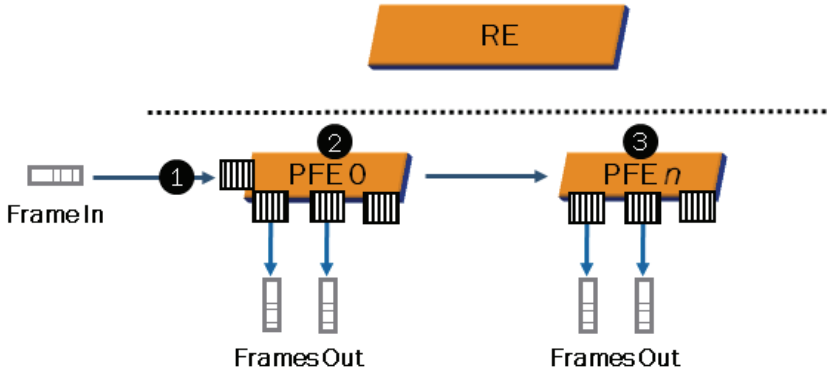


In the example illustrated on the graphic, the destination MAC address exists in the bridge table. If the egress port belongs to the ingress PFE, the frame is switched locally. If the egress port belongs to a PFE other than the ingress PFE (as shown in the example on the graphic), the frame is forwarded on through the switch fabric to the egress PFE where the egress switch port resides. This PFE might be a different PFE on the same switch or a remote PFE belonging to a separate member switch within the same Virtual Chassis system. We cover Virtual Chassis details in a subsequent chapter.

As illustrated on the previous graphic, if the source MAC address does not exist in the bridge table, the PFE extracts and sends the header to the RE to update the bridge table, which is part of the MAC learning process.

Frame Processing: Unknown Destination MAC Address

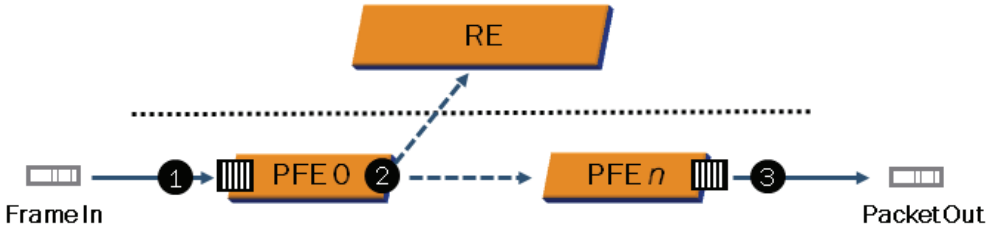
Processing steps for transit frames with an unknown destination MAC address:
1. Frame enters ingress port and attached ingress PFE.
2. Ingress PFE performs MAC address lookup, determines no entry exists then replicates frame out to other PFEs and all other local ports in the same broadcast domain (VLAN).
3. All other PFEs replicate frame and forward those frames out all egress ports in the same broadcast domain. No additional lookup is needed.



When the ingress PFE performs a lookup on the destination MAC address and no entry exists in the bridge table, the frame is flooded out all ports in the same broadcast domain. The frame is also flooded to other PFEs. However, the frame is not flooded out the port on which it was received. Once the switch sees return traffic from this MAC address, it adds the address to the bridge table. Frames with broadcast and multicast destination MAC addresses are also flooded in a similar fashion. Subsequent chapters of this study guide provide more details on MAC administration.

Frame Processing: Routed Packet

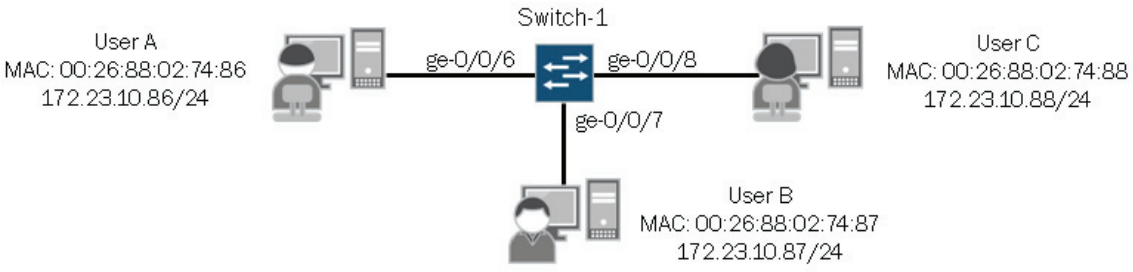
Processing steps for frames destined to the switch's MAC address:
1. Frame enters ingress port and attached ingress PFE.
2. Ingress PFE performs MAC address lookup. Because the destination MAC address belongs to the switch, PFE performs a Layer 3 lookup. a. If the destination IP address belongs to the switch, the decapsulated packet is sent to the RE for processing. b. If the destination IP address does not belong to the switch, the packet is forwarded to the egress PFE.
3. Egress PFE forwards packet out egress port toward destination. No additional lookup is needed.



When the PFE detects its own address as the destination MAC address, a Layer 3 lookup is performed. If the destination IP address belongs to the switch, the packet is forwarded to the RE. If the destination IP address does not belong to the switch but a Layer 3 forwarding table entry exists on the ingress PFE, the packet is forwarded to the egress PFE. If the destination IP address is not the switch and no Layer 3 forwarding table entry exists, the packet is discarded.

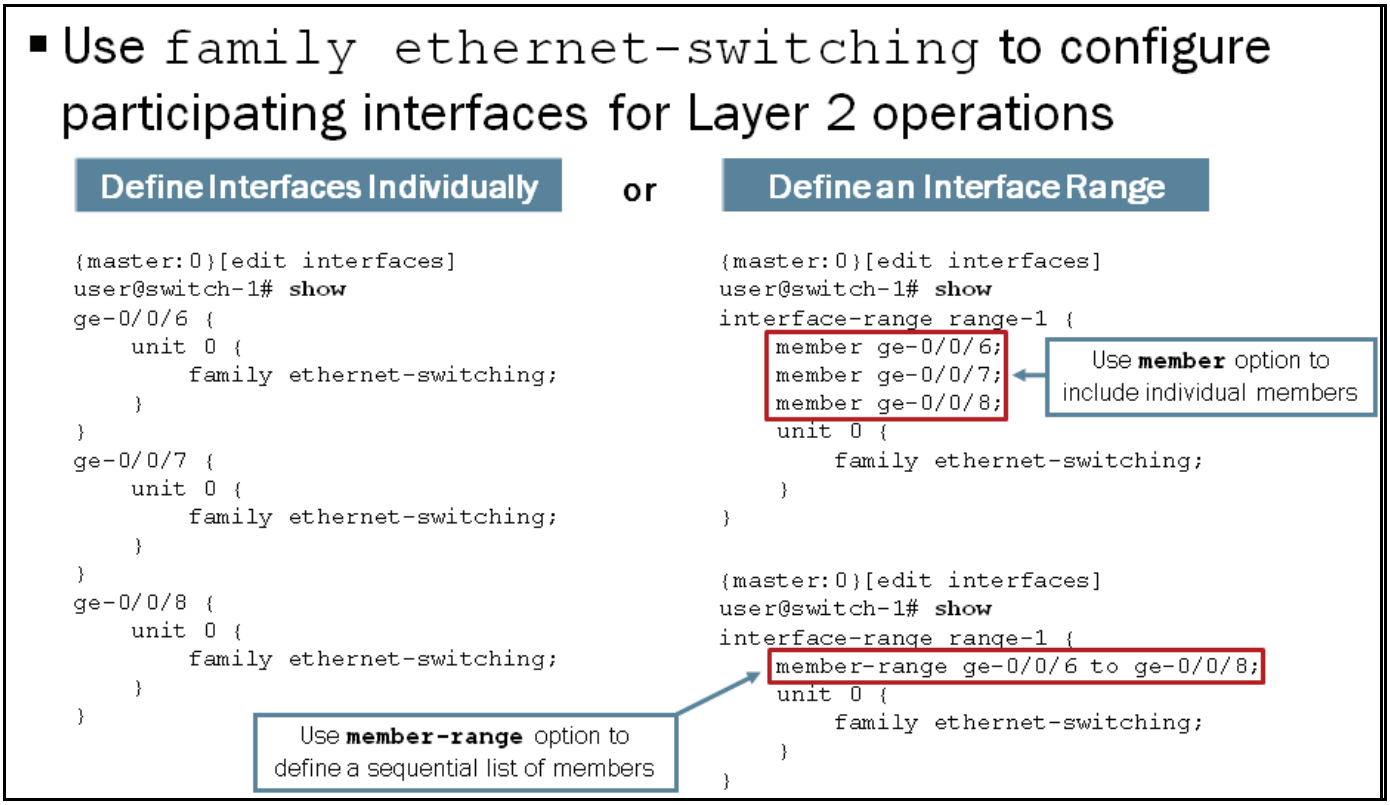
Case Study: Topology and Objectives

- Enable switching on Switch-1 to facilitate Layer 2 access for the users illustrated in the diagram below
- Use operational mode commands to verify proper Layer 2 switching operations



The graphic displays the topology and objectives for our case study.

Enabling Basic Layer 2 Functionality



The Ethernet switching process (eswd) is enabled by default on EX Series switches:

```
{master:0}
user@switch-1> show system processes | match "pid|eswd"
  PID  TT  STAT      TIME COMMAND
  823  ??  S      0:00.25 /usr/sbin/eswd -N
```

In addition to the Ethernet switching process, you must enable interfaces for Layer 2 operations.

The graphic illustrates Layer 2 interface configuration examples. You can define each interface individually, as shown on the left side of the graphic, or you can define a range of interfaces that share common configuration parameters, as shown on the right side of the graphic. If you define an interface range, you can specify individual interfaces belonging to the interface range using the **member** option or, if the member interfaces are sequentially ordered, you can specify an interfaces range in the <start-interface> to <end-interface> format using the **member-range** option.

You can also combine the two options within the same interface range as shown in the following example:

```
{master:0}[edit interfaces]
user@switch-1# show
interface-range range-1 {
    member ge-0/0/10;
    member-range ge-0/0/6 to ge-0/0/8;
    unit 0 {
        family ethernet-switching;
    }
}
```

Regardless of the configuration method you use, you must specify **family ethernet-switching** for interfaces operating in Layer 2 mode. All other interface configuration options are optional. Note that the factory-default configuration file for EX Series switches with built-in interfaces (excludes the EX8200 devices), all interfaces are configured for Layer 2 operations.

Verifying Interface State: Part 1

- Once configuration changes are activated, use the **show interfaces terse** command to verify interface status:

```
{master:0}[edit interfaces]
user@switch-1# commit and-quit
configuration check succeeds commit complete
Exiting configuration mode

{master:0}
user@switch-1> show interfaces terse | match "interface|0/6|0/7|0/8"
Interface               Admin Link Proto    Local                 Remote
ge-0/0/6                up    up
ge-0/0/6.0              up    up   eth-switch
ge-0/0/7                up    up
ge-0/0/7.0              up    up   eth-switch
ge-0/0/8                up    up
ge-0/0/8.0              up    up   eth-switch
```

Layer 2 interfaces should show the eth-switch value under the Proto column

Admin and Link state should show up for physical and logical interfaces

The graphic shows the expected status and details for Layer 2 interfaces. Note that the highlighted command is helpful in obtaining high-level status and protocol information. For usage statistics, errors, and detailed information, such as default interface settings, you should use the **show interfaces extensive** command. We illustrate the **show interfaces extensive** command on the next graphic.

Verifying Interface State: Part 2

- Use the **show interfaces extensive** command to view detailed interface information including default settings and error conditions:

```
{master:0}
user@switch-1> show interfaces extensive ge-0/0/6
Physical interface: ge-0/0/6, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 118, Generation: 138
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:19:e2:51:65:86, Hardware address: 00:19:e2:51:65:86
  Last flapped   : 2010-03-20 17:11:51 UTC (01:17:25 ago)
  Statistics last cleared: 2010-03-20 18:29:14 UTC (00:00:02 ago)
…
```

Default settings

This graphic illustrates the **show interfaces extensive** command which is helpful for determining detailed information such as the default interface settings, error conditions, and usage statistics.

In this example, you can see that the default `Speed` and `Duplex` settings are set to `Auto`. Generally, it is best to leave these default settings but some situations might exist where you must alter some settings. For example, in rare situations interface conflicts might occur, typically when interoperating with other vendors, which prohibits proper interface operation. In these cases, you might need to hard-code the speed and duplex settings on both sides to match.

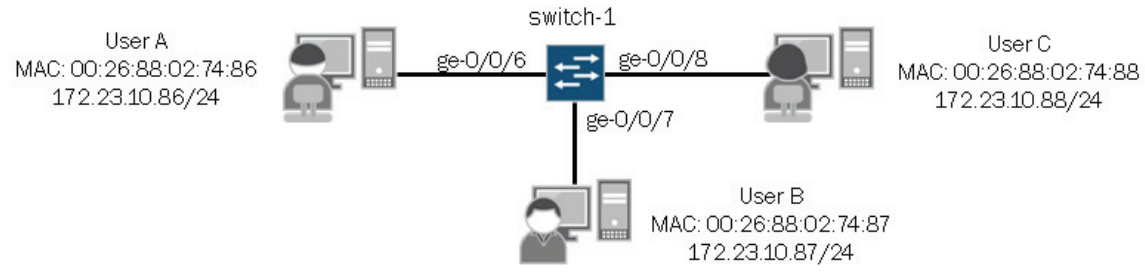
The following example shows the interface configuration where auto-negotiation is disabled and the speed and duplex settings are hard-coded to 1000 mbps and full-duplex respectively:

```
{master:0}
user@switch-1> show configuration interfaces ge-0/0/6
ether-options {
    no-auto-negotiation;
    link-mode full-duplex;
    speed {
        1g;
    }
}
unit 0 {
    family ethernet-switching;
}

{master:0}
user@switch-1> show interfaces extensive ge-0/0/6
Physical interface: ge-0/0/6, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 124, Generation: 138
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, Duplex: Full-Duplex,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Disabled,
...
```

Viewing Bridge Table Entries

- Use the `show ethernet-switching table` command to view contents of the bridge table:

```
{master:0}
user@switch-1> show ethernet-switching table
Ethernet-switching table: 4 entries, 3 learned
  VLAN              MAC address       Type          Age Interfaces
  default           *                 Flood           - All-members
  default           00:26:88:02:74:86 Learn           0 ge-0/0/6.0
  default           00:26:88:02:74:87 Learn           0 ge-0/0/7.0
  default           00:26:88:02:74:88 Learn           0 ge-0/0/8.0
```



Note: The capture was taken after traffic was passed between the three end-user devices.

Use the **`show ethernet-switching table`** command to view the contents of the bridge table. This command lists learned MAC addresses along with the corresponding VLAN, age, and interface. All entries are organized based on their associated VLAN. The sample output on the graphic also highlights each VLAN's flood entry, which is associated with all

interfaces for the VLAN. This entry is used to flood traffic, destined to an unknown destination, through all interfaces that belong to the same VLAN.

You can add the **extensive** option to view additional details:

```
{master:0}
user@switch-1> show ethernet-switching table extensive
Ethernet-switching table: 4 entries, 3 learned

    VLAN: default, Tag: 0, MAC: *, Interface: All-members
    Interfaces:
                ge-0/0/6.0, ge-0/0/7.0, ge-0/0/8.0
    Type: Flood
    Nexthop index: 1304

    VLAN: default, Tag: 0, MAC: 00:26:88:02:74:86, Interface: ge-0/0/6.0
    Type: Learn, Age: 1:16, Learned: 1:30
    Nexthop index: 1303

    VLAN: default, Tag: 0, MAC: 00:26:88:02:74:87, Interface: ge-0/0/7.0
    Type: Learn, Age: 0, Learned: 1:30
    Nexthop index: 1305

    VLAN: default, Tag: 0, MAC: 00:26:88:02:74:88, Interface: ge-0/0/8.0
    Type: Learn, Age: 1:00, Learned: 1:25
    Nexthop index: 1306
```

To view the Layer 2 forwarding table, issue the **show route forwarding-table family ethernet-switching** command:

```
{master:0}
user@switch-1> show route forwarding-table family ethernet-switching
Routing table: default.ethernet-switching
ETHERNET-SWITCHING:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd    66     1
2, *               user     0                    comp  1304     2
2, *               intf     0                    rslv  1302     1
2, 00:26:88:02:74:86 user     0                 ucst  1303     3 ge-0/0/6.0
2, 00:26:88:02:74:87 user     0                 ucst  1305     3 ge-0/0/7.0
2, 00:26:88:02:74:88 user     0                 ucst  1306     3 ge-0/0/8.0
```

Clearing Bridge Table Entries

- Use the `clear ethernet-switching table` commands to clear bridge table entries
  - You can clear entries based on interface, MAC, or VLAN

```
{master:0}
user@switch-1> show ethernet-switching table
Ethernet-switching table: 4 entries, 3 learned
  VLAN              MAC address       Type          Age Interfaces
  default           *                 Flood           - All-members
  default           00:26:88:02:74:86 Learn           0 ge-0/0/6.0
  default           00:26:88:02:74:87 Learn          35 ge-0/0/7.0
  default           00:26:88:02:74:88 Learn          33 ge-0/0/8.0

{master:0}
user@switch-1> clear ethernet-switching table interface ge-0/0/6.0

{master:0}
user@switch-1> show ethernet-switching table
Ethernet-switching table: 3 entries, 2 learned
  VLAN              MAC address       Type          Age Interfaces
  default           *                 Flood           - All-members
  default           00:26:88:02:74:87 Learn       1:04 ge-0/0/7.0
  default           00:26:88:02:74:88 Learn       1:02 ge-0/0/8.0
```

Use the **`clear ethernet-switching table`** command to clear all entries within the MAC address table. Optionally, you can clear individual MAC entries or all MAC entries associated with a specific VLAN using the available options shown in the following output:

```
{master:0}
user@switch-1> clear ethernet-switching table ?
Possible completions:
  <[Enter]>            Execute this command
  interface            Name of interface
  mac                  MAC address
  management-vlan      Management VLAN
  vlan                 Name of VLAN
  |                    Pipe through a command
```

Defining Static Bridge Table Entries

- You can define static bridge table entries under [edit ethernet-switching-options]:

```
{master:0}[edit ethernet-switching-options]
user@switch-1# show
static {
    vlan default {
        mac 00:26:88:02:74:86 next-hop ge-0/0/6.0;
        mac 00:26:88:02:74:87 next-hop ge-0/0/7.0;
        mac 00:26:88:02:74:88 next-hop ge-0/0/8.0;
    }
}

{master:0}[edit ethernet-switching-options]
user@switch-1# run show ethernet-switching table
Ethernet-switching table: 4 entries, 0 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:26:88:02:74:86 Static         - ge-0/0/6.0
  default           00:26:88:02:74:87 Static         - ge-0/0/7.0
  default           00:26:88:02:74:88 Static         - ge-0/0/8.0
```

Normally, MAC addresses are learned and added to the bridge table dynamically when traffic enters an interface. You can add static MAC addresses to the MAC address table if desired. The graphic illustrates the configuration used to statically define bridge table entries as well as the expected output for statically defined bridge table entries.

Review Questions

1. What are the key differences between shared and switched LANs?
2. List and describe the bridging mechanisms.
3. What layers exist in hierarchical Layer 2 networks and what functions are associated with each layer?

Answers

1.

Switched LANs break a single environment into multiple smaller collision domains which minimizes the chance of collisions. Shared LANs place all devices into a single collision domain which increases the chance of collisions; especially if a large number of devices exist. Switched LANs perform intelligent forwarding decisions based on the contents of the bridge table while shared LANs always flood traffic, which consumes resources unnecessarily and can pose some security risk.

2.

Learning is a process the switch uses to obtain the MAC addresses of nodes on the network. The forwarding mechanism is used by the switch to deliver traffic, passing it from an incoming interface to an outgoing interface that leads to (or toward) the destination. Flooding is a transparent mechanism used to deliver packets to unknown MAC addresses. The filtering mechanism is used to limit traffic to its associated broadcast domain or VLAN. Finally, the switch uses aging to ensure that only active MAC address entries are in the bridge table.

3.

Hierarchical Layer 2 networks can have access, aggregation, and core layers depending on the size and implementation approach. The access layer facilitates end-user and device access to the network and enforces access policy. The aggregation layer connects access switches together and often provides inter-VLAN routing and policy-based connectivity. The core layer switches packets between aggregation switches and functions as the gateway to the WAN edge device.

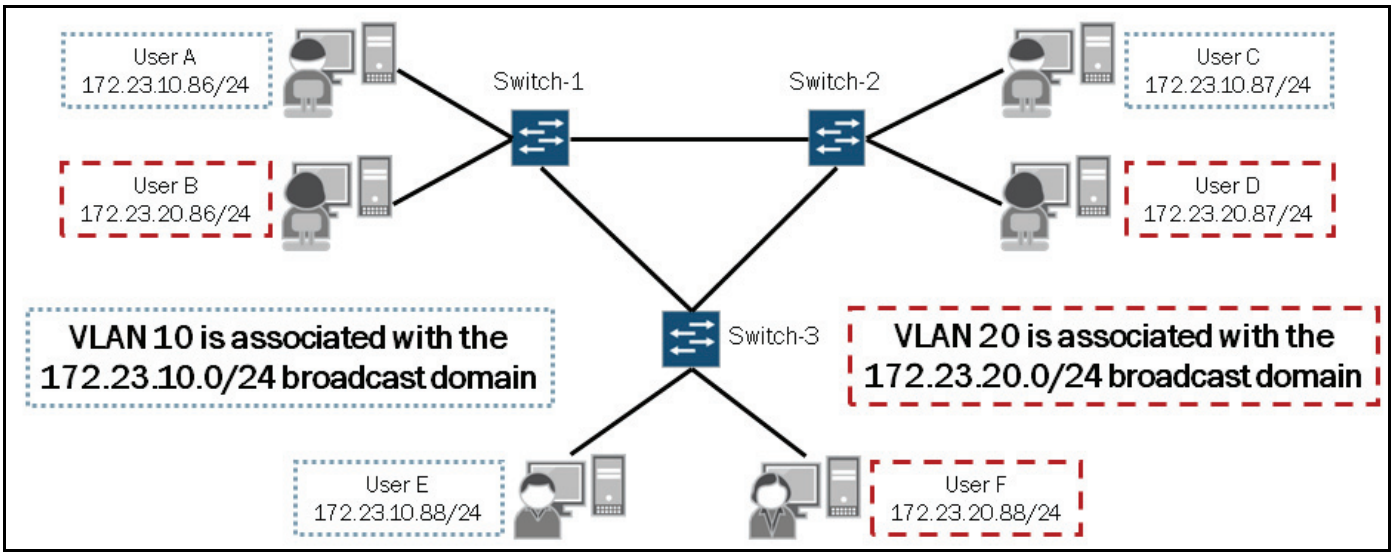
JNCIS-ENT Switching Study Guide

Chapter 2: Virtual Networks

This Chapter Discusses:

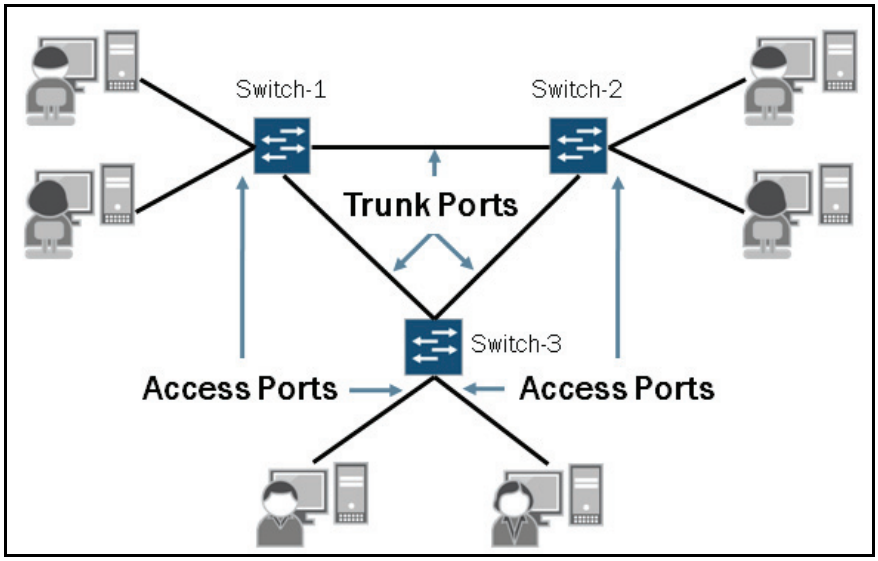
- The concept of a virtual network;
- Access and trunk ports;
- The configuration and monitoring of virtual LANs (VLANs);
- Voice and native VLAN concepts and configuration;
- Inter-VLAN routing operations; and
- The configuration and monitoring of inter-VLAN routing.

VLAN Defined



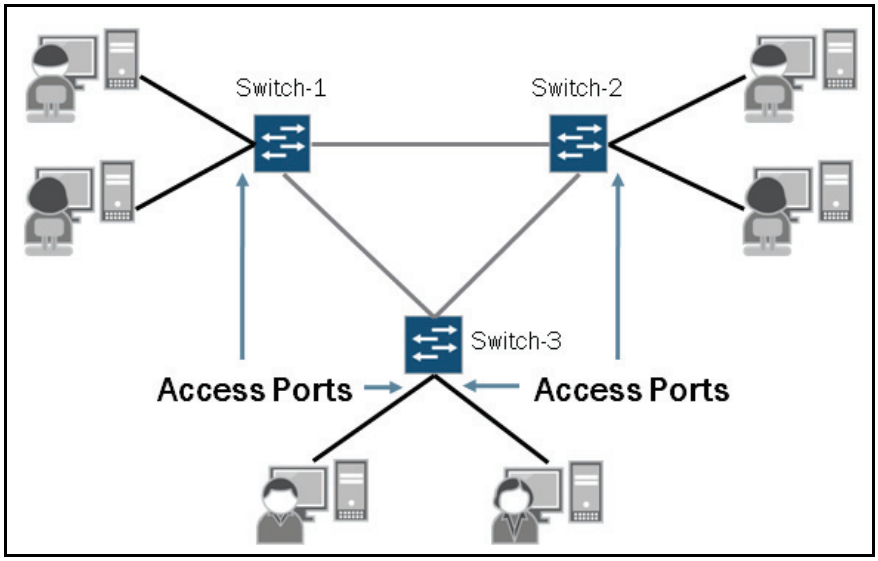
A virtual LAN is a collection of network nodes that are logically grouped together to form separate broadcast domains. A VLAN has the same general attributes as a physical LAN, but it allows all nodes for a particular VLAN to be grouped together, regardless of physical location. One advantage of using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

Layer 2 Switch Port Designations



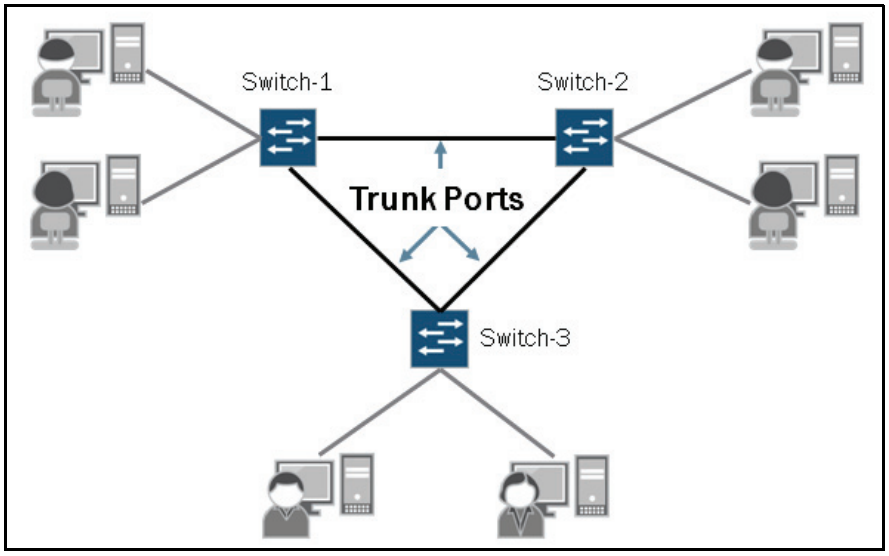
Layer 2 interfaces can be assigned to operate in either access or trunk mode. By default, all installed switch ports on an EX Series switch are configured as access ports. These same switch ports are associated with the default VLAN, which is an untagged VLAN. We discuss the port modes and default VLAN in more detail on subsequent graphics in this chapter.

Access Ports



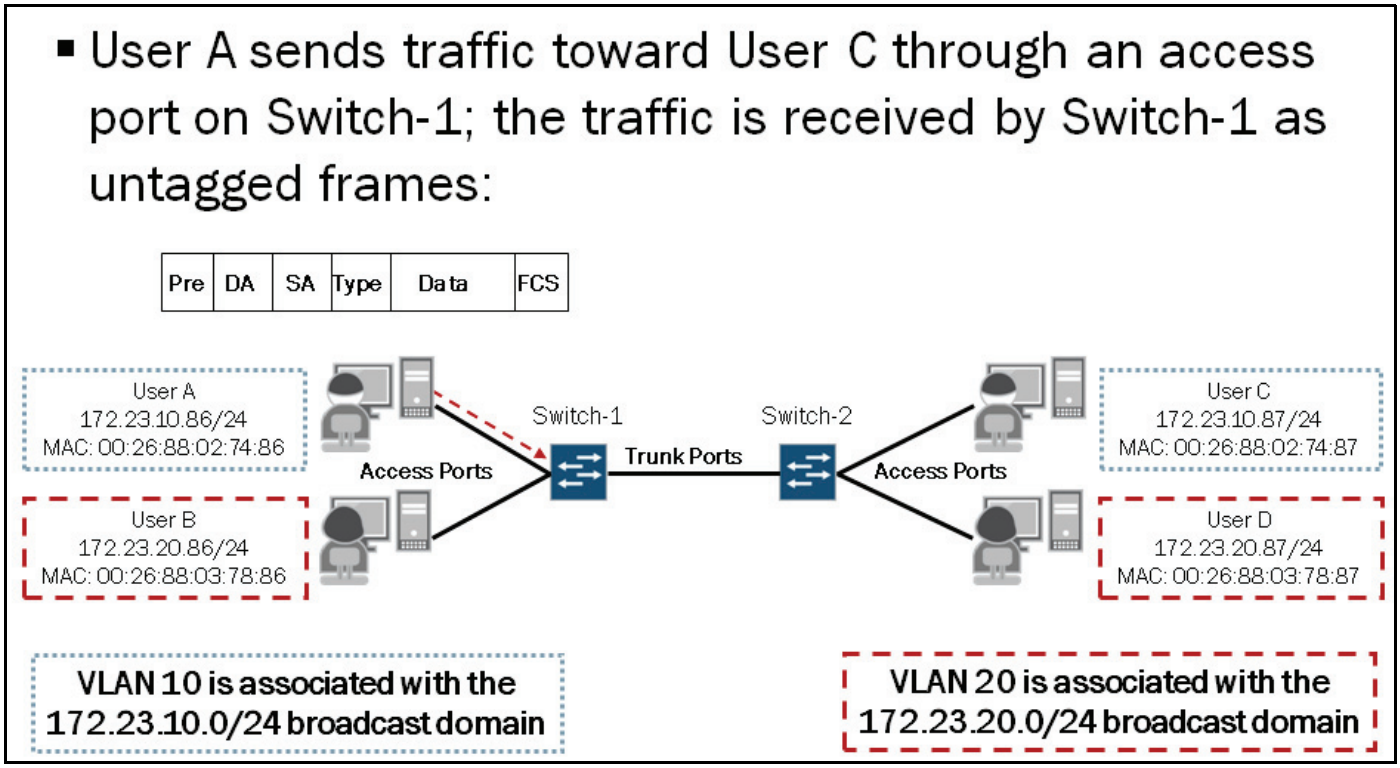
As shown in the illustration on the graphic, access ports typically connect to end-user devices such as computers, IP phones, and printers. Access ports typically belong to a single VLAN and send and receive untagged Ethernet frames. We will discuss the voice VLAN, which is an exception to this operational norm, in a later section in this chapter. All installed switch ports default to access mode in the factory-default configuration and belong to the default VLAN.

Trunk Ports



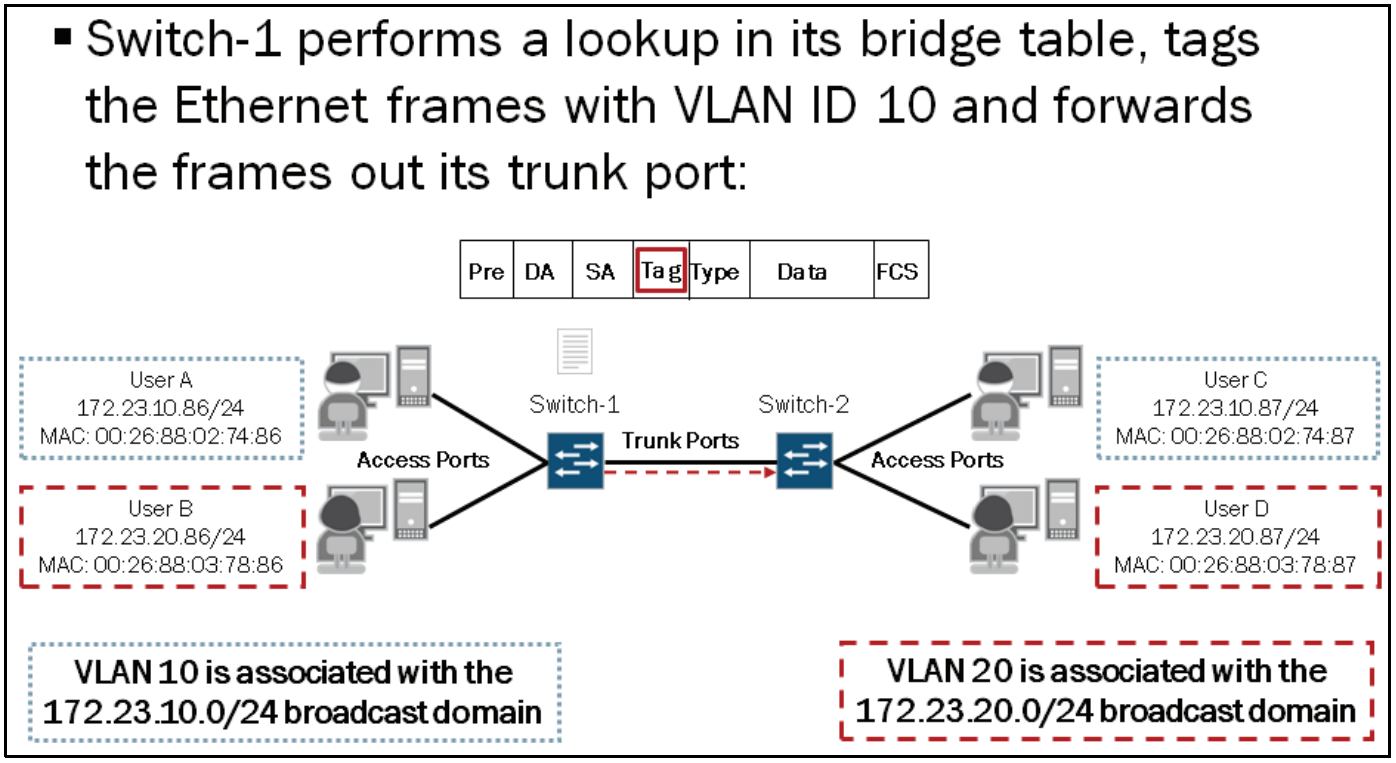
A trunk port typically connects to another switch or to an edge router. Interfaces configured for trunk mode handle traffic for multiple VLANs, multiplexing the traffic for all configured VLANs over the same physical connection, and separating the traffic by tagging it with the appropriate VLAN ID. Trunk ports can also carry untagged traffic when configured with the **`native-vlan-id`** statement. We cover the **`native-vlan-id`** configuration option later in this chapter.

Tagging Traffic Example: Part 1



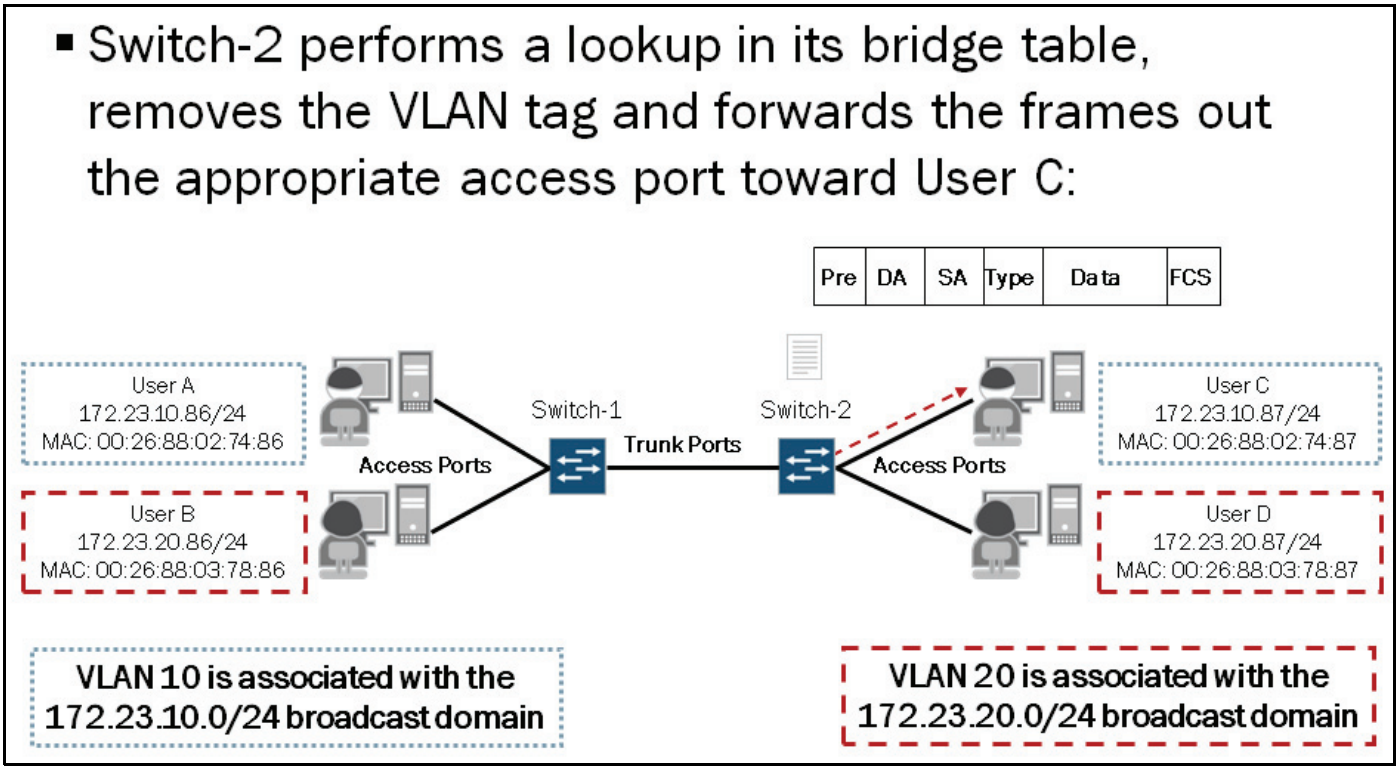
This graphic and the next two graphics illustrate the basic steps involved in sending traffic through a switched network where both access and trunk ports are used. On this graphic we see that User A is sending traffic toward User C through Switch-1 and Switch-2. As the traffic arrives at Switch-1, the frames are untagged. In this example we assume that both Switch-1 and Switch-2 already have the MAC addresses of the end-user devices in their bridge tables.

Tagging Traffic Example: Part 2



Switch-1 examines the source and destination MAC addresses and performs a lookup in its bridge table to determine how the frames should be handled. Switch-1 finds a matching entry for the destination MAC address in its bridge table, tags each Ethernet frame with VLAN-ID 10, and forwards the tagged frames out the appropriate egress interface; the trunk port connected to Switch-2 in this case.

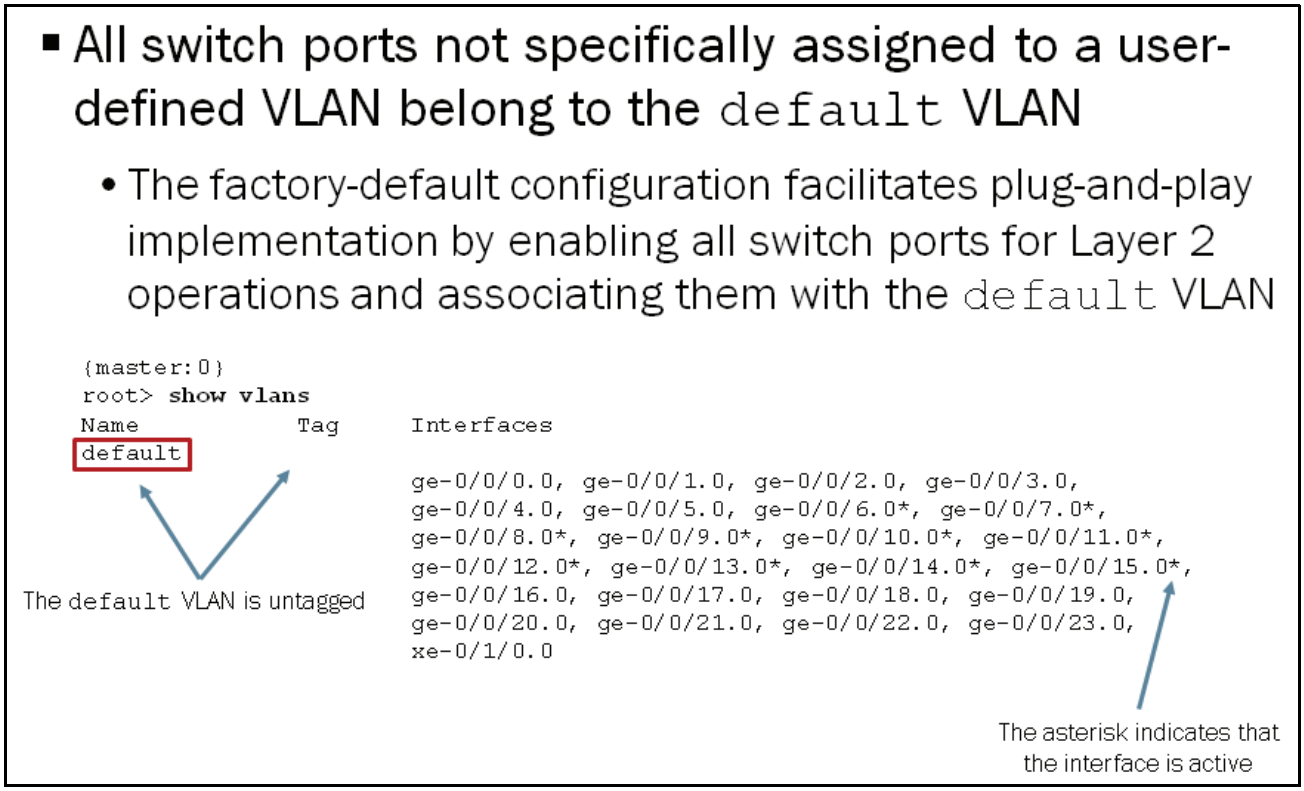
Tagging Traffic Example: Part 3



Once Switch-2 receives the frames, it examines the source and destination MAC addresses and performs a lookup in its bridge table to determine how the frames should be forwarded. Switch-2 finds a matching entry for the destination MAC address,

removes the tag from each Ethernet frame, and forwards the untagged frames out the appropriate egress interface; the access port connected to User C in this case.

Default VLAN



The factory-default configuration associates all installed interfaces with the `default` VLAN. In this sample output shown on the graphic we can see that the `default` VLAN does not use an 802.1Q tag.

Because all installed interfaces are pre-configured for Layer 2 operations and are associated with the `default` VLAN, you can simply insert an EX Series switch in basic single-broadcast domain environments without much or any configuration. If multiple broadcast domains are required within a single switch, you must define additional VLANs.

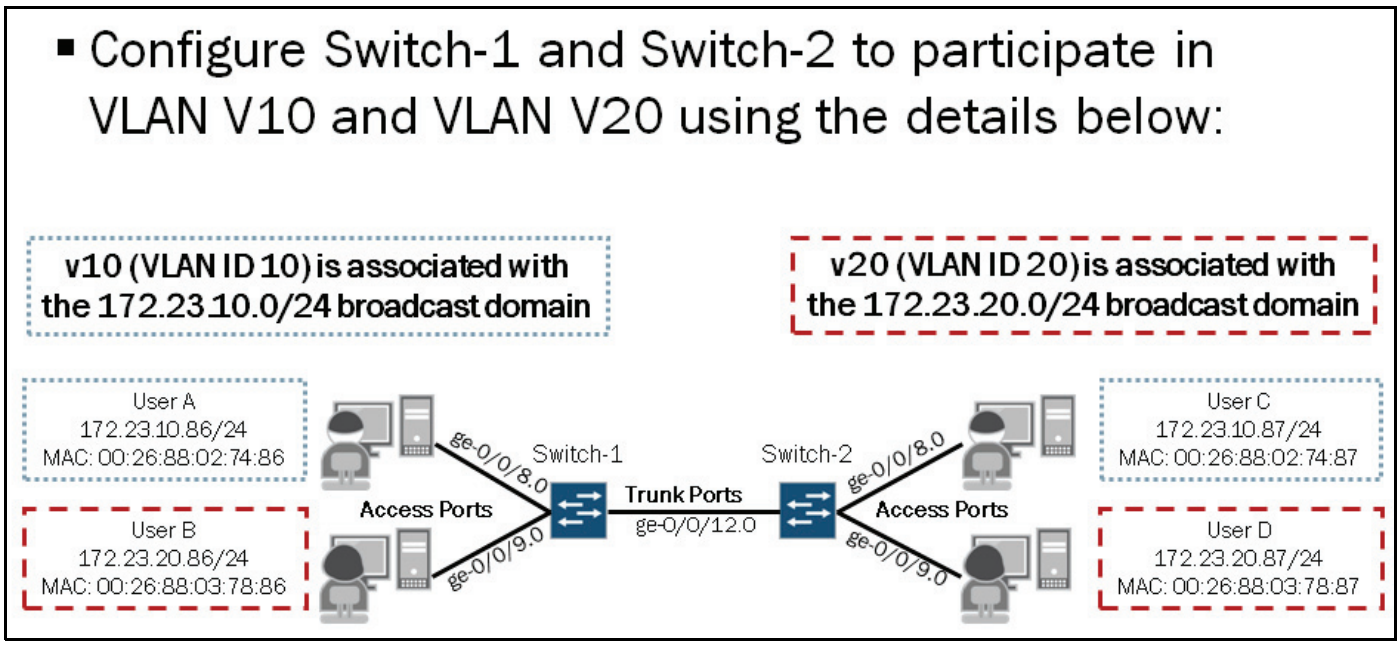
Note that you can make changes to the preconfigured interfaces by manually altering the configuration file directly on the device or automatically by retrieving a configuration file across the network from a Dynamic Host Configuration Protocol (DHCP) server using EZ Touchless Provisioning. This feature is used when you physically connect a switch to the network and boot it with a default configuration. The switch attempts to upgrade software automatically and autoinstall a configuration file from the network. The switch uses information that you configure on a DHCP server to determine whether to perform these actions and where to locate the necessary software image and configuration files on the network. If you do not configure the DHCP server to provide this information, the switch boots with the pre-installed software and default configuration. For more information about this feature, please refer to the technical documentation for your platform and Junos version.

You can manually assign an 802.1Q tag with the default VLAN as shown in the following output:

```
{master:0}[edit]
root# set vlans default vlan-id 100

{master:0}[edit]
root# commit and-quit
configuration check succeedscommit complete
Exiting configuration mode

{master:0}
root> show vlans
```

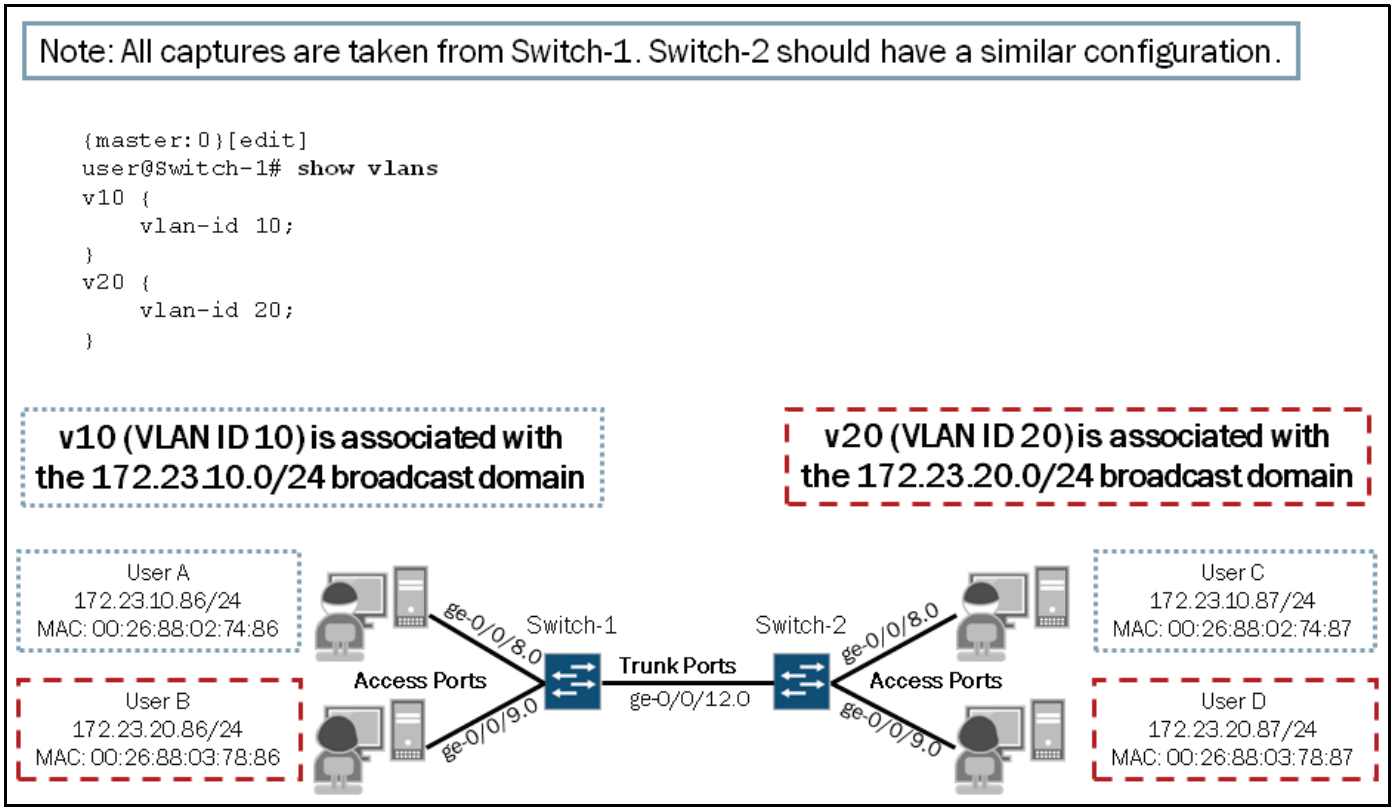
```
Name            Tag     Interfaces
default         100
                        ge-0/0/0.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
                        ge-0/0/4.0, ge-0/0/5.0, ge-0/0/6.0*, ge-0/0/7.0*,
                        ge-0/0/8.0*, ge-0/0/9.0*, ge-0/0/10.0*, ge-0/0/11.0*,
                        ge-0/0/12.0*, ge-0/0/13.0*, ge-0/0/14.0*, ge-0/0/15.0*,
                        ge-0/0/16.0, ge-0/0/17.0, ge-0/0/18.0, ge-0/0/19.0,
                        ge-0/0/20.0, ge-0/0/21.0, ge-0/0/22.0, ge-0/0/23.0,
                        xe-0/1/0.0
```

Case Study: Topology and Objectives



The graphic displays the topology and objectives for our case study.

Configuring VLANs

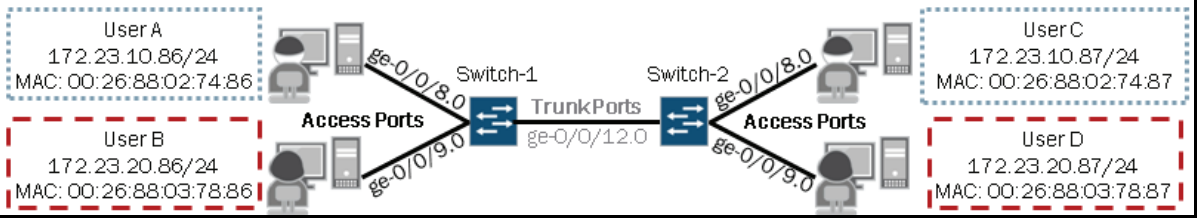


This graphic shows the required VLAN definitions for our case study. Note that additional configuration options are available under the `[edit vlans]` hierarchy level. We cover some of the listed configuration options in subsequent sections and chapters:

```
{master:0}[edit]
user@Switch-1# set vlans v10 ?
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
  description          Text description of the VLAN
> dot1q-tunneling      Dot1q-tunneling parameters
> filter               Packet filtering
> interface            Name of interface that uses this VLAN
  l3-interface         Layer 3 interface for this VLAN
  mac-limit            Number of MAC addresses allowed on this VLAN (1..65535)
  mac-table-aging-time  MAC aging time (60..1000000 seconds)
  no-local-switching   Disable local switching
  no-mac-learning      Disable mac learning
  primary-vlan         Primary VLAN for this community VLAN
  vlan-id              802.1q tag (1..4094)
  vlan-range           VLAN range in the form '<vlan-id-low>-<vlan-id-high>'
  |                    Pipe through a command
```

Configuring Access Ports

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/8
unit 0 {
    family ethernet-switching {
        port-mode access;
        vlan {
            members v10;
        }
    }
}

{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/9
unit 0 {
    family ethernet-switching {
        port-mode access;
        vlan {
            members v20;
        }
    }
}
```



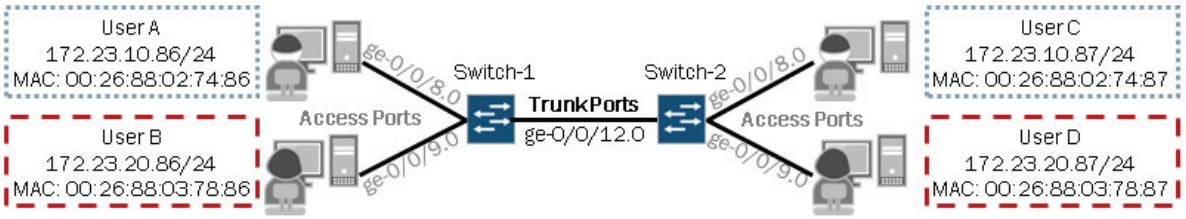
The sample configuration shown on the graphic illustrates one method you can use to associate an interface with a VLAN. Note that the illustrated method is the same method used by the J-Web user interface. Because Layer 2 interfaces default to access mode, including the **`port-mode access`** statement is not strictly required. You can also associate interfaces with VLANs under the `[edit vlans]` hierarchy as shown in the following capture:

```
{master:0}[edit vlans]
user@Switch-1# show
v10 {
    vlan-id 10;
    interface {
        ge-0/0/8.0;
    }
}
v20 {
    vlan-id 20;
    interface {
        ge-0/0/9.0;
    }
}
```

Both methods accomplish the same task. We recommend you use a consistent method when associating interfaces with VLANs to avoid configuration errors and confusion.

Configuring Trunk Ports

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/12
unit 0 {
    family ethernet-switching {
        port-mode trunk;
        vlan {
            members [ v10 v20 ];
        }
    }
}
```



This graphic shows the configuration required for the trunk ports on Switch-1 and Switch-2. Here you can see the **trunk** port-mode option in use and both of the defined VLANs assigned to this interface.

Optionally, you can use the keyword **all** to associate all configured VLANs with a given trunk port. The following example accomplishes the same goal as the configuration shown on the graphic:

```
{master:0}[edit interfaces ge-0/0/12]
user@Switch-1# show
unit 0 {
    family ethernet-switching {
        port-mode trunk;
        vlan {
            members all;
        }
    }
}
```

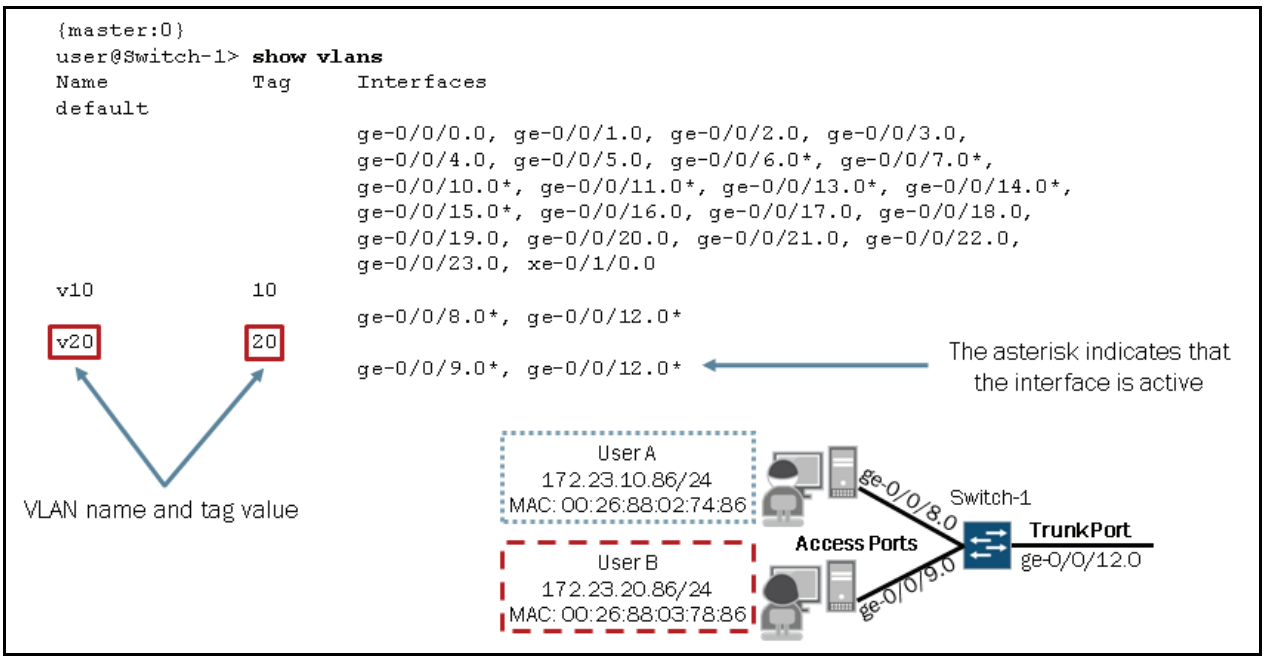
As noted earlier, you can optionally associate interfaces with VLANs under the `[edit vlans]` hierarchy. The following configuration shows this alternative method for a trunk port.

```
{master:0}[edit vlans]
user@Switch-1# show
v10 {
    vlan-id 10;
    interface {
        ge-0/0/12.0;
    }
}
v20 {
    vlan-id 20;
    interface {
        ge-0/0/12.0;
    }
}
```

Because Layer 2 interfaces default to the access port-mode, you must specify the **trunk** port-mode option for trunk interfaces regardless of the configuration method you choose. If you omit the **port-mode trunk** statement or attempt to associate an access interface with multiple standard VLANs, you will see the following error when attempting to activate the configuration:

```
{master:0}[edit interfaces ge-0/0/12]
user@Switch-1# show
unit 0 {
    family ethernet-switching {
        vlan {
            members [ v10 v20 ];
        }
    }
}

{master:0}[edit interfaces ge-0/0/12]
user@Switch-1# commit
error: Access interface <ge-0/0/12.0> has more than one vlan member: <v20> and <v10>
error: configuration check-out failed
```

Verifying VLAN Assignments

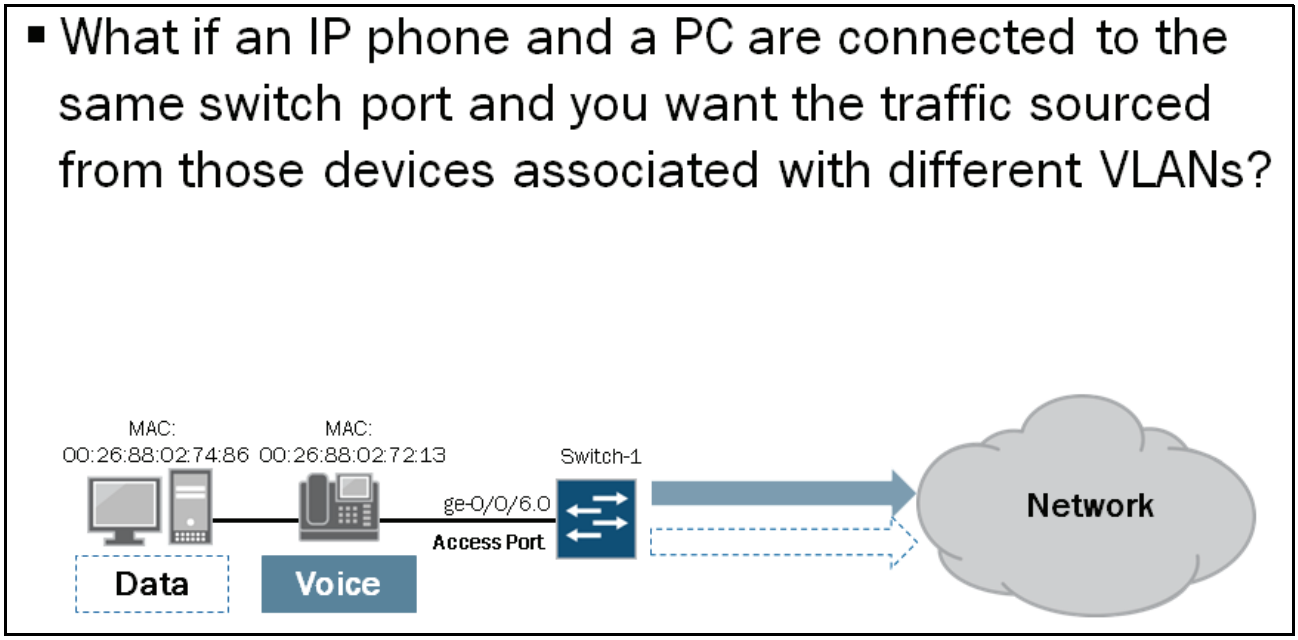


You can use the **show vlans** command to verify VLAN assignments and other details. Optionally you can filter the output or increase the amount of detail generated by adding options to the **show vlans** command. The available options are shown in the following output:

```
{master:0}
user@Switch-1> show vlans ?
Possible completions:
  <[Enter]>            Execute this command
  <vlan-name>          Show information for a particular VLAN
  brief                Display brief output
  default
  detail               Display detailed output
  dot1q-tunneling      Show dot1q-tunneling vlan information
  extensive            Display extensive output
  management-vlan      Show management vlan information
  sort-by              Specify display order
```

```
summary              Display summary output
v10
v20
|                    Pipe through a command
```

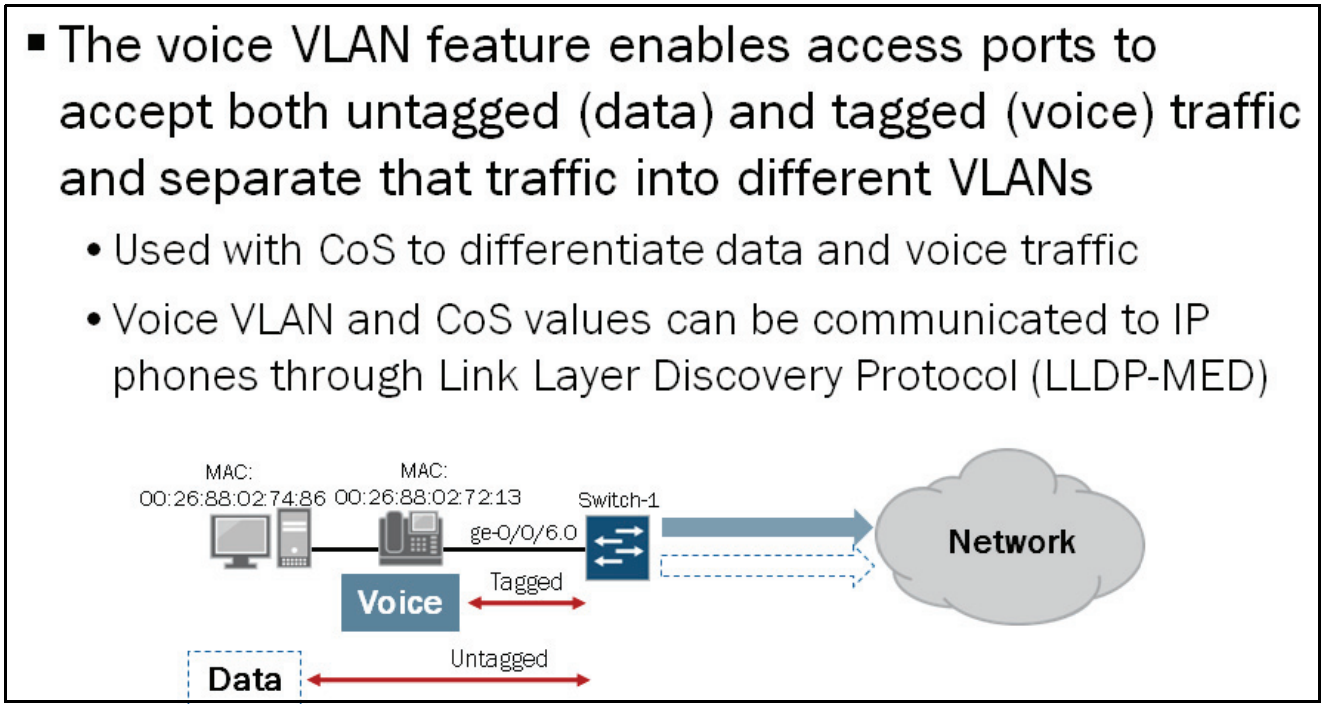
What If...?



This graphic presents a common implementation scenario where two end-user devices, an IP phone and a PC, are connected to a single switch port. In this implementation, it is typically recommended to separate the data and voice traffic so that differing levels of service can be provided by network devices, such as switches and routers, throughout the network.

The next several graphics introduce the voice VLAN configuration option, which can be used to address this exact situation.

Voice VLAN

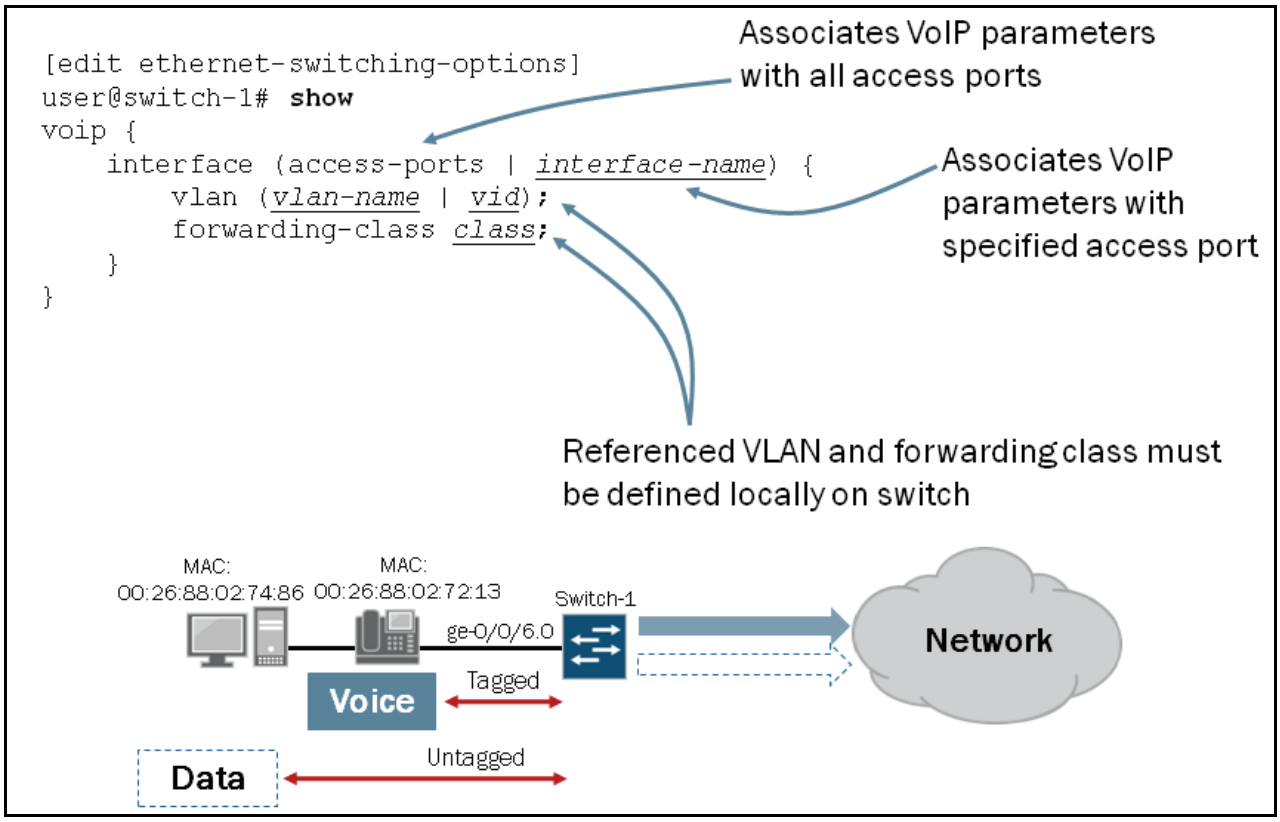


Typically, network administrators choose to treat VoIP traffic differently from user data traffic. To treat these types of traffic differently, you must be able to separate common user data traffic from voice traffic. The voice VLAN feature is used for this

purpose. The voice VLAN enables a single access port to accept untagged data traffic as well as tagged voice traffic and associate each type of traffic with distinct and separate VLANs. By doing this, a network's class-of-service (CoS) implementation can treat voice traffic differently, generally with a higher priority than common user data traffic. CoS is outside the scope of this study guide.

You can use LLDP-MED to dynamically provide the voice VLAN ID and 802.1p values to the attached IP phones. This dynamic method associates each IP phone with the appropriate voice VLAN and assigns the necessary 802.1p values, which are used by CoS, to differentiate service for voice traffic within a network. Note that LLDP-MED is not strictly necessary to associate the voice VLAN ID and 802.1p values with an IP phone. With most vendors, you can manually assign these values to the IP phone directly without the use of LLDP-MED. LLDP-MED is outside the scope of this study guide.

Voice VLAN Configuration: Part 1

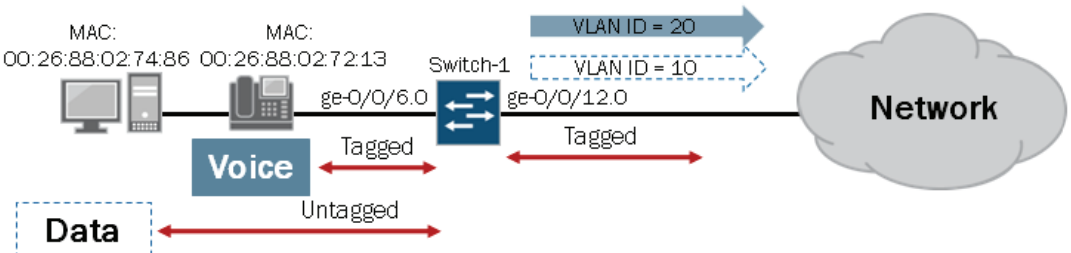


This graphic illustrates the basic hierarchy structure along with the available configuration options associated with the voice VLAN feature.

