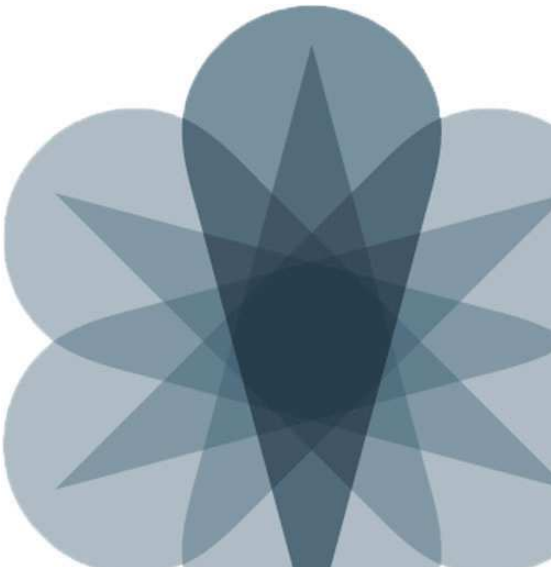
JNCIS-ENT Switching Study Guide

Worldwide Education Services

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



JUNIPER NETWORKS

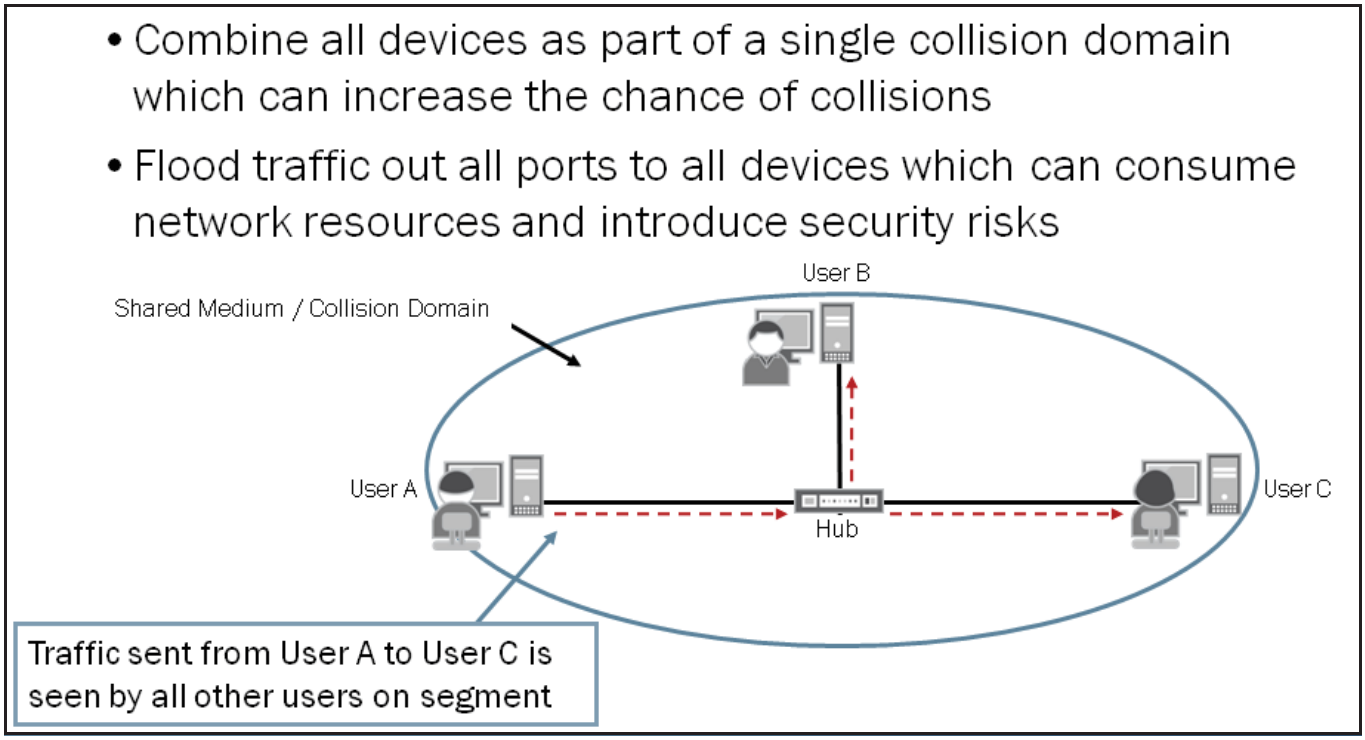
JNCIS-ENT Routing Study Guide

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.

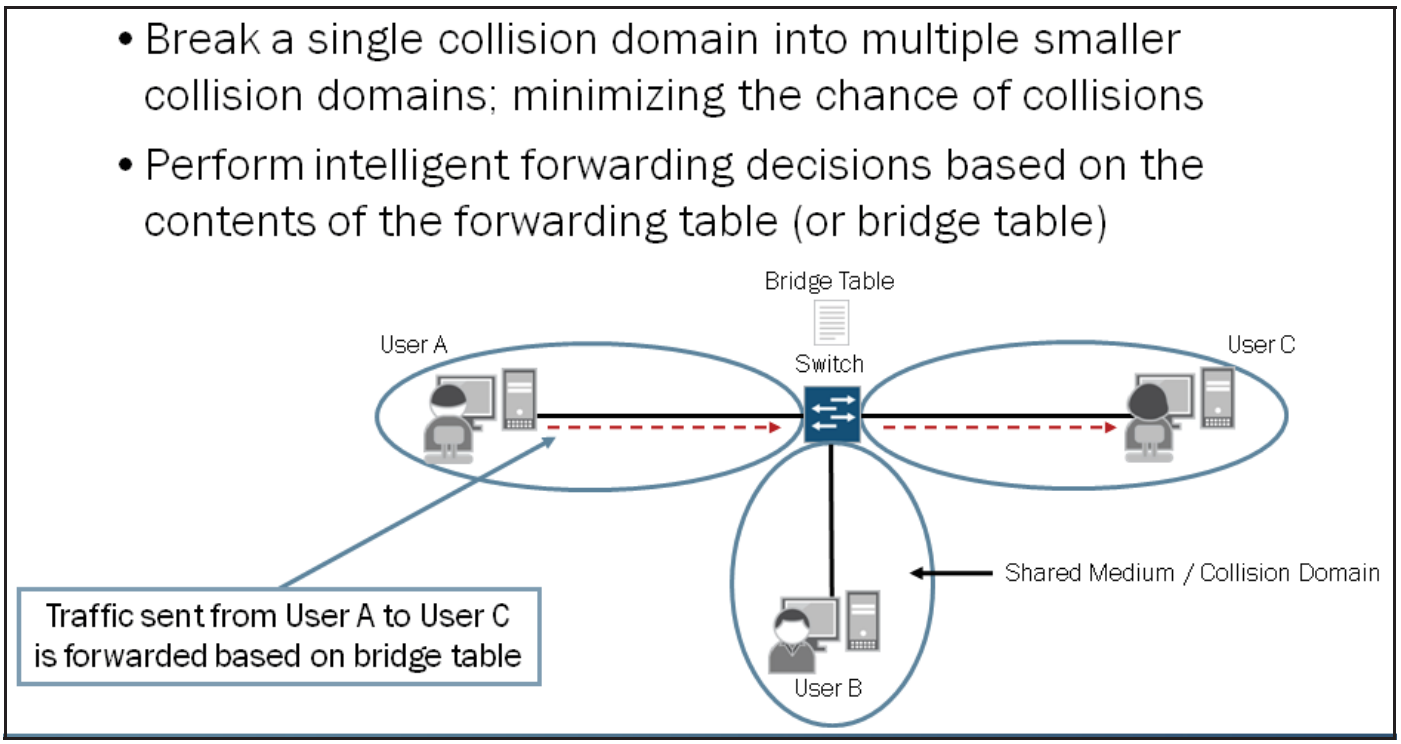
Ethernet uses the carrier-sense multiple access with collision detection (CSMA/CD) protocol to avoid and manage frame collisions. The sample topology on the graphic shows a series of nodes connected through a hub using a copper-based physical medium. This type of implementation only allows a single stream 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.

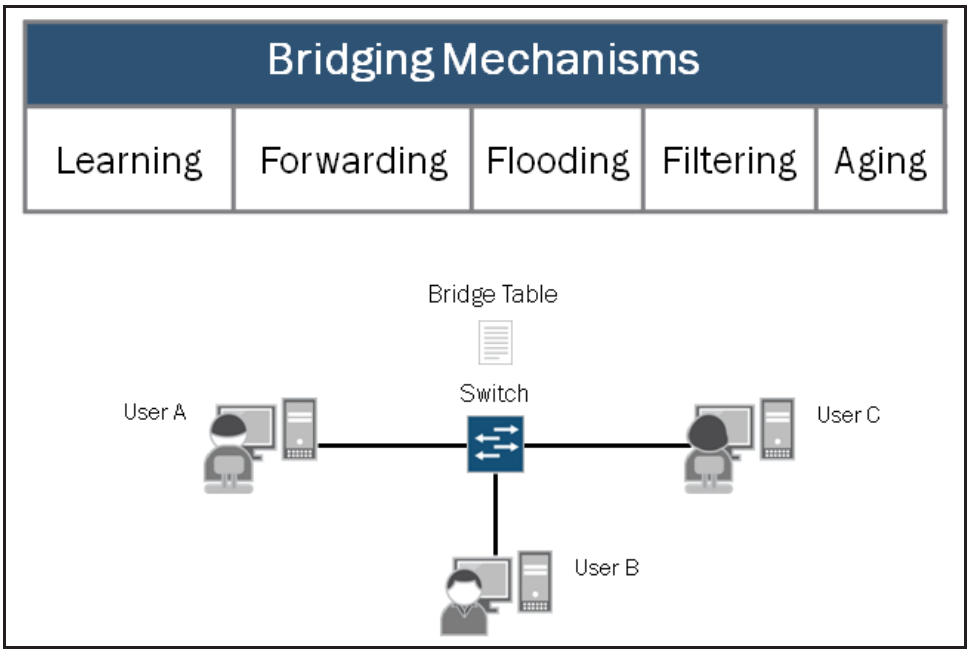
Switched LANs



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?

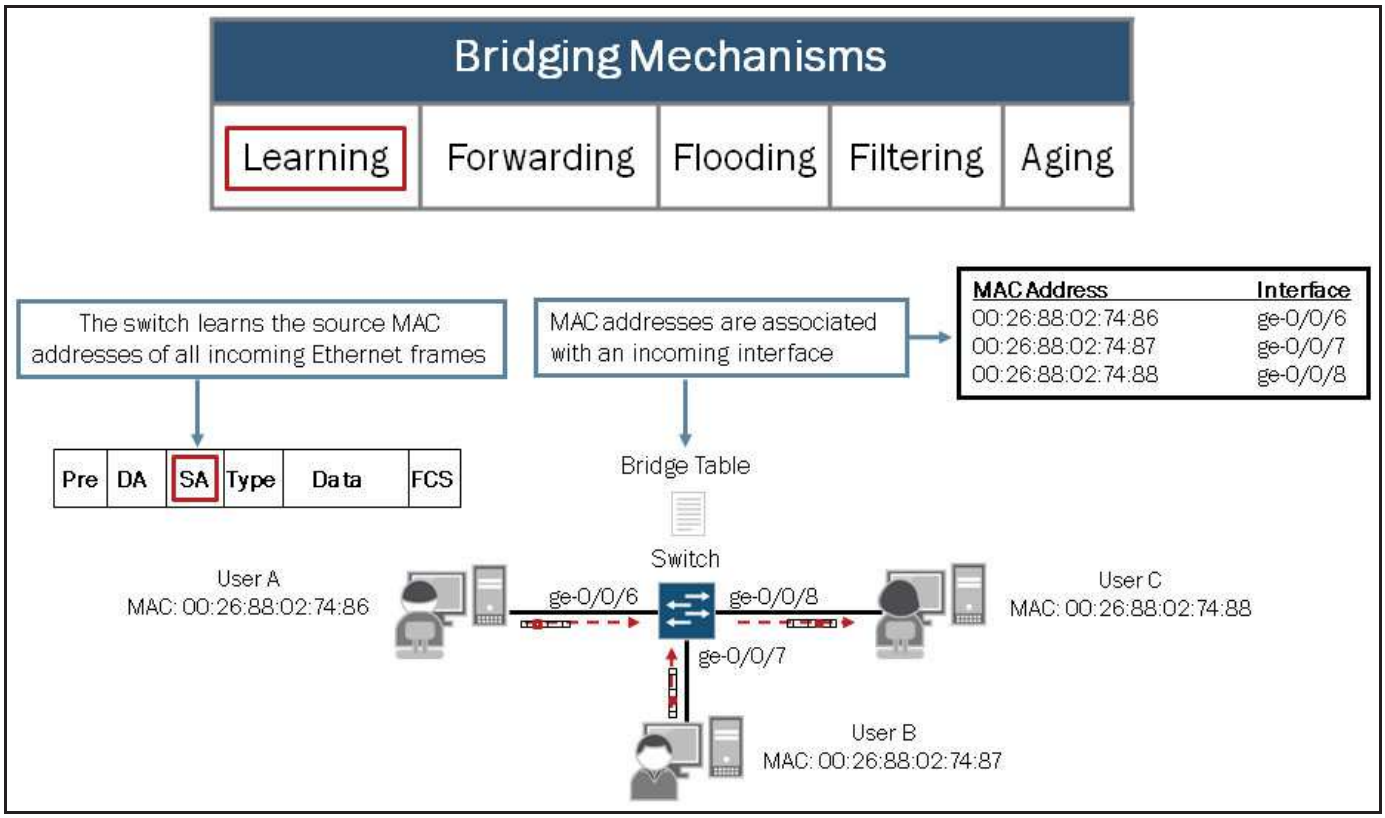


Defined in the IEEE 802.1D-2004 standard, bridging addresses some of the inherent problems of large shared Ethernet LANs. Bridging uses microsegmentation to divide a single collision domain into multiple, smaller bridged collision domains. Reducing the size of a collision domain effectively reduces the 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

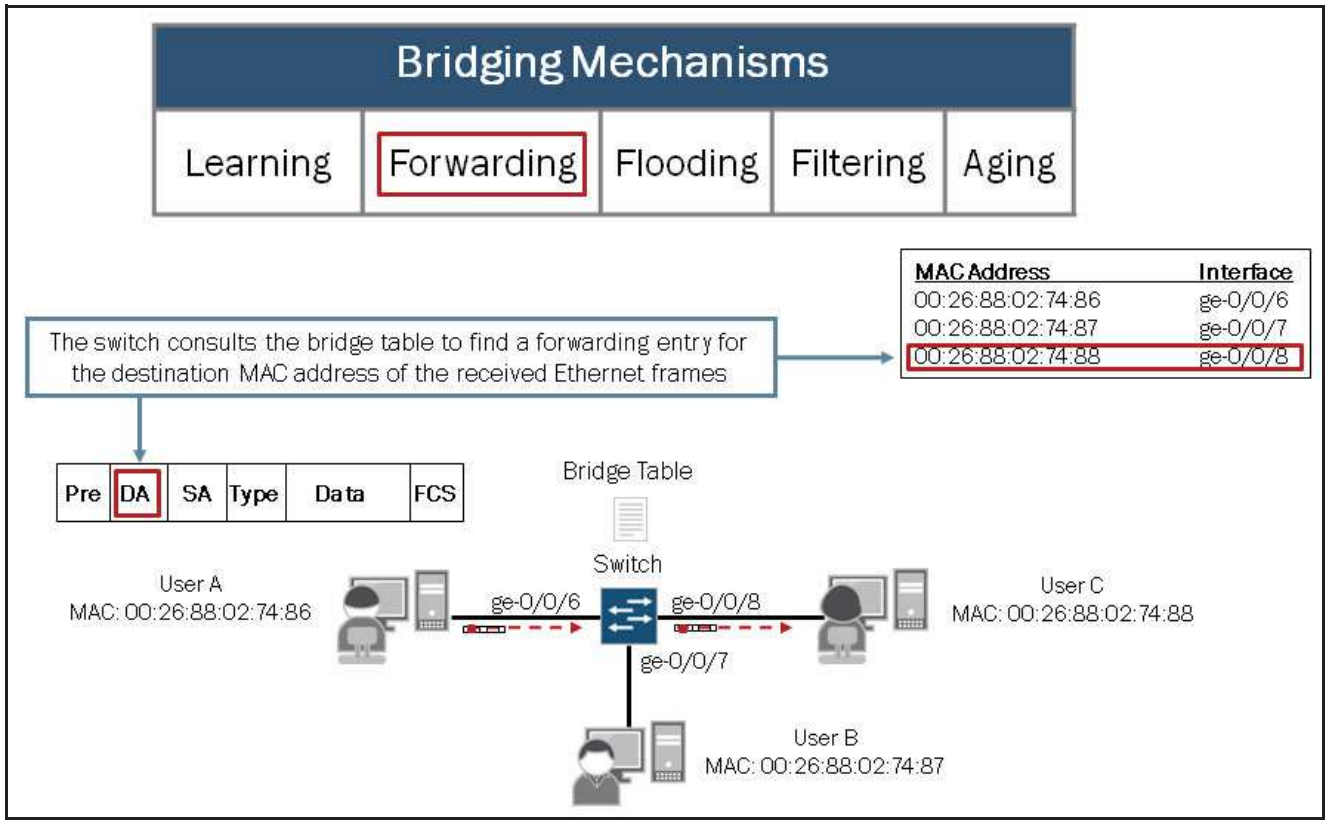


When a switch is first connected to an Ethernet LAN, it has no information about the devices connected to the network. *Learning* is the process a switch uses to obtain the MAC addresses of nodes on the network. The switch stores all learned MAC address in the bridge table. To learn MAC addresses, the 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:

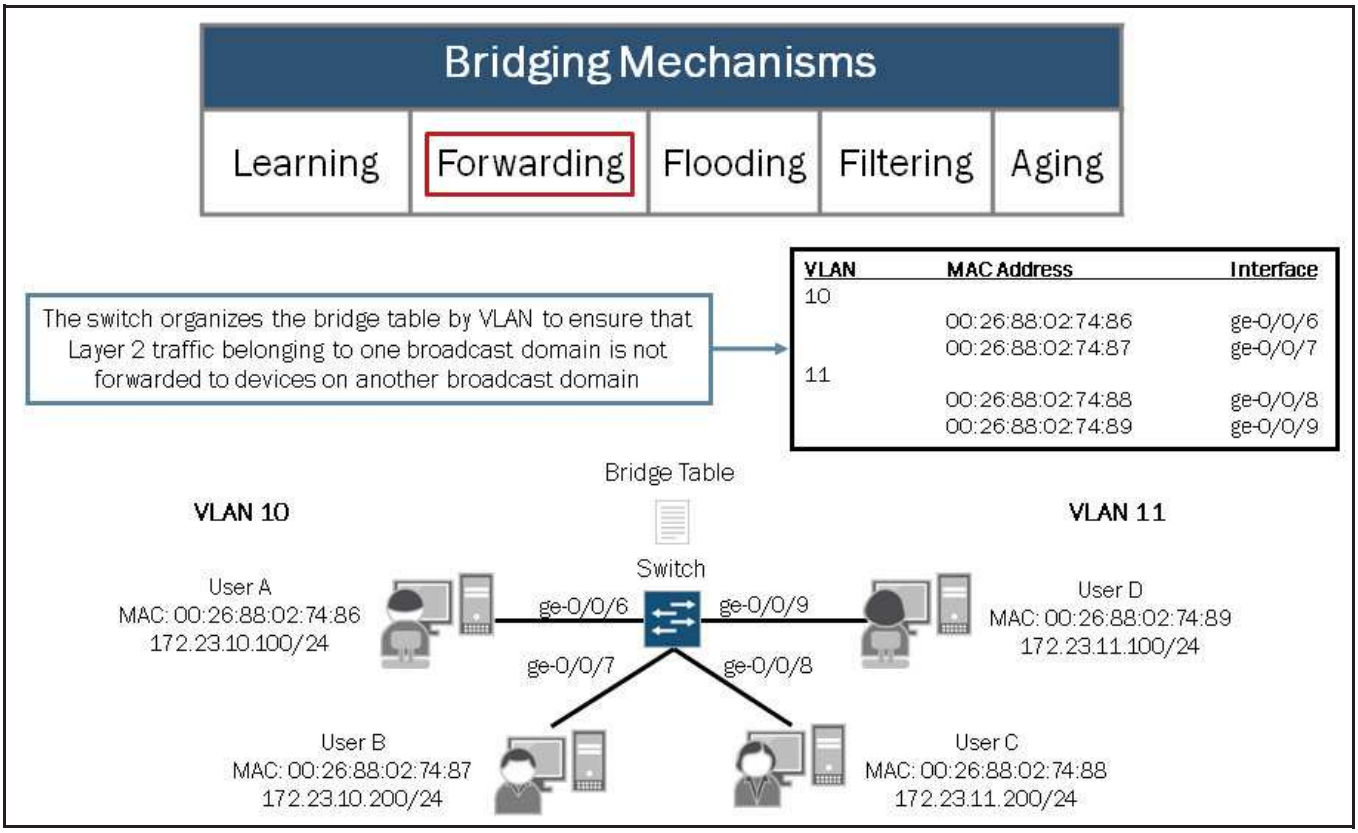
```
{master:0}[edit]
user@Switch# set ethernet-switching-options interfaces ge-0/0/0.0 no?
Possible completions:
  no-mac-learning      Disable mac learning for this interface
```

Forwarding: Part 1



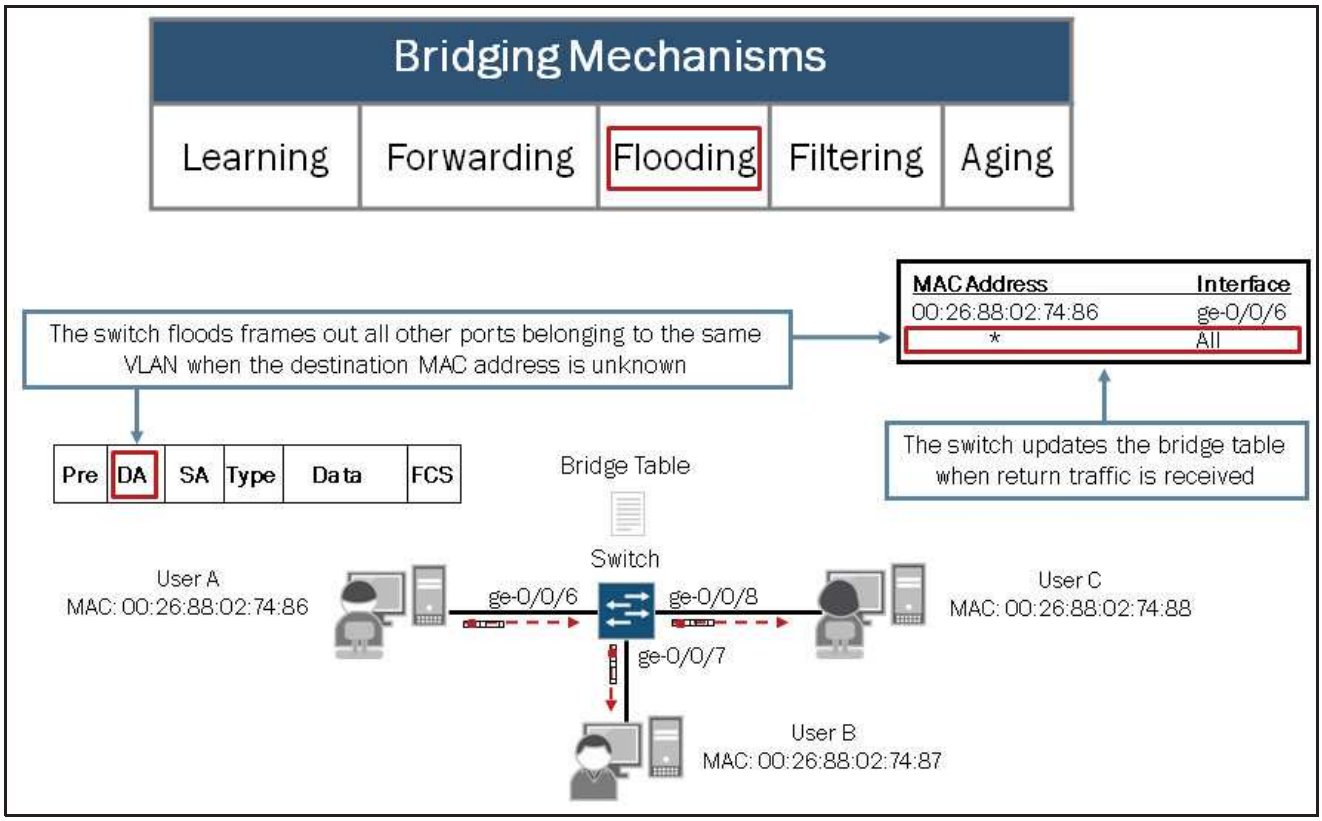
The *forwarding* mechanism is used by the switch to deliver traffic, passing it from an incoming interface to an outgoing interface that leads to (or toward) the destination. To forward frames, the switch consults the bridge table to see whether the table contains the MAC address corresponding to the 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



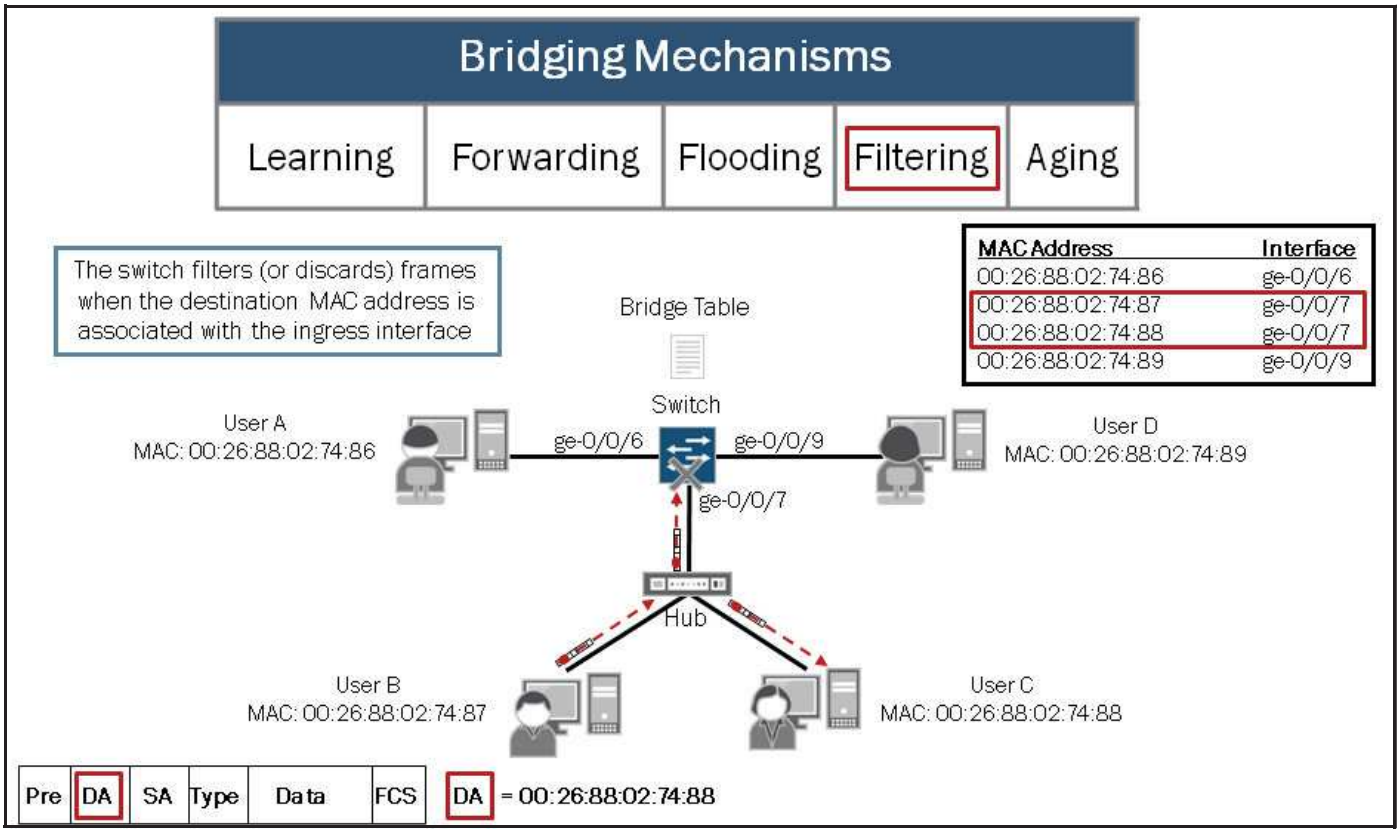
To forward frames, the switch consults the bridge table to see whether the table contains the MAC address corresponding to the frames' destination. The bridge table is organized by VLAN to ensure Layer 2 traffic is only forwarded out switch ports belonging to the same broadcast domain (VLAN) as the interface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

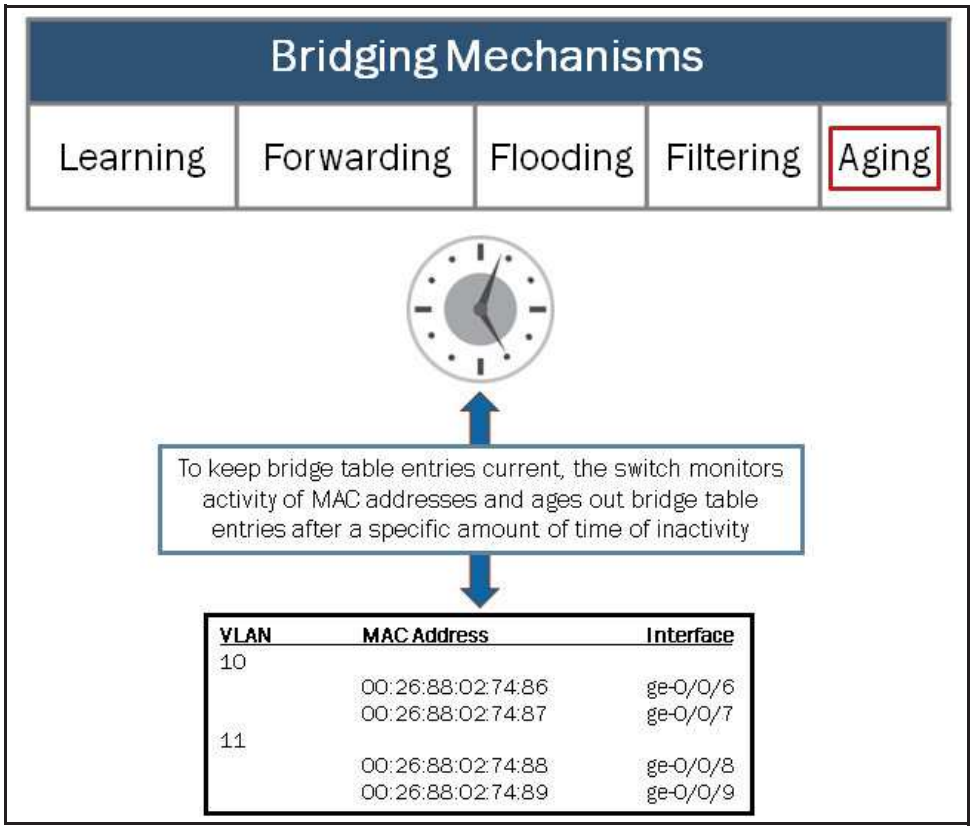
Filtering



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.