Voice VLAN Configuration: Part 2

```
{master:0}[edit]
user@Switch-1# show ethernet-switching-options
voip {
    interface ge-0/0/6.0 {
        vlan voice;
        forwarding-class assured-forwarding;
    }
}

{master:0}[edit]
user@Switch-1# show vlans
data {
    vlan-id 10;
}
voice {
    vlan-id 20;
}
```

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/6
unit 0 {
    family ethernet-switching {
        port-mode access;
        vlan {
            members data;
        }
    }
}

{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/12
unit 0 {
    family ethernet-switching {
        port-mode trunk;
        vlan {
            members [ data voice ];
        }
    }
}
```

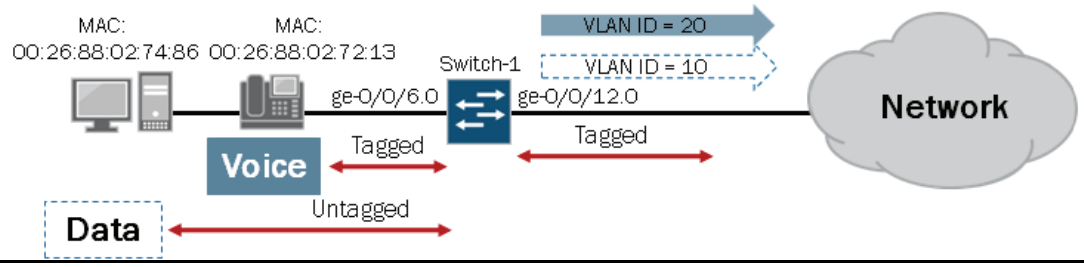


This graphic provides a more complete configuration example based on our sample topology which is also shown on this graphic.

Monitoring the Voice VLAN

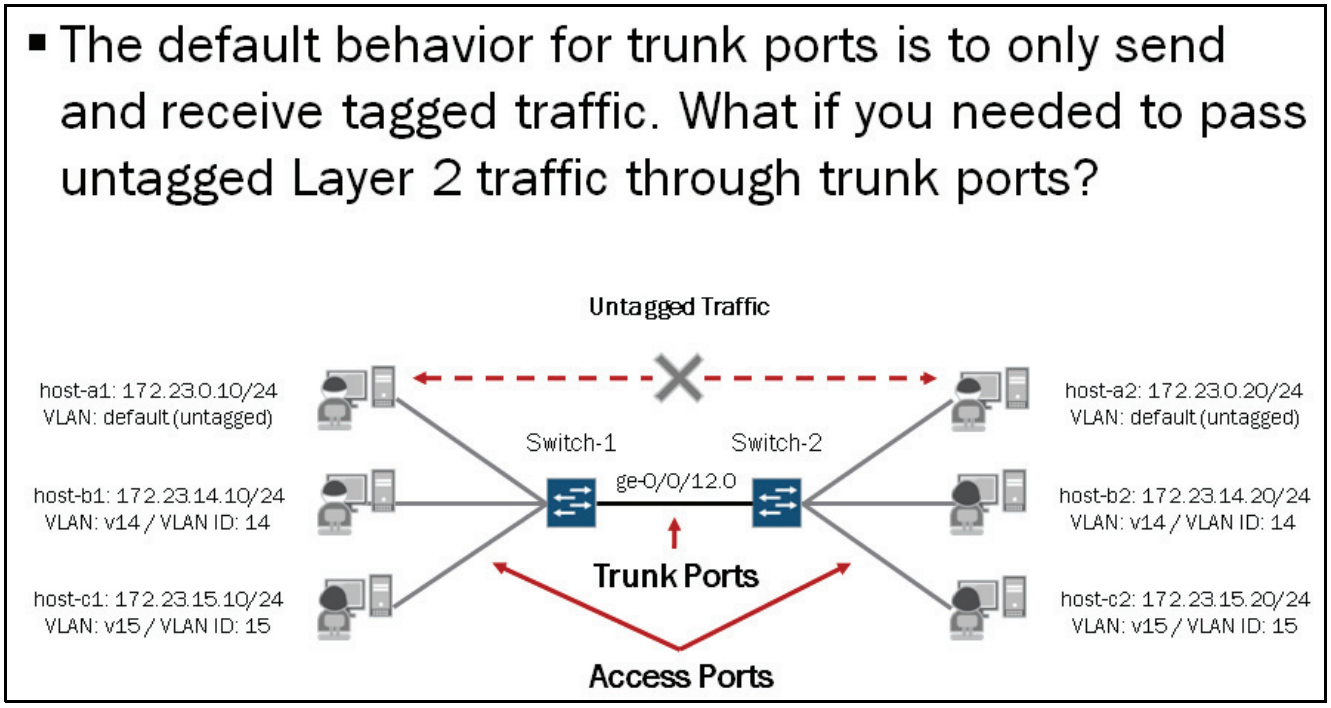
```
{master:0}
user@Switch-1> show vlans sort-by tag
Name            Tag     Interfaces
default
                        ge-0/0/0.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
                        ge-0/0/4.0, ge-0/0/5.0, ge-0/0/7.0*, ge-0/0/8.0*,
                        ge-0/0/9.0*, ge-0/0/10.0*, ge-0/0/11.0*, ge-0/0/13.0*,
                        ge-0/0/14.0*, ge-0/0/15.0*, ge-0/0/16.0, ge-0/0/17.0,
                        ge-0/0/18.0, ge-0/0/19.0, ge-0/0/20.0, ge-0/0/21.0,
                        ge-0/0/22.0, ge-0/0/23.0, xe-0/1/0.0
data            10
                        ge-0/0/6.0*,  ge-0/0/12.0*
voice           20
                        ge-0/0/6.0*,  ge-0/0/12.0*
```

Interface is associated with the `data` and `voice` VLANs



This graphic illustrates the expected output based on our sample configuration shown on the previous graphic. Here you can see that the access port (ge-0/0/6.0) is associated with the `data` and `voice` VLANs.

What If...?

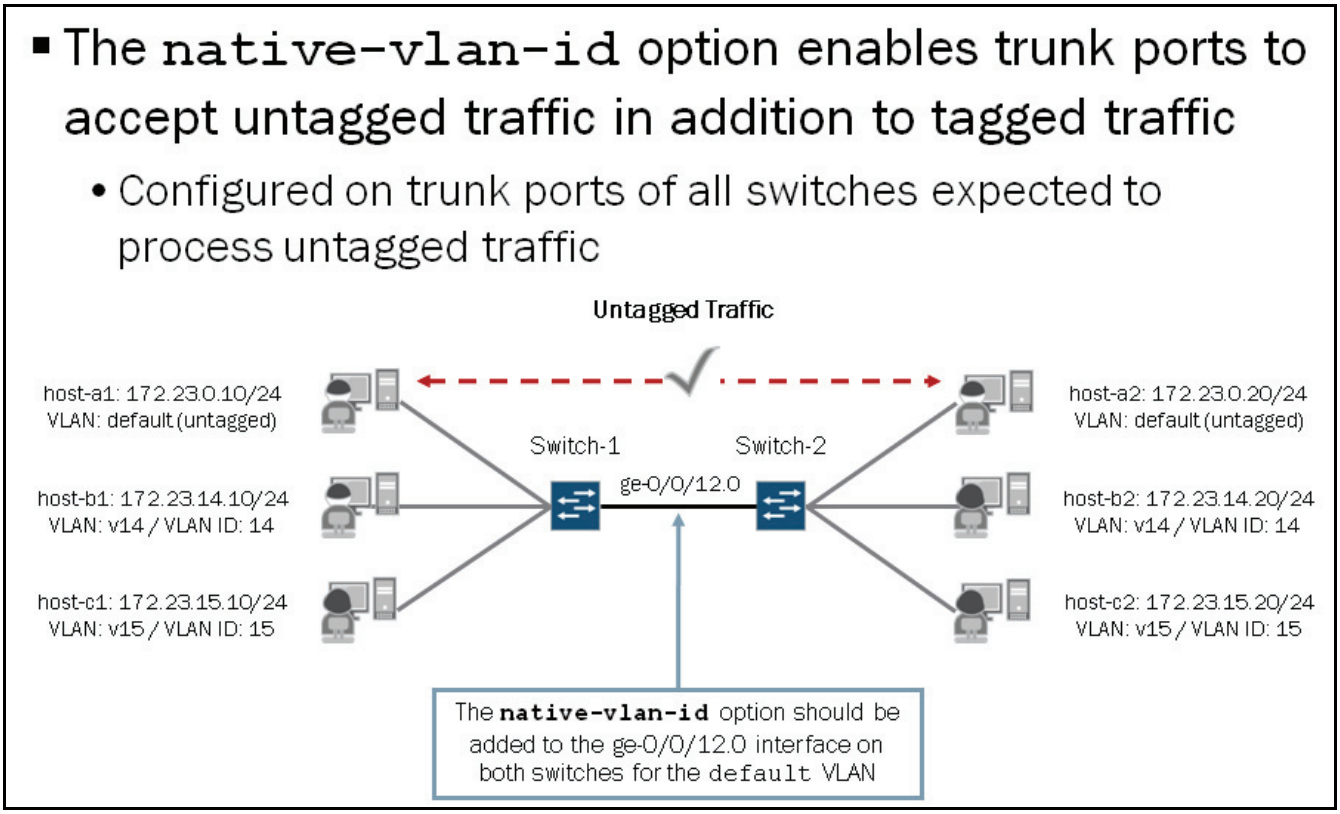


The default behavior on EX Series switches for trunk ports is to only send and receive tagged traffic. This means that you cannot assign an untagged VLAN, such as the default VLAN, to a trunk port. The configuration will not commit as shown here:

```
{master:0}[edit]
user@Switch-1# show interfaces ge-0/0/12
unit 0 {
    family ethernet-switching {
        port-mode trunk;
        vlan {
            members [ v14 v15 default ];
        }
    }
}

{master:0}[edit]
user@Switch-1# commit
error: Trunk interface ge-0/0/12.0 should not have a vlan default with tag value 0
error: configuration check-out failed
```

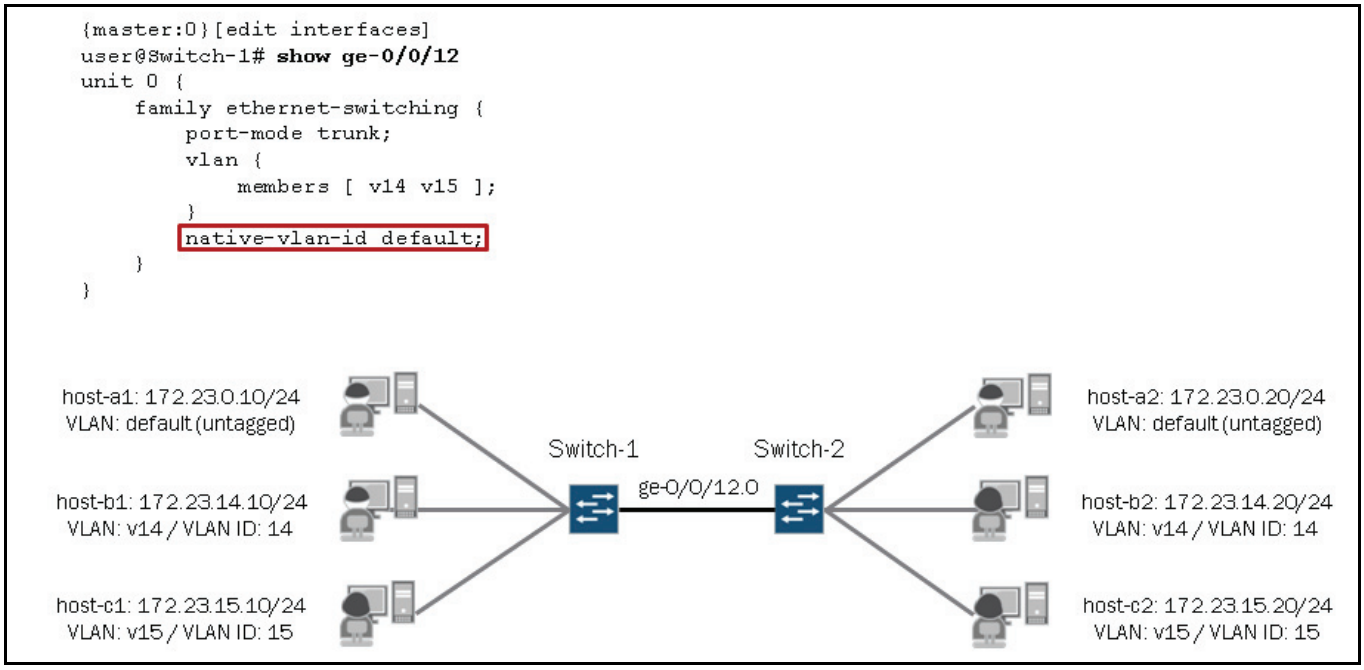
So, what can you do if you needed to pass untagged Layer 2 traffic through trunk ports? You must use the **native-vlan-id** configuration option. We cover the **native-vlan-id** option throughout the remainder of this section.

The `native-vlan-id` Option



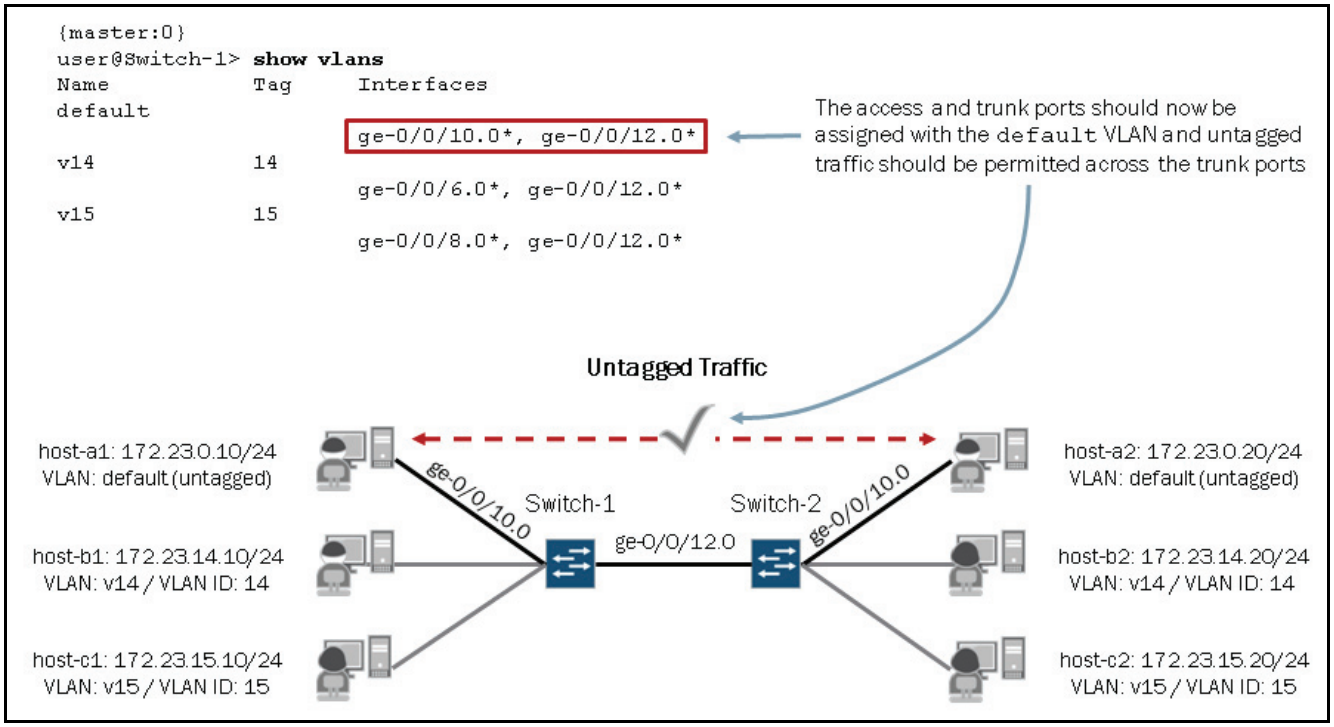
As previously mentioned, a trunk port typically connects one switch to another switch or to an edge router. Interfaces configured for trunk mode handle traffic for multiple VLANs, multiplexing the traffic for all configured VLANs over the same physical connection, and separating the traffic by tagging it with the appropriate VLAN ID. Trunk ports can also carry untagged traffic when configured with the **`native-vlan-id`** configuration option. This option must be enabled on all trunk ports expected to pass untagged traffic. Note that in some vendor's implementation, the native VLAN (also referred to as the default VLAN) is tagged (typically with VLAN-ID 1).

A Configuration Example



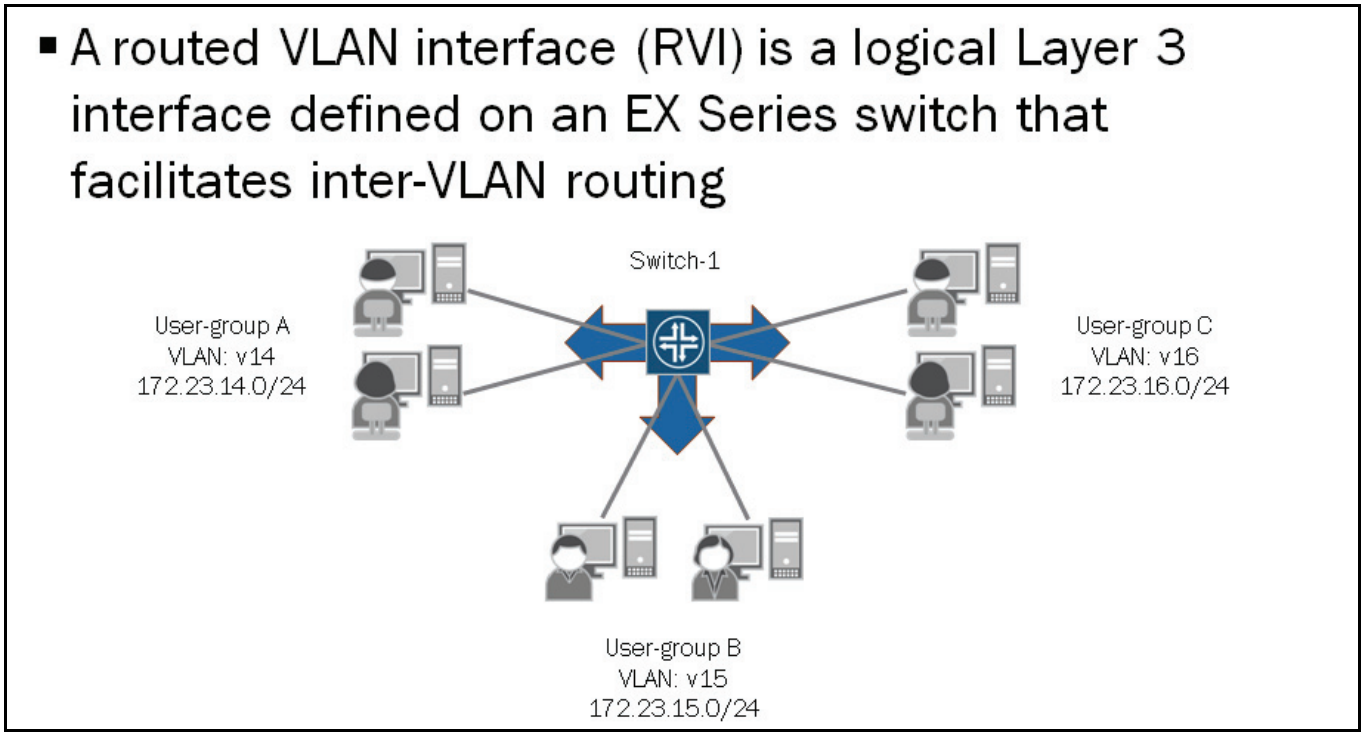
This graphic provides a configuration example using the **native-vlan-id** option for the trunk ports that connect Switch-1 and Switch-2. With this configuration, the ge-0/0/12 interfaces are configured as a trunk ports and are able to carry tagged traffic for the `v14` and `v15` VLANs as well as untagged traffic for the `default` VLAN.

Monitoring the Native VLAN Assignment



This graphic shows the current VLAN assignments on Switch-1. Although not shown on the graphic, Switch-2 has a similar set of VLAN assignments. In this sample output we see that the access port (ge-0/0/10.0) and the trunk port (ge-0/0/12.0) are now associated with the `default` VLAN. With this setup in place, host-a1 and host-a2, should now be able to communicate through the switched network.

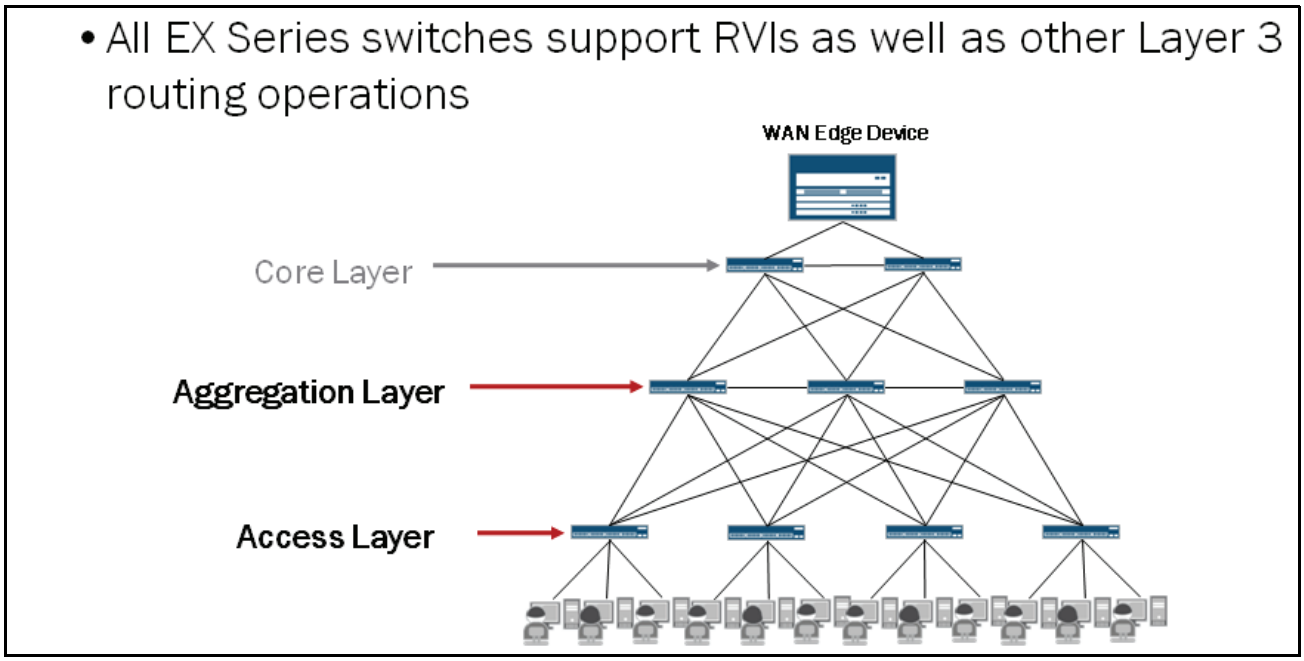
What Is an RVI?



A routed VLAN interface (RVI) is a logical Layer 3 VLAN interface used to route traffic between VLANs. The Layer 3 VLAN interface functions as the gateway IP address for end-user devices on the subnet associated with the corresponding VLAN. Note that proper routing information must exist on the end-user devices, which typically comes in the form of a default gateway.

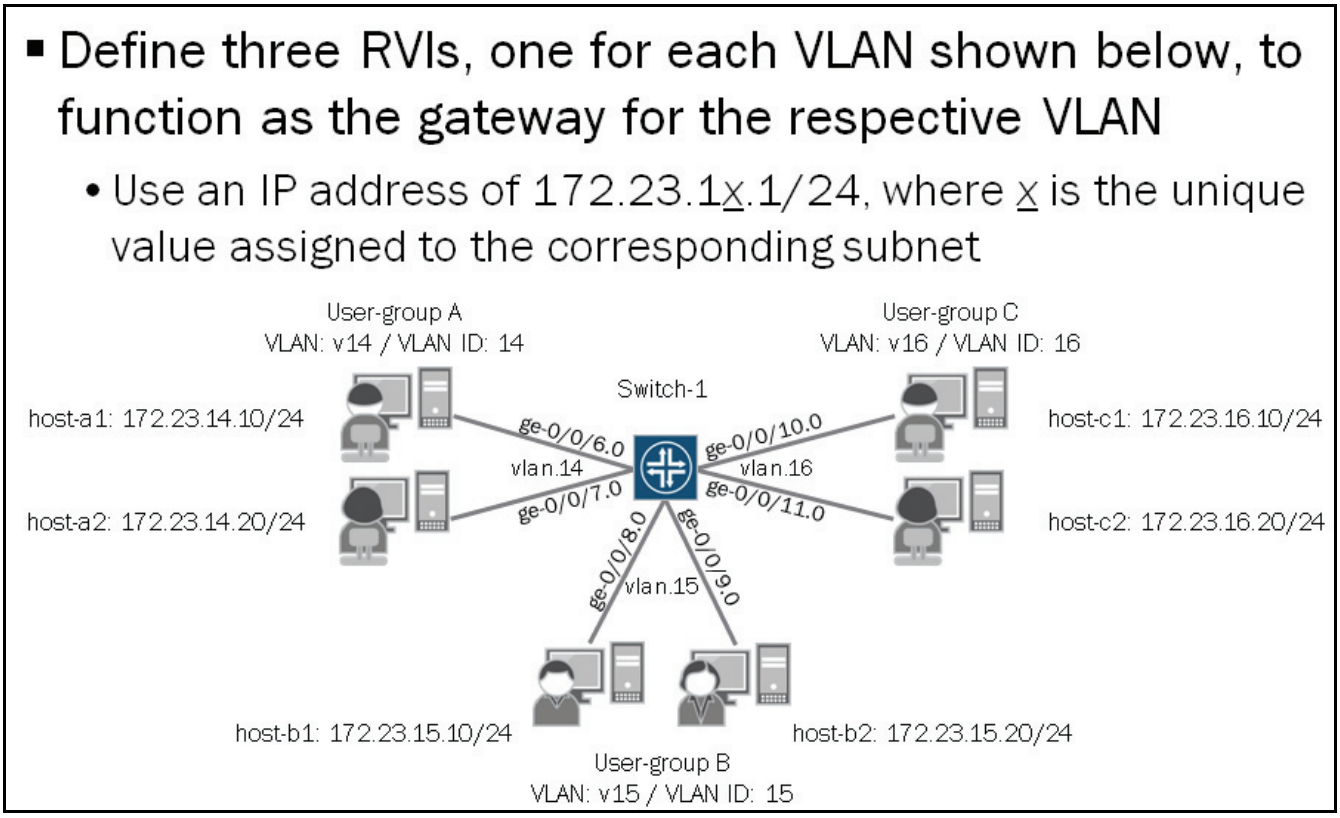
The following graphics provide a configuration and monitoring example for an RVI.

Implementing RVIs



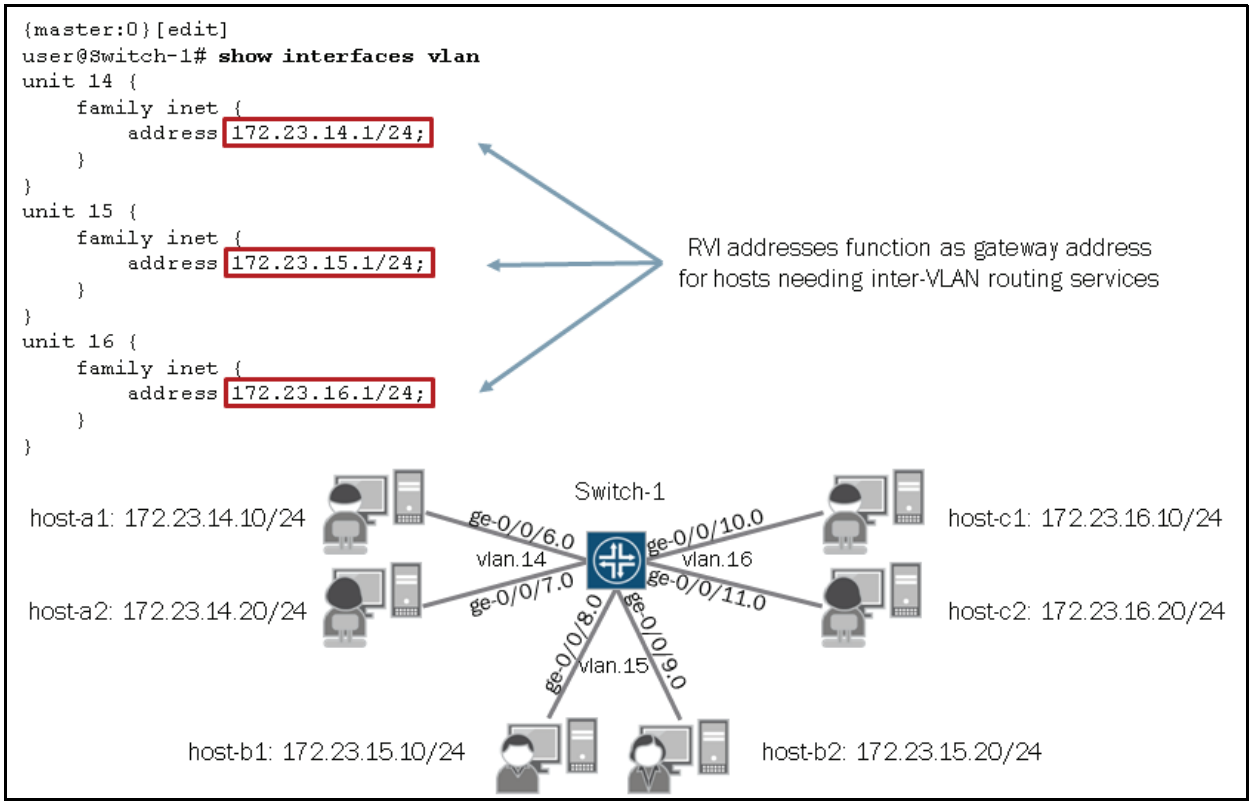
As indicated on the graphic, RVIs are typically implemented in either the aggregation layer or the access layer, depending on the network design and implementation. All EX Series switches support RVIs as well as other Layer 3 routing operations. Check your platform specific documentation for support details.

Case Study: Topology and Objectives



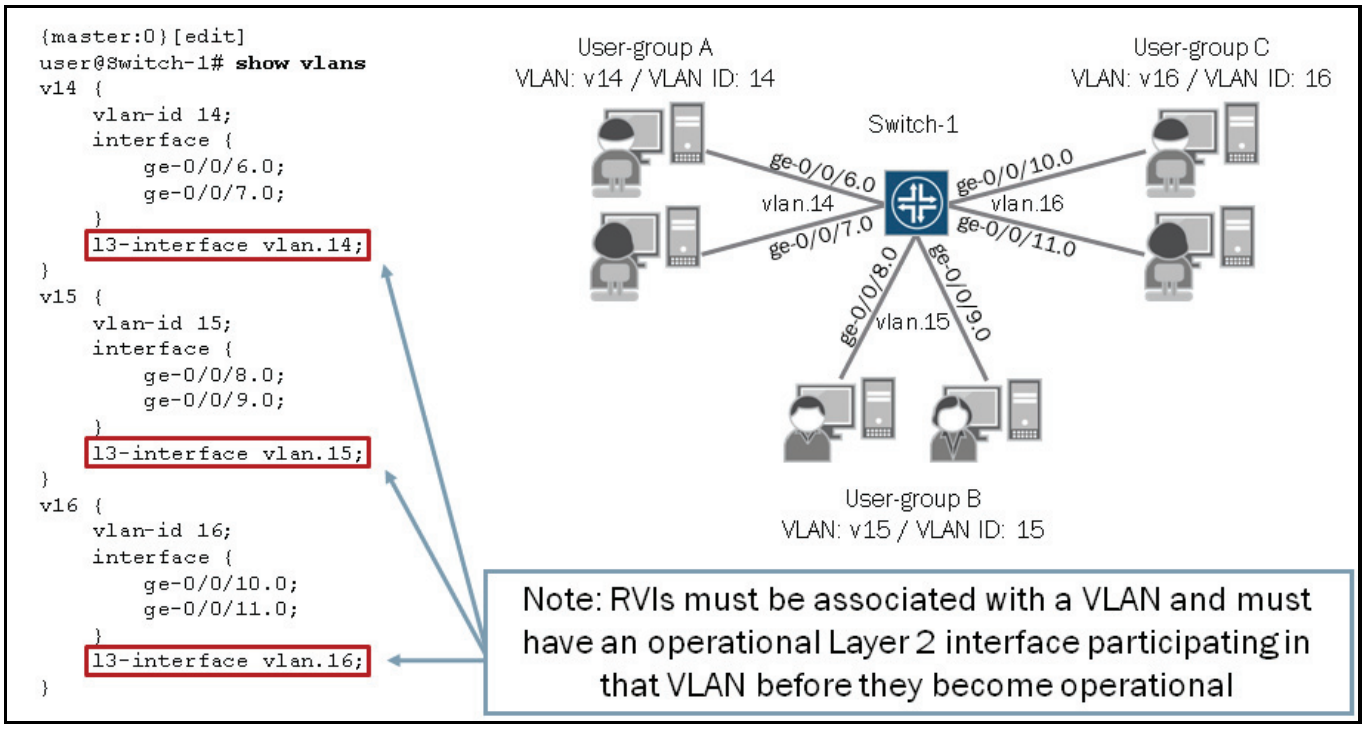
The graphic displays the topology and objectives for our case study.

Configuring RVIs



The graphic shows the RVI configuration required on Switch-1. The vlan.14, vlan.15 and vlan.16 RVIs function as gateways for VLANs v14, v15, and v16 respectively. Although not shown in this example, the access interfaces on Switch-1 that connect to the three VLANs must also be properly configured to permit communications.

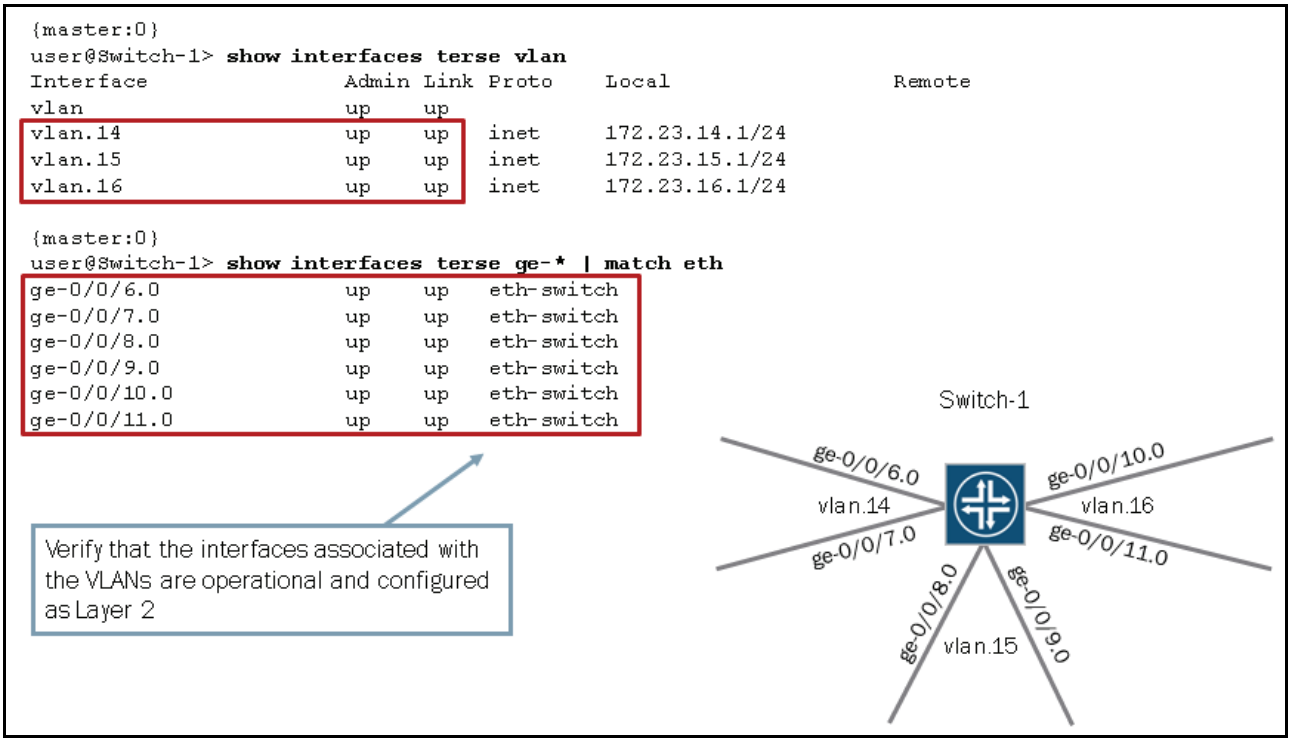
Associating RVIs with VLANs



This graphic shows the association previously defined RVIs with their respective VLANs. This association allows the referenced RVIs to provide Layer 3 services to end-user devices participating on the three VLANs displayed on the graphic. Inter-VLAN

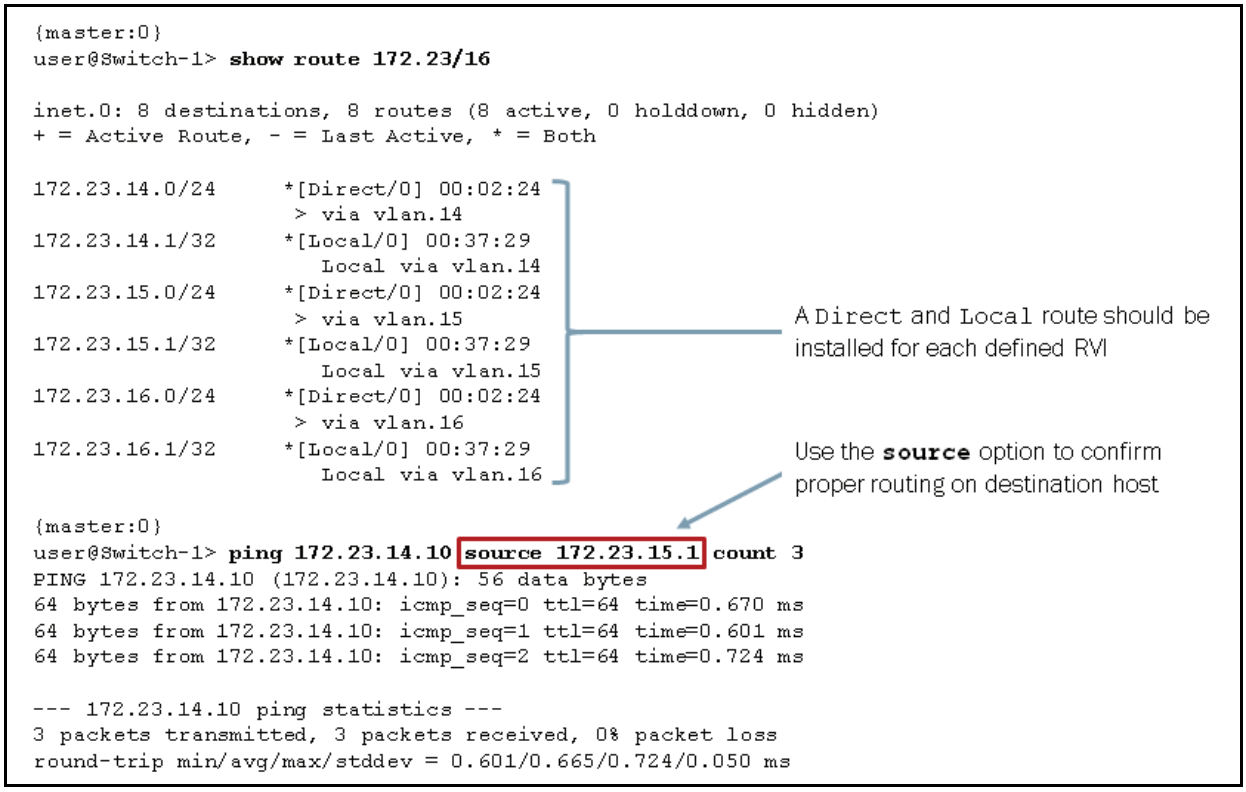
routing cannot occur without this RVI to VLAN association. As mentioned on the graphic, an RVI must be associated with a VLAN and that VLAN must have at least one operational Layer 2 interface before the RVI becomes operational.

Verifying Interface State



This graphic illustrates the commands and a sample output showing the desired interface state for the RVIs and the Layer 2 interfaces associated with the VLANs defined on the previous graphic.

Verifying Routing and Reachability



This graphic shows the command used to verify the proper routing information is present on Switch-1 as well as the command used to test reachability between VLANs.

Review Questions

1. What Layer 2 port modes can be assigned to a switch port? Describe the operations of each.
2. What is the purpose of the voice VLAN?
3. When is the `native-vlan-id` option used?
4. Describe how inter-VLAN routing can be implemented on a switch.

Answers

1.

Switch ports can either be in access or trunk mode. By default, Layer 2 interfaces on EX Series switches are in access mode, which means the connect to end-user devices and pass untagged traffic. You can configure Layer 2 interfaces for trunk mode, which means the interface passes tagged traffic. Switch ports in trunk mode typically connect to other switches or edge routers.

2.

Typically, access ports only relay untagged traffic and are associated with a single VLAN. In some implementations you can have an IP phone and a PC both connected to a single switch port, in a daisy-chained fashion. The voice VLAN feature allows you to associate a data VLAN and a voice VLAN with the same switch port and permits both untagged (data VLAN) and tagged (voice VLAN) traffic to pass through the access port.

3.

The **native-vlan-id** option allows you to associate a specific VLAN with untagged traffic on a specific trunk port. This option is most often used with the default VLAN because the default VLAN's default VLAN ID of 0 is not allowed to appear in the tag field of a tagged packet.

4.

You can use RVIs to implement inter-VLAN routing on an EX Series switch. An RVI is a logical Layer 3 interface and is associated with a specific VLAN. The IP address assigned to an RVI function as the gateway address for end-user devices within a given VLAN.

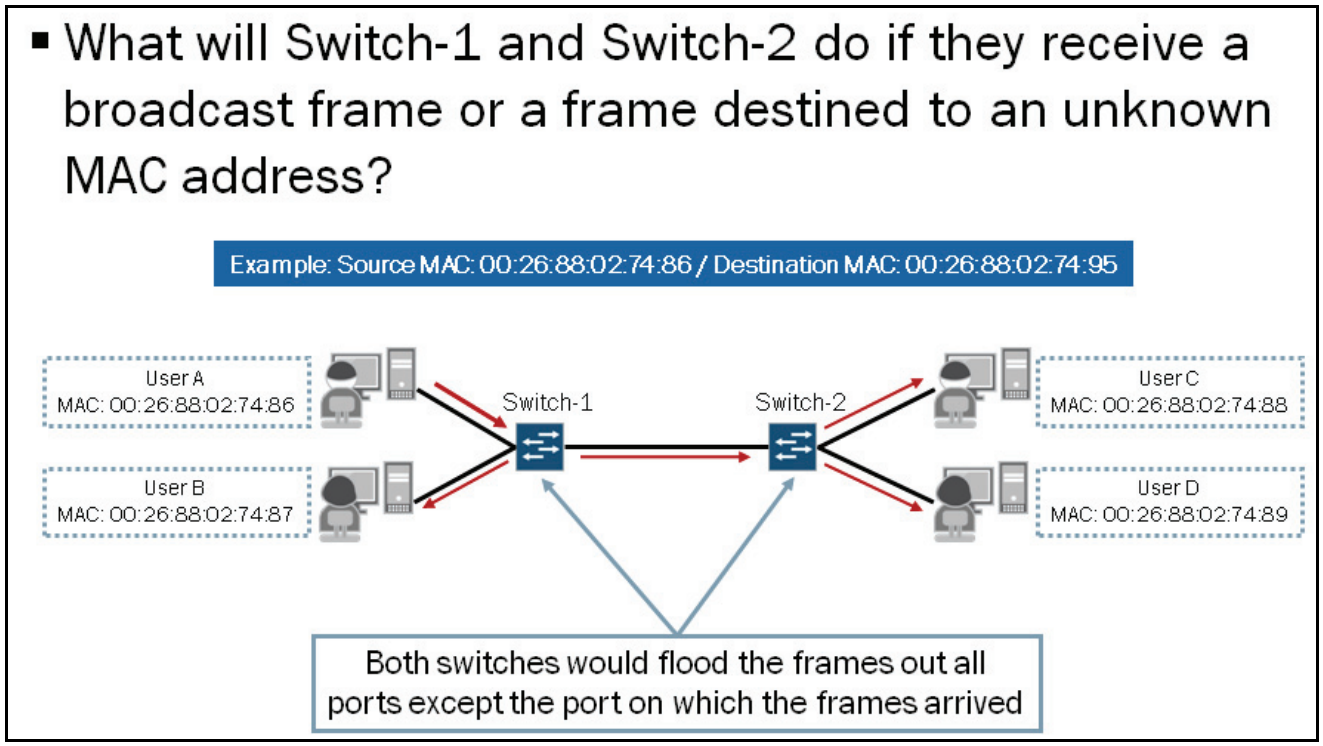
JNCIS-ENT Switching Study Guide

Chapter 3: Spanning Tree

This Chapter Discusses:

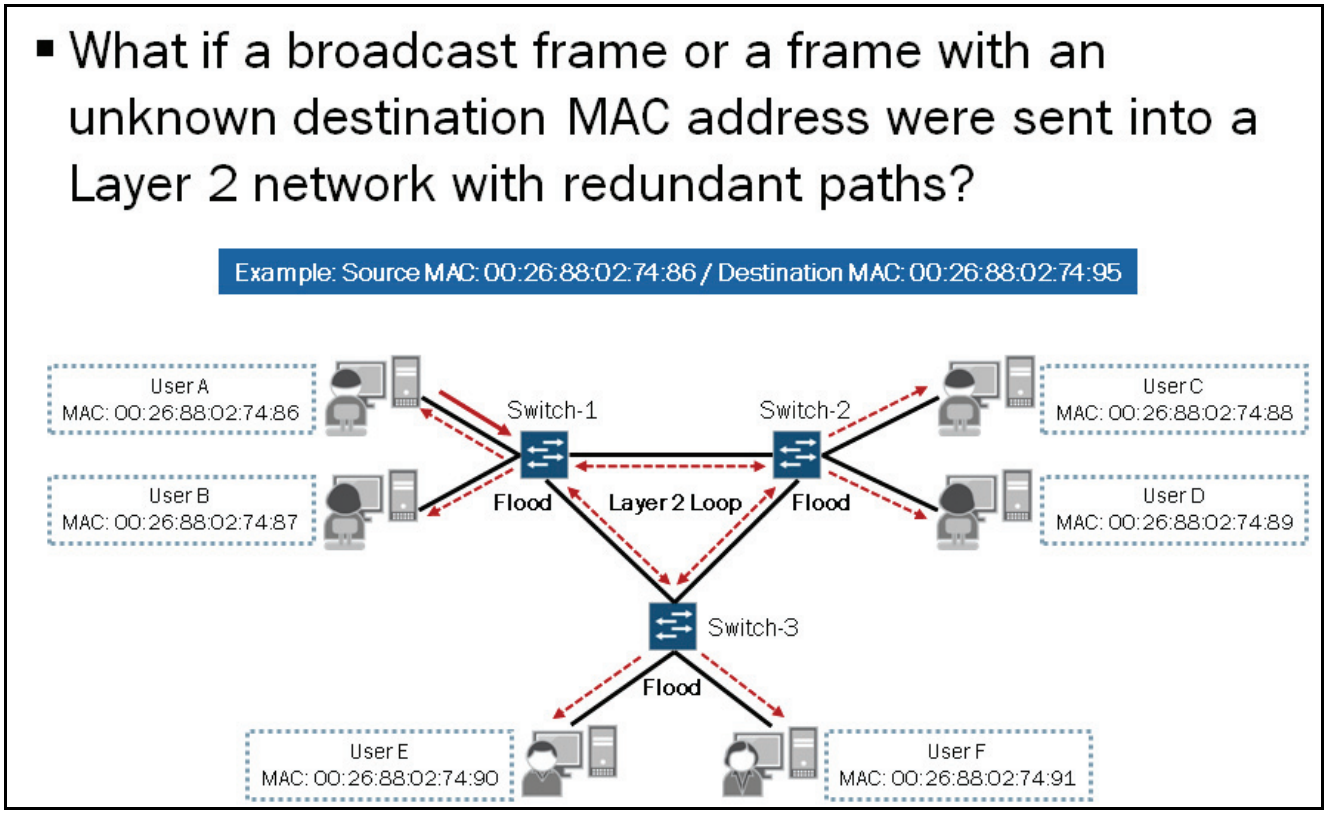
- Instances when a spanning tree is required;
- Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP) operations;
- The advantages of using RSTP over STP;
- The configuration and monitoring of STP and RSTP;
- Bridge protocol data unit (BPDU), loop, and root protection features; and
- The configuration and monitoring of BPDU, loop, and root protection features.

Test Your Knowledge



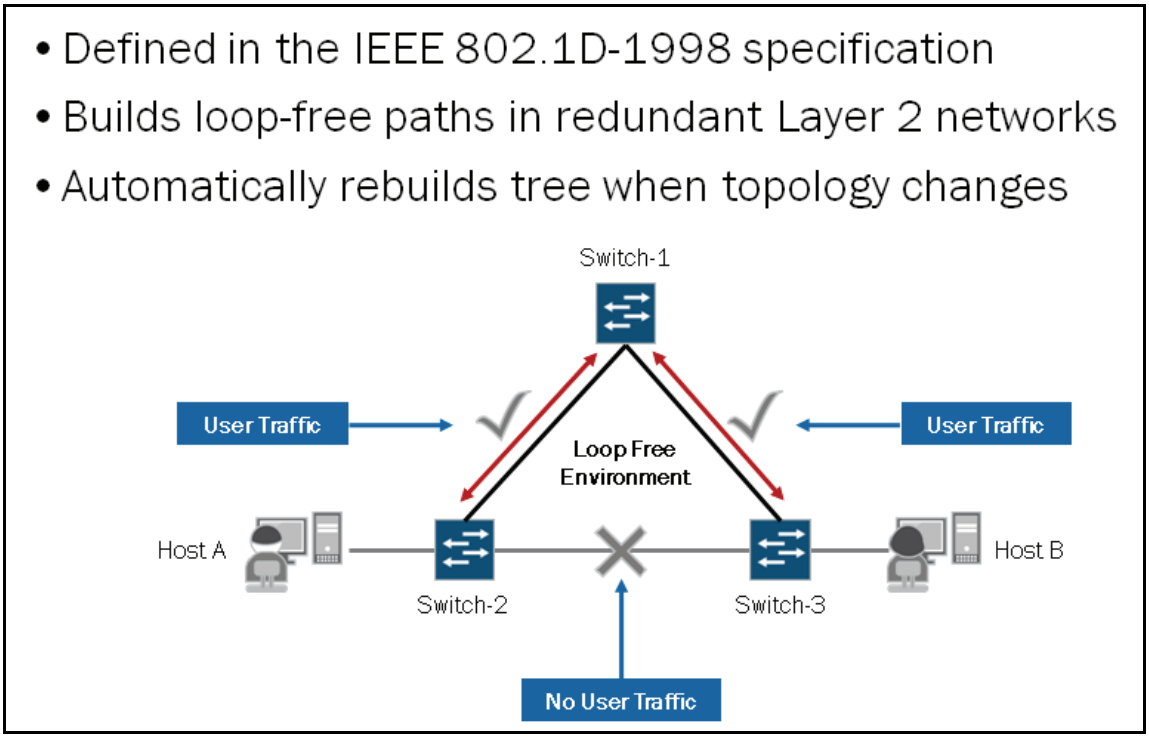
This graphic serves as a review of a previously covered concept. The graphic illustrates the expected behavior when a switch receives a broadcast frame or a frame destined to an unknown MAC address. You can see in the example that both Switch-1 and Switch-2 flood the frame out all interfaces except the interface on which the frame was received. This is an important concept to understand going forward.

What If ...?



As previously mentioned, switches flood broadcast frames and frames for unknown MAC addresses out all ports except the port on which those frames were received. In Layer 2 networks with redundant paths, such as the one illustrated on the graphic, switches will continuously flood these types of frames throughout the network. When a frame is continuously flooded throughout a Layer 2 network, a Layer 2 loop exists. Layer 2 loops can be extremely harmful to a network's operation and should be avoided. To avoid Layer 2 loops, you must implement a Layer 2 loop-prevention mechanism such as the spanning tree protocol (STP). We cover STP on subsequent graphics in this chapter.

STP

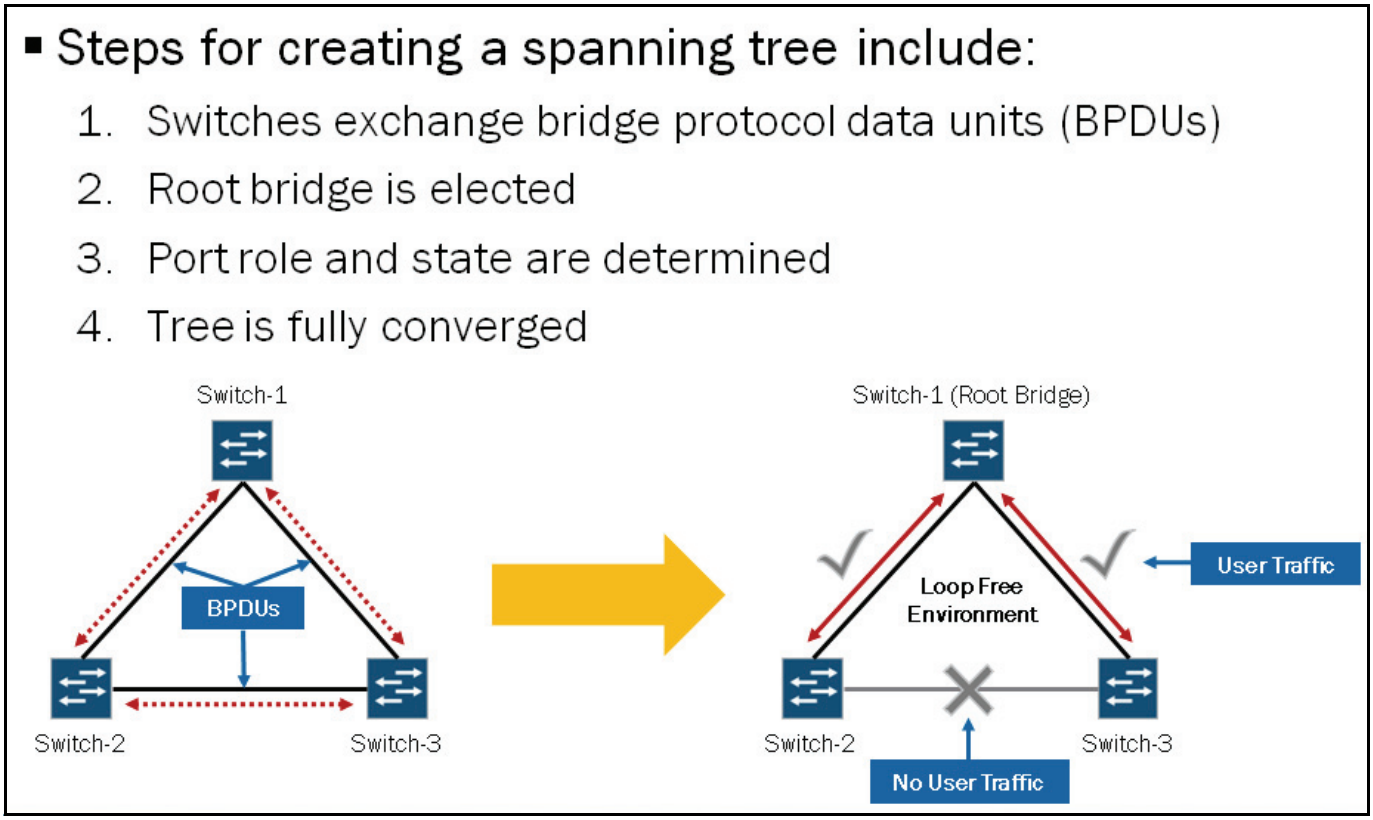


STP is defined in the Institute of Electrical and Electronics Engineers (IEEE) 802.1D 1998 specification. STP is a simple Layer 2 protocol that prevents loops and calculates the best path through a switched network that contains redundant paths. STP is highly recommended in any Layer 2 network environment where redundant paths exist or might exist. When topology changes occur, STP automatically rebuilds the tree.

Note that newer versions of STP exist including Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP), and VLAN Spanning Tree Protocol (VSTP). These newer versions of STP include enhancements over the original STP. We cover the RSTP in detail later in this chapter.

MSTP allows you to run a separate instance of spanning tree for a group of VLANs while VSTP allows you to run one or more spanning tree instances for each VLAN. MSTP and VSTP are outside the scope of this study guide.

How Does it Work?



This graphic highlights the basic steps for creating a spanning tree. We highlight each of these steps in more detail on subsequent graphics.

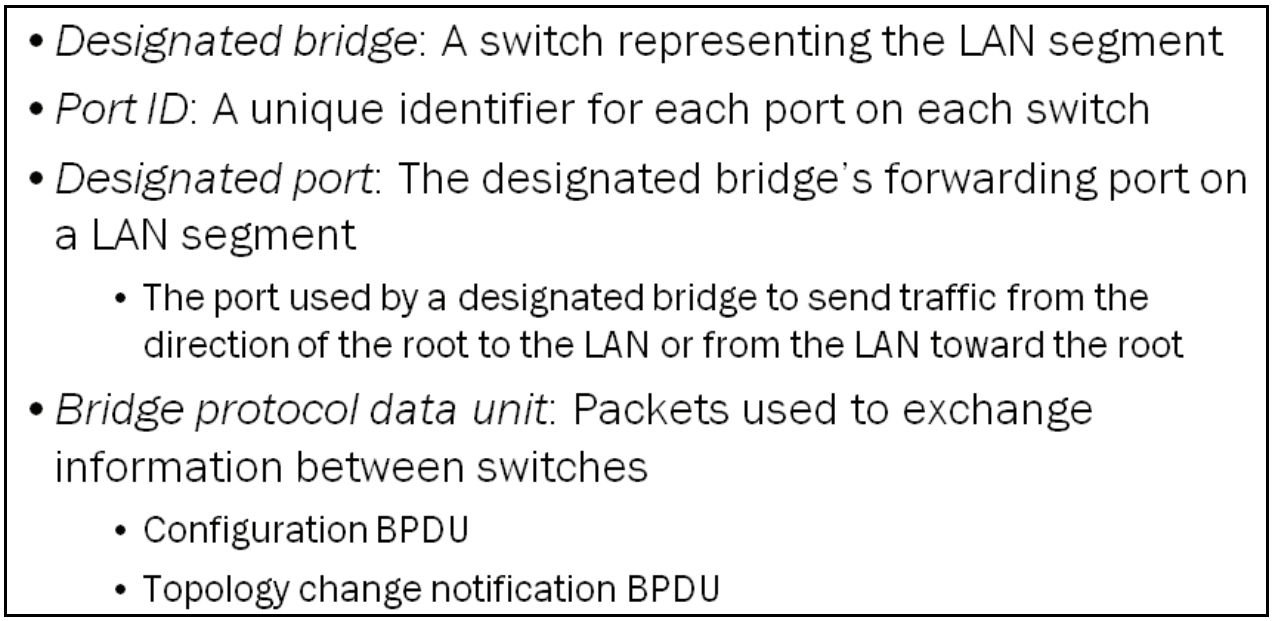
Key Terms and Concepts: Part 1

- *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 (1–200000000)
  - The default value is 20000 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 graphic—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.

Key Terms and Concepts: Part 2



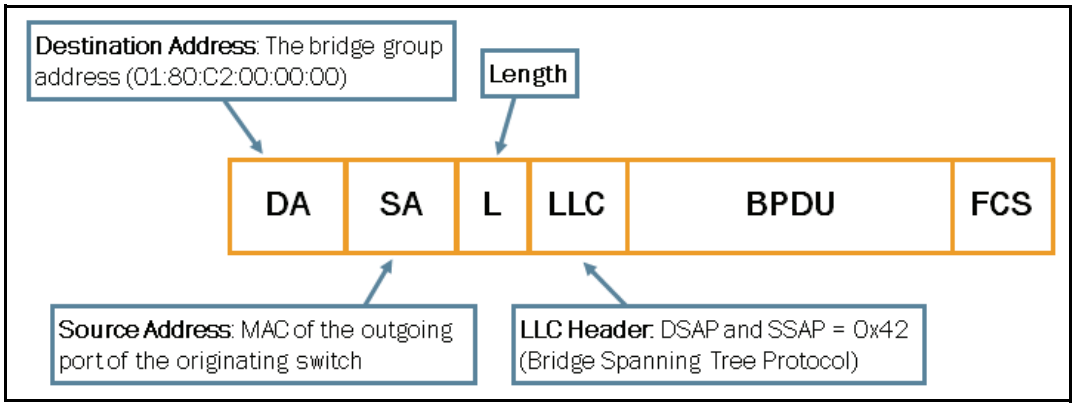
All switches participating on a common network segment must determine which switch offers the least-cost path from the network segment to the root bridge. The switch with the best path becomes the *designated bridge* for the LAN segment, and the port connecting this switch to the network segment becomes 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.

STP Port States

- Each individual port of each bridge can be in one of four 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 graphic, 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



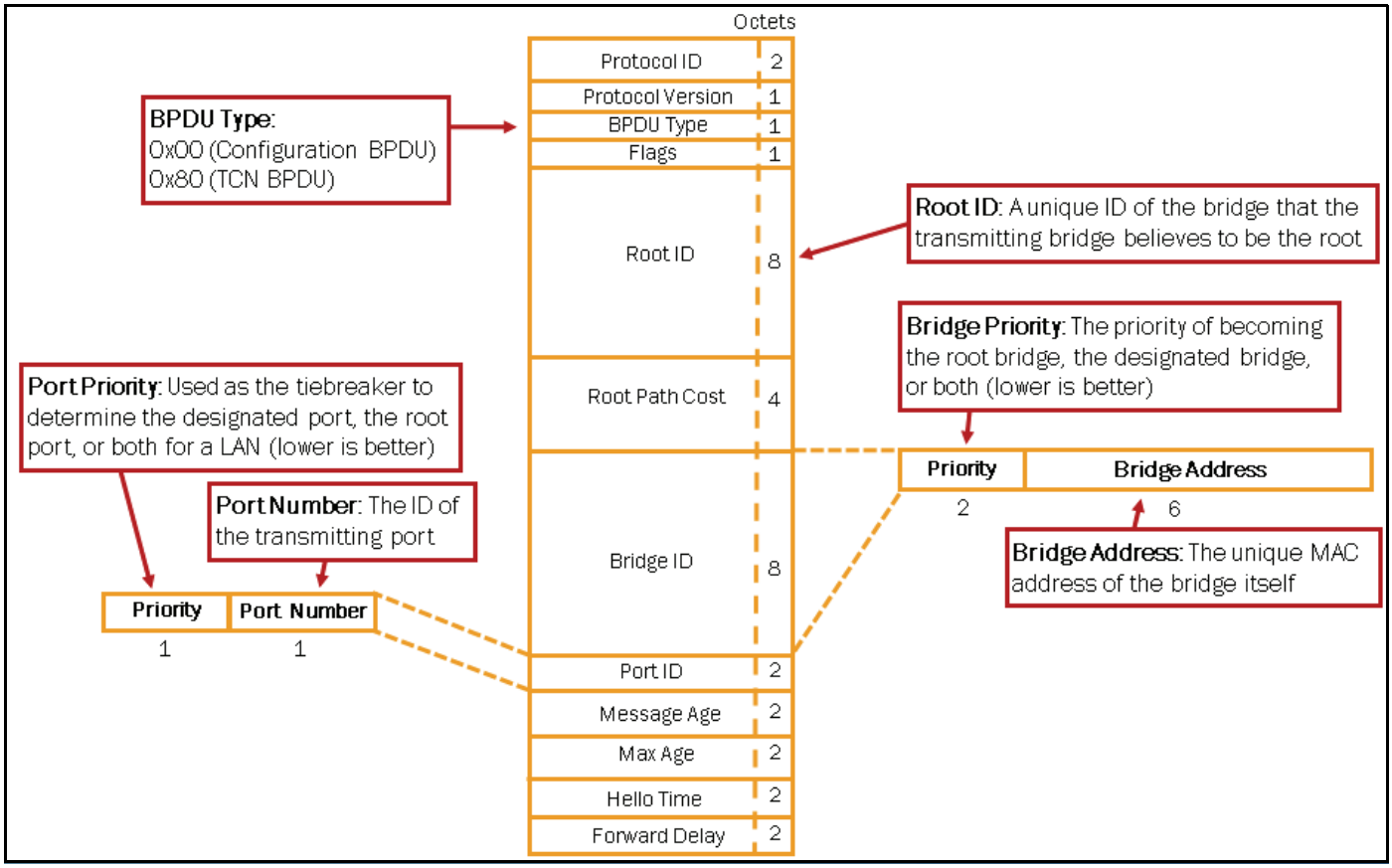
The graphic shows the Ethernet frame format of an STP BPDU. Notice that the Ethernet frame does not contain any 802.1Q-type VLAN tagging. The source address of the frame is the MAC address of the outgoing port of the sending switch. The destination address is the multicast MAC address that is reserved 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

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



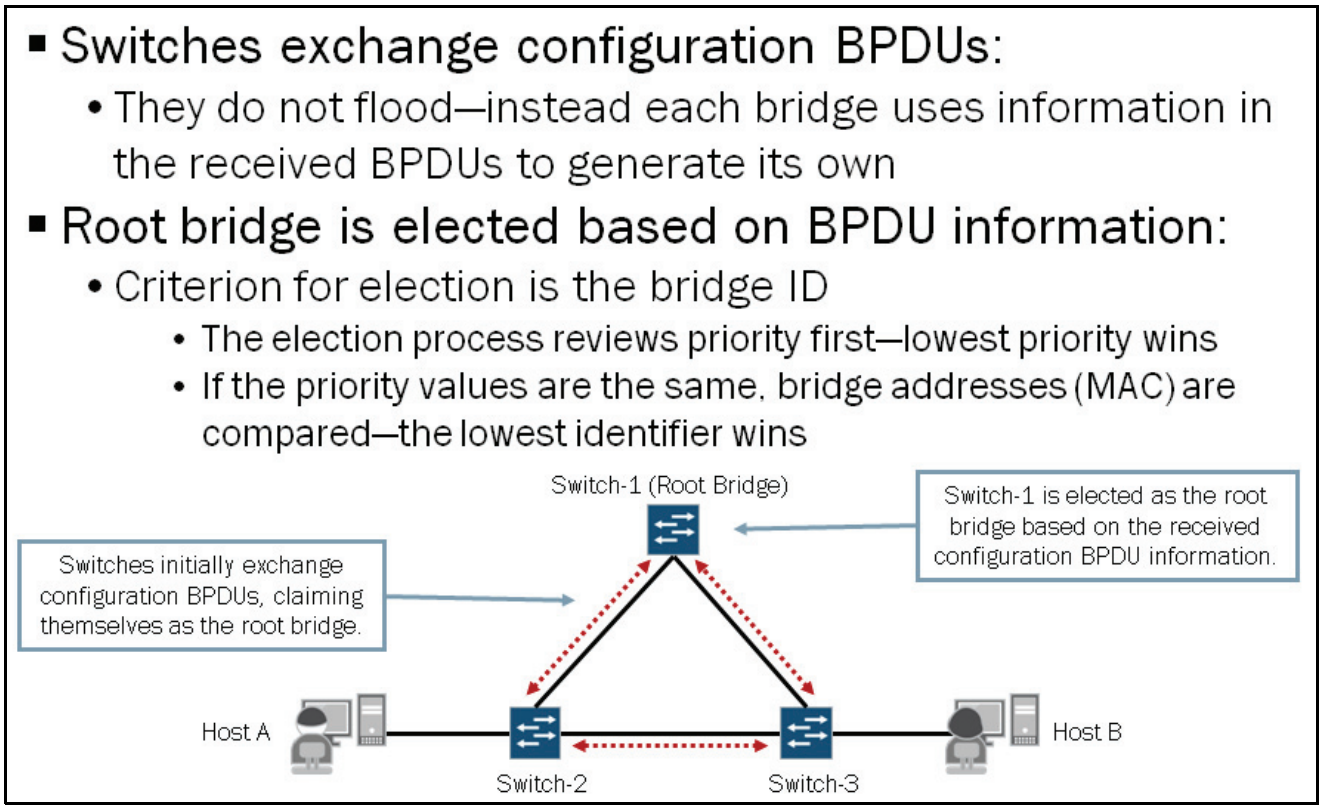
When an STP network is first turned up, all participating bridges send out configuration BPDUs to advertise themselves as candidates for the root bridge. Each bridge uses the received BPDUs to help build the spanning tree and elect the root bridge, root ports, and designated ports for the network. Once 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 (0x00) or TCN BPDU (0x80).
- *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.
- *Bridge ID (BID)*: This value is the identifier 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 EX Series switch. The switch automatically generates the port number and you cannot configure it. For example, ge-1/0/0 contains the value 128:513, whereas ge-1/0/1 contains the value 128:514.

- *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

Exchange of BPDUs



Switches participating in a switched network running STP 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 frame.

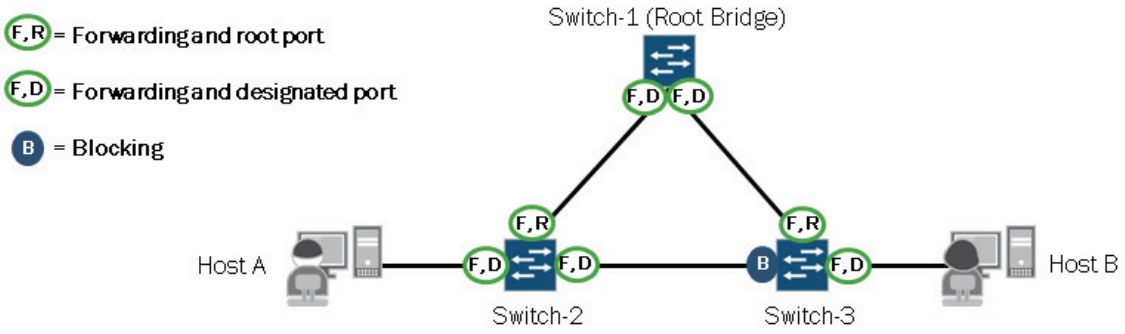
Root Bridge Election

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 switch reviews the priority values first to determine the root bridge. If the priority value of one switch is lower than the priority value of all other switches, that switch is elected as the root bridge. If the priority values are equal for multiple switches, STP evaluates the system MAC addresses of the remaining switches and elects the switch with the lowest MAC address as the root bridge.

Port Role and State Determination

- Least-cost path calculation to root bridge determines port role; port role determines port state:

Port Role and State Designations
All ports on root bridge assume designated port role and forwarding state
Root ports on switches are placed in the forwarding state; root bridge has no root ports
Designated ports on designated bridges are placed in the forwarding state
All other ports are placed in the blocking state

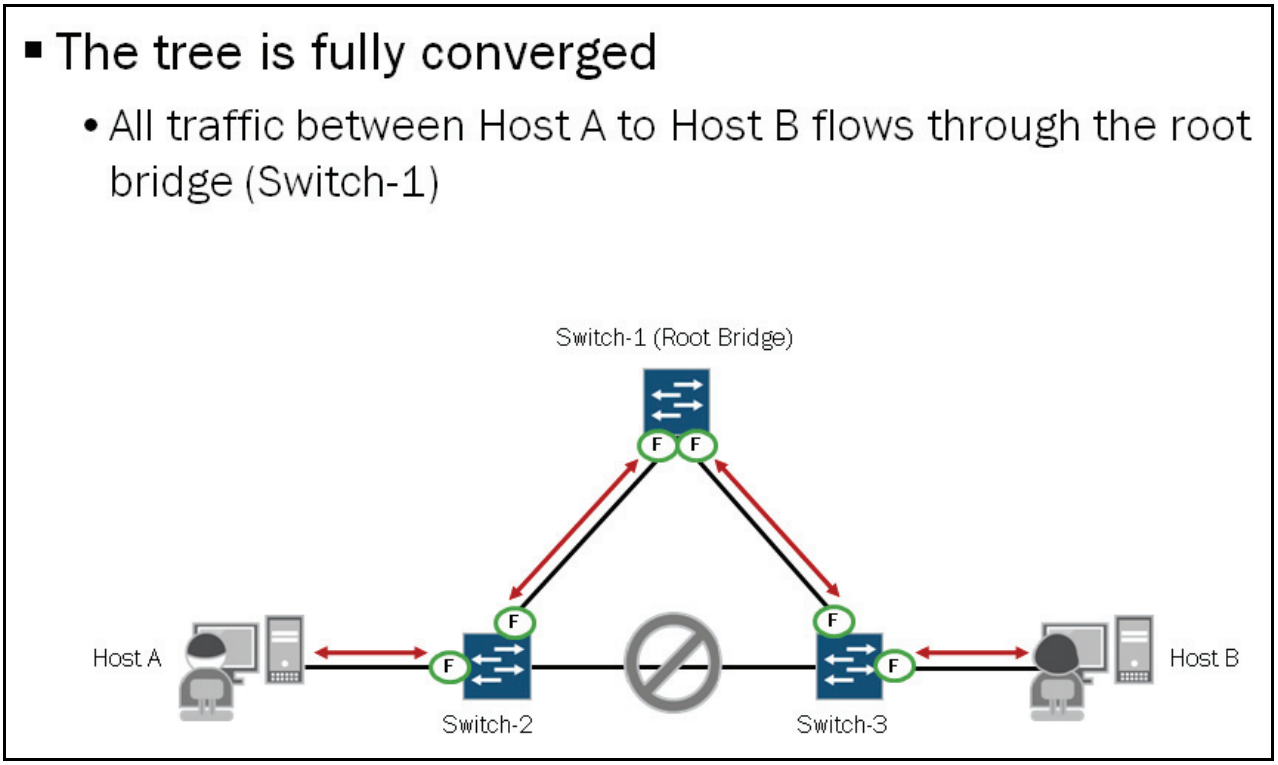


Once the root bridge election occurs, all nonroot devices perform a least-cost path calculation to the root bridge. The results of these calculations determine the role of the switch ports. The role of the individual switch ports determines the port state.

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.

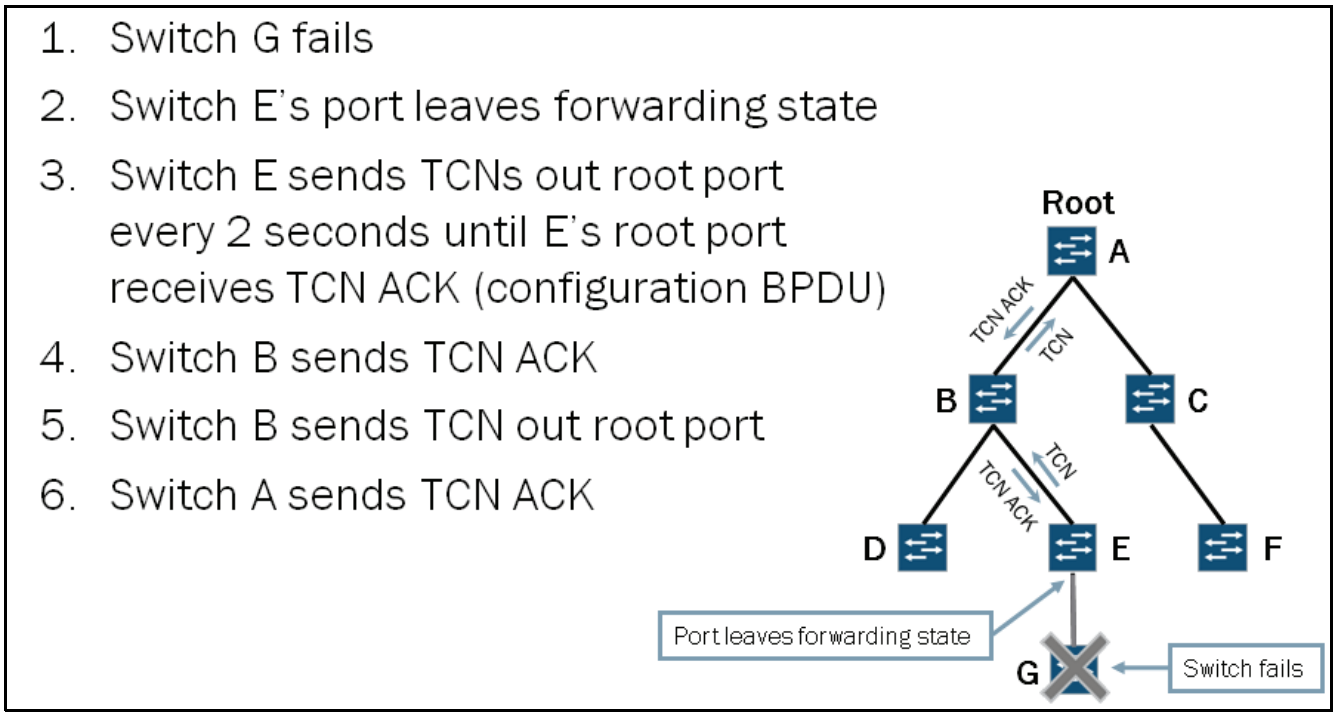
STP selects a designated bridge on each LAN segment. This selection process is also based on the least-cost path calculation from each switch to the root bridge. Once the designated bridge selection occurs, its port, which connects to the LAN segment, is chosen as the designated port. If the designated 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 switch determines the role and state for its 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 graphic, all traffic passing between Host A and Host B transits the root bridge (Switch-1).

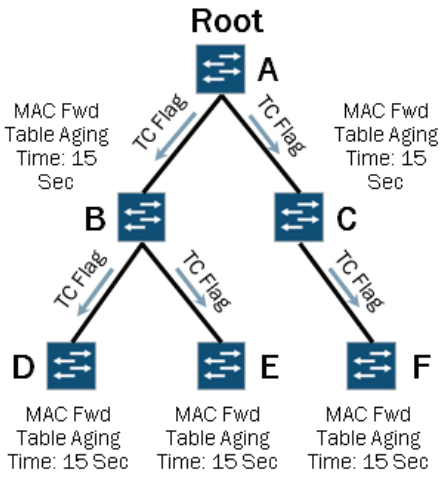
Reconvergence Example: Part 1



The graphic shows the first several steps during a failure and reconvergence scenario.

Reconvergence Example: Part 2

7. The root bridge sets the topology change flag and sends an updated configuration BPDU
8. Switches 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 graphic shows the remainder of 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.

Drawbacks of STP

- Slow convergence time
  - STP uses timers to transition between port states
    - STP can take 30 to 50 seconds to respond to a topology change (20 seconds for a BPDU to age out, 15 seconds for the listening state, and 15 seconds for the learning state)
  - Root bridge is responsible for communicating the current tree topology

For STP to recover from a link failure, it takes approximately 50 seconds: 20 seconds for a BPDU to age out, 15 seconds for the listening state, and 15 seconds for the learning state. This recalculation of the spanning tree is a time-consuming process and can result in delayed message delivery as ports transition between states. Users perceive these delays as service interruptions and certain applications, protocols, or processes can time out. These results are unacceptable in current high-availability networks, which led to the evolution of STP to RSTP.

STP and RSTP maintain the spanning tree differently. Both use BPDUs to communicate the current tree topology. With STP, the root bridge initiates these messages and they propagate throughout the tree every hello time interval. With RSTP, a non-root bridge sends a BPDU with its current information every hello time interval, regardless of receiving BPDUs from the root bridge. Note that EX Series switches configured to use STP actually run RSTP force version 0, which is compatible with STP, so BPDU behavior is the same.

RSTP Defined

Rapid Spanning Tree Protocol (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.

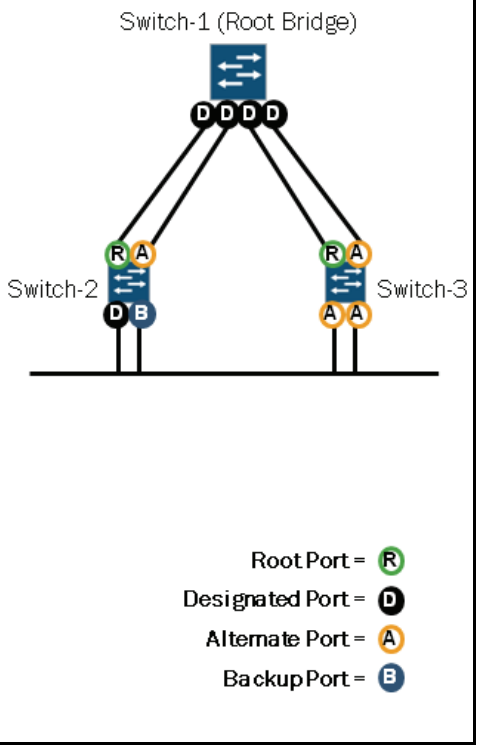
RSTP Convergence Improvements

- Point-to-point link designation
- Edge port designation
  - A port that connects to a LAN with no other bridges attached
  - It is always in the forwarding state
- Allows for rapid recovery from failures
  - A new root port or designated port can transition to forwarding without waiting for the protocol timers to expire
- Direct and indirect link failure and recovery

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 new port roles:
  - Alternate port:
    - Provides an alternate path to the root bridge (essentially a backup root port)
    - Blocks traffic while receiving superior BPDUs from a neighboring switch
  - Backup port:
    - Provides a redundant path to a segment (on designated switches only)
    - Blocks traffic while a more preferred port functions as the designated port
- RSTP continues to use the root and designated 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.

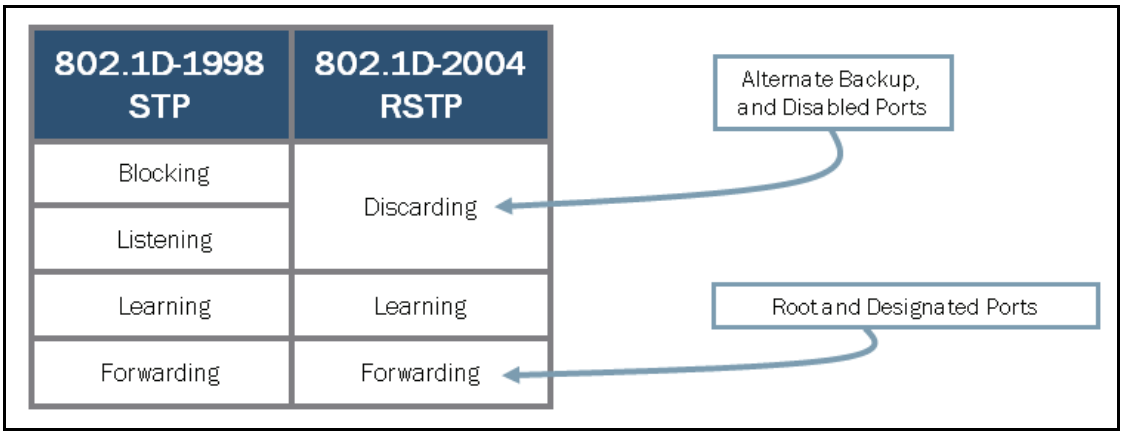
When a designated bridge has multiple ports connected to a shared LAN segment, it selects one of those ports as the designated port. The designated port is typically the port with the lower port ID. RSTP considers all other ports on the designated switch that connects to that same shared LAN segment 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.

STP and RSTP Port States



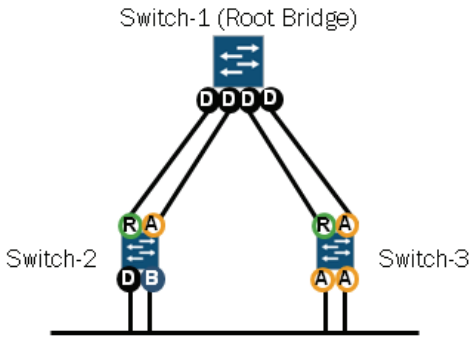
The graphic highlights the STP and RSTP port states. In addition to the states listed on the graphic, an interface can have STP administratively disabled. An administratively disabled port does not participate in the spanning tree but does flood any BPDUs it receives to other ports associated with the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

- *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.

RSTP uses fewer port states than STP. 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 simultaneously are in the forwarding state. As the graphic indicates, only root and designated ports use the forwarding state.

Rapid Spanning Tree BPDUs

- Act as keepalives
  - RSTP-designated ports send Configuration BPDUs every hello time (default of 2 seconds)
- Provide faster failure detection
  - If a neighboring bridge receives no BPDU within 3 times the hello interval (3 x 2 = 6 seconds), connectivity to the neighbor is faulty

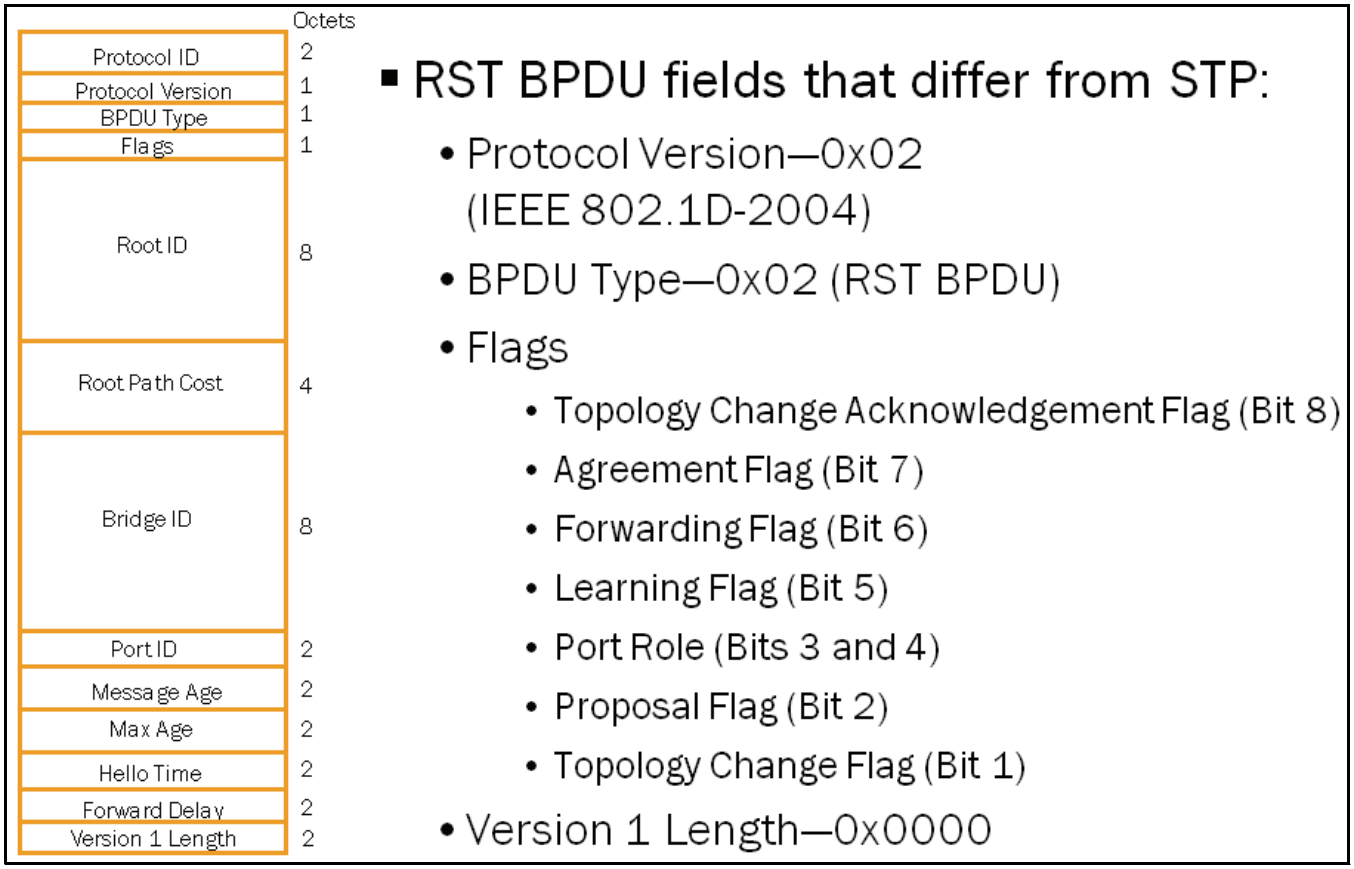


As previously mentioned, STP uses BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, prune specific redundant links to create a loop-free tree topology, and report and acknowledge topology changes. RSTP configuration BPDUs also 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 switch receives no BPDU within three times the hello interval, it assumes connectivity is faulty and updates the tree. By default, RSTP detects a failure within 6 seconds, whereas it might take up to 50 seconds when using STP (maximum age of 20 seconds plus the listening and learning states of 30 seconds).

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 timers to expire as with STP. 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 graphic. 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.

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 EX 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

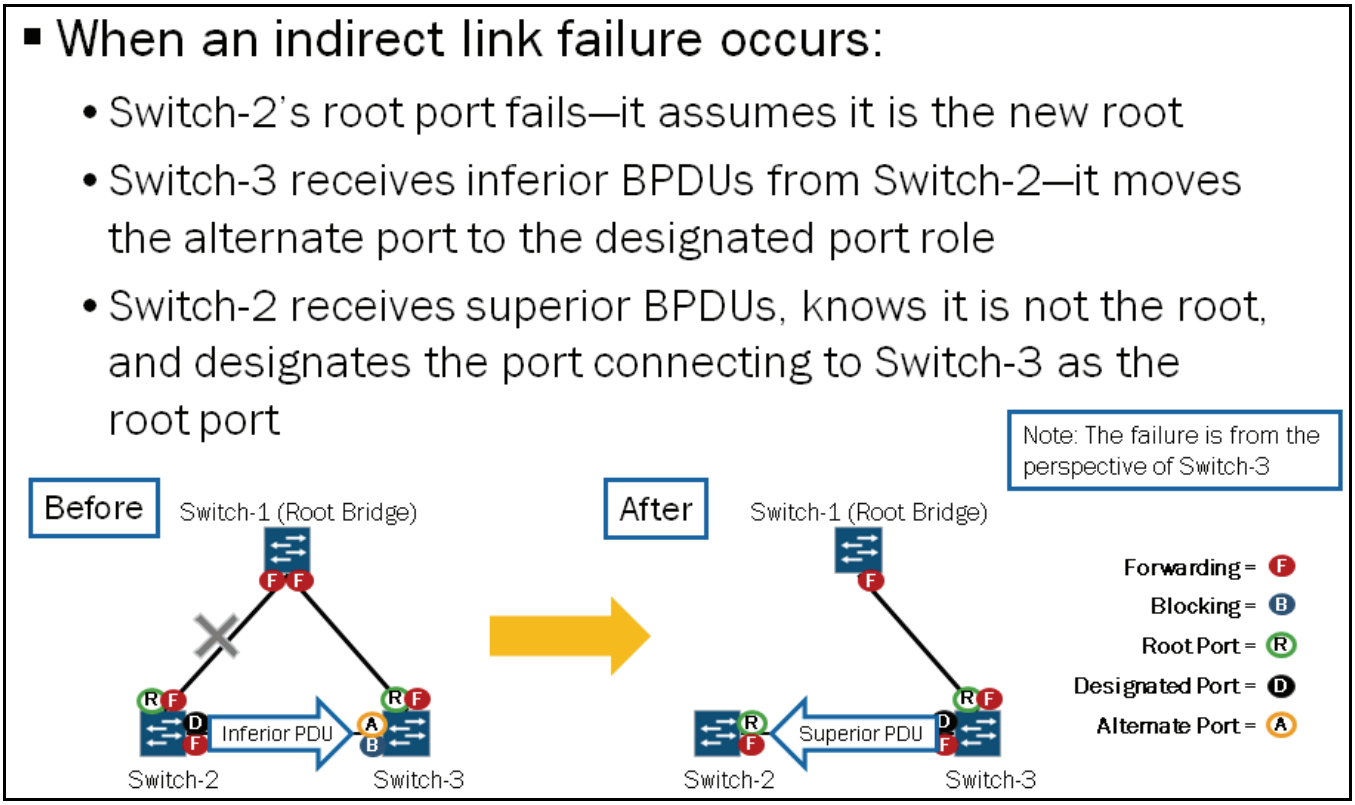
- Port transitions to the discarding state no longer trigger the STP TCN/TCN Acknowledgment sequence
- The initiator sends RSTP TCNs (RST BPDU with TCN flag set) out of all designated ports as well as out of the root port
- Because of the received RSTP TCN, switches flush the majority of MAC addresses in the bridge table
  - Switches do not flush MAC addresses learned from edge ports
  - Switches do not flush MAC addresses learned on port receiving TCN

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.

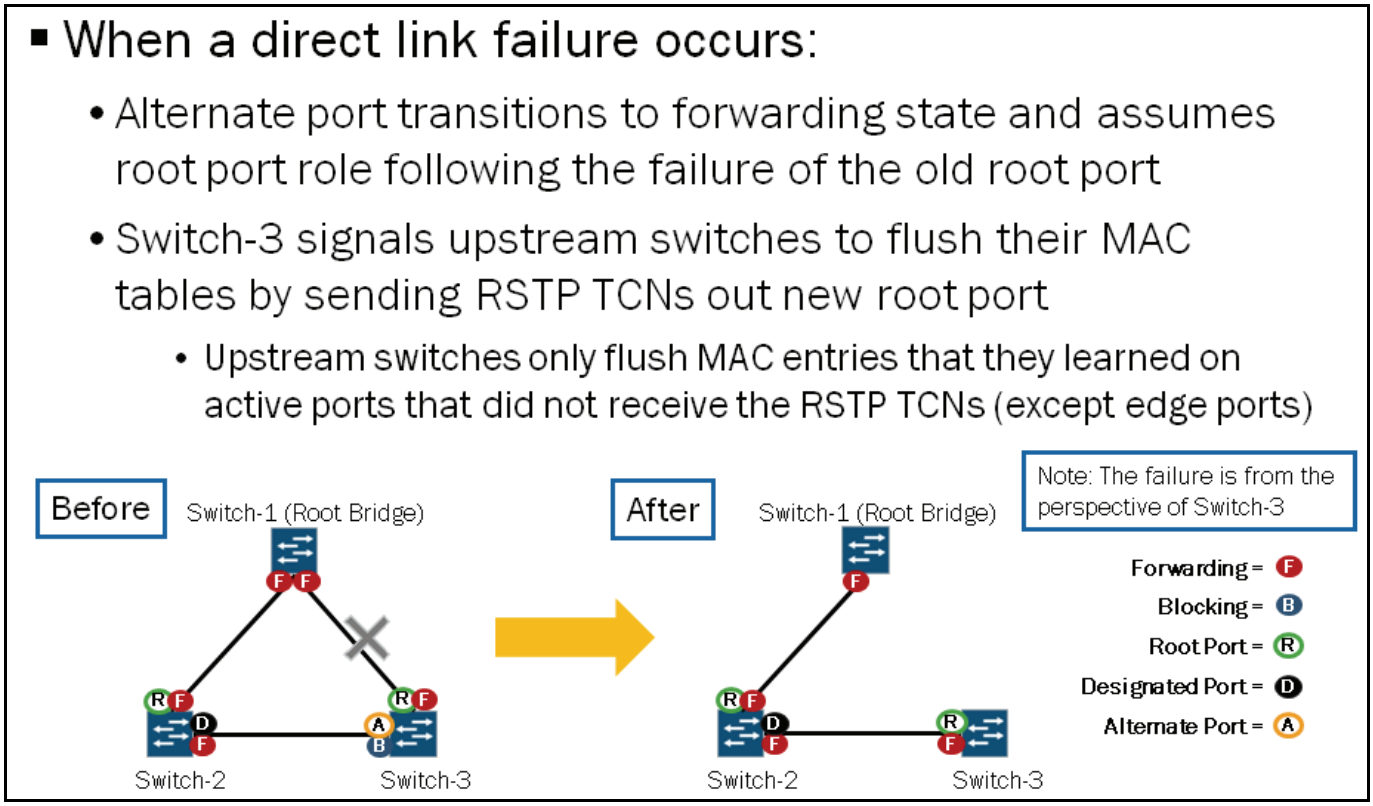
When a TCN is necessary and is generated, the initiating device floods all designated ports as well as the root port. Unlike traditional STP, neighboring switches that are not in the path of the initiator to the root bridge do not need to wait for this information from the root bridge. As the changes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

Indirect Link Failure



RSTP performs rapid recovery for link failures. The graphic illustrates a typical scenario for an indirect link failure from the perspective of Switch-3.

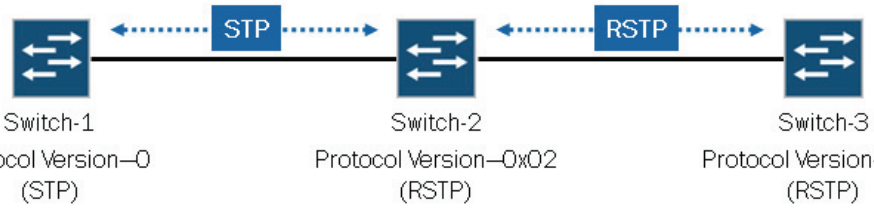
Direct Link Failure



The graphic illustrates a typical scenario in which a direct link failure occurs, from the perspective of Switch-3.

Interoperability Considerations

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



Switches configured for STP and RSTP will 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 will discard the RST BPDUs. The RSTP-capable switch, upon receiving STP BPDUs, reverts to STP mode, thus allowing interoperability between the two devices.

Configuring RSTP