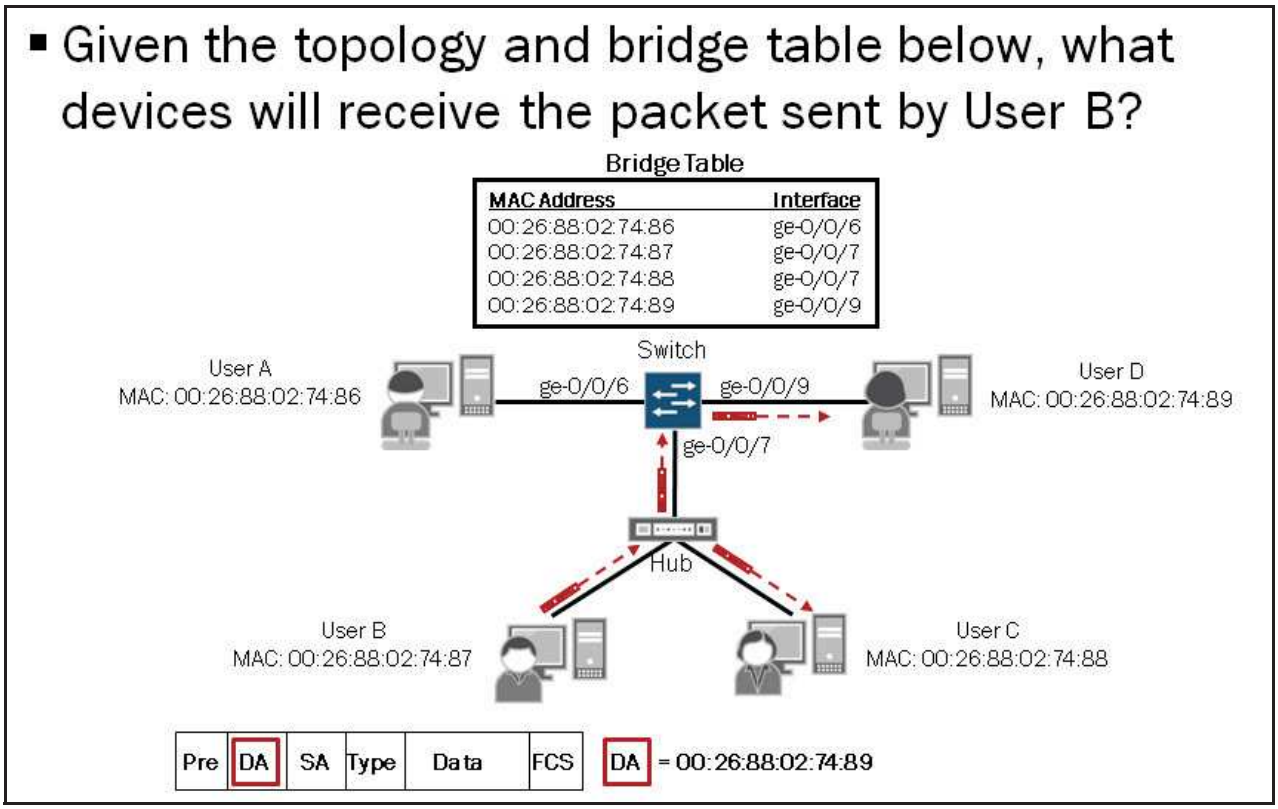
Aging



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:

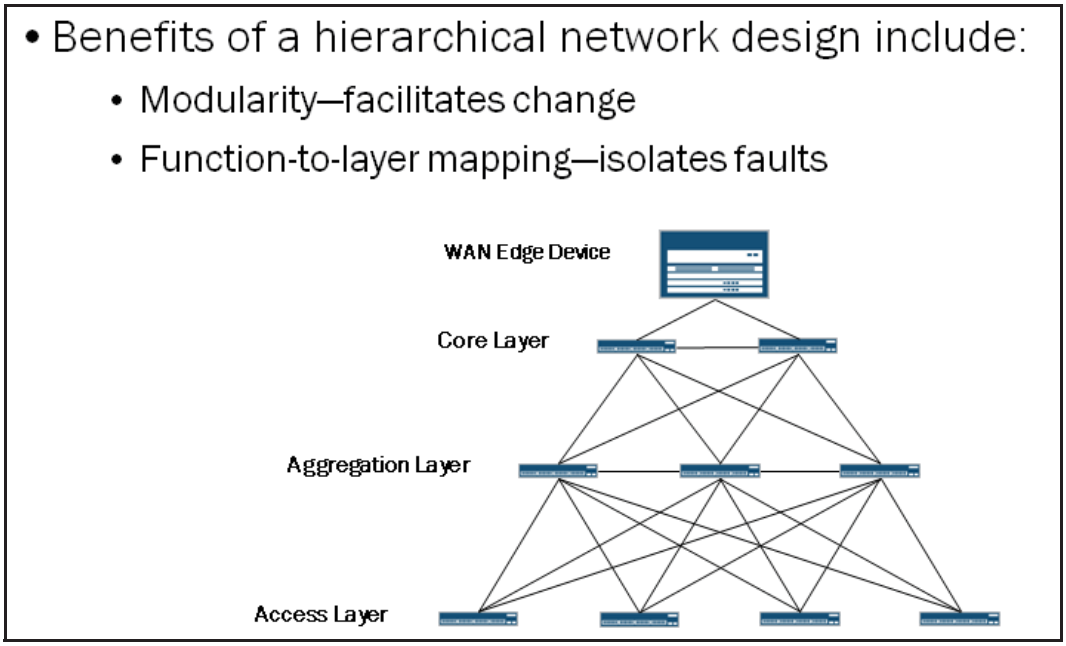
```
{master:0}[edit]
user@switch# set ethernet-switching-options mac-table-aging-time ?
Possible completions:
  <mac-table-aging-time>  MAC aging time (60..1000000 seconds)

{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



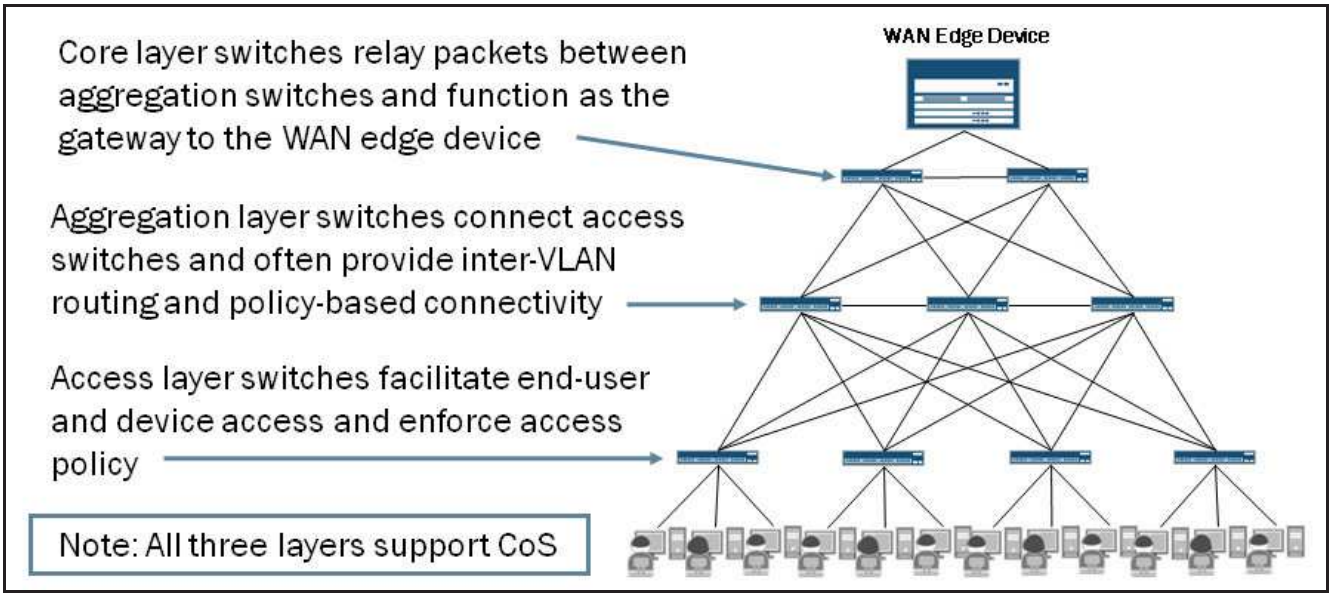
Switched networks are often hierarchical and consist of multiple layers. The diagram on the graphic illustrates the typical layers, which include access, aggregation (or distribution), and core. Each of these layers performs unique responsibilities.

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.

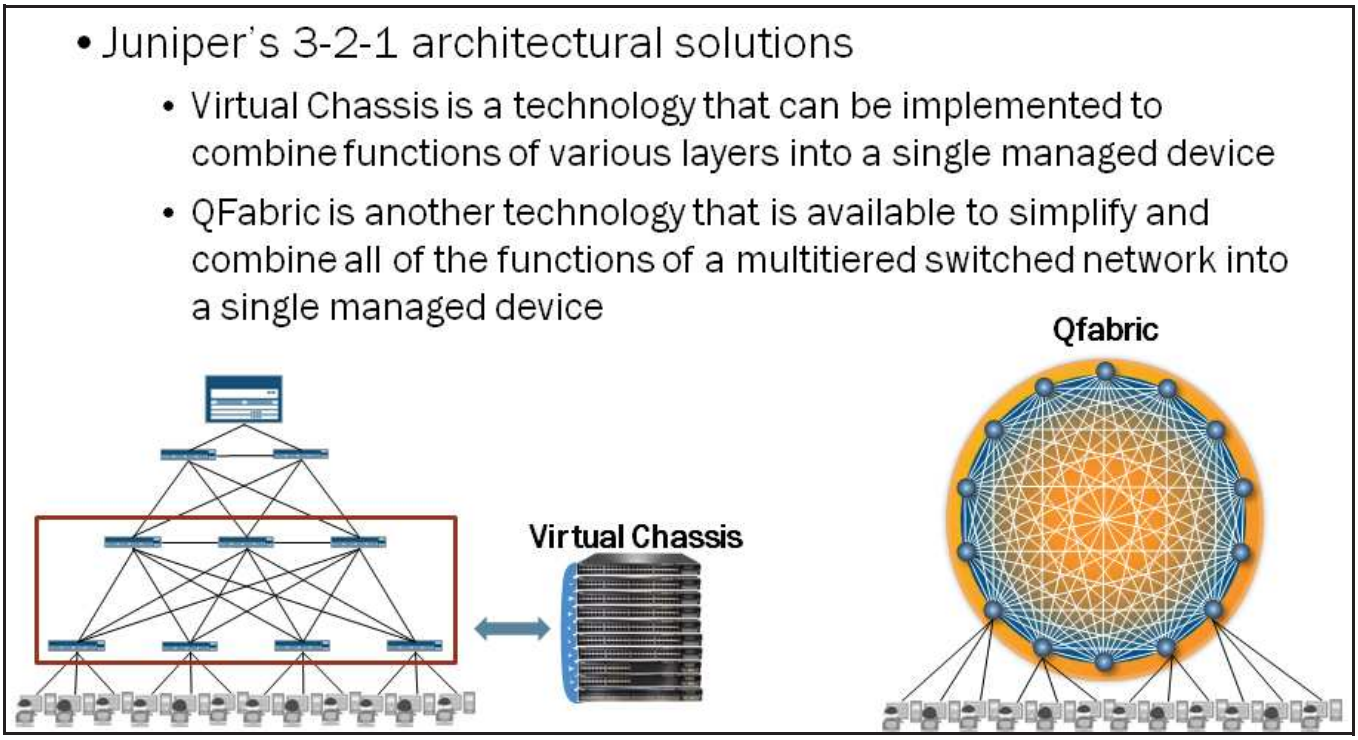
Functions of Layers



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



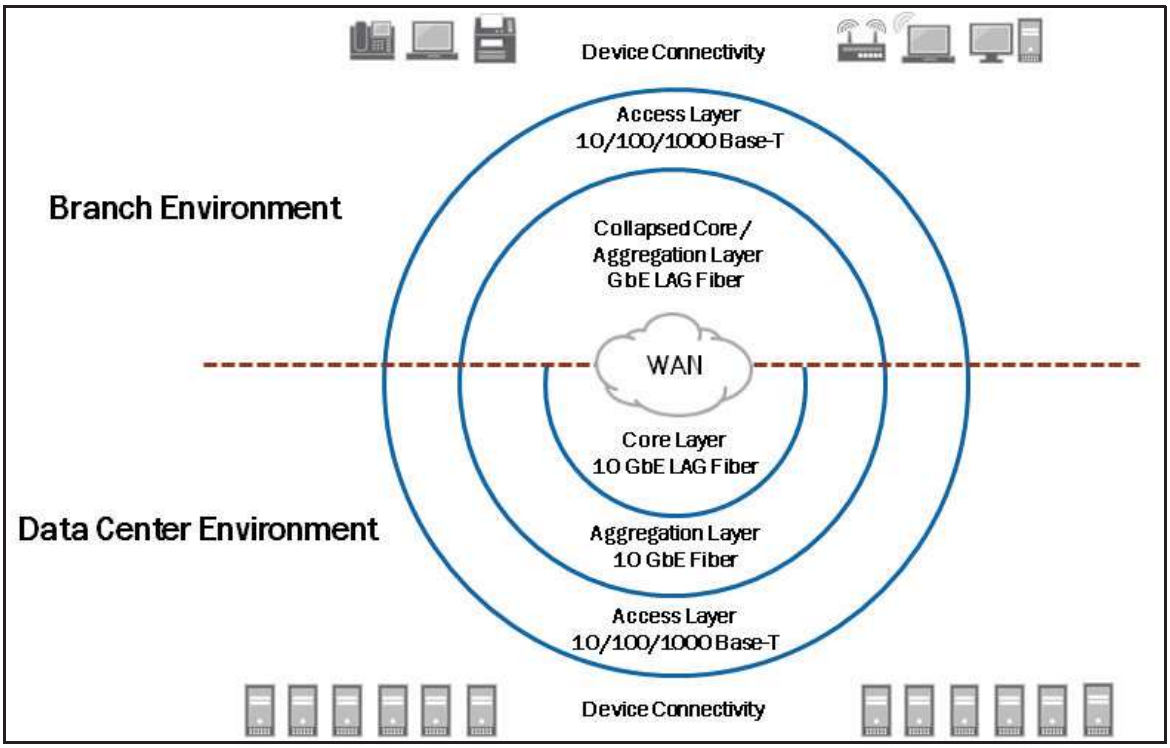
As networks and data centers continue grow, so does the complexity and overall number of devices that must be managed. We deliver a strategy for simplifying the data center network called the 3-2-1 data center network architecture. The 3-2-1 architecture eliminates layers of switching to flatten and 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



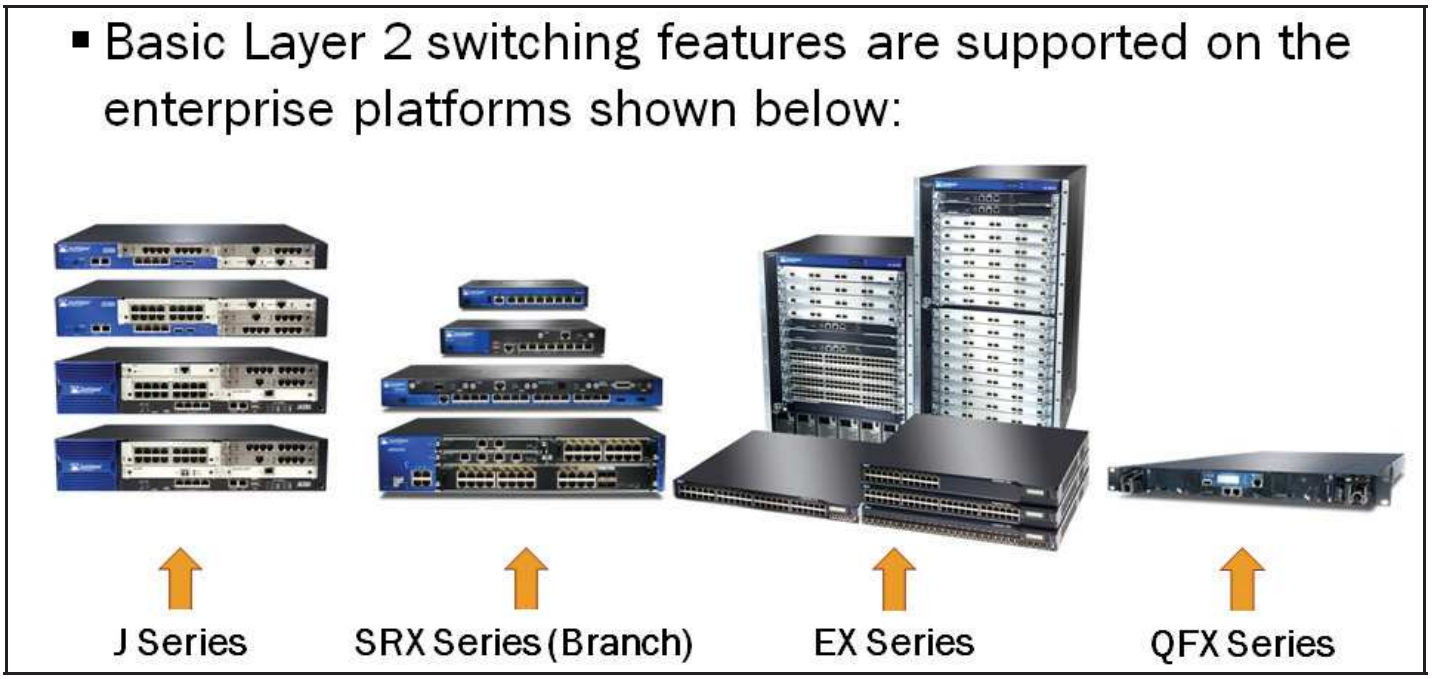
This graphic illustrates some points of comparisons between branch and data center environments. As shown on the graphic, branch environments typically do not have the three distinct hierarchical layers while data center (and many campus) environments do. In many branch environments, the core and aggregation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

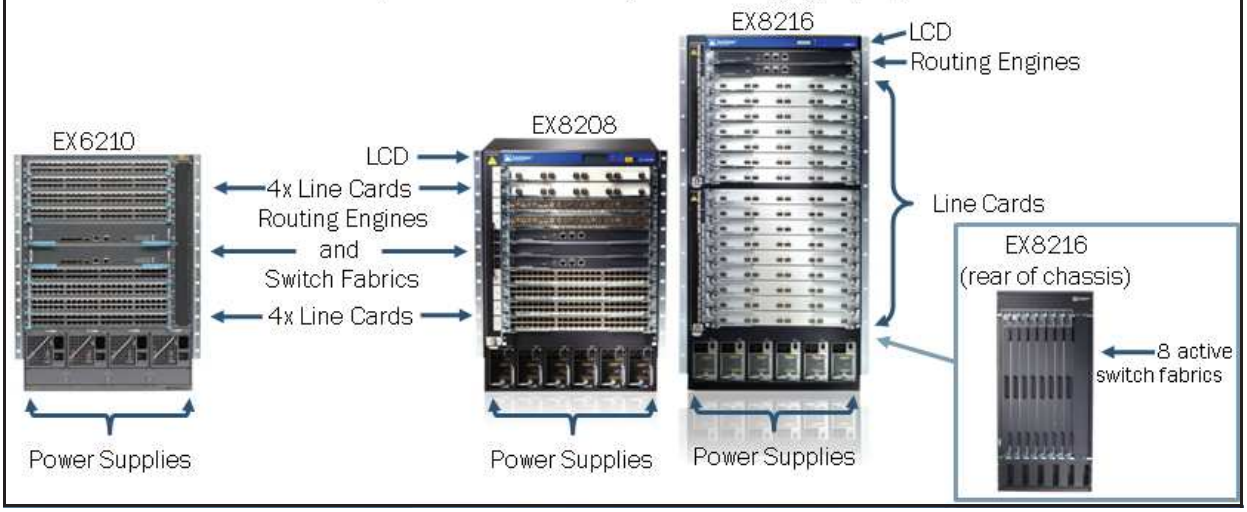
A brief description of the fixed chassis EX Series switches that run the Junos OS follows:

- 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])



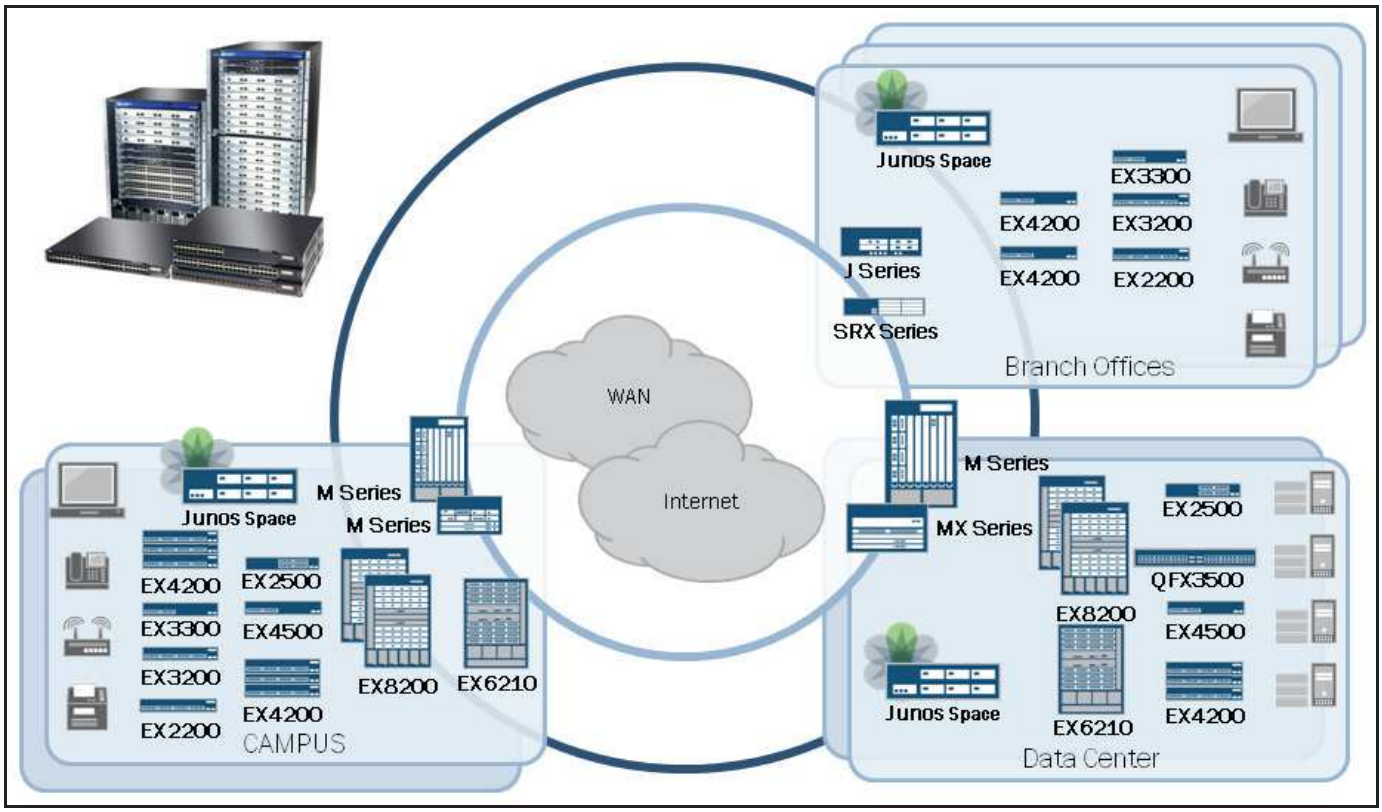
The EX6200 line of Ethernet switches is ideal for large enterprise campus and data center environments. The EX6210 is a 10-slot chassis featuring eight dedicated line-card slots that can accommodate a combination of the two available 48-port 10/1000/1000BASE-T line-card options. One option comes with 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.

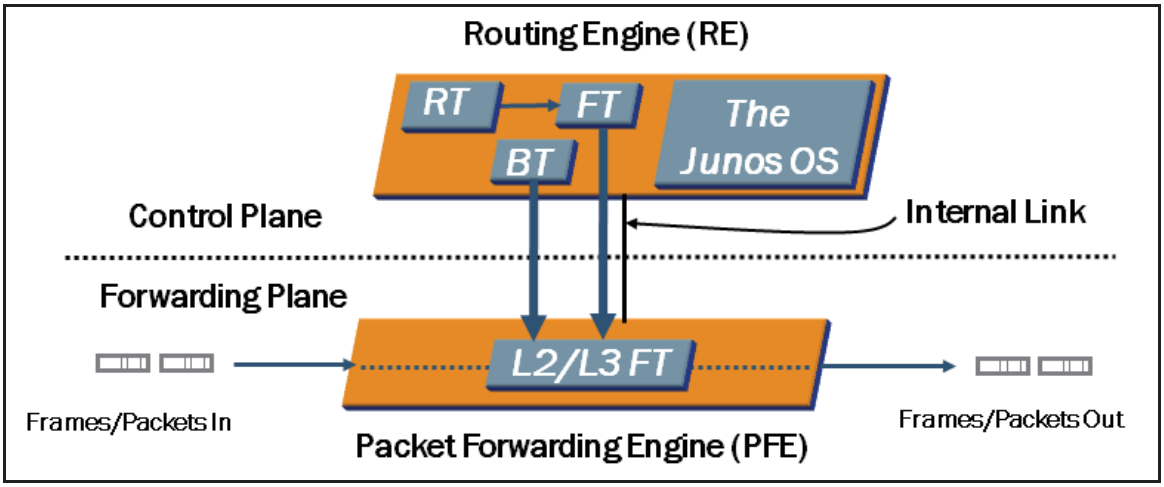
Support of the various Layer 2 switching features varies between platforms. For support information or more details on a specific EX Series platform, refer to the technical publications or the product-specific datasheets and literature found at: http://www.juniper.net/techpubs/ and http://www.juniper.net/us/en/products-services/switching/ex-series/,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

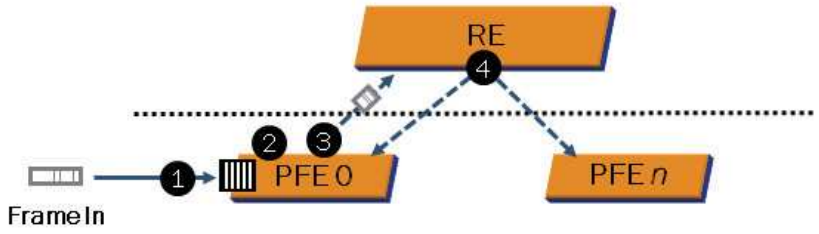
The PFE receives the Layer 2 and Layer 3 forwarding table from the RE by means of an internal link. Forwarding table updates are a high priority for the Junos OS kernel and are performed incrementally. The internal link that connects the RE and PFE is rate-limited to protect the RE from DoS attacks. 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.

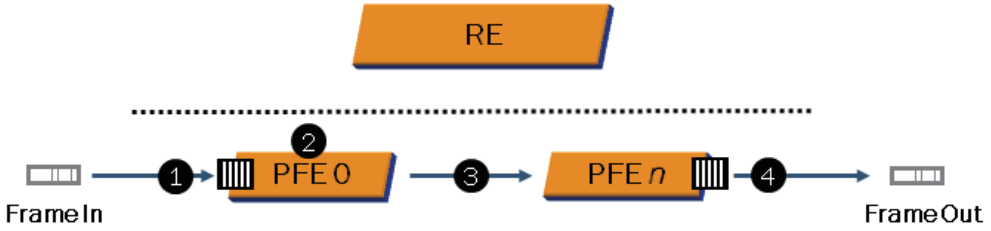


When frames enter a switch port, they are processed by the ingress PFE associated with that port. The ingress PFE determines how transit frames are processed and which lookup table is used when determining next-hop information. The PFE performs a lookup on the source and destination MAC address. In the 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.

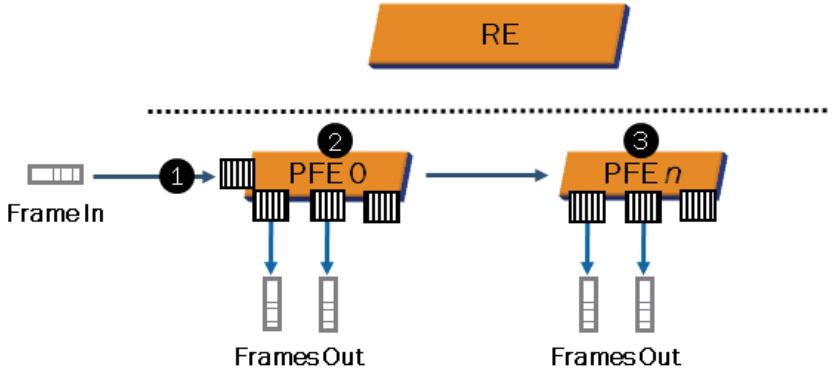


In the example illustrated on the graphic, the destination MAC address exists in the bridge table. If the egress port belongs to the ingress PFE, the frame is switched locally. If the egress port belongs to a PFE other than the ingress PFE (as shown in the example on the graphic), the frame is forwarded 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.

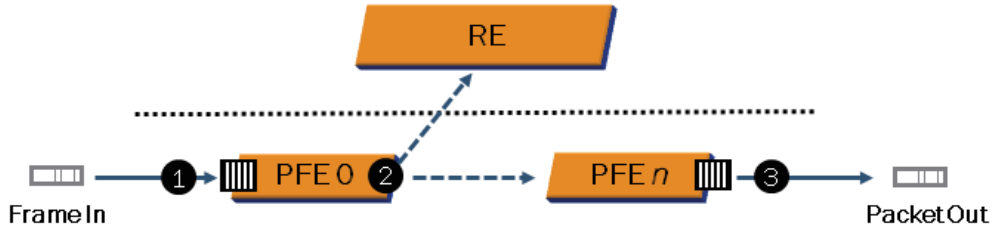


When the ingress PFE performs a lookup on the destination MAC address and no entry exists in the bridge table, the frame is flooded out all ports in the same broadcast domain. The frame is also flooded to other PFEs. However, the frame is not flooded out the port on which it was received. Once the switch 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:

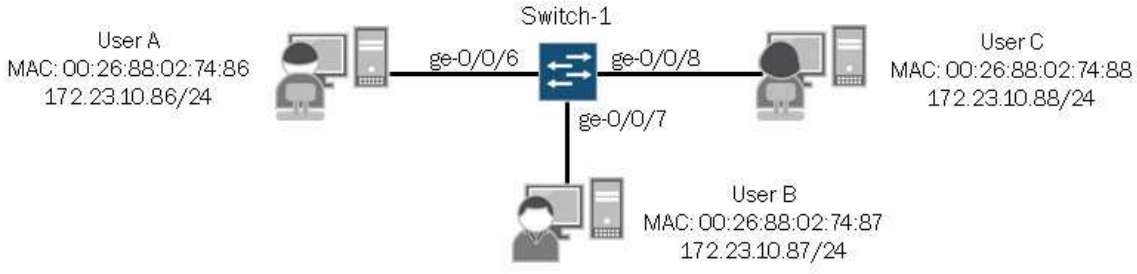
1. Frame enters ingress port and attached ingress PFE.
2. Ingress PFE performs MAC address lookup. Because the destination MAC address belongs to the switch, PFE performs a Layer 3 lookup.
   a. If the destination IP address belongs to the switch, the decapsulated packet is sent to the RE for processing.
   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.

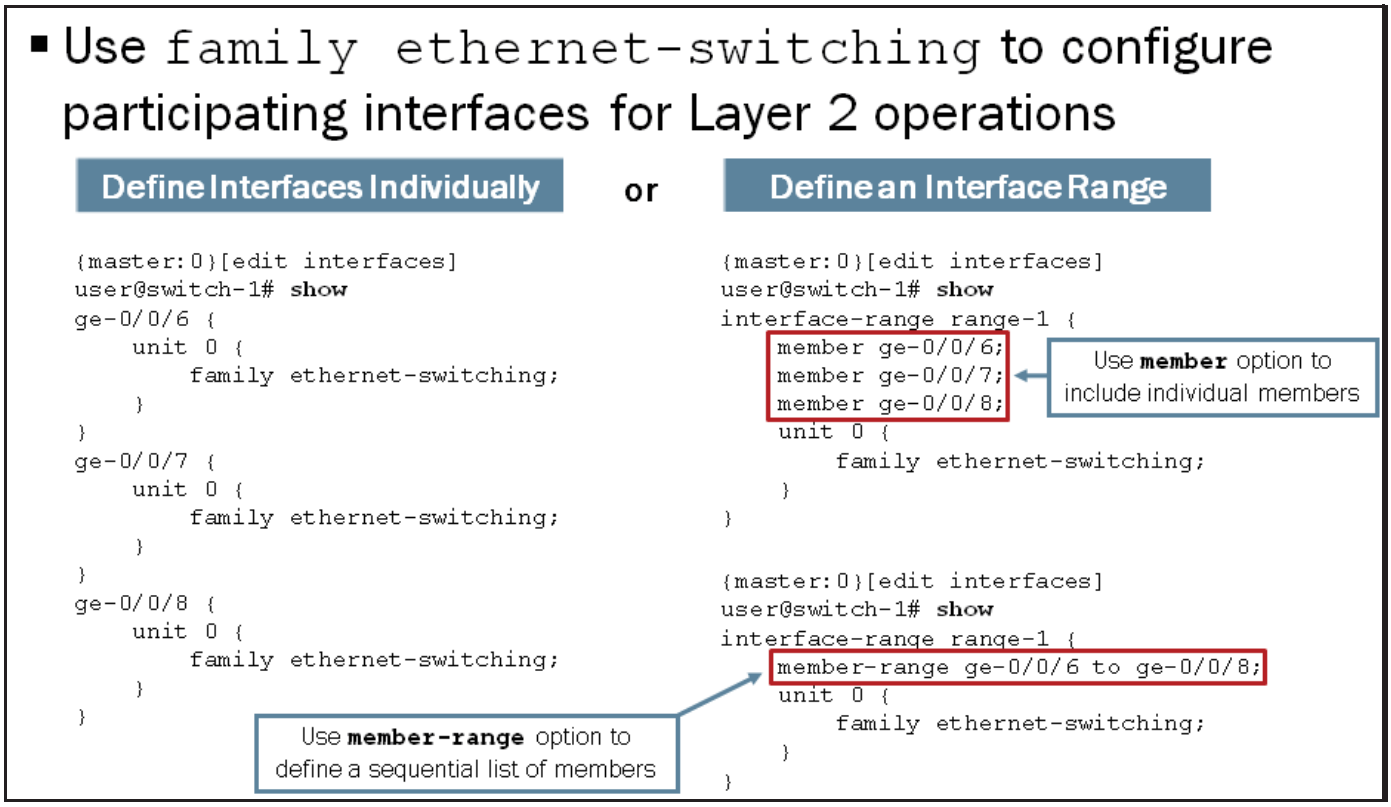
Case Study: Topology and Objectives

- 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.

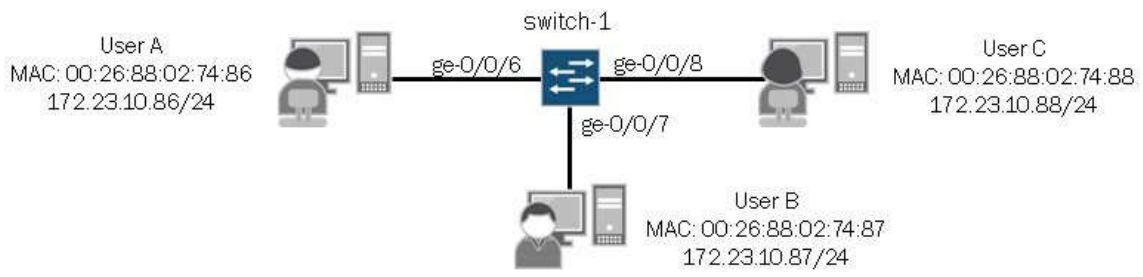
The following example shows the interface configuration where auto-negotiation is disabled and the speed and duplex settings are hard-coded to 1000 mbps and full-duplex respectively:

```
{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.

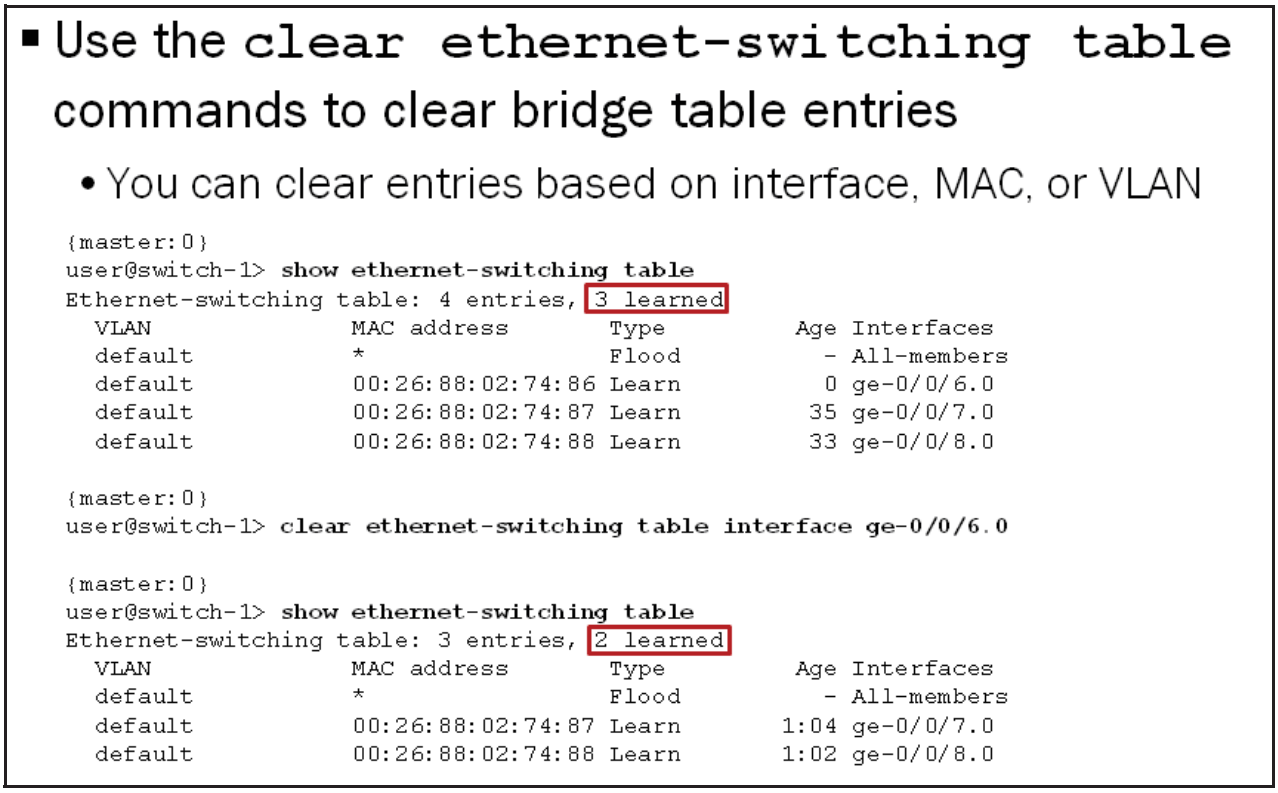
You can add the **extensive** option to view additional details:

```
{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** 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.

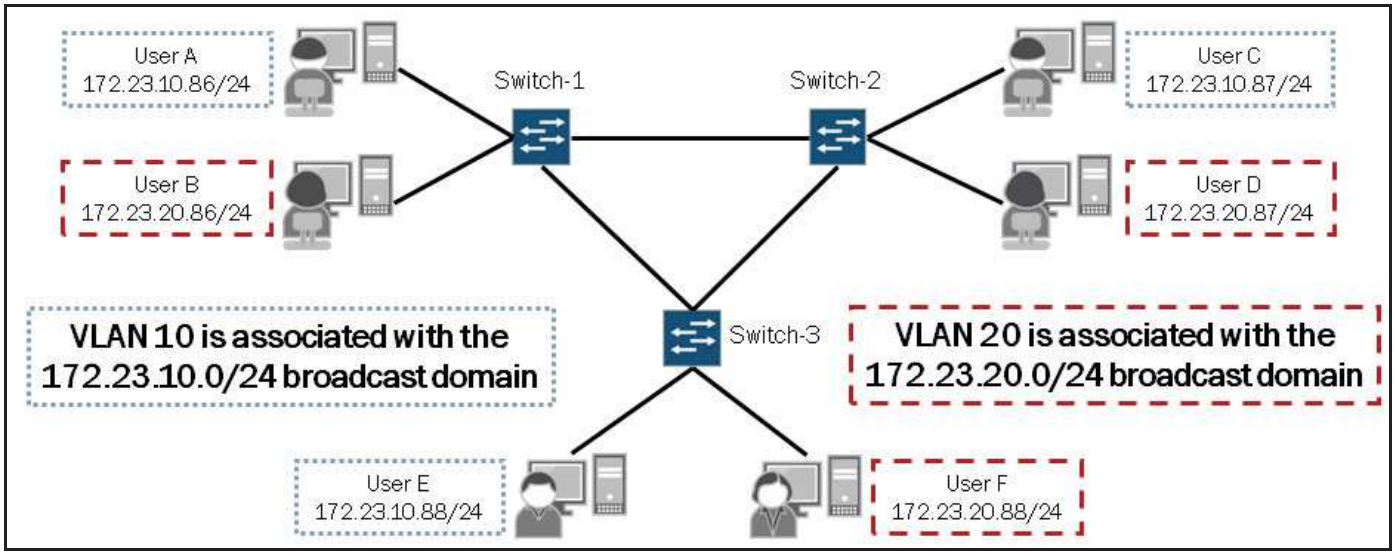
JNCIS-ENT Routing Study Guide

Chapter 2: Virtual Networks

This Chapter Discusses:

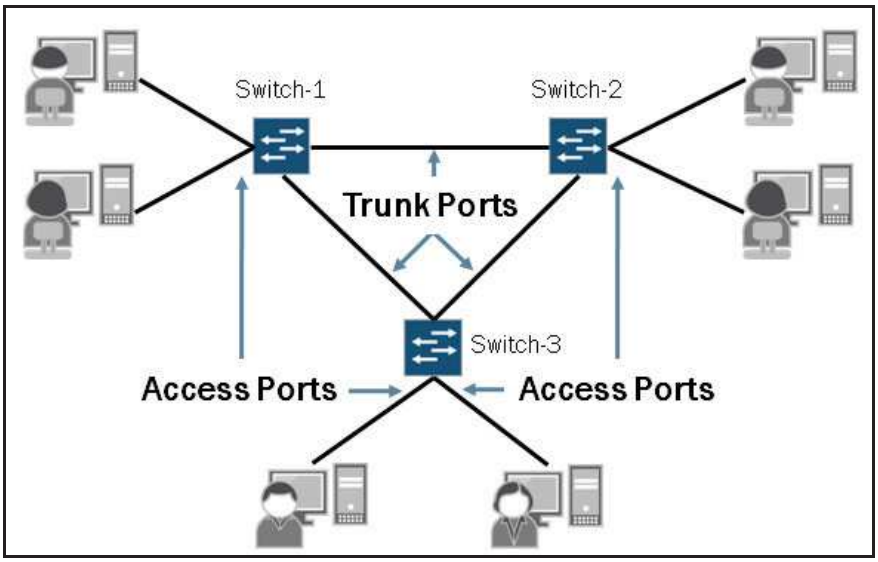
- The concept of a virtual network;
- Access and trunk ports;
- The configuration and monitoring of virtual LANs (VLANs);
- Voice and native VLAN concepts and configuration;
- Inter-VLAN routing operations; and
- The configuration and monitoring of inter-VLAN routing.

VLAN Defined



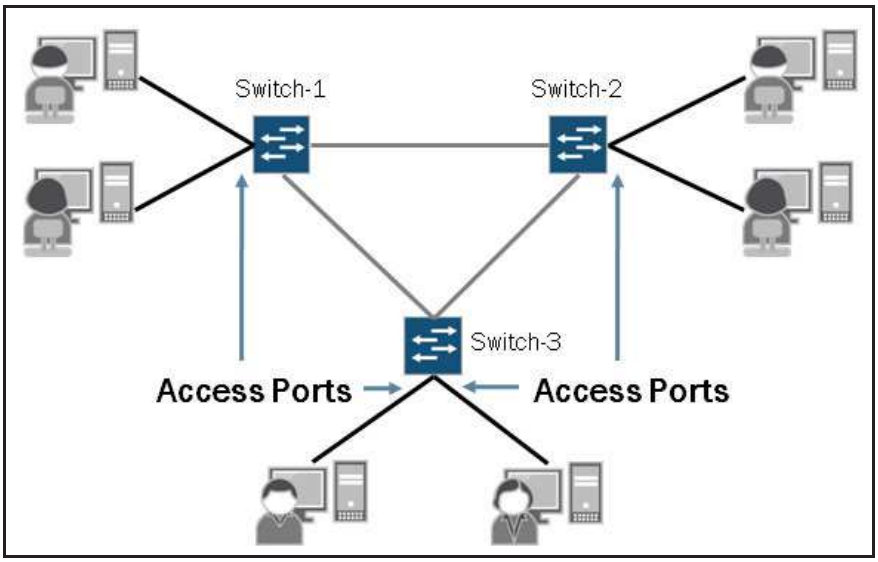
A virtual LAN is a collection of network nodes that are logically grouped together to form separate broadcast domains. A VLAN has the same general attributes as a physical LAN, but it allows all nodes for a particular VLAN to be grouped together, regardless of physical location. One advantage of using 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



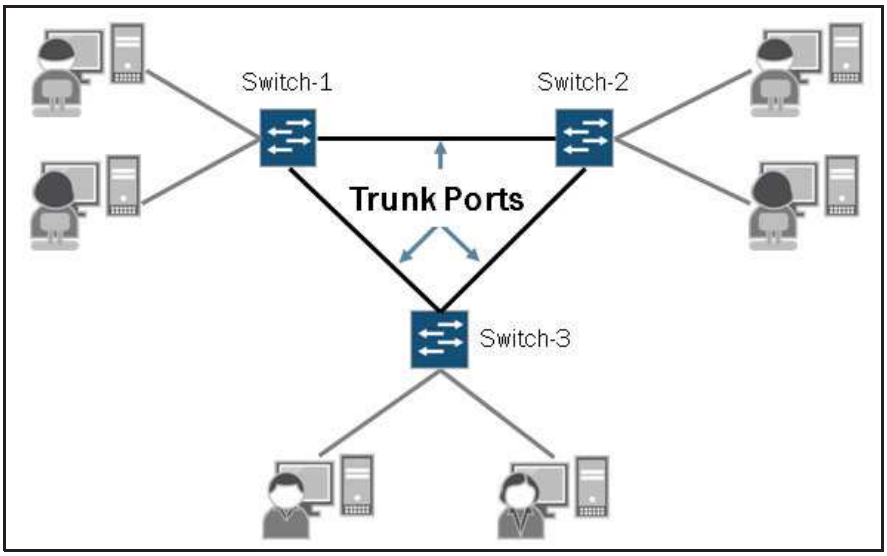
Layer 2 interfaces can be assigned to operate in either access or trunk mode. By default, all installed switch ports on an EX Series switch are configured as access ports. These same switch ports are associated with the default VLAN, which is an untagged VLAN. We discuss the port modes and default VLAN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

Access Ports



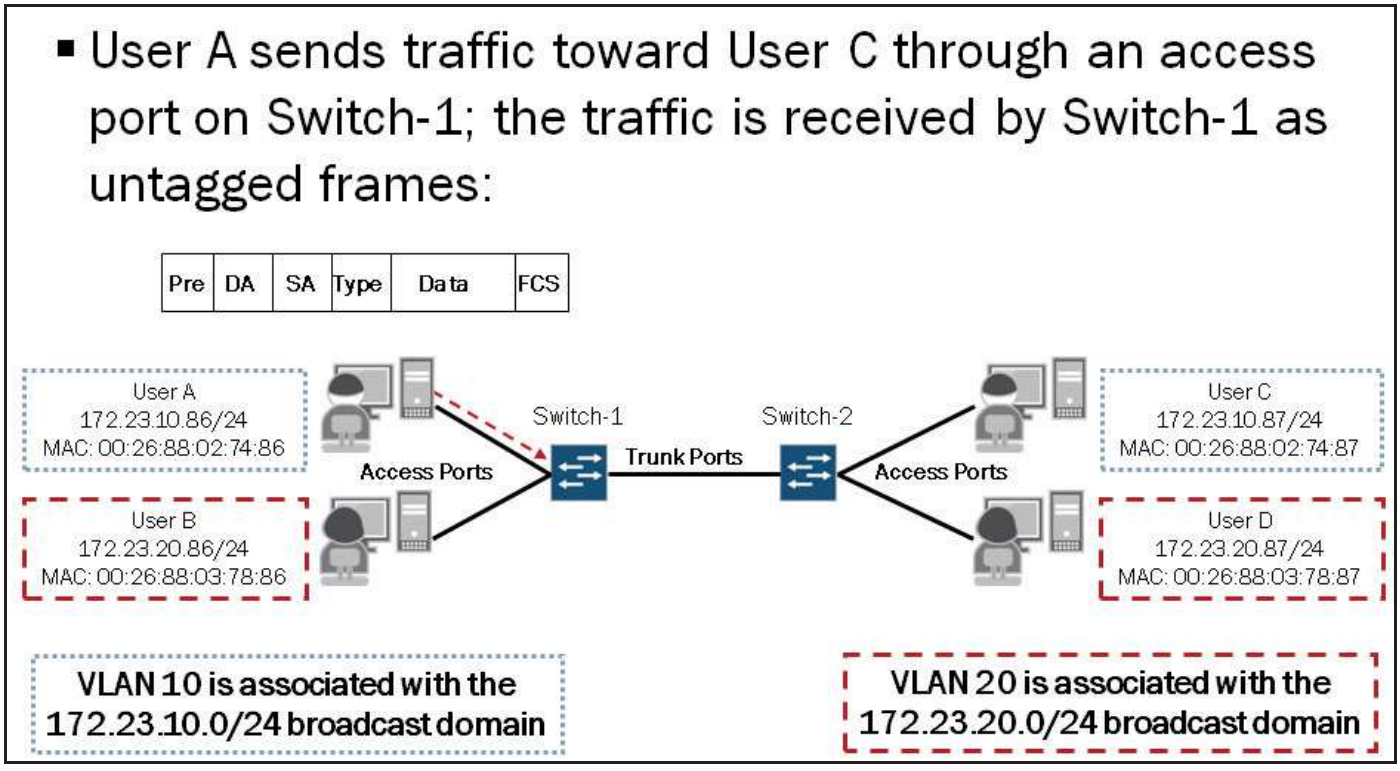
As shown in the illustration on the graphic, access ports typically connect to end-user devices such as computers, IP phones, and printers. Access ports typically belong to a single VLAN and send and receive untagged Ethernet frames. We will discuss the voice VLAN, which is an exception to this operational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



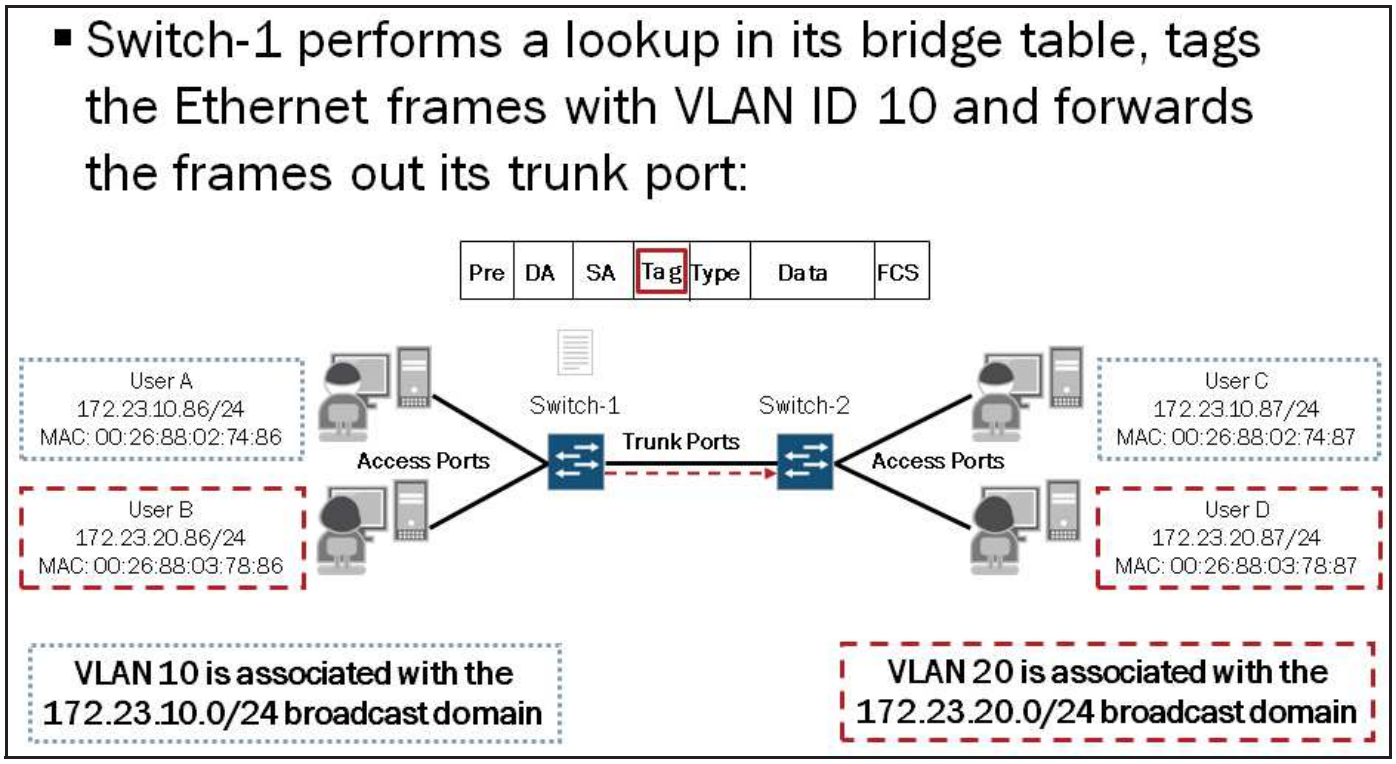
A trunk port typically connects to another switch or to an edge router. Interfaces configured for trunk mode handle traffic for multiple VLANs, multiplexing the traffic for all configured VLANs over the same physical connection, and separating the traffic by tagging it with the appropriate VLAN ID. Trunk 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



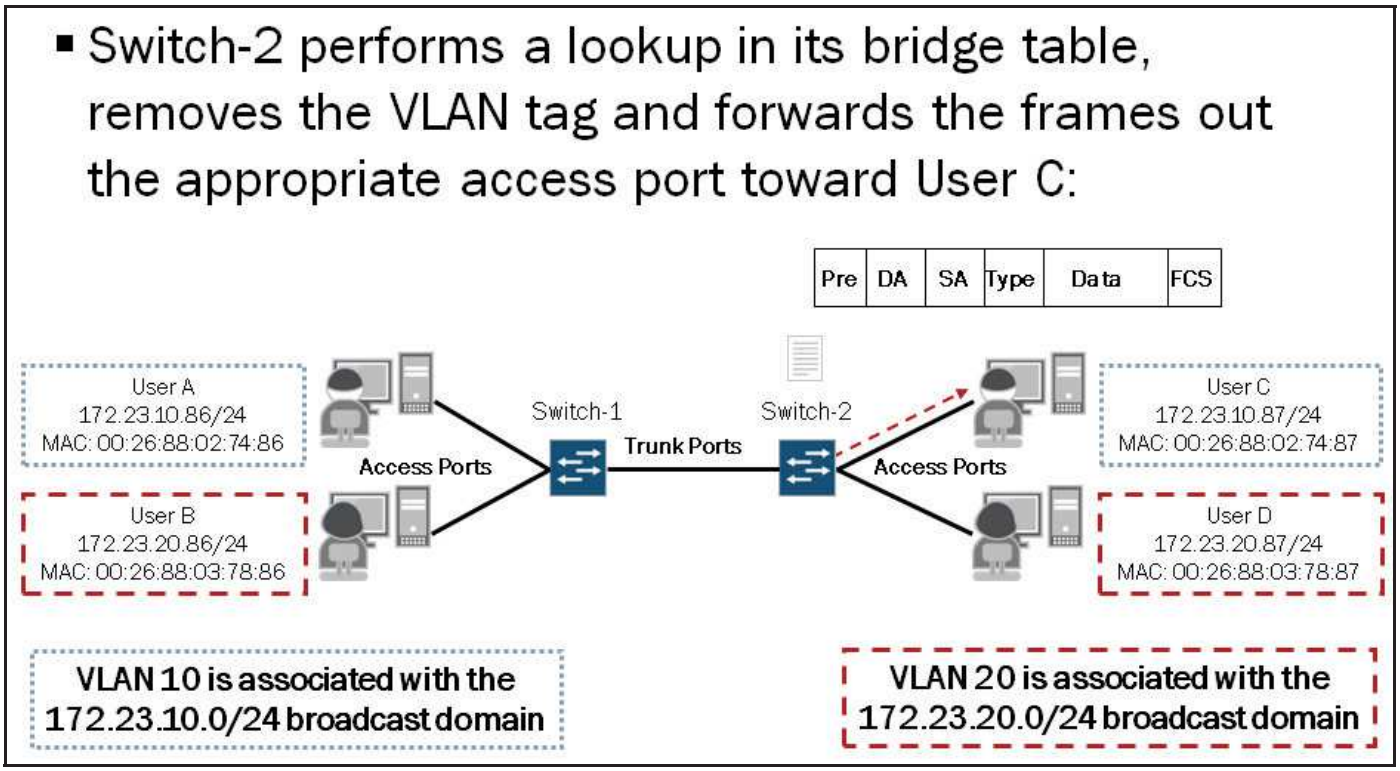
This graphic and the next two graphics illustrate the basic steps involved in sending traffic through a switched network where both access and trunk ports are used. On this graphic we see that User A is sending traffic toward User C through Switch-1 and Switch-2. As the traffic arrives at Switch-1, the 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.

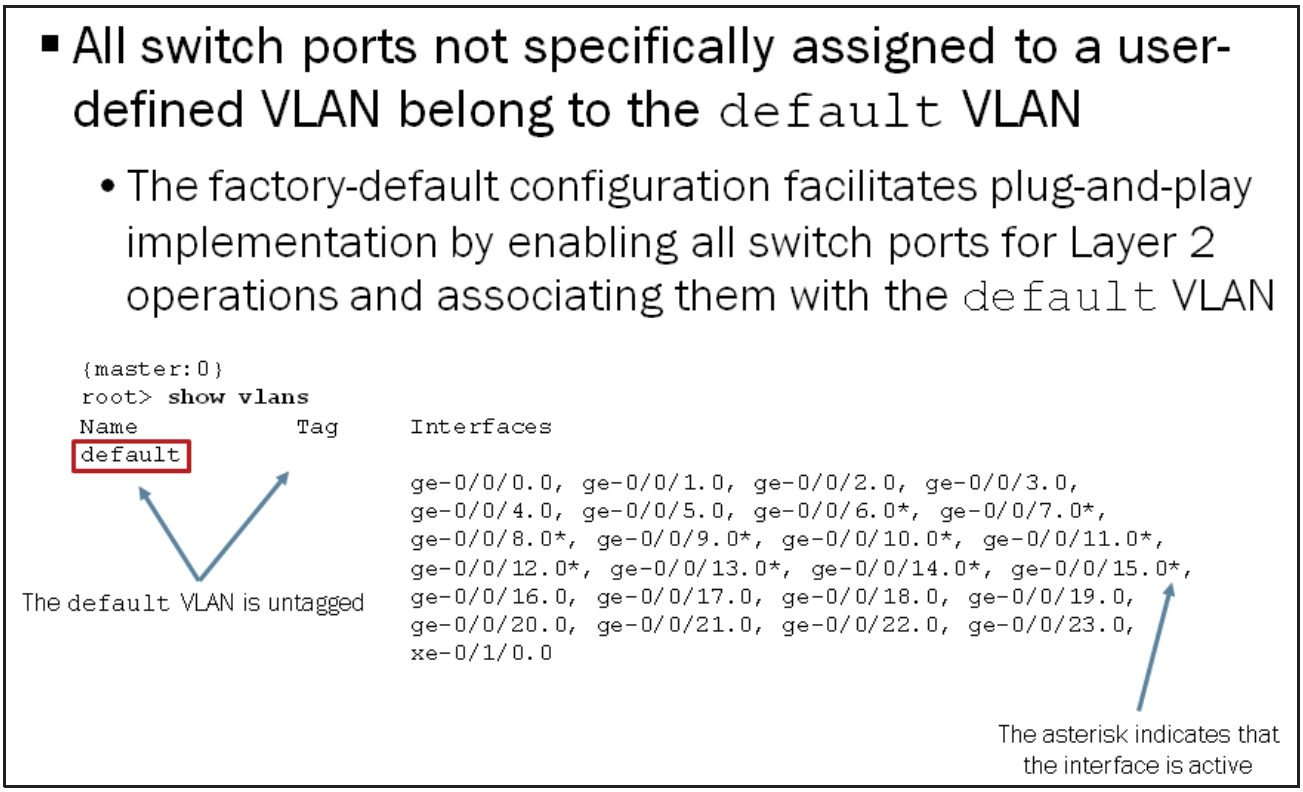
Tagging Traffic Example: Part 3



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.

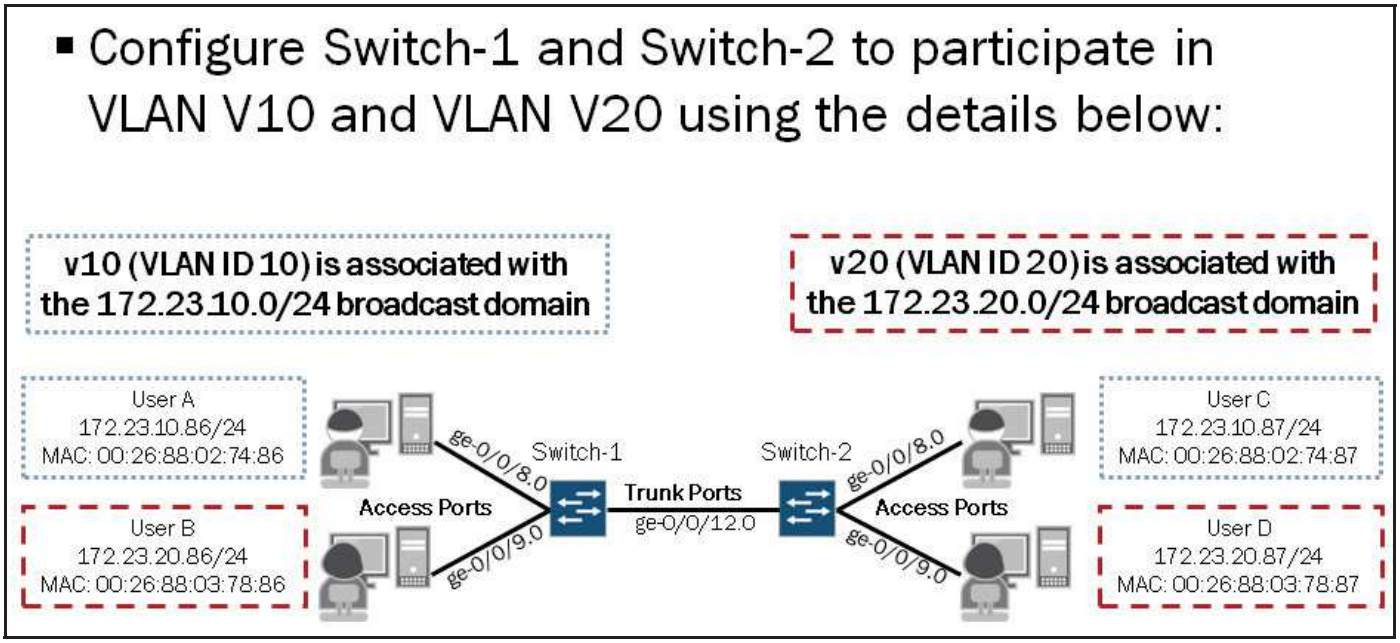
Note that you can make changes to the preconfigured interfaces by manually altering the configuration file directly on the device or automatically by retrieving a configuration file across the network from a Dynamic Host Configuration Protocol (DHCP) server using EZ Touchless Provisioning. This feature 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
```