```
[edit protocols rstp]
user@switch# show
bridge-priority 32k;
max-age 20;
hello-time 2;
forward-delay 15;
interface ge-0/0/10.0 {
    disable;
}
interface ge-0/0/13.0 {
    cost 20000;
    mode point-to-point;
}
interface ge-0/0/14.0 {
    priority 128;
    mode shared;
}
interface ge-0/0/2.0 {
    edge;
}
```

Default RSTP settings

Excludes interface from participating in RSTP

Default cost value for interfaces operating at 1 Gbps

Default interface mode for interfaces operating in full-duplex mode

Default priority value (used to influence downstream device's least-cost path calculation to root bridge—lower is better)

Default interface mode for interfaces operating in half-duplex mode

Default value for interfaces that do not connect to STP-enabled devices

The graphic illustrates a sample RSTP configuration along with several highlighted settings. Note that the max age and forwarding delay values used by a switch always match the values defined on the root bridge device.

The following sample configuration shows some basic STP configuration. EX Series switches 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 stp]` configuration hierarchy

```
[edit protocols stp]
user@switch# show
bridge-priority 32k;
max-age 20;
hello-time 2;
forward-delay 15;
```

Monitoring Spanning Tree Operation: Part 1

```
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 bridge
STP bridge parameters
Context ID                              : 0
Enabled protocol                        : RSTP
  Root ID                               : 4096.00:19:e2:55:36:00
  Root cost                             : 40000
  Root port                             : ge-0/0/13.0
  Hello time                            : 2 seconds
  Maximum age                           : 20 seconds
  Forward delay                         : 15 seconds
  Message age                           : 2
  Number of topology changes            : 2
  Time since last topology change       : 72 seconds
  Local parameters
    Bridge ID                           : 32768.00:19:e2:55:1d:40
    Extended system ID                  : 0
    Internal instance ID                : 0
```

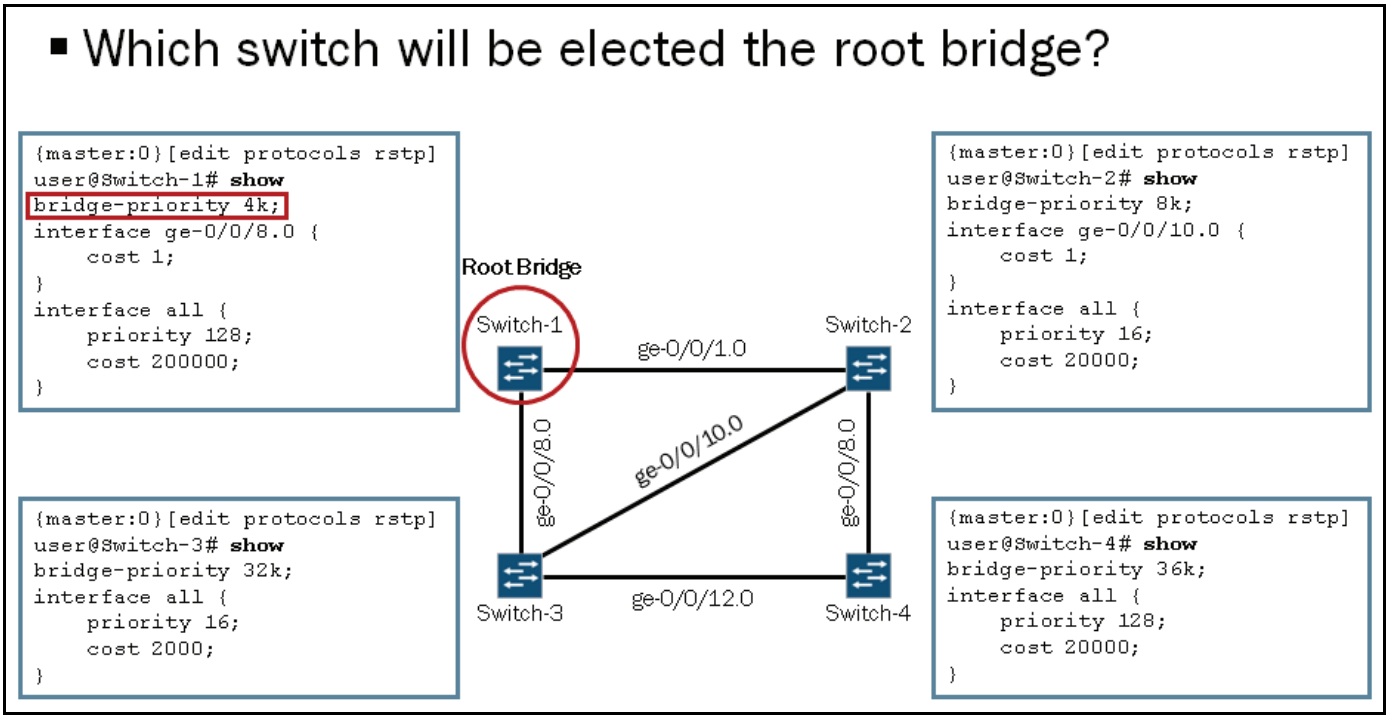

This graphic and the next illustrate some common operational-mode commands used to monitor the operation of STP and RSTP.

Monitoring Spanning Tree Operation: Part 2

```
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-0/0/10.0     128:523      128:523  32768.0019e2507c00  20000      BLK    ALT
ge-0/0/11.0     128:524      128:524  32768.0019e2507c00  20000      BLK    ALT
ge-0/0/12.0     128:525      128:525  32768.0019e2507c00  20000      BLK    ALT
ge-0/0/13.0     128:526      128:526  32768.0019e2503fe0  20000      FWD    ROOT
ge-0/0/14.0     128:527      128:527  32768.0019e2503fe0  20000      BLK    ALT
ge-0/0/15.0     128:528      128:528  32768.0019e2503fe0  20000      BLK    ALT

user@switch> show spanning-tree statistics interface

Interface    BPDUs sent   BPDUs received      Next BPDU
                                              transmission
ge-0/0/10.0          7              5                 0
ge-0/0/11.0          7              5                 0
ge-0/0/12.0          7              5                 0
ge-0/0/13.0          7              4                 0
ge-0/0/14.0          7              5                 0
ge-0/0/15.0          7              5                 0
```

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

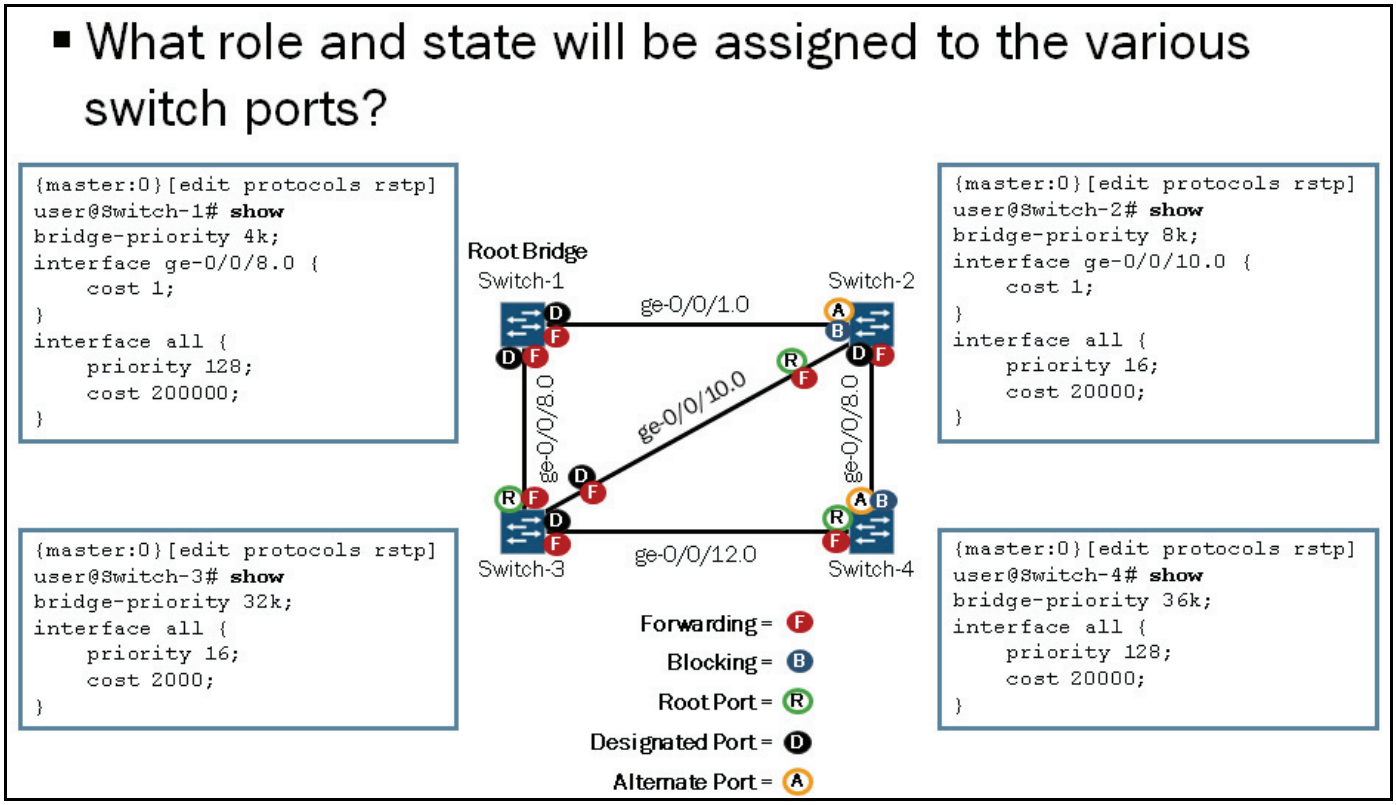
Test Your Knowledge: Part 1



This graphic is designed to test your understanding of the various configuration options and how they relate to the root bridge election process. As shown in the following output, you can use the **show spanning-tree bridge** command to verify root bridge information:

```
user@Switch-1> show spanning-tree bridge
STP bridge parameters
Context ID                              : 0
Enabled protocol                        : RSTP
  Root ID                               : 4096.00:26:88:02:74:90
  Hello time                            : 2 seconds
  Maximum age                           : 20 seconds
  Forward delay                         : 15 seconds
  Message age                           : 0
  Number of topology changes            : 1
  Time since last topology change       : 2114 seconds
  Topology change initiator             : ge-0/0/1.0
  Topology change last recvd. from      : 00:26:88:02:6b:81
  Local parameters
    Bridge ID                           : 4096.00:26:88:02:74:90
    Extended system ID                  : 0
    Internal instance ID                : 0
```

Test Your Knowledge: Part 2

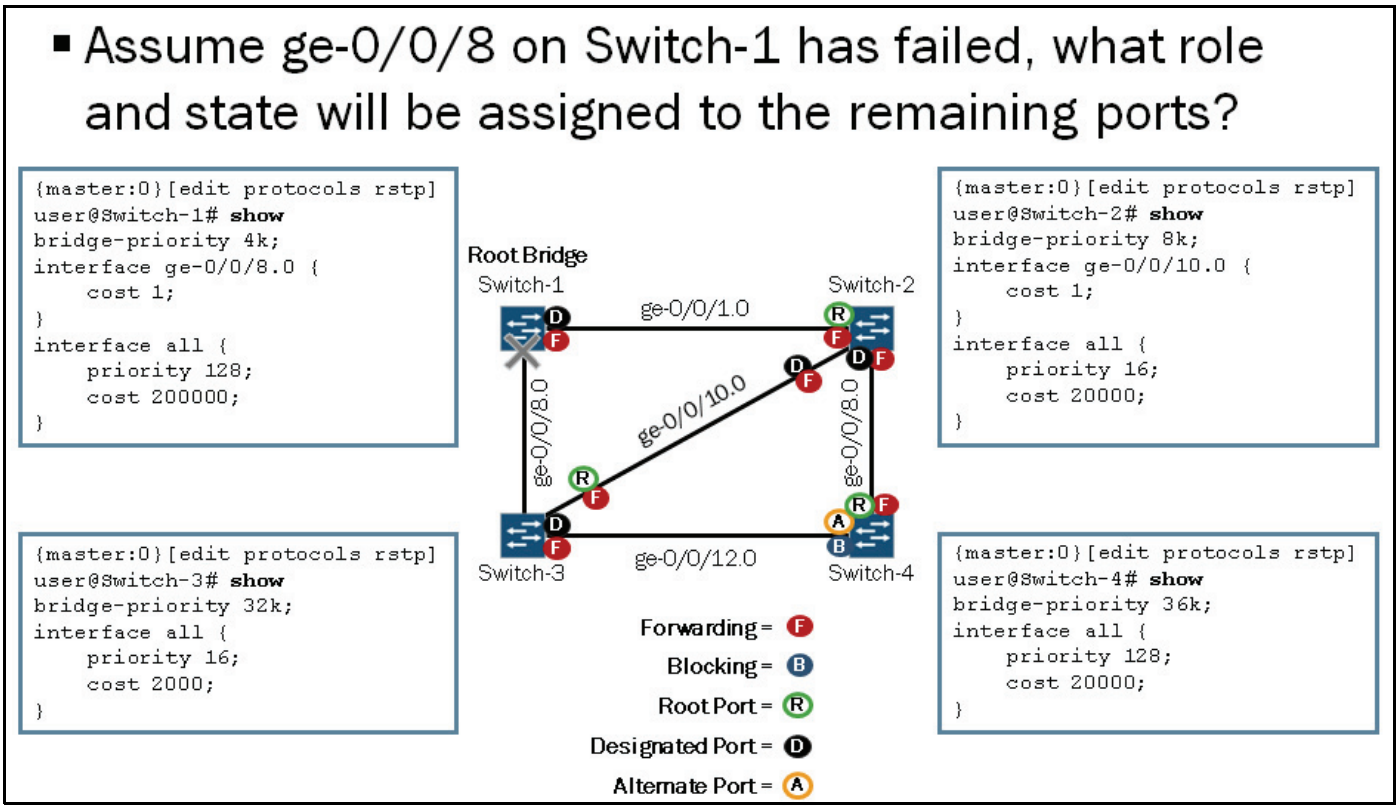


This graphic is designed to test your understanding of the various configuration options and how they relate to port role and state determination. As shown in the following output, you can use the **show spanning-tree interface** command to verify spanning tree interface information:

```
user@Switch-2> 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-0/0/1.0      16:514      128:514   4096.002688027490      20000  BLK    ALT
ge-0/0/8.0      16:521       16:521   8192.002688026b90      20000  FWD    DESG
ge-0/0/10.0    128:523       16:523  32768.0019e2516580          1  FWD    ROOT

user@Switch-3> 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-0/0/8.0      16:521      128:521   4096.002688027490       2000  FWD    ROOT
ge-0/0/10.0     16:523       16:523  32768.0019e2516580       2000  FWD    DESG
ge-0/0/12.0     16:525       16:525  32768.0019e2516580       2000  FWD    DESG
```

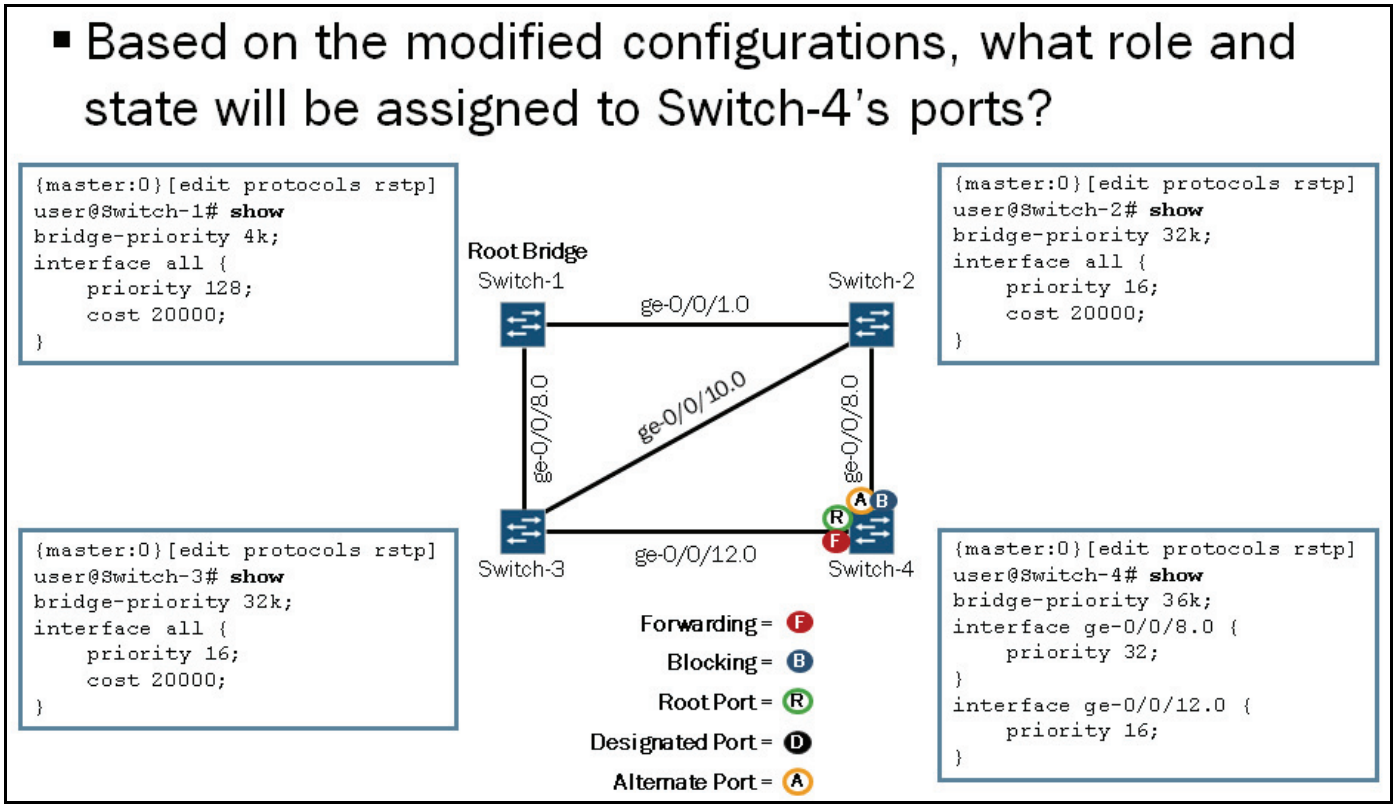
Test Your Knowledge: Part 3



This graphic is designed to test your understanding of the various configuration options and how they relate to port role and state determination. As shown in the following output, you can use the **show spanning-tree interface** command to verify spanning tree interface information:

```
user@Switch-2> 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-0/0/1.0      16:514      128:514   4096.002688027490      20000  FWD    ROOT
ge-0/0/8.0      16:521       16:521   8192.002688026b90      20000  FWD    DESG
ge-0/0/10.0    128:523      128:523   8192.002688026b90          1  FWD    DESG

user@Switch-3> 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-0/0/10.0     16:523      128:523   8192.002688026b90       2000  FWD    ROOT
ge-0/0/12.0     16:525       16:525  32768.0019e2516580       2000  FWD    DESG
```

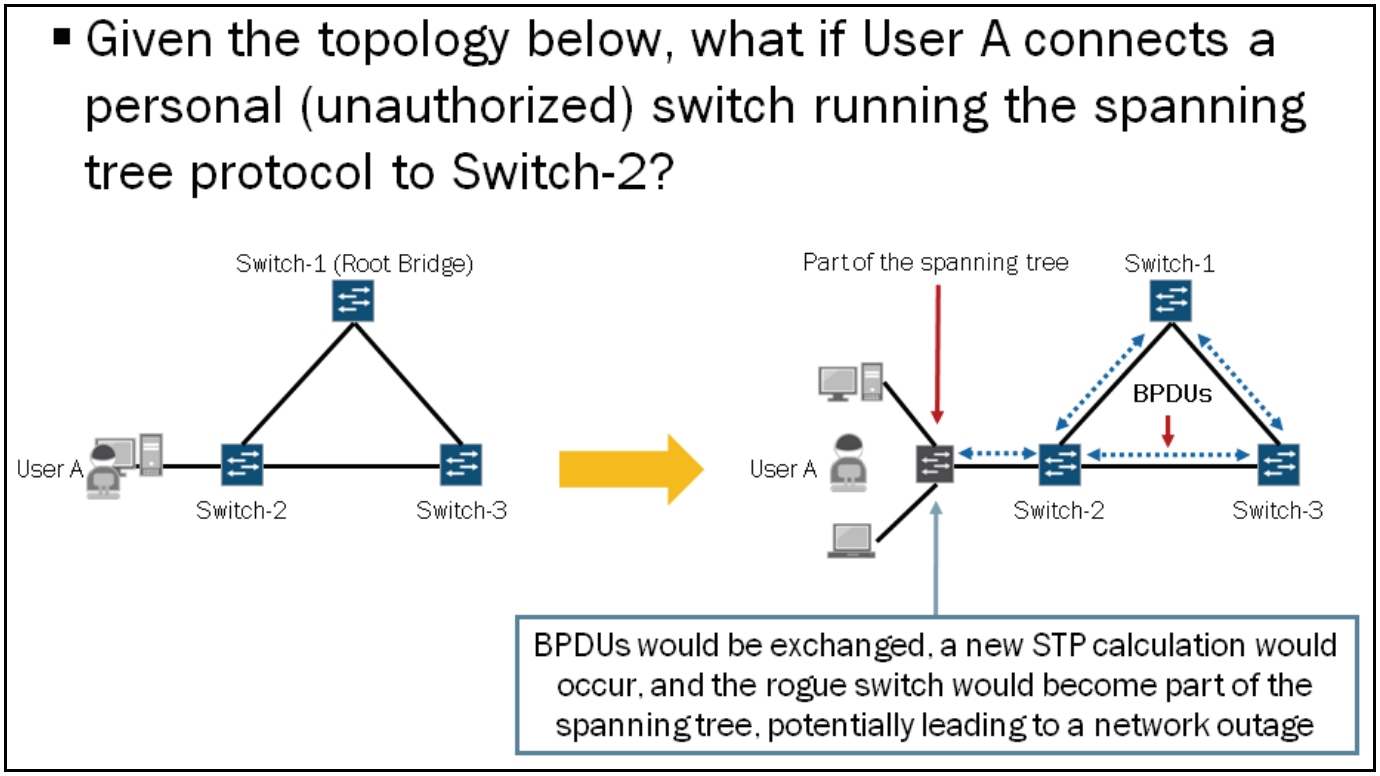
Test Your Knowledge: Part 4



This graphic is designed to test your understanding of the various configuration options and how they relate to port role and state determination. As shown in the following output, you can use the **show spanning-tree interface** command to verify spanning tree interface information:

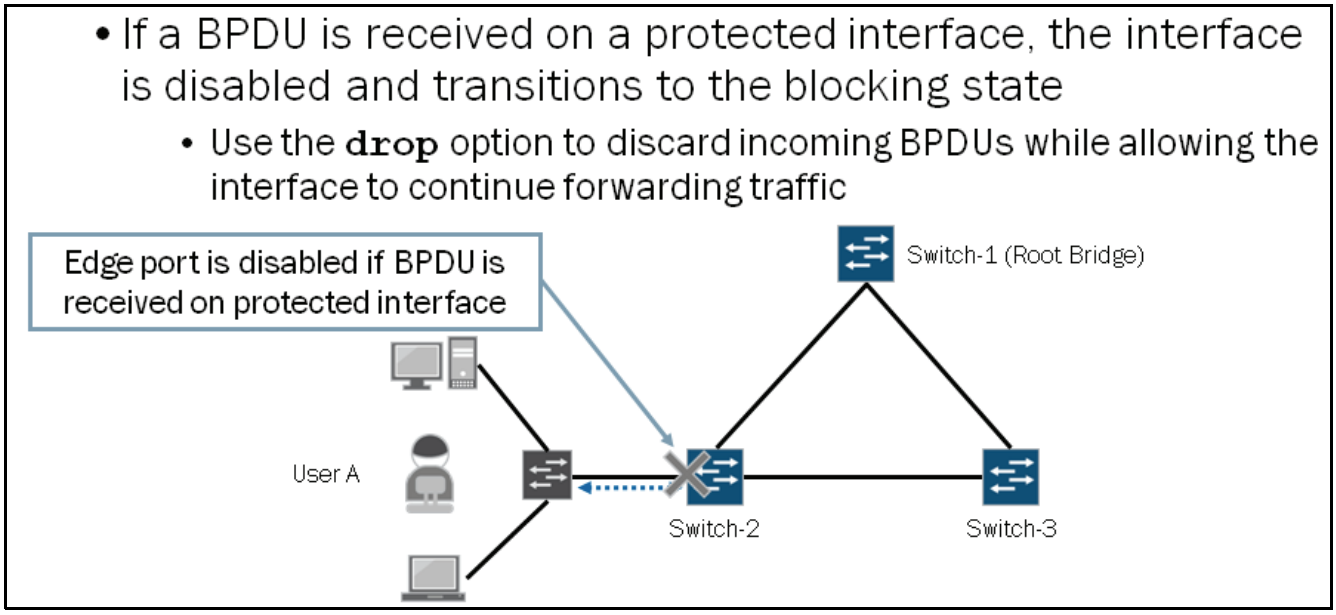
```
user@Switch-4> show spanning-tree interface
Spanning tree interface parameters for instance 0
Interface    Port ID    Designated      Designated         Port    State  Role
                        port ID         bridge ID          Cost
ge-0/0/8.0      32:521       16:521  32768.002688026b90     20000  BLK    ALT
ge-0/0/12.0     16:525       16:525  32768.0019e2516580     20000  FWD    ROOT
```

What If...?



The graphic illustrates a scenario where User A connects a rogue switch to the network so multiple devices can participate on the network. Assuming the rogue switch has spanning tree running, it would exchange BPDUs with Switch-2 causing a new spanning tree calculation to occur. Once the spanning tree calculation is complete, the rogue switch would then become part of the spanning tree. Having an unauthorized device become part of the spanning tree could have some negative impact on the network and its performance. For example, a rogue device could trigger a spanning-tree miscalculation and potentially cause a Layer 2 loop or even a complete network outage.

BPDU Protection

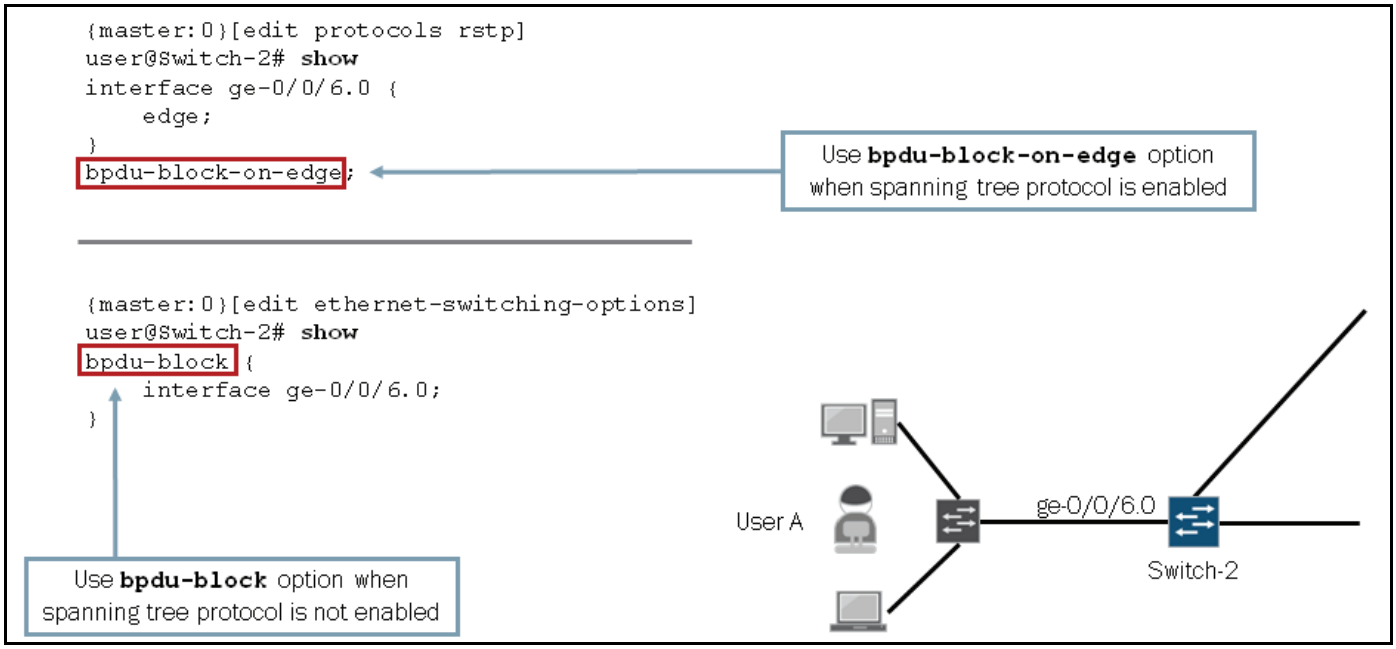


You can enable BPDU protection on switch interfaces on which no BPDUs are expected. If a protected interface receives BPDUs, the switch disables the interface and stops forwarding frames by transitioning the interface to a blocking state. In some situations you might not want the interface to become unavailable if BPDUs are received. You can include the **drop** option to

discard any incoming BPDUs while allowing the interface to remain up and functional. You can only configure the **drop** statement on interfaces that do not have any type of spanning tree protocol enabled.

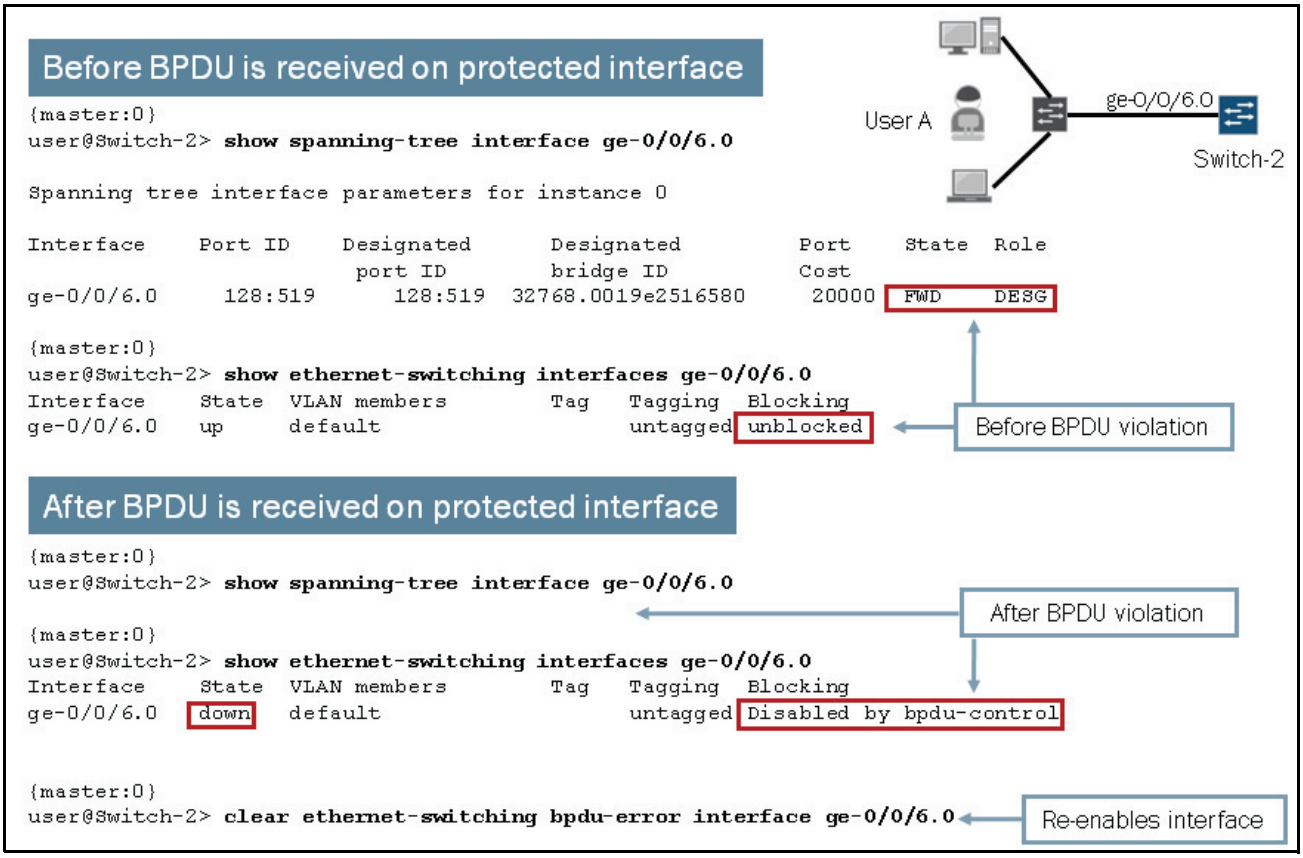
You can configure BPDU protection on a switch with a spanning tree as well as on a switch that is not running STP. We discuss the configuration of BPDU protection in the next section.

Configuring BPDU Protection



You can configure BPDU protection on edge ports to block incoming BPDUs. The graphic illustrates two configuration examples; the top configuration example is used when a spanning tree protocol is enabled and the bottom configuration example is used when no spanning tree protocol is in use. With this configuration enabled, if Switch-2 receives a BPDU from the rogue switch connected to ge-0/0/6.0, Switch-2 would transition the ge-0/0/6.0 interface to the blocking state and stop forwarding frames.

Monitoring BPDU Protection



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 ethernet-switching interfaces** operational mode command.

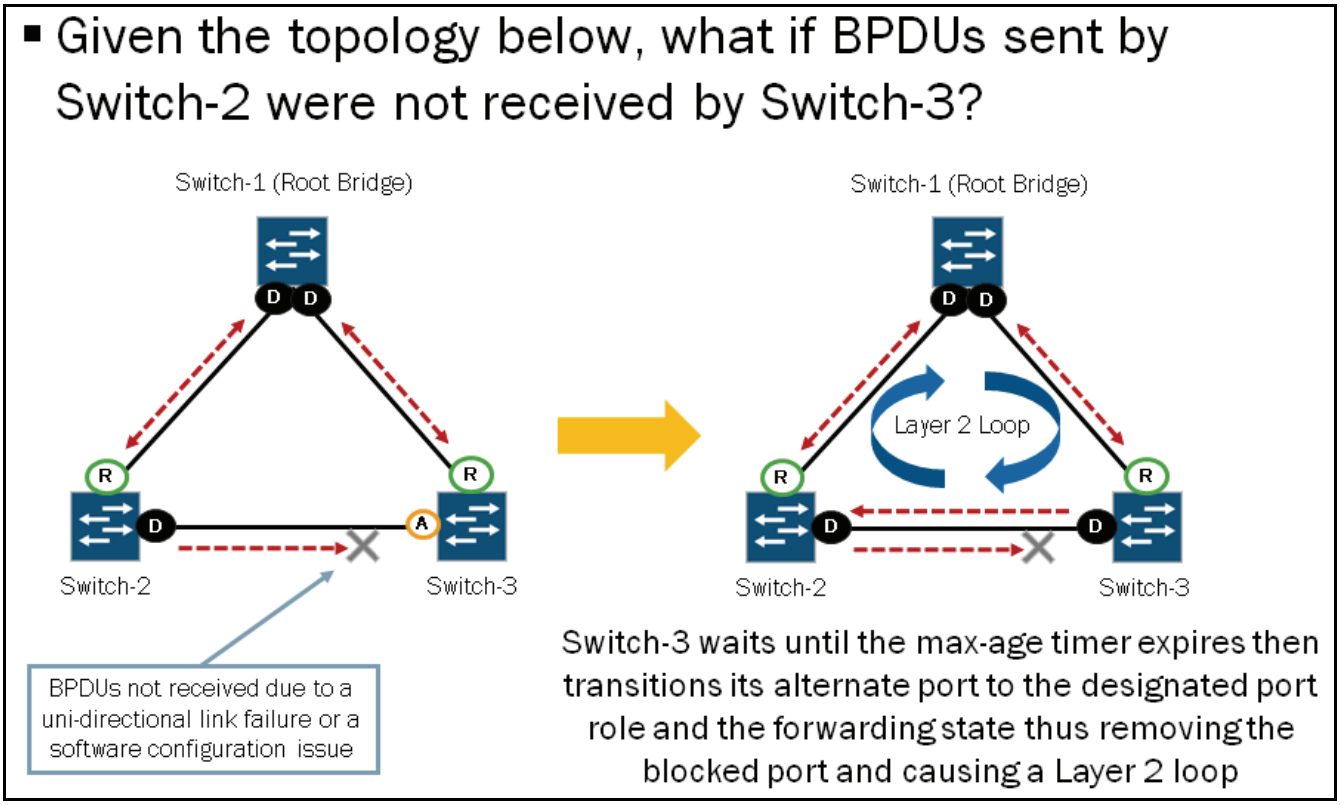
These commands provide the information on the state and role changes on the protected interfaces. Specifically, once the BPDUs are sent from an offending device to the protected interface, the interface transitions to the DIS role, meaning that it becomes a BPDU inconsistent state. The BPDU inconsistent state changes the interfaces' state to blocking (BLK), preventing them from forwarding traffic.

To unblock the interfaces, use the **clear ethernet-switching bpdu-error** operational mode command. Alternatively, you can use the disable-timeout option to allow the interface to return to service automatically after the timer expires. The following configuration example illustrates the disable-timeout option:

```
{master:0}[edit ethernet-switching-options]
user@Switch-2# set bpdu-block disable-timeout ?
Possible completions:
  <disable-timeout>    Disable timeout for BPDU Protect (10..3600 seconds)
```

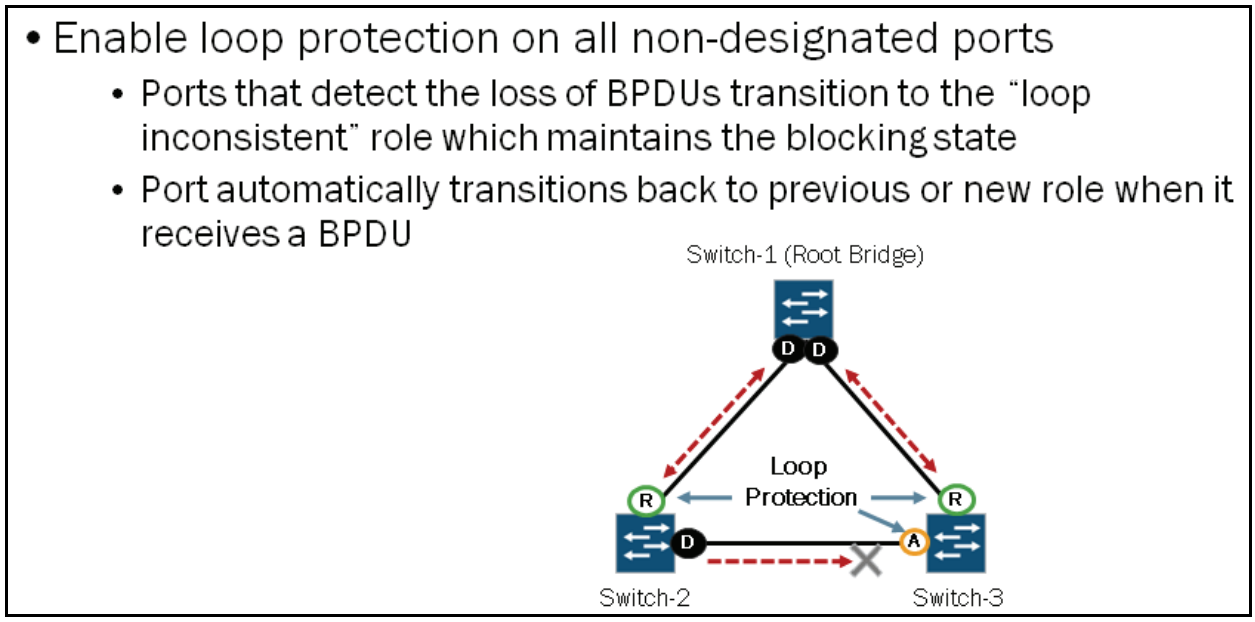
Disabling the BPDU protection configuration for an interface does not unblock the interface. You must clear the violation using the **clear ethernet-switching bpdu-error** command or wait for the configured timer to expire.

What If...?



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 graphic illustrates this point.

Loop Protection



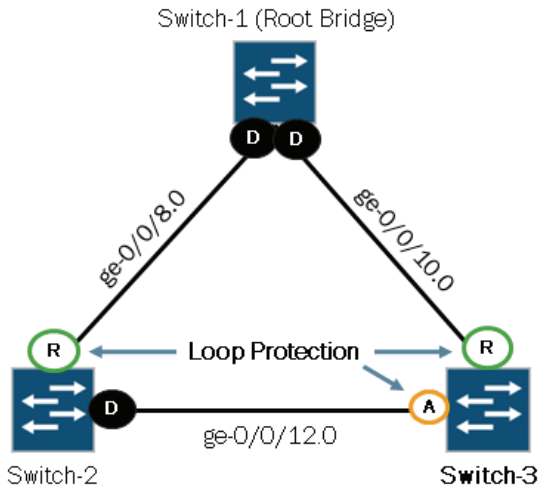
When loop protection is enabled, the spanning-tree topology detects root ports and blocked ports, and ensures that both are receiving BPDUs. If an interface with the loop protection feature enabled stops receiving BPDUs from its designated port, it reacts as it would react to a problem with the physical connection on this 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 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.

Configuring Loop Protection

```
{master:0}[edit protocols rstp]
user@Switch-3# show
interface ge-0/0/10.0 {
    bpdu-timeout-action {
        block;
    }
}
interface ge-0/0/12.0 {
    bpdu-timeout-action {
        block;
    }
}
```

Use the `block` or `alarm` action in conjunction with the loop protection feature



The graphic illustrates the required configuration for loop protection on Switch-3's root and alternate ports. The example configuration illustrates the use of the `block` option, which, if a violation occurs, the affected interface immediately transitions to the DIS (Loop-Incon) role and remain in the blocking (BLK) state. The `block` option also writes related log entries to the `messages` log file.

You can alternatively use the `alarm` option, which does not force a change of the port's role but simply writes the related log entries to the `messages` log file. If the `alarm` option is used, the switch port assumes the designated port role and transitions its state to the forwarding (FWD) state once the max-age timer expires.

Note that an interface can be configured for either loop protection or root protection, but not both. We discuss root protection in the next section.

Monitoring Loop Protection

When BPDUs are received on protected interface:

```
{master:0}
user@Switch-3> 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-0/0/10.0     128:523      128:523   4096.002688027490      20000  FWD    ROOT
ge-0/0/12.0     128:525      128:525  16384.0019e2516580      20000  BLK    ALT
```

When BPDUs are not received on protected interface:

```
{master:0}
user@Switch-3> 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-0/0/10.0     128:523      128:523   4096.002688027490      20000  FWD    ROOT
ge-0/0/12.0     128:525      128:525  32768.0019e2553600      20000  BLK    DIS (Loop-Incon)
```

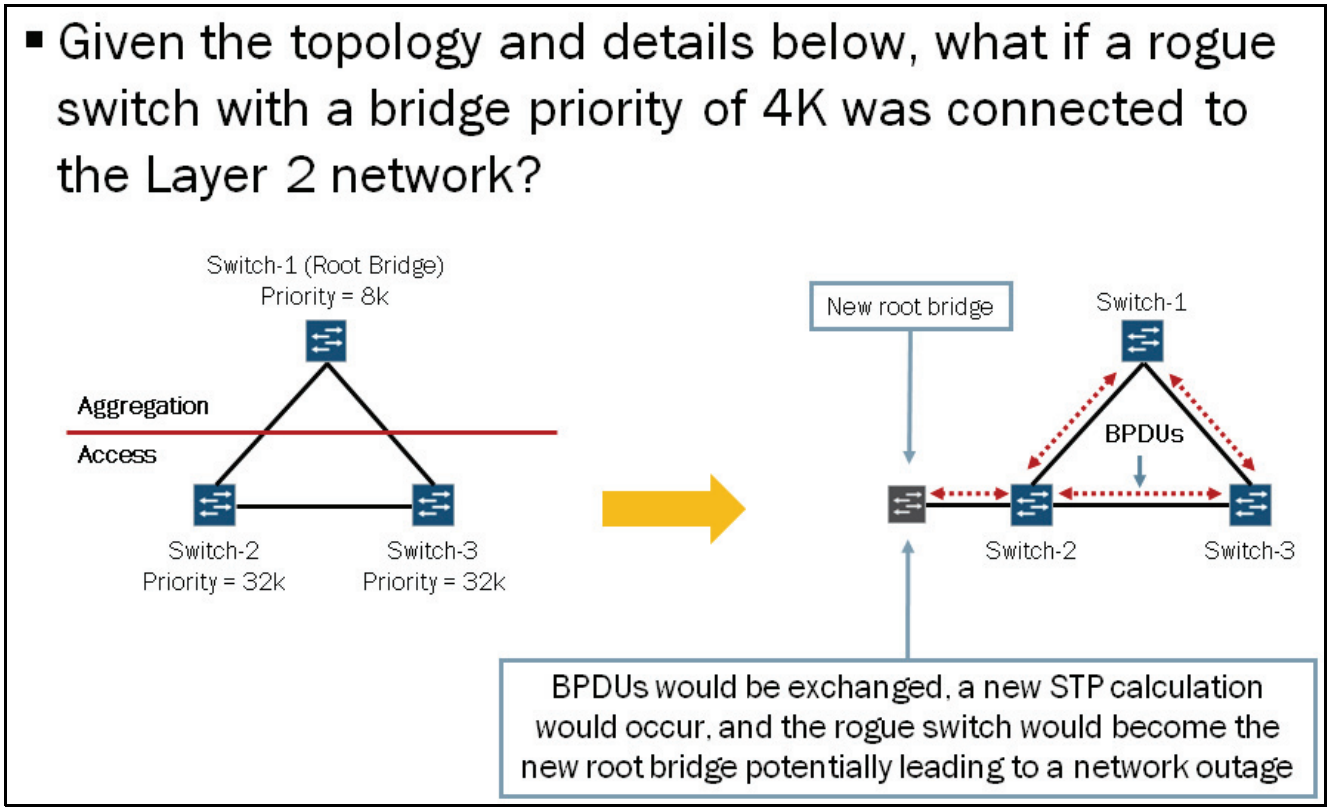
To confirm that the configuration is working properly on the STP-running switch, use the **show spanning-tree interface** operational mode command prior to configuring loop protection. This command provides information for the interface's spanning-tree state, which should be blocking (BLK).

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 now shows that the protected interface has transitioned to the DIS (Loop-Incon) role and remains in the blocking (BLK) state, which prevents the interface from transitioning to the forwarding state. The interface recovers and transitions back to its original state when it receives BPDUs.

You can also monitor the interface role transitions using the **show log messages** command as shown in the following capture:

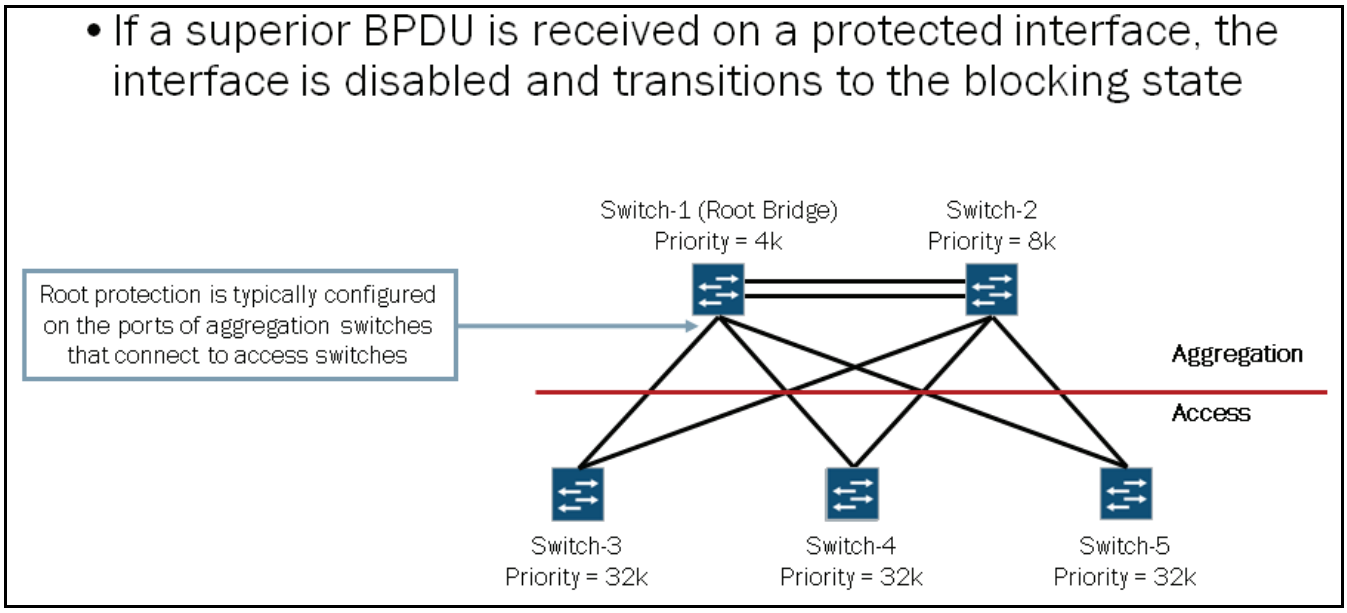
```
{master:0}
user@Switch-3> show log messages | match "loop|protect"
Apr 27 20:04:49  Switch-3 eswd[40744]: Loop_Protect: Port ge-0/0/12.0: Received
information expired on Loop Protect enabled port
Apr 27 20:04:49  Switch-3 eswd[40744]: ESWD_STP_LOOP_PROTECT_IN_EFFECT: ge-0/0/12.0:
loop protect in effect for instance 0
Apr 27 20:05:27  Switch-3 eswd[40744]: ESWD_STP_LOOP_PROTECT_CLEARED: ge-0/0/12.0:
loop protect cleared for instance 0
```

What If...?



The graphic illustrates a scenario where a rogue switch running a spanning tree protocol is connected to the network. Once connected to the network, the rogue switch exchanges BPDUs with Switch-2 which in turn causes a new spanning tree calculation to occur. Once the spanning tree calculation is complete, the rogue switch is the new root bridge for the spanning tree. Having an unauthorized device become part of the spanning tree or worse become the root bridge for the Layer 2 network could have some negative impact and affect the network's overall performance or even cause a complete network outage.

Root Protection

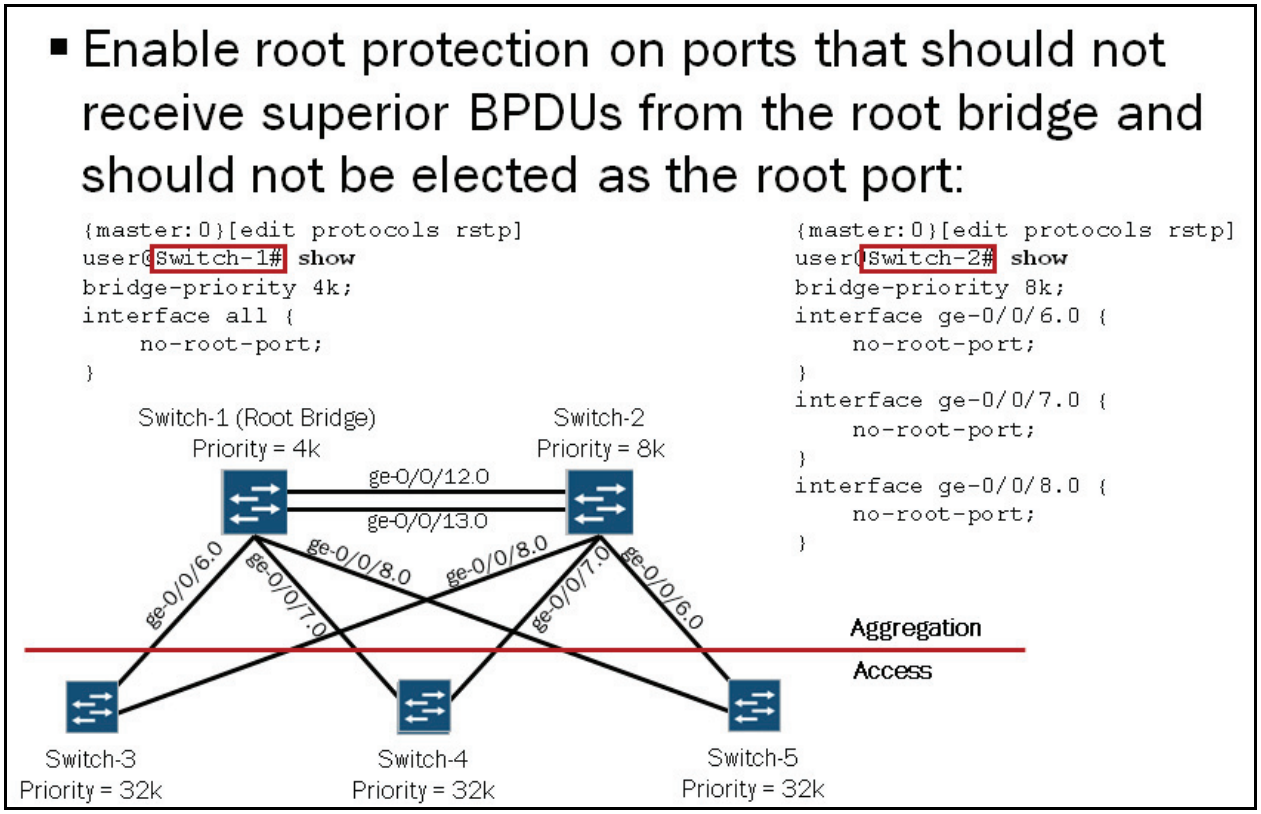


Enable root protection on interfaces that should not receive superior BPDUs and should not be elected as the root port. These interfaces become designated ports. If the bridge receives superior BPDUs on a port that has root protection enabled, that port transitions to an inconsistency state, blocking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

After the switch stops receiving superior BPDUs on the interface with root protection, the interface returns to a listening state, followed by a learning state, and ultimately back 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. Interface is blocked only for instances for which it receives superior BPDUs. Otherwise, it participates in the spanning-tree topology.

Configuring Root Protection



This graphic illustrates a sample topology and configuration for the two aggregation switches (Switch-1 and Switch-2). In this example, you can see that root protection has been enabled on all ports that should not receive superior BPDUs or be elected as the root port. On Switch-1, all ports should be elected as designated ports. On Switch-2, ge-0/0/6.0, ge-0/0/7.0, and ge-0/0/8.0 should be designated ports.

As previously mentioned, you can configure an interface for either loop protection or root protection, but not both. If both features are configured, the configuration will not commit as shown in the following output:

```
{master:0}[edit protocols rstp]
user@Switch-1# show interface ge-0/0/6.0
bpdu-timeout-action {
    block;
}
no-root-port;

{master:0}[edit protocols rstp]
user@Switch-1# commit
[edit protocols rstp]
  'interface ge-0/0/6.0'
    Loop Protect cannot be enabled on a Root Protect enabled port
error: configuration check-out failed
```

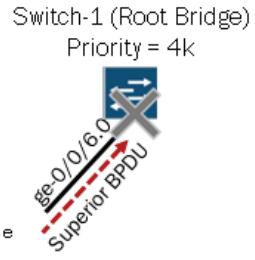
Monitoring Root Protection

Before superior BPDU is received on protected interface

```
{master:0}
user@Switch-1> 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-0/0/6.0      128:519      128:519   4096.0019e2516580      20000   FWD    DESG
ge-0/0/7.0      128:520      128:520   4096.0019e2516580      20000   FWD    DESG
ge-0/0/8.0      128:521      128:521   4096.0019e2516580      20000   FWD    DESG
ge-0/0/12.0     128:525      128:525   4096.0019e2516580      20000   FWD    DESG
ge-0/0/13.0     128:526      128:526   4096.0019e2516580      20000   FWD    DESG
```

After superior BPDU is received on protected interface

```
{master:0}
user@Switch-1> 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-0/0/6.0      128:519      128:519     0.002688027490       20000   BLK    ALT (Root-Incon)
ge-0/0/7.0      128:520      128:520   4096.0019e2516580      20000   FWD    DESG
ge-0/0/8.0      128:521      128:521   4096.0019e2516580      20000   FWD    DESG
ge-0/0/12.0     128:525      128:525   4096.0019e2516580      20000   FWD    DESG
ge-0/0/13.0     128:526      128:526   4096.0019e2516580      20000   FWD    DESG
```



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 again 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 how to build a spanning tree.
3. How are STP and RSTP different?
4. What is the purpose of the BPDU protection feature?

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.

The basic steps involved in building a spanning tree are that switches exchange BPDUs, all participating switches elect a single root bridge based on the received BPDUs, and the switches determine the role and state of their individual ports. Once these steps are complete, the tree is considered fully converged.

3.

RSTP provides a number of advantages of STP. These advantages help to significantly improve link-convergence time over that found with STP. Some differences include the point-to-point and edge port designations and fewer port state designations.

4.

BPDU protection prevents rogue switches from connecting to a Layer 2 network and causing undesired Layer 2 topology changes and possible outages.

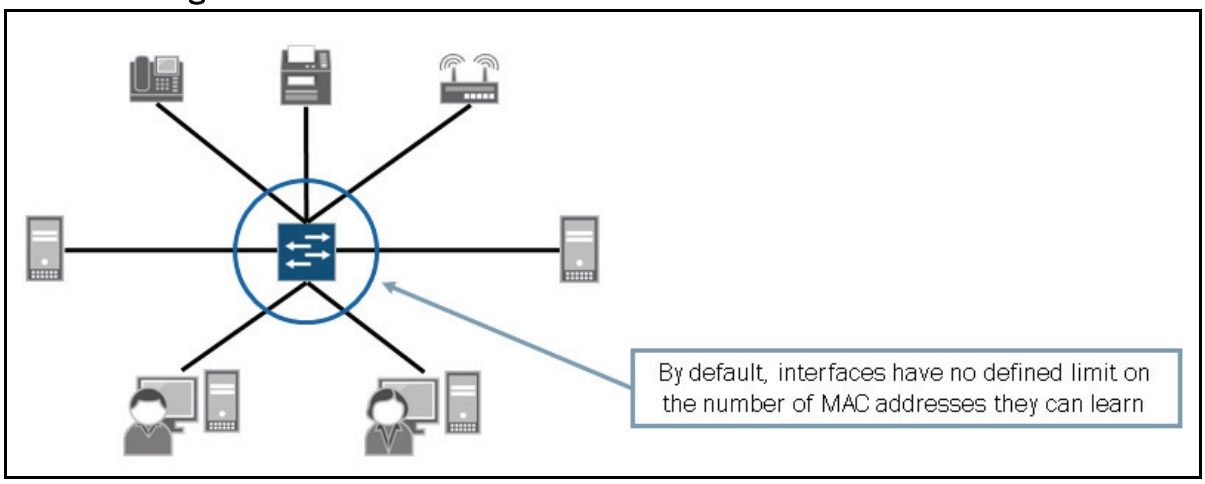
JNCIS-ENT Switching Study Guide

Chapter 4: Port Security

This Chapter Discusses:

- Various port security features including MAC limiting, DHCP snooping, Dynamic ARP Inspection (DAI), and IP source guard; and
- The configuration and monitoring of the previously listed port security features.

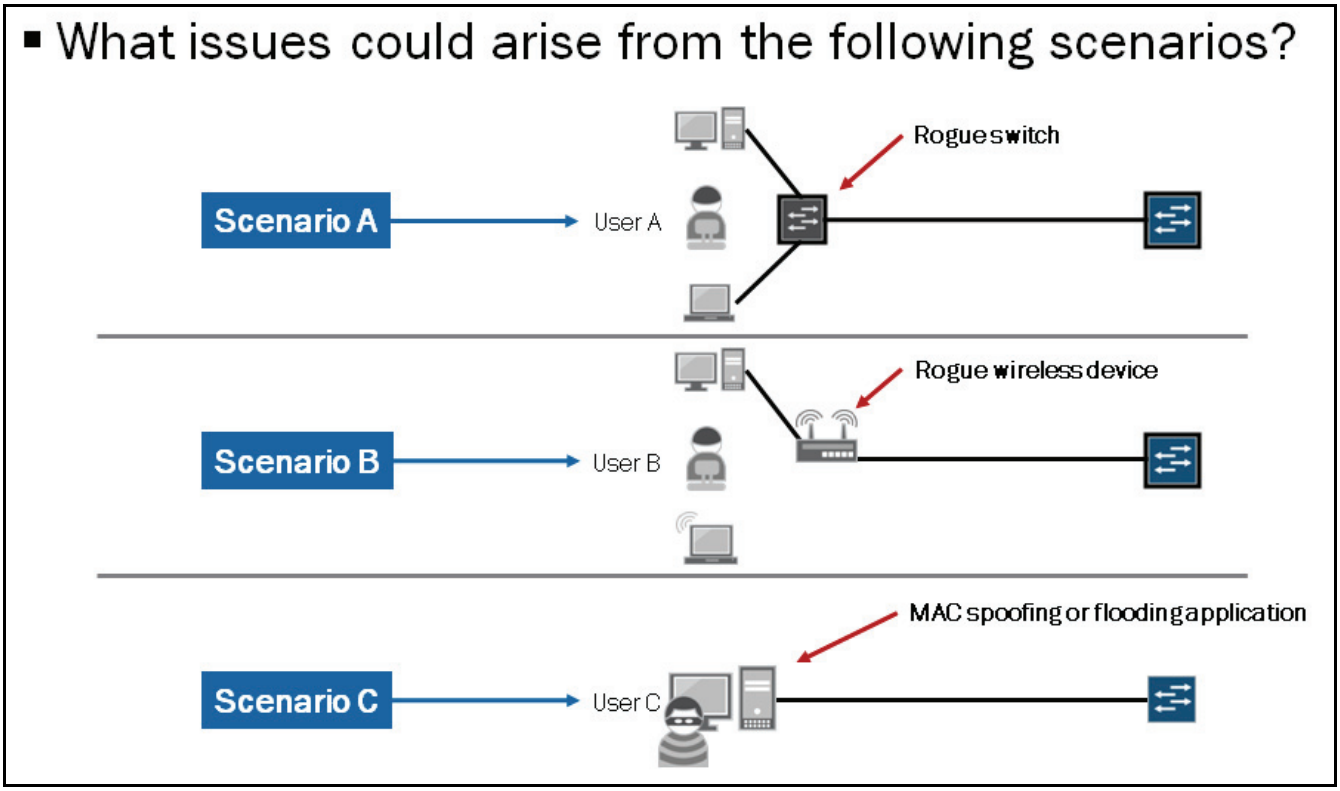
Factory Default Settings



By default, all installed interfaces in an EX Series switch are configured for Layer 2 operations. These Layer 2 interfaces do not have a defined limit on the number of MAC addresses that can be learned. In some environments, this default implementation can be problematic and prone to security risks. Once a physical connection is passed to an end-user, that user can, if proactive measures are not taken, connect a rogue switch or wireless device to the network allowing access for unauthorized devices to the network and its resources.

We discuss several port security features throughout this chapter that combat the potential security risks that are inherent with the default configuration settings. Note that not all features are supported on all EX Series devices currently. For details for your specific device, refer to the features overview found in the technical publications (http://www.juniper.net/techpubs/en_US/junos10.1/topics/concept/ex-series-software-features-overview.html).

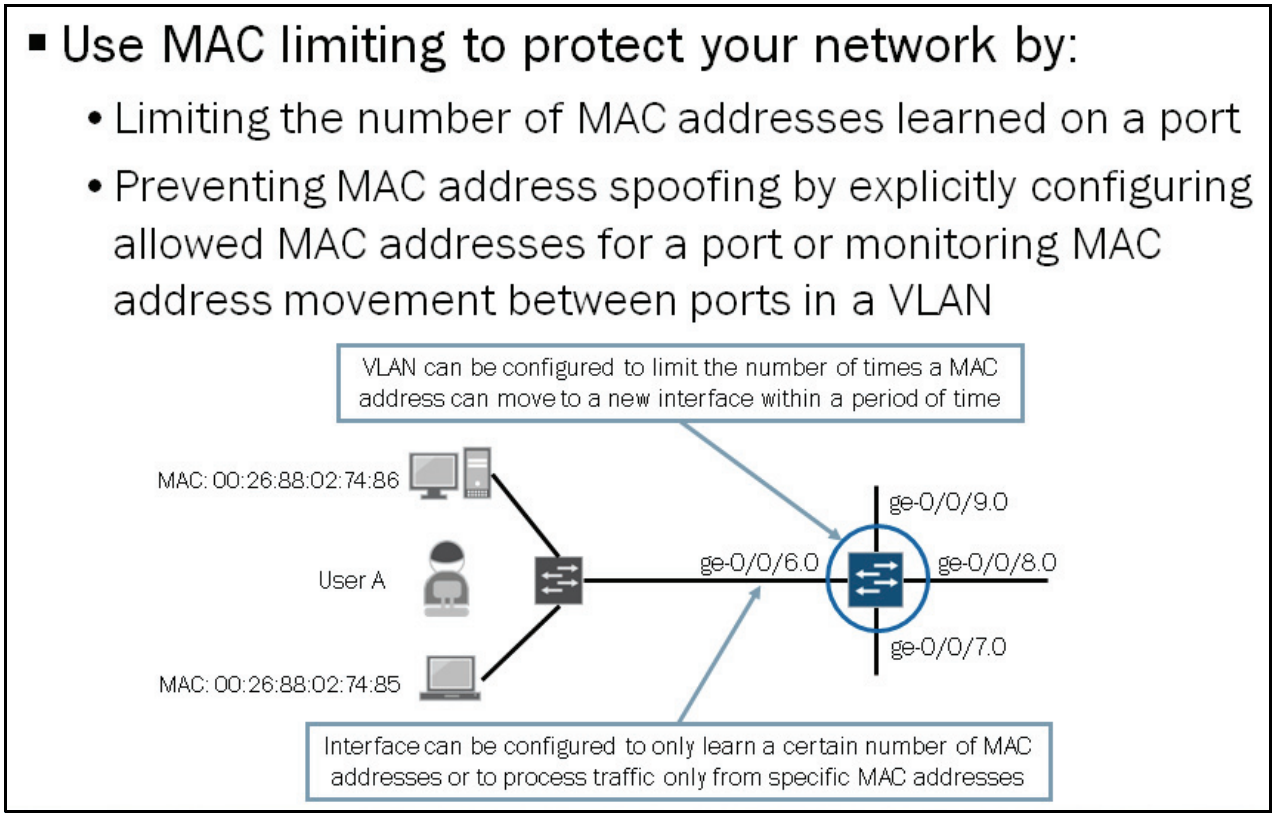
Think About It



This graphic is designed to get you thinking about the potential issues that can arise from the scenarios shown on the graphic.

Among other things, you could see Layer 2 topology changes that affect network performance or cause complete outages, unauthorized access to your network and its resources, or a network outage caused by resource overload through a DoS attack. In reality many potential issues can arise if you do not protect your network and its resources. The port security features covered in this section and throughout the chapter can help mitigate some of these potential issues.

MAC Limiting



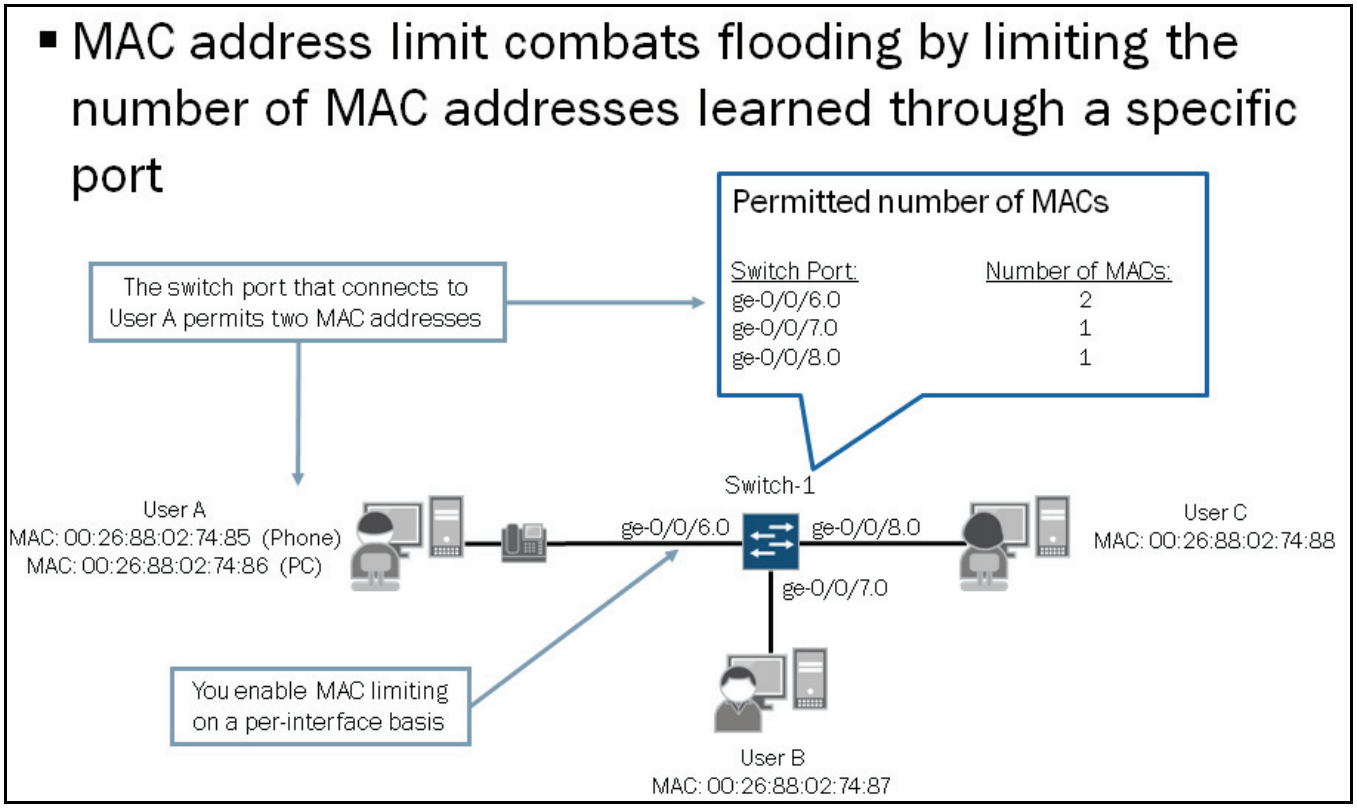
MAC limiting protects Ethernet switches, as well as other network resources, against attacks that use MAC addresses. Some examples of attacks that use MAC addresses to disrupt network operations include MAC flooding and MAC spoofing. Both MAC flooding and MAC spoofing can be quite harmful because they facilitate a denial-of-service (DoS) attack, which renders users, systems, or entire networks useless. MAC limiting can be implemented using two different methods.

The first method allows you to specify the maximum number of MAC addresses that can be learned on a single Layer 2 access port. Once the switch reaches the MAC limit, all traffic sourced from new MAC addresses is subject to being dropped based on the configured action.

The second method allows you to define allowed MAC addresses for a specific access port. Any MAC address that is not listed will not be learned or permitted network access.

You can also enable MAC move limiting which allows the switch to track the number of times a MAC address can move to a new interface (port) within a VLAN.

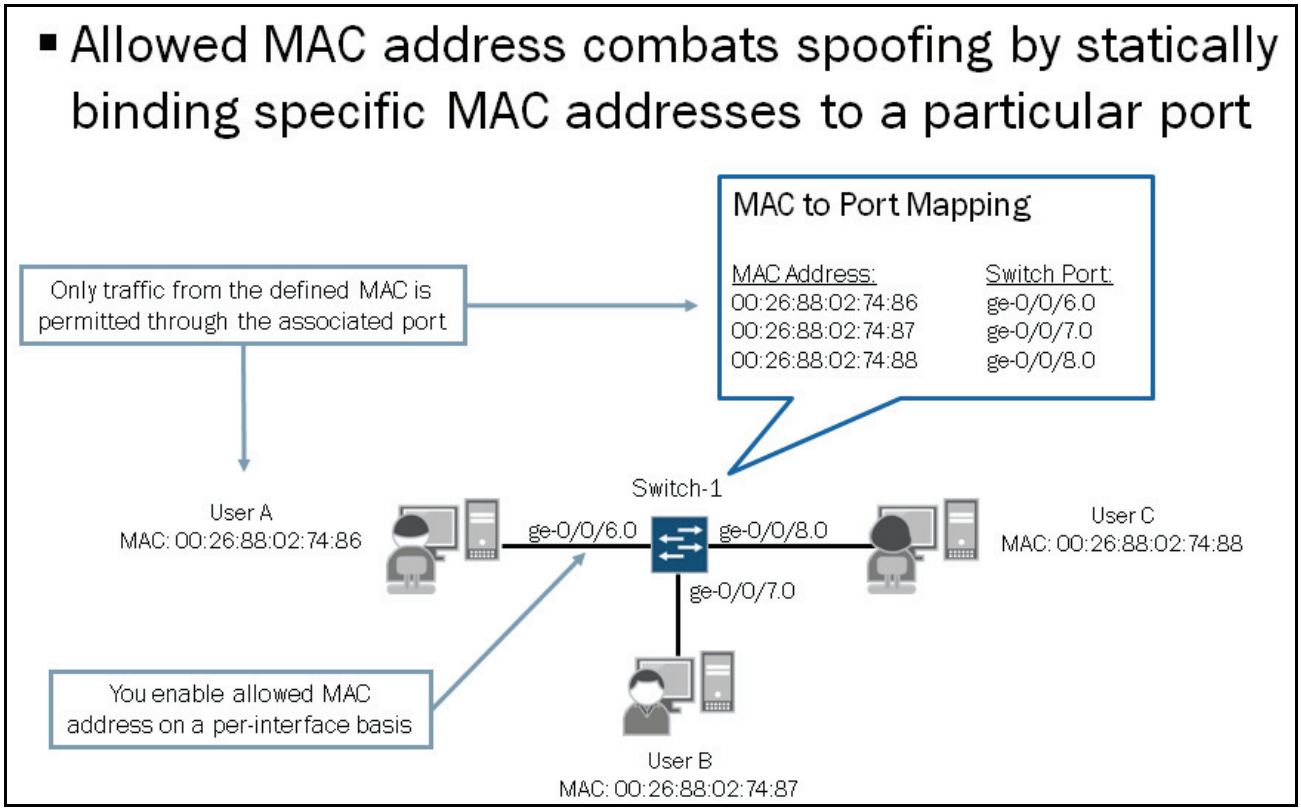
MAC Address Limit



MAC limiting protects the MAC forwarding table against flooding. You enable this feature on individual interfaces. The MAC limit is user defined and varies depending on the needs within each environment. In environments that use IP telephony, the limit specified should be two when an IP phone and a user's PC are attached to the same switch port. In data-only environments, you can typically specify a limit of one to account for the user's PC connection.

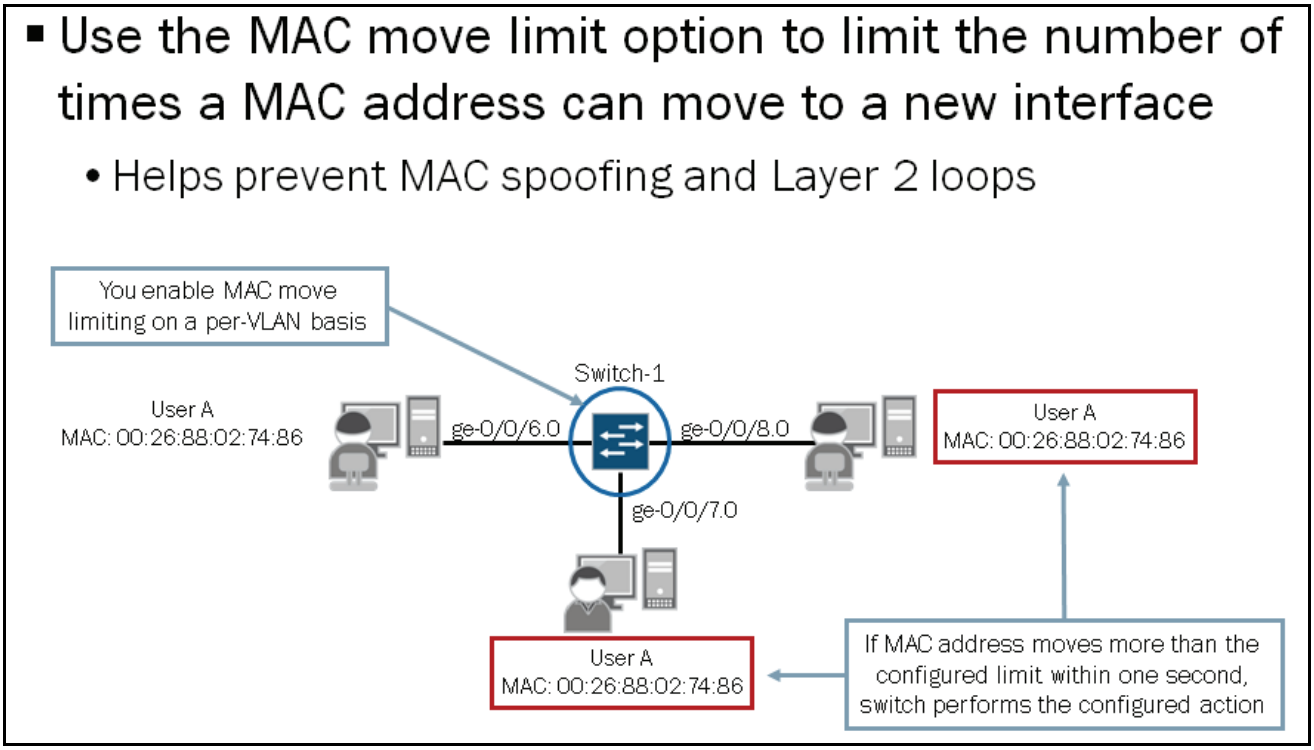
On the graphic we see that two devices; a PC and an IP phone, require access to the network through ge-0/0/6.0. To accommodate this access requirement, ge-0/0/6.0 is configured with a MAC limit of two. All subsequent MACs attempting to access the network through this port are subject to the configured action. We cover the configurable actions for MAC limiting later in this section.

Allowed MAC Address



With the allowed MAC address option, a switch permits or denies hosts network access through their attached network ports based on their MAC addresses. This requires knowledge of the node's MAC address and is not ideal in environments where end-users move from switch port to switch port.

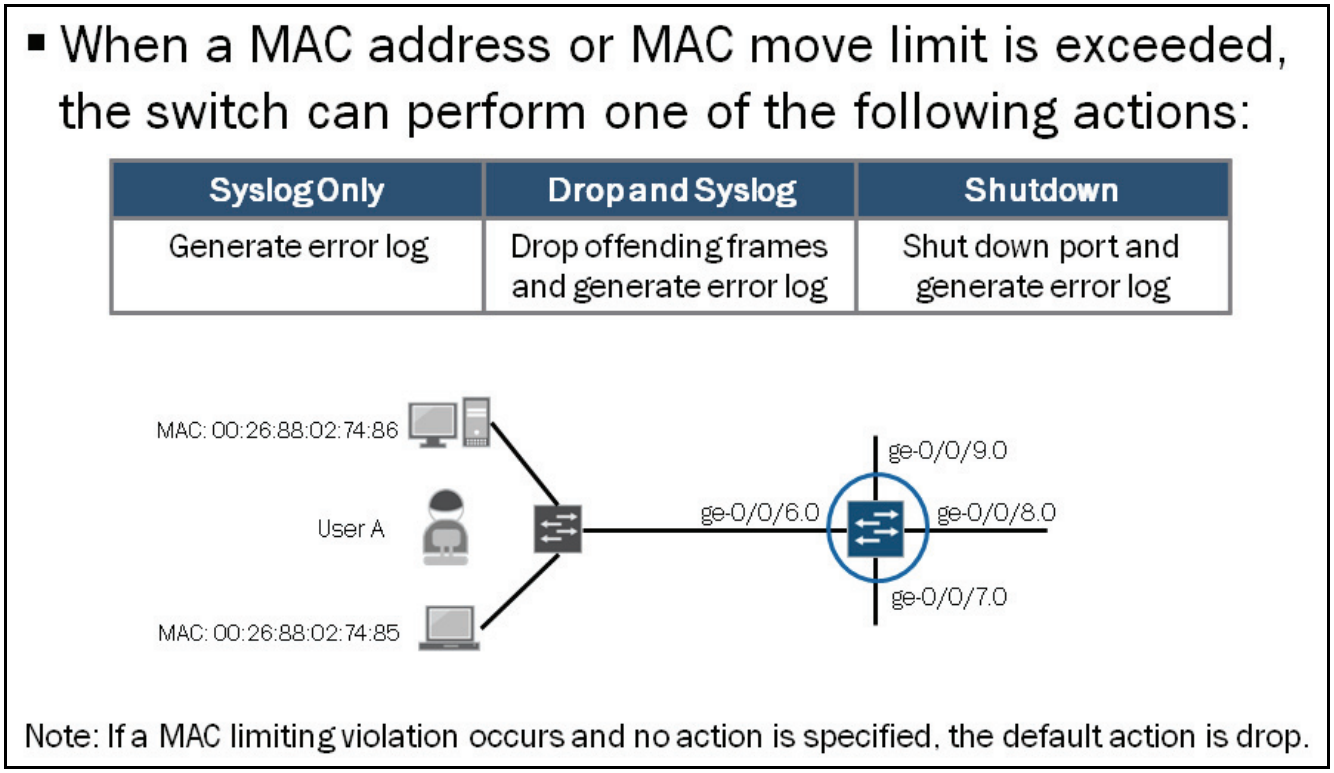
MAC Move Limiting



MAC move limiting is used to limit the number of times a MAC address can move to a new interface. This feature is used to prevent MAC spoofing attacks as well as Layer 2 loops. You enable MAC move limiting on a per-VLAN basis rather than on a

per-interface basis like the allowed MAC and MAC limit options. Once enabled, the switch tracks the number of times a MAC address moves to a new interface. If the number of moves within one second exceeds the defined limit, the switch performs the configured action. We cover the configurable actions and show a configuration example later in this section.

MAC Limiting Actions



When a MAC limiting violation occurs, the switch performs one of the following actions:

- `none`: Does nothing. If you set a MAC limit to apply to all interfaces or a MAC move limit to apply to all VLANs on the EX Series switch, you can override that setting for a particular interface or VLAN by specifying an action of `none`.
- `drop`: Drops the packet and generates an alarm, an SNMP trap, or a system log entry. This is the default action for a MAC limiting violation (MAC limit or MAC move limit).
- `log`: Does not drop the packet but generates a system log entry.
- `shutdown`: Disables the port, blocks data traffic, and generates a system log entry.

Autorecovery

You can configure a switch with the **`port-error-disable`** statement to allow disabled interfaces to recover automatically upon expiration of the specified disable timeout. An example configuration using the **`port-error-disable`** statement with a specified disable timeout follows:

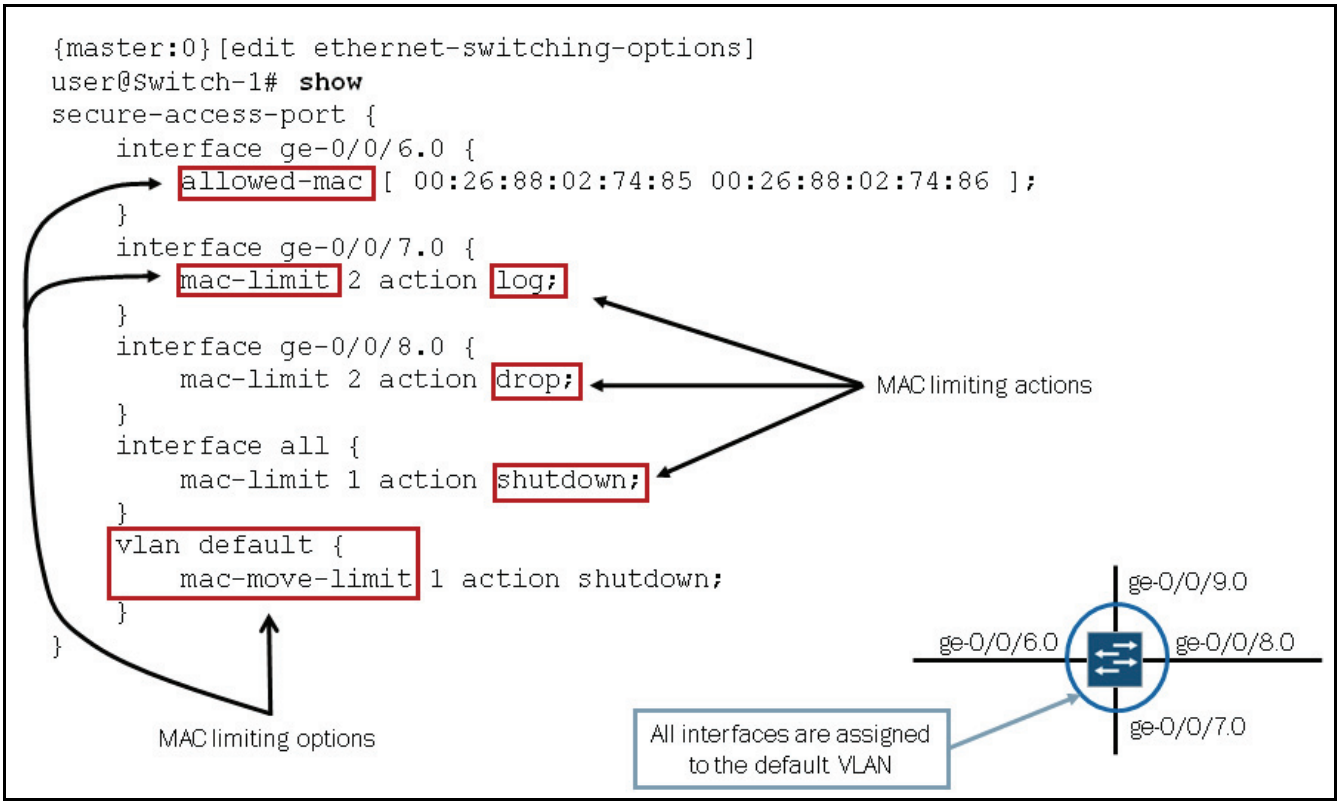
```
{master:0}[edit ethernet-switching-options]
user@Switch-1# show
port-error-disable {
    disable-timeout 3600;
}
```

This autorecovery feature is disabled by default and you must specify a valid timeout range of 10 to 3600 seconds. If you do not specify a timeout value, you will still need to manually clear the disabled port. Enabling this autorecovery option does not affect pre-existing error conditions but rather only impacts error conditions detected after the `port-error-disable` option has been enabled and committed.

If you have not configured the switch for autorecovery or you need to clear a port error disabled condition detected before the autorecovery feature was enabled, you can bring a disabled interface back into service by issuing the **`clear`**

ethernet-switching port-error interface command. We provide an example of clearing MAC limiting violations later in this section.

Configuring MAC Limiting



The graphic illustrates a sample configuration for the various MAC limiting options previously covered. In this example, we see all three enforcement methods, as well as the common actions invoked when a limit violation occurs.

As mentioned previously, in addition to the actions of `log`, `drop`, and `shutdown`, a fourth action of `none` exists. The action `none` allows you to exclude individual interfaces or VLANs from a MAC limiting configuration when the **interface all** or **vlan all** statements are used with the MAC limit or MAC move limit options respectively.

The following example illustrates the use of the `none` action in this scenario:

```
{master:0}[edit ethernet-switching-options]
user@switch# show
secure-access-port {
    interface ge-0/0/15.0 {
        mac-limit 1 action none;
    }
    interface all {
        mac-limit 1 action shutdown;
    }
    vlan all {
        mac-move-limit 1 action shutdown;
    }
    vlan default {
        mac-move-limit 1 action none;
    }
}
```

When the `interface all` or `vlan all` statements are used in conjunction with individual interface or VLAN statements, the Junos OS considers the individual interface or VLAN statements to be more specific, and they always take precedence.

Monitoring MAC Limiting

- Use `show log messages` to view violations:

```
{master:0}
user@Switch-1> show log messages | match limit | match "0/6|0/9"
...
May  7 12:52:17  Switch-1 eswd[821]: ESWD_MAC_MOVE_LIMIT_BLOCK: MAC move limit (1) exceeded
at default for MAC address 00:26:88:02:74:86; shutting down interface ge-0/0/6.0
May  7 13:14:56  Switch-1 eswd[821]: ESWD_MAC_LIMIT_BLOCK: MAC limit (1) exceeded at ge-
0/0/9.0: shutting down the interface
```

- Use `show ethernet-switching interfaces` to view interface state details:

```
{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State   VLAN members        Tag    Tagging  Blocking
ge-0/0/6.0    down    default                    untagged MAC move limit exceeded
ge-0/0/7.0    up      default                    untagged unblocked
ge-0/0/8.0    up      default                    untagged unblocked
ge-0/0/9.0    down    default                    untagged MAC limit exceeded
me0.0         up      mgmt                       untagged unblocked
```

This graphic illustrates some sample outputs used to determine the affects of the MAC limiting configuration options.

Clearing MAC Limiting Violations

```
{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State   VLAN members        Tag    Tagging  Blocking
ge-0/0/6.0    down    default                    untagged MAC move limit exceeded
ge-0/0/7.0    up      default                    untagged unblocked
ge-0/0/8.0    up      default                    untagged unblocked
ge-0/0/9.0    down    default                    untagged MAC limit exceeded
me0.0         up      mgmt                       untagged unblocked

{master:0}
user@Switch-1> clear ethernet-switching port-error interface ge-0/0/6.0

{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State   VLAN members        Tag    Tagging  Blocking
ge-0/0/6.0    up      default                    untagged unblocked
ge-0/0/7.0    up      default                    untagged unblocked
ge-0/0/8.0    up      default                    untagged unblocked
ge-0/0/9.0    down    default                    untagged MAC limit exceeded
me0.0         up      mgmt                       untagged unblocked
```

If you have not configured the switch for autorecovery from port error disabled conditions, you can bring up a disabled interfaces manually by issuing the **`clear ethernet-switching port-error interface`** command as shown on the graphic.

Persistent MAC Learning

Persistent MAC learning, also known as sticky MAC, is a port security feature that allows retention of dynamically learned MAC addresses on an interface across reboots of the switch and interface-down events. Persistent MAC address learning is disabled by default. By enabling persistent MAC learning along with MAC limiting, you can enable interfaces to learn MAC addresses of trusted workstations and servers during the period from when you connect the interface to your network until the limit for MAC addresses is reached, and ensure that after this initial period with the limit reached, new devices are not allowed even if the switch reboots. The alternatives to using persistent MAC learning with MAC limiting are to statically configure each MAC address on each port or to allow the port to continuously learn new MAC addresses after restarts or interface-down events.

If you move a device within your network that has a persistent MAC address entry on the switch, use the **clear ethernet-switching table persistent-mac** command to clear the persistent MAC address entry from the interface. If you move the device and do not clear the persistent MAC address from the original port it was learned on, then the new port will not learn the MAC address and the device will not be able to connect.

If the original port is down when you move the device, then the new port will learn the MAC address and the device can connect. However, if you do not clear the MAC address on the original port, then when the port comes back up, the system reinstalls the persistent MAC address in the forwarding table for that port. If this occurs, the address is removed from the new port and the device loses connectivity.\

Guidelines for Implementing Persistent MAC Learning

Consider the following configuration guidelines when configuring persistent MAC learning:

- Interfaces must be configured in access mode (use the port-mode configuration statement).
- You cannot enable persistent MAC learning on an interface that is part of a redundant trunk group.
- You cannot enable persistent MAC learning on an interface on which 802.1x authentication is configured.
- You cannot enable persistent MAC learning on an interface on which `no-mac-learning` is enabled.

Configuring Persistent MAC Learning

- Enable persistent MAC learning on an interface under the [edit ethernet-switching-options secure-access-port] hierarchy

```
{master:0}[edit ethernet-switching-options secure-access-port]
user@Switch-1# show
interface ge-0/0/6.0 {
    persistent-learning;
}
```

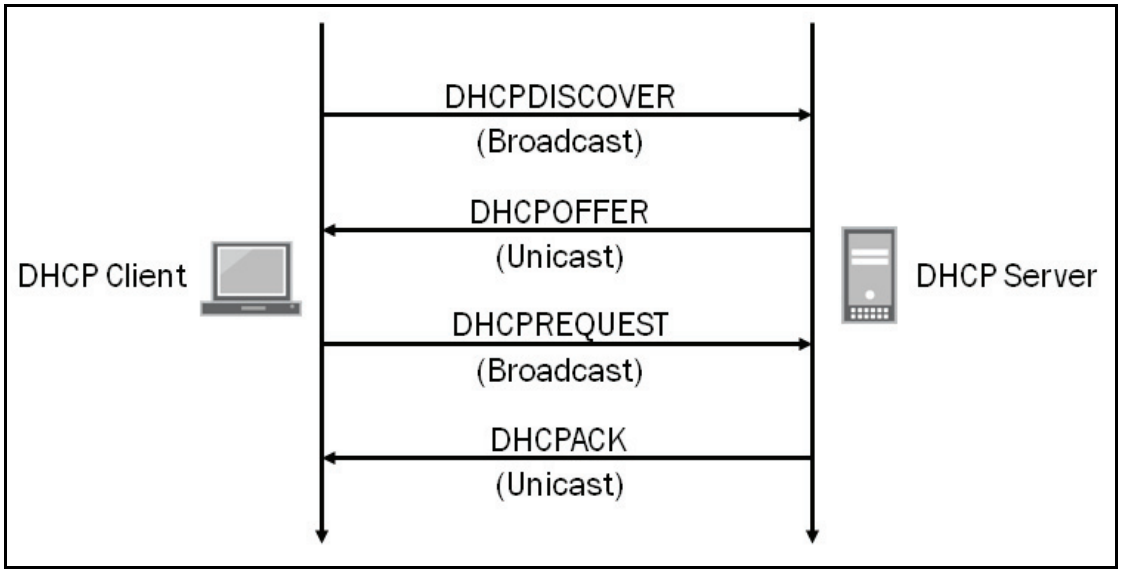
The graphic provides a sample configuration for enabling persistent MAC learning on the ge-0/0/6 interface.

Monitoring Persistent MAC Limiting

```
{master:0}
user@Switch-1> show ethernet-switching table

Ethernet-switching table: 8 entries, 2 learned, 5 persistent entries
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:10:94:00:00:02 Persistent       0 ge-0/0/6.0
  default           00:10:94:00:00:03 Persistent       0 ge-0/0/6.0
  default           00:10:94:00:00:04 Persistent       0 ge-0/0/6.0
  default           00:10:94:00:00:05 Persistent       0 ge-0/0/6.0
  default           00:10:94:00:00:06 Persistent       0 ge-0/0/6.0
  default           00:21:59:c8:0c:50 Learn            0 ae0.0
  default           02:21:59:c8:0c:44 Learn            0 ae0.0
```

As shown on the graphic, you use the **show ethernet-switching table** command to verify which learned MAC address will be retained during an interface down situation or reboot. As highlighted on the graphic, these MAC addresses will be marked `Persistent`.

DHCP Review

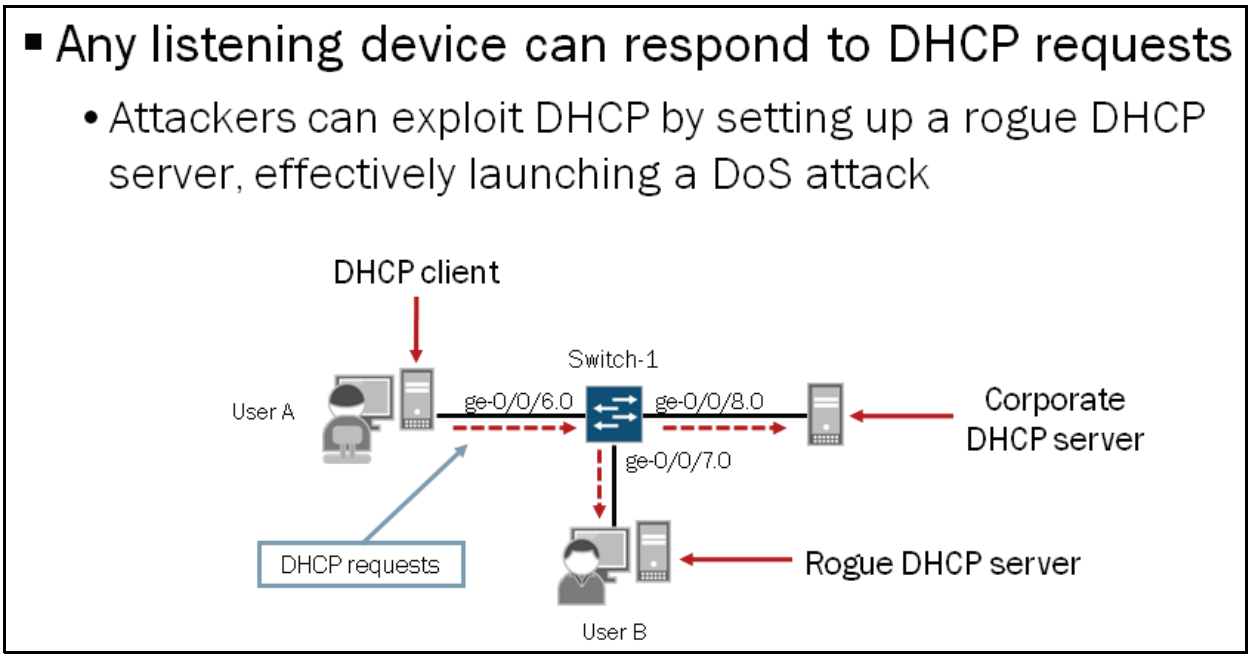


The Dynamic Host Configuration Protocol (DHCP) is used to dynamically configure hosts on a network. An administrator defines network parameters on a DHCP server. Based on individual requests from the DHCP clients, the DHCP server dynamically assigns network parameters that facilitate network access for the individual hosts, or DHCP clients. The graphic illustrates the basic communication process between DHCP clients and a DHCP server, including the various messages types sent between clients and a DHCP server.

DHCP Requests

DHCP, like many other protocols, has inherent vulnerabilities, which can be exploited either intentionally or unintentionally. When a client sends a DHCP request, it is sent as a broadcast packet and is seen by all other devices participating on the subnet.

Who Is Calling?



Because all DHCP requests can be viewed by any other device participating on the same subnet, it makes sense that any device on that subnet can also respond to that DHCP request. This inherent DHCP behavior facilitates opportunities for attackers to disrupt normal network operations and effectively launch a DoS attack. One such attack might include the use of a rogue DHCP server that responds to legitimate requests from authorized clients and provides bogus network parameters to those clients.

DHCP Snooping

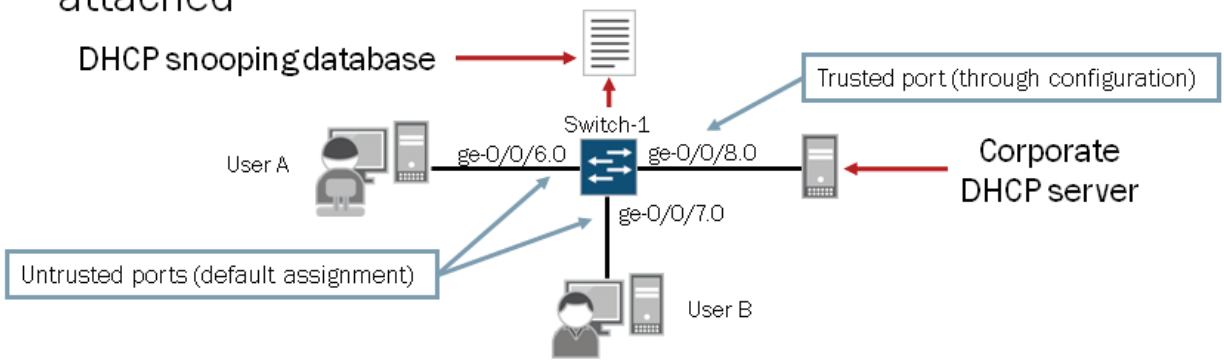
- DHCP snooping combats DHCP vulnerabilities (not always malicious) by:
  - Building and maintaining a database of valid DHCP bindings
  - Inspecting DHCP packets received on untrusted ports
    - By default, access ports are untrusted and trunk ports are trusted;

You can use DHCP snooping to combat some of the inherent DHCP vulnerabilities and protect your network and its resources. DHCP snooping builds and maintains a database of valid IP addresses assigned to downstream network devices by a trusted DHCP server. DHCP snooping reads the lease information, which is sent from the DHCP server to the individual DHCP clients. From this information it creates the DHCP snooping database. This database is a mapping between IP address, MAC address, and the associated VLAN. When a DHCP client releases an IP address (by sending a DHCPRELEASE message), the associated mapping entry is deleted from the database. The switch also tracks the lease time, as assigned by the DHCP server, and purges all expired entries.

DHCP snooping protects the switch, as well as other network components, by inspecting all DHCP packets on untrusted ports. By default, the Junos OS treats access ports as untrusted and trunk ports as trusted. If a server is connected to a local access port, you must configure that port as a trusted port to accommodate the DHCP server traffic it receives. Note that DHCP snooping occurs only on interfaces for which an entry exists. If a switch port is connected to a device with a statically defined IP address, no inspection occurs.

DHCP Option 82

- Used to identify the switch and port to which the client is attached



DHCP snooping includes support for DHCP option 82, also known as the DHCP relay agent information option. DHCP option 82 helps protect the switch against attacks such as spoofing (forging) of IP addresses and MAC addresses, as well as DHCP IP address pool exhaustion. When a DHCP client that is connected to the switch on an untrusted interface sends a DHCP request, the switch inserts information about the client's network location into the packet header of that request. The switch then sends the request to the DHCP server. The DHCP server reads the option 82 information contained in the packet header and uses it to implement the IP address or another parameter for the client. The DHCP server sends this information back toward the switch with the same option 82 information in the header. The switch removes the option 82 information in the header before sending the response packet to the client.

You can enable DHCP option 82 on a single VLAN or on all VLANs on the switch. You can also configure it on Layer 3 interfaces (routed VLAN interfaces [RVIs]) when the switch is functioning as a relay agent.

The EX Series switch implementation of option 82 contains three suboptions: circuit ID, remote ID, and vendor ID. These suboptions are configurable fields within the packet header:

- `circuit-id`: This suboption identifies the circuit (interface, VLAN, or both) on the switch on which the request was received. The circuit ID contains the interface name, the VLAN name, or both, with the two elements separated by a colon. For example, ge-0/0/10:vlan1, where ge-0/0/10 is the interface name and vlan1 is the VLAN name. If

the request packet is received on a Layer 3 interface, the circuit ID is just the interface name, for example, ge-0/0/10.

Use the `prefix` option to add an optional prefix to the circuit ID. If you enable the `prefix` option, the hostname for the switch is used as the prefix; for example, `switch1:ge-0/0/10:vlan1`, where `switch1` is the hostname.

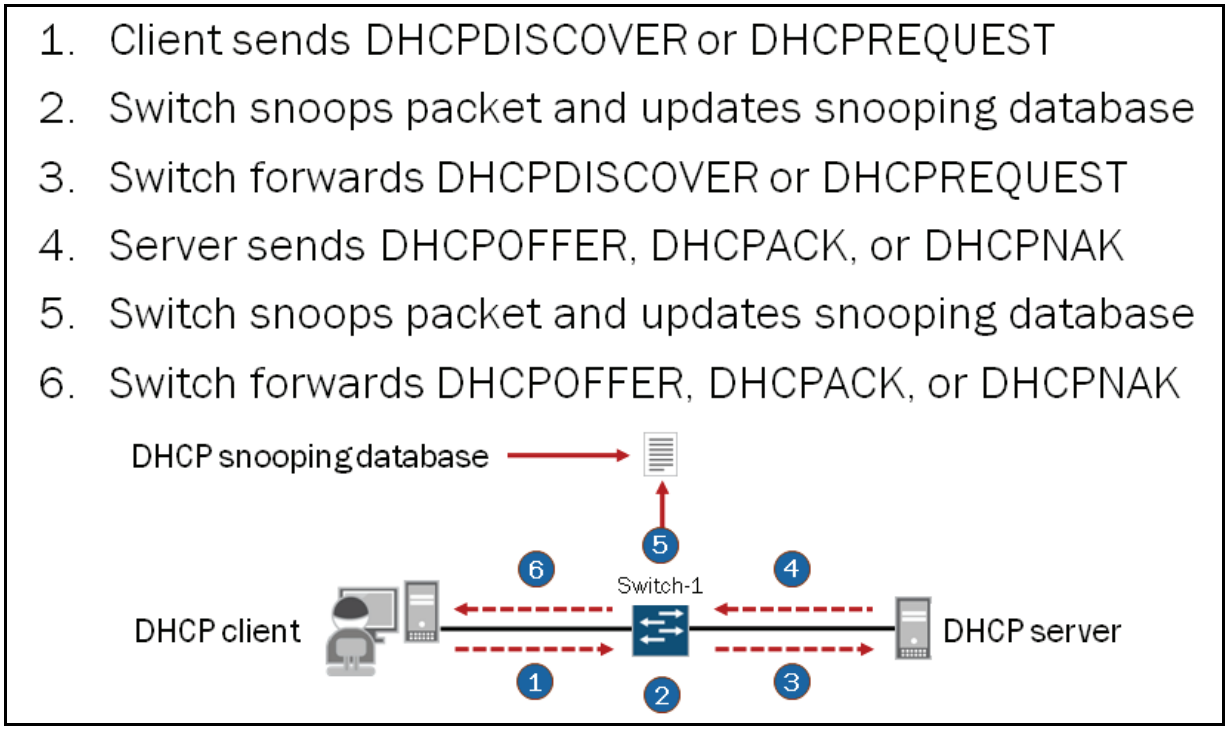
You can also specify that the interface description be used rather than the interface name, or that the VLAN ID be used rather than the VLAN name, or both.

- `remote-id`: This suboption identifies the host. By default, the remote ID is the MAC address of the switch. You can specify that the remote ID be the hostname of the switch, the interface description, or a character string of your choice. You can also add an optional prefix to the remote ID.
- `vendor-id`: This suboption identifies the vendor of the host. If you specify the vendor-id option but do not enter a value, the default value, `Juniper`, is used. To specify a value, type a character string.

Note that if you are going to use DHCP option 82, you must ensure that the DHCP server is configured to accept option 82. If it is not configured to accept option 82, then when it receives requests containing option 82 information, it does not use the information in setting parameters and it does not echo the information in its response message. If the DHCP option 82 information is not echoed back to the switch, the DHCP packets are not forwarded to the client, resulting in a DHCP failure.

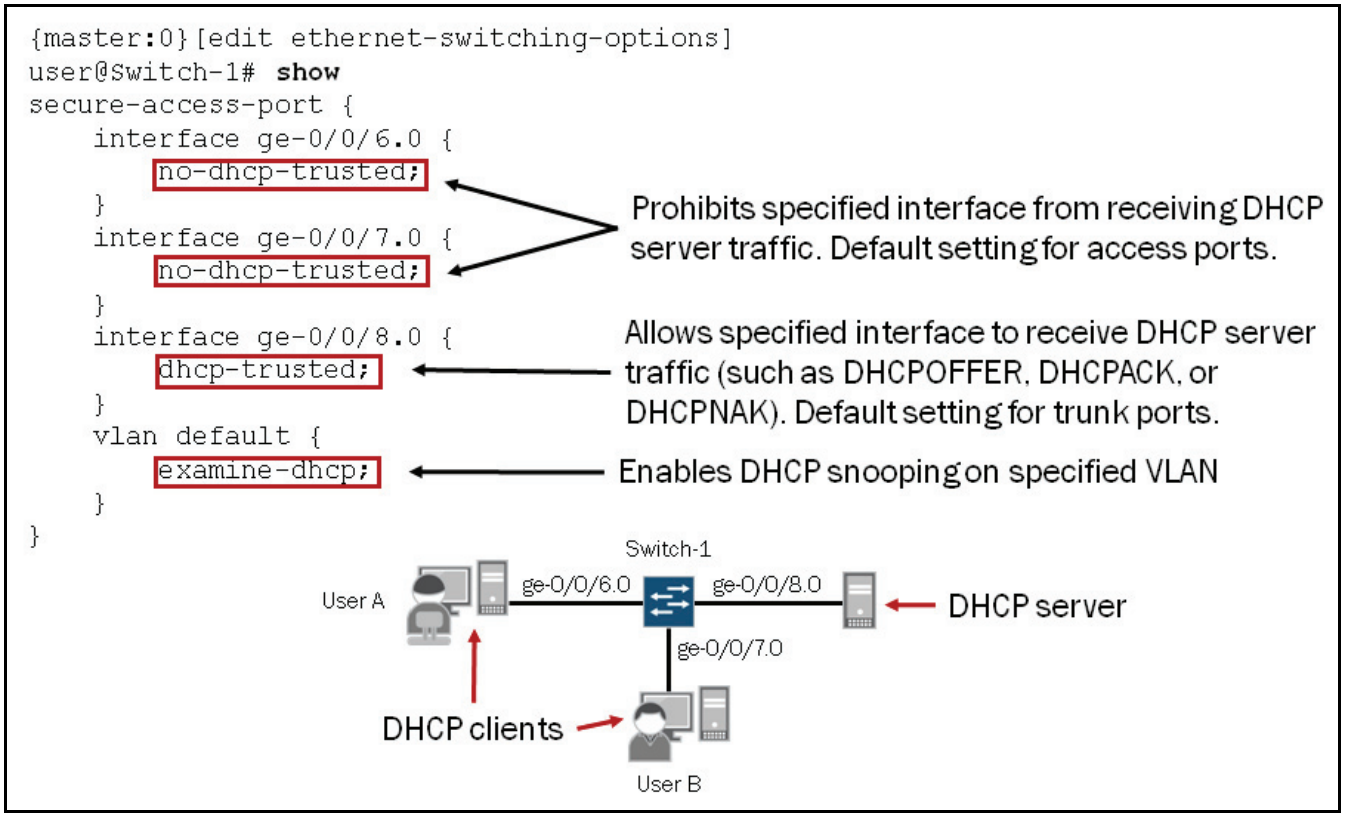
For additional information about DHCP option 82 and how to configure this feature, please refer to the technical publications at http://www.juniper.net/techpubs/.

DHCP Snooping Process



This graphic illustrates the basic steps involved with the DHCP snooping process. Note that in previous versions of the Junos OS, EX Series switches snooped DHCPDISCOVER and DHCPOFFER packets. These packets are no longer snooped in current software versions.

Configuring DHCP Snooping



This graphic provides a basic DHCP snooping configuration example. This example shows the required configuration to enlist an access interface (ge-0/0/8.0), which connects to a DHCP server, as a trusted interface, as well as how to enable DHCP snooping on an individual VLAN.

When DHCP snooping is enabled, the DHCP lease information learned by the switch is used to create the DHCP snooping database, a mapping of IP address to VLAN–MAC-address pairs. For each VLAN–MAC-address pair, the database stores the corresponding IP address.

By default, the IP-MAC bindings are lost when the switch is rebooted and DHCP clients (the network devices, or hosts) must reacquire bindings. You can configure the DHCP bindings to persist through a reboot by setting the **dhcp-snooping-file** statement. This configuration option stores the database file either locally or remotely, depending on user preference, and is configured under the `[edit ethernet-switching-options secure-access-port]` hierarchy level.

The following sample configuration illustrates the configurable options:

```
{master:0}[edit ethernet-switching-options secure-access-port]
user@switch# set dhcp-snooping-file ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  location             Location of DHCP snooping entries file
  timeout              Timeout for remote read and write operations (seconds)
  write-interval       Time interval for writing DHCP snooping entries (seconds)
```

Junos OS requires you to set the `location` to where the file entries will be logged as well as the `write-interval`. The `write-interval` statement determines the interval for which the snooping entries are written to the local or remote file. The `timeout` option determines the timeout interval for remote read and write operations when the DHCP snooping file is written to a remote server.

```
{master:0}[edit ethernet-switching-options]
user@Switch-1# show
secure-access-port {
    interface ge-0/0/8.0 {
        dhcp-trusted;
    }
```

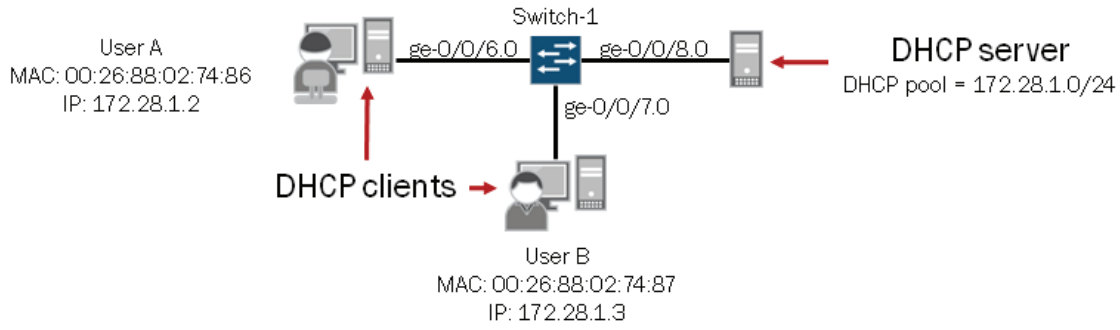
```
        vlan default {
            examine-dhcp;
        }
        dhcp-snooping-file {
            location /var/tmp/snoop-dawg-n-co;
            write-interval 60;
        }
    }
```

To view the DHCP snooping file contents, use the operational **file show** command along with the path and file name. You can also view the transfer and read and write statistics for the DHCP snooping file using the operational **show dhcp snooping statistics** command. The following output illustrates a sample output:

```
{master:0}[edit ethernet-switching-options]
user@Switch-1# run file show /var/tmp/snoop-dawg-n-co
Version : 1
00:26:88:02:74:86       172.28.1.2      Tue May 11 19:10:20 2010        ge-0/0/6.0
default 32a945b6 4054d4d7 2343702b fcdd52ee
00:26:88:02:74:87       172.28.1.3      Tue May 11 19:10:20 2010        ge-0/0/7.0
default f7dea107 cee9565a 92030701 8e348a02

{master:0}[edit ethernet-switching-options]
user@Switch-1# run show dhcp snooping statistics
DHCP Snoop Persistence statistics
Successful Remote Transfers: 0          Failed Remote Transfers: 0
Successful Record Reads    : 0          Failed Record Reads    : 0
Successful Record Writes   : 2          Failed Record Writes   : 0
```

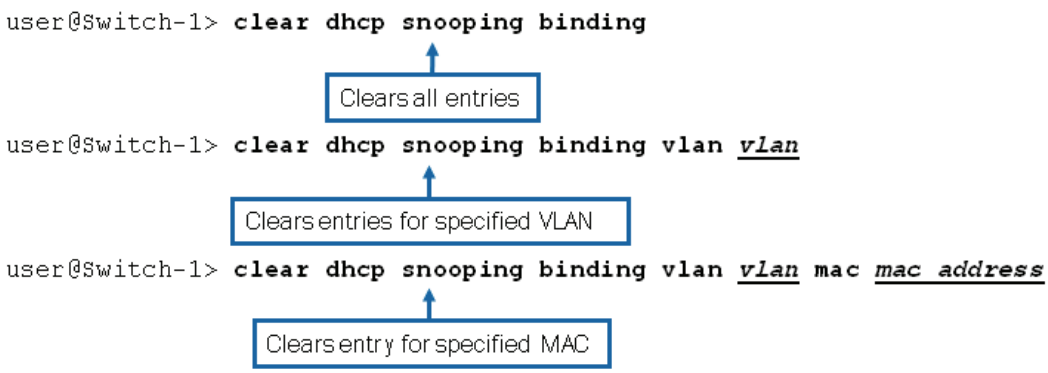
Monitoring DHCP Snooping

```
{master:0}
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address         IP address        Lease (seconds)  Type      VLAN     Interface
00:26:88:02:74:86   172.28.1.2             86113       dynamic   default  ge-0/0/6.0
00:26:88:02:74:87   172.28.1.3             86378       dynamic   default  ge-0/0/7.0
```



When DHCP snooping is enabled, the DHCP lease information learned by the switch is used to create the DHCP snooping database, a mapping of IP address to VLAN–MAC-address pairs. For each VLAN–MAC-address pair, the database stores the corresponding IP address. You use the **show dhcp snooping binding** command to view the registered details within the DHCP snooping database.

Clearing the DHCP Snooping Database



Use the **clear dhcp snooping binding** commands to clear entries within the DHCP snooping database. This command offers various options that allow the user to clear all entries, all entries for a particular VLAN, or individual entries within the DHCP snooping database. You can use the **show dhcp snooping binding** command before and after clearing database entries to monitor the results. The following output illustrates this point and clears an individual MAC address:

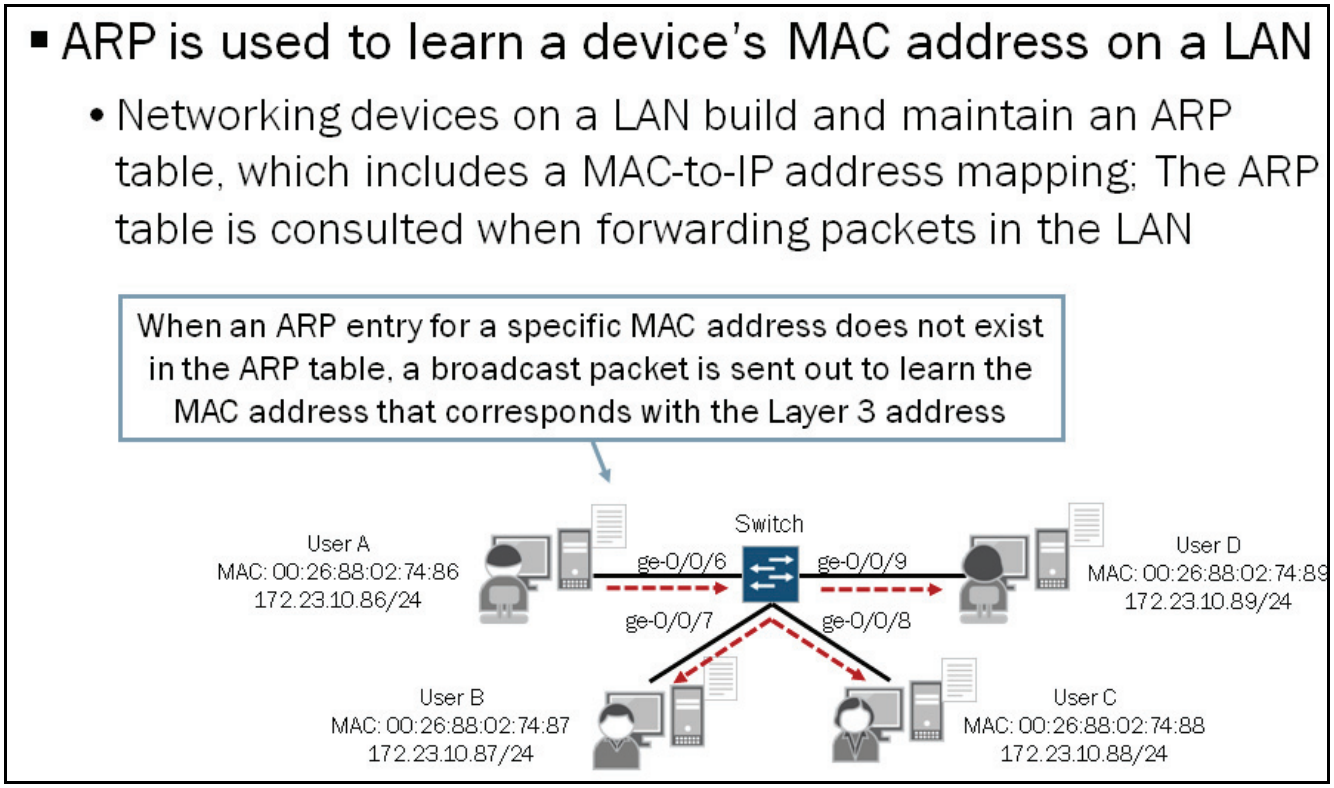
```
{master:0}
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address        Lease (seconds)  Type     VLAN    Interface
00:26:88:02:74:86  172.28.1.2                  85243 dynamic  default ge-0/0/6.0
00:26:88:02:74:87  172.28.1.3                  85243 dynamic  default ge-0/0/7.0

{master:0}
user@Switch-1> clear dhcp snooping binding vlan default mac 00:26:88:02:74:87

{master:0}
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address        Lease (seconds)  Type     VLAN    Interface
00:26:88:02:74:86  172.28.1.2                  85212 dynamic  default ge-0/0/6.0
```

Adding Static Entries

```
{master:0}[edit ethernet-switching-options]
user@Switch-1# show
secure-access-port {
    interface ge-0/0/9.0 {
        static-ip 172.28.1.4 vlan default mac 00:26:88:02:74:89;
    }
    vlan default {
        examine-dhcp;
    }
}

{master:0}[edit ethernet-switching-options secure-access-port]
user@Switch-1# run show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address        Lease (seconds)  Type     VLAN    Interface
00:26:88:02:74:89  172.28.1.4                      -  static   default ge-0/0/9.0
```

As shown on the graphic, you can add static entries to the DHCP snooping database. This might be helpful in situations where a network device does not support ARP or must have ARP disabled. To remove static DHCP snooping database entries, you must manually delete the static definition.

ARP Review



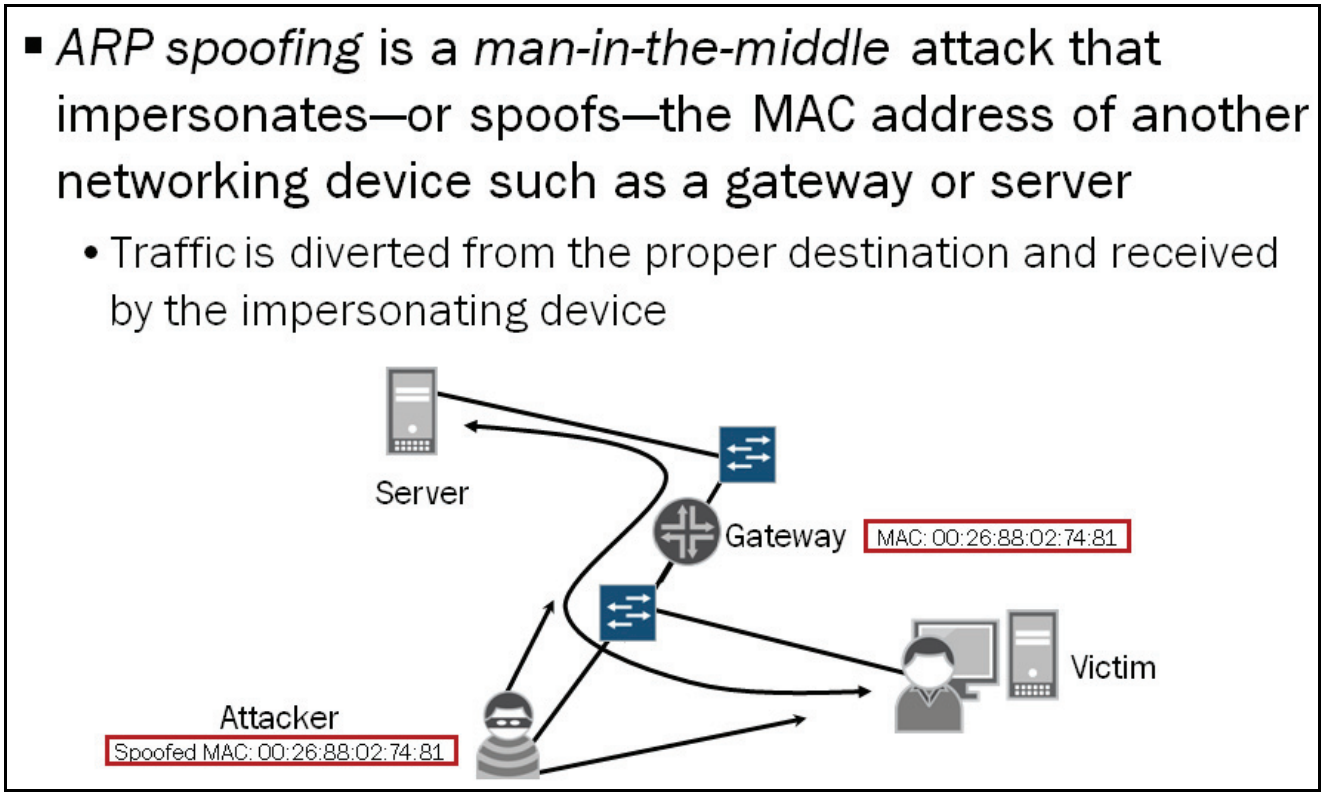
Sending IP packets on a multiaccess network requires mapping an IP address to an Ethernet MAC address. Ethernet LANs use the Address Resolution Protocol (ARP) to map MAC addresses to IP addresses.

All devices participating on an Ethernet LAN in a Layer 3 capacity maintain an ARP table with these IP address to Ethernet MAC address mappings. Each Layer 3 device participating on an Ethernet LAN maintains its ARP table in cache and consults the stored information when forwarding packets to other Layer 3 devices on the same LAN.

If the ARP cache does not contain an entry for the destination device, the host broadcasts an ARP request for that device's Ethernet MAC address and stores the response in the ARP table. An example of an ARP table follows:

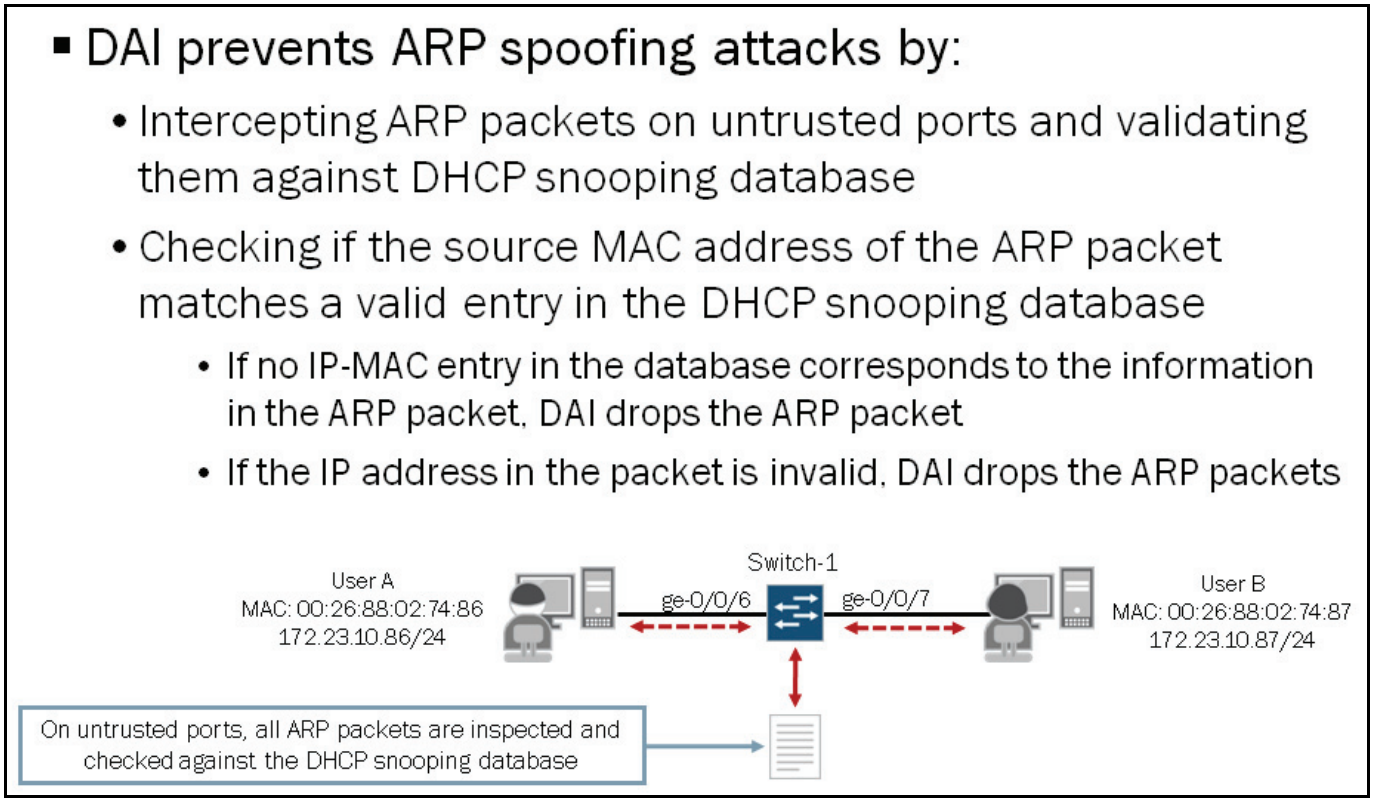
```
user@switch> show arp
MAC Address       Address         Name                      Interface     Flags
00:26:88:02:74:86 172.28.1.2      172.28.1.2                ge-0/0/8.0    none
00:26:88:02:74:87 172.28.1.3      172.28.1.3                ge-0/0/8.0    none
00:26:88:02:74:89 172.28.1.4      172.28.1.4                ge-0/0/8.0    none
Total entries: 3
```

ARP Spoofing



ARP spoofing, also known as *ARP poisoning*, is commonly used to initiate *man-in-the-middle* attacks. In these types of attacks, the attacker sends an ARP packet that spoofs the MAC address of another device on the LAN such as a gateway device or server. Instead of the switch sending traffic to the proper network device, it sends the traffic to the impersonating device with the spoofed address. The result is that traffic from the switch is diverted from the proper destination and received by the impersonating device.

Dynamic ARP Inspection



Dynamic ARP Inspection (DAI) examines ARP requests and responses on the LAN. Each ARP packet received on an untrusted access port is validated against the DHCP snooping database. By validating each ARP packet received on untrusted access ports, DAI can prevent ARP spoofing.

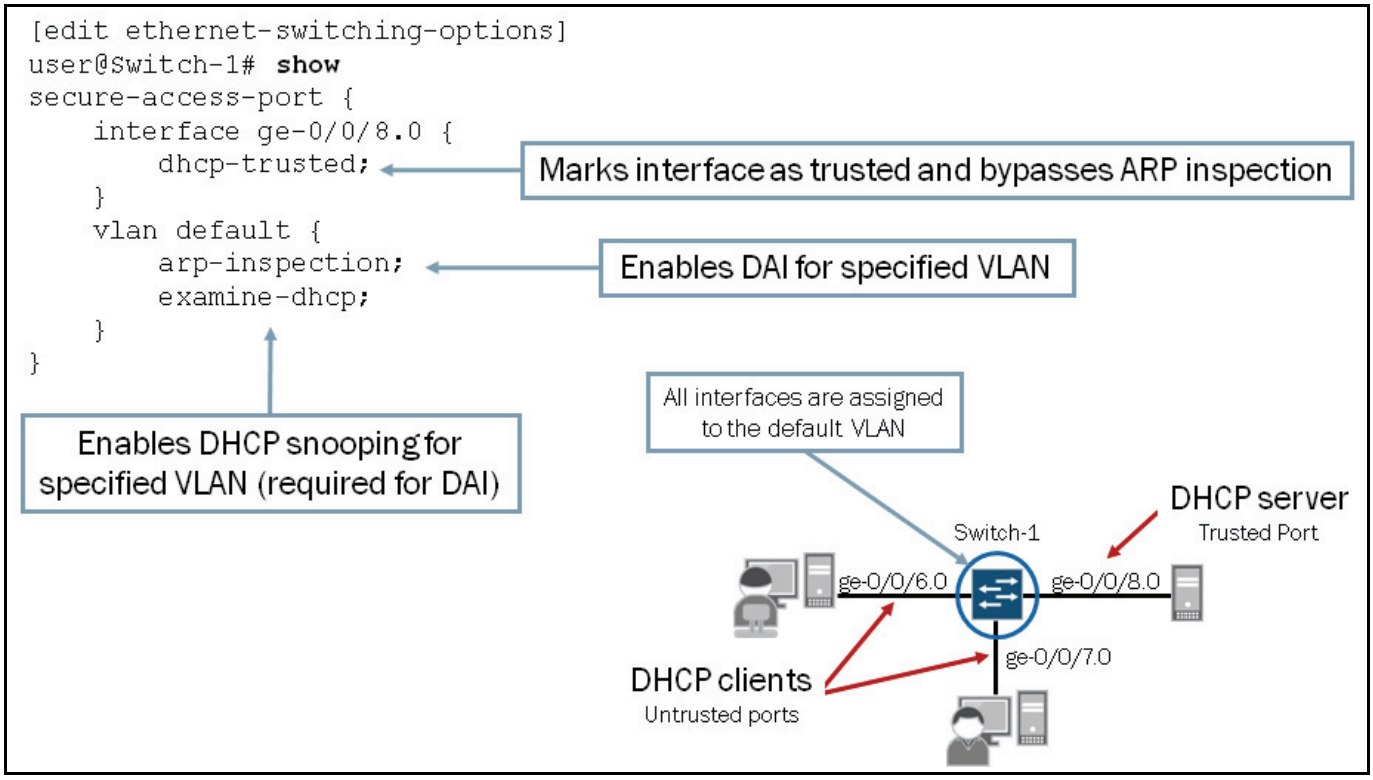
If the DHCP snooping database does not contain an IP address-to-MAC address entry for the information within the ARP packet, DAI drops the ARP packet, thus preventing the propagation of invalid host address information. DAI also drops ARP packets when the IP address in the packet is invalid. Because DAI depends on the entries found within the DHCP snooping database, you must enable DHCP snooping. DAI inspects ARP packets received on untrusted interfaces. Access ports are untrusted by default but can be changed to trusted ports through user configuration. ARP packets bypass DAI on trusted interfaces. Trunk ports are trusted by default.

By default, DAI is disabled on EX Series switches. You enable DAI on individual VLANs and not for each port. If an access port is connected to a host with a statically defined IP address within a VLAN that has DHCP snooping and DAI enabled, you must configure that port as a trusted port to allow ARP packets to pass. You can set individual ports as trusted by adding the **dhcp-trusted** option on a given port, as shown in the following example:

```
[edit ethernet-switching-options]
user@switch# show
secure-access-port {
    interface ge-0/0/8.0 {
        dhcp-trusted;
    }
}
```

Junos OS broadcasts all ARP queries directed to the switch out all ports assigned to the associated VLAN. The software subjects ARP responses of those queries to the DAI check. ARP packets are sent to and reviewed by the RE. To prevent CPU overloading, Junos OS rate-limits ARP packets destined for the RE.

Configuring DAI



This graphic illustrates a basic DAI configuration. As mentioned previously, DAI is configured on a per-VLAN basis. In the sample configuration on the graphic, DAI is enabled for the `default` VLAN.

Note that before DAI is configured, DHCP snooping must be configured. You can set the interface as dhcp-trusted, as shown on the graphic and mentioned previously.

If you have devices that do not support DHCP and choose to implement DAI, you must define a static entry in the DHCP snooping database for those devices. The following example configuration illustrates how to manually define a static DHCP snooping database entry:

```
{master:0}[edit]
user@Switch-1# show ethernet-switching-options
secure-access-port {
    interface ge-0/0/9.0 {
        static-ip 172.28.1.4 vlan default mac 00:26:88:02:74:89;
    }
    interface ge-0/0/8.0 {
        dhcp-trusted;
    }
    vlan default {
        arp-inspection;
        examine-dhcp;
    }
}
```

Monitoring DAI

- Use the **show dhcp snooping binding** command to view DHCP snooping database details:

```
{master:0}
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address        Lease (seconds)  Type     VLAN    Interface
00:26:88:02:74:86  172.28.1.2                  86113  dynamic  default ge-0/0/6.0
00:26:88:02:74:87  172.28.1.3                  86378  dynamic  default ge-0/0/7.0
```

- Use the **show arp inspection statistics** command to view DAI statistics:

```
{master:0}
user@Switch-1> show arp inspection statistics | match "Interface|0/6|0/7|0/8"
Interface     Packets received     ARP inspection pass  ARP inspection failed
  ge-0/0/6                   4                       4                      0
  ge-0/0/7                   4                       4                      0
  ge-0/0/8                   5                       5                      0
```

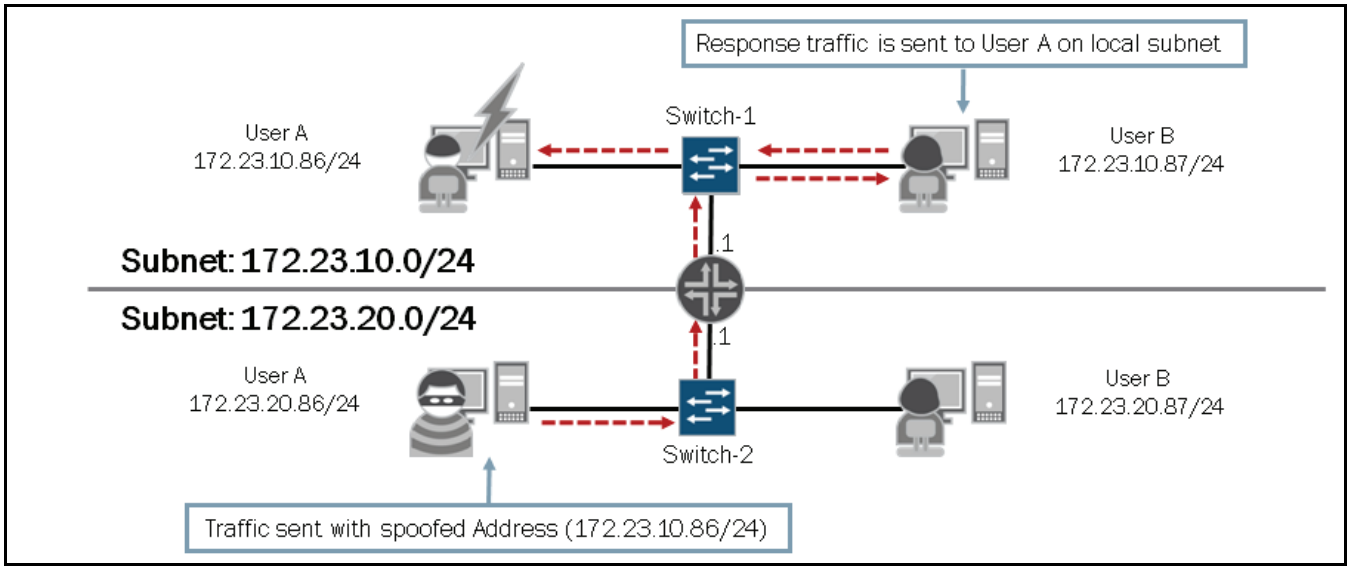
All ARP packets have passed inspection

This graphic highlights some key commands used to monitor the operation of DAI. Use the **show dhcp snooping binding** command to view the recorded details within the DHCP snooping database. Use the **show arp inspection statistics** command to view DAI statistics.

If you have included a static entry in the DHCP snooping database, that entry will show a type of `static` rather than `dynamic`. The following capture illustrates a static entry:

```
{master:0}
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address    Lease (seconds) Type     VLAN    Interface
00:26:88:02:74:86  172.28.1.2      86277         dynamic  default ge-0/0/6.0
00:26:88:02:74:87  172.28.1.3      86277         dynamic  default ge-0/0/7.0
00:26:88:02:74:89  172.28.1.4        -           static   default ge-0/0/9.0
```

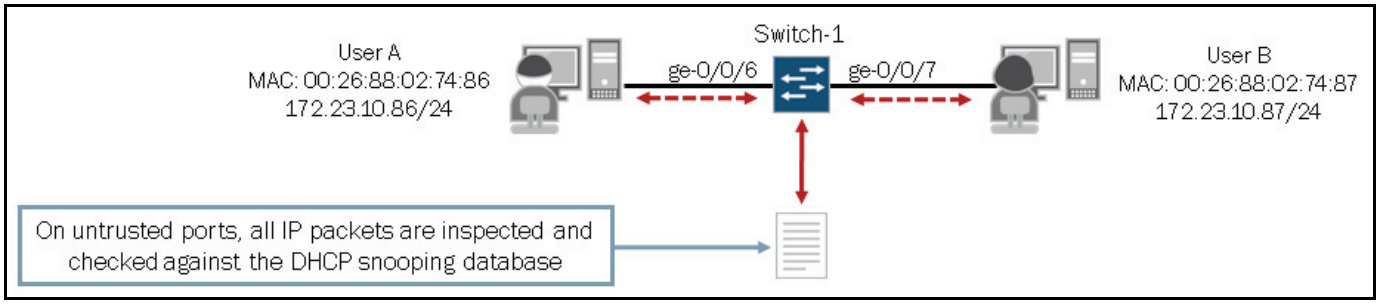
IP Address Spoofing



Users can change or spoof source IP addresses and/or source MAC addresses by flooding the switch with packets containing invalid addresses. Combined with other techniques, such as TCP SYN flood attacks, address spoofing can deny legitimate service and render a network useless.

Identifying the source of an attack that uses source IP address or source MAC address spoofing can be difficult for system administrators. As illustrated on the graphic, attackers can spoof addresses on the same subnet or on a different subnet.

IP Source Guard

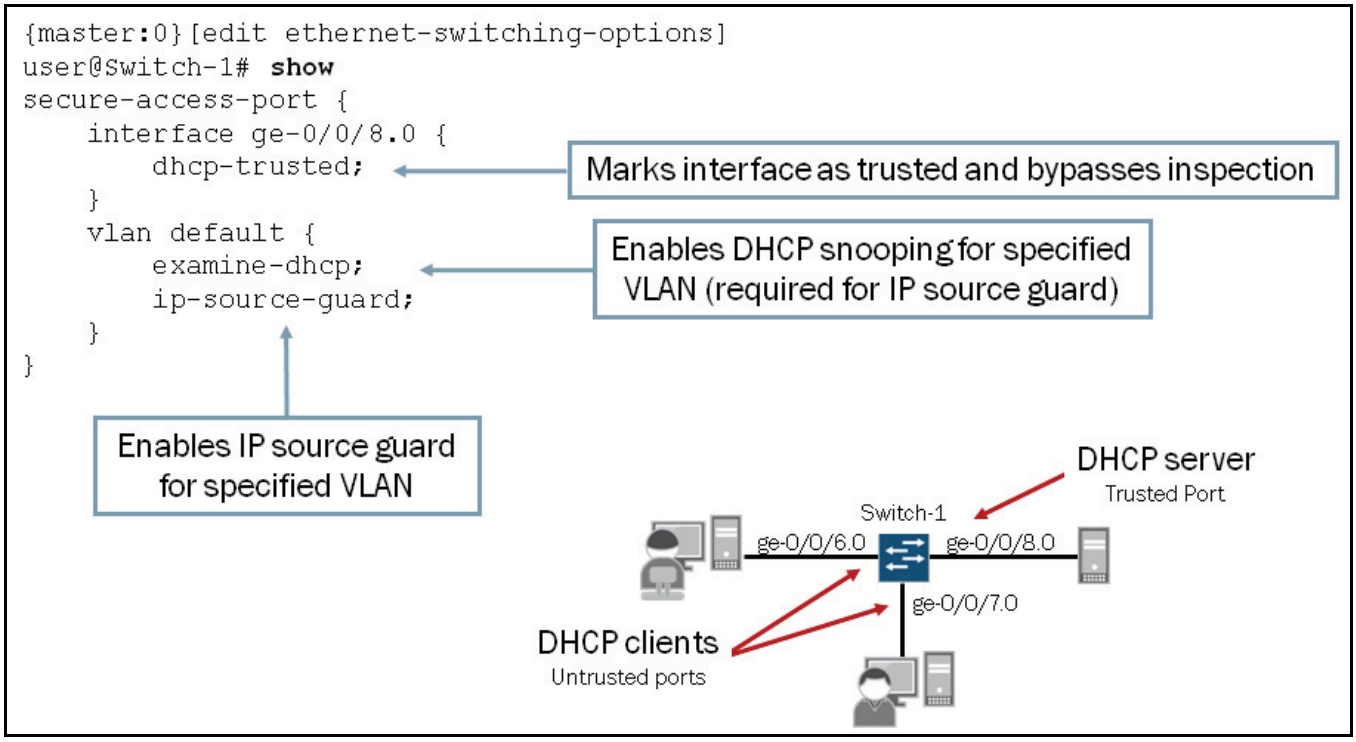


A switch, with the IP source guard feature enabled, checks the source IP and MAC addresses in a packet entering untrusted access interfaces against the entries stored in the DHCP snooping database. If IP source guard determines that the packet header contains an invalid source IP or MAC address, the switch does not forward the packet—that is, the packet is discarded.

You can enable IP source guard on one or more VLANs. IP source guard applies its checking rules to packets sent from untrusted access interfaces on those VLANs. By default, on EX Series switches, access interfaces are untrusted and trunk interfaces are trusted. IP source guard does not check packets that have been sent to the switch by devices connected to either trunk interfaces or trusted access interfaces.

IP source guard obtains information about IP-address/MAC-address/VLAN bindings from the DHCP snooping database. It causes the switch to validate incoming IP packets against the entries in that database. After the DHCP snooping database has been populated either through dynamic DHCP snooping or through configuration of specific static IP address/MAC address bindings, the IP source guard feature builds its database. It then checks incoming packets from access interfaces on the VLANs on which it is enabled. If the source IP addresses and source MAC addresses match the IP source guard binding entries, the switch forwards the packets to their specified destination addresses. If no matches are found, the switch discards the packets.

Configuring IP Source Guard



This graphic illustrates the required configuration to enable the IP source guard feature. As mentioned previously, IP source guard is configured on a per-VLAN basis. In the sample configuration on the graphic, IP source guard is enabled for the `default` VLAN.

If you have devices that do not support DHCP and choose to implement IP source guard, you must define a static entry in the DHCP snooping database for those devices. The following example configuration illustrates how to manually define a static DHCP snooping database entry:

```
{master:0}[edit ethernet-switching-options]
user@Switch-1# show
secure-access-port {
    interface ge-0/0/8.0 {
        dhcp-trusted;
    }
    interface ge-0/0/9.0 {
        static-ip 172.28.1.4 vlan default mac 00:26:88:02:74:89;
    }
    vlan default {
        examine-dhcp;
        ip-source-guard;
    }
}
```

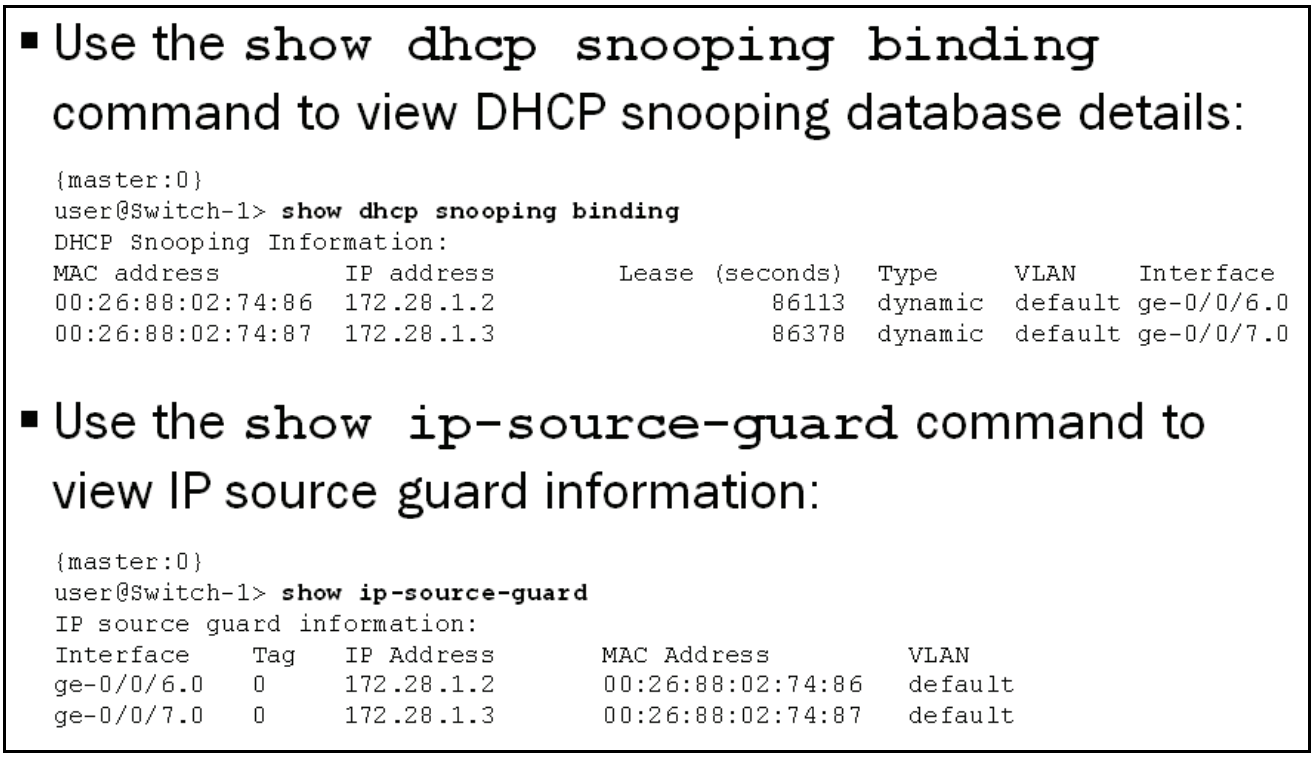
You can use the **`no-ip-source-guard`** and **`no-examine-dhcp`** statements to disable IP source guard and DHCP snooping respectively for a specific VLAN after you have enabled those features for all VLANs. A configuration example that uses these statements follows:

```
{master:0}[edit ethernet-switching-options]
user@switch# show
secure-access-port {
    vlan all {
        examine-dhcp;
        ip-source-guard;
    }
    vlan default {
```

```
            no-examine-dhcp;
            no-ip-source-guard;
        }
    }
```

You can configure IP source guard with various other features on EX Series switches to provide increased access port security. One such feature, which provides end-user authentication services, is 802.1X. The 802.1X user authentication feature is applied in one of three modes: single supplicant, single-secure supplicant, or multiple supplicant. Single supplicant mode works with IP source guard, but single-secure and multiple supplicant modes do not. Complete coverage of 802.1X is outside the scope of this study guide.

Monitoring IP Source Guard



This graphic highlights some key commands used to monitor the operation of IP source guard. Use the **show dhcp snooping binding** command to view the recorded details within the DHCP snooping database. Use the **show ip-source-guard** command to view IP source guard information. Remember that the IP source guard database is created based on the contents of the DHCP snooping database. For this reason the output displayed when issuing these two commands is nearly identical.

The IP source guard database table contains the VLANs enabled for IP source guard, the untrusted access interfaces on those VLANs, the VLAN 802.1Q tag IDs if any exist, and the IP addresses and MAC addresses that are bound to one another. If a switch interface is associated with multiple VLANs and some of those VLANs are enabled for IP source guard and others are not, the VLANs that are not enabled for IP source guard have an asterisk (*) in the IP Address and MAC Address fields.

Static entries added to the DHCP snooping database show a type of `static` rather than `dynamic`. The following capture illustrates a static entry:

```
user@Switch-1> show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address   Lease (seconds) Type     VLAN    Interface
00:26:88:02:74:86  172.28.1.2     86277         dynamic  default ge-0/0/6.0
00:26:88:02:74:87  172.28.1.3     86277         dynamic  default ge-0/0/7.0
00:26:88:02:74:89  172.28.1.4       -           static   default ge-0/0/9.0
```

No type indication exists for entries in the IP source guard database:

```
user@Switch-1> show ip-source-guard
IP source guard information:
```

```
Interface    Tag   IP Address       MAC Address         VLAN
ge-0/0/6.0   0     172.28.1.2       00:26:88:02:74:86   default
ge-0/0/7.0   0     172.28.1.3       00:26:88:02:74:87   default
ge-0/0/9.0   0     172.28.1.4       00:26:88:02:74:89   default
```

Review Questions

1. How does MAC limiting protect against MAC flooding and MAC spoofing?
2. What is the purpose of DHCP snooping, and how does it work?
3. What is spoofing, and what features are available to protect against it?

Answers

1.

MAC limiting can restrict the number of MAC addresses learned by a given port, limit the number of times a MAC address can move between ports, or specify which MAC addresses are learned on a given port.

2.

DHCP snooping protects network resources from attacks by inspecting DHCP packets received on untrusted ports. To aid in this effort, DHCP snooping builds and maintains a database of valid IP address-to-MAC address bindings called the DHCP snooping database.

3.

Spoofing is where one device impersonates another device. This can include impersonating a MAC address or an IP address. EX Series switches support DAI and IP source guard to help combat spoofing.

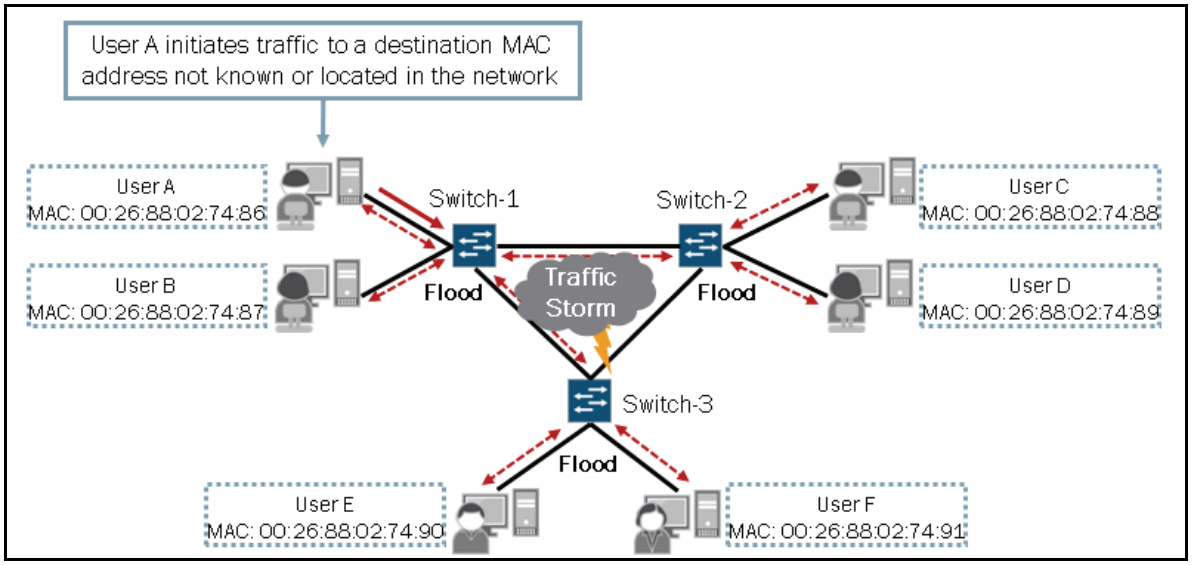
JNCIS-ENT Switching Study Guide

Chapter 5: Device Security and Firewall Filters

This Chapter Discusses:

- Traffic storms and the storm control feature;
- The configuration and monitoring of the storm control feature;
- Firewall filter support for the EX Series switches; and
- The implementation and monitoring of firewall filters.

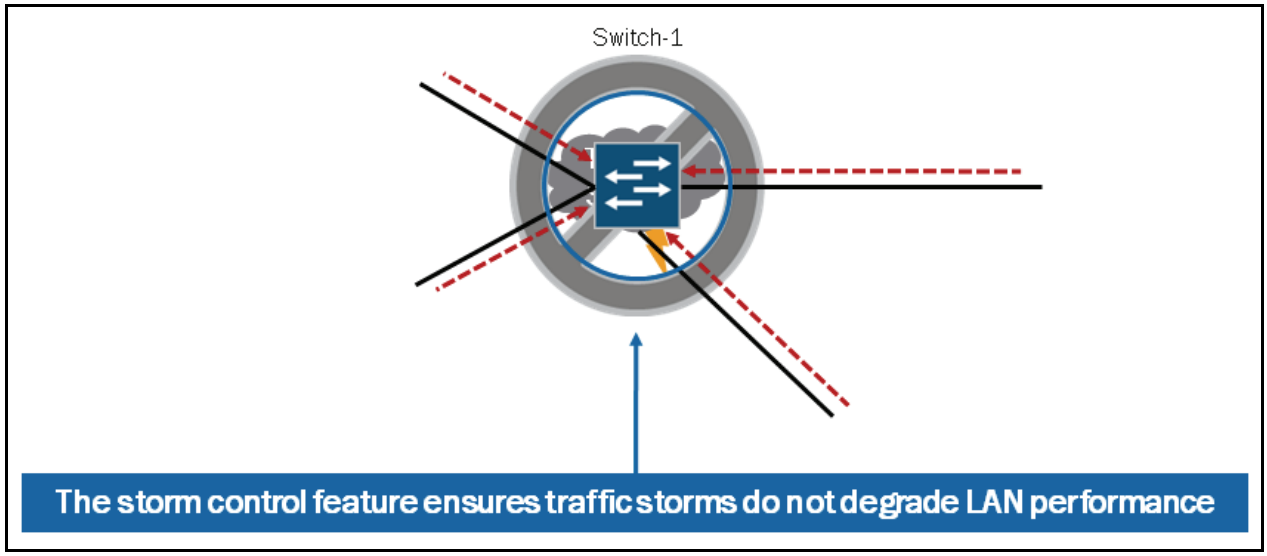
Traffic Storms



A traffic storm is generated when certain types of traffic (broadcast, multicast, and unknown unicast) is flooded throughout a network at a level significant enough that network resources and the end-users' experience is negatively affected. Some traffic prompts a receiving node to respond by broadcasting its own messages on the network. This, in turn, prompts further responses, creating a snowball effect. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service.

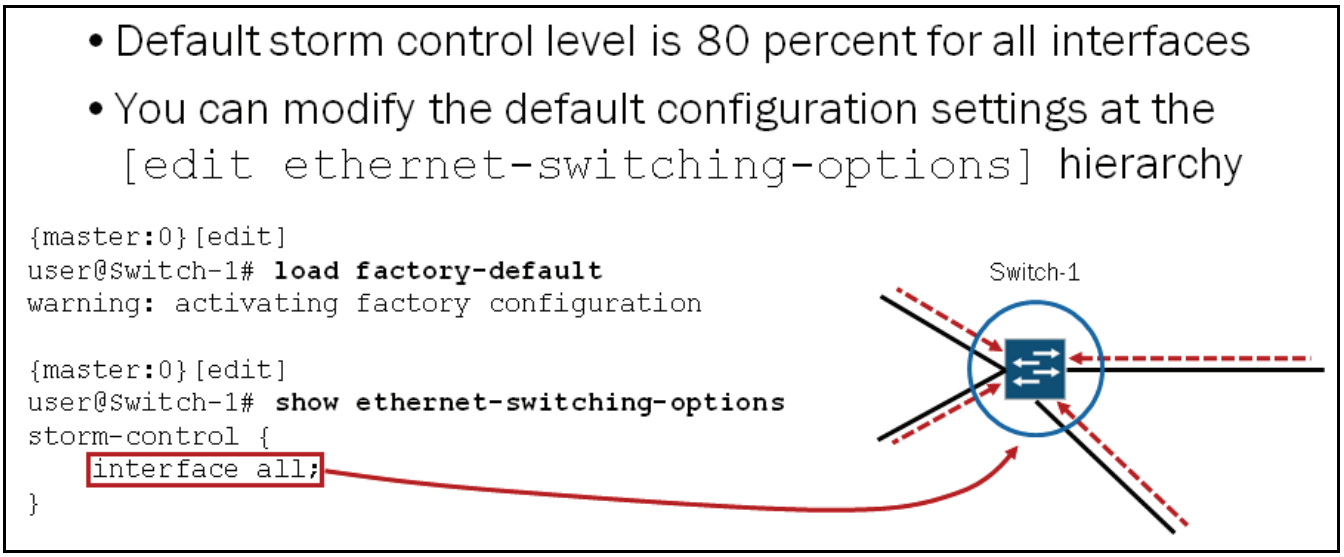
Broadcast, multicast, and unicast packets are part of normal LAN operations. To recognize a traffic storm, you must be able to identify when traffic has reached a level that is abnormal for your LAN. You should suspect a storm when operations begin timing out and network response times slow down. As more packets flood the LAN, network users might be unable to access servers or e-mail.

Storm Control



Storm control enables the switch to monitor traffic levels and to drop broadcast, multicast, and unknown unicast packets when a specified traffic level—called the storm control level—is exceeded. By dropping packets that contribute to a traffic storm, a switch can prevent those packets from proliferating and degrading the LAN.

Storm Control Configuration



Storm control enables you to prevent network outages caused by broadcast storms on the LAN. You can configure storm control on EX Series switches to rate limit broadcast traffic, multicast, and unknown unicast traffic at a specified level and to drop packets when the specified traffic level is exceeded, thus preventing packets from proliferating and degrading the LAN

The factory default configuration enables storm control on all switch interfaces, with the storm control level set to 80 percent of the combined broadcast, multicast, and unknown unicast streams. You can change the storm control level for an interface by specifying a bandwidth value for the combined broadcast, multicast, and unknown unicast traffic streams. You can also selectively disable storm control for broadcast, multicast, or unknown unicast streams.

Continued on next page.

Storm Control Configuration (contd.)

The following capture illustrates the commands used to disable storm control for broadcast, multicast, and unknown unicast streams:

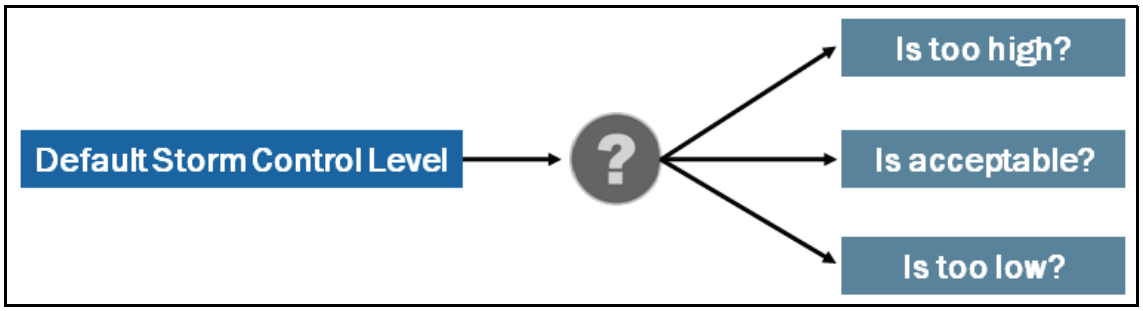
```
{master:0}[edit ethernet-switching-options storm-control]
user@Switch-1# set interface all ?
```



```
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
  bandwidth            Link bandwidth (100..10000000 kbps)
  no-broadcast         Disable broadcast storm control
  no-multicast         Disable multicast storm control
  no-unknown-unicast   Disable unknown unicast storm control
  |                    Pipe through a command
```

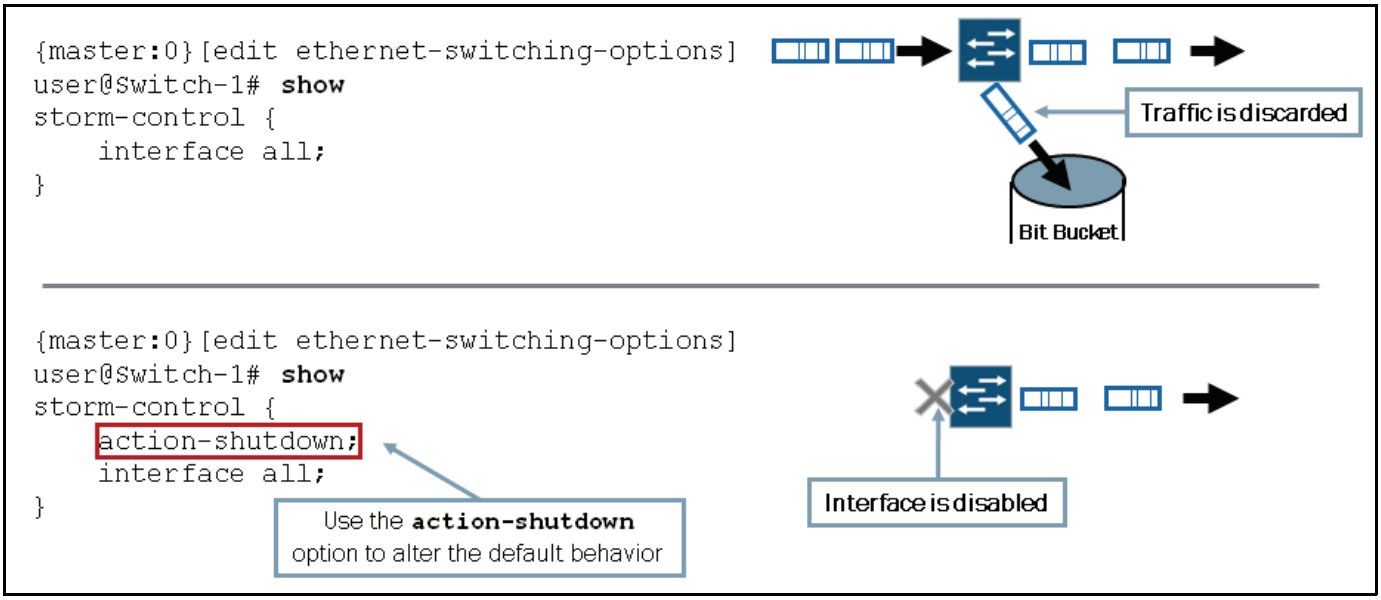
When you configure storm control bandwidth on an aggregated Ethernet interface, the storm control level for each member of the aggregated Ethernet interface is set to that bandwidth. For example, if you configure a storm control bandwidth of 15000 Kbps on ae1, and ae1 has two member links, ge-0/0/0 and ge-0/0/1, each member has a storm control level of 15000 Kbps. Thus, the storm control level on ae1 allows a traffic rate of up to 30000 Kbps of combined broadcast, multicast, and unknown unicast traffic.

Changing the Default Storm Control Configuration



Broadcast, multicast, and unicast packets are part of normal LAN operation, so to recognize a storm, you must be able to identify when traffic has reached a level that is abnormal for your LAN. Before altering the default storm control configuration you should monitor the level of broadcast, multicast, and unknown unicast traffic in the LAN when it is operating normally. You should then use this data as a benchmark to determine when traffic levels are too high. Once the benchmark data has been compiled and evaluated, you should then configure storm control and set the level at which you want to drop broadcast traffic, multicast, unknown unicast traffic, or all three.

Storm Control Actions



By default, when the storm control level is exceeded the switch drops all offending traffic. You can alter the default behavior so that interfaces through which a storm control level violation occurs are shut down. The graphic illustrates the required configuration to alter storm control's default action.

Automatic Recovery from an Error Condition

```
{master:0}[edit ethernet-switching-options]
user@Switch-1# show
port-error-disable {
    disable-timeout 300;
}
storm-control {
    action-shutdown;
    interface all;
}
```

Specify a disable timeout value between 10 and 3600 seconds

By default, when the storm control level is exceeded, the switch drops unknown unicast, multicast, and broadcast messages on the specified interfaces. If you set the **action-shutdown** and the **port-error-disable** statements, an interface through which the control level has been exceeded is disabled temporarily and recovers automatically when the disable timeout expires.

If you set the **action-shutdown** statement and do not specify the **port-error-disable** statement, interfaces through which violations occur are shut down when the storm control level is exceeded and they do not recover automatically from the port-error condition. In this situation, you must issue the **clear ethernet-switching port-error** command to clear the port error and restore the interfaces to service.

Monitoring Automatic Recovery

- Using **show ethernet-switching interfaces** to view interface state details:

```
{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State  VLAN members        Tag   Tagging  Blocking
ge-0/0/6.0    up     v11                 11    untagged unblocked
ge-0/0/8.0    up     v11                 11    tagged   unblocked
ge-0/0/9.0    down   v11                 11    tagged   Storm control in effect
                                                        (00:03:57) remaining
me0.0         up     mgmt                      untagged unblocked
```

- Using **show log messages** to view violation details:

```
{master:0}
user@Switch-1> show log messages | match storm | match ge-0/0/9
Jul 29 09:38:23  Switch-1 eswd[856]: ESWD_ST_CTL_ERROR_DISABLED: ge-0/0/9.0: storm control
disabled port
Jul 29 09:43:23  Switch-1 eswd[856]: ESWD_ST_CTL_ERROR_ENABLED: ge-0/0/9.0: storm control
enabled port
```

Interface was re-enabled after disable timeout period (5 minutes)

This graphic illustrates some key commands when monitoring storm control and its automatic recovery option. In this example you can see that the ge-0/0/9.0 interface was shut down because the storm control level was exceeded. After the disable timeout period (5 minutes in this example) the interface was re-enabled. You can confirm the interface has been re-enabled by simply issuing the **show ethernet-switching interfaces** command once again, as shown in the following capture:

```
{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State  VLAN members        Tag   Tagging  Blocking
ge-0/0/6.0    up     v11                 11    untagged unblocked
ge-0/0/8.0    up     v11                 11    tagged   unblocked
ge-0/0/9.0    up     v11                 11    tagged   unblocked
me0.0         up     mgmt                      untagged unblocked
```

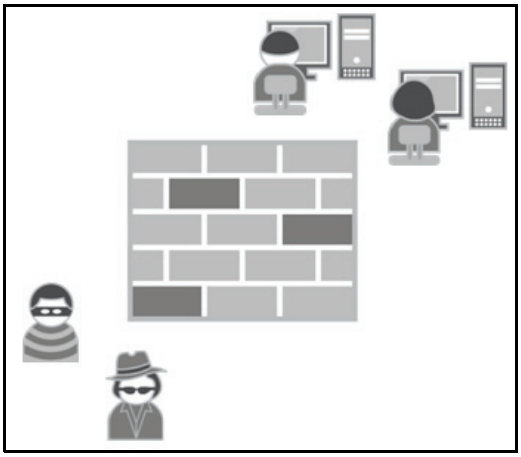
Clearing Violations Manually

- Use `clear ethernet-switching port-error interface` to clear violations manually:

```
{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State  VLAN members         Tag   Tagging  Blocking
ge-0/0/6.0    up     v11                  11    untagged unblocked
ge-0/0/8.0    up     v11                  11    tagged   unblocked
ge-0/0/9.0    down   v11                  11    tagged   Storm control in effect
                                                         (00:04:17) remaining
me0.0         up     mgmt                       untagged unblocked

{master:0}
user@Switch-1> clear ethernet-switching port-error interface ge-0/0/9

{master:0}
user@Switch-1> show ethernet-switching interfaces
Interface     State  VLAN members         Tag   Tagging  Blocking
ge-0/0/6.0    up     v11                  11    untagged unblocked
ge-0/0/8.0    up     v11                  11    tagged   unblocked
ge-0/0/9.0    up     v11                  11    tagged   unblocked
me0.0         up     mgmt                       untagged unblocked
```

This graphic illustrates the process and command used to manually clear a storm control violation.

A Review of Firewall Filters



Firewall filters are often referred to as access control lists (ACLs) by other vendors. The Junos firewall filters are stateless in nature and are used by the software to control traffic passing through a Junos device.

Stateless firewall filters examine each packet individually. Thus, unlike a stateful firewall that tracks connections and allows you to specify an action to take on all packets within a flow, a stateless firewall filter has no concept of connections. The stateless nature of these filters can impact the way you write your firewall filters. Because the system does not keep state information on connections, you must explicitly allow traffic in both directions for each connection that you want to permit. By contrast, stateful firewall filters (not supported on EX Series switches) only require you to permit the initial connection and then automatically permit bidirectional communications for this connection.

You can use firewall filters to restrict certain types of traffic from passing into and out of your network. You can also use firewall filters to perform monitoring tasks that help you formulate an effective security strategy for your environment.

Unlike some other vendors, EX Series switches always perform firewall filter checks in hardware. Firewall filters are programmed in the Packet Forwarding Engine (PFE) ternary content addressable memory (TCAM). Because firewall filters are implemented in hardware rather than a software process, the result is a very efficient match and enforcement rate when performing packet filtering operations.

Firewall Filter Types

Filter Type	Application Description
Port-based	Applied to Layer 2 switch ports in ingress and egress directions
VLAN-based	Applied to Layer 2 VLANs in the ingress and egress directions
Router-based	Applied to Layer 3 routed interfaces in ingress and egress directions

```
{master:0}[edit firewall]
user@Switch-1# edit family ?
Possible completions:
> any                   Protocol-independent filter
> ethernet-switching    Protocol family Ethernet Switching for firewall filter
> inet                  Protocol family IPv4 for firewall filter
> inet6                 Protocol family IPv6 for firewall filter
```

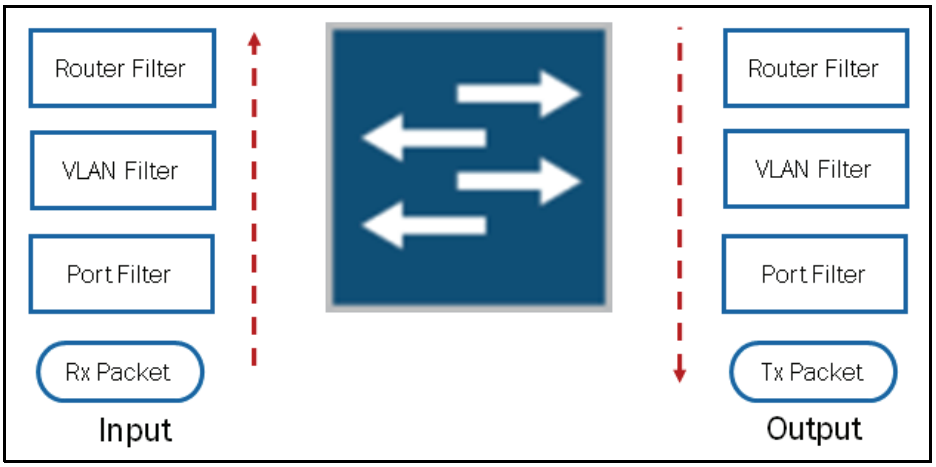
Port-based and VLAN-based filters use **family ethernet-switching** option while router-based filters use **family inet** or **family inet6** depending on the traffic type

EX Series switches support three types of firewall filters, port-based, VLAN-based, and router-based filters. Port-based and VLAN-based filters are configured using the **family ethernet-switching** option, as shown on the graphic. These filter types are Layer 2 in nature but can call upon elements found within the higher layers of the TCP/IP protocol stack. You apply port-based filters to Layer 2 switch ports and VLAN-based filters to the desired VLAN.

Router-based filters are Layer 3 in nature and are configured using the **family inet** or **family inet6** depending on the type of traffic traversing the network. Router-based filters are applied to Layer 3 interfaces, including physical ports defined for Layer 3 operations, aggregated Ethernet interfaces operating in a Layer 3 capacity, loopback interfaces, and RVIs. Unlike some Junos devices, firewall filters applied to the loopback interface do not affect traffic traversing the management Ethernet interface (me0).

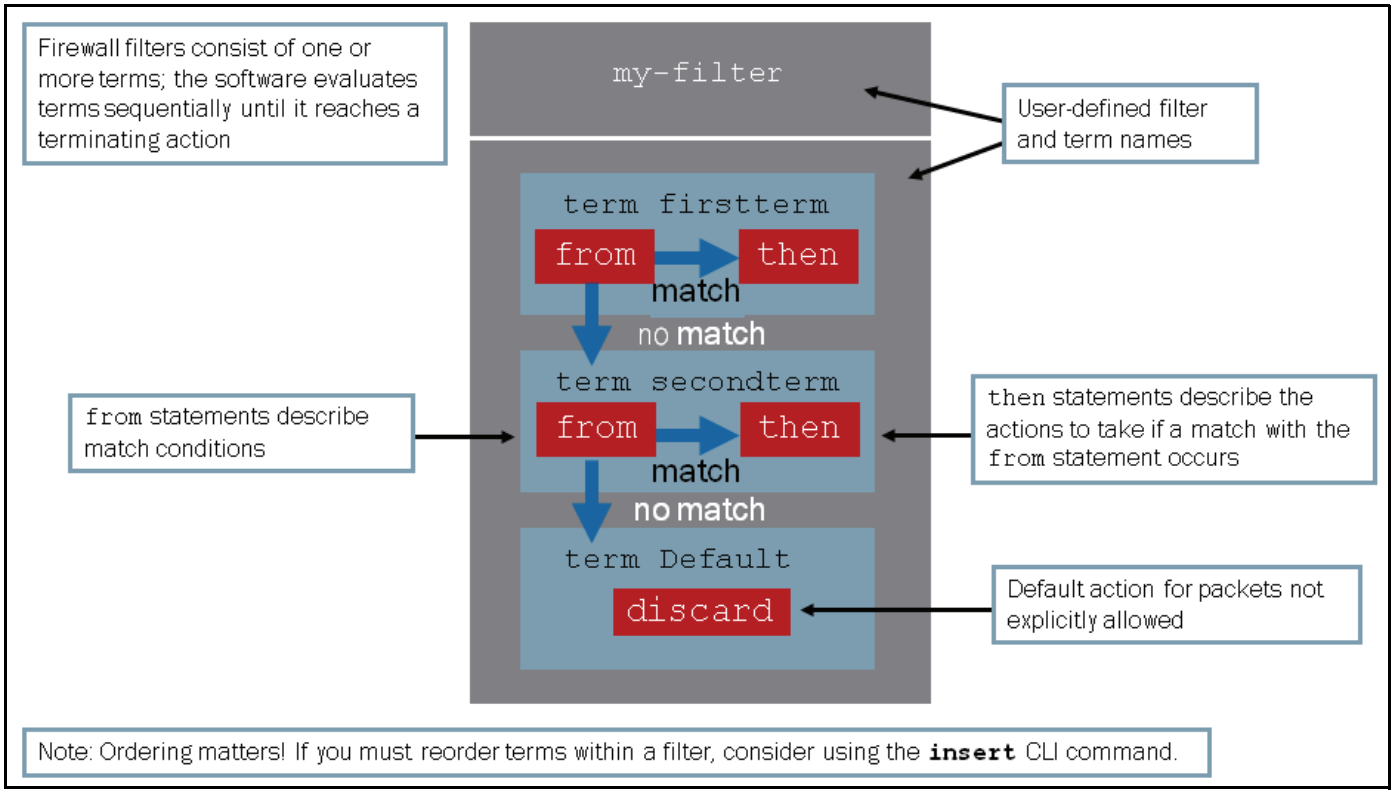
You can configure and apply no more than one firewall filter per port, VLAN, or router interface, per direction. The supported number of firewall filter terms allowed per filter varies by product. All firewall filter types support ingress and egress filters. An ingress firewall filter is a filter that is applied to packets entering an interface or VLAN. An egress firewall filter is a filter that is applied to packets exiting an interface or VLAN. Note that egress firewall filters do not affect the flow of locally generated control packets from the Routing Engine.

Processing Order of Firewall Filters



This graphic illustrates and describes the processing order of firewall filters.

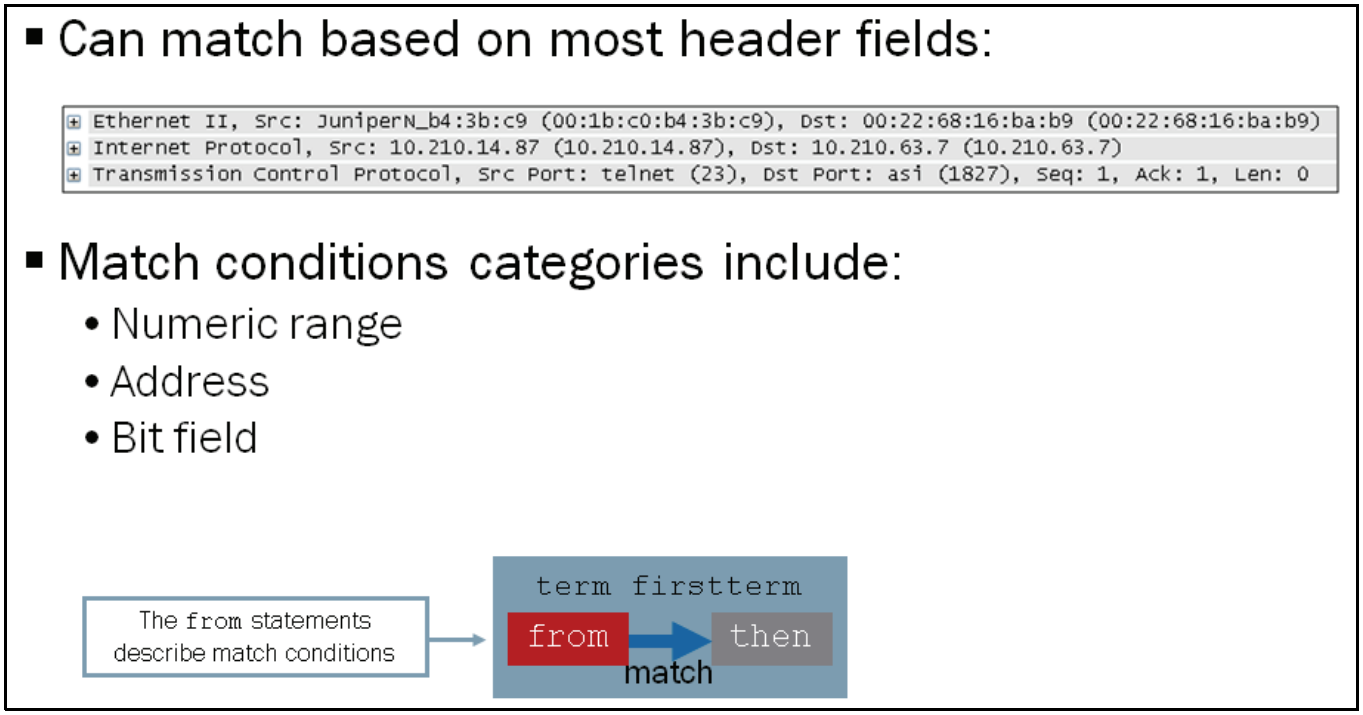
Building Blocks of Firewall Filters



The fundamental building block of a firewall filter is the *term*. A term contains zero or more match conditions and one or more actions. If all the match conditions are true, the Junos OS takes the specified action or actions within the term. If no match conditions are specified, all traffic matches the firewall filter term and is subjected to the stated action. You use a filter to group together multiple terms and establish the order in which the system evaluates the terms. The Junos firewall filters require at least one term.

Firewall filters always include a default term that discards all packets not explicitly permitted through the defined terms. When implementing firewall filters, keep in mind that the order of the terms is important and can impact the results. If you must reorder terms within a given firewall filter, consider using the **insert** command to simplify the task.

Match on Header Fields



The switch processes each packet through the firewall filters independently of all other packets. This processing affects the way you craft firewall filters and also has implications on the information that is available to a switch when it processes packets through these filters.

Because the switch does not keep state information on connections, you must explicitly allow traffic in both directions for each connection that you want to permit. By contrast, stateful firewall filters require you to permit only the initial connection and then permit bidirectional communications for this connection.

The stateless nature of these filters also affects the information available to the switch when processing these filters. For example, if you want to allow all *established* TCP sessions through a switch, you can have the firewall filter permit all TCP packets that have the acknowledgement (ACK) flag set in the TCP header. However, looking for this match condition provides no guarantee that the session was properly established. This packet might instead have been maliciously crafted to have the ACK flag set for an unestablished TCP session.

You specify the criteria to be used to match packets in *from* clauses within firewall filter terms. You can use many header fields as match criteria. However, you must remember that all header fields might not be available to you because of the way firewall filters are processed.

When you specify a header field, the JUNOS software looks for a match at the location in the header where that field should exist. However, it does not check to ensure that the header field makes sense in the given context. For example, if you specify that the software should look for the ACK flag in the TCP header, the software looks for that bit to be set at the appropriate location, but it does not check that the packet was actually a TCP packet. Therefore, you must account for how the software looks for a match when writing your filters. In this case, you should have the switch check both that the packet was a TCP packet and that the TCP ACK flag was set.

The stateless nature of firewall filters can affect the information available in the processing of fragmented packets. Processing fragments is trickier with stateless firewall filters than with a stateful firewall filter. The first fragment should have all the Layer 4 headers; however, subsequent fragments will not. Additionally, attempting to check Layer 4 headers in fragments produces unpredictable results. As was explained previously, the JUNOS software still attempts to evaluate the Layer 4 headers; however, the second and subsequent fragments do not contain these headers, so the matches are unpredictable.

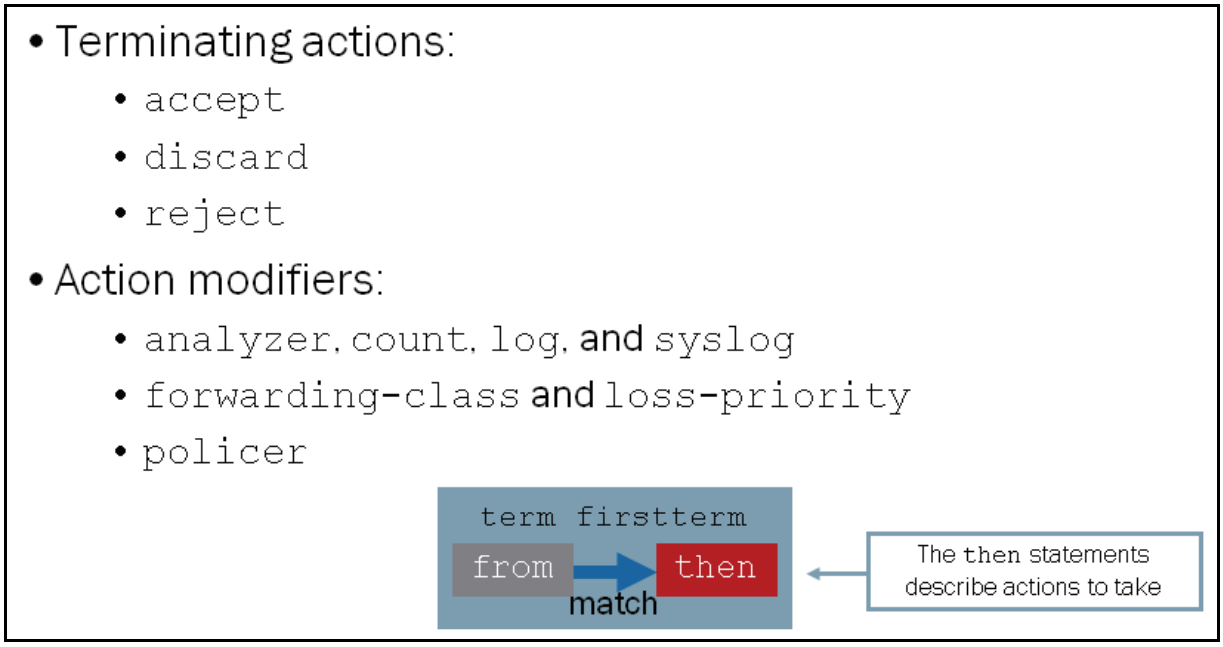
Categories of Match Conditions

Match conditions generally fall into three categories: numeric range, address, and bit field match conditions. You can generally use the same evaluation options for each condition within the category. There are also several *synonyms*, which are match conditions that are equivalent to one or more of these match conditions. The actual match conditions available for each of the

listed categories is dependant on the firewall filter type used. The following output illustrates the supported match conditions for Layer 2 port-based and VLAN-based filters:

```
{master:0}[edit firewall]
user@Switch-1# set family ethernet-switching filter test term test from ?
Possible completions:
+ apply-groups          Groups from which to inherit configuration data
+ apply-groups-except   Don't inherit configuration data from these groups
> destination-address   Match IP destination address
> destination-mac-address  Match MAC destination address
+ destination-port      Match TCP/UDP destination port
> destination-prefix-list  Match IP destination prefixes in named list
+ dot1q-tag             Match Dot1Q Tag Value
+ dot1q-user-priority   Match Dot1Q user priority
+ dscp                  Match Differentiated Services (DiffServ) code point
+ ether-type            Match Ethernet Type
  fragment-flags        Match fragment flags (in symbolic or hex formats) - (Ingress
only)
+ icmp-code             Match ICMP message code
+ icmp-type             Match ICMP message type
> interface             Match interface name
  is-fragment           Match if packet is a fragment
+ precedence            Match IP precedence value
+ protocol              Match IP protocol type
> source-address        Match IP source address
> source-mac-address    Match MAC source address
+ source-port           Match TCP/UDP source port
> source-prefix-list    Match IP source prefixes in named list
  tcp-established       Match packet of an established TCP connection
  tcp-flags             Match TCP flags (in symbolic or hex formats) - (Ingress only)
  tcp-initial           Match initial packet of a TCP connection - (Ingress only)
+ vlan                  Match Vlan Id or Name
```

Common Actions



You specify actions in the `then` clause of a term. You can specify terminating actions or action modifiers. Terminating actions cause the policy evaluation to stop. The `accept` action causes the switch to accept the packet and continue the input or output processing of the packet. The `discard` action causes the switch to silently discard the packet, without sending an Internet Control Message Protocol (ICMP) message to the source address.

You can specify one or more action modifiers with any terminating action. If you specify an action modifier but do not specify a terminating action, the switch imposes an action of `accept`. You can use the `count` action modifier to count the number of packets processed through the filter that match the specified criteria defined in the `from` statement. The `forwarding-class` and `loss-priority` action modifiers are used to specify class-of-service (CoS) information. The `policer` action modifier allows you to invoke a traffic policer. The `analyzer` action modifier specifies that the switch should mirror the packets for additional analysis. Note that the supported actions vary based on the firewall filter type and direction in which the filter is applied. For support information, refer to the technical publications at http://www.juniper.net/techpubs/.

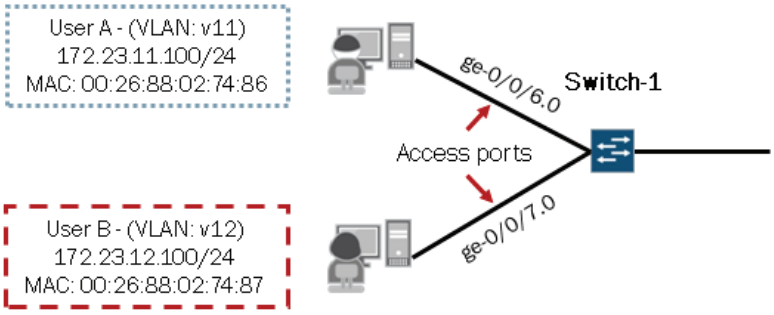
Default Action

The default action when a firewall filter is configured is *discard*. Therefore, if a packet fails to match any term within a firewall filter or chain of firewall filters, the switch silently discards it.

Unlike routing policy, the default action is different when a firewall filter is configured than when no firewall filter is configured. If no firewall filter is applied, the default action is *accept*.

Case Study: Topology and Objectives

- Implement filters on the access ports so that only frames using the expected source MAC addresses are permitted
  - Discard and count frames sourced from any other MAC addresses
- Implement a filter on both VLANs to block frames destined to MAC address 01:80:c2:00:00:00
  - Discard and count frames destined to the referenced MAC address

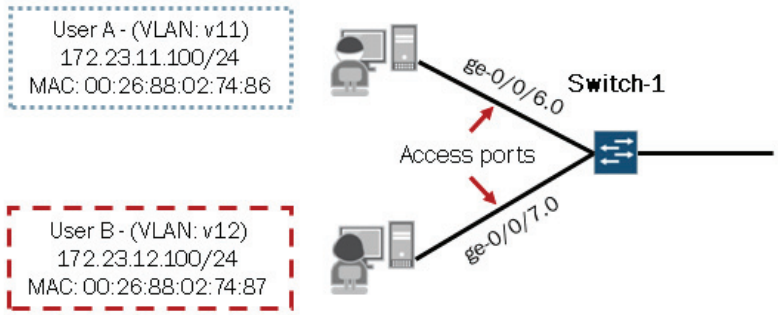


This graphic introduces a case study related to firewall filters and provides the related topology and objectives.

Configuring the Filters: Part 1

```
{master:0}[edit firewall family ethernet-switching]
user@Switch-1# show filter limit-MAC-ge006
term 1 {
    from {
        source-mac-address {
            00:26:88:02:74:86;
        }
    }
    then accept;
}
term 2 {
    then {
        discard;
        count ge006-invalid-MAC;
    }
}
```

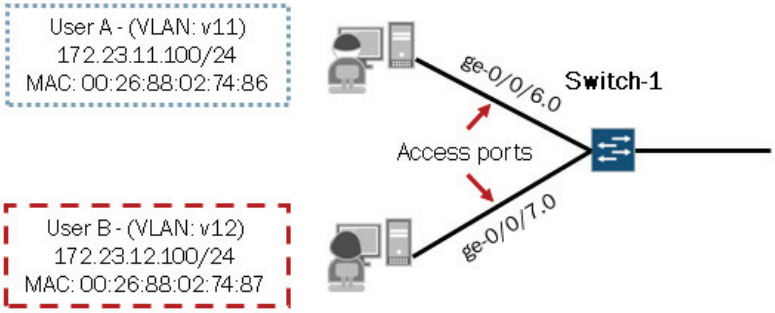
```
{master:0}[edit firewall family ethernet-switching]
user@Switch-1# show filter limit-MAC-ge007
term 1 {
    from {
        source-mac-address {
            00:26:88:02:74:87;
        }
    }
    then accept;
}
term 2 {
    then {
        discard;
        count ge007-invalid-MAC;
    }
}
```



This and the next graphic illustrate sample configurations used to meet the objectives listed on the previous graphic.

Configuring the Filters: Part 2

```
{master:0}[edit firewall family ethernet-switching]
user@Switch-1# show filter block-dest-MAC-01:80:c2:00:00:00
term 1 {
    from {
        destination-mac-address {
            01:80:c2:00:00:00;
        }
    }
    then {
        discard;
        count block-stp-bpdus;
    }
}
term 2 {
    then accept;
}
```

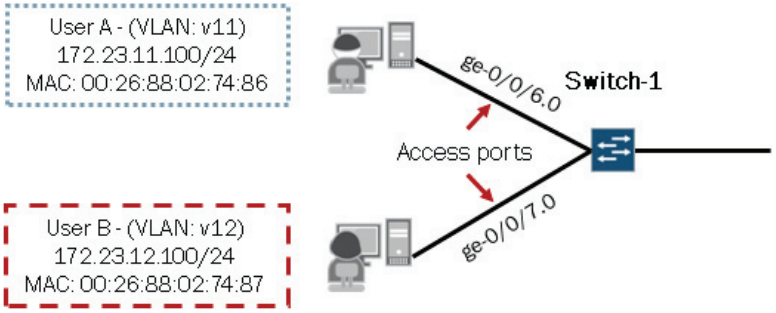


This graphic illustrates the remaining firewall filter configuration used to meet the stated objectives for this case study.

Applying the Filters: Part 1

```
{master:0}[edit interfaces]
user@Switch-1# show ge-0/0/6
unit 0 {
    family ethernet-switching {
        vlan {
            members v11;
        }
        filter {
            input limit-MAC-ge006;
        }
    }
}
```

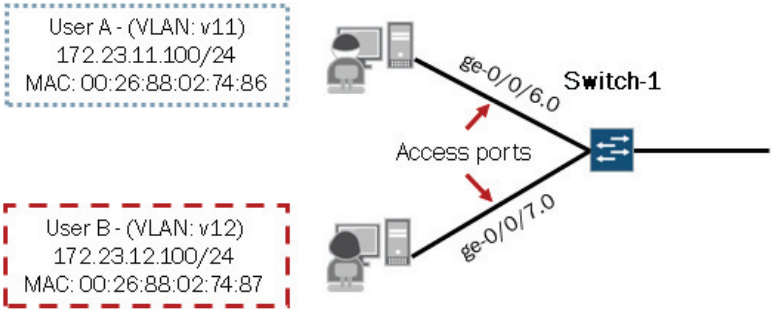
```
{master:0}[edit interfaces]
user@Switch-1# show ge-0/0/7
unit 0 {
    family ethernet-switching {
        vlan {
            members v12;
        }
        filter {
            input limit-MAC-ge007;
        }
    }
}
```



This graphic illustrates the application of port-based firewall filters.

Applying the Filters: Part 2

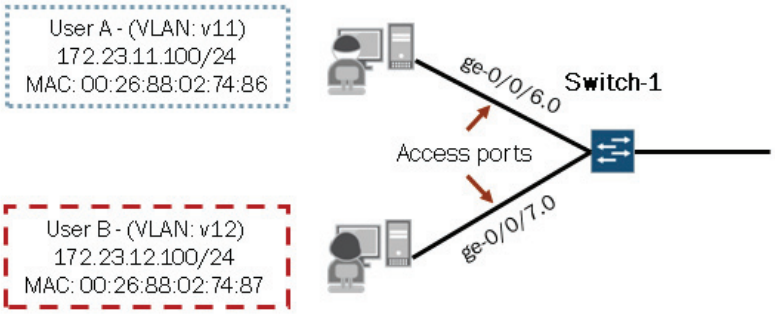
```
{master:0}[edit vlans]
user@Switch-1# show
v11 {
    vlan-id 11;
    filter {
        input block-dest-MAC-01:80:c2:00:00:00;
    }
    l3-interface vlan.11;
}
v12 {
    vlan-id 12;
    filter {
        input block-dest-MAC-01:80:c2:00:00:00;
    }
    l3-interface vlan.12;
}
```



This graphic illustrates the application of the VLAN-based firewall filter.

Monitoring Firewall Filters

```
{master:0}
user@Switch-1> show firewall

Filter: block-dest-MAC-01:80:c2:00:00:00
Counters:
Name                                                Bytes              Packets
block-stp-bpdus                                       472                    7

Filter: limit-MAC-ge006
Counters:
Name                                                Bytes              Packets
ge006-invalid-MAC                                    1148                   12

Filter: limit-MAC-ge007
Counters:
Name                                                Bytes              Packets
ge007-invalid-MAC                                     842                    9
```



This graphic illustrates a sample output from the **`show firewall`** command. In this example, you can see the individual counters incrementing which indicates devices, using unexpected MAC addresses, are attempting to gain access to the network using the highlighted access ports and that unauthorized STP BPDUs are being received on one or both of the defined VLANs. Note that you could further identify from where the unauthorized STP BPDUs are sourced by defining unique VLAN-based filters with unique counters or even better by combining that filtering functionality in to the port-based filters. By including the filtering functionality in to the port-based filters you can learn the exact interfaces through which the unauthorized BPDUs are entering the switch.

Review Questions

1. What is a traffic storm and how is it created?
2. What actions can be taken when a storm control level is exceeded?
3. Which types of firewall filters are supported on EX Series switches? Where are they applied?

Answers

1.

A traffic storm is an undesirable state in a network that impacts a network's performance. Traffic storms are generated when certain types of traffic (broadcast, multicast, and unknown unicast) is continuously flooded throughout a network. Some traffic prompts a receiving node to respond by broadcasting its own messages on the network. This, in turn, prompts further responses, creating a snowball effect. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service.

2.

By default, when the storm control level is exceeded the switch drops all offending traffic. You can alter this default behavior so that interfaces through which a storm control level violation occurs are shut down.

3.

EX Series switches support three types of firewall filters, port-based, VLAN-based, and router-based firewall filters. Port-based and VLAN-based filters are Layer 2 filters and are applied to individual switch ports and VLANs respectively. Router-based filters are Layer 3 filters and are applied to Layer 3 interfaces, including RVIs.

JNCIS-ENT Switching Study Guide

Chapter 6: Virtual Chassis

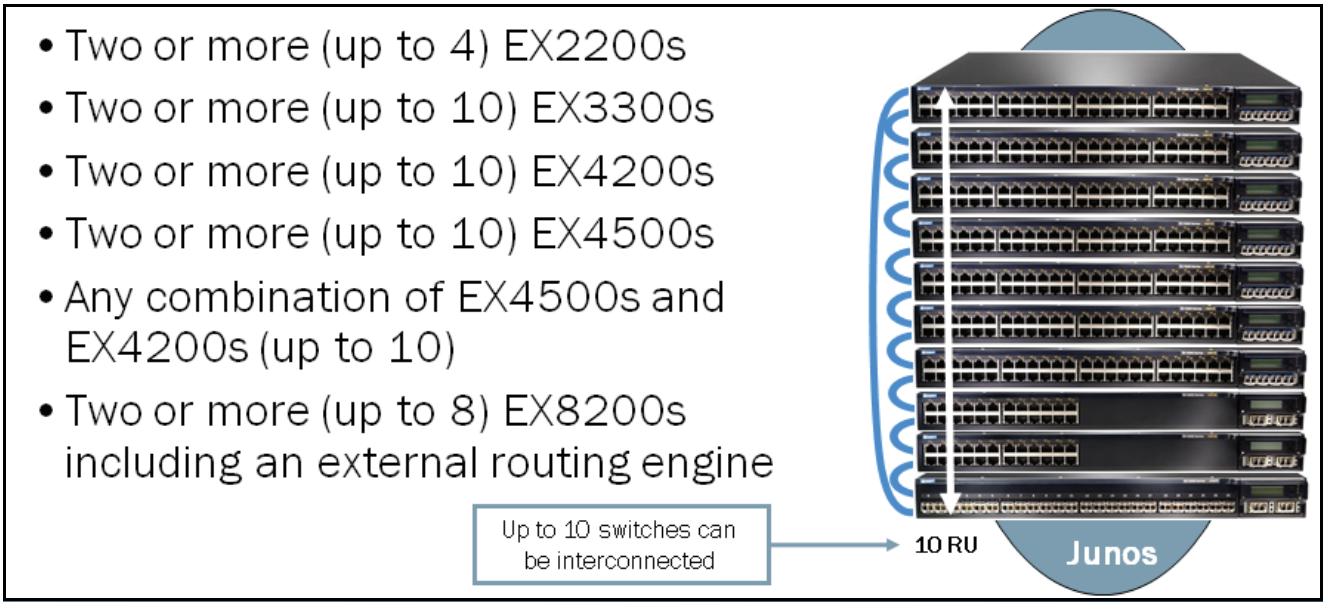
This Chapter Discusses:

- The key concepts and components of a Virtual Chassis;
- The operational details of a Virtual Chassis; and
- The implementation and operational verification of a Virtual Chassis.

Virtual Chassis Defined

You can connect two or more EX Series switches together to form one unit and manage the unit as a single chassis, called a Virtual Chassis. The Virtual Chassis system offers you *add-as-you-grow* flexibility. A Virtual Chassis can start with two switches and grow, based on your needs, to as many as ten interconnected switches. This ability to grow and expand within and across wiring closets is a key advantage in many enterprise environments.We discuss additional benefits and design and operational considerations on subsequent graphics in this chapter.

Valid Chassis Combinations



A few different switch models can be used to create a Virtual Chassis. You can combine two or more EX2200 (up to 4) switches together to create a Virtual Chassis. You can also combine two or more EX3300 (up to 10) switches to create a Virtual Chassis. You can combine two or more EX4200 (up to 10) switches together within a single Virtual Chassis. In many environments, this allows administrators to collapse the access and aggregation layers into a single layer making the network more efficient and less complicated to manage.

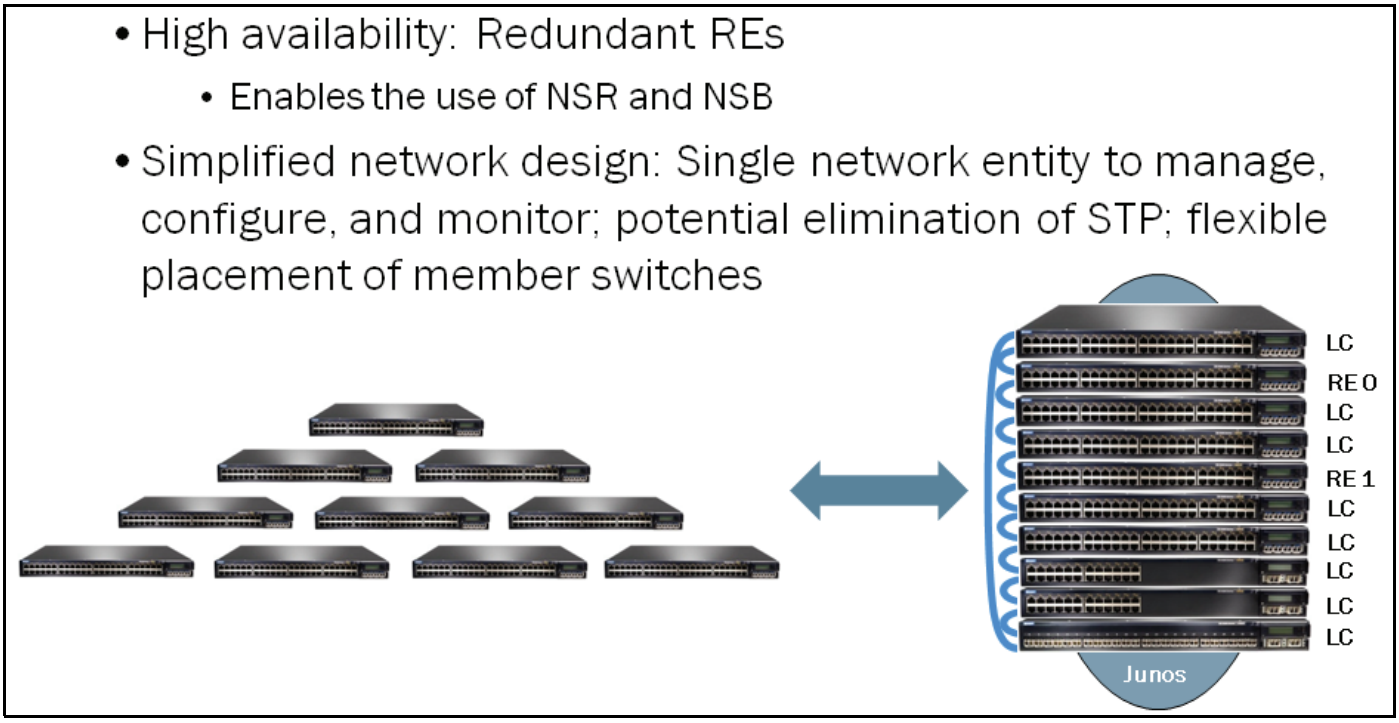
In addition to the previous models, the EX4500 Series switches support the Virtual Chassis technology. With the EX4500s you have the ability to combine two or more EX4500s (up to 10) into a Virtual Chassis. You can also combine EX4500s and EX4200s (up to 10 switches) within the same Virtual Chassis.

The EX8200 Series switches also support Virtual Chassis functionality, but you must include an external Routing Engine (XRE) to handle most of the Routing Engine functions. The XRE is responsible for the Virtual Chassis configuration as well as verifying all functionality for devices within the Virtual Chassis. An EX8200 Virtual Chassis can combine up to eight EX8200 Series switches. You can mix and match the specific EX8200 platforms to suit your needs. You can also add a second XRE for Routing Engine redundancy. If you choose to use two XREs, the master and backup roles are strictly decided based on the XREs uptime. The XRE with the highest uptime is chosen to act as the master Routing Engine for the Virtual Chassis. In many environments, this allows administrators to collapse the aggregation and core layers into a single layer making the network more efficient and less complicated to manage.

Currently only the EX2200, EX3300, EX4200, EX4500, and EX8200 Series switches support the Virtual Chassis technology. The EX4200 was the first chassis to support the Virtual Chassis functionality and will be the primary focus throughout the remainder of the chapter. Many of the topics discussed relate to the EX2200, EX3300, EX4500 as well as the EX4200 mixed platform Virtual Chassis scenarios.

For detailed information regarding a specific EX Series platform, refer to the technical publications or the product-specific datasheets and literature found at: http//www.juniper.net/techpubs/ as well as http://www.juniper.net/us/en/products-services/switching/ex-series/, respectively.

Control Plane Redundancy



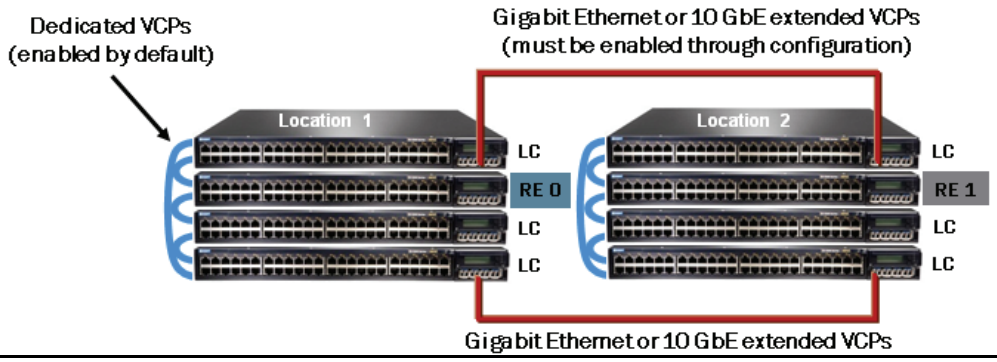
In a Virtual Chassis configuration, one of the member switches is elected as the master RE and a second member switch is elected as the backup RE. This design approach provides control plane redundancy and is a requirement in many enterprise environments.

Having redundant REs enables you to implement nonstop active routing (NSR), and nonstop bridging (NSB) which allows for a transparent switchover between REs without requiring restart of supported routing protocols and supported Layer 2 protocols respectively. Both REs are fully active in processing protocol sessions, so each can take over for the other. We discuss both NSR and NSB in more detail in a subsequent chapter.

You can connect certain EX Series switches together to form a Virtual Chassis system, which you then manage as a single device. Comparatively speaking, managing a Virtual Chassis system is much simpler than managing up to ten individual switches. For example, when upgrading the software on a Virtual Chassis system, only the master switch must have the software upgraded. However, if all members function as standalone switches, all individual members must have the software upgraded separately. Also, in a Virtual Chassis scenario, it is not necessary to run the Spanning Tree Protocol (STP) between the individual members because in all functional aspects, a Virtual Chassis system is a single device.

Virtual Chassis Components

- EX Series switches
  - Participating switches serve as REs or line cards (LCs)
- Virtual Chassis ports (can be dedicated VCPs or uplink ports converted to extended VCPs through configuration)
  - VCPs interconnect PFEs from one switch to another to form a single backplane



You can interconnect one to ten EX4200 switches to form a Virtual Chassis. A Virtual Chassis can consist of any combination of model numbers within the EX4200 family of switches.

Each EX4200 switch has two or three Packet Forwarding Engines (PFEs) depending on the platform. All PFEs are interconnected, either through internal connections or through the Virtual Chassis ports (VCPs). Collectively, the PFEs and their connections constitute the Virtual Chassis backplane.

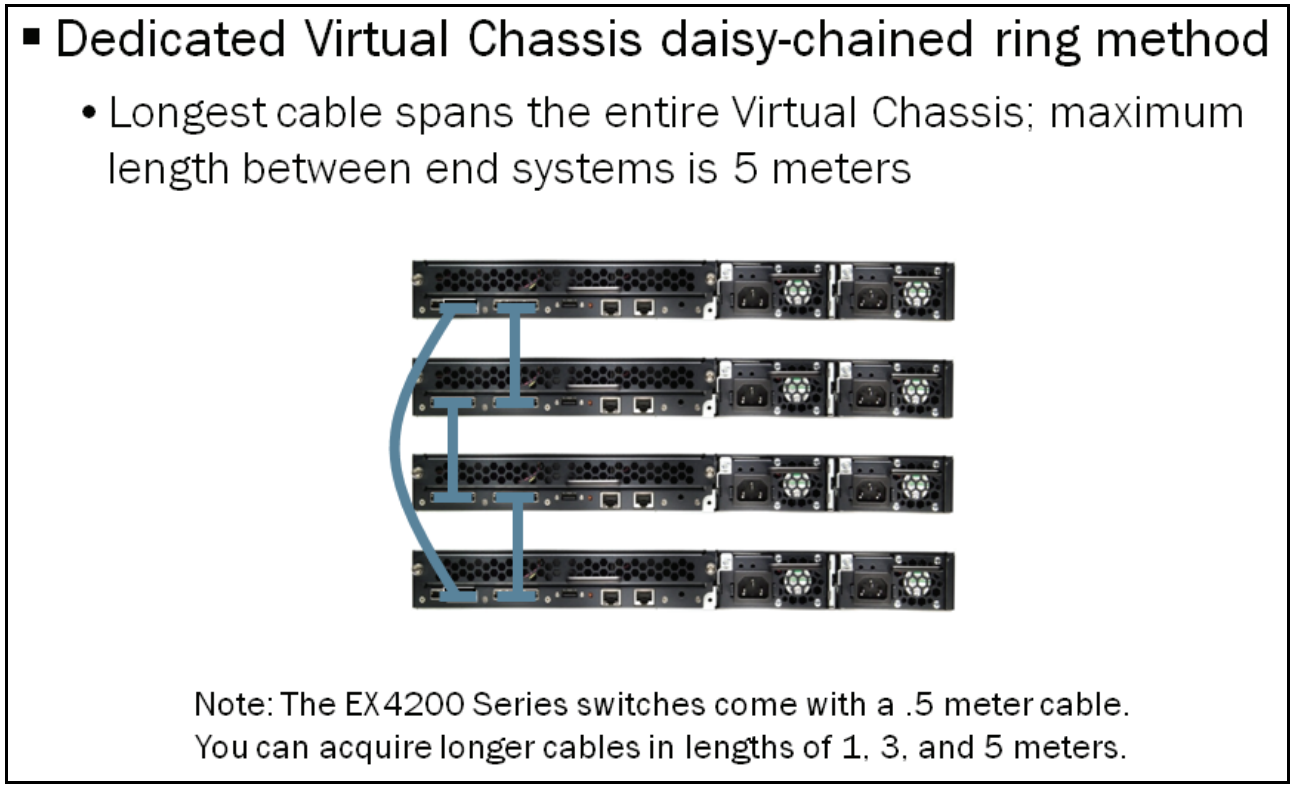
You can use the built-in VCPs on the rear of the EX4200 switches or uplink ports, converted to VCPs, to interconnect the member switches' PFEs. To use an uplink port as a VCP, explicit configuration is required. The following example shows the operational mode command used to convert the
xe-0/1/0 uplink port to a VCP.

```
{master:0}
user@Switch-1> request virtual-chassis vc-port set pic-slot 1 port 0
```

Note that once the illustrated command is issued the xe-0/1/0 interface is converted to the vcp-255/1/0 interface:

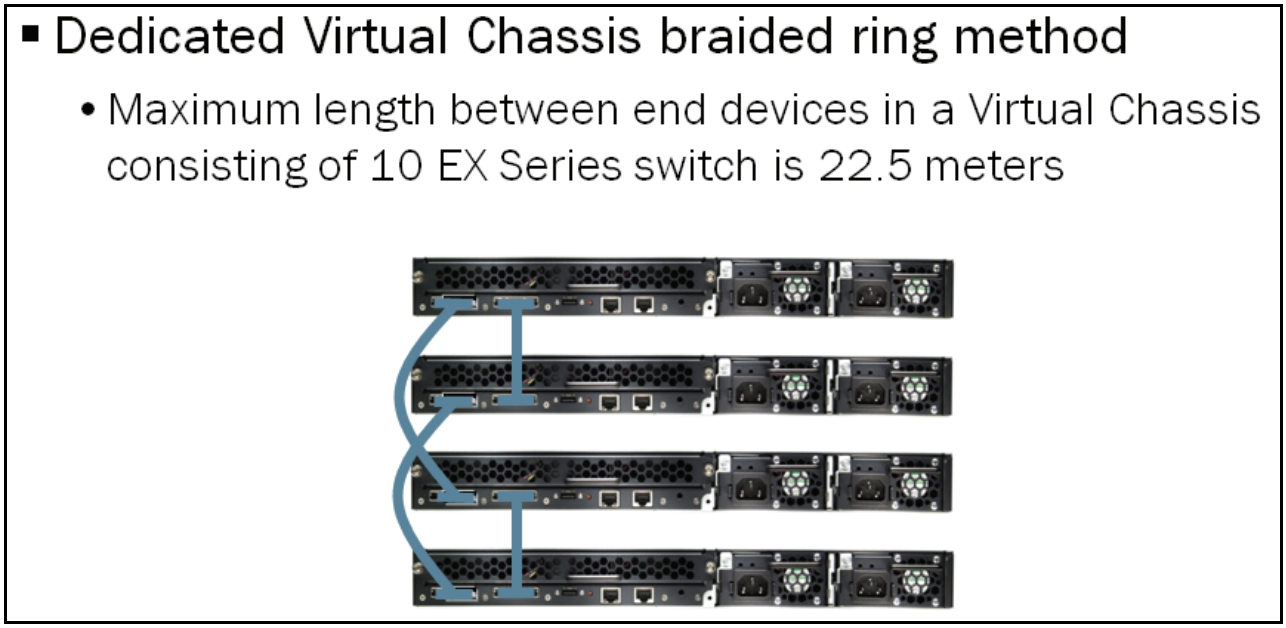
```
{master:0}
user@Switch-1> show interfaces terse | match vcp-255
vcp-255/1/0             up    down
```

Virtual Chassis Cabling Options: Part 1



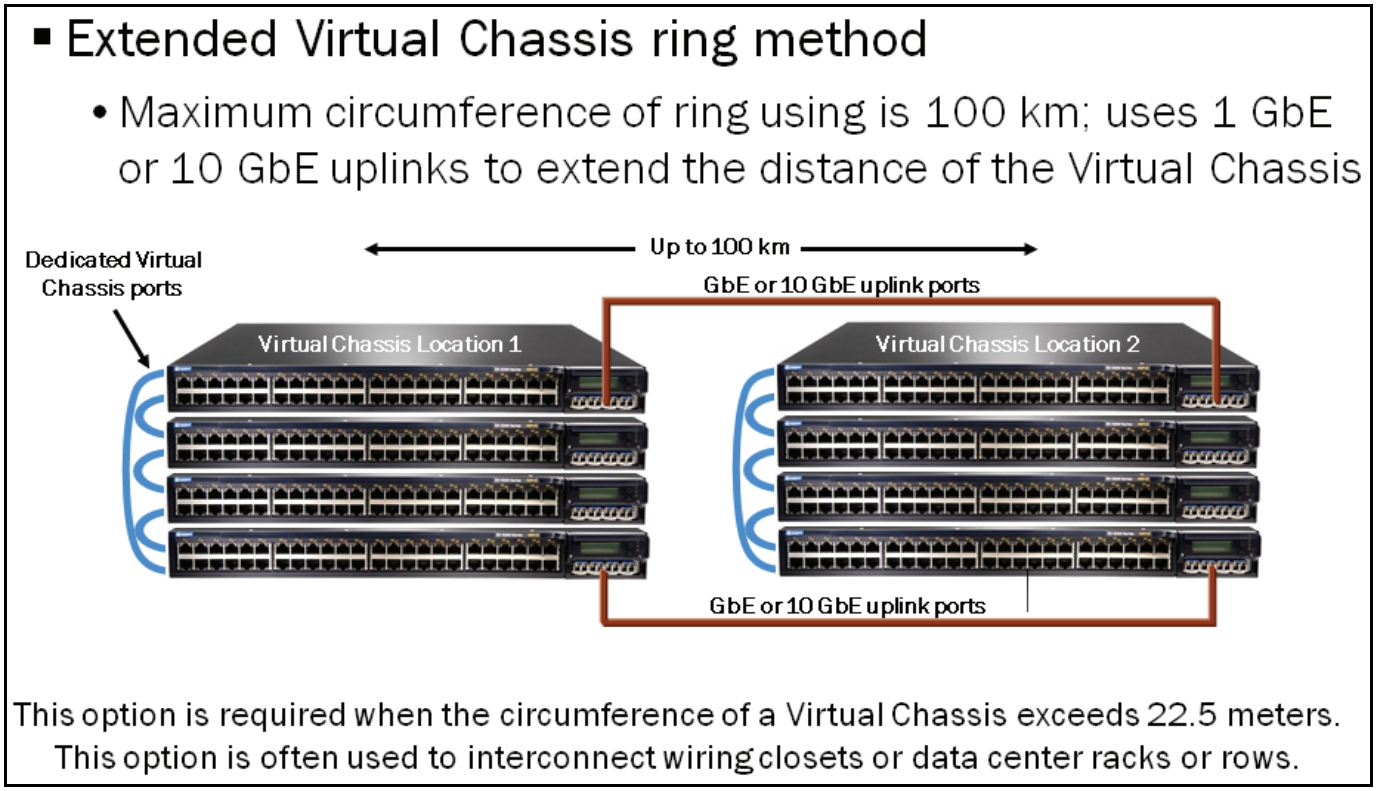
This graphic illustrates one of the recommended cabling options and provides some related information.

Virtual Chassis Cabling Options: Part 2



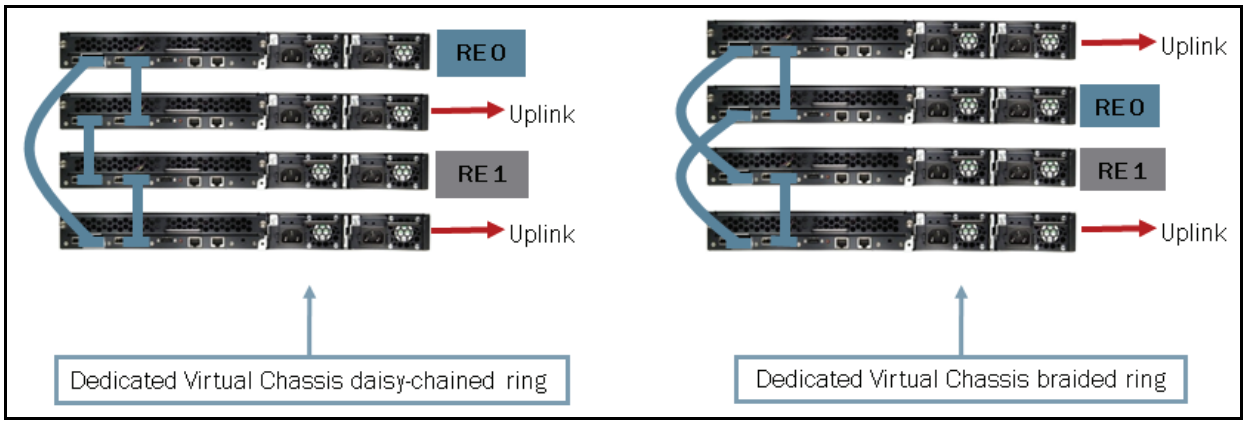
This graphic illustrates another recommended cabling option and provides some related information.

Virtual Chassis Cabling Options: Part 3



This graphic illustrates another recommended cabling option and provides some related information.

Recommended RE Placement

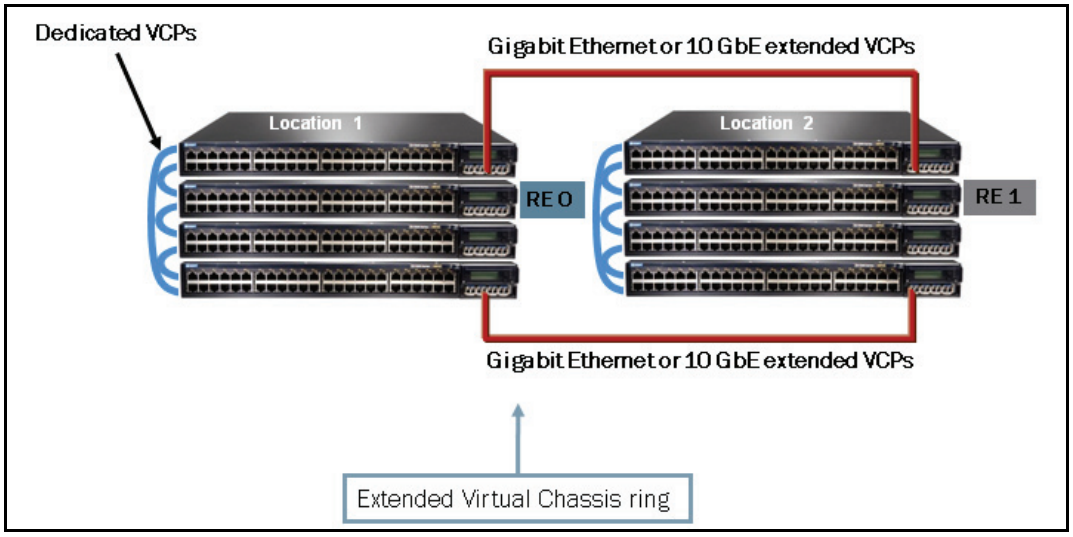


If a disruption to the Virtual Chassis configuration occurs due to member switches failing or being removed from the configuration, the Virtual Chassis configuration can split into two separate Virtual Chassis systems (known as a split Virtual Chassis). This situation could cause disruptions in the network if the two separate configurations share common resources, such as global IP addresses.

EX4200 switches support the split and merge feature which provides a method to prevent the separate Virtual Chassis configurations from adversely affecting the network and also allows the two parts to merge back into a single Virtual Chassis configuration. The split and merge feature is enabled by default on EX4200 switches. If desired, you can disable the split and merge feature through configuration. We cover the configuration option used to disable this feature on a subsequent graphic.

This and the next graphic illustrate the recommended placement of the Routing Engines (REs)—the master and backup switches for the Virtual Chassis. Specifically, this graphic illustrates the recommended positioning for the daisy-chained ring and braided ring cabling methods. You should follow these RE placement recommendations to help avoid potential issues related to a split Virtual Chassis and other failure conditions.

Recommended RE Placement



This graphic illustrates the recommended placement of the REs when implementing a extended Virtual Chassis.

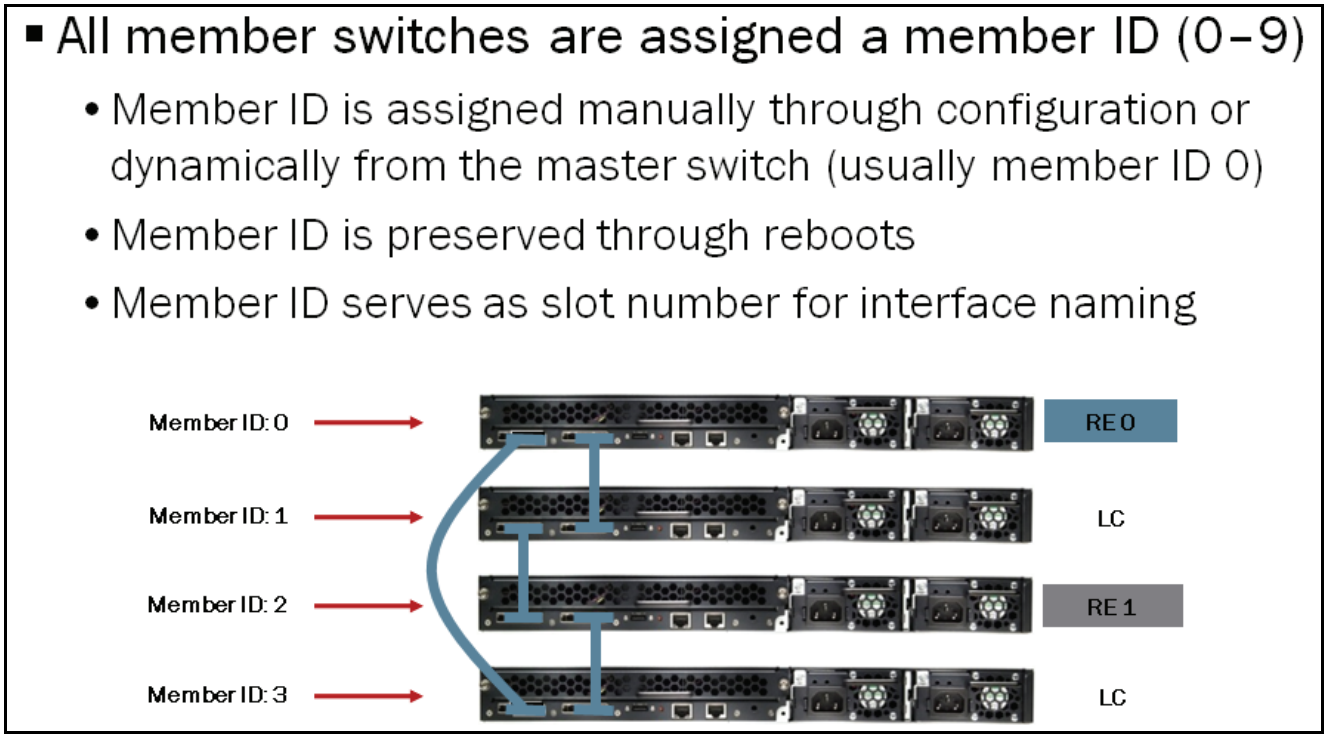
Determining Mastership

1. Member with the highest user-configured priority
   - Priority range is 1–255, factory-default value is 128
2. Member previously functioning as master prior to reboot
3. Member with the longest standing uptime
   - Difference must be greater than 1 minute
4. Member with the lowest MAC address
   - Used as tie breaker if all is equal through the first three determination steps
5. Second member in election decision tree becomes backup switch; all other members are line cards

Note: If a master or backup fails, one of the line card switches is elected the new backup switch using the same criteria.

The graphic outlines the steps used to determine mastership within a Virtual Chassis along with some related considerations.

Member ID Assignment and Considerations



The master switch typically assumes a member ID of 0 because it is the first switch powered on. Member IDs can be assigned manually using the preprovisioned configuration method or dynamically from the master switch.

If assigned dynamically, the master switch assigns each member added to the Virtual Chassis a member ID from 1 through 9, making the complete member ID range 0–9. The master assigns each switch a member ID based the sequence that the switch was added to the Virtual Chassis system. The member ID associated with each member switch is preserved, for the sake of consistency, across reboots. This preservation is helpful because the member ID is also a key reference point when naming individual interfaces. The member ID serves the same purpose as a slot number when configuring interfaces.

Note that when the member ID is assigned by the master switch, you can change the assigned ID values using the CLI. The LCD and CLI prompt displays the member ID and role assigned to that switch. The following sequence shows an example of the CLI prompt and how to change the member ID:

```
{master:0}
user@Switch-1> request virtual-chassis renumber member-id 0 new-member-id 5

To move configuration specific to member ID 0 to member ID 5, please
use the replace command. e.g. replace pattern ge-0/ with ge-5/

Do you want to continue ? [yes,no] (no) yes


{master:0}
user@Switch-1>
Switch-1 (ttyu0)

login: user
Password:

--- JUNOS 10.1R2.8 built 2010-05-11 04:08:08 UTC
{master:5}
user@Switch-1>
```

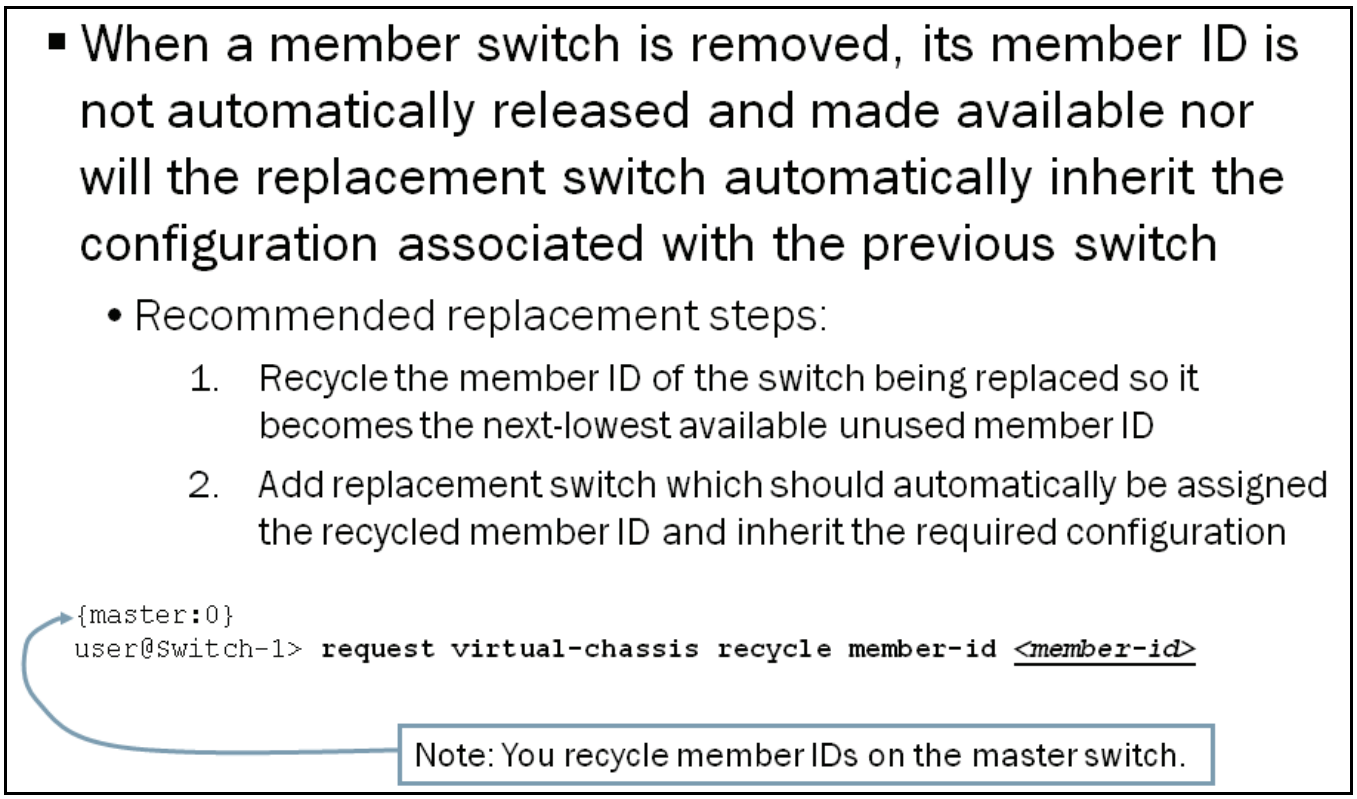
If you must shutdown a specific member of a Virtual Chassis, you can use the **request system halt member** command as shown in the following example:

```
{master:5}
user@Switch-1> request system halt member ?
Possible completions:
  <member>             Halt specific virtual chassis member (0..9)
```

If needed, you can access individual members of a Virtual Chassis system using the **request session** operational command as shown in the following example:

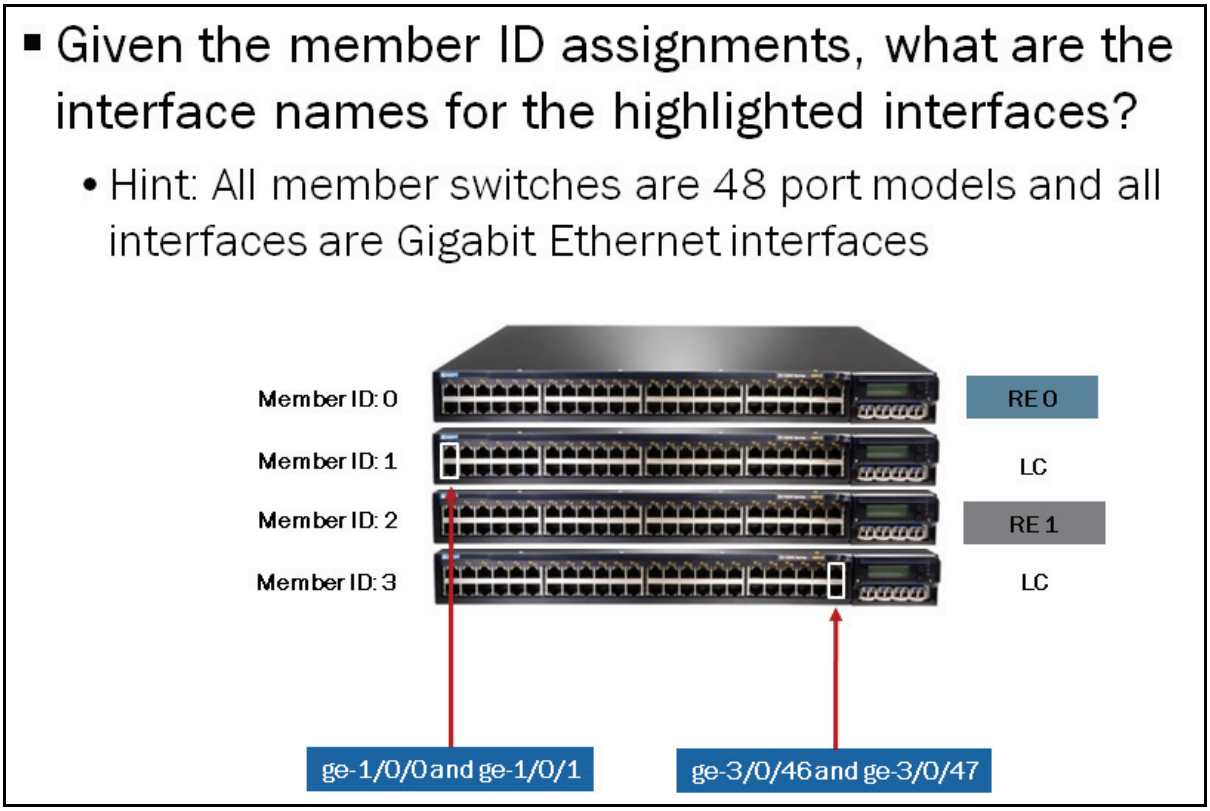
```
{master:5}
user@Switch-1> request session member 1

--- JUNOS 10.1R2.8 built 2010-05-11 04:08:08 UTC
{backup:1}
user@Switch-1>
```

Replacing a Member Switch



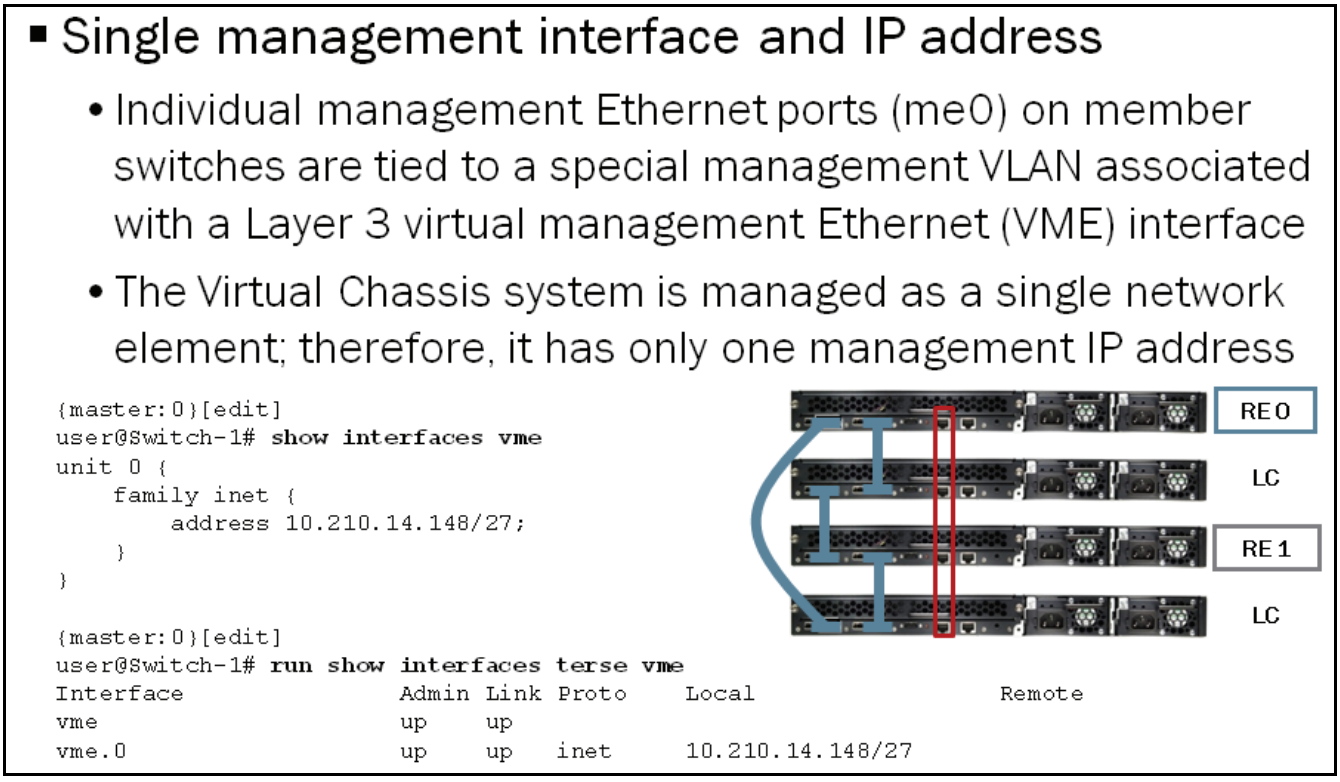
This graphic provides the recommended steps and some related details for replacing a member switch within a Virtual Chassis when using the dynamic installation method. If you are using the preprovisioned installation and configuration method, you will simply modify the configuration on the master switch to account for the new switch's serial number before adding the replacement switch. We cover the dynamic and preprovisioned installation methods later in this chapter.

Think About It!



This graphic provides an opportunity to discuss and think about how interfaces are named within a Virtual Chassis.

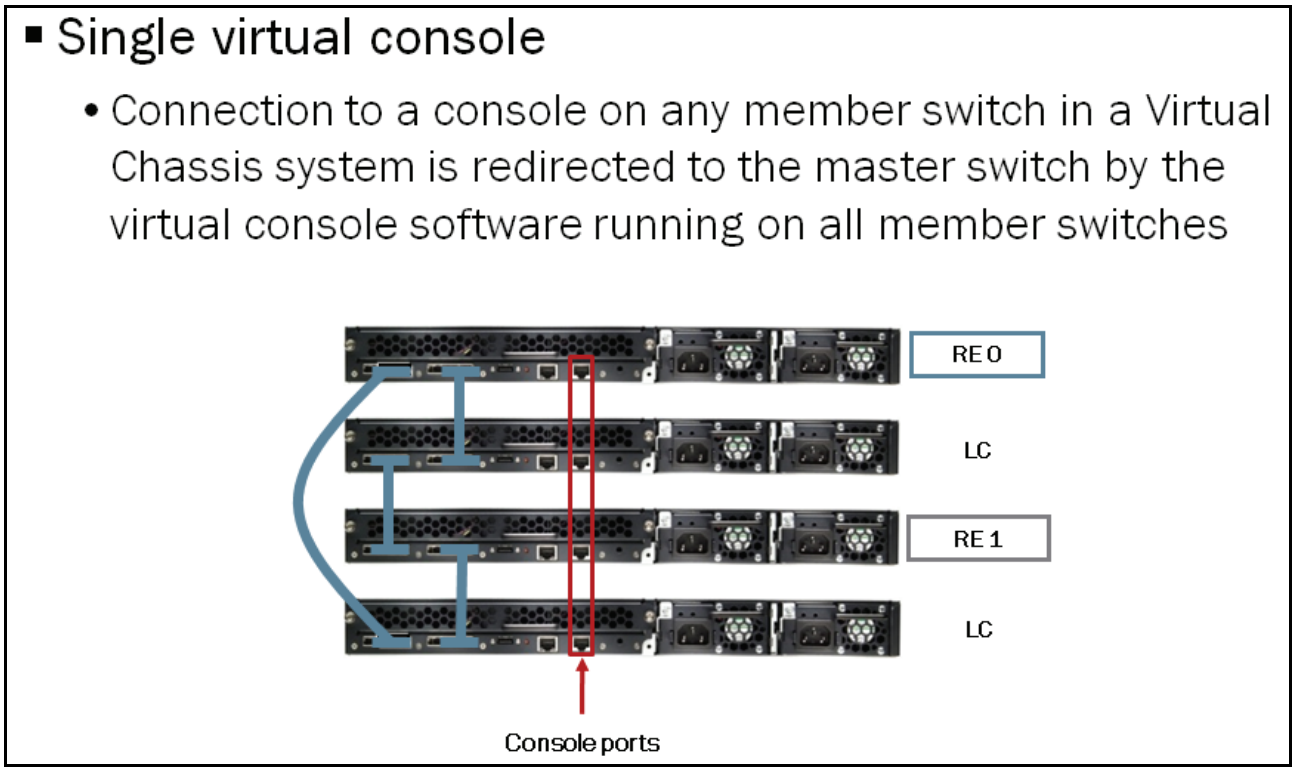
Management Connectivity



The management Ethernet ports on the individual member switches are automatically associated with a management VLAN. This management VLAN uses a Layer 3 virtual management interface called vme which facilitates communication through the

Virtual Chassis system to the master switch even if the master switch's physical Ethernet port designated for management traffic is inaccessible.

When you set up the master switch, you specify an IP address for the vme interface. This single IP address allows you to configure and monitor the Virtual Chassis system remotely through Telnet or SSH regardless of which physical interface the communications session uses.



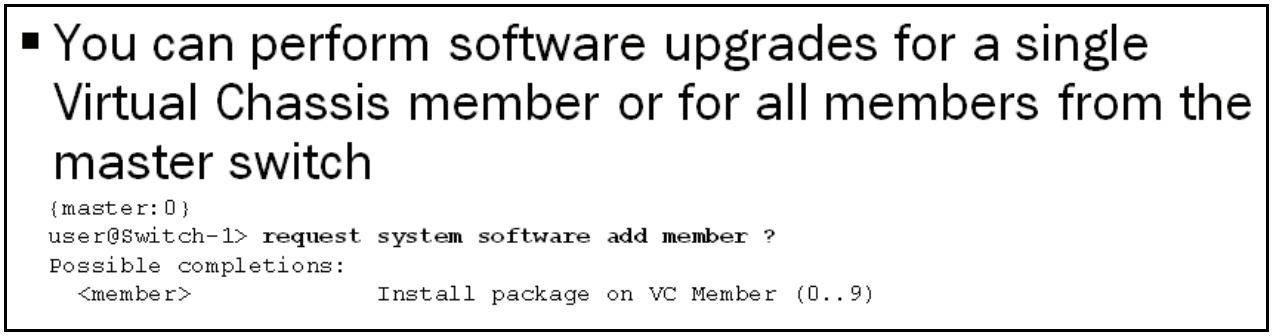
All member switches participating in a Virtual Chassis system run virtual console software. This software redirects all console connections to the master switch regardless of the physical console port through which the communications session is initiated.

The ability to redirect management connections to the master switch simplifies Virtual Chassis management tasks and creates a level of redundancy. Generally speaking, you can obtain all status-related information for the individual switches participating in a Virtual Chassis system through the master switch. It is, however, possible to establish individual virtual terminal (vty) connections from the master switch to individual member switches.

If needed, you can access individual members of a Virtual Chassis system using the **request session** operational command as shown in the following example:

```
{master:5}
user@Switch-1> request session member 1

--- JUNOS 10.1R2.8 built 2010-05-11 04:08:08 UTC
{backup:1}
user@Switch-1>
```


Software Upgrades



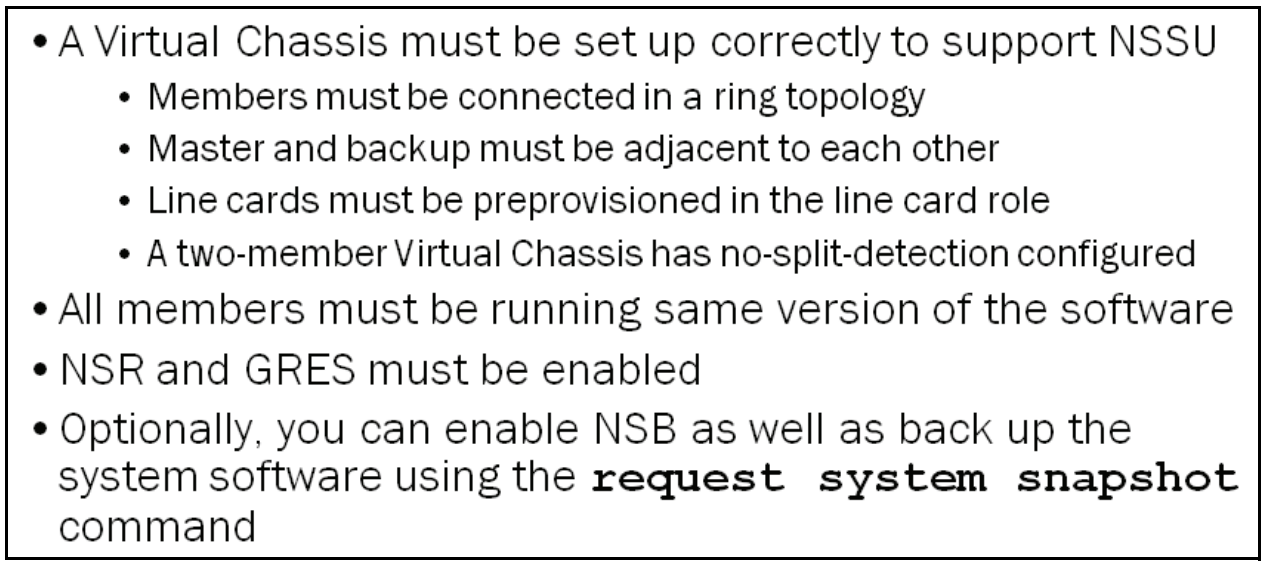
You perform software upgrades within a Virtual Chassis system on the master switch. Using the **request system software add** command, all member switches are automatically upgraded. Alternatively, you can add the **member** option with the desired member ID, as shown on the graphic, to upgrade a single member switch.

Software Compatibility Check

For a new member switch to be added to and participate in a Virtual Chassis system, that switch must be running the same software version as the master switch. The master switch checks the Junos OS version on all newly added switches before allowing them to participate with the Virtual Chassis system. If a software version mismatch exists, the Virtual Chassis master will assign a member ID to the new switch, generate a syslog message and place the newly added switch in the inactive state. Any member switch in this state must be upgraded before actively joining and participating with the Virtual Chassis.

You can upgrade individual switches manually or you can enable the automatic software upgrade feature. The automatic software update feature automatically updates software on prospective member switches as they are added to a Virtual Chassis. This method allows new member switches to immediately join and participate with the Virtual Chassis. We discuss this configuration option in a subsequent section.

Software Upgrades Using NSSU

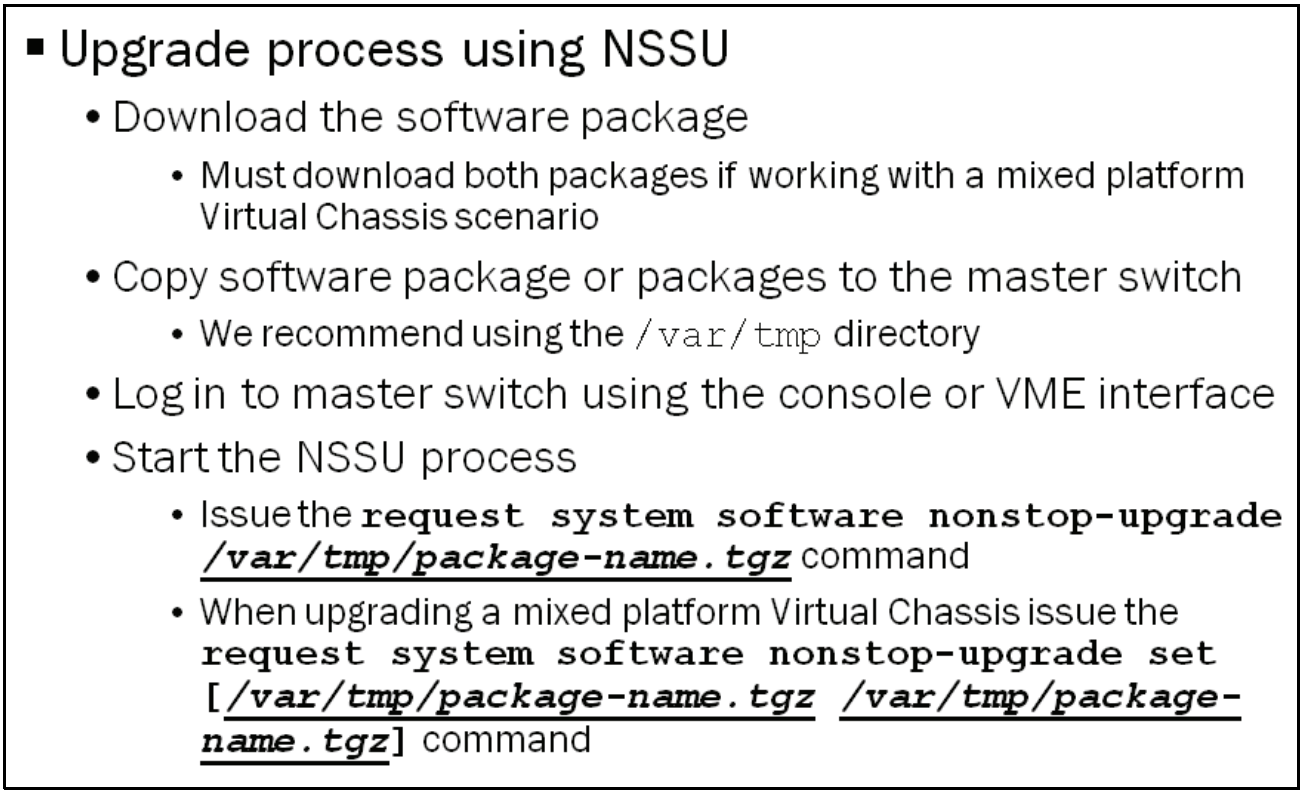


Nonstop software upgrade (NSSU) can be used to upgrade the software running on all member switches participating in a Virtual Chassis while minimizing traffic disruption during the upgrade. NSSU is supported on most EX Series switches that support Virtual Chassis. Refer to specific documentation for your particular platform to verify that the current Junos OS version supports NSSU.

Before attempting an NSSU, you must ensure that the Virtual Chassis is set up correctly. First, the Virtual Chassis members must be connected in a ring topology. The ring topology prevents the Virtual Chassis from splitting during the NSSU process. Next, the master and backup must be adjacent to each other. Adjacency allows the master and backup switches to always be in sync. Last, ensure that the line cards in the Virtual Chassis are explicitly preprovisioned. During an NSSU, the members must maintain their roles—the master and backup must maintain their master and backup roles, although mastership will change,

and the other member switches must maintain their line card roles. When you are upgrading a two-member Virtual Chassis, `no-split-detection` must be configured so that the Virtual Chassis does not split when an NSSU upgrades a member.

All members of the Virtual Chassis must be running the same version of the Junos software. NSR and graceful Routing Engine switchover (GRES) must be enabled. Optionally, you can enable NSB. Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU. Another step that you might want to consider is to back up the current system software—Junos OS, the active configuration, and log files—on each member to an external storage device with the **`request system snapshot`** command.



The graphic describes how to upgrade the software running on all Virtual Chassis members using NSSU. When the upgrade completes, all members are running the new version of the software. Because a GRES occurs during the upgrade, the original Virtual Chassis backup is the new master.

The first step is to download the appropriate software package from Juniper Networks. If you are upgrading the software running on a mixed Virtual Chassis, download the software packages for both switch types. The next step is to copy the software package or packages to the Master switch of the Virtual Chassis. We recommend that you copy the file to the `/var/tmp` directory on the master. Next, log in to the Virtual Chassis using the console connection or the virtual management Ethernet (VME) interface. Using a console connection allows you to monitor the progress of the master switch reboot.

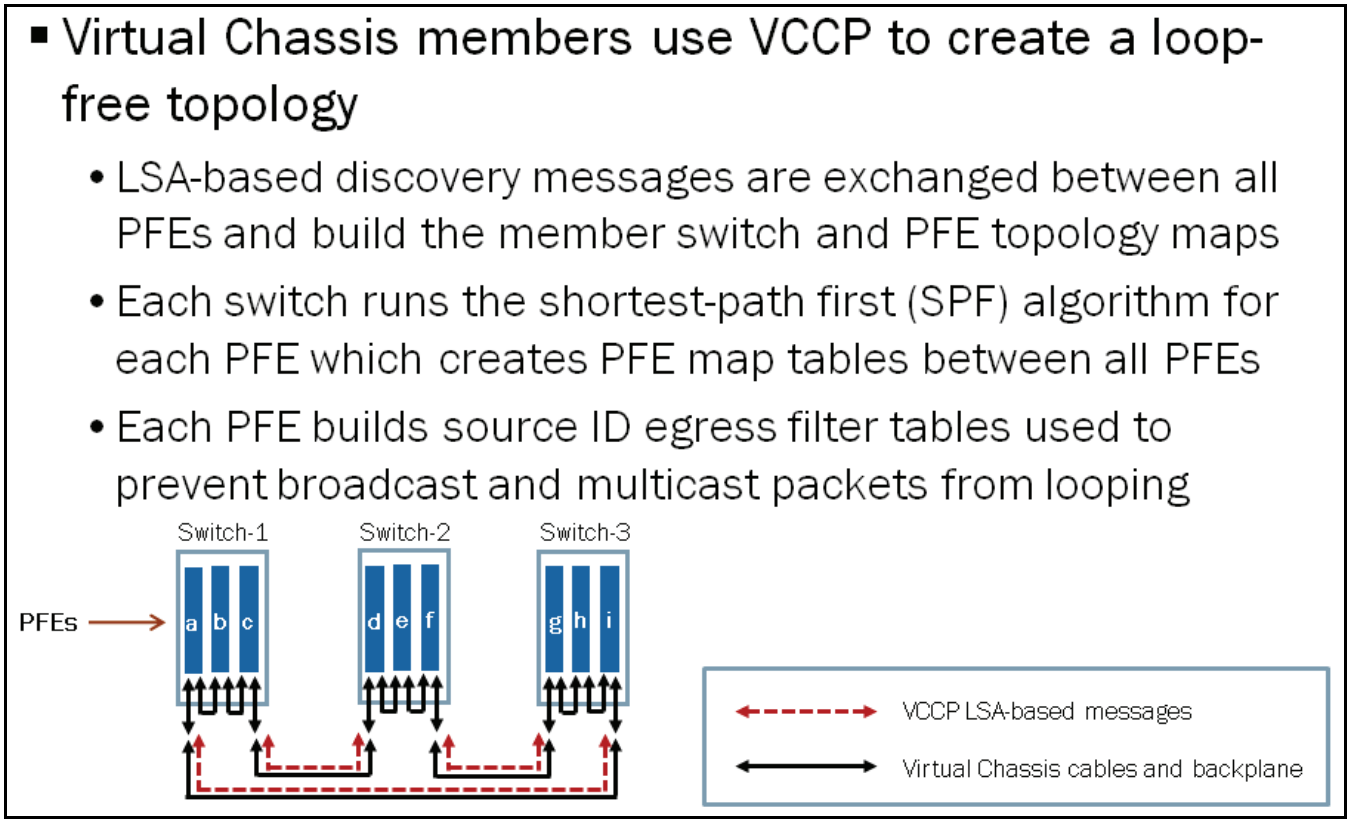
The following command should be used to upgrade a Virtual Chassis using NSSU:

```
{master:0}
user@Switch> request system software nonstop-upgrade /var/tmp/package-name.tgz
```

The following command should be used to upgrade a Virtual Chassis with mixed platforms using NSSU:

```
{master:0}
user@Switch> request system software nonstop-upgrade set [/var/tmp/package1-name.tgz /
var/tmp/package2-name.tgz]
```

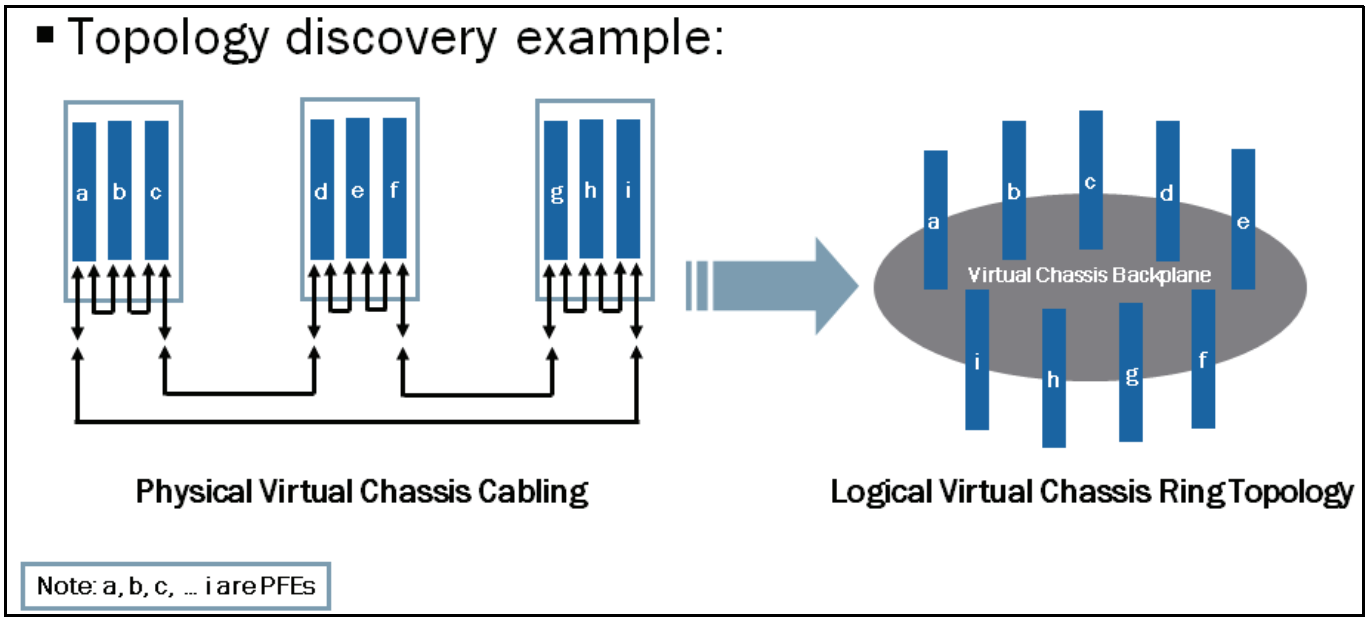

Topology Discovery



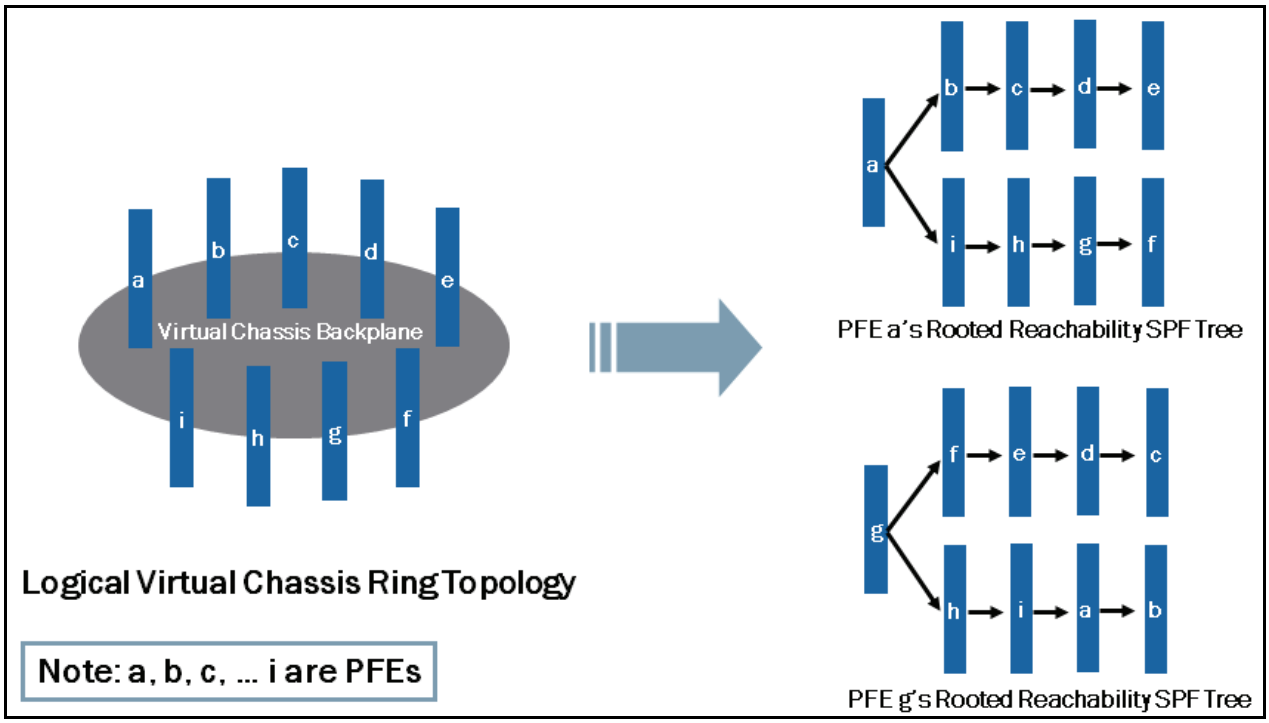
All switches participating in the Virtual Chassis system use the Virtual Chassis Control Protocol (VCCP) to discover the system's topology and ensure the topology is free of loops. Each member exchanges link-state advertisement (LSA) based discovery messages between all interconnected PFEs within a Virtual Chassis system. Based on these LSA-based discovery messages, each PFE builds a member switch topology in addition to a PFE topology map. These topology maps are used when determining the best paths between individual PFEs.

Once the PFE topology map is built, the individual switches run a shortest-path algorithm for each PFE. This algorithm is based on hop count and bandwidth. The result is a map table for each PFE that outlines the shortest path to all other PFEs within the Virtual Chassis system. In the event of a failure, a new SPF calculation is performed.

To prevent broadcast and multicast loops, each switch creates a unique source ID egress filter table on each PFE.

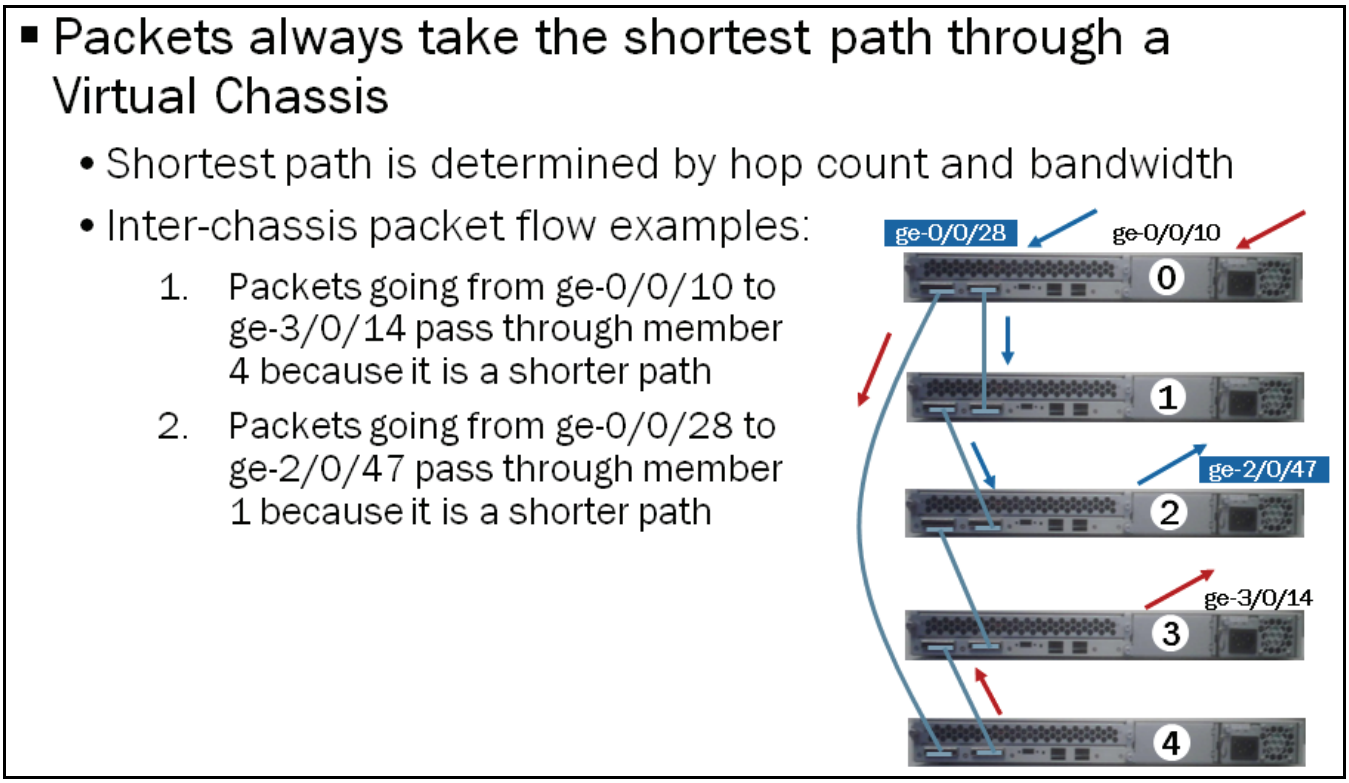


The graphic illustrates the physical cabling and logical ring topology of a Virtual Chassis.



Using the SPF algorithm, each PFE builds its own shortest-path tree to all other PFEs, based on hop count and bandwidth. This process is automatic and is not configurable. The graphic illustrates the basics of this process.

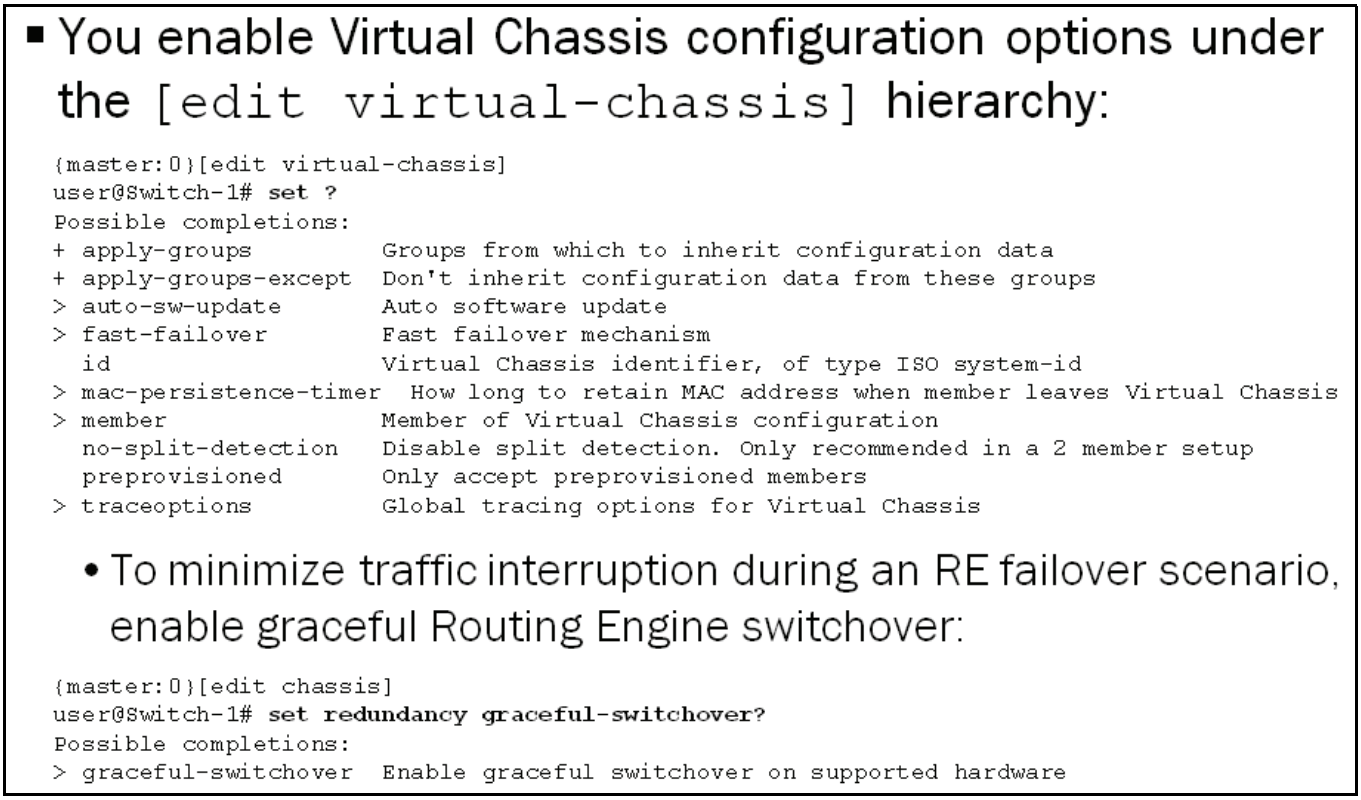
Inter-Chassis Packet Flow



As packets flow from one member switch to another through the Virtual Chassis system, they always takes the shortest path. The shortest path within a Virtual Chassis is based on a combination of hop count and bandwidth. The first example on the graphic shows a packet that enters the Virtual Chassis through port ge-0/0/10, which is a fixed Gigabit Ethernet port on the member switch assigned member ID 0. The packet is destined for the egress port ge-3/0/14, which is a fixed Gigabit Ethernet port on the member switch assigned member ID 3. Based on the physical topology, this packet passes through member switch 4 to member switch 3, which owns the egress port in this example.

In the second example, we see similar results in which the shortest path, which traverses the member switch assigned member ID 1, is selected.

Virtual Chassis Configuration



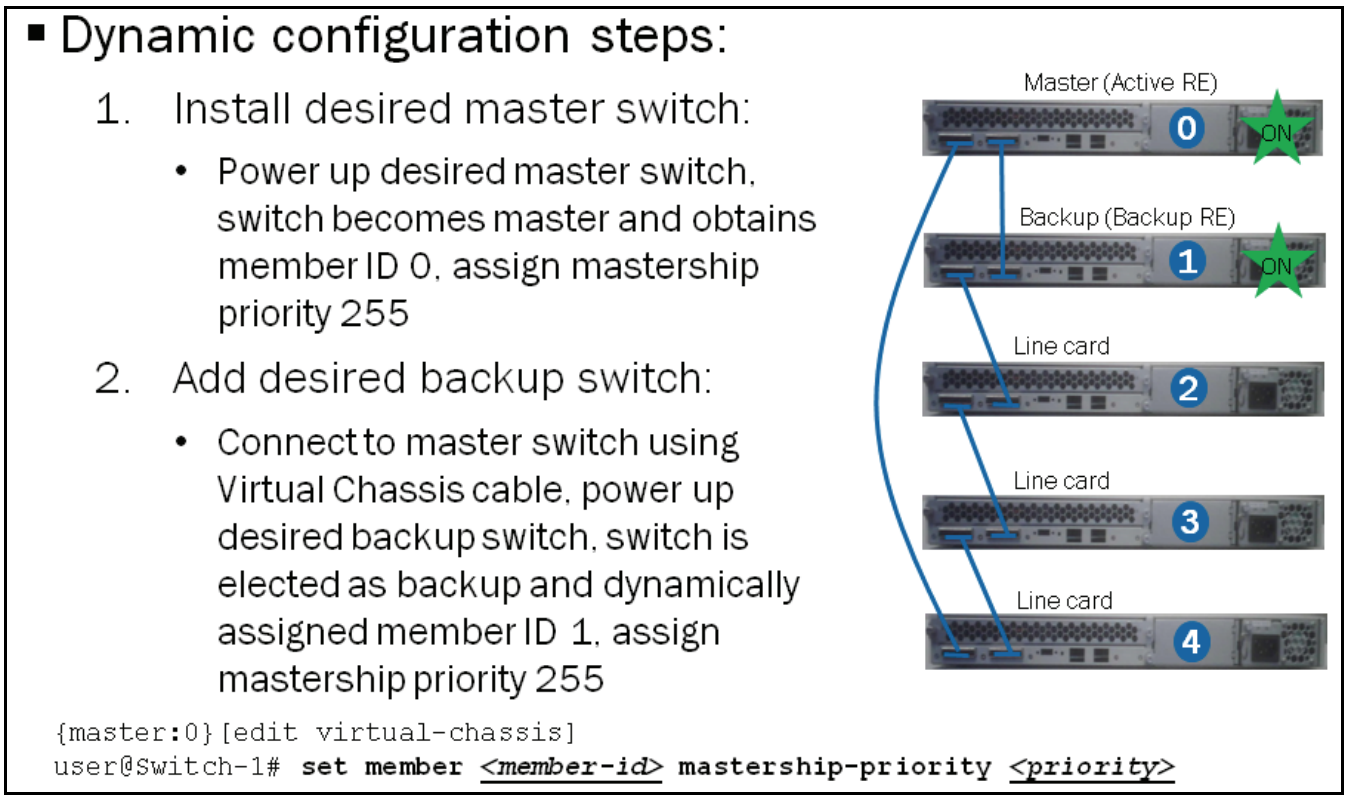
This graphic illustrates the hierarchy used and configurable options available when configuring a Virtual Chassis. A more detailed explanation for some key configuration options follows:

- `auto-sw-update`: This option enables the auto software update feature which is used to automatically upgrade member switches that have a software mismatch (for example, the member switch's version does not match the version currently used by the master switch elected for a given Virtual Chassis). This feature is not enabled by default.
- `fast-failover`: This option enables the fast failover feature which is a hardware-assisted failover mechanism that automatically reroutes traffic and reduces traffic loss in the event of a link failure or switch failure. If a link between two members fails, traffic flow between those members must be rerouted quickly to minimize traffic loss. When fast failover is activated, each VCP is automatically configured with a backup port of the same type (dedicated VCP, SFP uplink VCP, or XFP uplink VCP). If a VCP fails, its backup port is used to send traffic. These backup ports act as standby ports and are not meant for load-balancing purposes. Fast failover is effective only for Virtual Chassis members configured in ring topologies using identical port types. This feature is enabled by default for built-in VCPs but must be enabled for uplink ports converted to VCPs, if desired.
- `id`: This feature allows you to explicitly assign a Virtual Chassis ID so that, if two Virtual Chassis configurations merge, the ID you assign takes precedence over the automatically assigned Virtual Chassis IDs and becomes the ID of the newly merged Virtual Chassis configuration.
- `mac-persistence-timer`: If the master switch is physically disconnected or removed from the Virtual Chassis, this feature determines how long the backup (new master) switch continues to use the address of the old master switch. When the MAC persistence timer expires, the backup (new master) switch begins to use its own MAC address. No minimum or maximum timer limits exist and the default timer is 10 minutes.
- `no-split-detection`: This feature is used to disable the split and merge feature which is enabled by default. The split and merge feature provides a method to prevent the two parts of a separated Virtual Chassis from adversely affecting the network. This feature also allows the two parts to merge back into a single Virtual Chassis configuration.

- `preprovisioned`: This feature is used to deterministically control both the role and the member ID assigned to each member switch in a Virtual Chassis. A preprovisioned configuration links the serial number of each member switch to a specified member ID and role. The serial number must be specified in the configuration file for the member to be recognized as part of the Virtual Chassis. Using this option, you select two member switches as eligible for mastership election process. When you list these two members in the preprovisioned configuration file, you designate both members as routing-engine. One member will function as the master switch and the other will function as the backup switch. You designate all other members with the line card role.

In addition to the previously listed features, you can also enable graceful Routing Engine switchover (GRES) feature as shown on the graphic. GRES enables a device running the Junos OS with redundant REs to continue forwarding traffic even if one RE fails. GRES preserves interface and kernel information and ensures minimal traffic interruption during a mastership change. Note that GRES does not preserve the control plane.

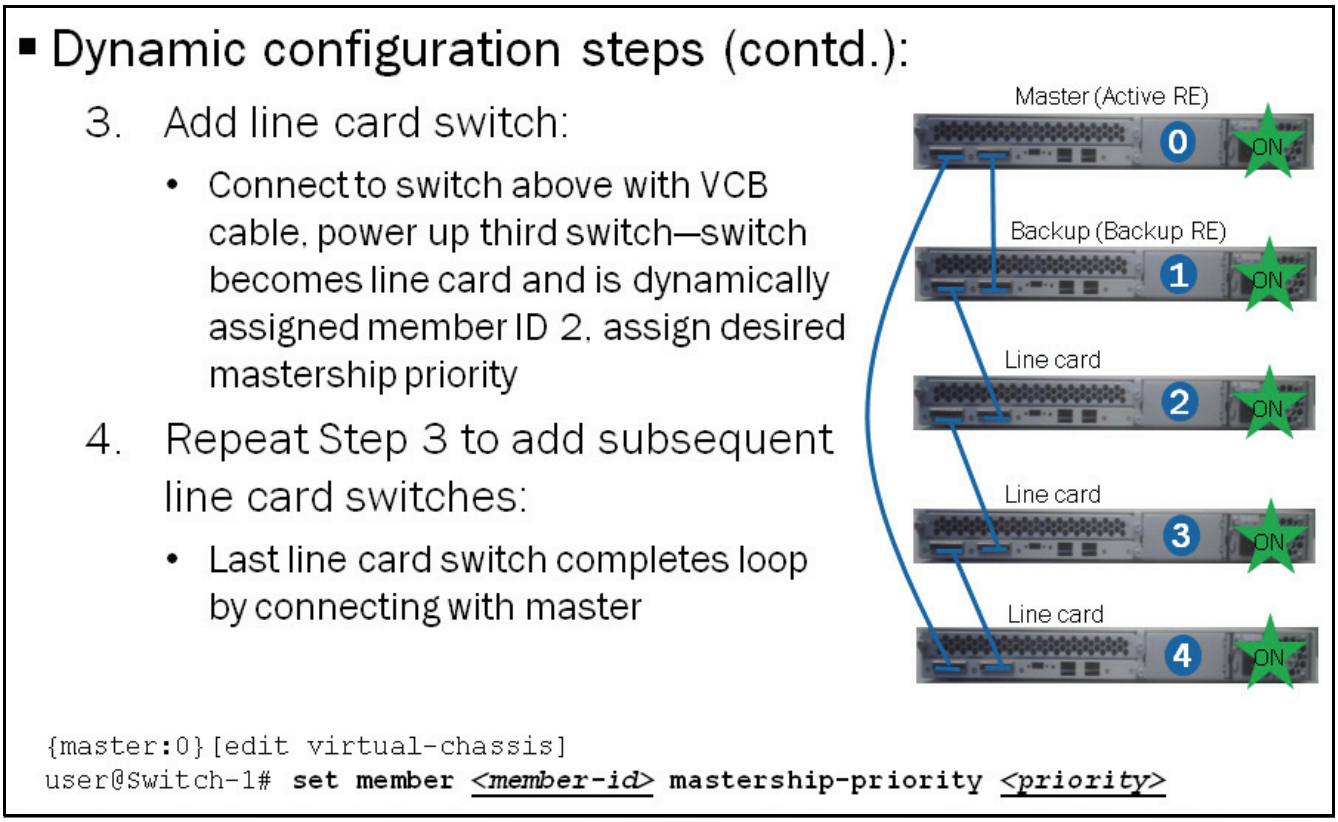
Dynamic Configuration Process



This graphic and the next graphic provide the steps and related details for the dynamic installation and configuration process.

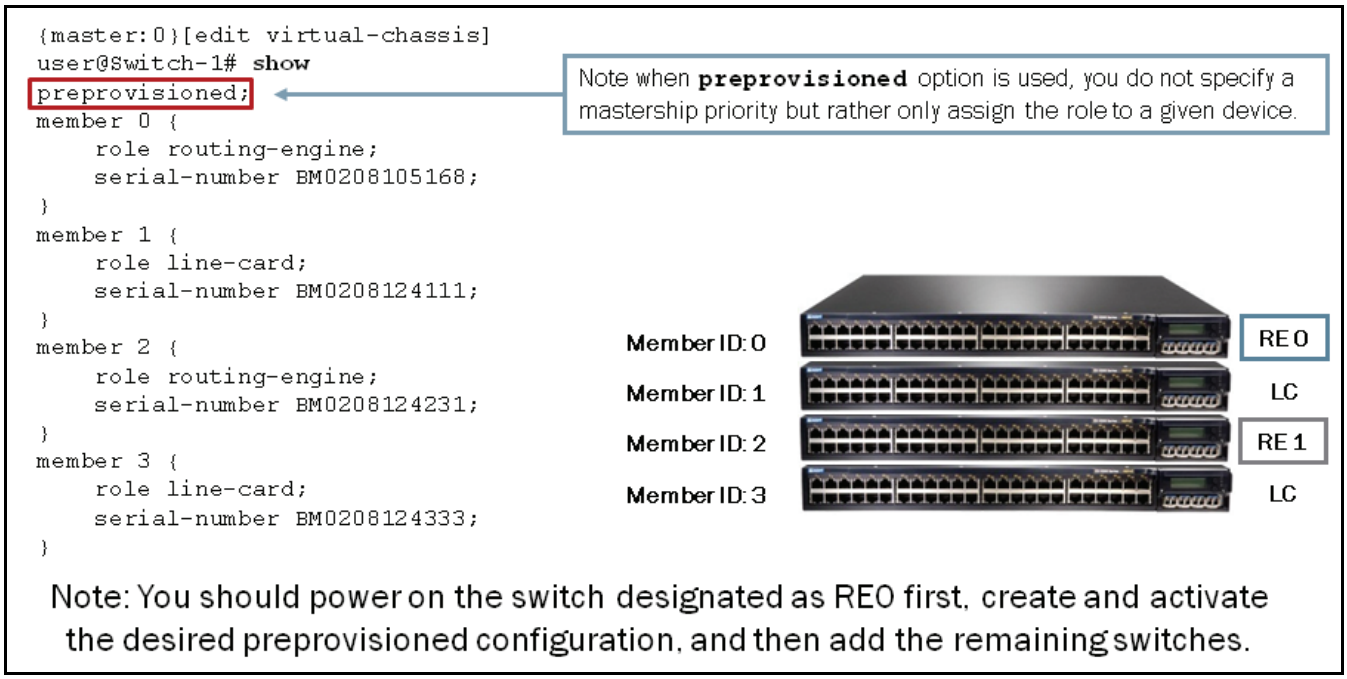
When an EX4200 switch powers on, it receives the default mastership priority value of 128. When a standalone EX4200 switch is connected to an existing Virtual Chassis configuration (which implicitly includes its own master), we recommend that you explicitly configure the mastership priority of the members that you want to function as the master and backup switches. We recommend the following guidelines for assigning mastership priority:

- Specify the same mastership priority value for the master and backup switches in a Virtual Chassis configuration. Doing so helps to ensure a smooth transition from master to backup if the master switch becomes unavailable. This configuration also prevents the original master switch from retaking control from the backup switch when the original master switch comes back online, a situation sometimes referred to as flapping or pre-emption that can reduce the efficiency of system operation.
- Configure the highest possible mastership priority value (255) for the master and backup switches. This configuration ensures that these members continue to function as the master and backup switches when new members are added to the Virtual Chassis configuration.



This graphic shows the remainder of the steps used when performing the dynamic installation and configuration process.

Preprovisioned Configuration Example



This graphic illustrates a preprovisioned configuration example along with some related details.

Monitoring Virtual Chassis Operations

- Use the `show virtual-chassis` commands to monitor Virtual Chassis operations:

```
{master:0}
user@Switch-1> show virtual-chassis ?
Possible completions:
  <[Enter]>            Execute this command
  active-topology      Virtual Chassis active topology
  device-topology      PFE device topology
  fast-failover        Fast failover status
  login
  protocol             Show Virtual Chassis protocol information
  status               Virtual Chassis information
  vc-path              Show virtual-chassis packet path
  vc-port              Virtual Chassis port information
  |                    Pipe through a command
```

This graphic illustrates some key commands used to monitor Virtual Chassis operations. We illustrate some of these key commands on subsequent graphics with sample output.

Verifying Virtual Chassis Port State

```
{master:0}
user@Switch-1> show virtual-chassis vc-port
fpc0:
--------------------------------------------------------------------------
Interface   Type              Trunk  Status       Speed        Neighbor
or                             ID                 (mbps)       ID  Interface
PIC / Port
vcp-0       Dedicated           2    Up           32000        1   vcp-0
vcp-1       Dedicated           1    Up           32000        1   vcp-1

fpc1:
--------------------------------------------------------------------------
Interface   Type              Trunk  Status       Speed        Neighbor
or                             ID                 (mbps)       ID  Interface
PIC / Port
vcp-0       Dedicated           2    Up           32000        0   vcp-0
vcp-1       Dedicated           1    Up           32000        0   vcp-1
```

This graphic illustrates the command used to view Virtual Chassis port and state information along with a sample output. In the sample output, you can see that the VCPs for both members participating in the Virtual Chassis are in the `up` state.

Enabling and Disabling Virtual Chassis Ports

```
{master:0}VC
user@Switch-1> request virtual-chassis vc-port set interface vcp-0 disable

{master:0}
user@Switch-1> show virtual-chassis vc-port
fpc0:
--------------------------------------------------------------------------
Interface   Type              Trunk  Status       Speed        Neighbor
or                             ID                (mbps)       ID  Interface
PIC / Port
vcp-0       Dedicated           2    Disabled     32000
vcp-1       Dedicated           1    Up           32000        1   vcp-1
...
{master:0}
user@Switch-1> request virtual-chassis vc-port set interface vcp-0

{master:0}
user@Switch-1> show virtual-chassis vc-port
fpc0:
--------------------------------------------------------------------------
Interface   Type              Trunk  Status       Speed        Neighbor
or                             ID                (mbps)       ID  Interface
PIC / Port
vcp-0       Dedicated           2    Down         32000
vcp-1       Dedicated           1    Up           32000        1   vcp-1
...
```

This graphic illustrates the commands used to disable or re-enable the built-in VCPs. In the sample output, you see two states; `Disabled` and `Down`. The `Disabled` state indicates that the associated VCP has been administratively disabled while the `Down` state indicates that the VCP is enabled but the Virtual Chassis cable is either not connected or it is connected to a remote VCP that is disabled.

The extended VCPs, or uplink ports converted to VCPs, must be created and enabled using the CLI. The command used to convert the xe-0/1/0 uplink interface to the vcp-255/1/0 interface follows:

```
{master:0}
user@Switch-1> request virtual-chassis vc-port set pic-slot 1 port 0 local
fpc0:
--------------------------------------------------------------------------

{master:0}
user@Switch-1> show interfaces terse vcp-255/1/0
Interface               Admin Link Proto    Local                 Remote
vcp-255/1/0             up    down
vcp-255/1/0.32768       up    down

{master:0}
user@Switch-1> show interfaces terse xe-0/1/0
error: device xe-0/1/0 not found
```

You can return the resulting VCP back to its original uplink interface format (xe-0/1/0) using the following command:

```
{master:0}
user@Switch-1> request virtual-chassis vc-port delete pic-slot 1 port 0 local
fpc0:
--------------------------------------------------------------------------

{master:0}
user@Switch-1> show interfaces terse vcp-255/1/0
error: device vcp-255/1/0 not found

{master:0}
user@Switch-1> show interfaces terse xe-0/1/0
Interface               Admin Link Proto    Local                 Remote
xe-0/1/0                up    down
```

Viewing Virtual Chassis Status Information

```
{master:0}
user@Switch-1> show configuration virtual-chassis
preprovisioned;
member 0 {
    role routing-engine;
    serial-number BM0208105168;
}
member 1 {
    role line-card;
    serial-number BM0208124231;
}

{master:0}
user@Switch-1> show virtual-chassis status

Preprovisioned Virtual Chassis
Virtual Chassis ID: 8d5c.a77f.8de8
                                          Mastership            Neighbor List
Member ID  Status   Serial No    Model     priority     Role     ID  Interface
0 (FPC 0)  Prsnt    BM0208105168 ex4200-24t      129  Master*    1  vcp-0
                                                                 1  vcp-1
1 (FPC 1)  Prsnt    BM0208124231 ex4200-24t        0  Linecard   0  vcp-0
                                                                 0  vcp-1
```

This graphic illustrates the command used to view Virtual Chassis status information along with a sample configuration and output. In the sample output, you can see that the members 0 and 1 have correctly been assigned to their designated roles (`Master` and `Linecard`, respectively).

Review Questions

1. List some benefits of implementing a Virtual Chassis.
2. What upgrade process discussed will upgrade all devices in a Virtual Chassis with minimal impact to traffic?

Answers

1.

Although there are a number of potential benefits of using a virtual chassis configuration, the highlighted benefits included control-plane redundancy and allowing administrators to manage many physical switches as a single entity.

2.

To minimize the impact to traffic you should use the nonstop software upgrade process (NSSU).

JNCIS-ENT Switching Study Guide

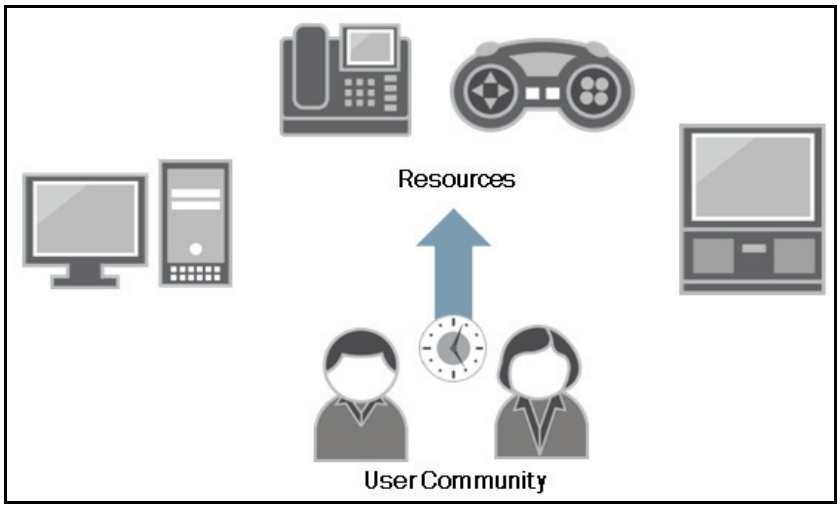
Chapter 7: High Availability Features

This Chapter Discusses:

- Various features that promote high availability; and
- The configuration and monitoring of some high availability features.

High Availability Defined

Users want their systems, (for example, computers, telephones, video games, or televisions) to work properly and at all times. Availability, in general terms, refers to the ability of a user group to access a desired system or resource. When the user group cannot access a given system or resource, the resource is considered unavailable. Often, the term *downtime* is used to describe a period when a system or resource is unavailable. Therefore, high availability is the ability to ensure a high degree of operational continuity for a given user community to a system or some other resource.



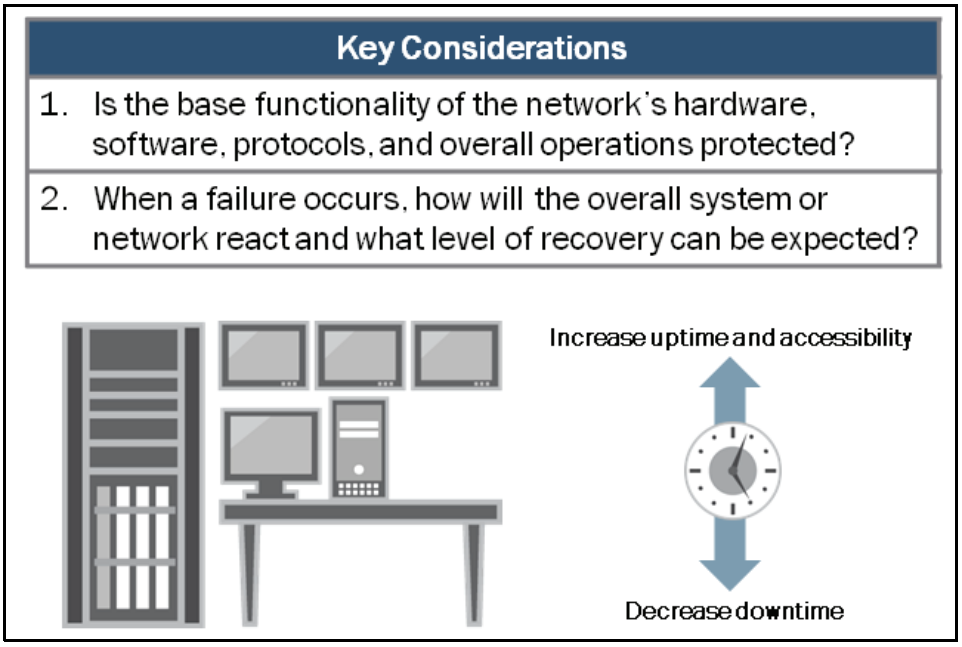
Note that uptime and availability are not the same thing. A system can be up, but unavailable because of other issues. Availability is typically measured as a percentage of uptime over a given duration. The following table provides a mapping for availability percentages and the corresponding amount of time a system is considered to be unavailable:

Availability and Downtime Mappings

Availability %	Downtime per year	Downtime per month (30 days)	Downtime per week
90%	36.5 days	72 hours	16.8 hours
99%	3.65 days	7.20 hours	1.68 hours
99.9%	8.76 hours	43.2 minutes	10.1 minutes
99.99%	52.6 minutes	4.32 minutes	1.01 minutes
99.999%	5.26 minutes	25.9 seconds	6.05 seconds
99.9999%	31.5 seconds	2.59 seconds	0.605 seconds

The obvious objective is to achieve the highest level of availability possible. Note that downtime calculations can vary between organizations, which can make them somewhat misleading. Because the manner in which downtime is calculated can vary, you might find that overall user satisfaction is a better method of evaluating success.

High Availability Networks



When designing high availability networks, you should ensure that all network components function properly and are available to their respective user community. As previously mentioned, there is a difference between uptime and availability, but both are equally important and must coexist for full operational continuity. All high availability networks include provisions in the form of features and physical characteristics that allow for maximum uptime and accessibility.

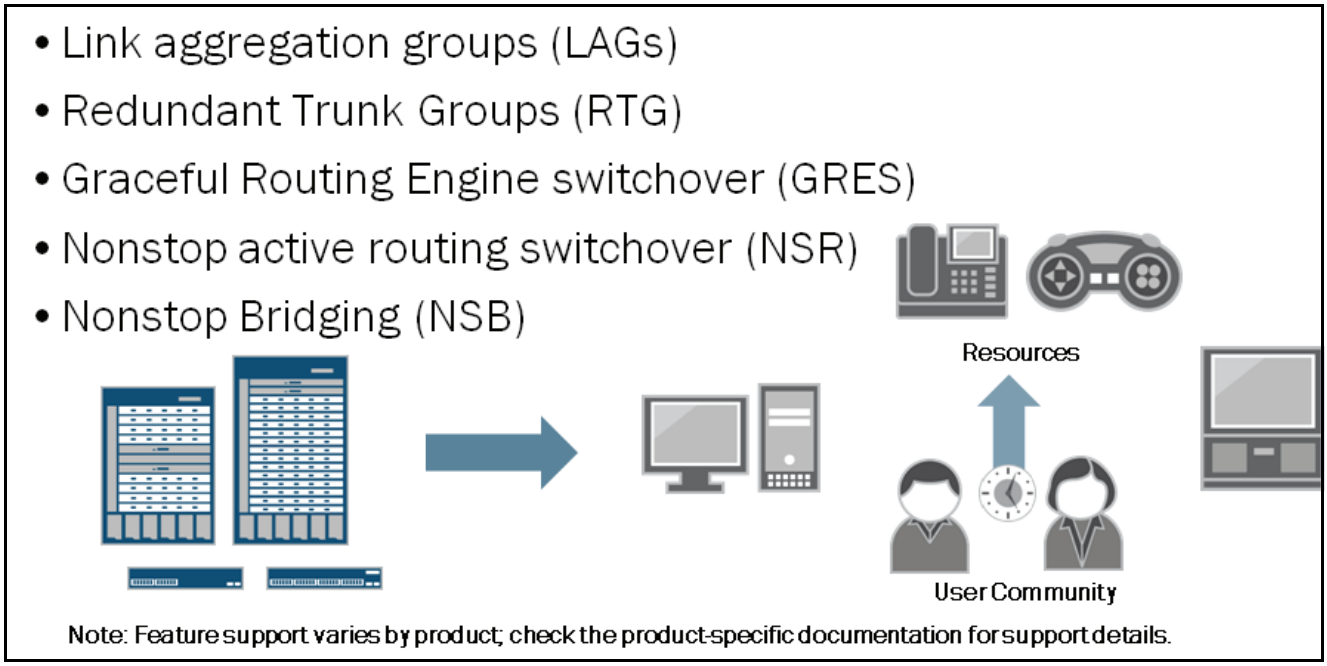
To maximize uptime and accessibility in a network, you should consider the following as you design and implement your network:

1. Is the base functionality of the network's hardware, software, protocols, and overall operations protected?
2. When a failure occurs, how will the overall system or network react, and what level of recovery can be expected?

To properly protect a network, your network design should include some level of redundancy. Although redundancy can provide a large amount of protection for the network, it does come with a cost. Many devices running the Junos operating system include redundant hardware components, such as Routing Engines (REs), control boards (CBs), power supplies, and cooling fans. Refer to the technical publications at http://www.juniper.net/techpubs/ for details on a specific Junos device.

In addition to redundant hardware, you can also use various software features that accommodate redundancy and rapid failure detection. We cover some of these high availability features on the next graphic.

Supported High Availability Features

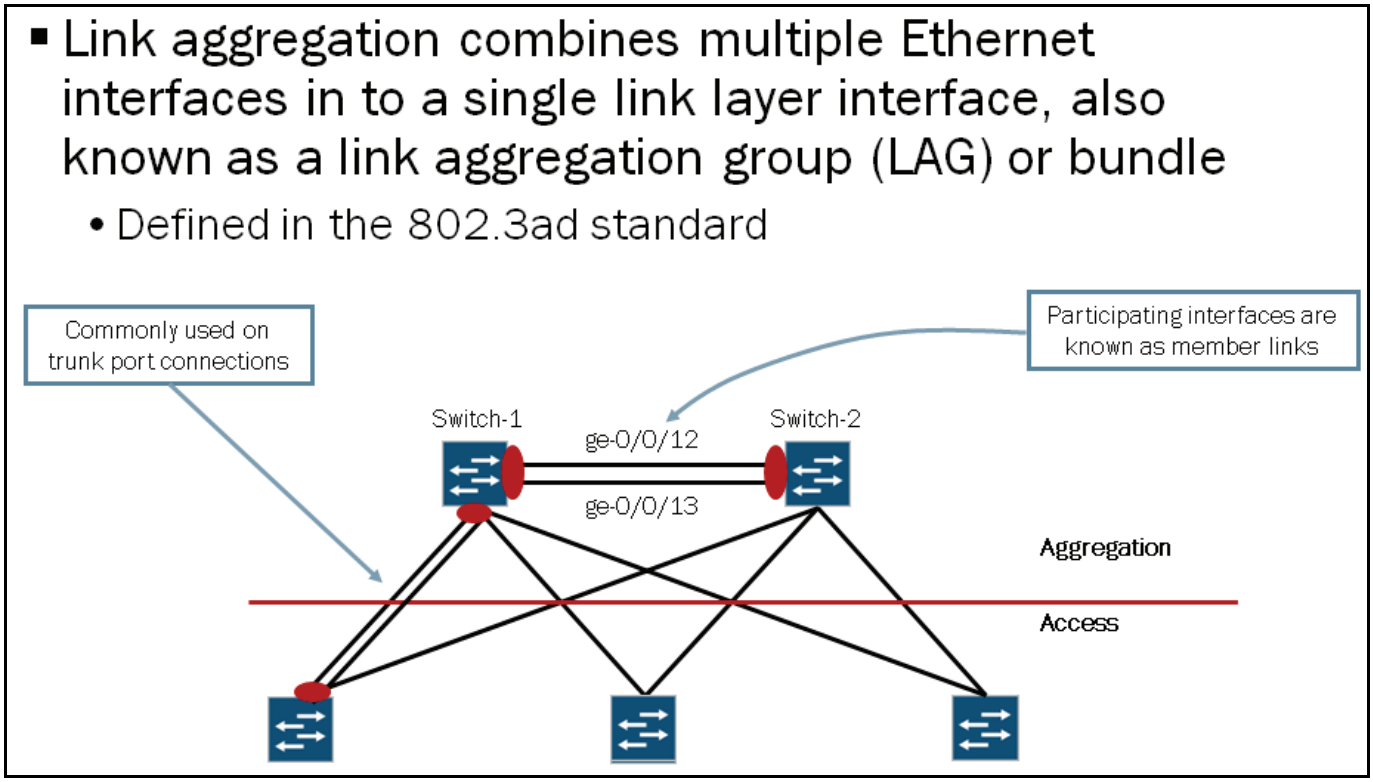


The EX Series Ethernet Switches support a number of features that can increase availability in a network. The following is a brief summary of the high availability features shown on the graphic:

- *Link aggregation groups (LAGs)*: This feature allows you to combine multiple Ethernet interfaces in to a single link layer interface. This feature is defined in the 802.3ad standard.
- *Redundant Trunk Groups (RTG):* This feature provides a quick and simple failover mechanism for redundant Layer 2 links. You can use this feature as a replacement for STP on access switches that are dual homed to multiple aggregation switches.
- *Graceful Routing Engine switchover (GRES):* This feature allows system control to switch from the master RE to the backup RE with minimal interruption to network communications by synchronizing the kernel tables and PFE tables. This feature requires redundant REs or Virtual Chassis.
- *Nonstop active routing (NSR):* This feature provides high availability in a switch with redundant REs or on a Virtual Chassis by enabling transparent switchover of the REs without requiring restart of supported routing protocols by synchronizing the `rpd` process and routing information.
- *Nonstop Bridging (NSB):* This feature provides high availability in a switch with redundant REs or on a Virtual Chassis by enabling transparent switchover of the REs without requiring restart of supported Layer 2 protocols by synchronizing the RE process and switching information.

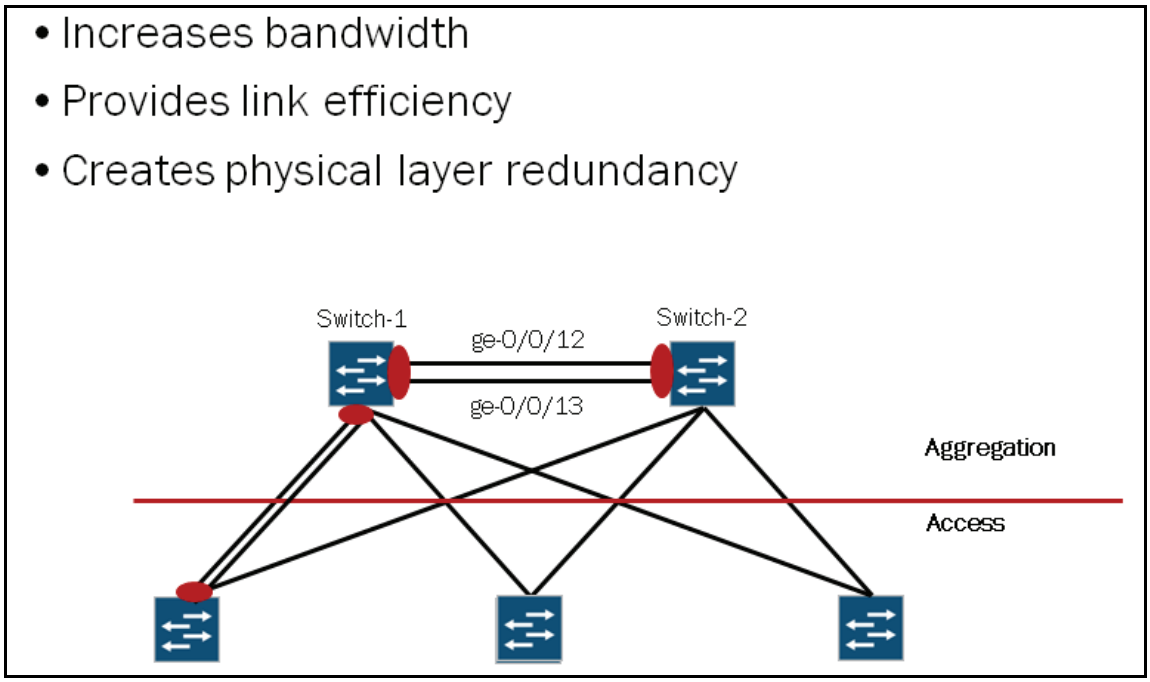
Note that support for these highlighted features can vary between EX Series switches. Refer to the documentation for your specific product for details. We cover the highlighted features in more detail in the subsequent sections. The Junos OS supports several other high availability features and supporting technologies not shown on the graphic. Refer to the technical publications at http://www.juniper.net/techpubs/ for more details.

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 on the graphic, LAGs are commonly used to aggregate trunk links between an access and aggregation switches.

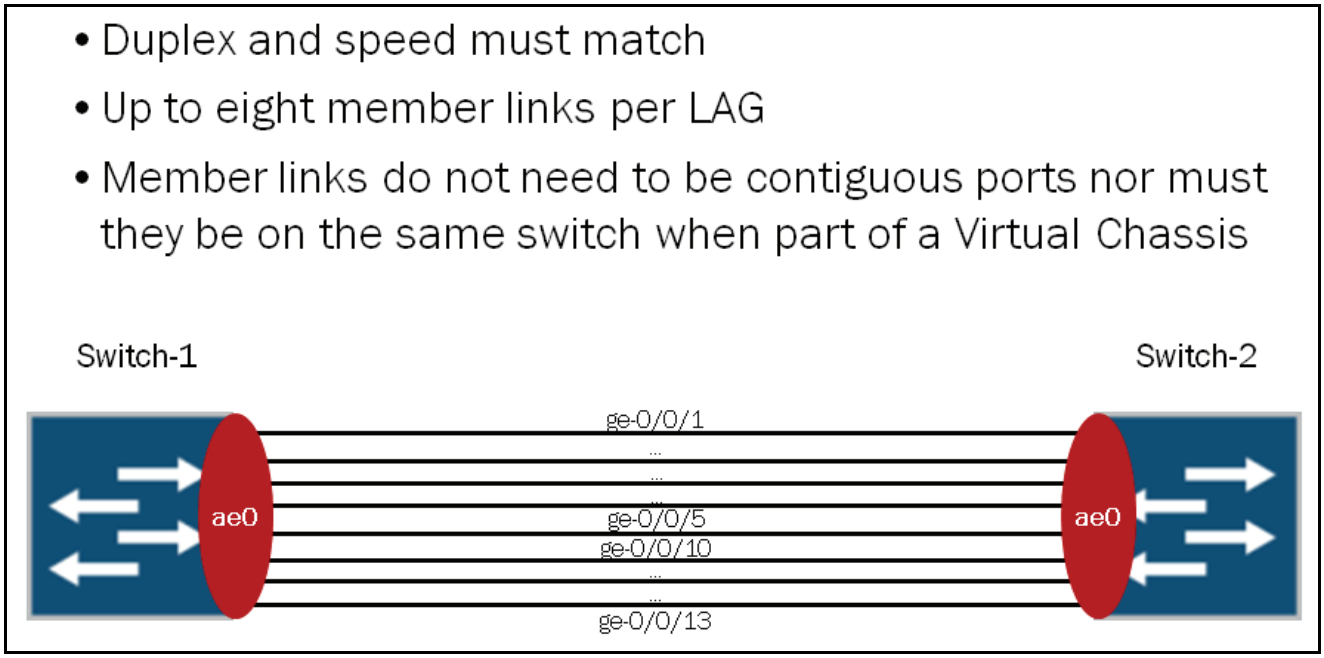
Benefits of Link Aggregation



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

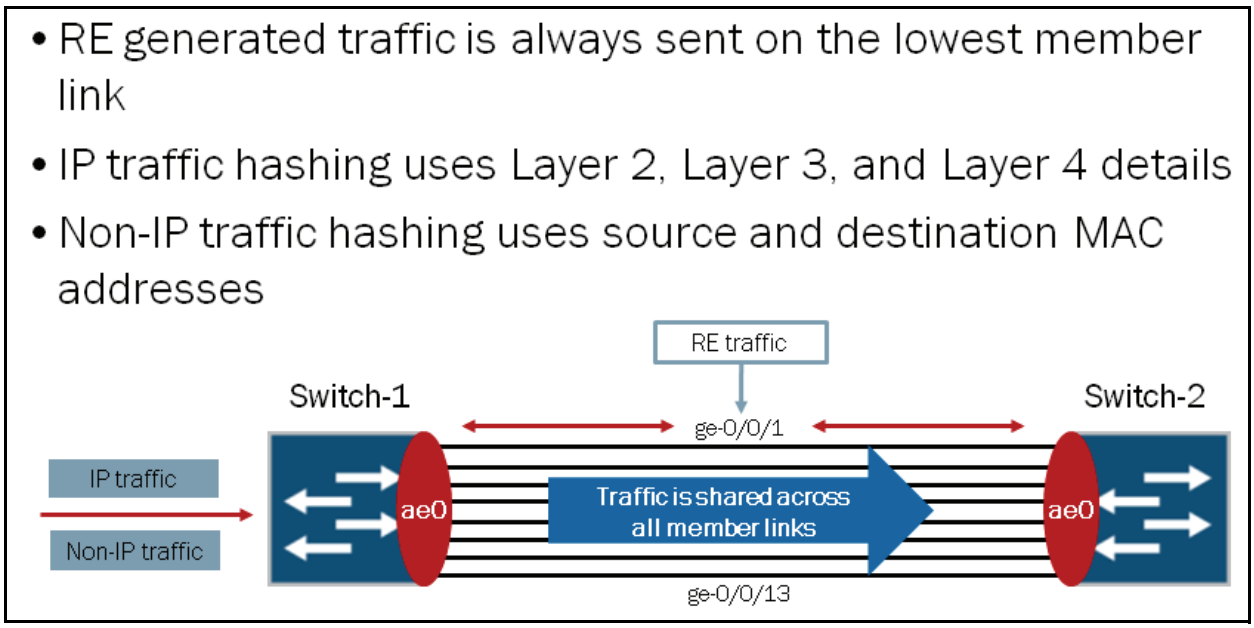


A number of hardware requirements and considerations exist when working with link aggregation. The following list highlights these details:

- 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 Virtual Chassis system.

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

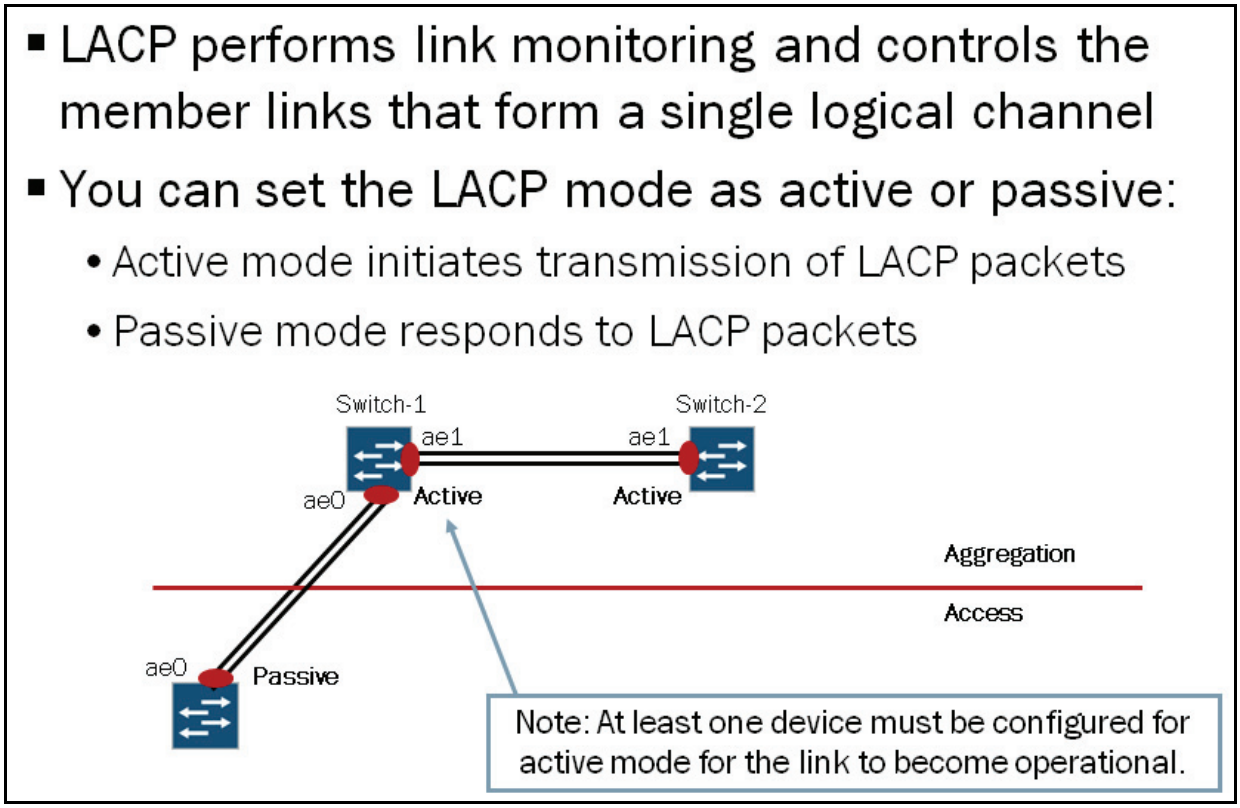
Processing and Forwarding Considerations



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:

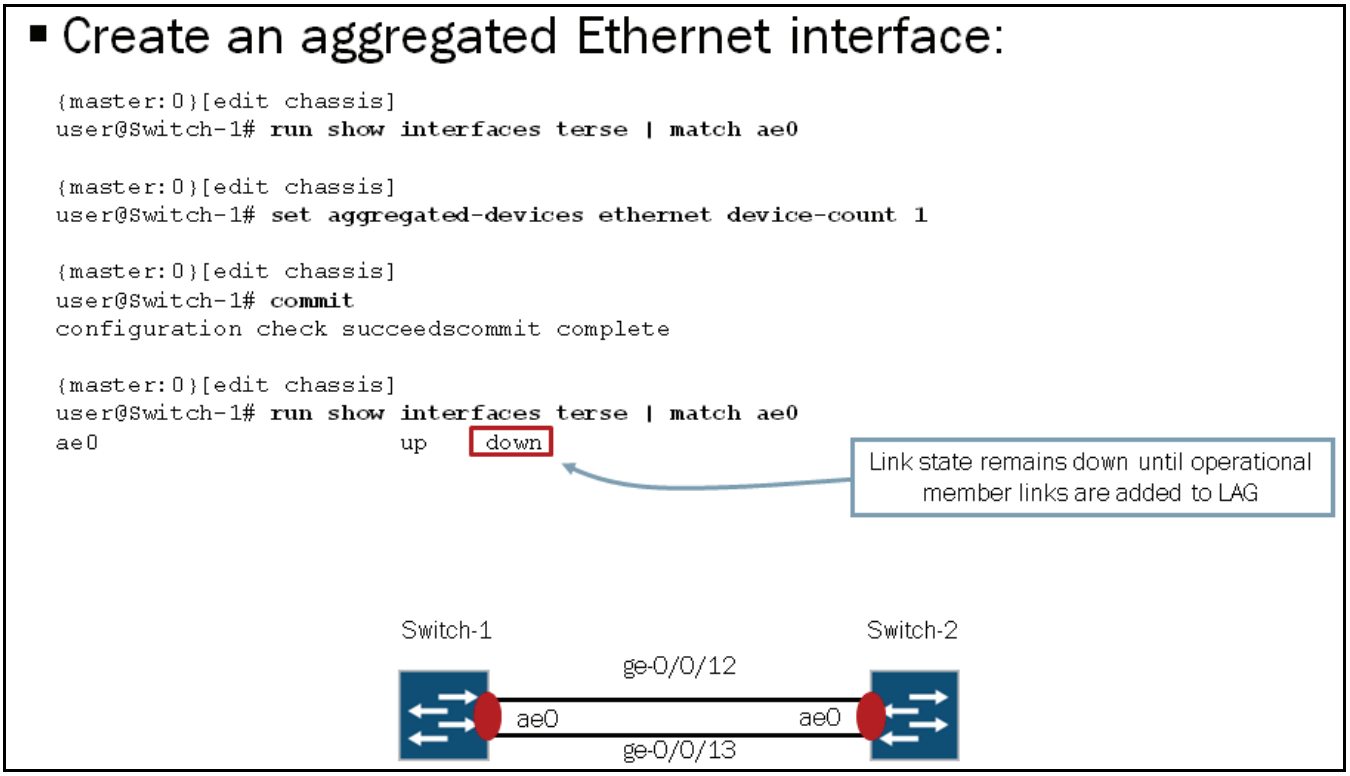
- Automatic addition and deletion of individual links to the aggregate bundle without user intervention
- Link monitoring to check whether both ends of the bundle are connected to the correct group

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.

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 graphic. 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:

```
{master:0}[edit]
user@Switch-1# show chassis
aggregated-devices {
    ethernet {
        device-count 2;
    }
}

{master:0}[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:

```
{master:0}[edit interfaces]
user@Switch-1# set ae0 unit 0 family ethernet-switching

{master:0}[edit interfaces]
user@Switch-1# set ae0 aggregated-ether-options lacp active

{master:0}[edit interfaces]
user@Switch-1# set ge-0/0/12 ether-options 802.3ad ae0

{master:0}[edit interfaces]
user@Switch-1# set ge-0/0/13 ether-options 802.3ad ae0

{master:0}[edit interfaces]
user@Switch-1# commit
configuration check succeedscommit complete

{master:0}[edit interfaces]
user@Switch-1# run show interfaces terse | match ae0
ge-0/0/12.0             up    up   aenet    --> ae0.0
ge-0/0/13.0             up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   eth-switch
```

This graphic illustrates the remainder of the configuration required to implement a LAG for Layer 2 operations. On this graphic, 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.

```
{master:0}[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

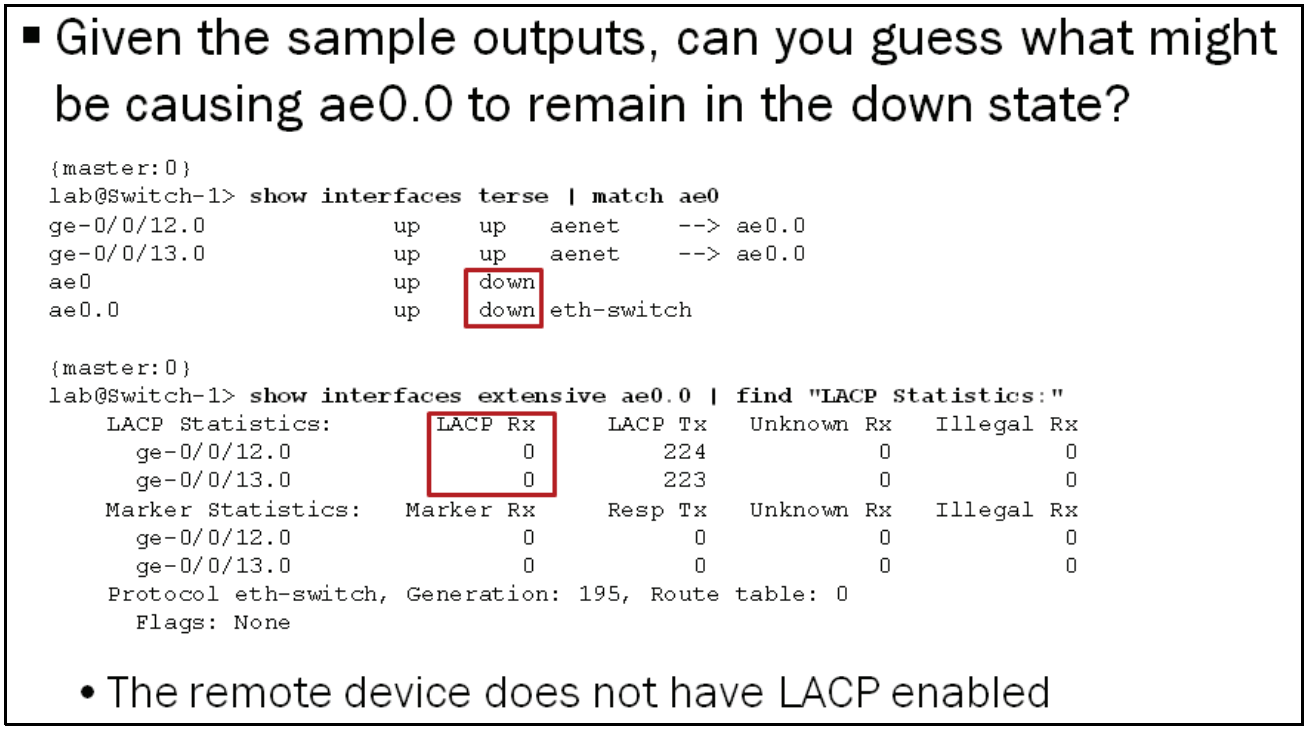
```
{master:0}
user@Switch-1> show interfaces terse | match ae0
ge-0/0/12.0             up    up   aenet    --> ae0.0
ge-0/0/13.0             up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   eth-switch

{master:0}
user@Switch-1> show interfaces extensive ae0.0 | find "LACP Statistics:"
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      ge-0/0/12.0               26         516            0            0
      ge-0/0/13.0               25         519            0            0
    Marker Statistics:   Marker Rx     Resp Tx   Unknown Rx   Illegal Rx
      ge-0/0/12.0                0           0            0            0
      ge-0/0/13.0                0           0            0            0
    Protocol eth-switch, Generation: 195, Route table: 0
      Flags: None
```

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:

```
{master:0}[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
```

Think About It

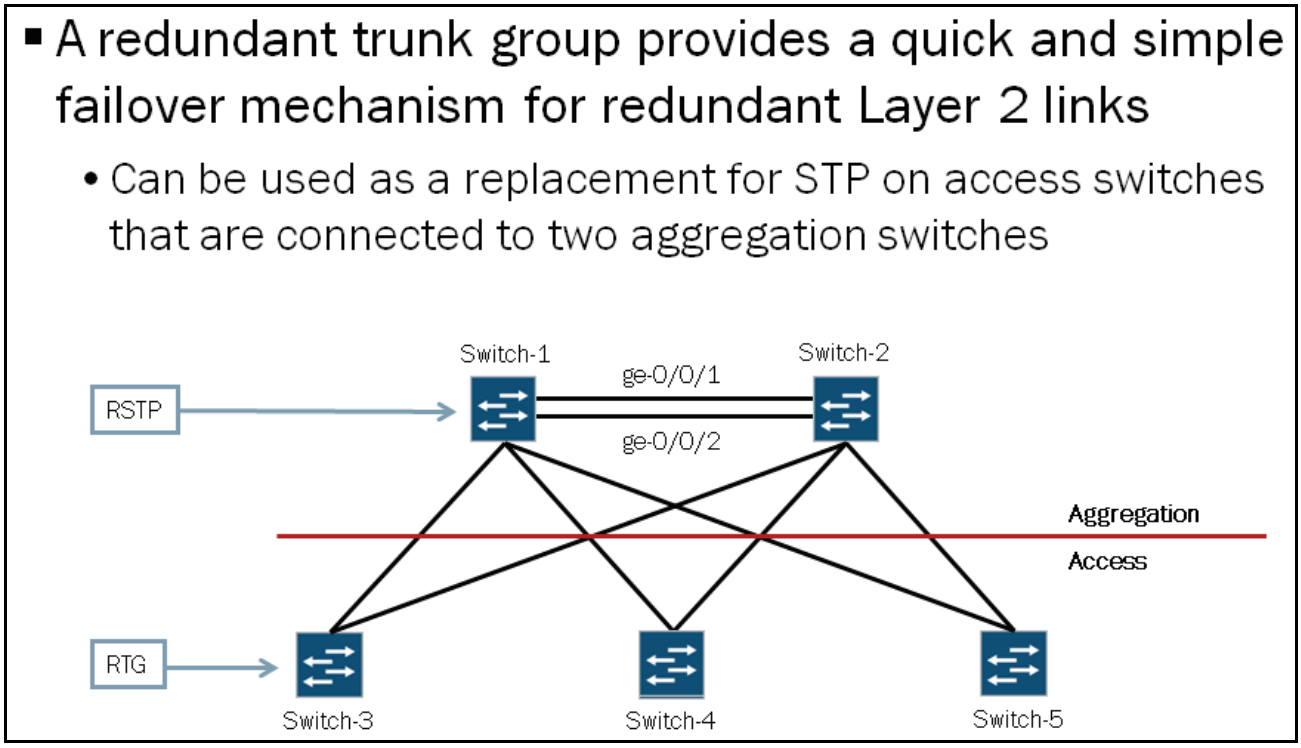


This graphic tests your knowledge on determining why the given LAG shows a state of down. Given the sample statistical output, we see that Switch-1 is not receiving any LACP messages. Although other possible reasons might explain why LACP messages are not being received, in this case a simple misconfiguration exists on Switch-2 (the remote device participating in the LAG).

Once LACP is enabled on Switch-2, we see the state change to up for ae0 and the received LACP statics increment as shown in the following output:

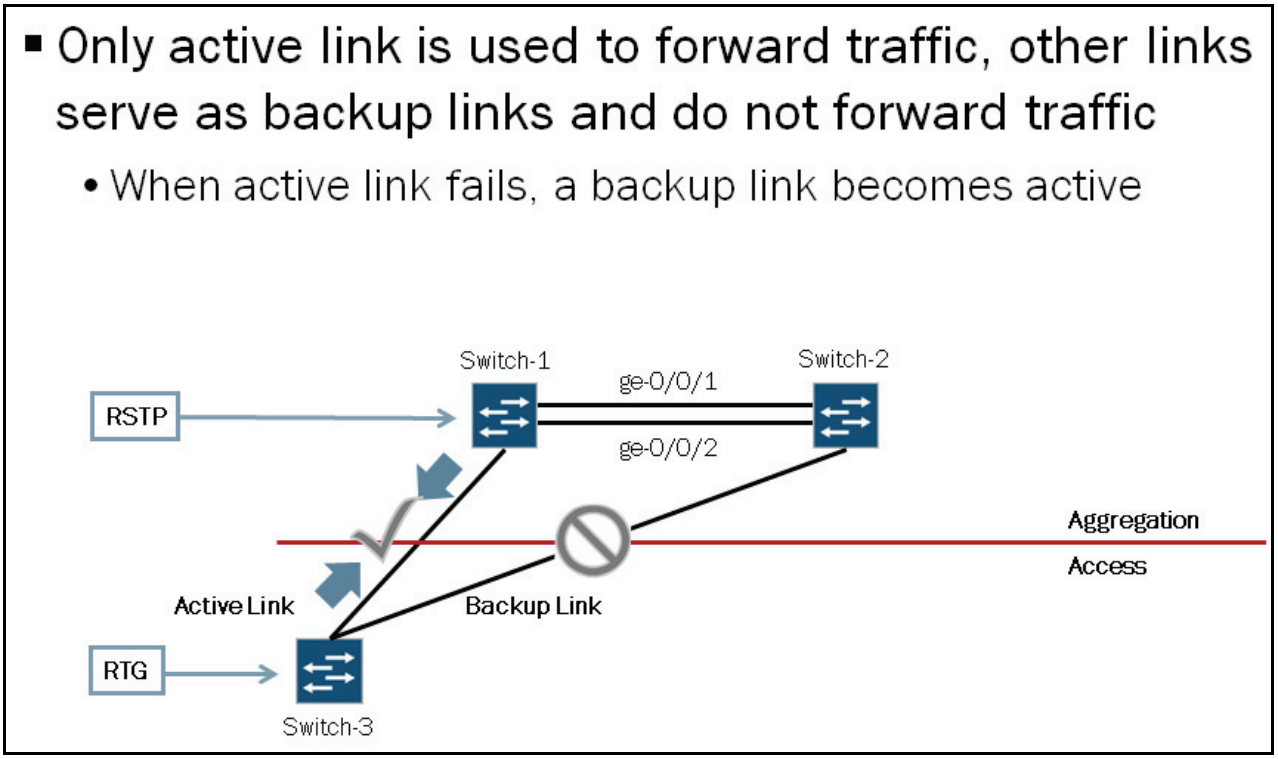
```
{master:0}
user@Switch-1> show interfaces terse | match ae0
ge-0/0/12.0             up    up   aenet    --> ae0.0
ge-0/0/13.0             up    up   aenet    --> ae0.0
ae0                     up    up
ae0.0                   up    up   eth-switch

{master:0}
user@Switch-1> show interfaces extensive ae0.0 | find "LACP Statistics:"
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      ge-0/0/12.0               26         516            0            0
      ge-0/0/13.0               25         519            0            0
```

Redundant Trunk Group



Redundant trunk groups (RTGs) provide a quick and simple failover mechanism between redundant Layer 2 links. This feature is a replacement for STP and is often used on access switches that are connected to two aggregation switches.

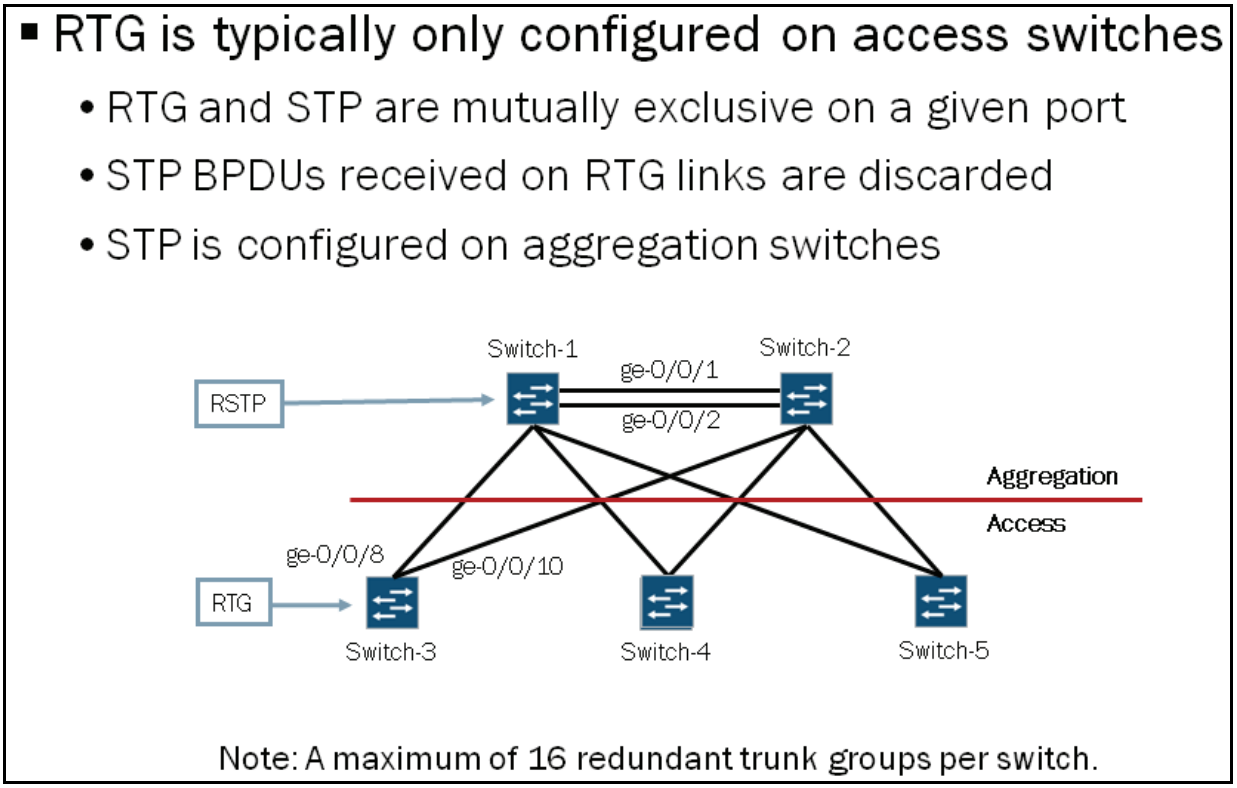
How Does It Work?



Redundant trunk groups are used as an alternative to STP in redundant enterprise networks. You configure the redundant trunk group on access switches, and it contains two links: a primary (or active) link, and a secondary (or backup) link. If the active link fails, the secondary link automatically assumes the active role and starts forwarding data traffic. Typically the convergence time associated with a failover scenario is less than one second.

The Junos OS forwards data traffic only on the active link. The software drops data traffic on the secondary link. The Junos OS tracks these dropped packets. You can view these packets by using the **show interfaces *interface-name*** command. While data traffic is blocked on the secondary link, Layer 2 control traffic is still permitted. For example, you can run LACP or the Link Layer Discovery Protocol (LLDP) between two EX Series switches on the secondary link.

Configuration Considerations



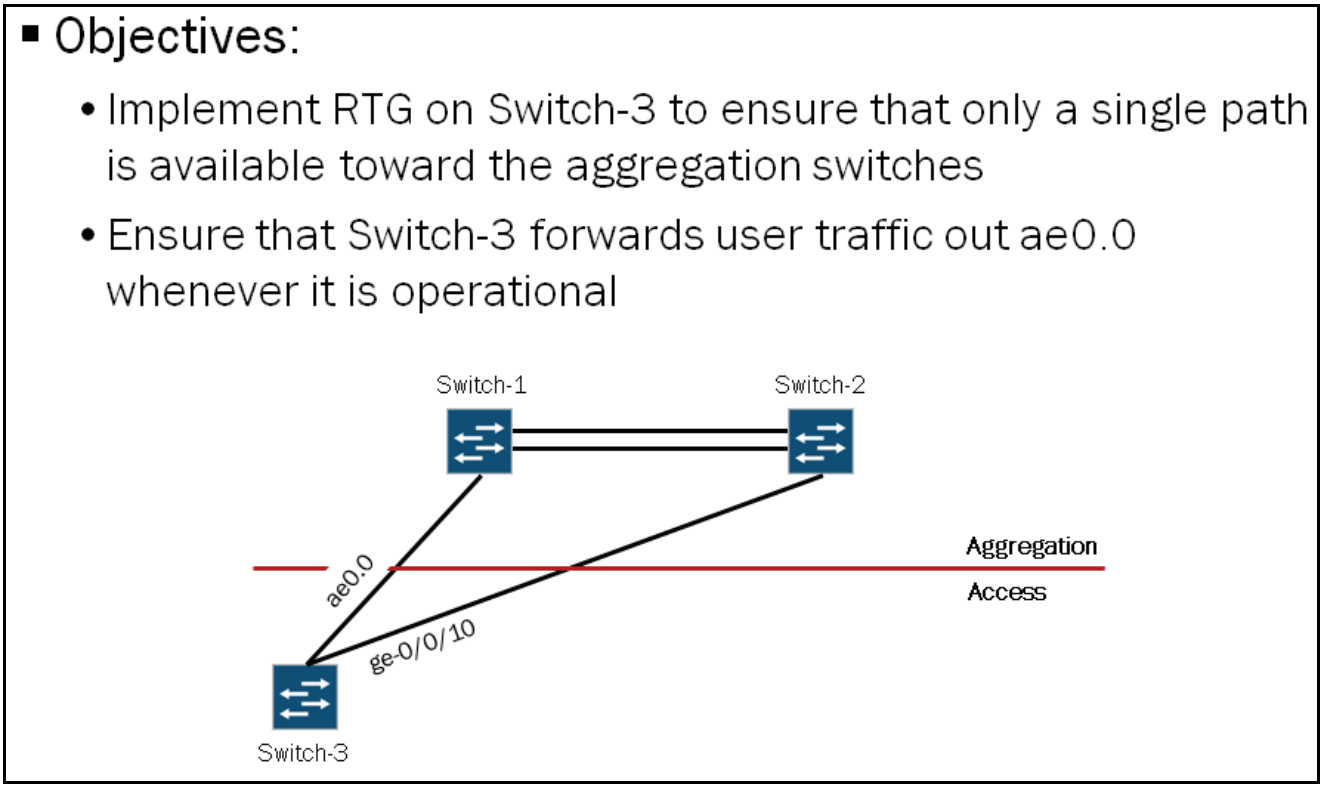
You typically configure redundant trunk groups on access switches. Redundant trunk groups are used as an alternative to STP on trunk ports in redundant enterprise networks. You cannot configure an interface to participate in a redundant trunk group and STP at the same time. If you do configure an interface to participate in a redundant trunk group and STP at the same time, the Junos OS will generate an error when the **commit** command is issued, as shown in the following output:

```
{master:0}[edit ethernet-switching-options]
user@Switch-3# commit
error: XSTP : msti 0 STP and RTG cannot be enabled on the same interface ae0.0
error: configuration check-out failed
```

Similarly, you must configure both interfaces within a redundant trunk group as trunk ports or access ports. If the redundant links are configured as trunk ports, they must be configured to service the same VLANs.

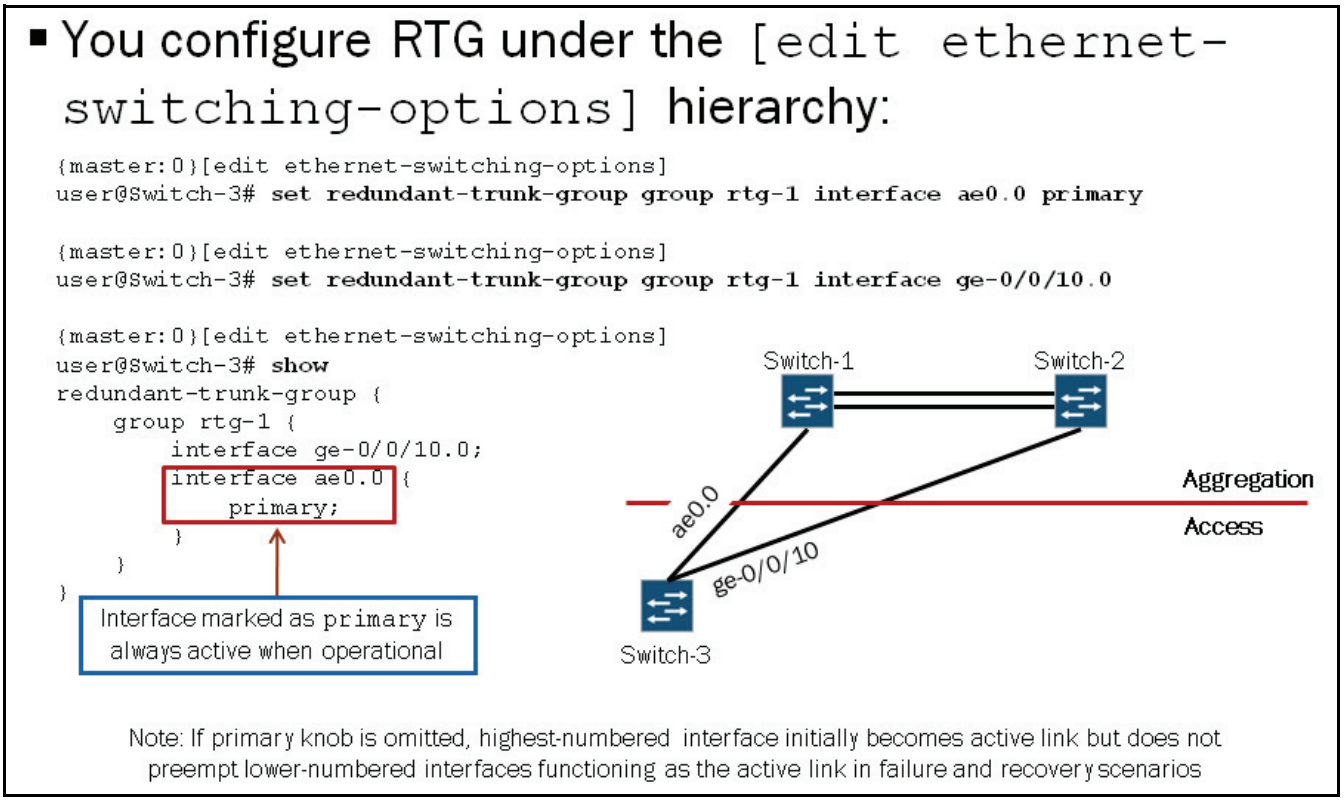
All STP BPDUs received through an interface participating in a redundant trunk group are dropped. Although not strictly necessary, you should configure STP (or a variant) on the aggregation switches. If redundant links exist between aggregation switches (as shown on the diagram) and you do not configure STP, a Layer 2 loop will form and the network performance and operations will be affected.

Case Study: Topology and Objectives



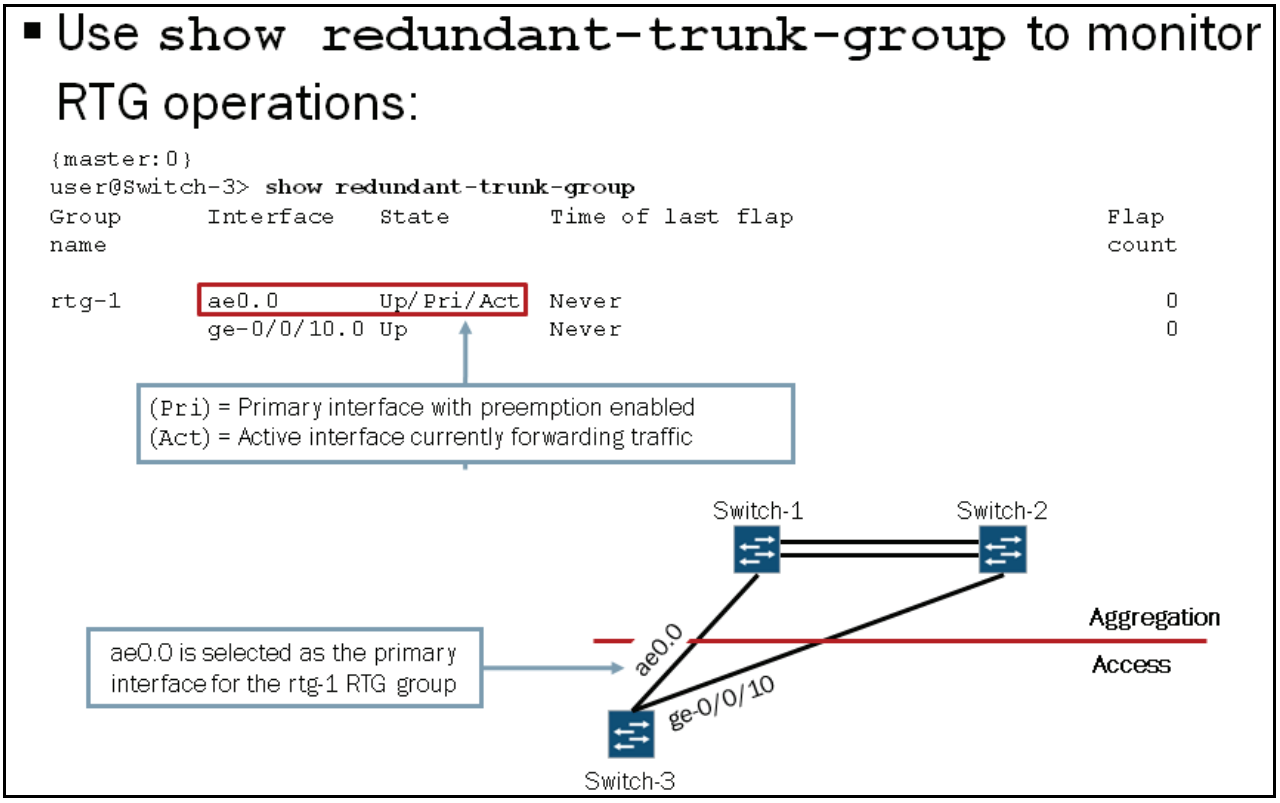
This graphic provides the topology and objectives for this case study.

Case Study: Configuring RTG



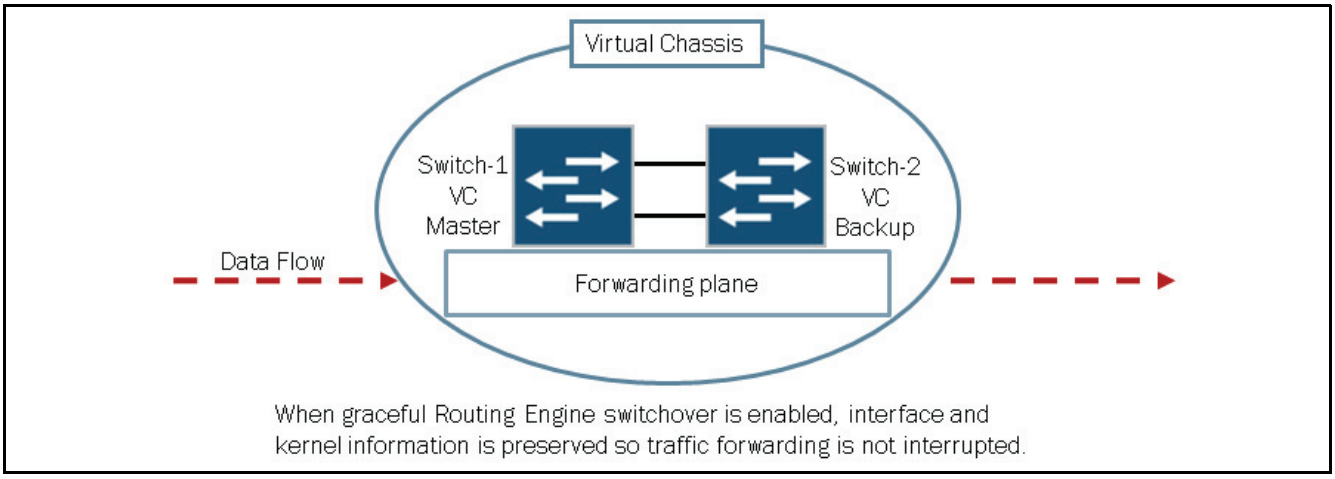
This graphic illustrates the configuration used for our sample topology and to accomplish the stated objectives. Some configuration options and considerations are also displayed on the graphic.

Case Study: Monitoring RTG



This graphic illustrates the primary method used to monitor RTG operations.

Case Study: Monitoring RTG

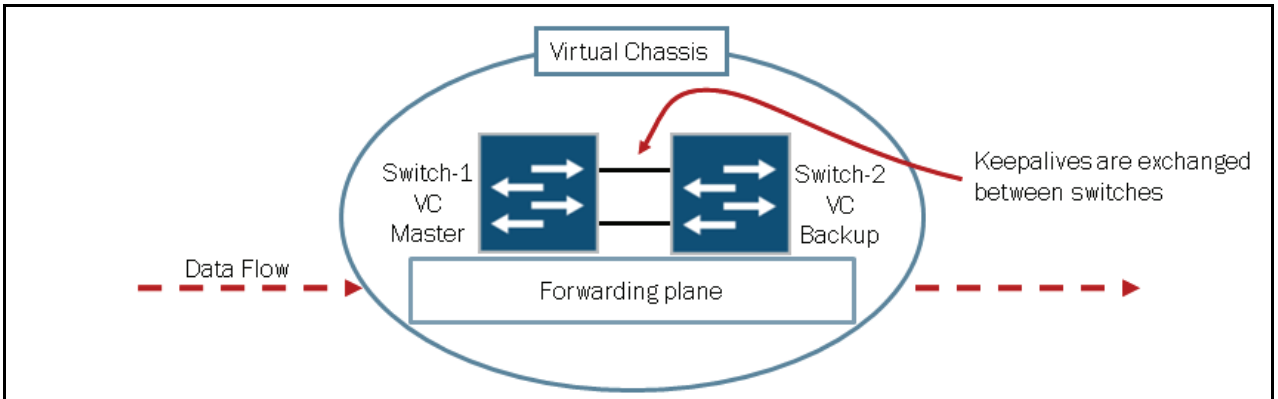


Graceful Routing Engine switchover enables a switch with redundant REs, or participating in a Virtual Chassis, to continue forwarding packets, even if one RE fails by allowing control to switch from the master RE to the backup RE with minimal interruption to network communications. Graceful Routing Engine switchover preserves interface and kernel information and ensures that traffic forwarding is not interrupted during a mastership change. Graceful Routing Engine switchover does not, however, preserve the control plane, which means the routing protocol process (`rpd`) must restart and the information learned through that process must be relearned (unless NSR is also configured). Any updates to the master RE are replicated to the backup RE as soon as they occur. If the kernel on the master RE stops operating, the master RE experiences a hardware failure, or the administrator initiates a manual switchover, mastership switches to the backup RE.

Without graceful Routing Engine switchover enabled, the PFE restarts and all interfaces are discovered by the new master RE when a mastership change occurs. The new master RE restarts `rpd`, so all adjacencies are aware of the topological changes. Because all interfaces go down during this process, the system generates interface alarms. The network also undergoes a topology change because all protocol adjacencies are affected, albeit temporarily.

With graceful Routing Engine switchover enabled, the PFE is not restarted and interface and kernel information is preserved. By allowing the PFE to remain up during a mastership switchover and preserving interface and kernel information, graceful Routing Engine switchover greatly reduces the time the RE failover process takes. With no other high availability features enabled, the new master RE must restart `rpd`, so all adjacencies are aware of the routing change.

Graceful Routing Engine Switchover Operations



A Summary of Graceful Routing Engine Switchover Operations
1. Once graceful Routing Engine switchover is configured and the REs are synchronized, keepalives are exchanged.
2. If the backup RE does not receive a keepalive from the master RE (typically after 2 seconds), it determines that the master RE has failed and takes mastership.
3. The PFE seamlessly disconnects from the old master RE and connects to the new master RE; note that the PFE remains operational.
4. The new master RE and the PFE synchronize, after which the new master RE sends state update messages to the PFE as needed.

Once the REs are synchronized, they exchange keepalives. If the backup RE does not receive a keepalive from the master RE after a specified timeout (typically 2 seconds), it determines that the master RE has failed and takes mastership. When a mastership change occurs, the PFE seamlessly disconnects from the old master RE and reconnects to the new master RE. The PFE does not reboot and continues forwarding traffic based on the existing forwarding table entries. The new master RE then synchronizes its state with the PFE. If the new master RE detects that the PFE state is not up to date, it resends state update messages.

If needed, you can perform a manual RE mastership switchover using the **`request chassis routing-engine master switch`** command, as follows:

```
{master:0}
user@Switch-1> request chassis routing-engine master switch
Toggle mastership between routing engines ? [yes,no] (no) yes

{master:0}
user@Switch-1>
Switch-1 (ttyu0)

login: lab

Logging to master
Password:

--- JUNOS 12.2R1.8 built 2012-08-25 01:27:13 UTC
{master:1}
user@Switch-1>
```

Additional options exist for switching mastership as illustrated in the following command:

```
{master:0}
user@Switch> request chassis routing-engine master ?
Possible completions:
  acquire              Attempt to become master Routing Engine
  release              Request that other Routing Engine become master
  switch               Toggle mastership between Routing Engines
```

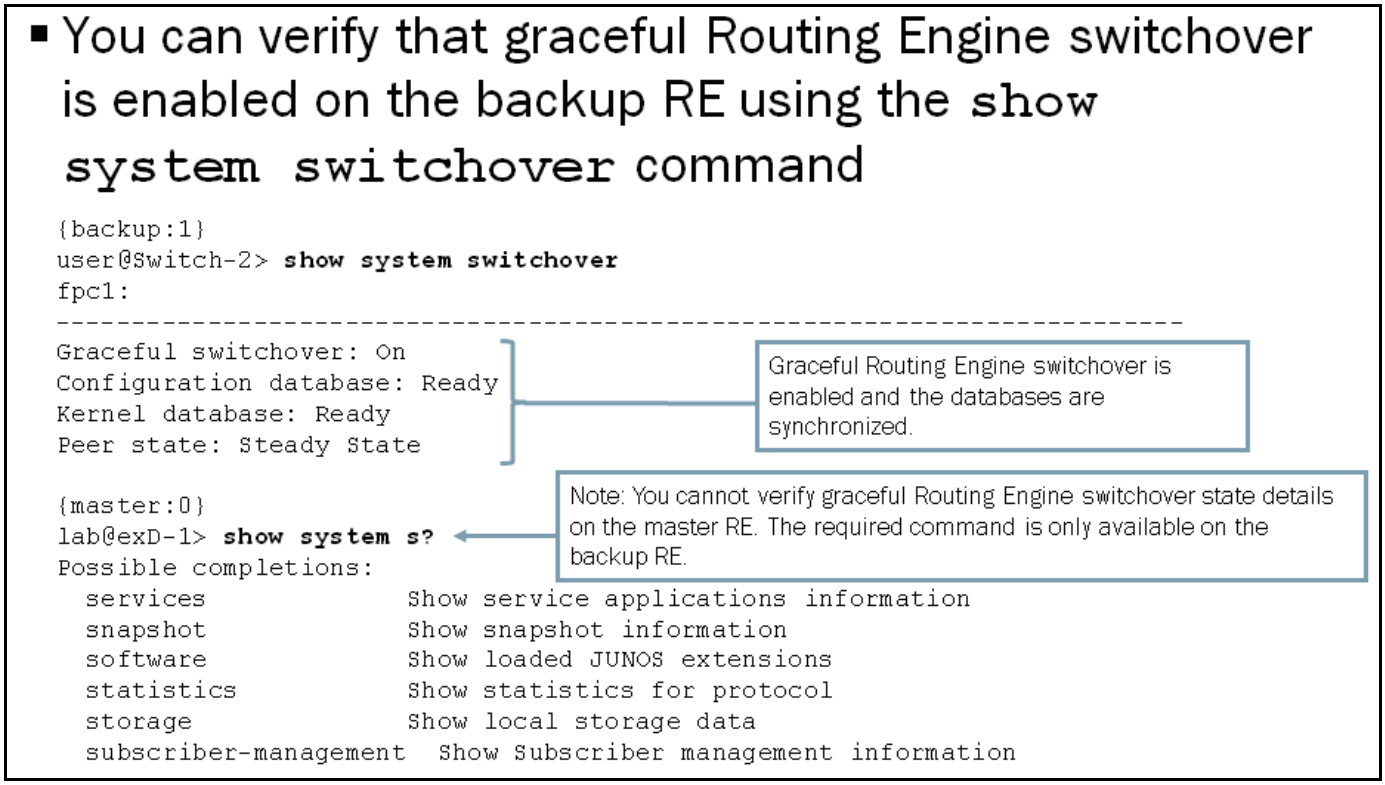
Configuring Graceful Routing Engine Switchover

- Enable graceful Routing Engine switchover under the `[edit chassis]` hierarchy

```
{master:0}[edit chassis]
user@Switch-1# set redundancy graceful-switchover

{master:0}[edit chassis]
user@Switch-1 # show
redundancy {
    graceful-switchover;
}

{master:0}[edit chassis]
user@Switch-1 # commit
fpc0:
configuration check succeeds
fpc1:
commit complete
fpc0:
commit complete
```

By default, graceful Routing Engine switchover is disabled. Enable graceful Routing Engine switchover under the `[edit chassis]` hierarchy using the **`set redundancy graceful-switchover`** command.

Monitoring Graceful Routing Engine Switchover

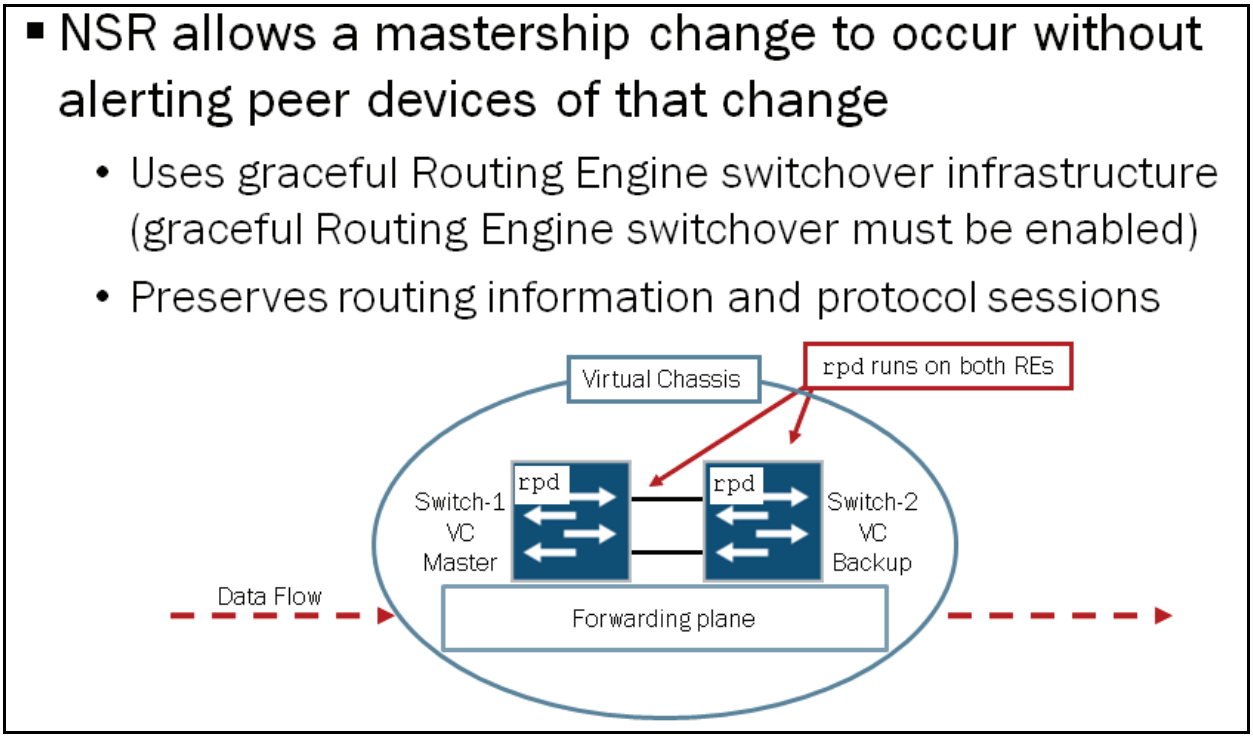


The graphic illustrates the **show system switchover** command, which is used to verify whether graceful Routing Engine switchover is enabled and that the databases are synchronized. Note that you can issue this command only on the backup RE.

For graceful Routing Engine switchover to function properly, the master RE replicates its state to the backup RE and PFE. The following are the three specific states that must be replicated:

- *Configuration database*: The configuration database, or repository of configuration files, is replicated through the **commit synchronize** process. Several system processes require the configuration database to perform their designated functions; for example, the device control daemon (`dcd`) checks this database when it brings interfaces online, the chassis process (`chassisd`) uses this database to manage hardware components, and `rpd` uses this database to control routing protocols.
- *Kernel and related entries*: Enabling graceful Routing Engine switchover starts a custom Junos process known as `ksyncd`. The `ksyncd` process is responsible for kernel state replication tasks between various hardware components.
- *PFE state*: The Junos OS uses `chassisd` to perform PFE state replication. When a mastership change occurs, `chassisd` performs a soft restart to query the system's hardware inventory. When the hardware components respond to the query, the system re-attaches them to the backup RE and brings them online without any disruption.

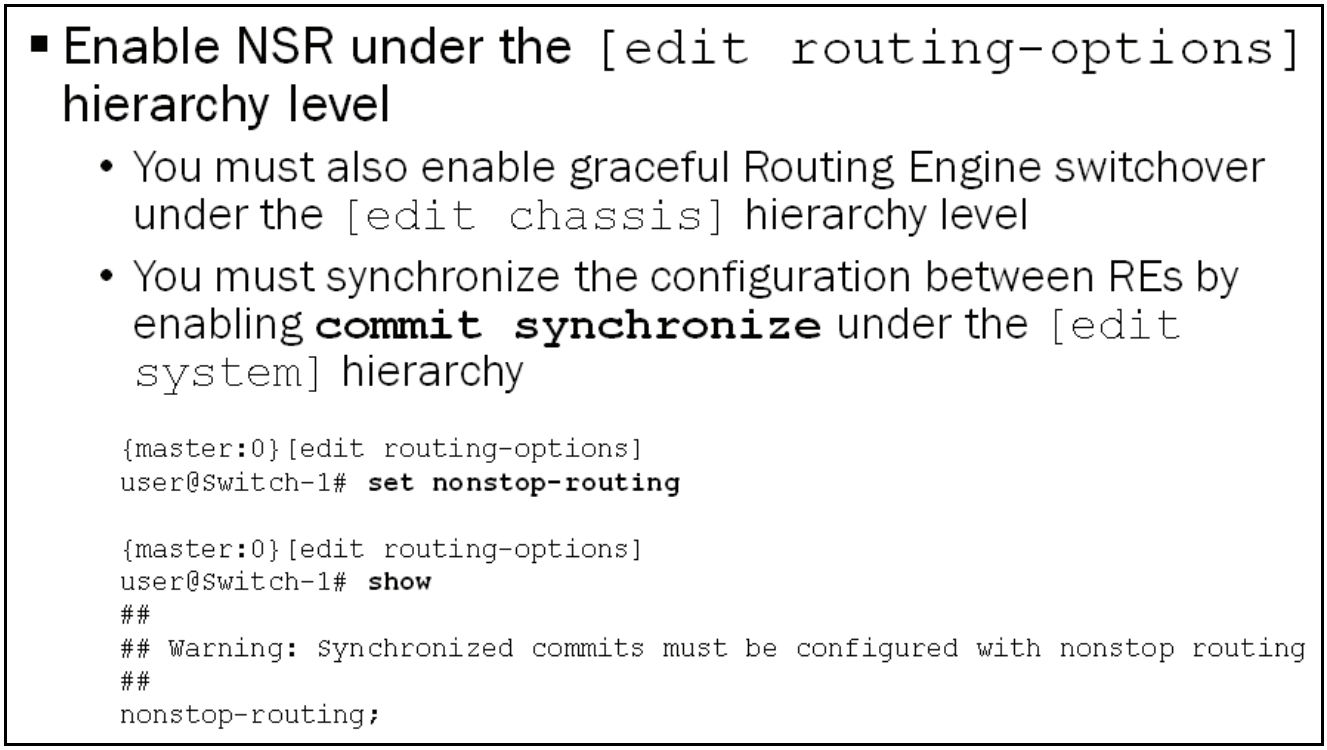
Nonstop Active Routing



Nonstop active routing (NSR) enables a routing platform with redundant REs, or participating in a Virtual Chassis, to switch from a primary RE to a backup RE without alerting peer devices. NSR uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. In addition to maintaining interface and kernel information, NSR also saves routing protocol information by running the `rpd` process on the backup RE. By saving this additional information, NSR is self-contained and does not rely on helper routers to assist the routing platform in restoring routing protocol information.

NSR requires that the participating REs run the same version of the Junos OS. Although NSR supports most protocols, it does not support all protocols. For all protocols supported by NSR, the state information is preserved during a switchover event. If you configure a protocol that is not supported by NSR, the protocol operates as usual. When a switchover occurs, the state information for the unsupported protocol is not preserved and must be refreshed using the normal recovery mechanisms inherent in the protocol. Also note that NSR is not supported on all EX Series switches. Refer to the technical publications for platform and protocol support details.

Configuring NSR



The graphic provides a sample configuration used to enable NSR. Note that you must also enable graceful Routing Engine switchover for NSR to function. We covered graceful Routing Engine switchover in detail in the previous section. In addition to enabling NSR and graceful Routing Engine switchover, you should also ensure that the commit operation synchronizes the configuration file by enabling the synchronize functionality in the configuration, as follows:

```
{master:0}[edit system]
user@Switch-1# set commit ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  synchronize          Synchronize commit on both Routing Engines by default
```

Once NSR is enabled and the configurations on the master and backup REs are synchronized, the routing protocol process on the backup RE actively gathers information sent to and from the routing protocol process on the master RE. This process allows the backup RE to keep its state up-to-date with the network just as the master RE does.

Monitoring NSR

- To verify the NSR synchronization status, use the **show task replication** command:

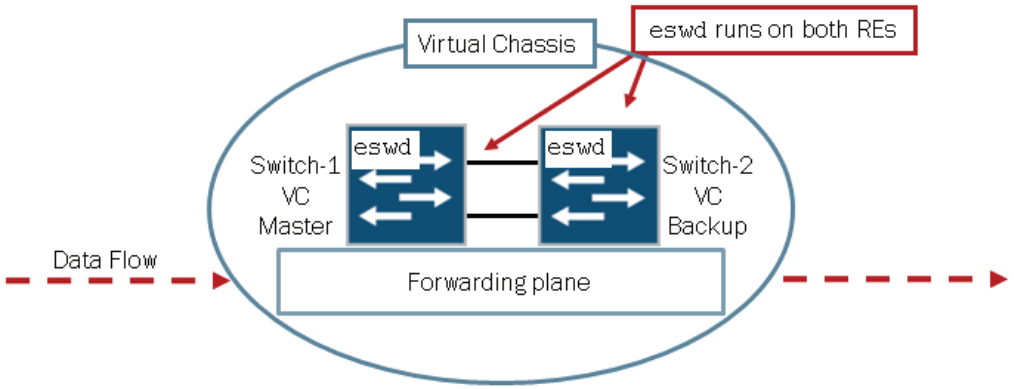
```
{master:0}
user@Switch-1> show task replication
        Stateful Replication: Enabled
        RE mode: Master

    Protocol                Synchronization Status
    OSPF                    Complete
    BGP                     Complete
```

  - You can also issue operational **show** commands such as **show ospf neighbor**, **show bgp summary**, and **show route** on the backup RE to verify state information

As shown on the graphic, you use the **show task replication** command to verify NSR synchronization. You should see all configured protocols that support NSR synchronization listed in the output, along with their complete status. Alternatively, you can log in to the backup RE and issue the same operational **show** commands you would issue on the master RE to determine protocol and routing information.

Nonstop Bridging

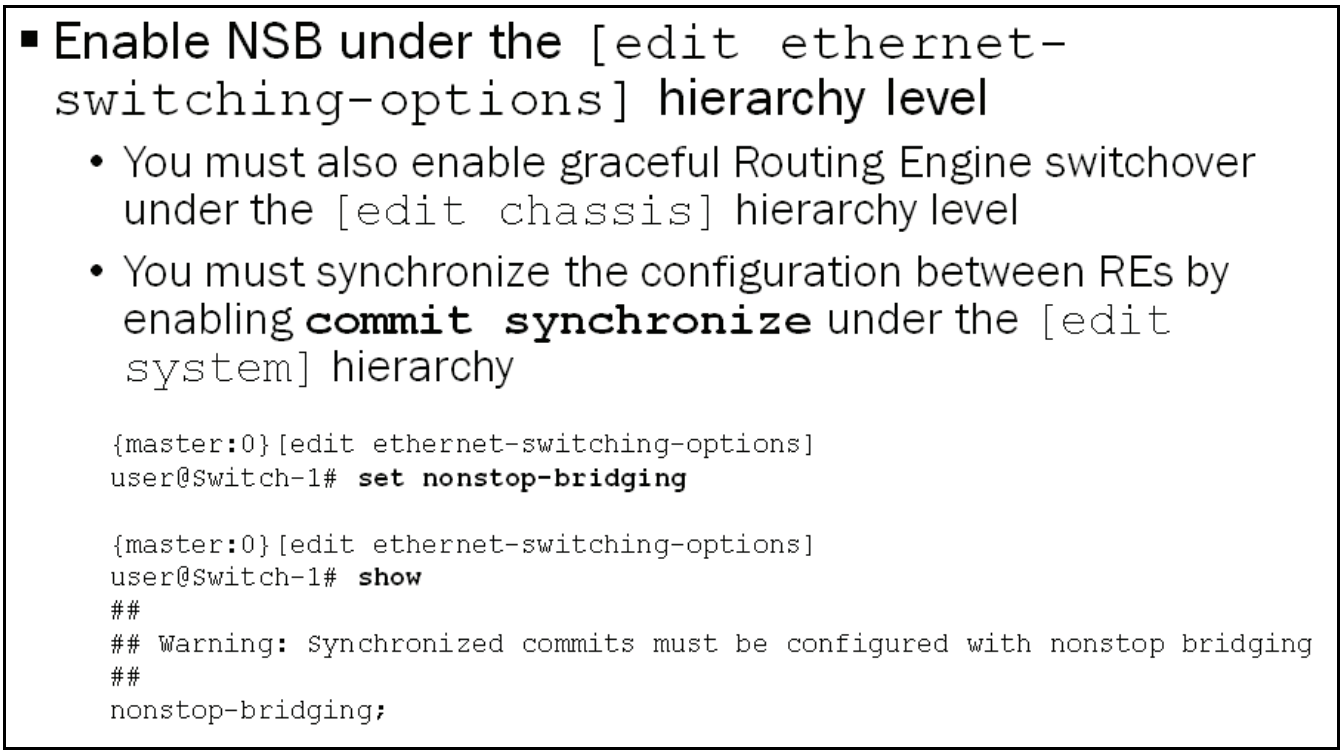
- NSB allows a mastership change to occur without alerting peer devices of that change
  - Uses graceful Routing Engine switchover infrastructure (graceful Routing Engine switchover must be enabled)
  - Preserves Layer 2 information and protocol sessions



Nonstop bridging (NSB) enables a switch with redundant REs, or participating in a Virtual Chassis, to switch from a primary RE to a backup RE without alerting peer devices. NSB uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. In addition to maintaining interface and kernel information, NSB also saves supported Layer 2 information by running the `eswd` process on the backup RE. NSB does the same for Layer 2 protocols that NSR does for Layer 3 routing protocols.

NSB requires that the participating REs run the same version of the Junos OS. Although NSB supports many Layer 2 protocols, it does not support all protocols. For all supported Layer 2 protocols, the state information is preserved during a switchover event. If you configure a protocol that is not supported by NSB, the protocol operates as usual. When a switchover occurs, the state information for the unsupported protocol is not preserved and must be refreshed using the normal recovery mechanisms inherent to the protocol. Also note that NSB is not supported on all EX Series switches. Refer to the technical publications for platform and protocol support details.

Configuring NSB



The graphic provides a sample configuration used to enable NSB. Note that you must also enable graceful Routing Engine switchover for NSB to function. We covered graceful Routing Engine switchover in detail in the previous section of this chapter. In addition to enabling NSB and graceful Routing Engine switchover, you should also ensure that the commit operation synchronizes the configuration file by enabling the synchronize functionality in the configuration. Once NSB is enabled and the configurations on the master and backup REs are synchronized, the supported Layer 2 protocol process on the backup RE actively gathers information sent to and from the Ethernet switching protocol process on the master RE. This process allows the backup RE to keep its state up-to-date with the network just as the master RE does. No current method exists to verify that NSB is synchronizing Layer 2 protocol information on the backup RE. You can log in to the backup RE and verify that the Ethernet switching subsystem is running. You will not see synchronized values though. The following output is from the backup RE with NSB running:

```
{backup:1}
user@Switch-1> show spanning-tree bridge
STP bridge parameters
Snipped...
```

If NSB is not enabled the Ethernet subsystem will not be running as illustrated in the following output:

```
{backup:1}
user@Switch-1> show spanning-tree bridge
error: the ethernet-switching subsystem is not running
```


Review Questions

1. What is the purpose of LACP?
2. In what situation would you likely find RTG?
3. Which high availability feature allows you synchronize Layer 2 protocol information between REs?

Answers

1.

LACP is used for automatic addition and deletion of individual links to an aggregated bundle and for link monitoring to check whether both ends of the aggregated bundle are connected to the correct group. The Junos OS implementation of LACP provides link monitoring but not automatic addition and deletion of links.

2.

Redundant trunk groups provide a quick and simple failover mechanism between redundant Layer 2 links. This feature is a replacement for STP and is often used on access switches that are connected to two aggregation switches.

3.

For the redundant REs to synchronize Layer 2 protocol information you must enable nonstop bridging.

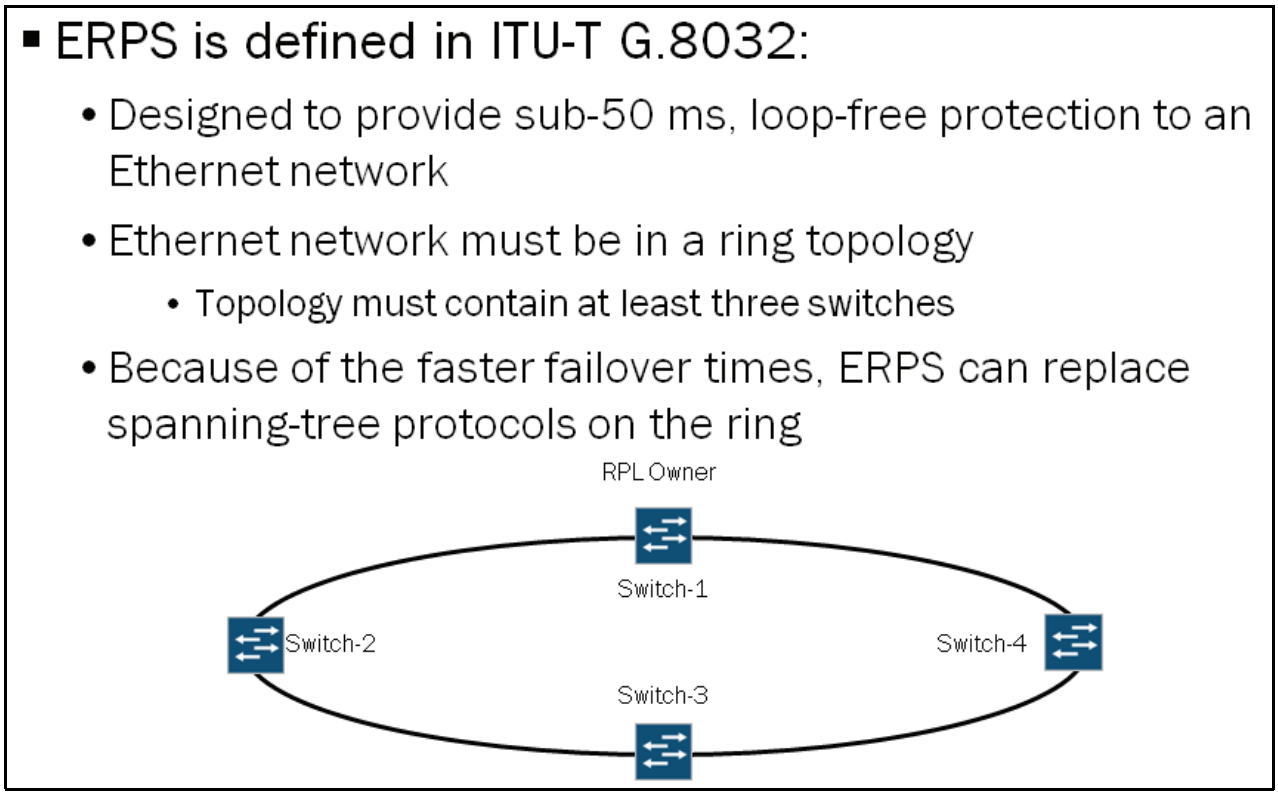
JNCIS-ENT Switching Study Guide

Appendix A: Ethernet Ring Protection Switching

This Appendix Discusses:

- The concepts of Ethernet Ring Protection Switching (ERPS); and
- Configuration and monitoring of ERPS.

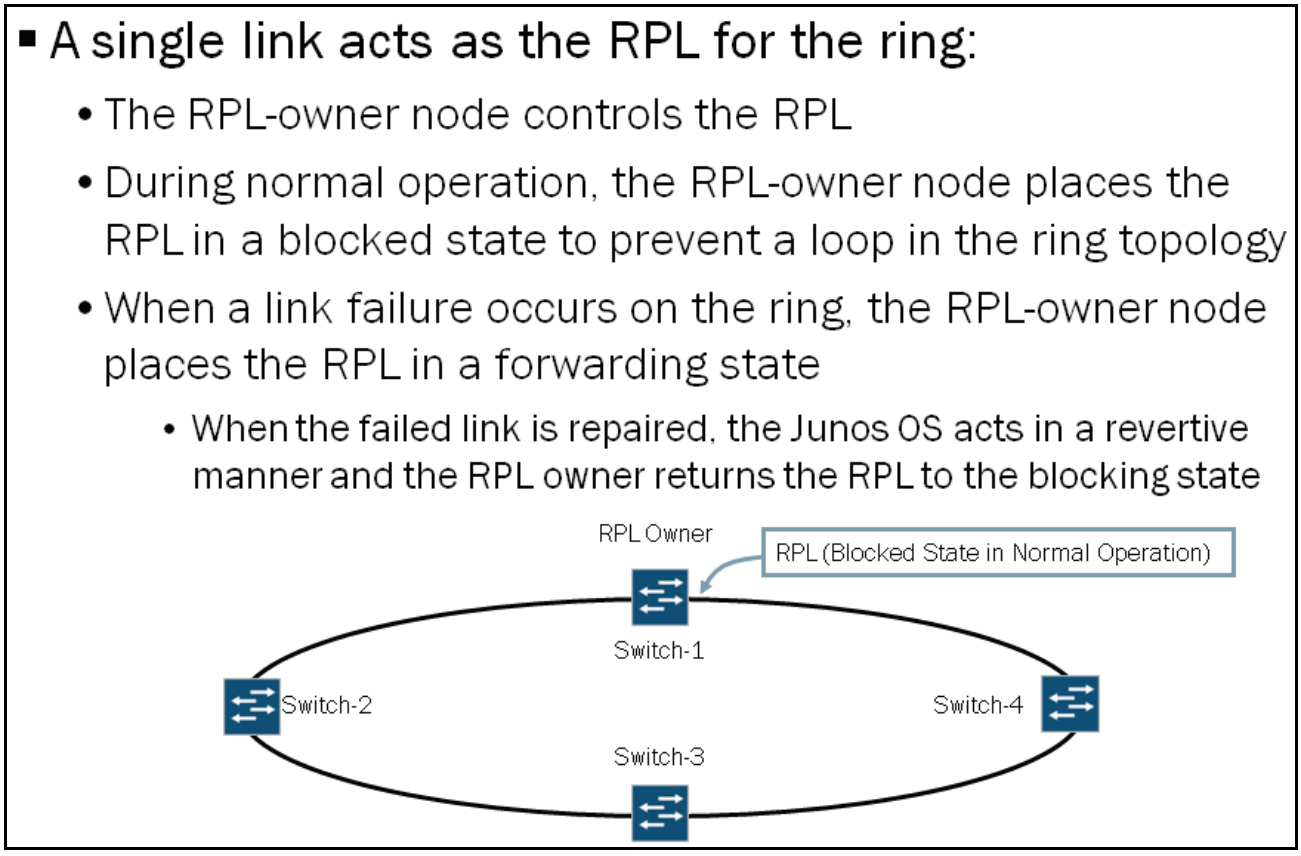
ERPS



Defined in the International Telecommunication Union Telecommunication Standardization (ITU-T) G.8032 recommendation, ERPS provides highly reliable, stable, and loop-free protection for Ethernet ring topologies. ERPS 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 three. Because ERPS can provide sub-50 ms, loop-free protection for a ring topology, it can viably replace any spanning-tree protocol on the ring. For optimal performance, we recommend you do not have more that 16 nodes in a single ERPS ring.

ERPS is not supported on all EX Series Ethernet Switches and you should refer to technical documentation to identify supportability on your platform and Junos version. For a support matrix of features on EX Series switches, refer to the following URL:http://www.juniper.net/techpubs/en_US/release-independent/junos/topics/concept/ex-series-software-features-overview.html.

Ring Protection Link

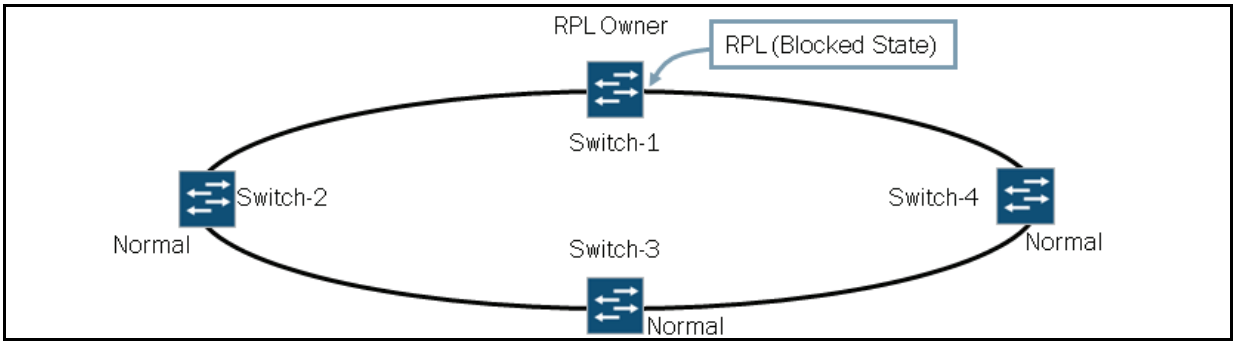


The basic idea of Ethernet ring protection is to use one specific link to protect the whole ring. This special link is the ring protection link (RPL). When all links are up and running, the RPL blocks traffic and remains idle. The RPL itself is controlled by the designated RPL owner. Only one RPL owner exists on the ring and the RPL owner is responsible for blocking the RPL interface under normal operating conditions. However, if a link failure occurs on the ring, the RPL owner is responsible for unblocking the RPL interface and begins forwarding the traffic on the alternate path around the ring. 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

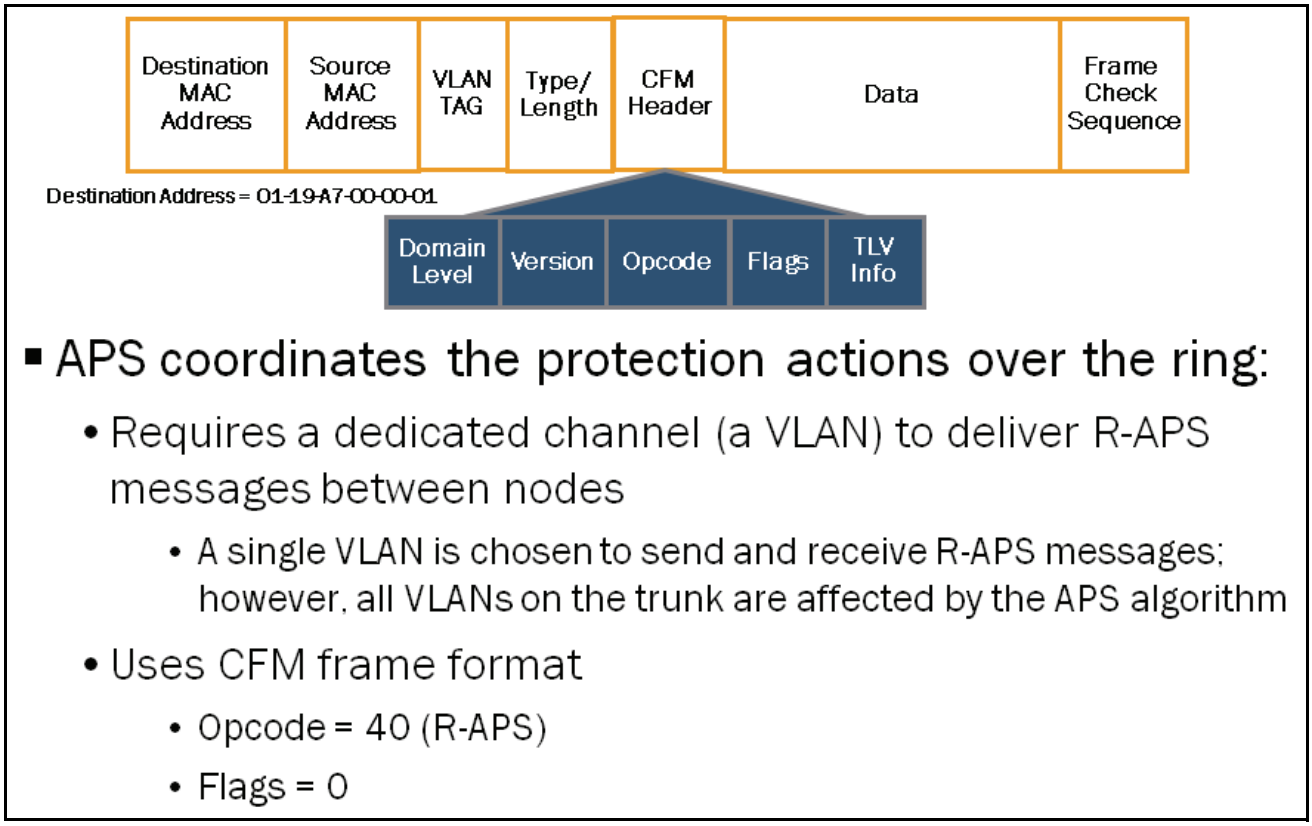
The RPL owner controls the state of the RPL. During the idle state, it is the only node that sends periodic Ring Automatic Protection Switching (R-APS) messages to notify the other nodes about the state of the RPL. The next few graphics discuss the details of the Automatic Protection Switching (APS)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

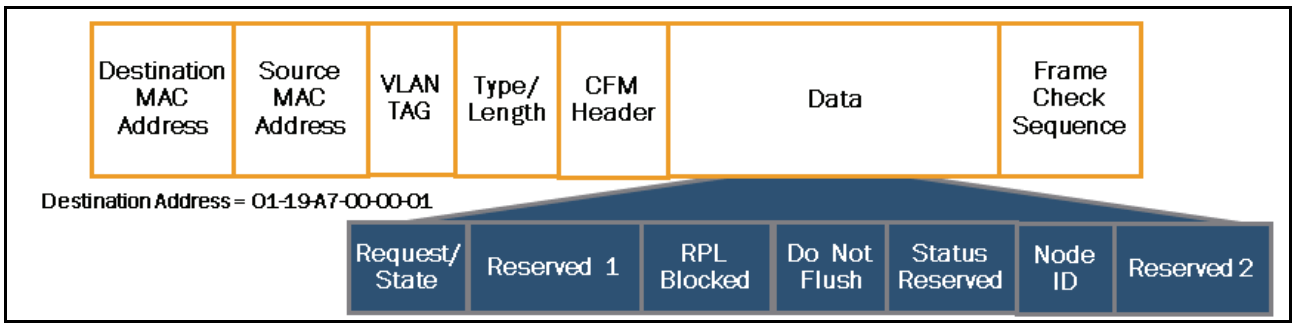
APS Protocol



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 using a virtual LAN (VLAN) 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 connectivity fault management (CFM) frame format. To allow differentiation between an R-APS message from other CFM message, an R-APS message uses a destination address of 01-19-A7-00-00-01, as well as an opcode of 40.

For additional information relating to CFM please consult the technical documentation for your platform and Junos version.

R-APS Data Fields

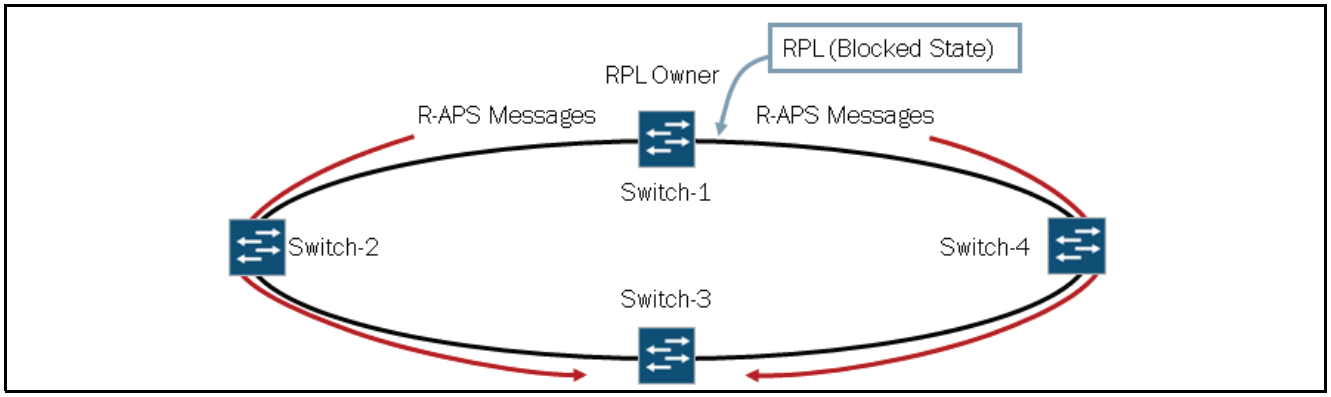


Currently, APS has no specified type, length, and values (TLVs). The graphic shows the data fields found in an R-APS message. The following list describes each data field:

- *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 graphic. Only the RPL owner can signal RPL Blocked.

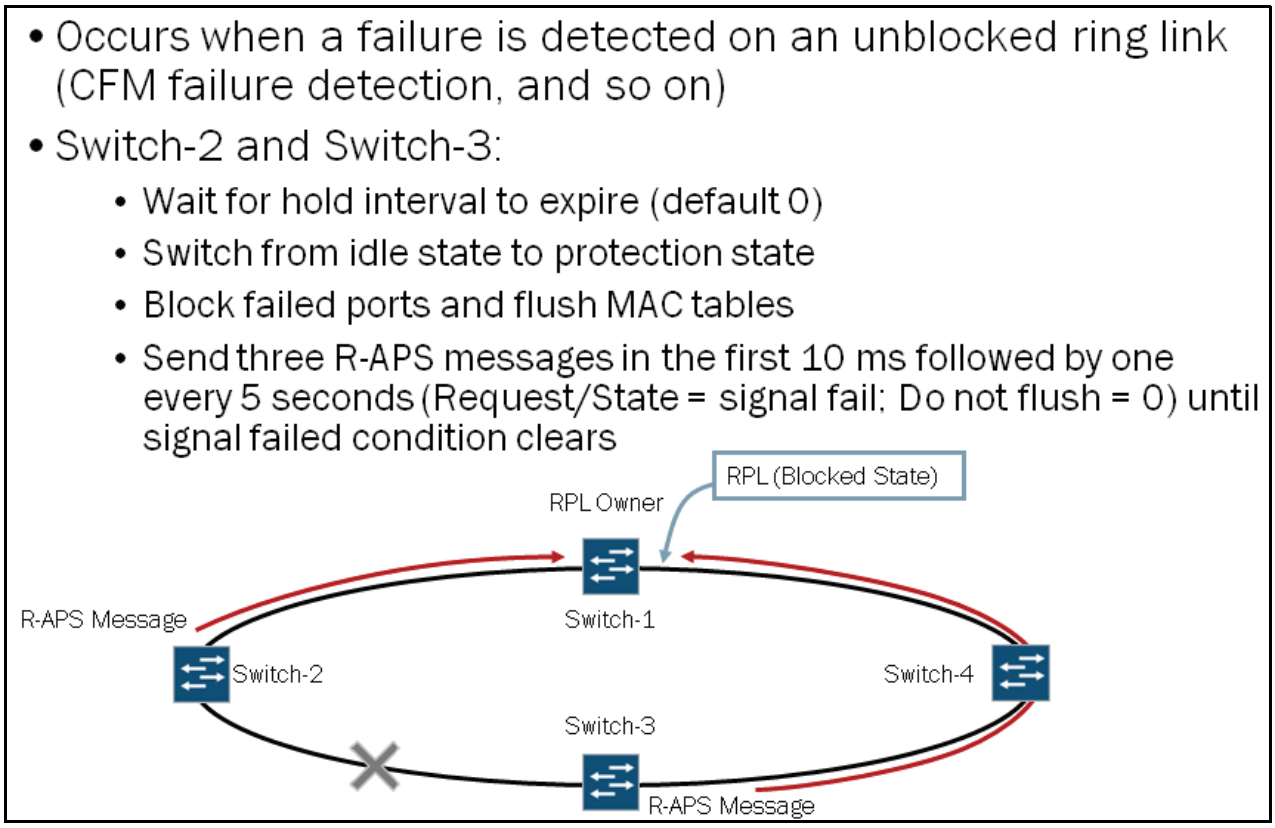
- *Do not flush* (1 bit): The value of 1 indicates that devices should not flush their MAC tables while a value of 0 indicates that devices should flush their MAC tables.
- *Status Reserved* (6 bits): This value is always 000000. This field is reserved for future use.
- *Node ID* (6 octets): This field 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



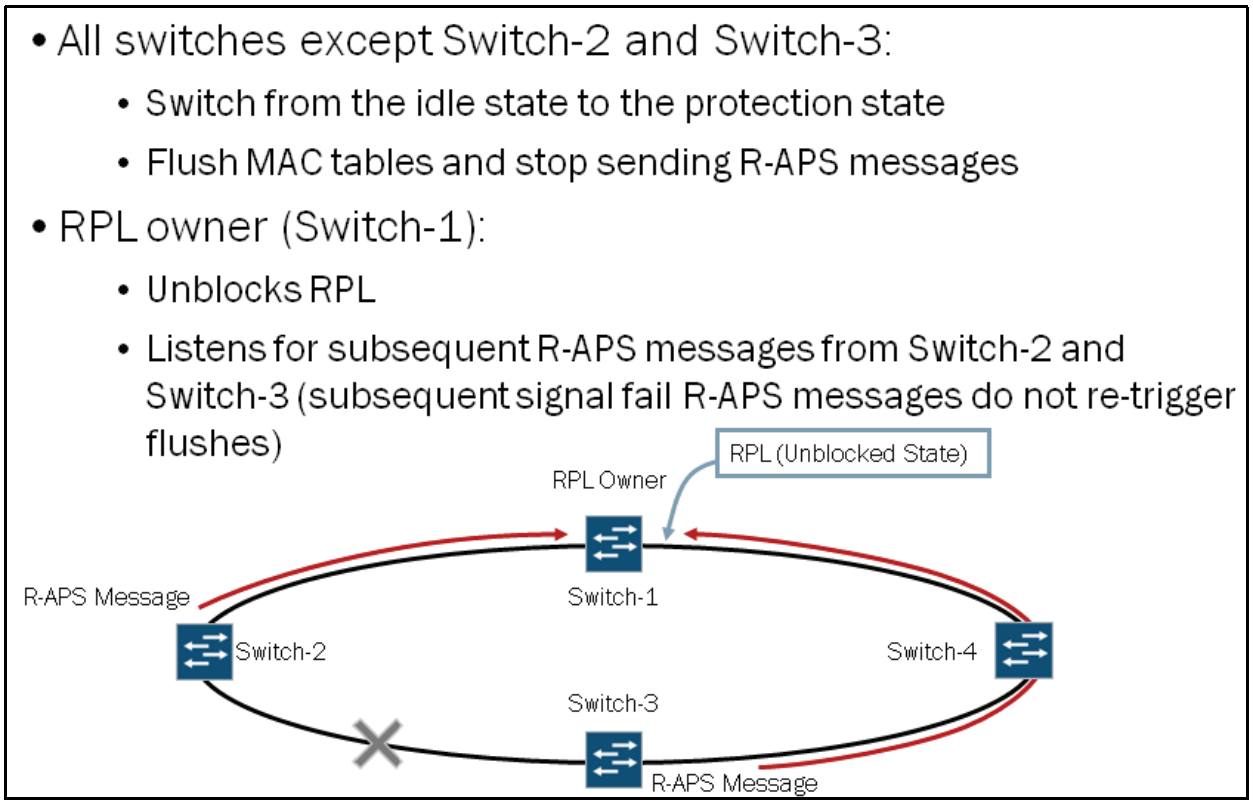
When no failures occur on the Ethernet ring, all nodes are in the idle state. During the idle state, the RPL owner places the RPL in a blocking state. Also, the RPL owner sends periodic (every 5 seconds) R-APS messages that signal that no failure is present on the ring (Request/State =no request), that 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



A signal failure occurs when a node detects a failure on a ring port. In the example, Switch-2 and Switch-3 detect a failure on the link between them. The Junos OS does not currently support hold interval. In other words, Switch-2 and Switch-3 react immediately to the failed link. The nodes switch from the idle state to the protection state, block the failed ports, flush their MAC

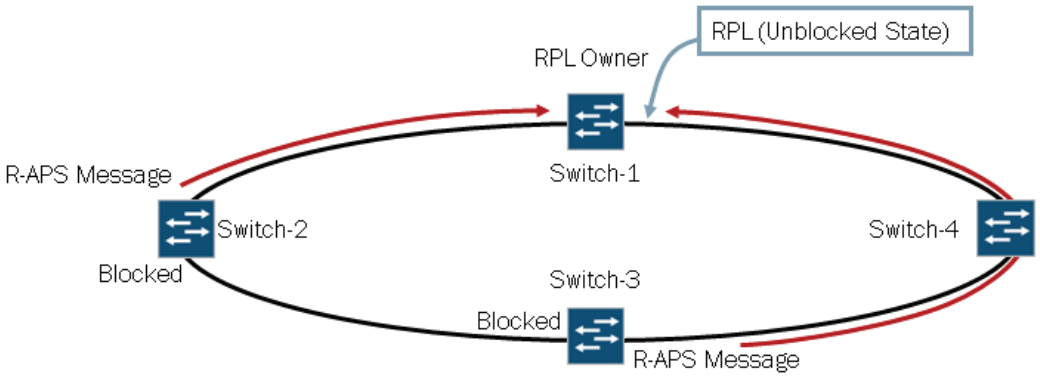
table, and signal to all the other nodes that a signal failure has occurred using R-APS messages. The R-APS messages tell the other nodes that a failure has occurred (Request/State = signal fail) and that the nodes should flush their MAC tables (Do not flush= 0). Switch-2 and Switch-3 continually send R-APS messages every 5 seconds until the signal failure condition clears.



Upon receiving the signal fail R-APS messages from Switch-2 and Switch-3, all other nodes (including the RPL owner) transition to the protection state, and flush their MAC tables. The RPL owner stops sending R-APS messages, unblocks the RPL, and listens for subsequent R-APS message from Switch-2 and Switch-3.

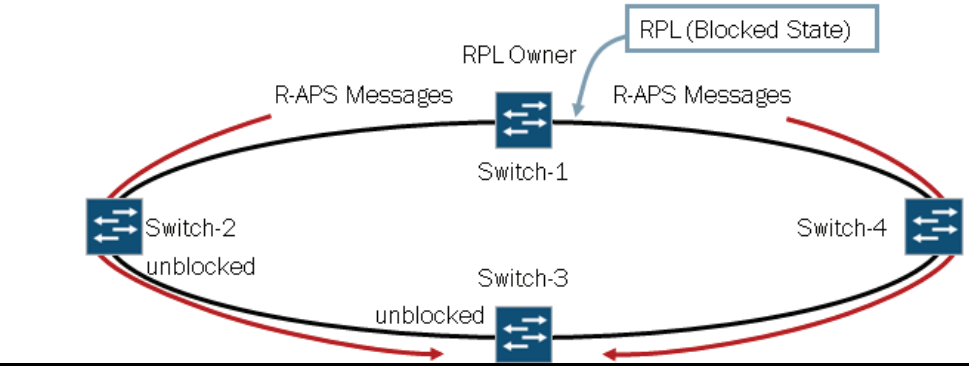
Restoration of a Failed Link

- Signal fail condition clears on the link between Switch-2 and Switch-3
  - Switch-2 and Switch-3:
    - Continue to block the previously failed link
    - Send no request R-APS messages (Request/State = no request; Do not flush = 1) and continue until they receive an R-APS message from Switch-1



When the failure is repaired between Switch-2 and Switch-3, they begin sending 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). Switch-2 and Switch-3 keep the previously failed ports in the blocked state (preventing a loop) until they receive R-APS messages from Switch-1 as described in the following graphic.

- Switch-1 (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 Switch-2 and Switch-3, Switch-1 starts a restore timer. The default is 5 minutes. You can configure the restore timer in 1-minute increments between 5 and 12 minutes. Once the restore timer expires, Switch-1 blocks the RPL and transmits R-APS 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 Switch-1, 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.

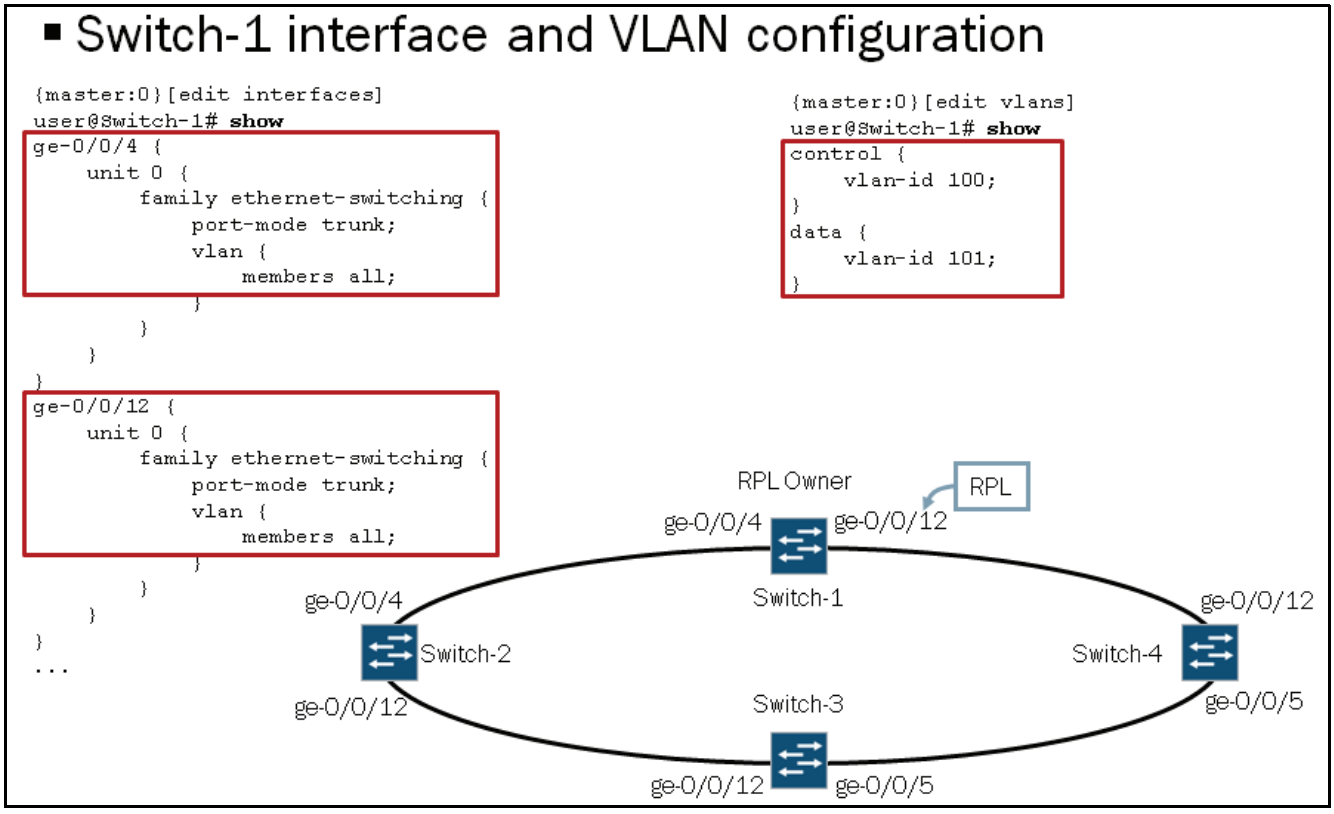
ERPS Configuration Options

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

- `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.
- `restore-interval`: Specifies the number of minutes that the node waits before processing ERP PDUs.

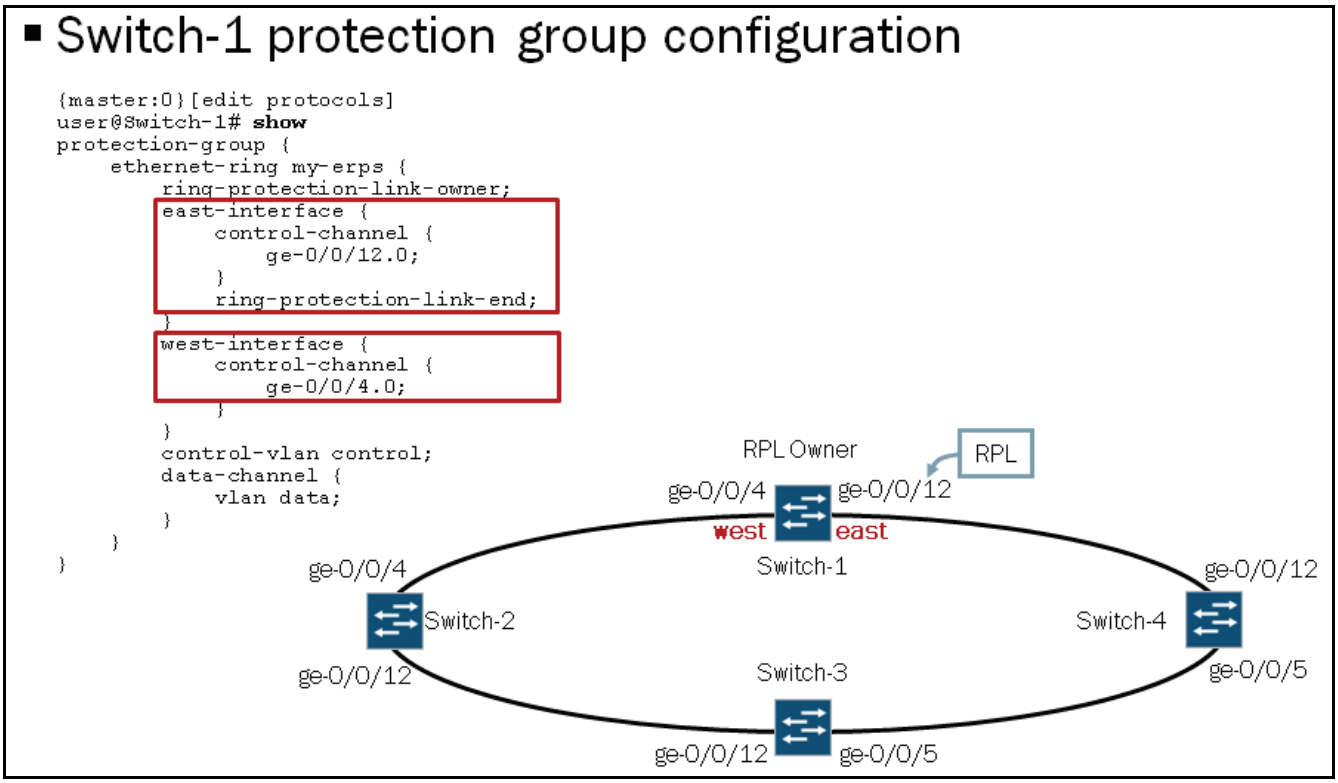
```
protection-group {
    ethernet-ring my-erps {
        ring-protection-link-owner;
        restore-interval interval;
        guard-interval interval;
        east-interface {
            control-channel {
                interface-name;
            }
            ring-protection-link-end;
        }
        west-interface {
            control-channel {
                interface-name;
            }
        }
        control-vlan (vlan-id | vlan-name);
        data-channel {
            vlan (vlan-id | vlan-name);
        }
    }
}
```

RPL Owner Configuration



The graphic shows a typical interface configuration for the RPL owner. First, you must configure the two interfaces that participate in the Ethernet ring for the APS channel. For trunk-mode interfaces, you must also specify the VLANs allowed across the interface. In this case, VLAN 100 is used as the communication channel between nodes and VLAN 101 is used to specify data traffic.

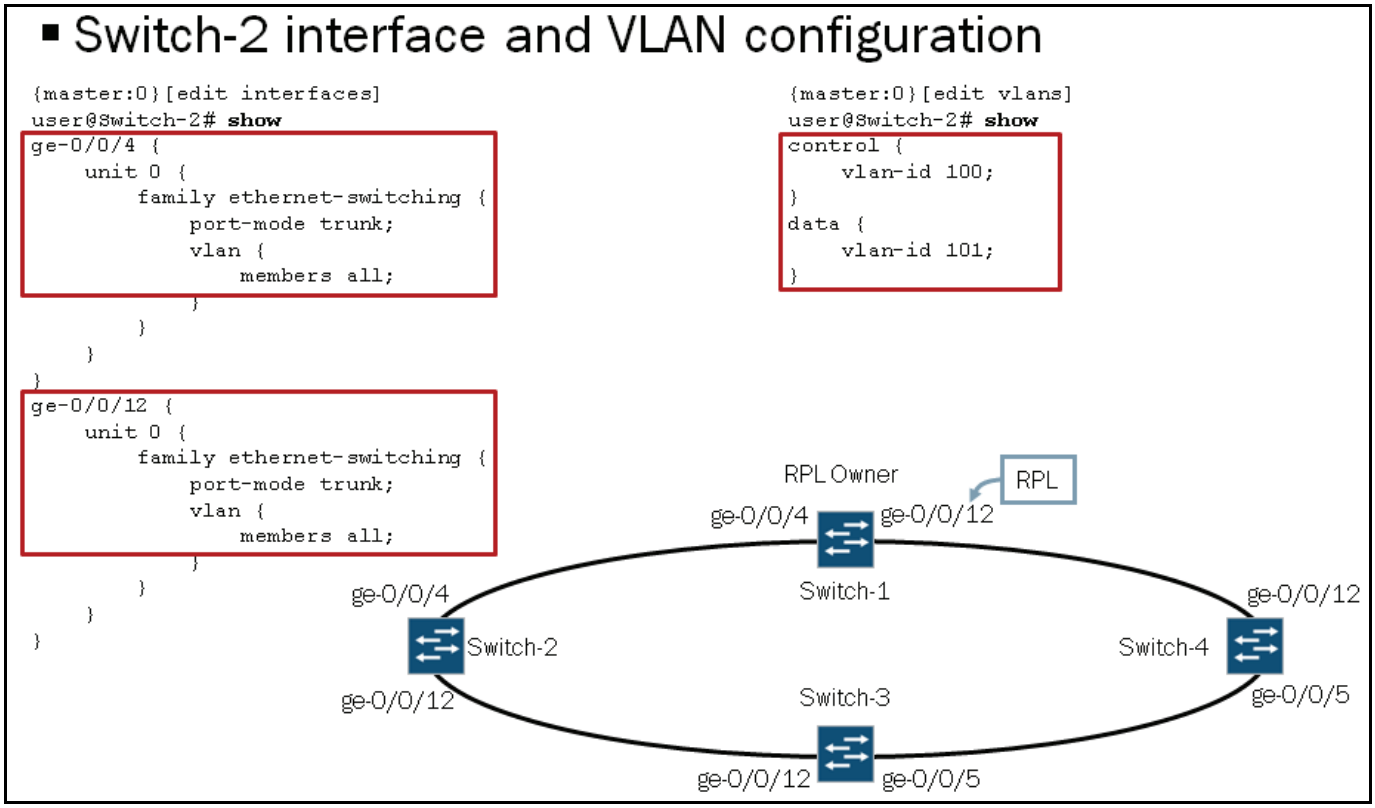
In the example provided on the graphic, you are using just standard Gigabit Ethernet interfaces, but depending on your environment you might want to implement LAG interfaces to ensure that your trunk interfaces can support the volume of data traffic.



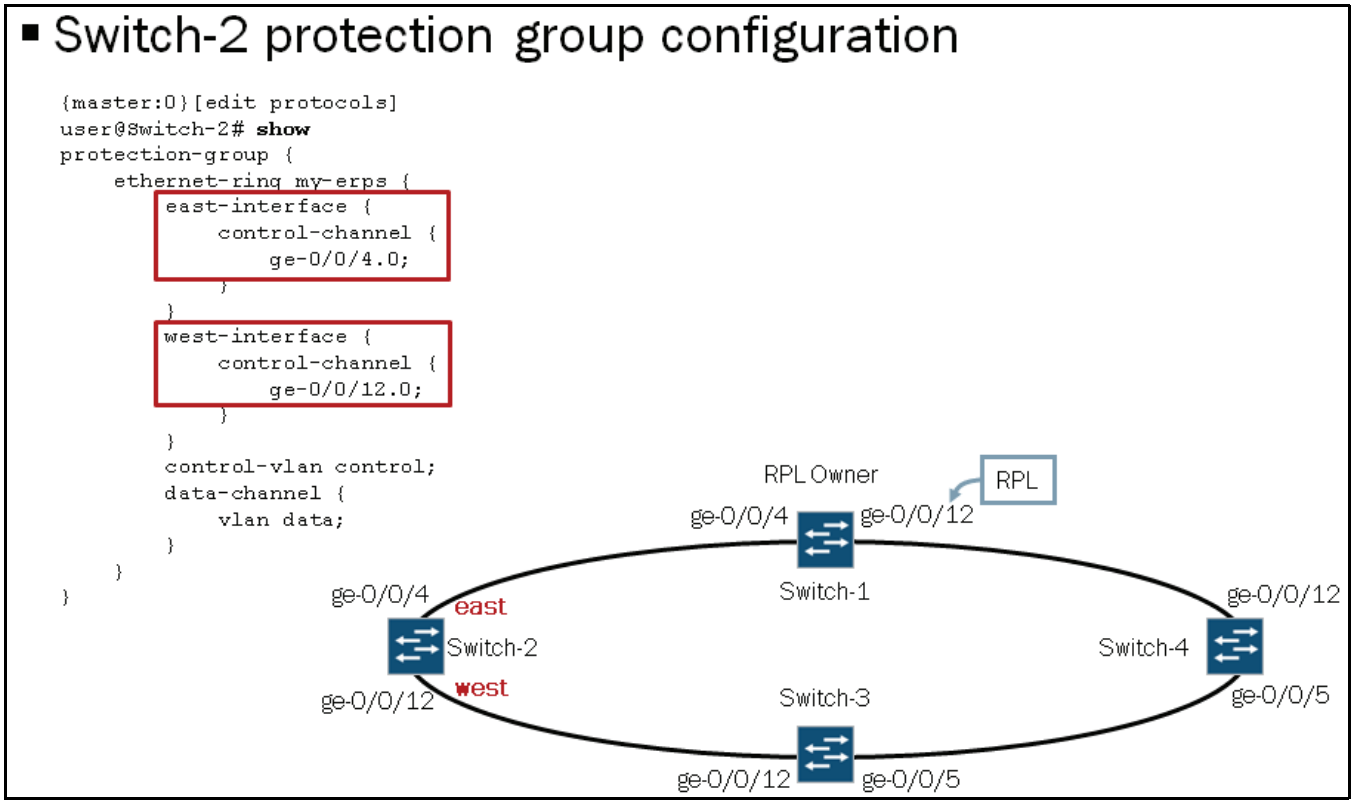
The graphic shows a typical ERPS configuration for the RPL owner. Configure ERPS parameters under `[edit protocols protection-group]`. Note the following information about the ERPS configuration for the RPL owner:

- You must configure the RPL owner specifically as the `ring-protection-link-owner`; and
- The interfaces are interchangeable with regard to selecting them to act as the `west-interface` and `east-interface` as long as you specify one as the `ring-protection-link-end`.

Normal Node Configuration



The graphic shows a typical interface configuration for a normal node. The interface configurations are basically the same as the interfaces configured on the RPL owner.



The graphic shows a typical ERPS configuration for a normal node. The configuration is very similar to what is configured for the RPL owner except you do not identify this node as the RPL owner and you do not configure one of the interfaces as the ring end link.

ERPS Status

```
user@Switch-1> 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
```

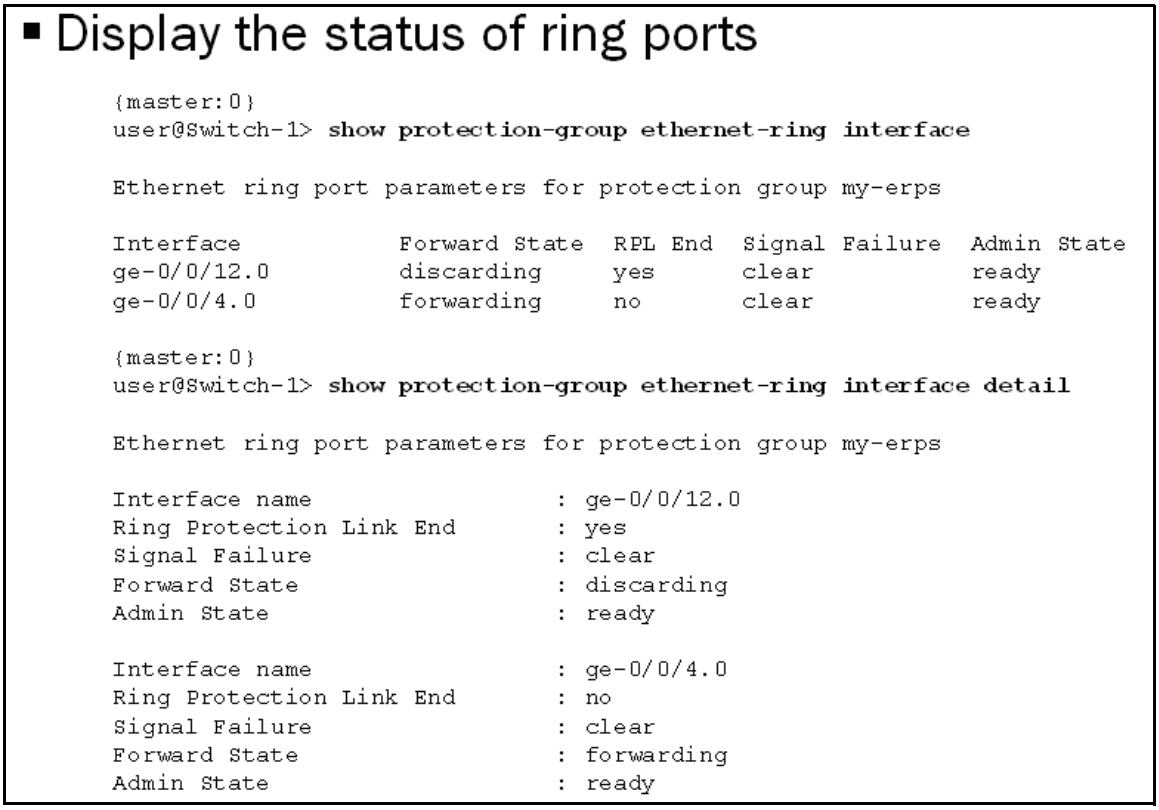
The graphic shows all of the possible commands to monitor ERPS. We discuss each one on the next few graphics.

R-APS Information

```
{master:0}
user@Switch-1> show protection-group ethernet-ring aps

Ring Name    Request/state  No Flush  RPL Blocked  Originator  Remote Node ID
my-erps      NR             no        yes          yes         NA

{master:0}
user@Switch-1> show protection-group ethernet-ring aps detail
Ethernet-Ring name             : my-erps
Request/State                  : NR
No Flush Flag                  : no
Ring Protection Link blocked   : yes
Originator                     : yes
Remote Node ID                 : NA
```

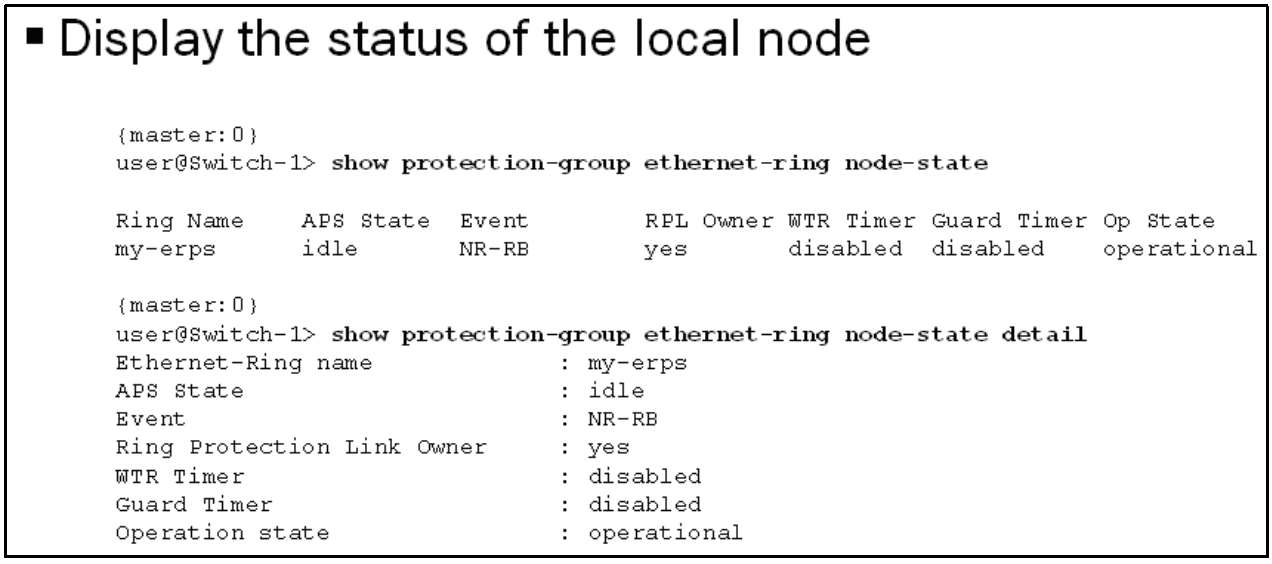
The command on the graphic shows the details of the R-APS messages to which the local node is currently listening or which it is forwarding. Based on the output, you can tell that the local node (Switch-1) is the RPL owner because the R-APS message originates from it and it is advertising that the RPL is currently blocked.

Interface Status



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

Local Node Details



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

ERPS Statistics

- Display statistics of the local node

```
{master:0}
user@Switch-1> show protection-group ethernet-ring statistics

Ring Name  Local SF  Remote SF NR Event NR-RB Event
my-erps    2         4         4        3

{master:0}
user@Switch-1> show protection-group ethernet-ring statistics detail

Ethernet ring statistics for protection group my-erps

PG level statistics:

Local SF      : 2
Remote SF     : 4
NR Event      : 4
NR-RB Event   : 3

Interface level statistics:

Interface   RAPS sent   RAPS received
ge-0/0/12.0        782              0
ge-0/0/4.0         784              0
```

The command on 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.

Review Questions

1. How does ERPS prevent forwarding loops?
2. What messages are exchanged between ERPS devices to coordinate protection actions?
3. Which configuration parameters must be used to identify a device as the RPL owner?

Answers

1.

ERPS prevents forwarding loops by placing one interface in a blocked state. This interface is known as the RPL and is only present on the RPL owner device. This interface remains in a blocked state unless there is a failure within the ring that prevents traffic from being forwarded to the final destination.

2.

ERPS uses R-APS messages to coordinate protection actions.

3.

You must configure the RPL owner specifically as the **`ring-protection-link-owner`**. You must also specify either the `east-interface` or `west-interface` as the **`ring-protection-link-end`**.

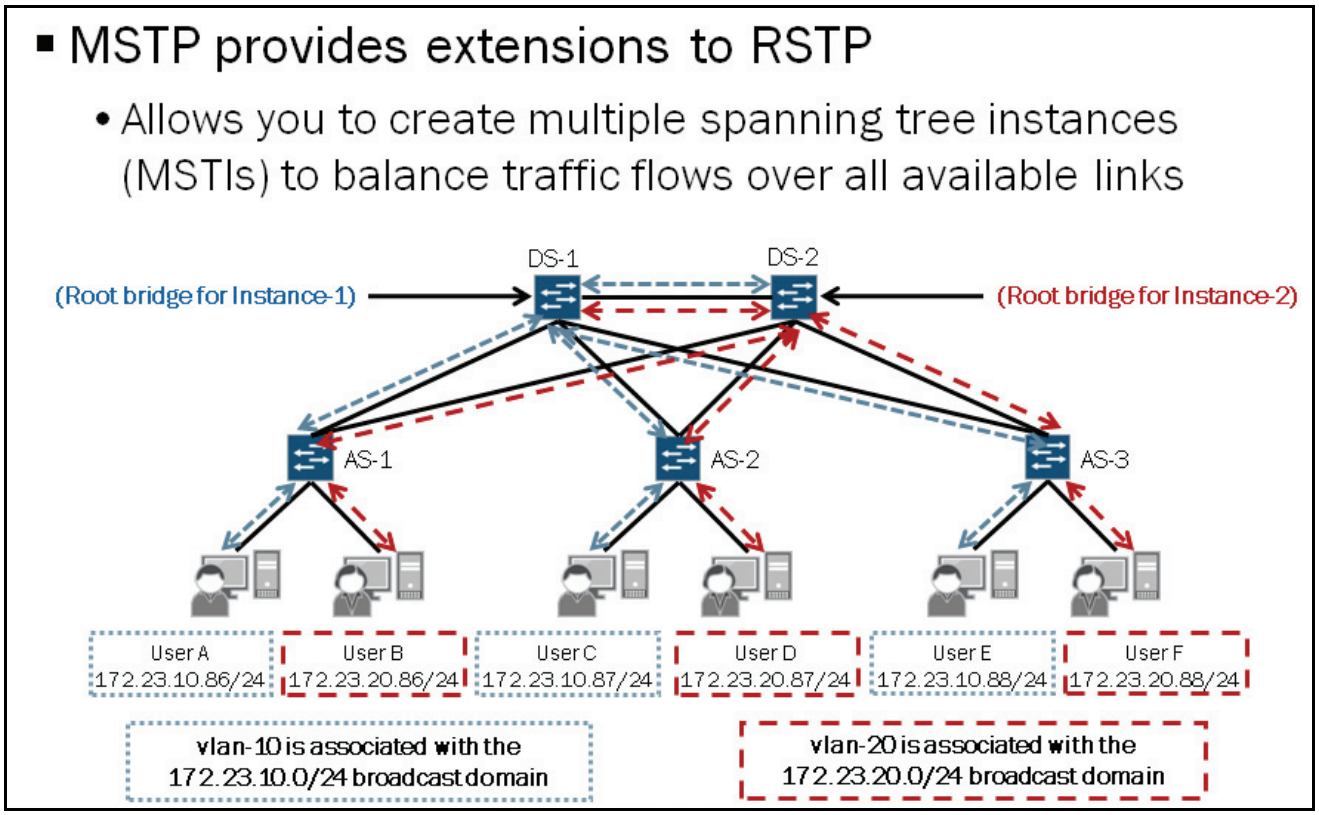
JNCIS-ENT Switching Study Guide

Appendix B: Multiple Spanning Tree Protocol

This Appendix Discusses:

- The concepts of Multiple Spanning Tree (MSTP); and
- Configuration and monitoring of MSTP.

MSTP

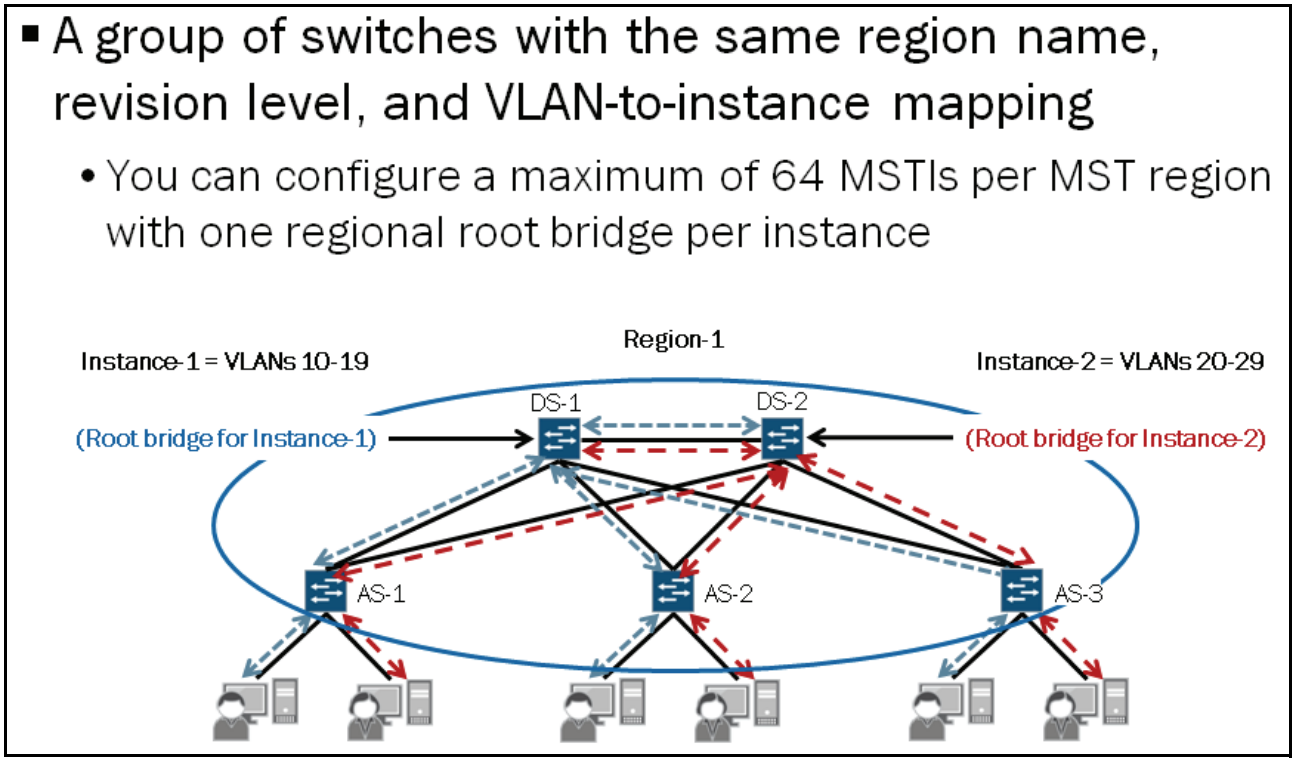


MSTP extends Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP) functionality by mapping multiple independent spanning-tree instances onto one physical topology. Each spanning-tree instance (STI) includes one or more VLANs. Each multiple spanning tree instance (MSTI) creates a separate 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.

Unlike in STP and RSTP configurations, a port can belong to multiple VLANs and be dynamically blocked in one spanning-tree instance but forwarding in another. This behavior significantly improves network resource utilization by load-balancing across the network and maintaining switch CPU loads at moderate levels. MSTP also leverages the fast re-convergence time of RSTP when a network, switch, or port failure occurs within a spanning-tree instance.

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

MST Region

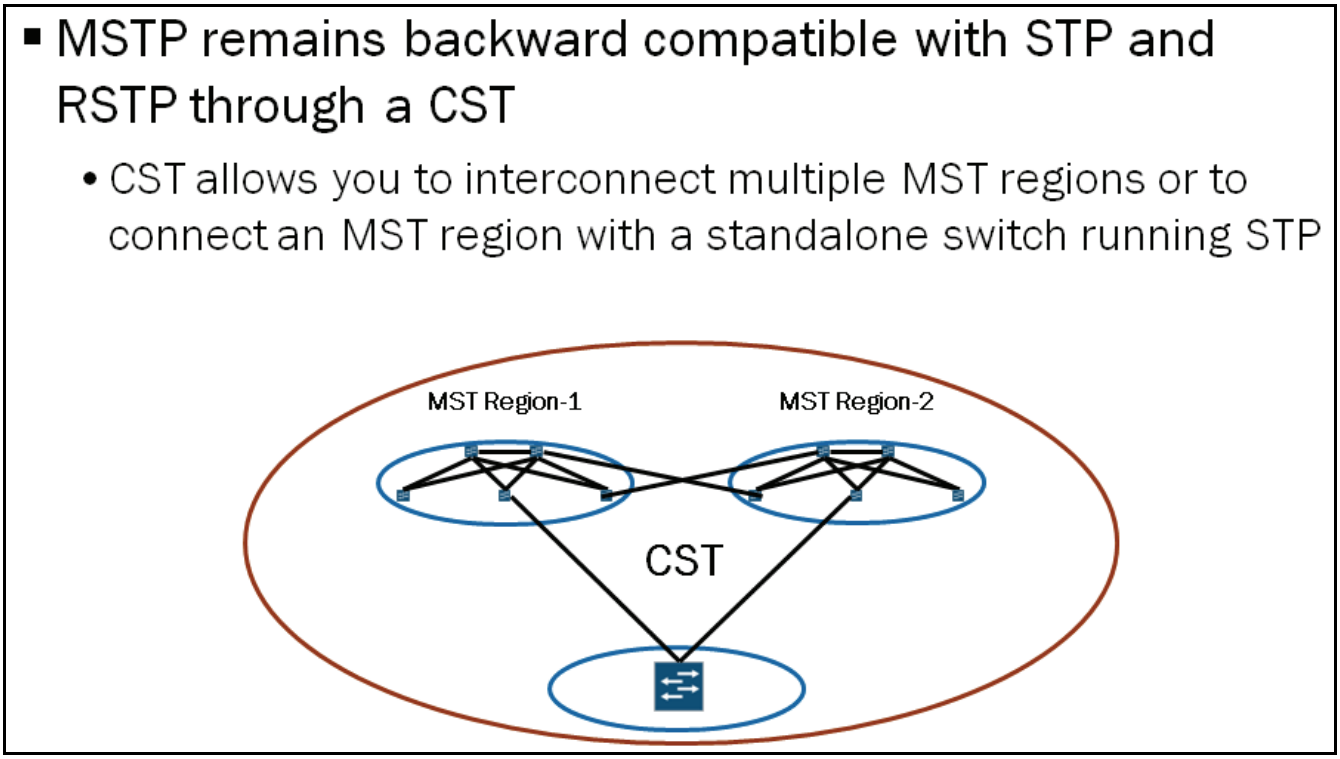


MSTP allows switches to be logically grouped into manageable clusters, known as multiple spanning tree (MST) regions. An MST region is a group of switches that share the same region name, revision level, and VLAN-to-instance mapping parameters.

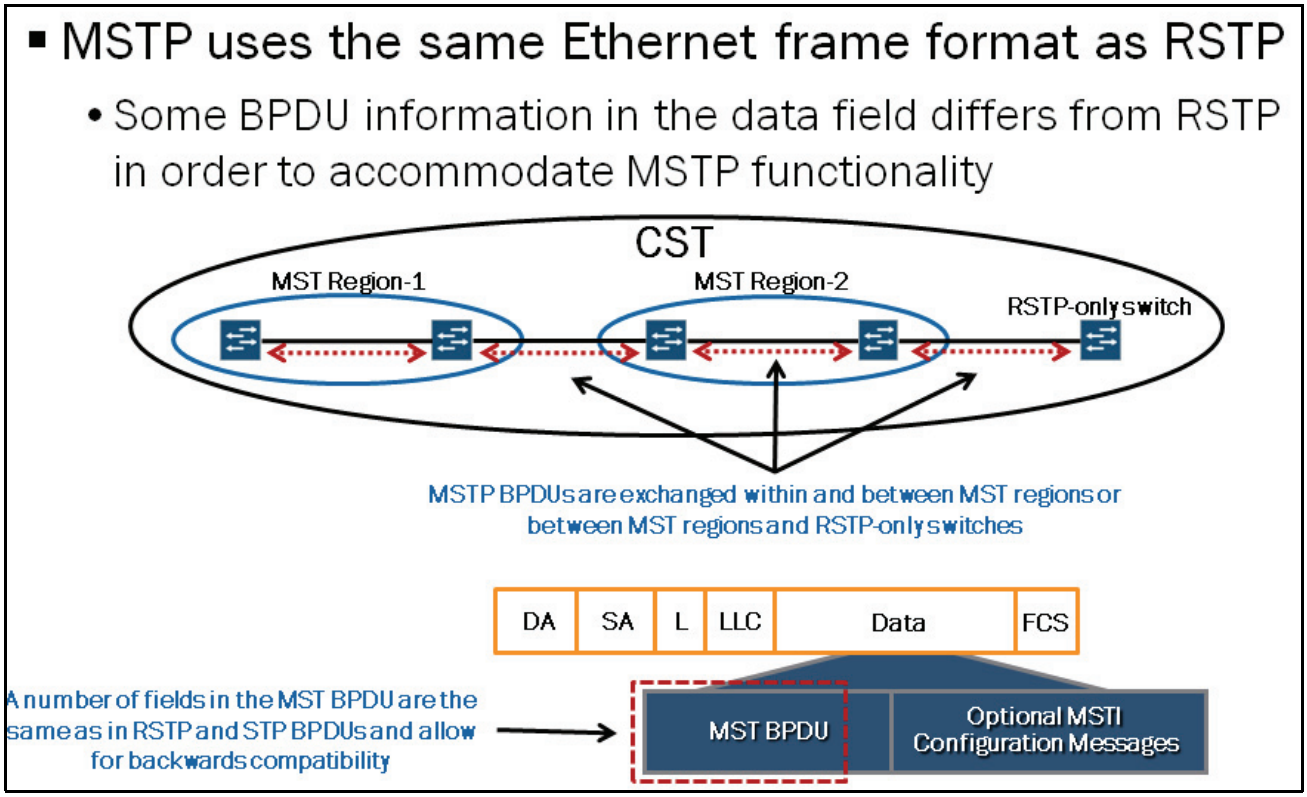
Each MST region supports up to 64 MSTIs. MSTP greatly reduces the number of bridge protocol data units (BPDUs) on a LAN by including the spanning tree information for all MSTIs in a single BPDU. MSTP encodes region information after the standard RSTP BPDU along with individual MSTI messages. The MSTI 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.

Common Spanning Tree



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. The CST facilitates backward compatibility with RSTP and STP.

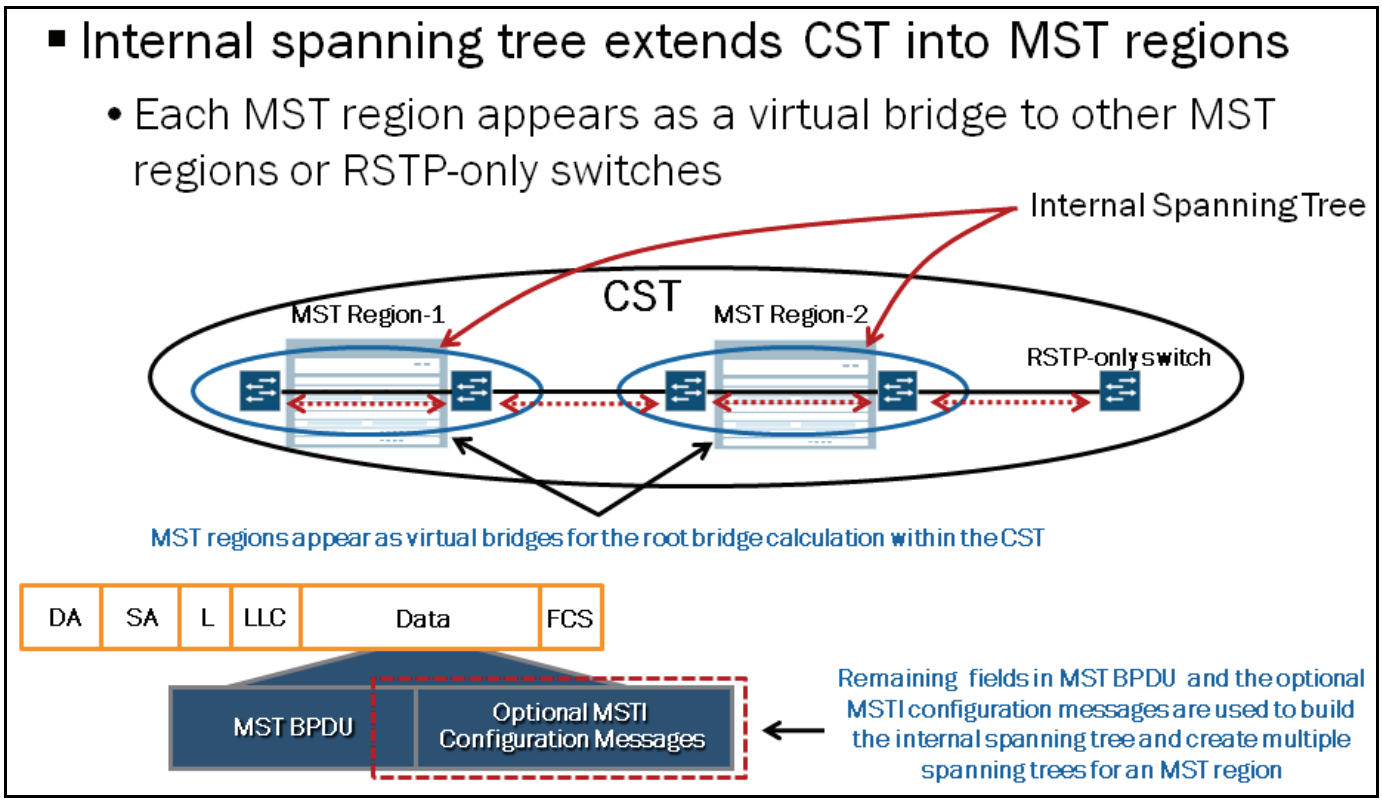


Because MSTP encodes region information after the standard RSTP BPDU, a switch running RSTP interprets MSTP BPDUs as RSTP BPDUs. This behavior facilitates full compatibility between devices running MSTP and devices running STP or RSTP. MSTP uses the same Ethernet frame as STP and RSTP. However, the BPDU information in the data field is different.

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 RSTP BPDU. To the outside world (other

MSTI regions or standalone RSTP devices), these fields are a representation of the virtual bridge that is an individual MSTP region. This information is used to build the CST.

Common and Internal Spanning Tree



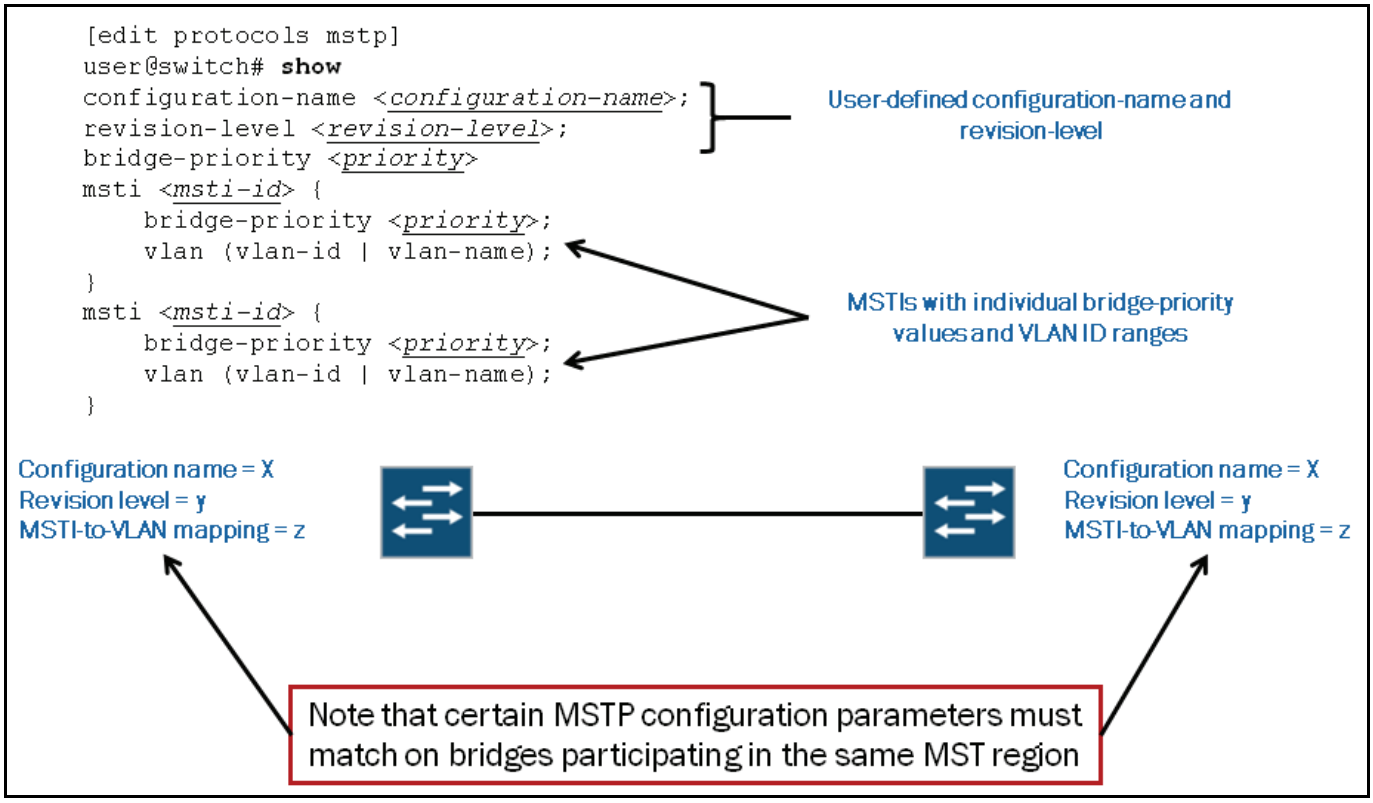
All MSTP environments contain a CST, which is used to interconnect individual MST regions and independent STP devices. All bridges in the CST elect a single root bridge. The root bridge is responsible for the path calculation for the CST. As illustrated on the graphic, bridges outside of the MST region 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.

Each MSTP region builds a spanning tree for the region, referred to as an internal spanning tree, based upon the remaining BPDU fields. 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 (illustrated on the previous graphic). Once the internal spanning tree is built, by default, all traffic on all VLANs will follow it.

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. We evaluate MSTP configuration on EX Series switches on a subsequent graphics.

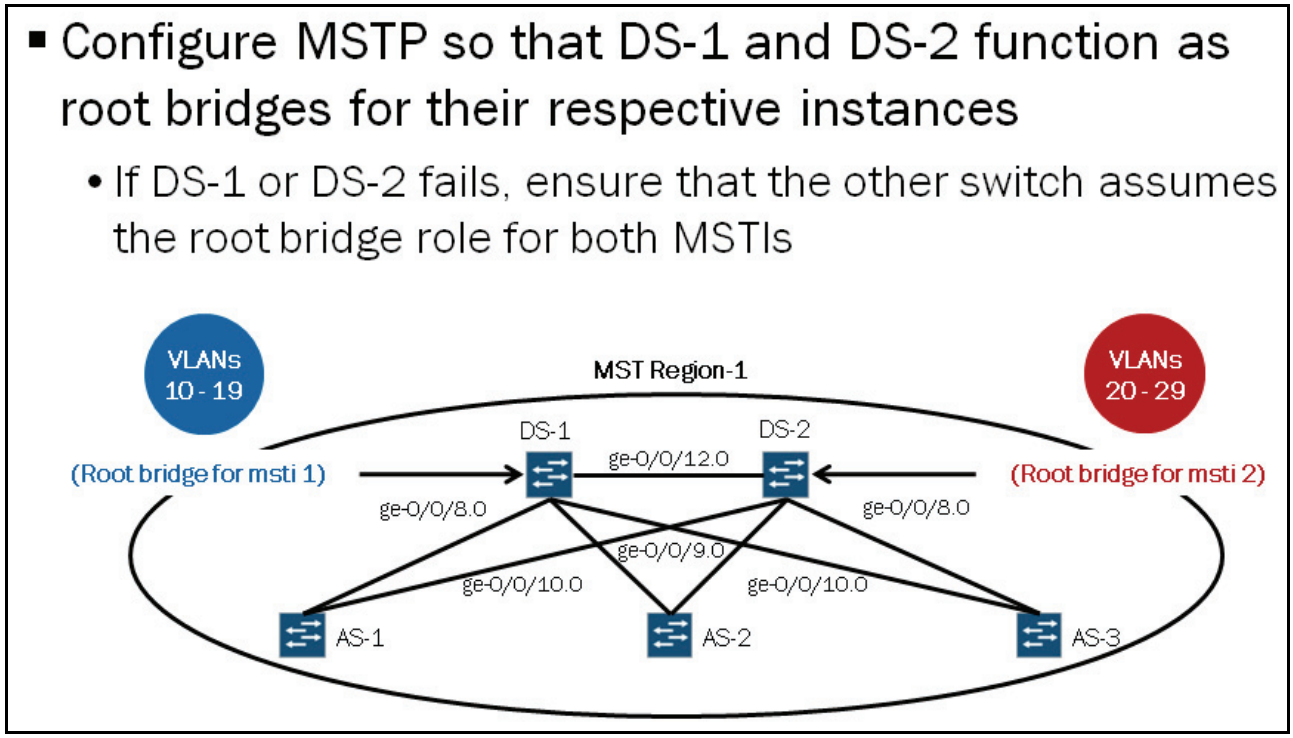
MSTP Configuration



This graphic illustrates the configuration structure for MSTP along with some of the key configuration parameters and considerations. Note that some of the MSTP configuration values must match on all devices participating in the same MSTP region. The MSTP configuration values that must match include:

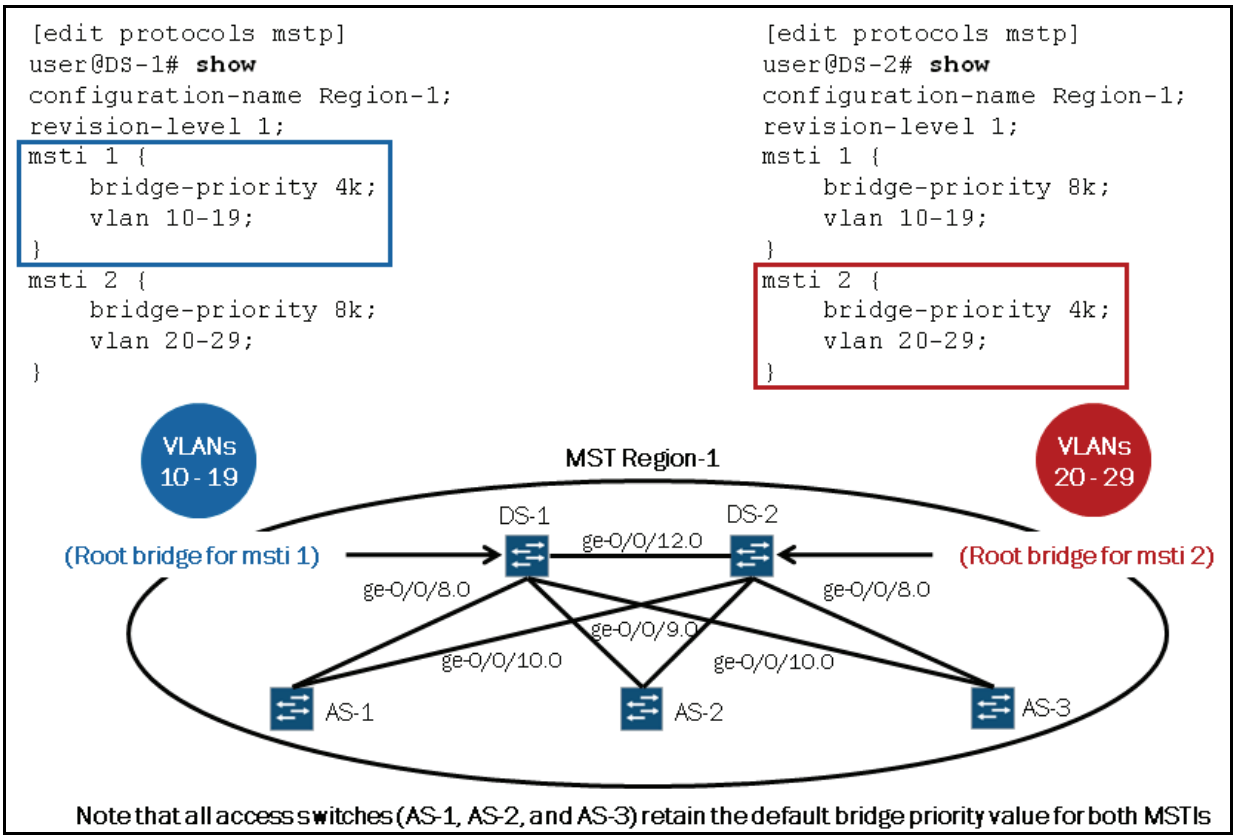
- Configuration name: A user-defined value used to represent the region. Note that this value can be left blank but must match on all devices in a common region.
- Revision level: A user-defined value that represents the MSTP configuration version number. By default this value is 0.
- MSTI-to-VLAN mapping: A mapping between a specific MSTI and the VLANs that MSTI will service. This value must match on all devices in a common MSTP region. All VLANs not specifically mapped to a user-defined MSTI are automatically associated with MSTI 0 (the common spanning tree instance).

Case Study: Topology and Objectives



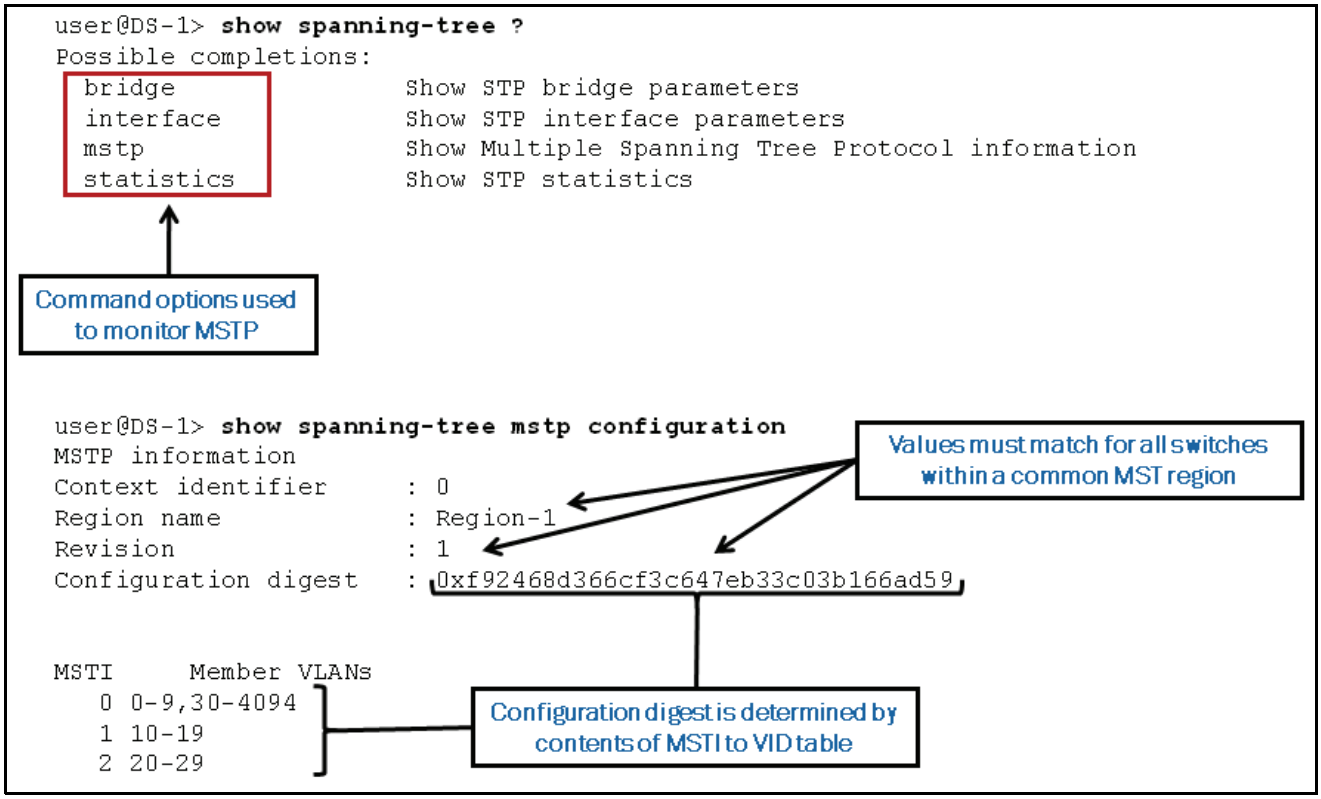
This graphic introduces the topology and objectives used throughout this case study.

Case Study: Configuring MSTP

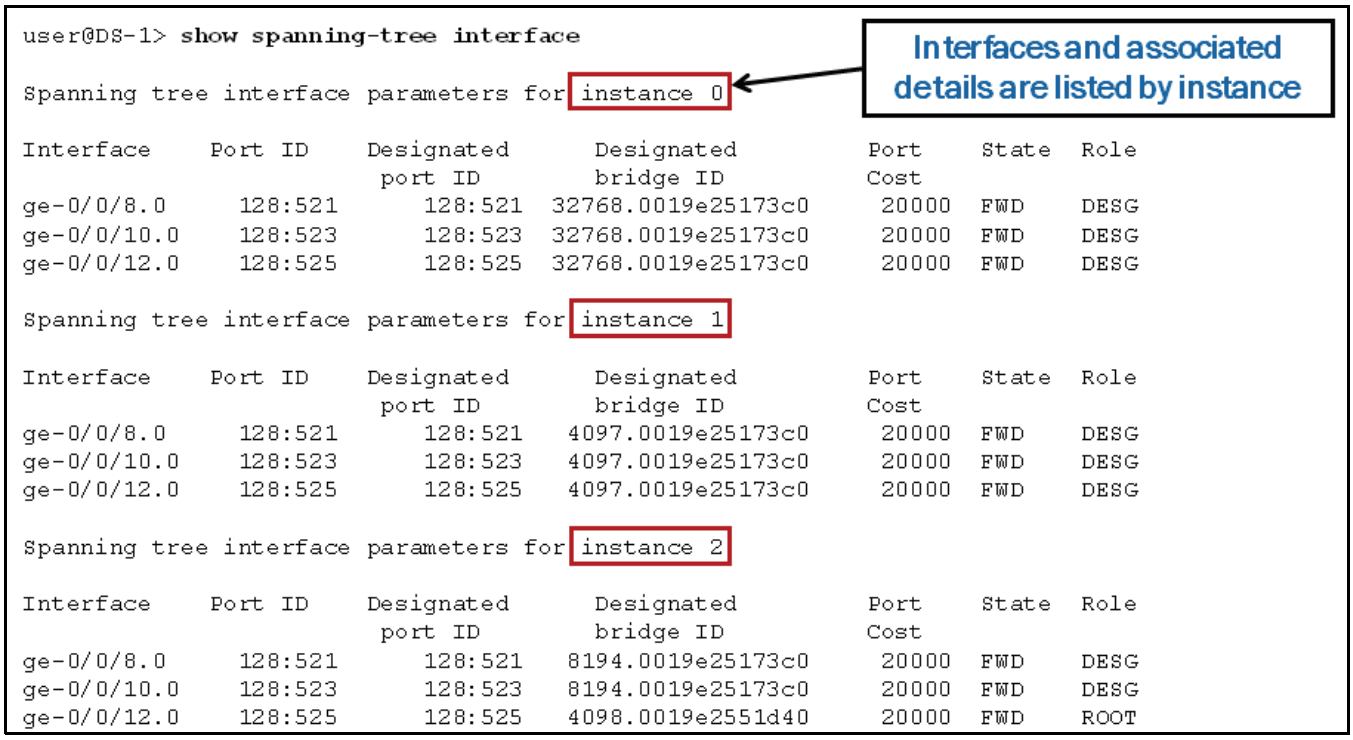


This graphic provides the configuration required on DS-1 and DS-2 to accomplish the objectives outlined on the previous graphic. Note that the configuration on AS-1, AS-2, and AS-3 is very similar to that shown on the graphic with the exception of the configured bridge priority values (AS-1, AS-2, and AS-3 all use the default bridge priority of 32K).

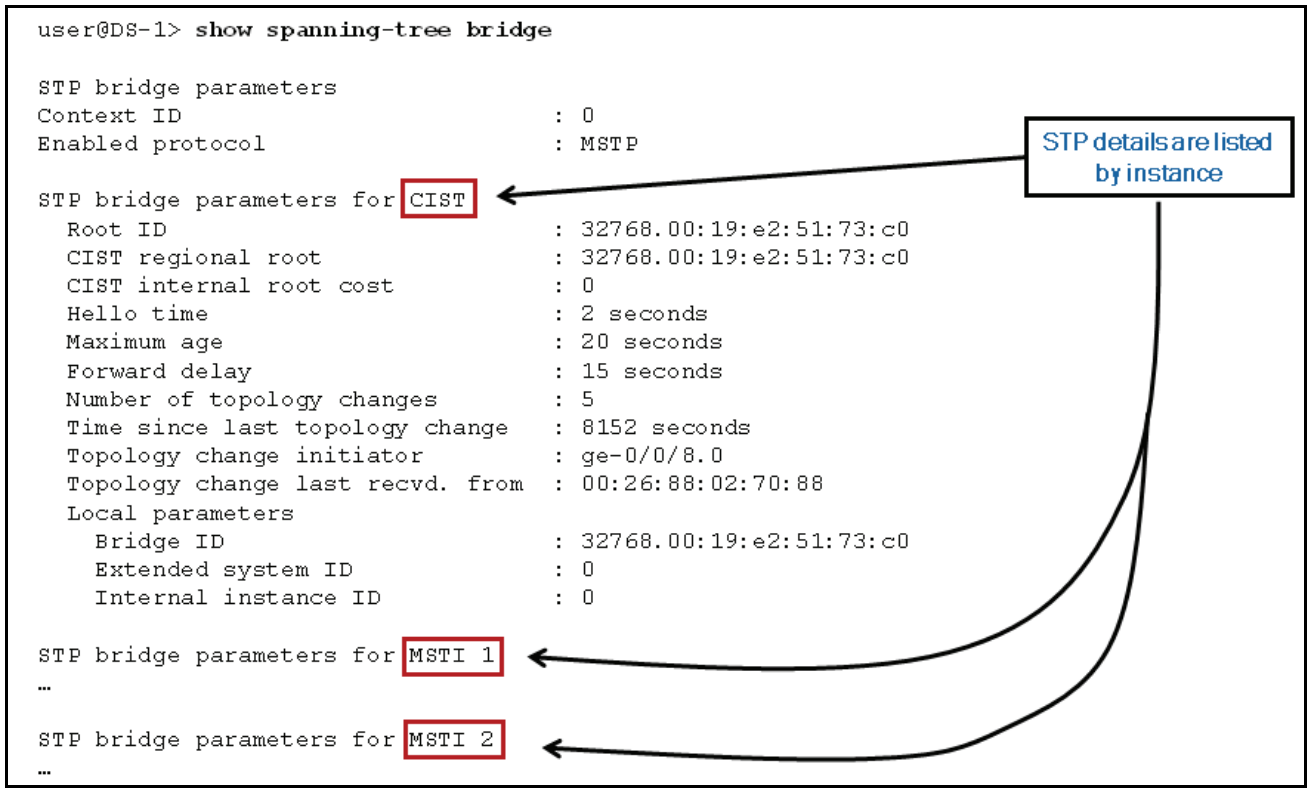
Case Study: Monitoring MSTP



This graphic illustrates the operational-mode commands used to monitor MSTP along with a sample output from the **show spanning-tree mstp configuration** command.



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.

Review Questions

1. What limitation with STP and RSTP is overcome when using MSTP?
2. What is the purpose of the CIST in MSTP?

Answers

1.

While RSTP provides several advantages over STP neither of these protocols allow for load balancing, which in some environments is a requirement. In environments where RSTP or STP is used, all VLANs within a LAN share the same spanning tree, which limits the number of forwarding paths for data traffic.

2.

The 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.