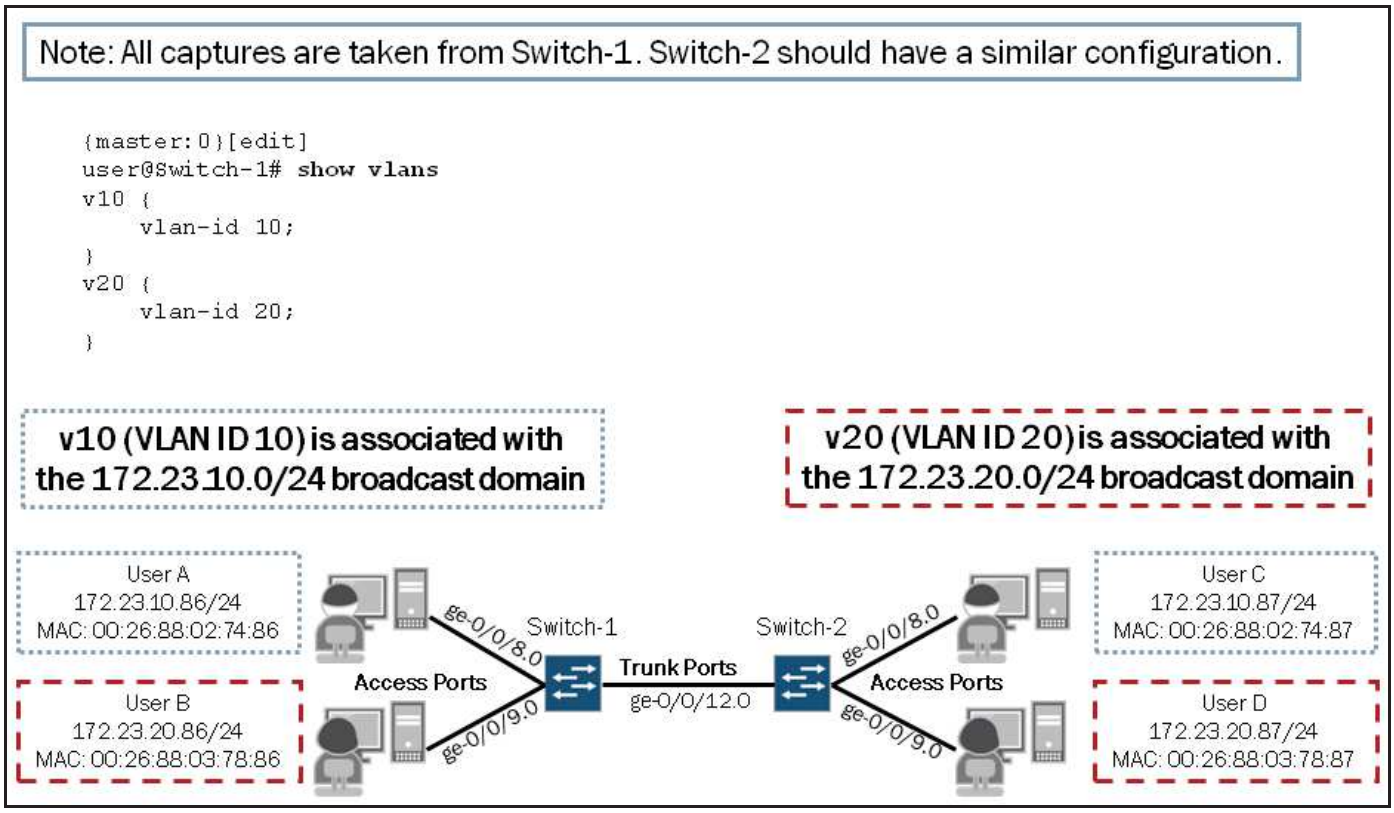
```
Name            Tag     Interfaces
default         100
                        ge-0/0/0.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
                        ge-0/0/4.0, ge-0/0/5.0, ge-0/0/6.0*, ge-0/0/7.0*,
                        ge-0/0/8.0*, ge-0/0/9.0*, ge-0/0/10.0*, ge-0/0/11.0*,
                        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

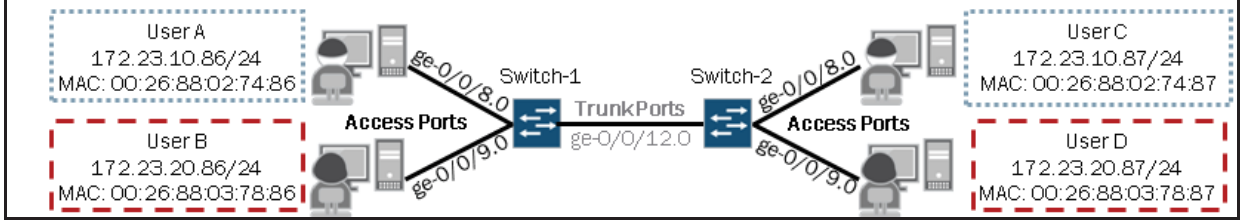


This graphic shows the required VLAN definitions for our case study. Note that additional configuration options are available under the `[edit vlans]` hierarchy level. We cover some of the listed configuration options in subsequent sections and chapters:

```
{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;
        }
    }
}
```

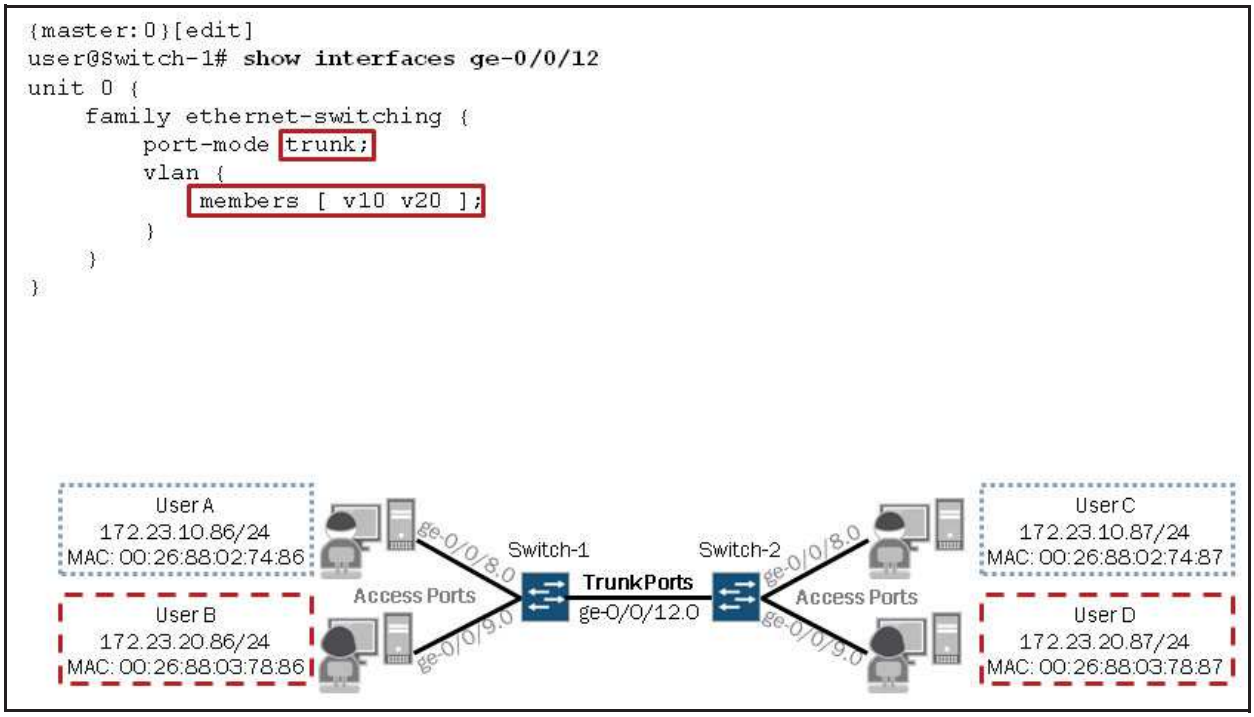


The sample configuration shown on the graphic illustrates one method you can use to associate an interface with a VLAN. Note that the illustrated method is the same method used by the J-Web user interface. Because Layer 2 interfaces default to access mode, including the **`port-mode access`** statement 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



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;
        }
    }
}
```

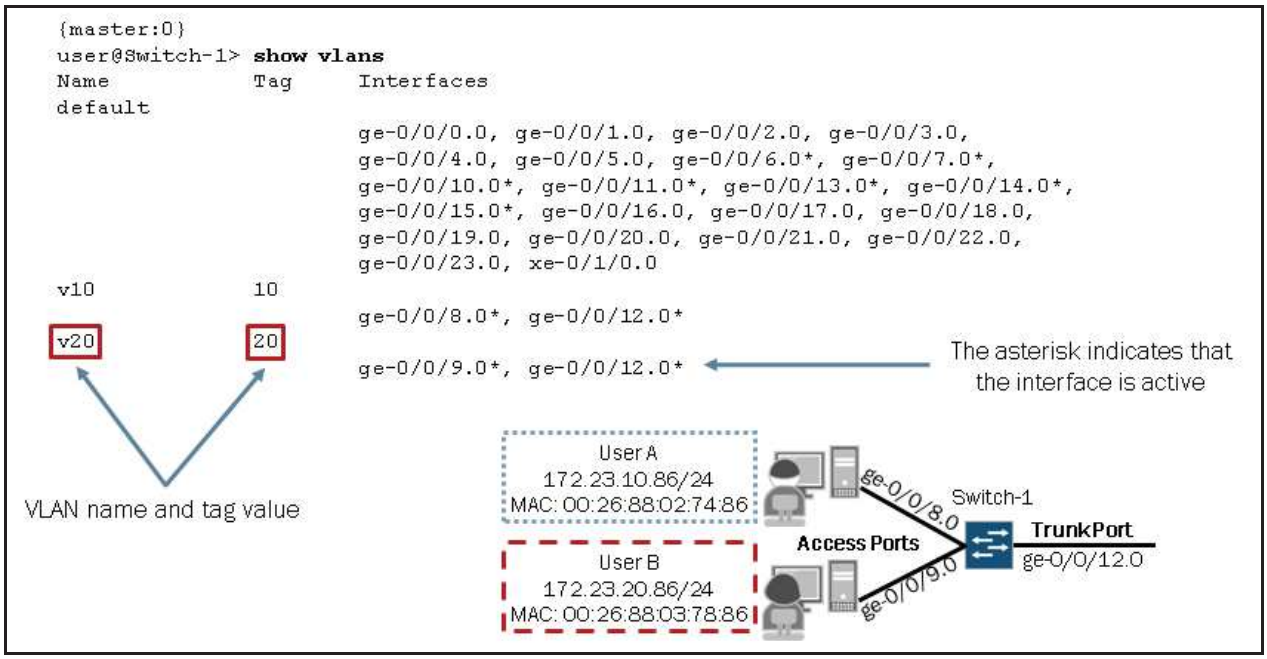
As noted earlier, you can optionally associate interfaces with VLANs under the `[edit vlans]` hierarchy. The following configuration shows this alternative method for a trunk port.

```
{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

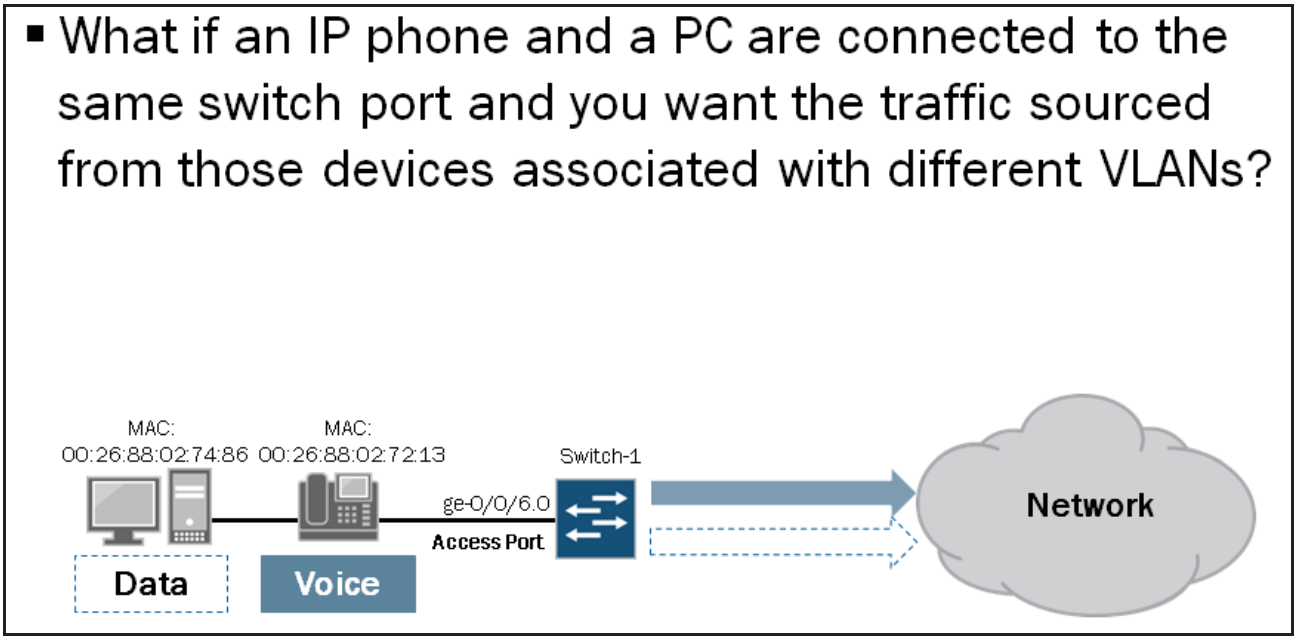


You can use the **show vlans** command to verify VLAN assignments and other details. Optionally you can filter the output or increase the amount of detail generated by adding options to the **show vlans** command. The available options are shown in the following output:

```
{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
```

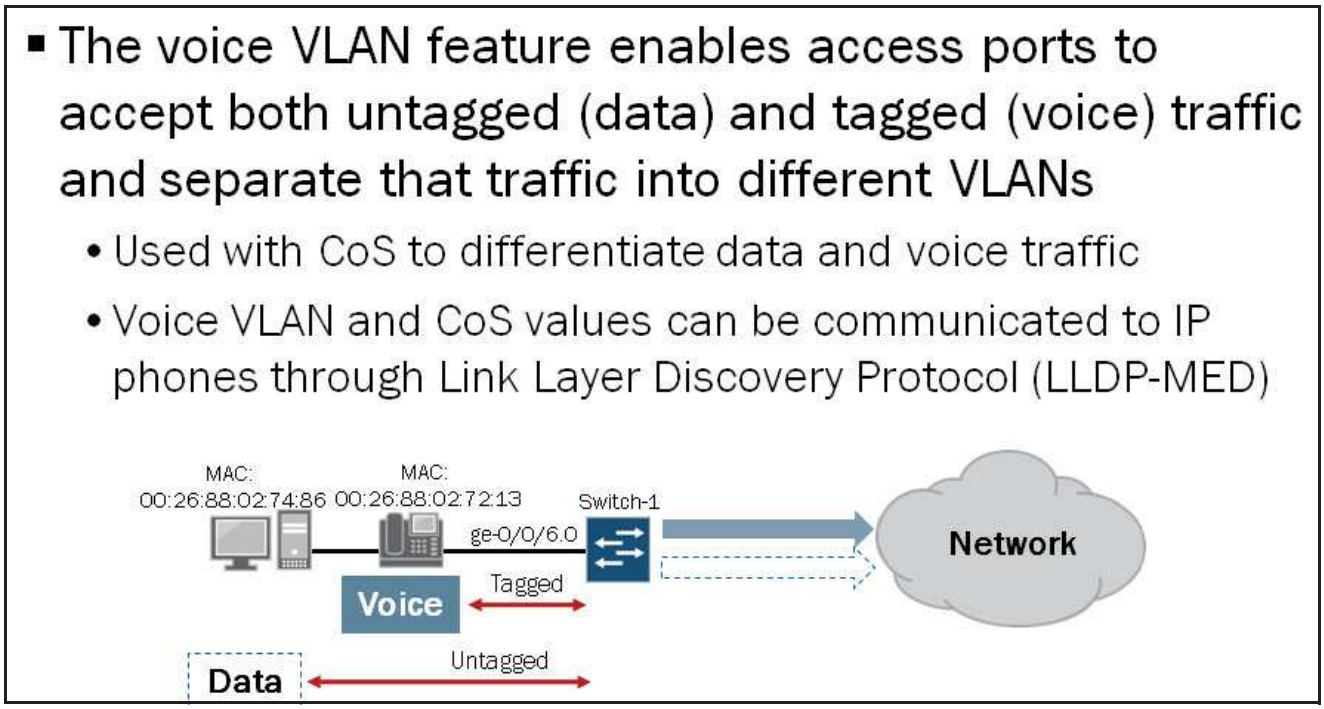
What If...?



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

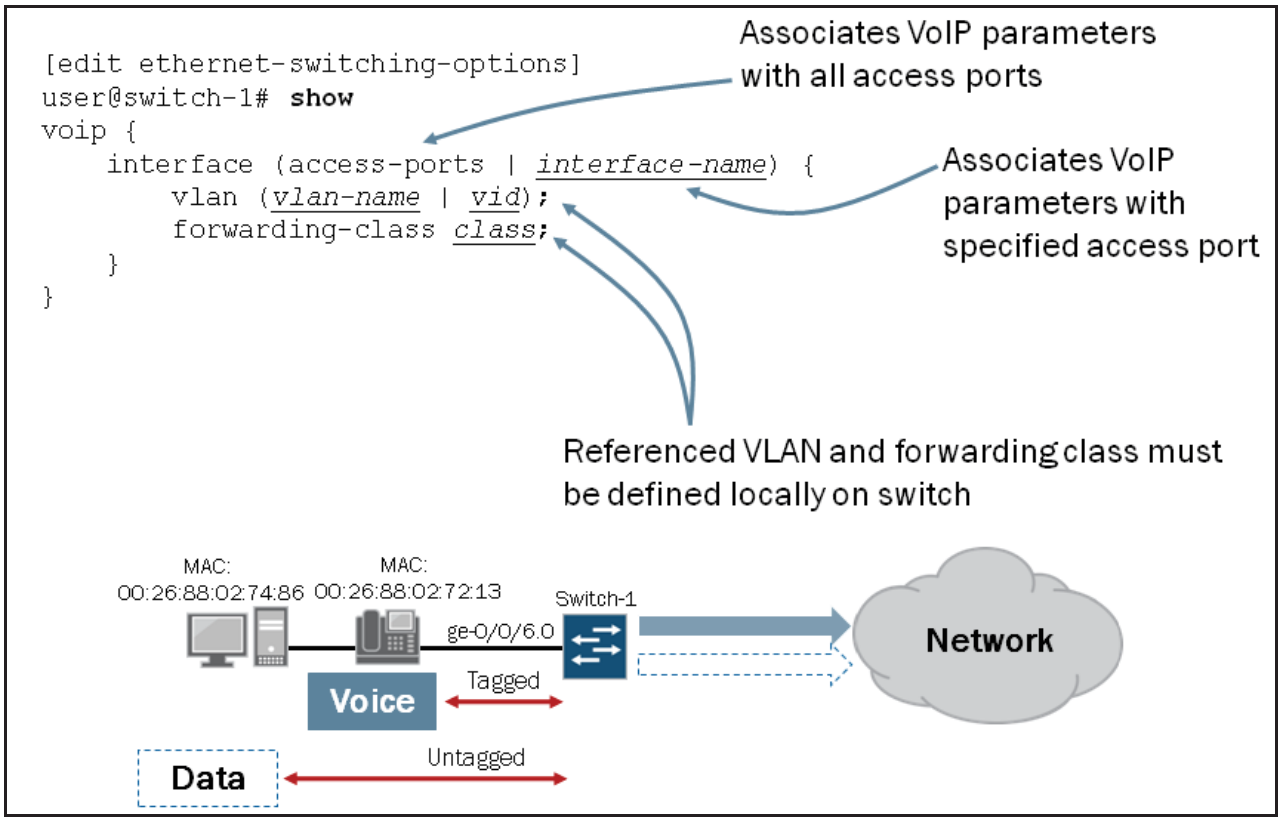


Typically, network administrators choose to treat VoIP traffic differently from user data traffic. To treat these types of traffic differently, you must be able to separate common user data traffic from voice traffic. The voice VLAN feature is used for this

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

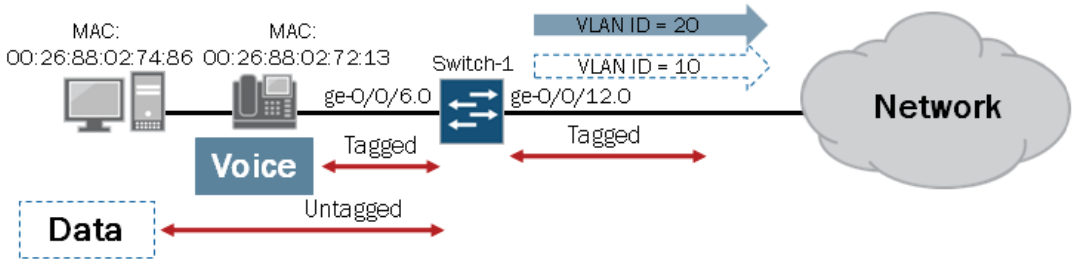


This graphic illustrates the basic hierarchy structure along with the available configuration options associated with the voice VLAN feature.

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 ];
        }
    }
}
```

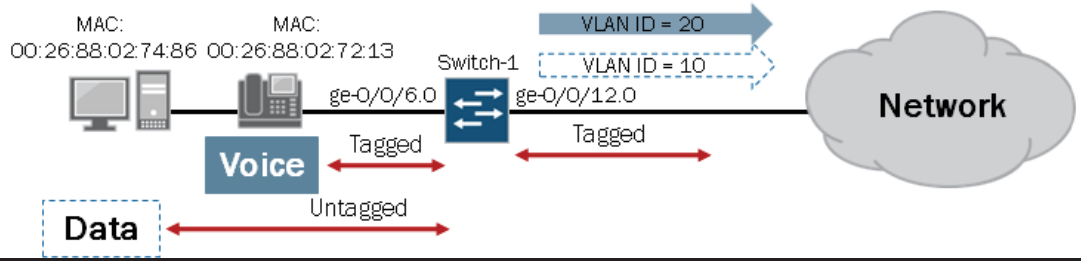


This graphic provides a more complete configuration example based on our sample topology which is also shown on this graphic.

Monitoring the Voice VLAN

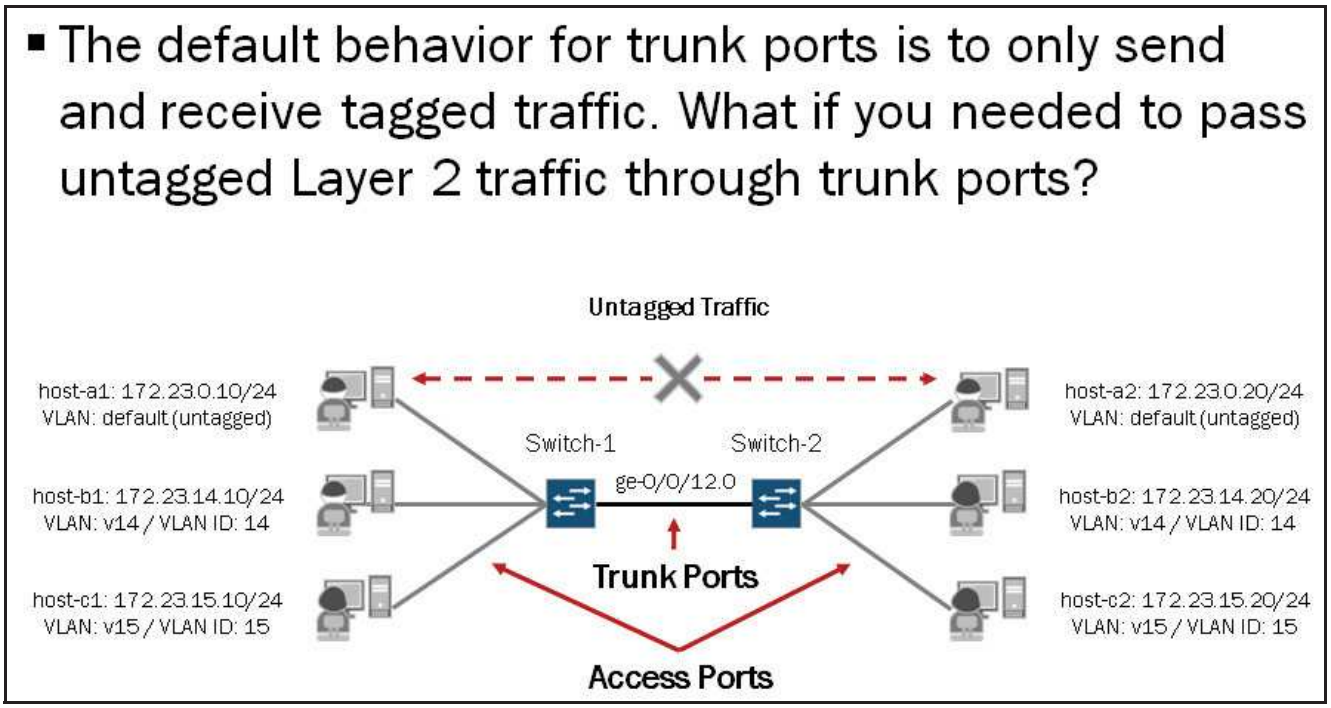
```
{master:0}
user@Switch-1> show vlans sort-by tag
Name            Tag     Interfaces
default
                        ge-0/0/0.0, ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0,
                        ge-0/0/4.0, ge-0/0/5.0, ge-0/0/7.0*, ge-0/0/8.0*,
                        ge-0/0/9.0*, ge-0/0/10.0*, ge-0/0/11.0*, 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...?

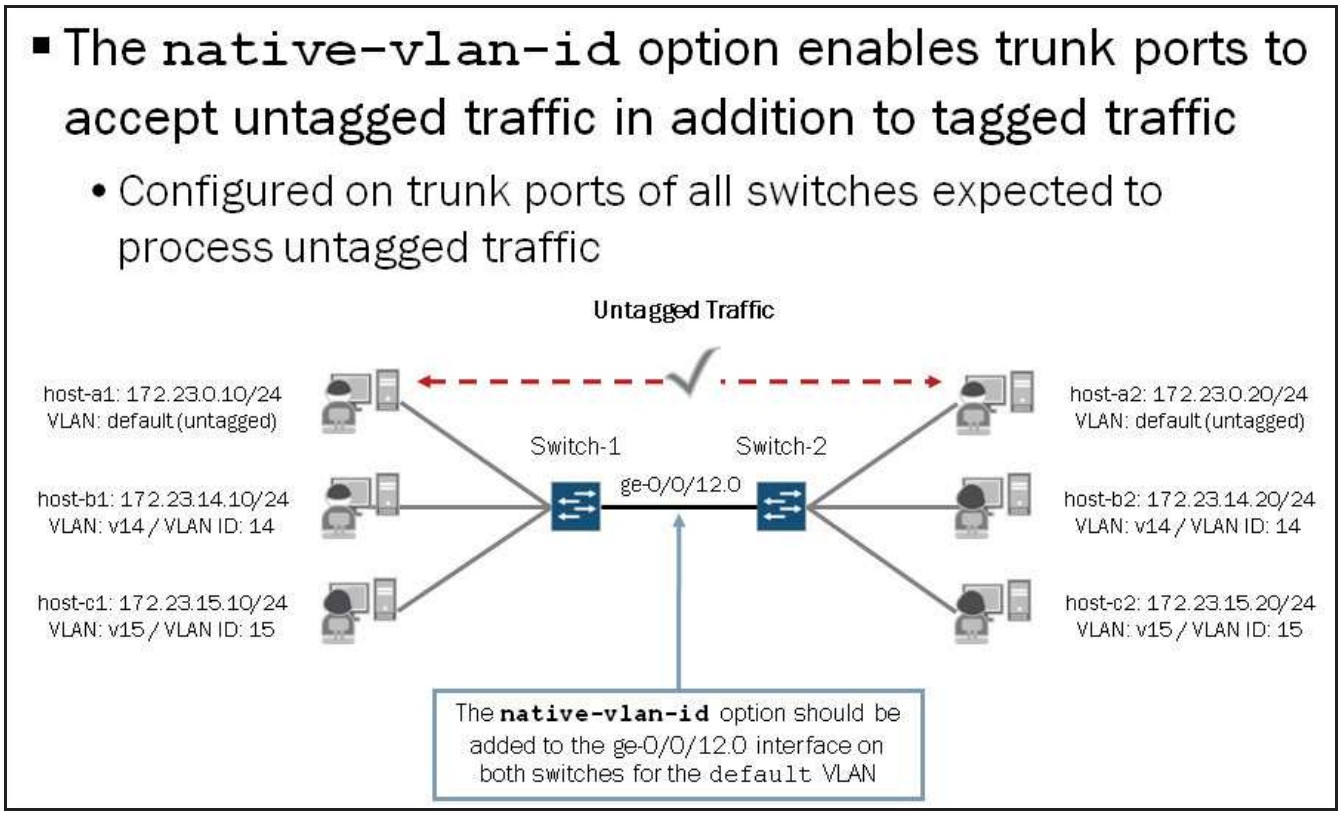


The default behavior on EX Series switches for trunk ports is to only send and receive tagged traffic. This means that you cannot assign an untagged VLAN, such as the default VLAN, to a trunk port. The configuration will not commit as shown here:

```
{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
```

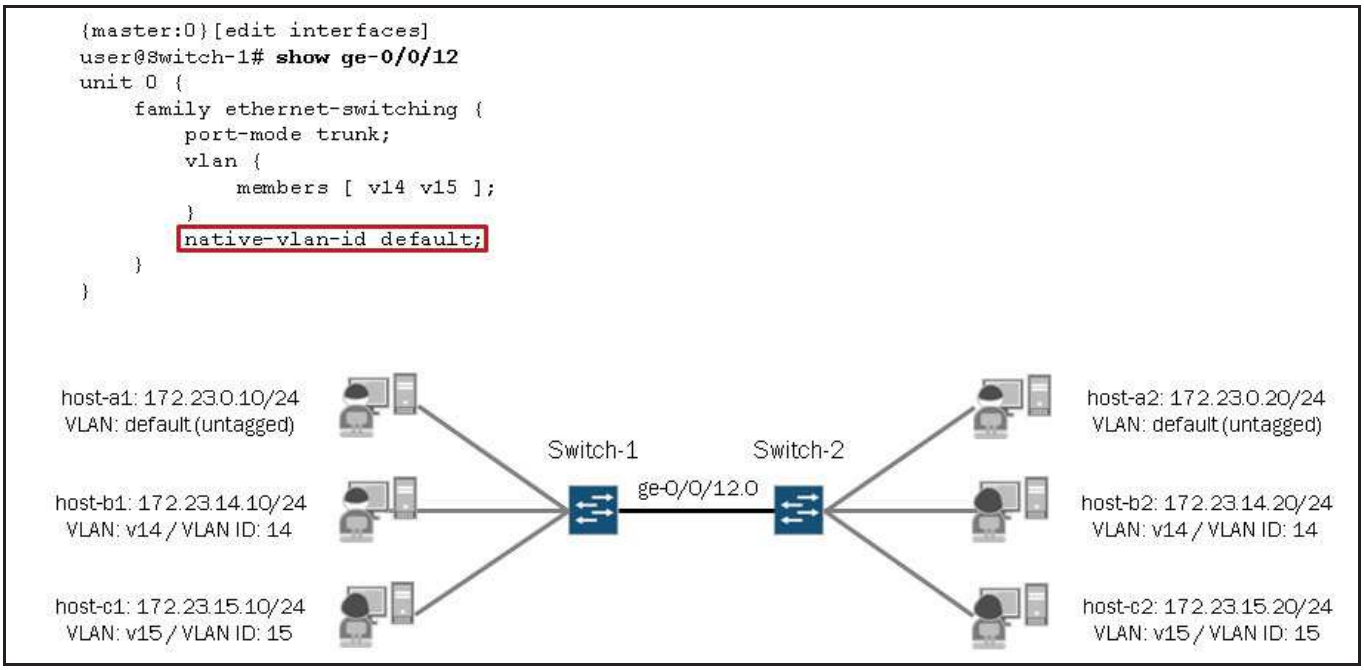
So, what can you do if you needed to pass untagged Layer 2 traffic through trunk ports? You must use the **native-vlan-id** configuration option. We cover the **native-vlan-id** option throughout the remainder of this section.

The `native-vlan-id` Option



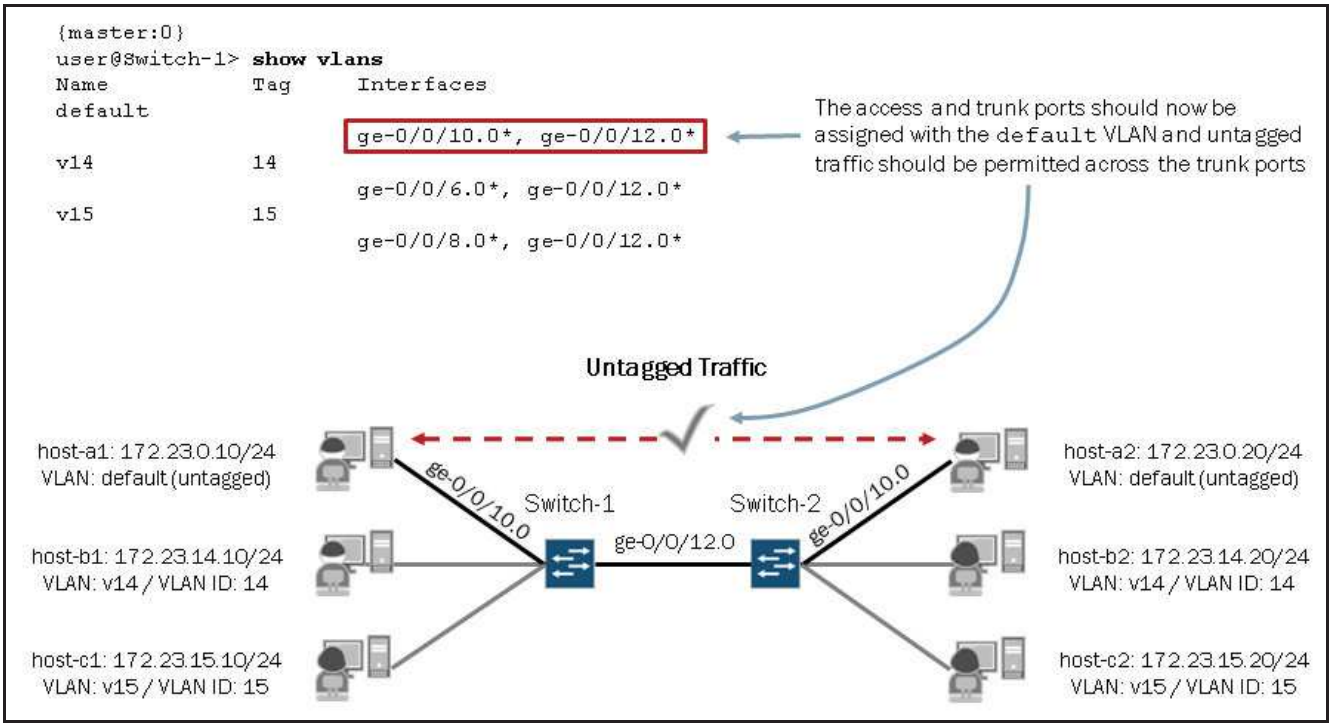
As previously mentioned, a trunk port typically connects one switch to another switch or to an edge router. Interfaces configured for trunk mode handle traffic for multiple VLANs, multiplexing the traffic for all configured VLANs over the same physical connection, and separating the traffic by tagging 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



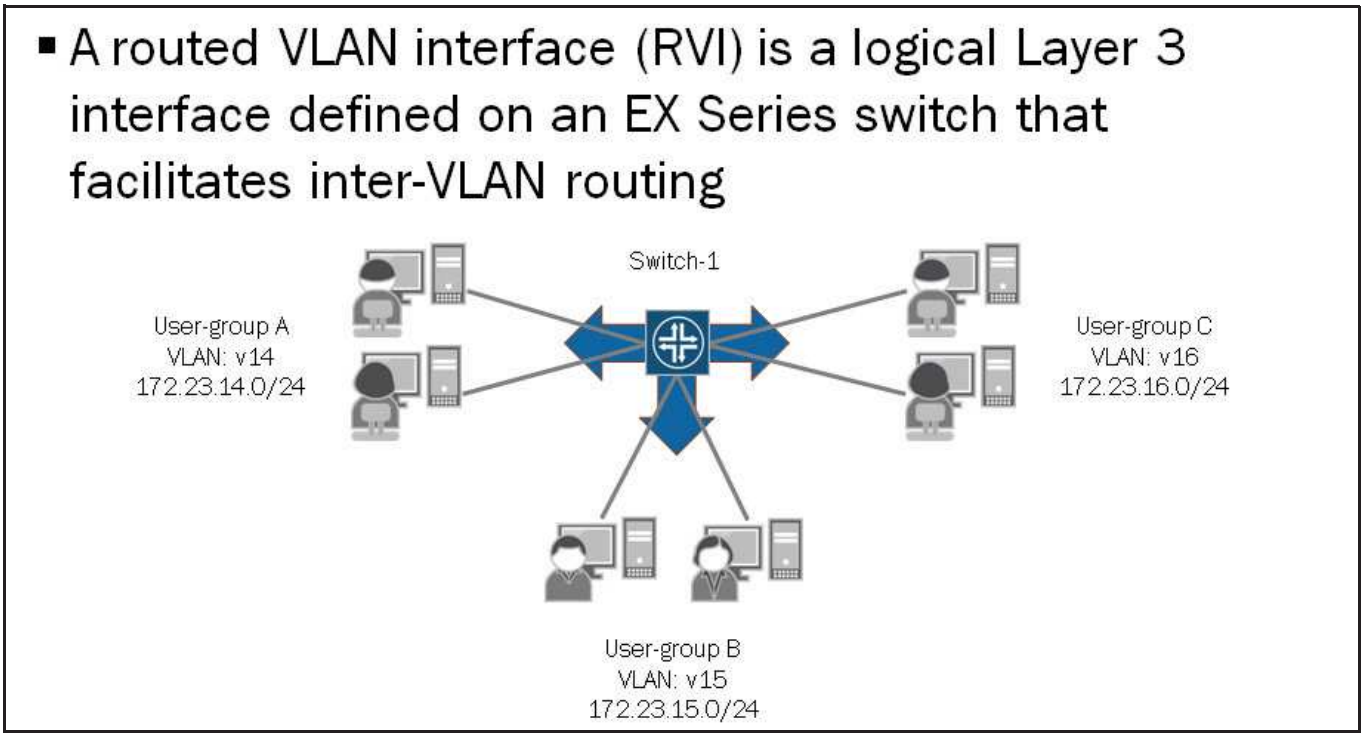
This graphic provides a configuration example using the **`native-vlan-id`** option for the trunk ports that connect Switch-1 and Switch-2. With this configuration, the ge-0/0/12 interfaces are configured as a trunk ports and are able to carry tagged traffic for the `v14` and `v15` VLANs as well as untagged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

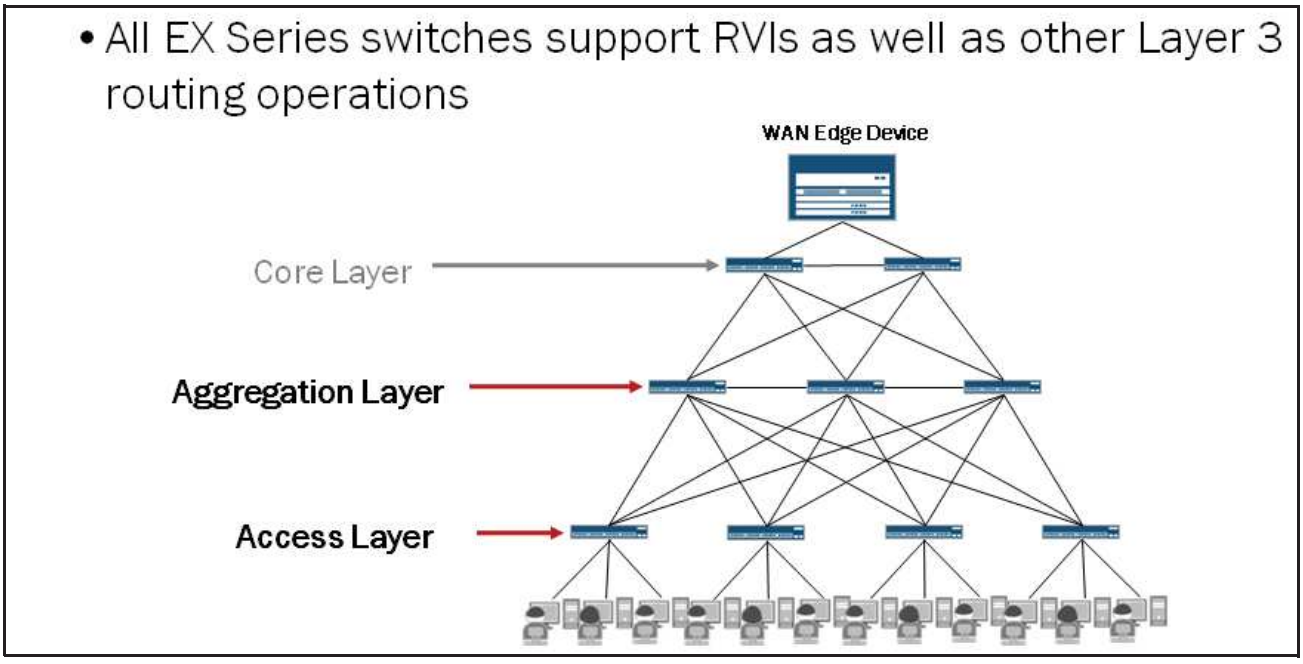
What Is an RVI?



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.

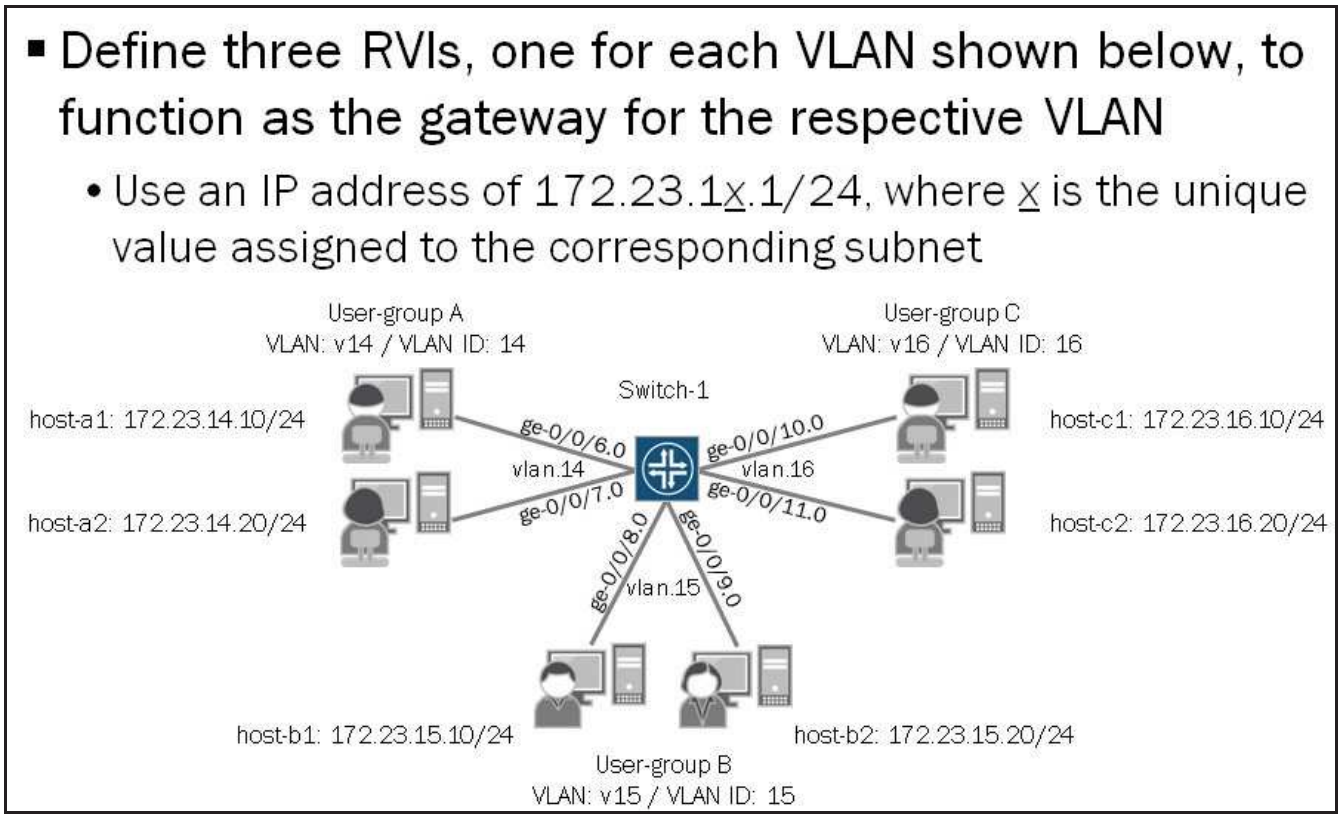
The following graphics provide a configuration and monitoring example for an RVI.

Implementing RVIs



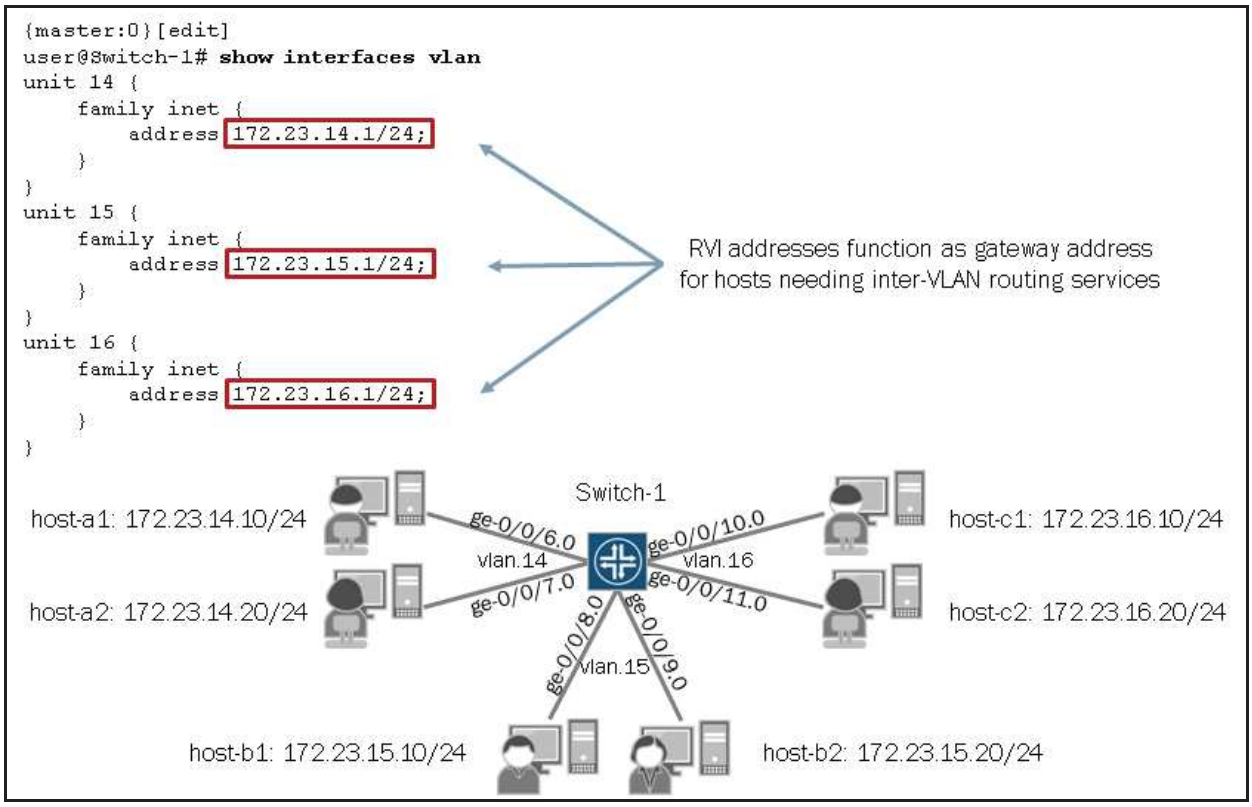
As indicated on the graphic, RVIs are typically implemented in either the aggregation layer or the access layer, depending on the network design and implementation. All EX Series switches support RVIs as well as other Layer 3 routing operations. Check your platform specific documentation for support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

Case Study: Topology and Objectives



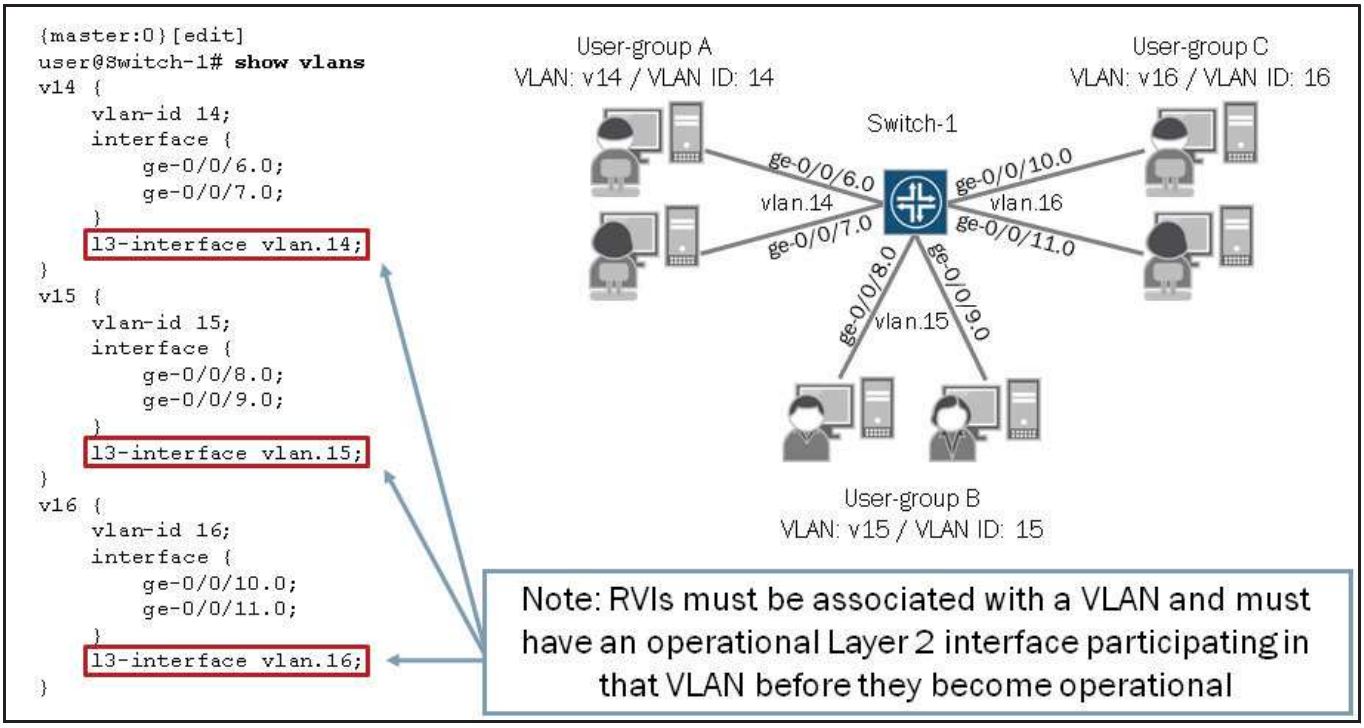
The graphic displays the topology and objectives for our case study.

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.

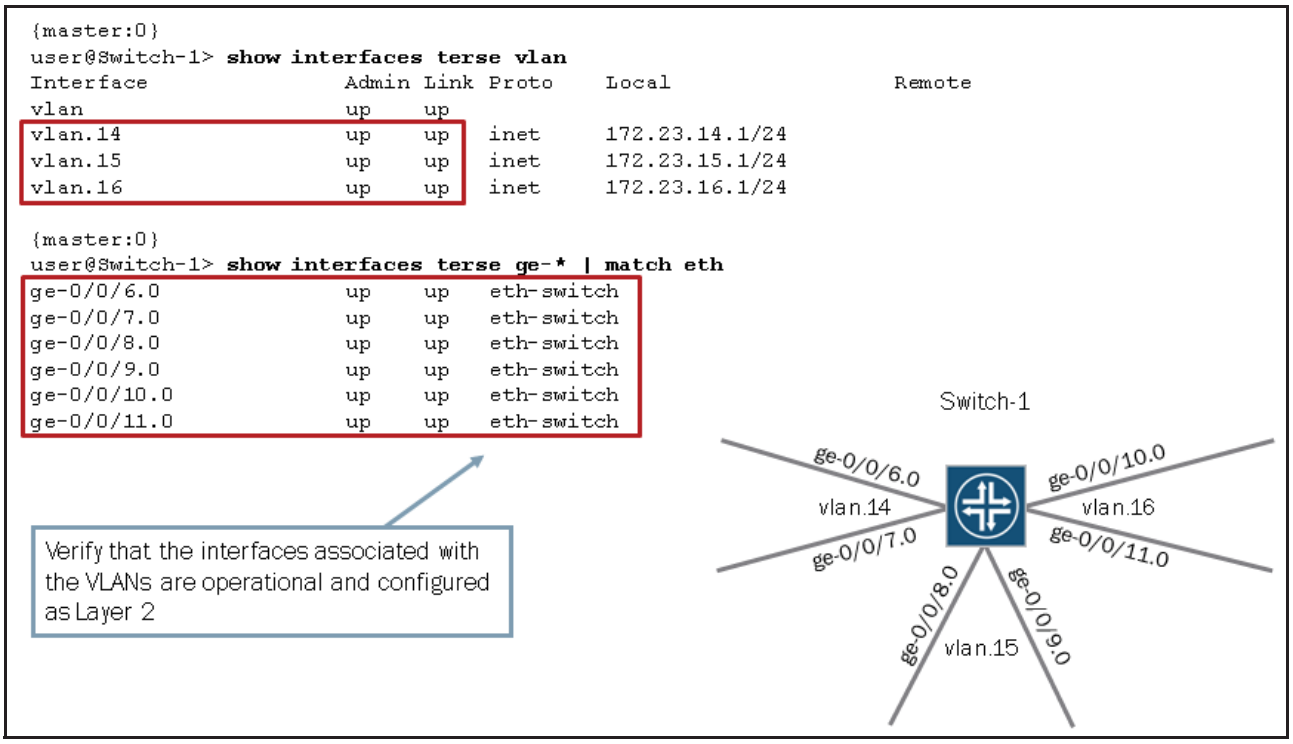
Associating RVIs with VLANs



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

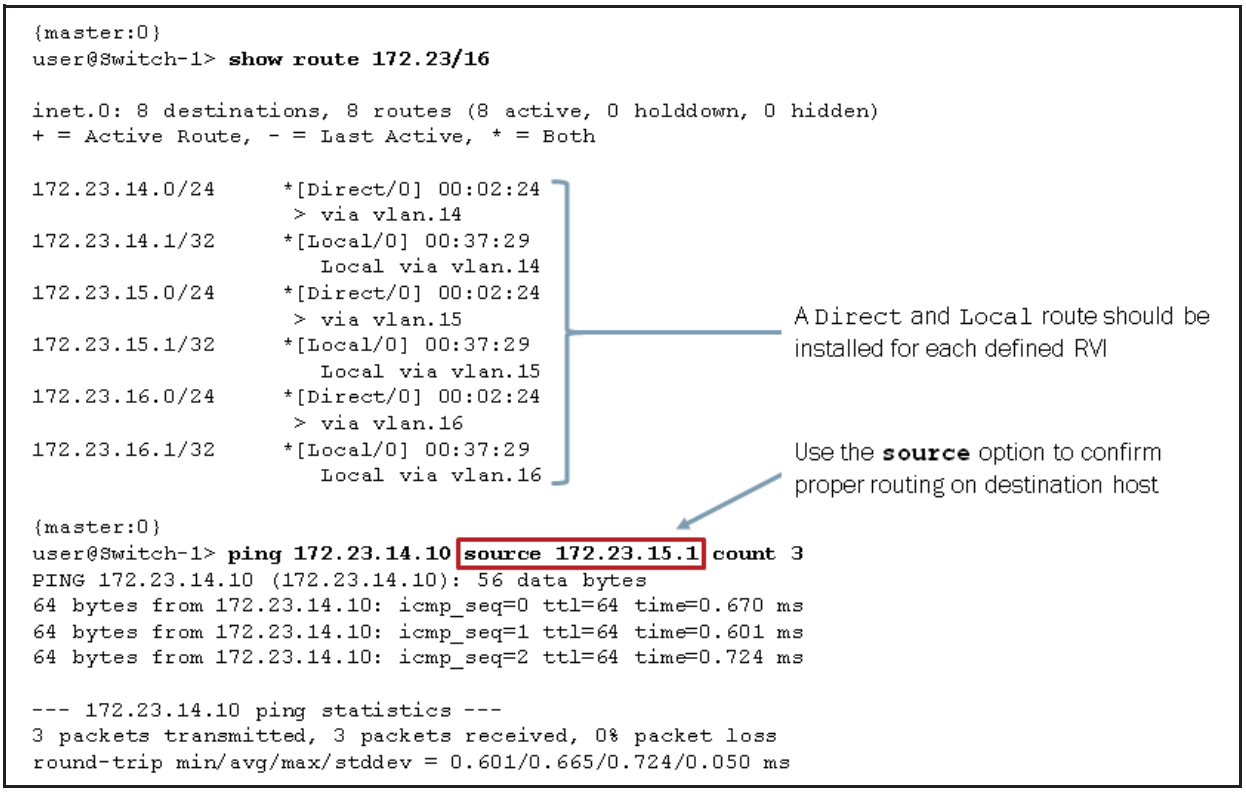
routing cannot occur without this RVI to VLAN association. As mentioned on the graphic, an RVI must be associated with a VLAN and that VLAN must have at least one operational Layer 2 interface before the RVI becomes operational.

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.

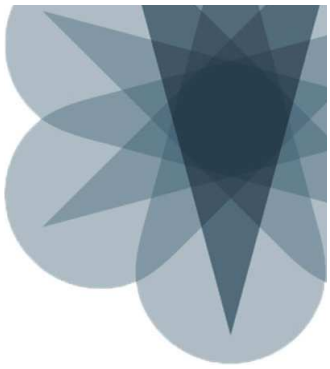
3.

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.

JUNIPER NETWORKS



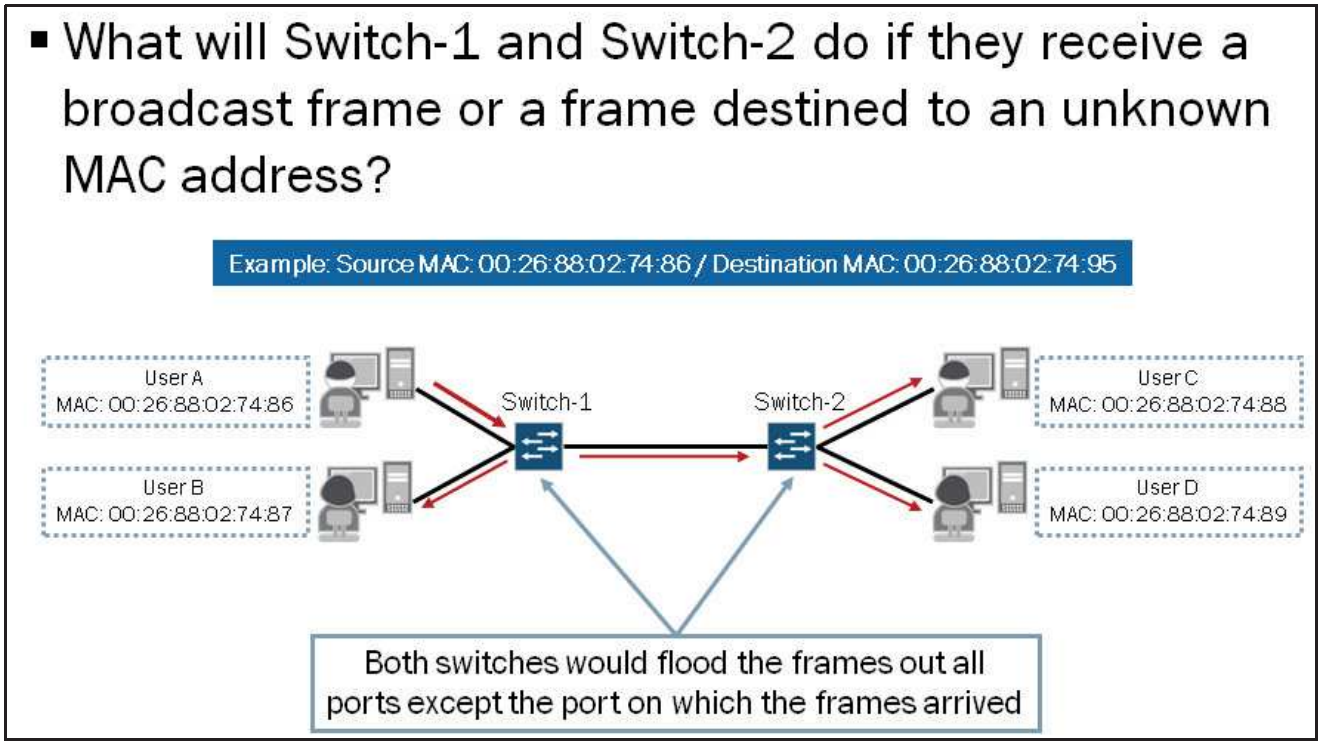
JNCIS-ENT Routing Study Guide

Chapter 3: Spanning Tree

This Chapter Discusses:

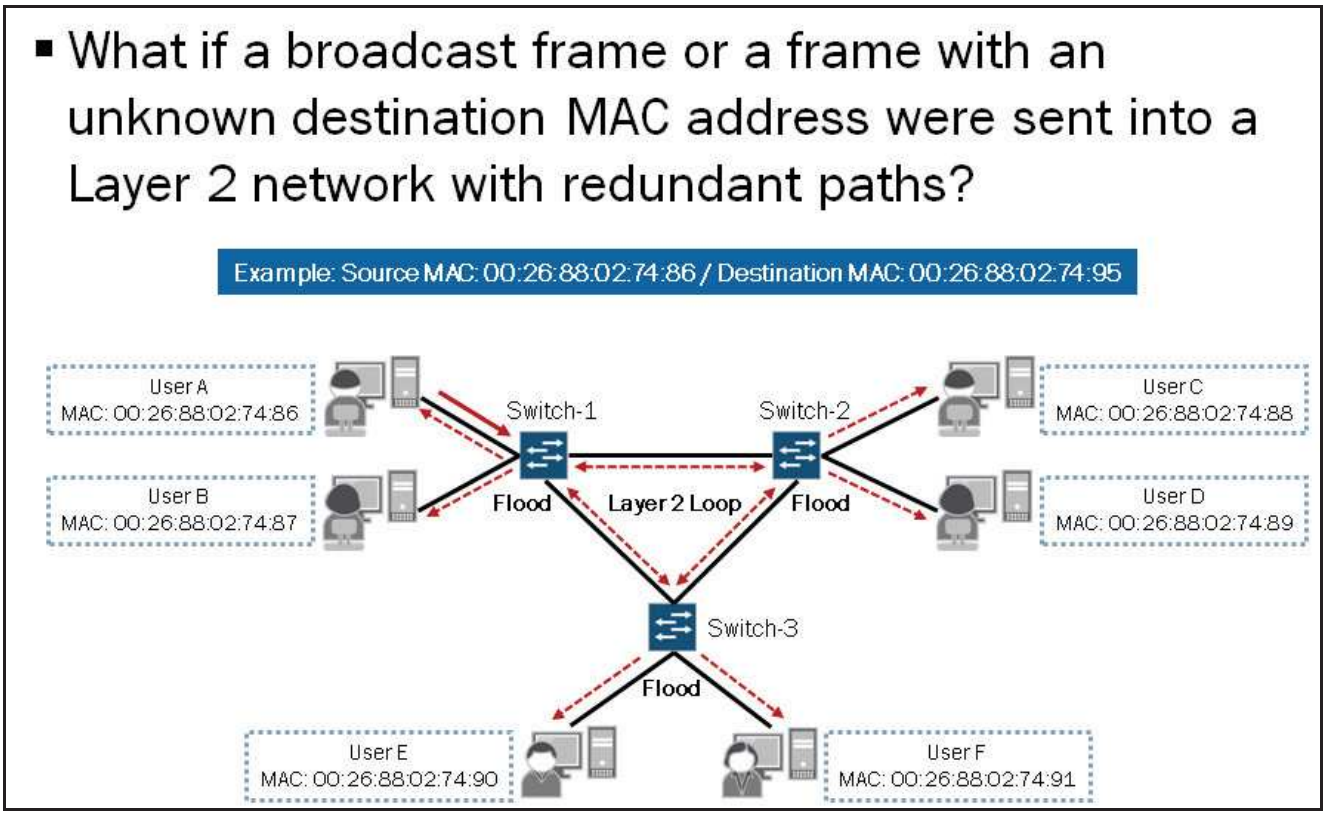
- Instances when a spanning tree is required;
- Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP) operations;
- The advantages of using RSTP over STP;
- The configuration and monitoring of STP and RSTP;
- Bridge protocol data unit (BPDU), loop, and root protection features; and
- The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

Test Your Knowledge



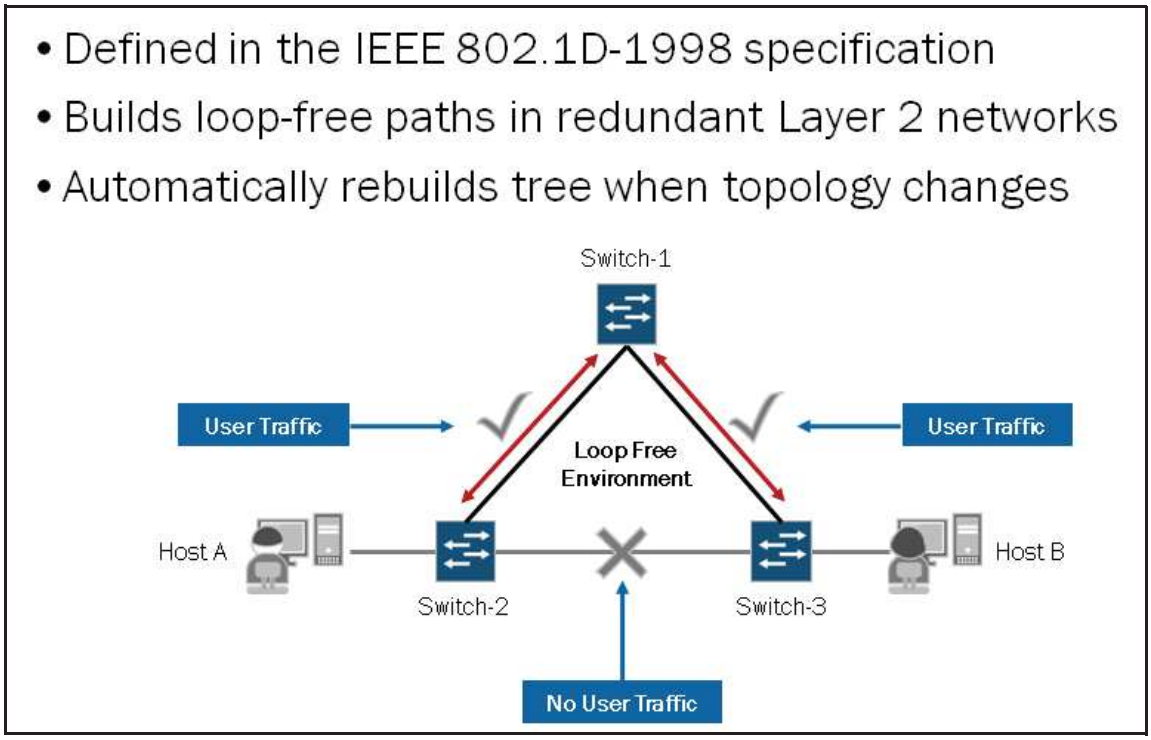
This graphic serves as a review of a previously covered concept. The graphic illustrates the expected behavior when a switch receives a broadcast frame or a frame destined to an unknown MAC address. You can see in the example that both Switch-1 and Switch-2 flood the frame out all interfaces except the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

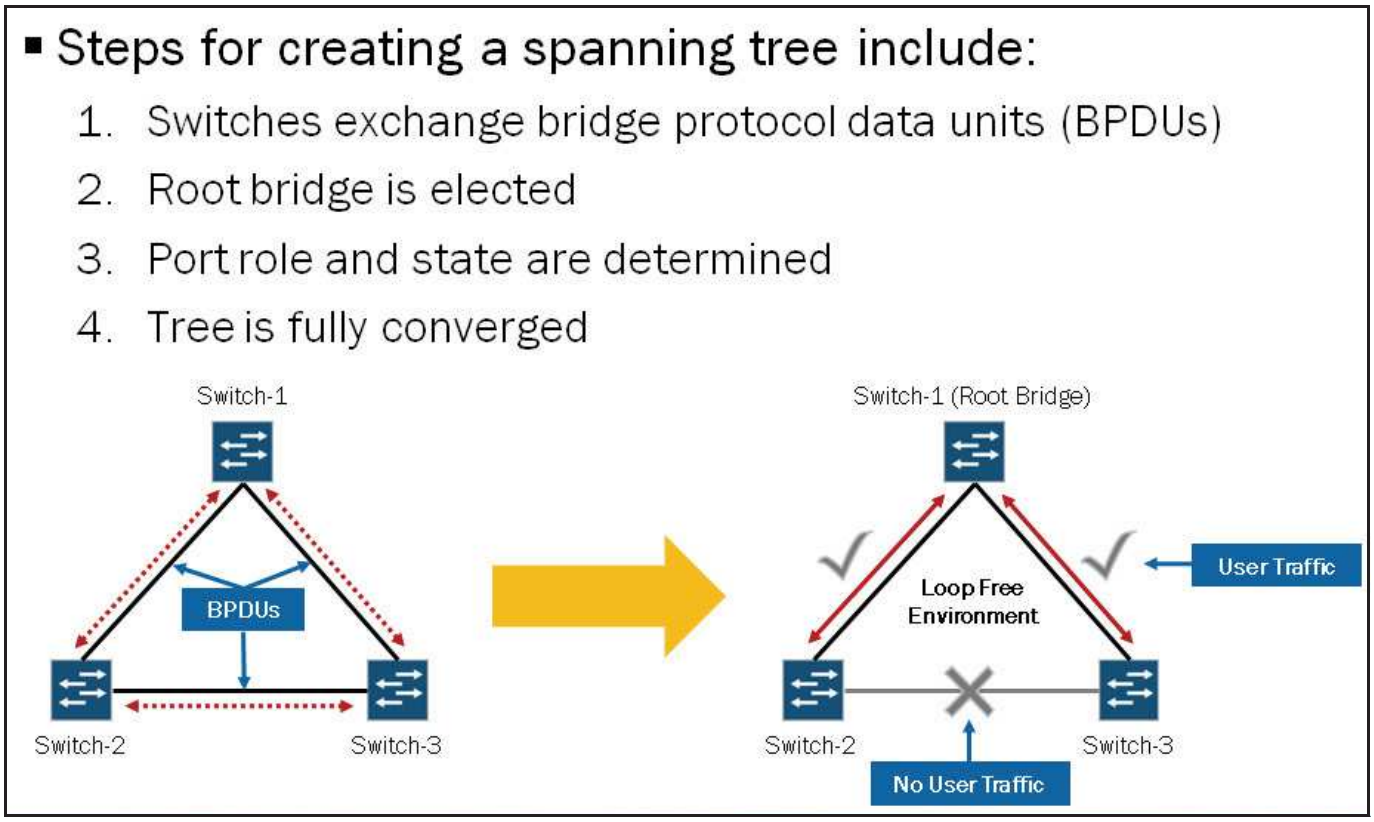


STP is defined in the Institute of Electrical and Electronics Engineers (IEEE) 802.1D 1998 specification. STP is a simple Layer 2 protocol that prevents loops and calculates the best path through a switched network that contains redundant paths. STP is highly recommended in any Layer 2 network environment 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

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

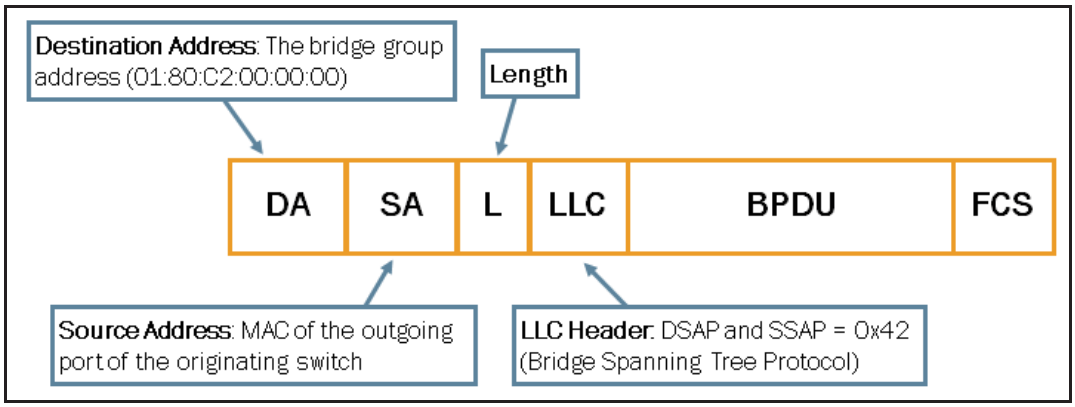
All switches participating on a common network segment must determine which switch offers the least-cost path from the network segment to the root bridge. The switch with the best path becomes the *designated bridge* for the LAN segment, and the port connecting this switch to the network segment becomes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

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



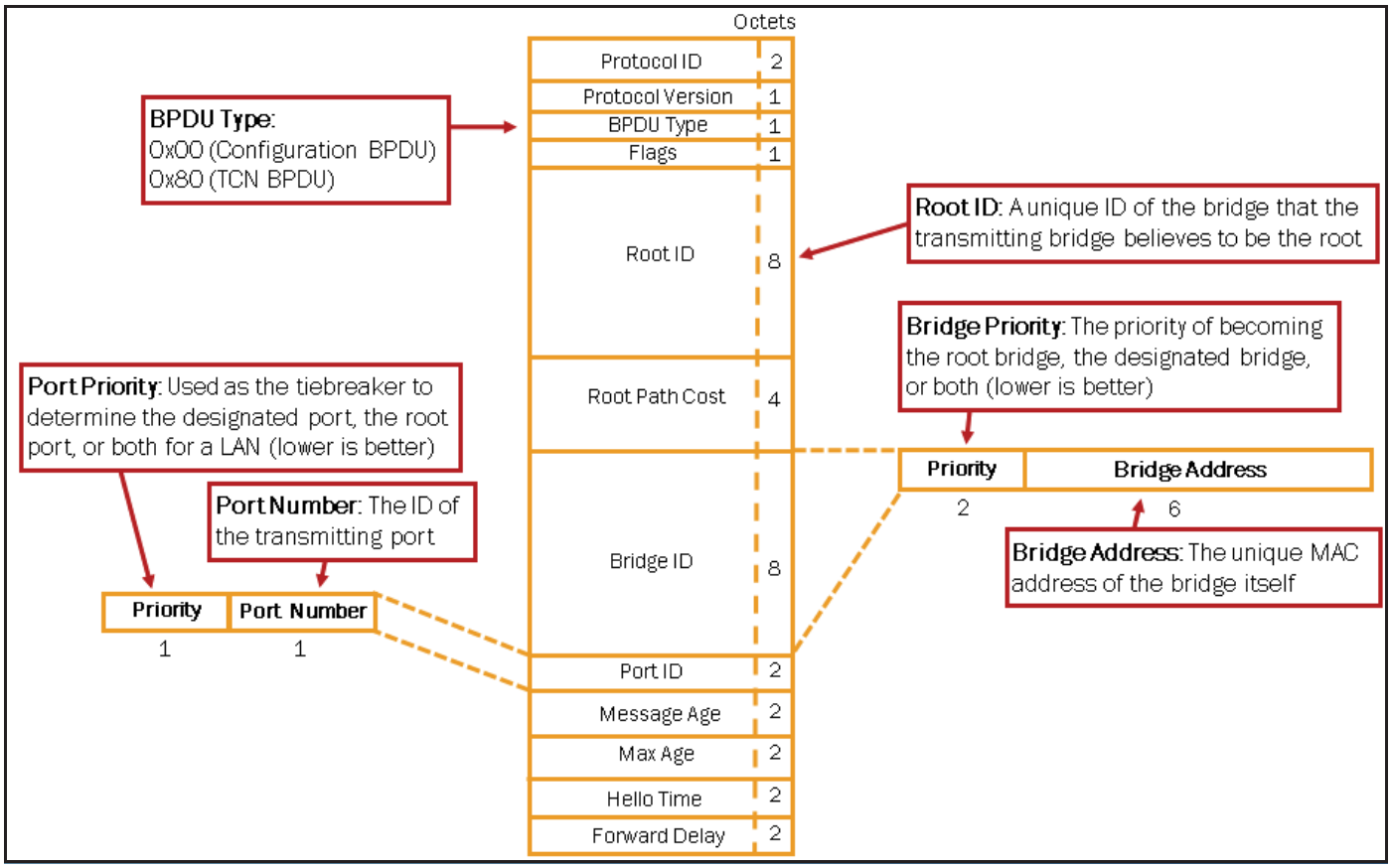
The graphic shows the Ethernet frame format of an STP BPDU. Notice that the Ethernet frame does not contain any 802.1Q-type VLAN tagging. The source address of the frame is the MAC address of the outgoing port of the sending switch. The destination address is the multicast MAC address that is reserved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



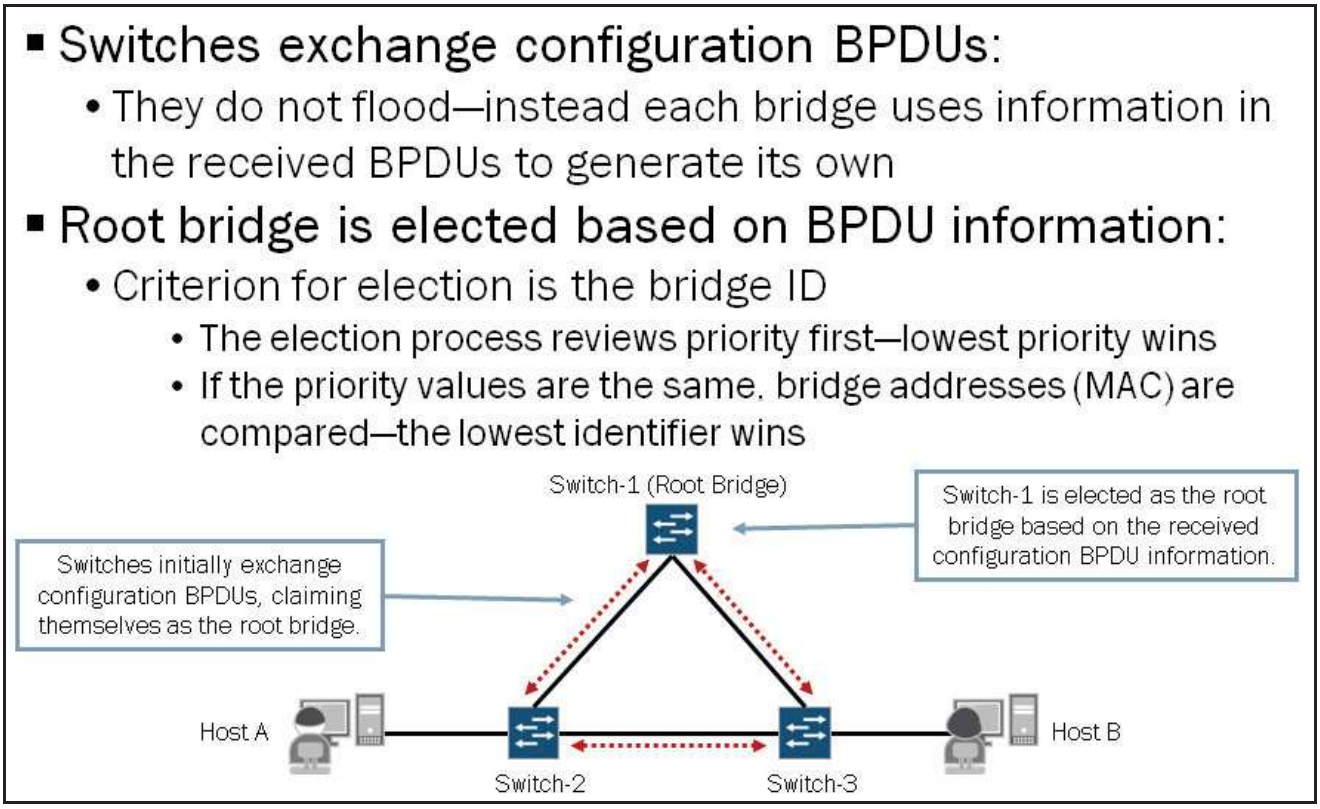
When an STP network is first turned up, all participating bridges send out configuration BPDUs to advertise themselves as candidates for the root bridge. Each bridge uses the received BPDUs to help build the spanning tree and elect the root bridge, root ports, and designated ports for the network. Once 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.

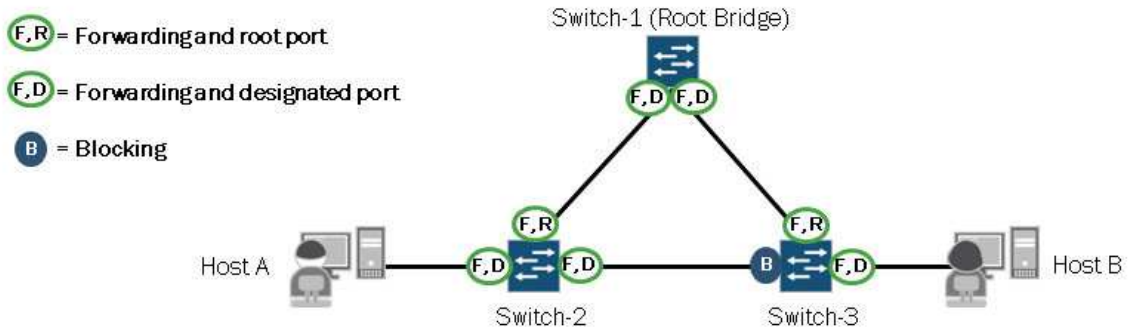
Root Bridge Election

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

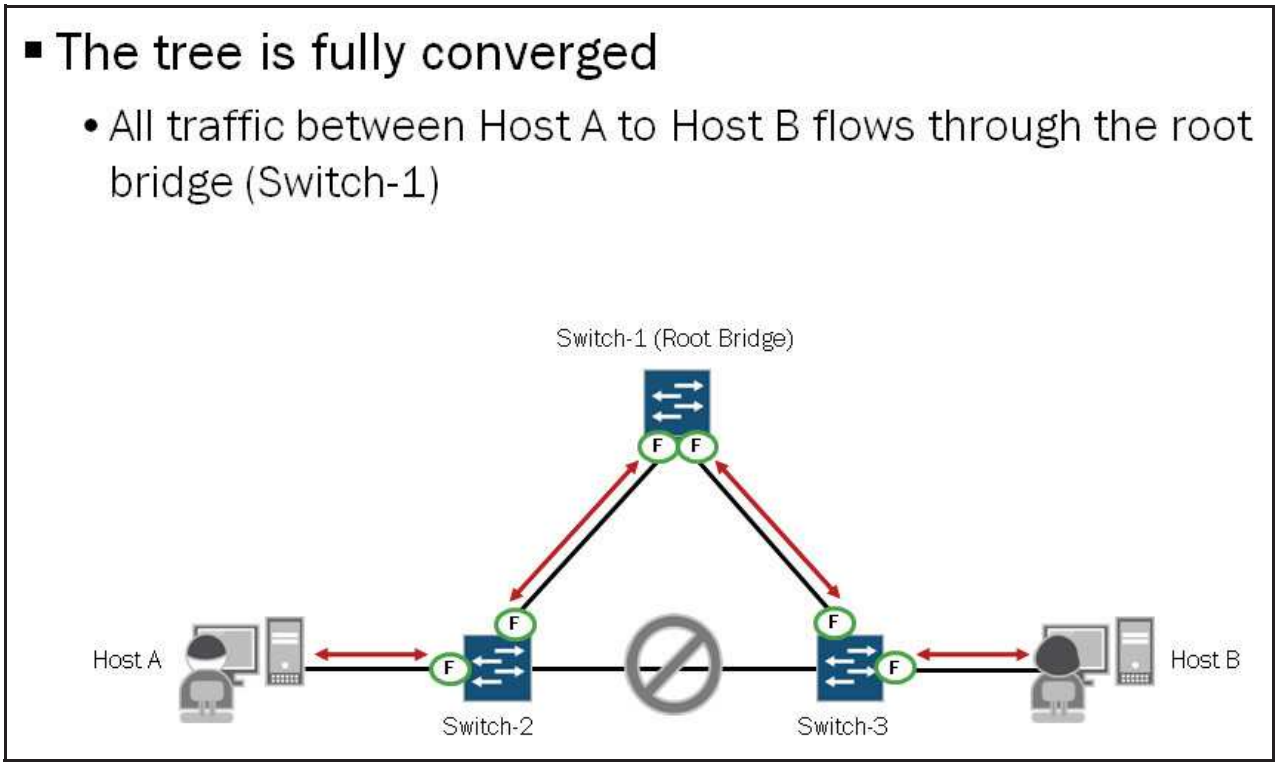


Once the root bridge election occurs, all nonroot devices perform a least-cost path calculation to the root bridge. The results of these calculations determine the role of the switch ports. The role of the individual switch ports determines the port state.

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.

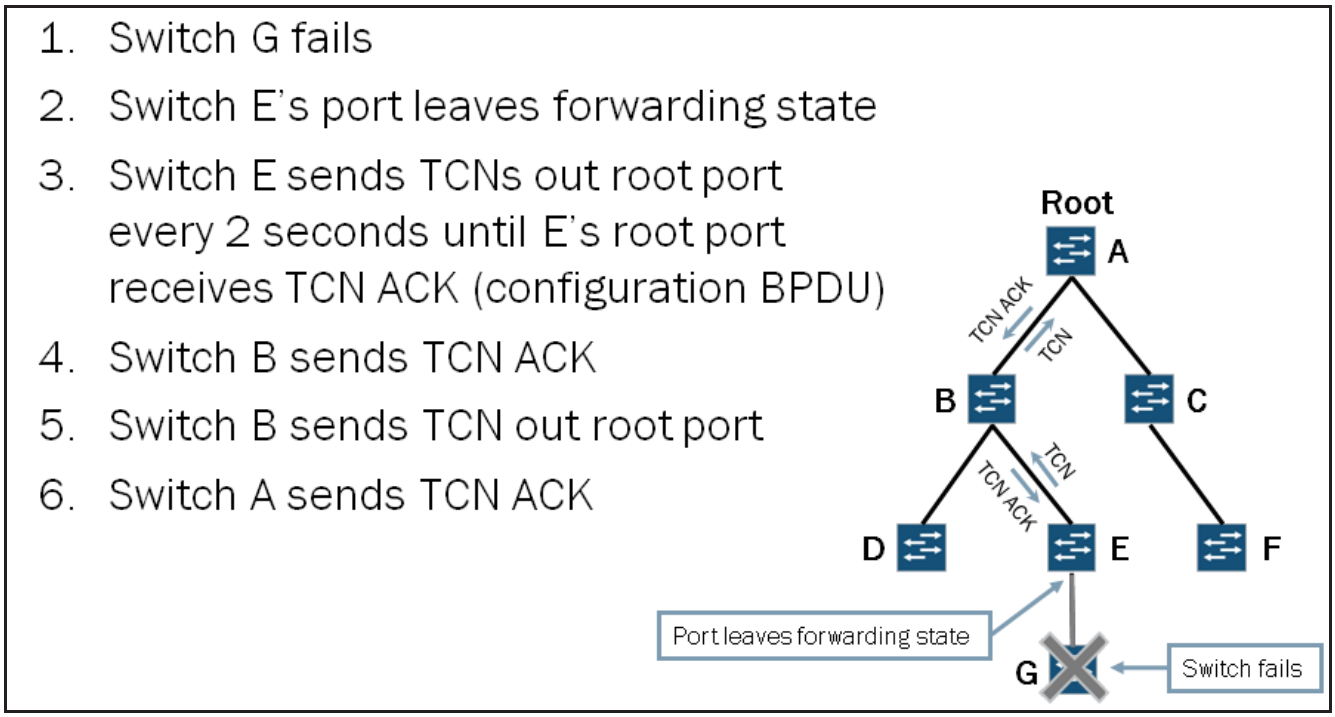
STP selects a designated bridge on each LAN segment. This selection process is also based on the least-cost path calculation from each switch to the root bridge. Once the designated bridge selection occurs, its port, which connects to the LAN segment, is chosen as the designated port. If the designated 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).

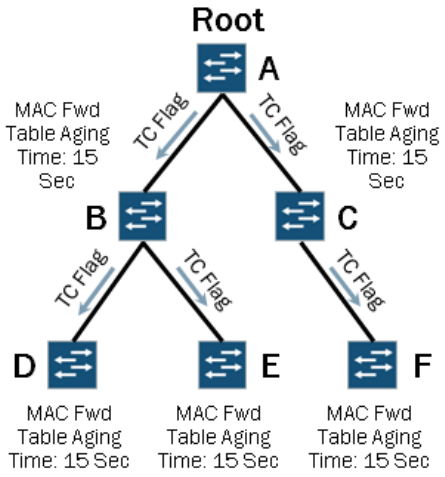
Reconvergence Example: Part 1



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.

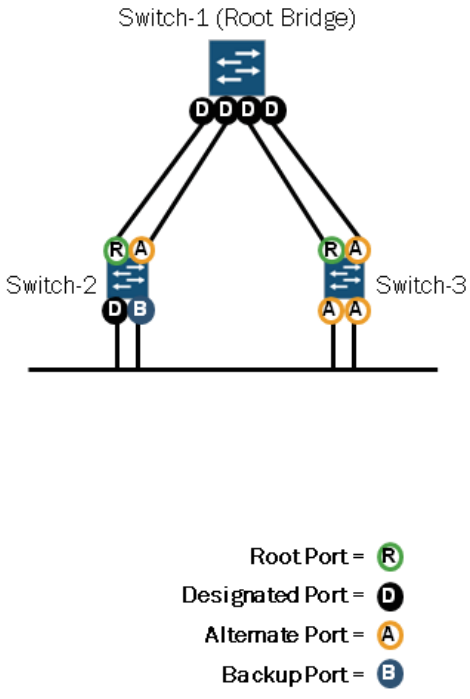
RSTP Convergence Improvements

- 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.

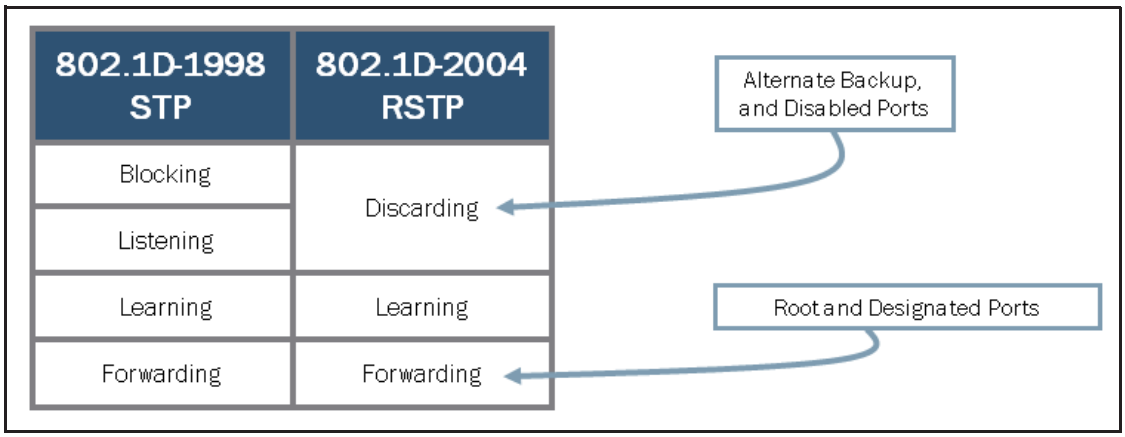
When a designated bridge has multiple ports connected to a shared LAN segment, it selects one of those ports as the designated port. The designated port is typically the port with the lower port ID. RSTP considers all other ports on the designated switch that connects to that same shared LAN segment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



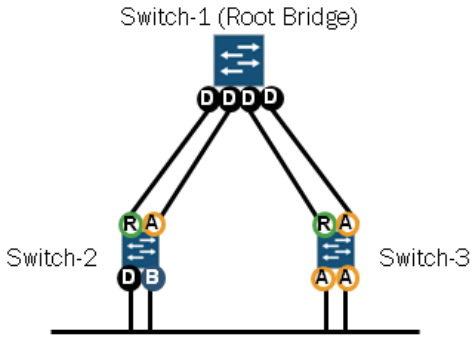
The graphic highlights the STP and RSTP port states. In addition to the states listed on the graphic, an interface can have STP administratively disabled. An administratively disabled port does not participate in the spanning tree but does flood any BPDUs it receives to other ports associated with the 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

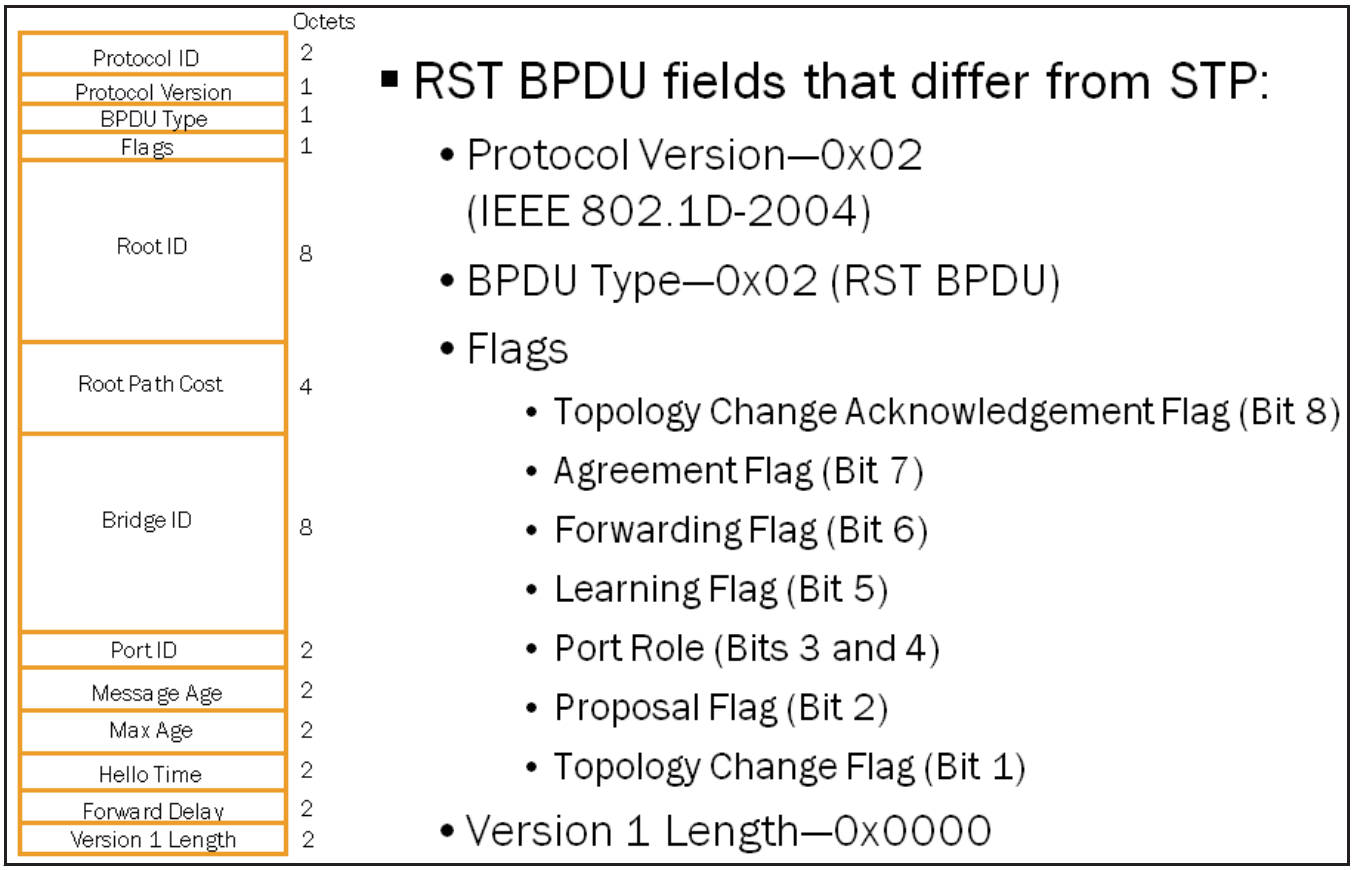


As previously mentioned, STP uses BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, prune specific redundant links to create a loop-free tree topology, and report and acknowledge topology changes. RSTP configuration BPDUs also 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

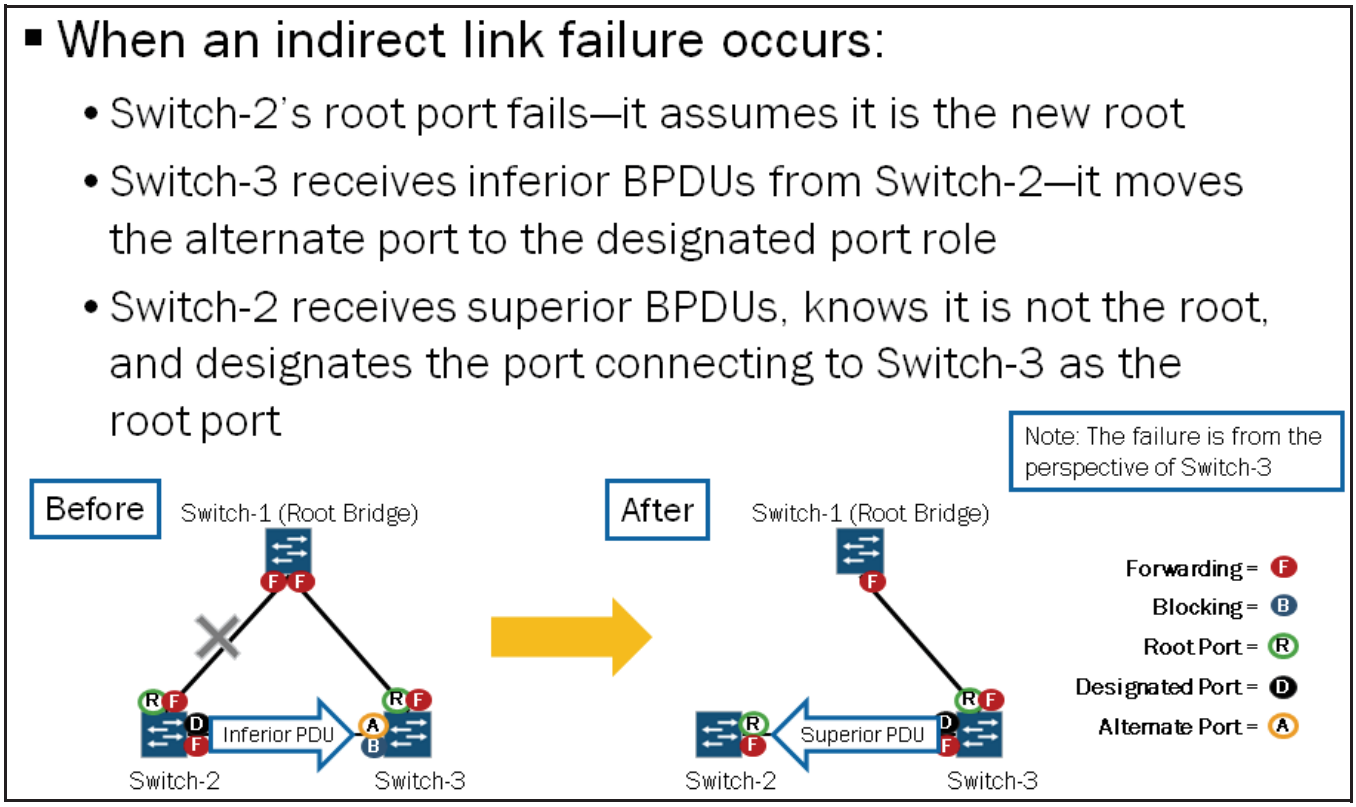
- Port transitions to the discarding state no longer trigger the STP TCN/TCN Acknowledgment sequence
- The initiator sends RSTP TCNs (RST BPDU with TCN flag set) out of all designated ports as well as out of the root port
- Because of the received RSTP TCN, switches flush the majority of MAC addresses 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.

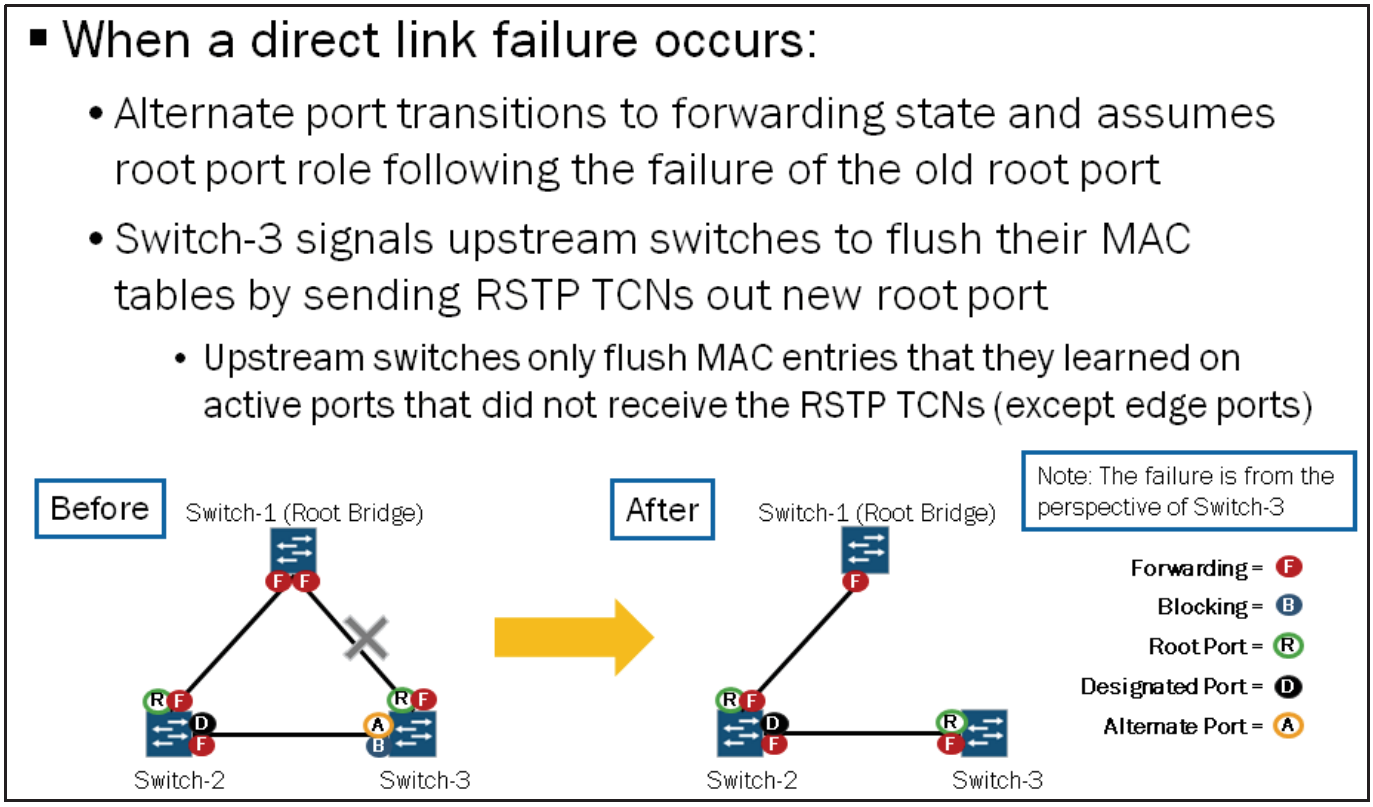
When a TCN is necessary and is generated, the initiating device floods all designated ports as well as the root port. Unlike traditional STP, neighboring switches that are not in the path of the initiator to the root bridge do not need to wait for this information from the root bridge. As the changes 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



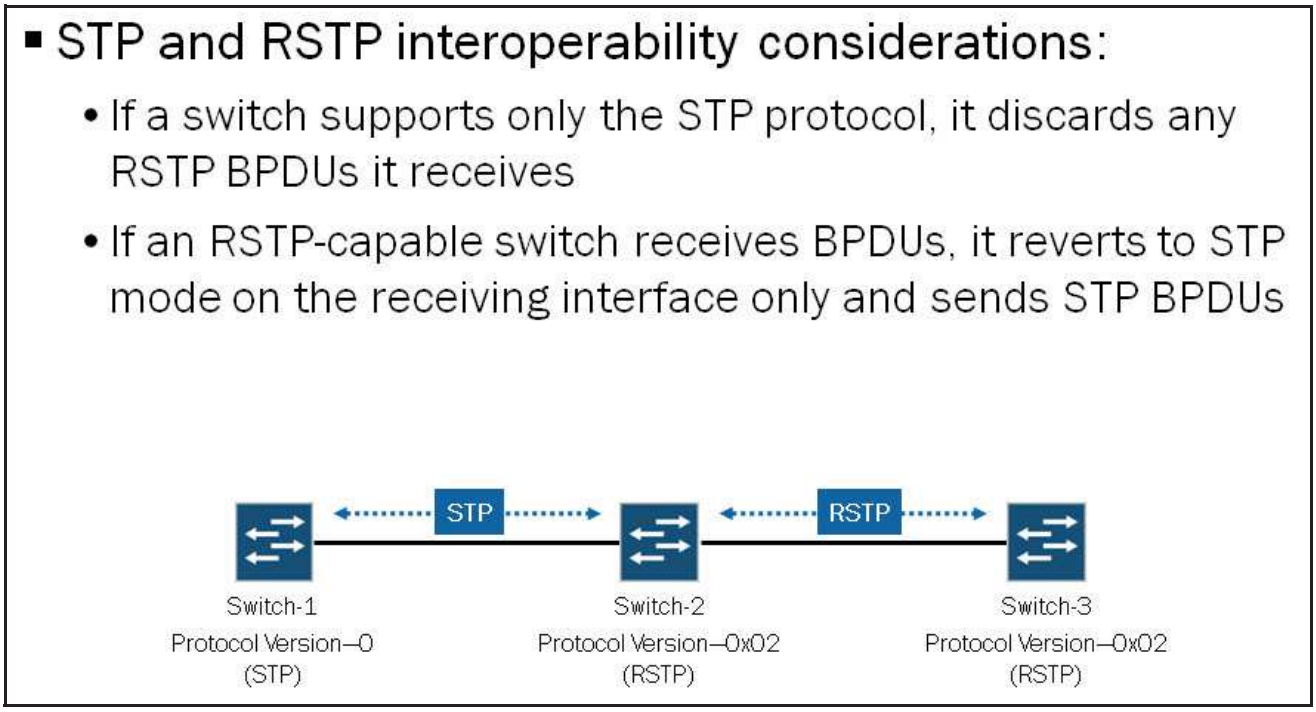
RSTP performs rapid recovery for link failures. The graphic illustrates a typical scenario for an indirect link failure from the perspective of Switch-3.

Direct Link Failure



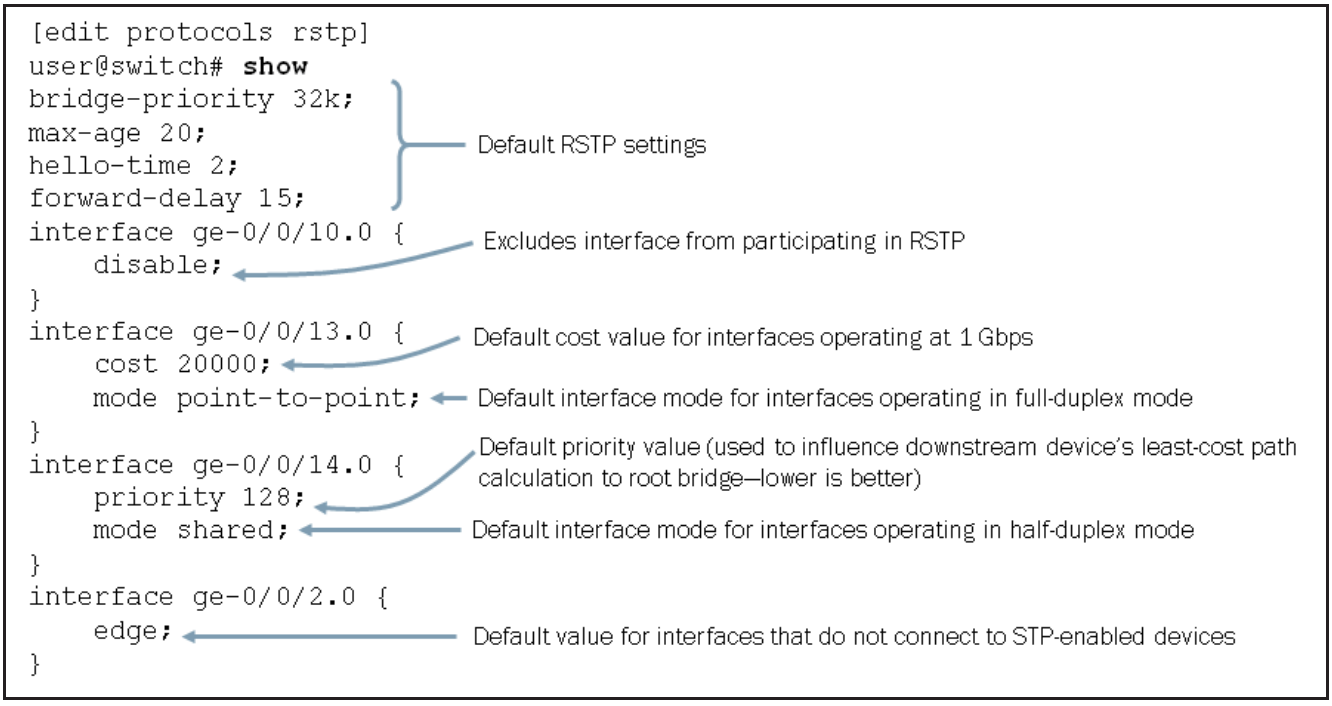
The graphic illustrates a typical scenario in which a direct link failure occurs, from the perspective of Switch-3.

Interoperability Considerations



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

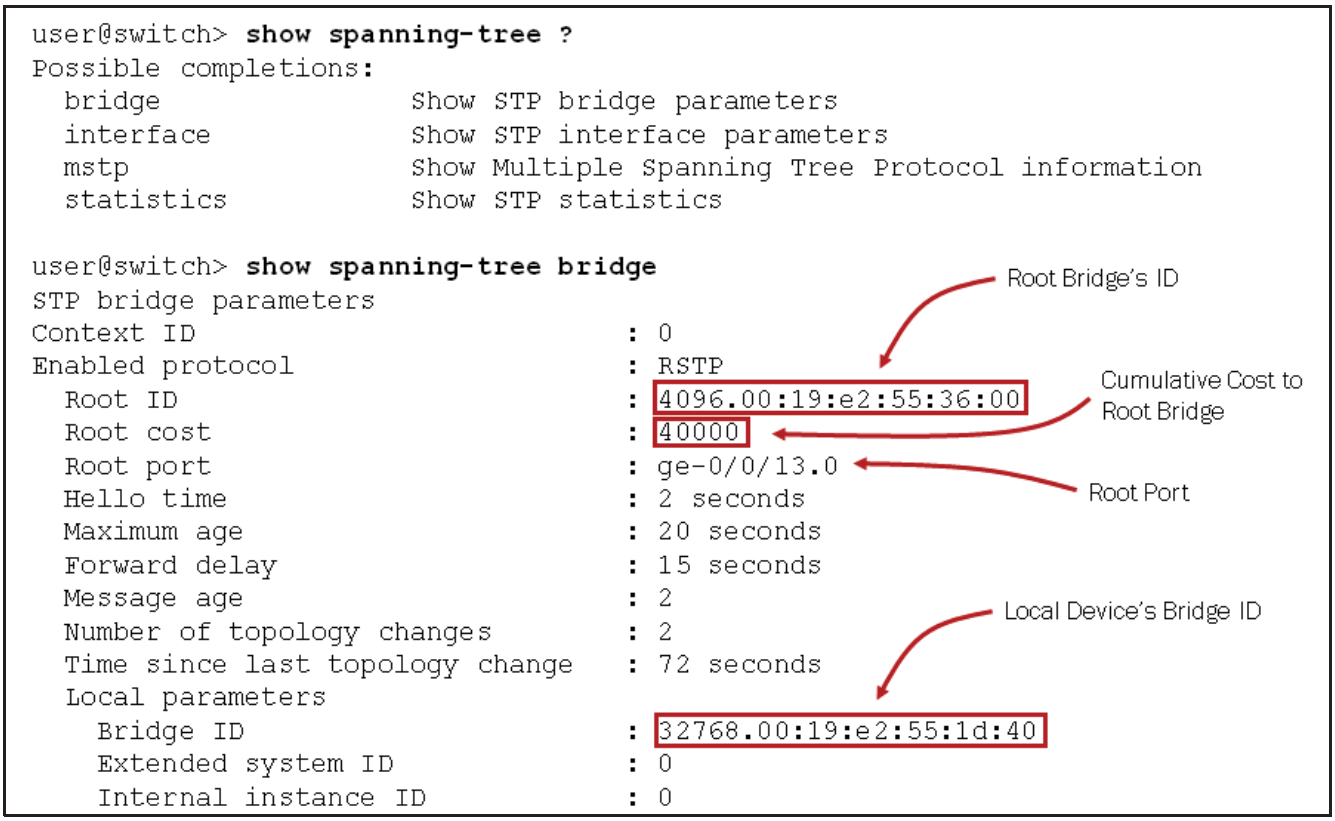


The graphic illustrates a sample RSTP configuration along with several highlighted settings. Note that the max age and forwarding delay values used by a switch always match the values defined on the root bridge device.

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



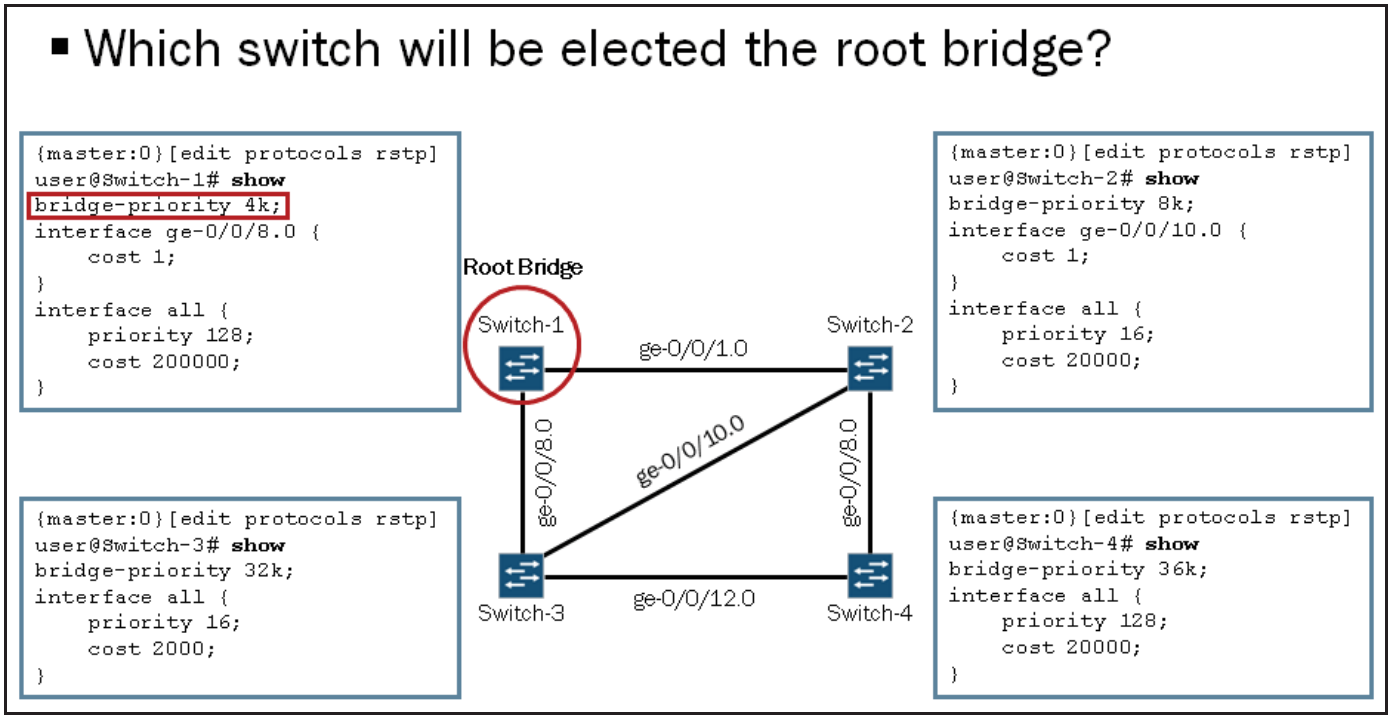
This graphic and the next illustrate some common operational-mode commands used to monitor the operation of STP and RSTP.

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.

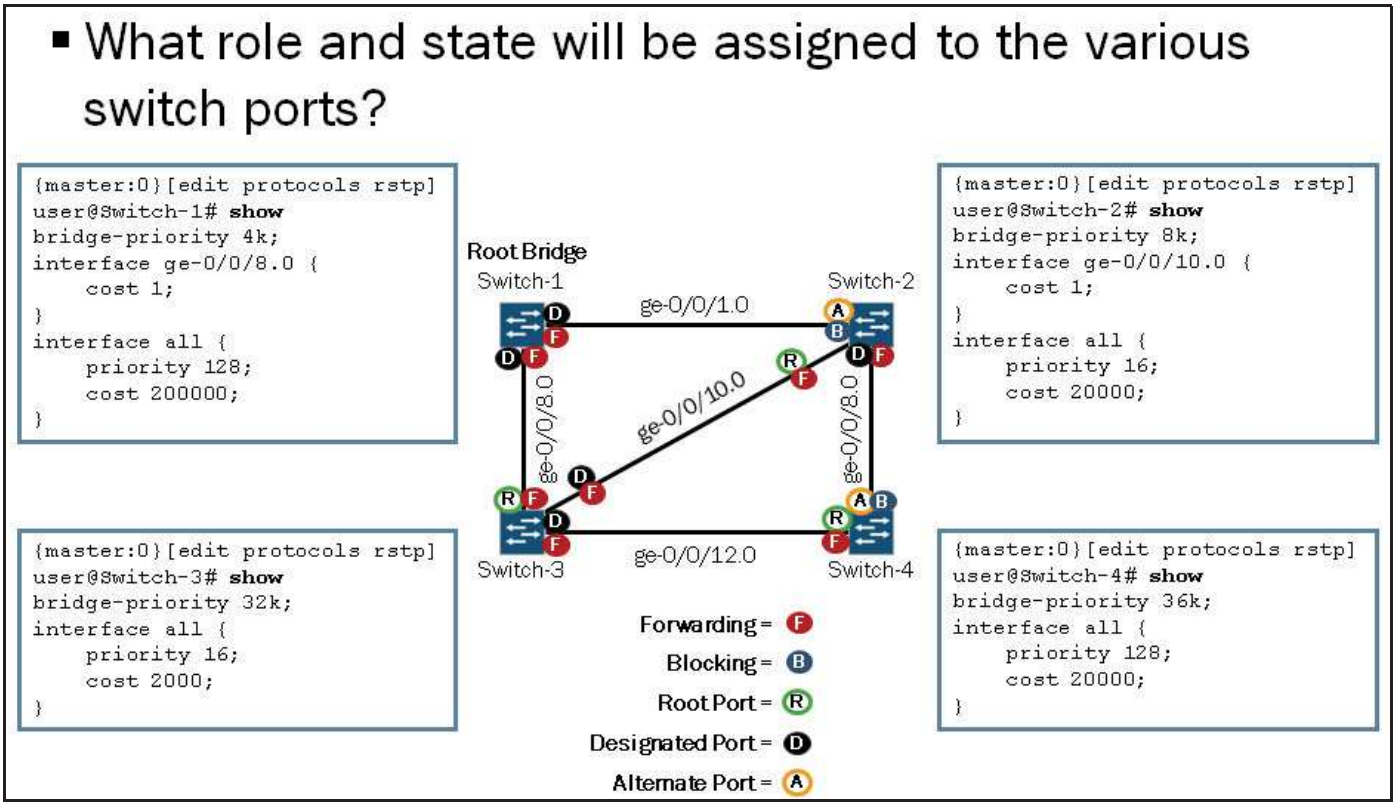
Test Your Knowledge: Part 1



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

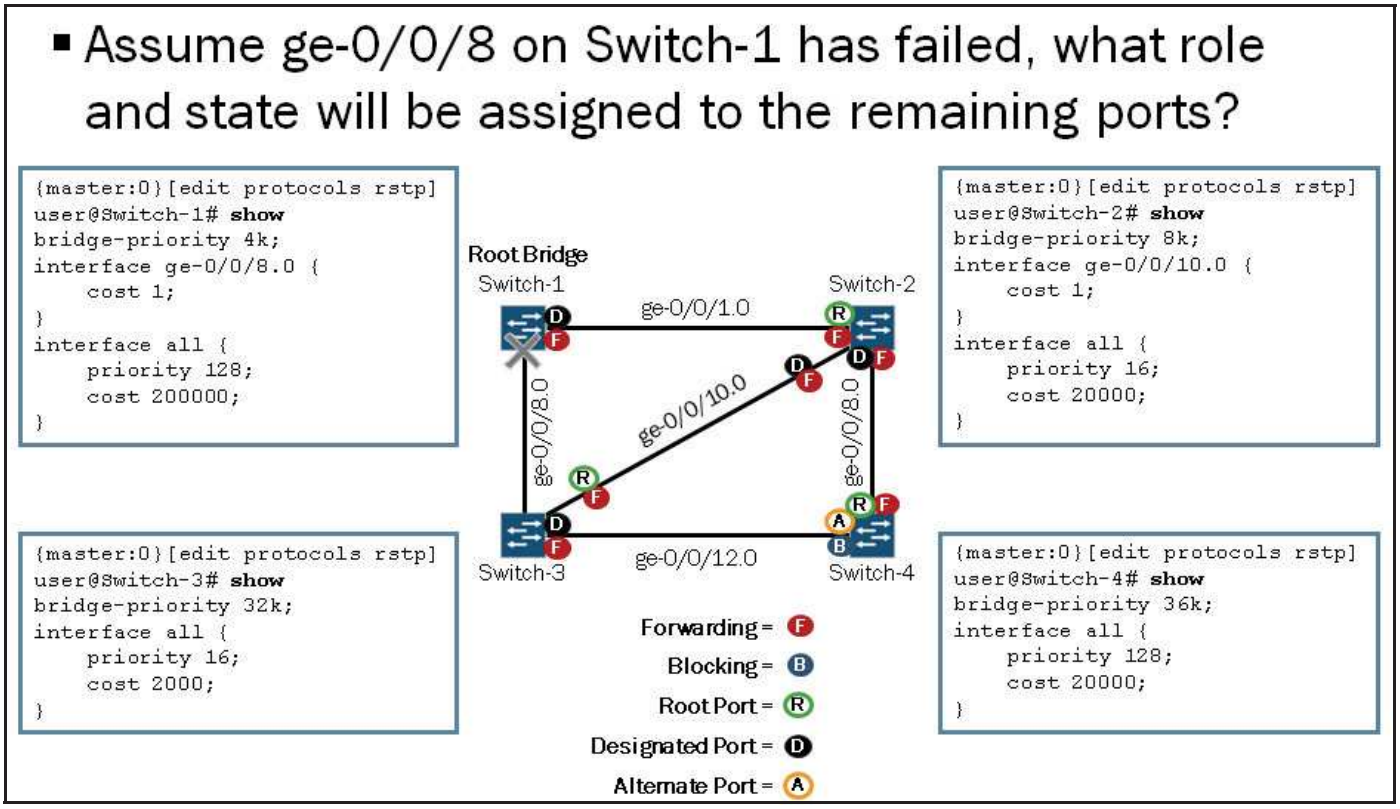


This graphic is designed to test your understanding of the various configuration options and how they relate to port role and state determination. As shown in the following output, you can use the **show spanning-tree interface** command to verify spanning tree interface information:

```
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
```

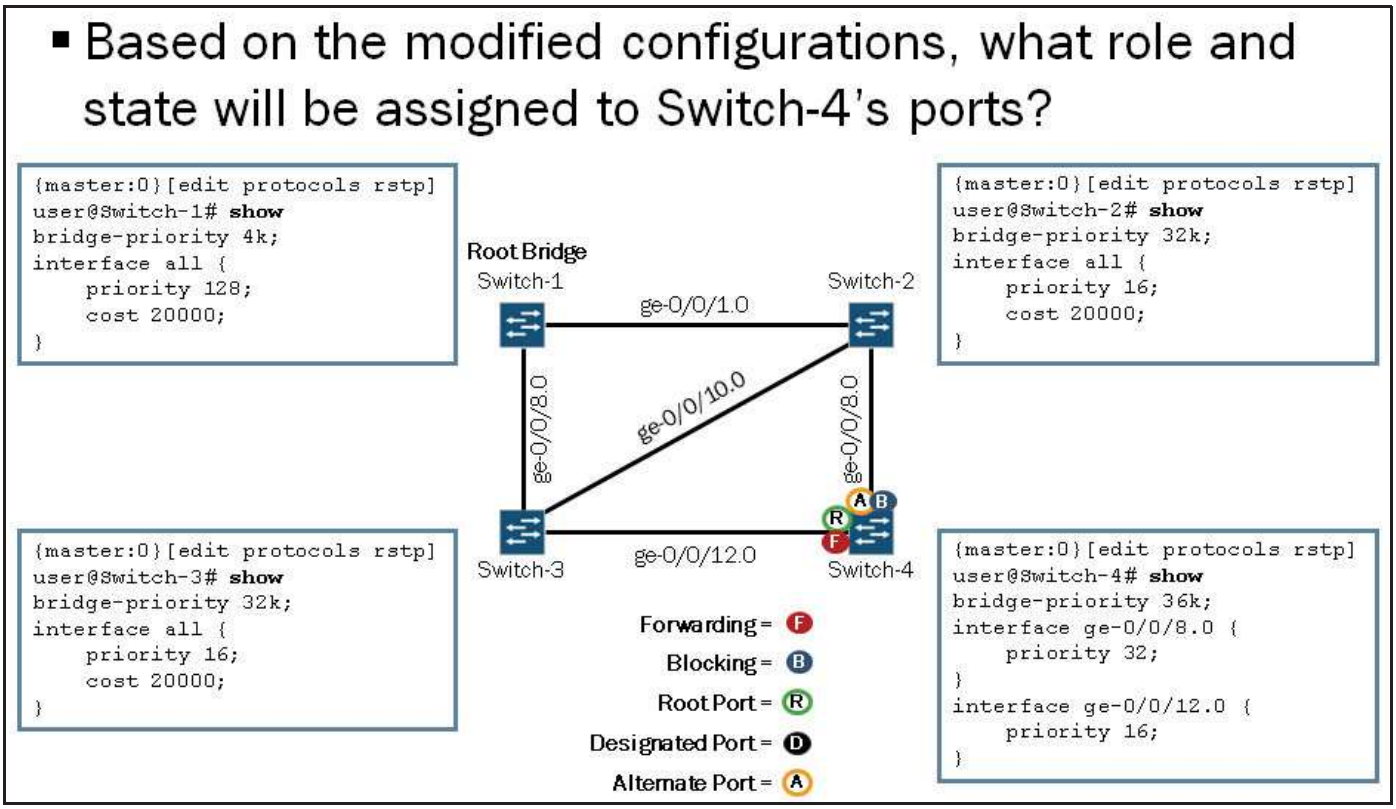
Test Your Knowledge: Part 3



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
```

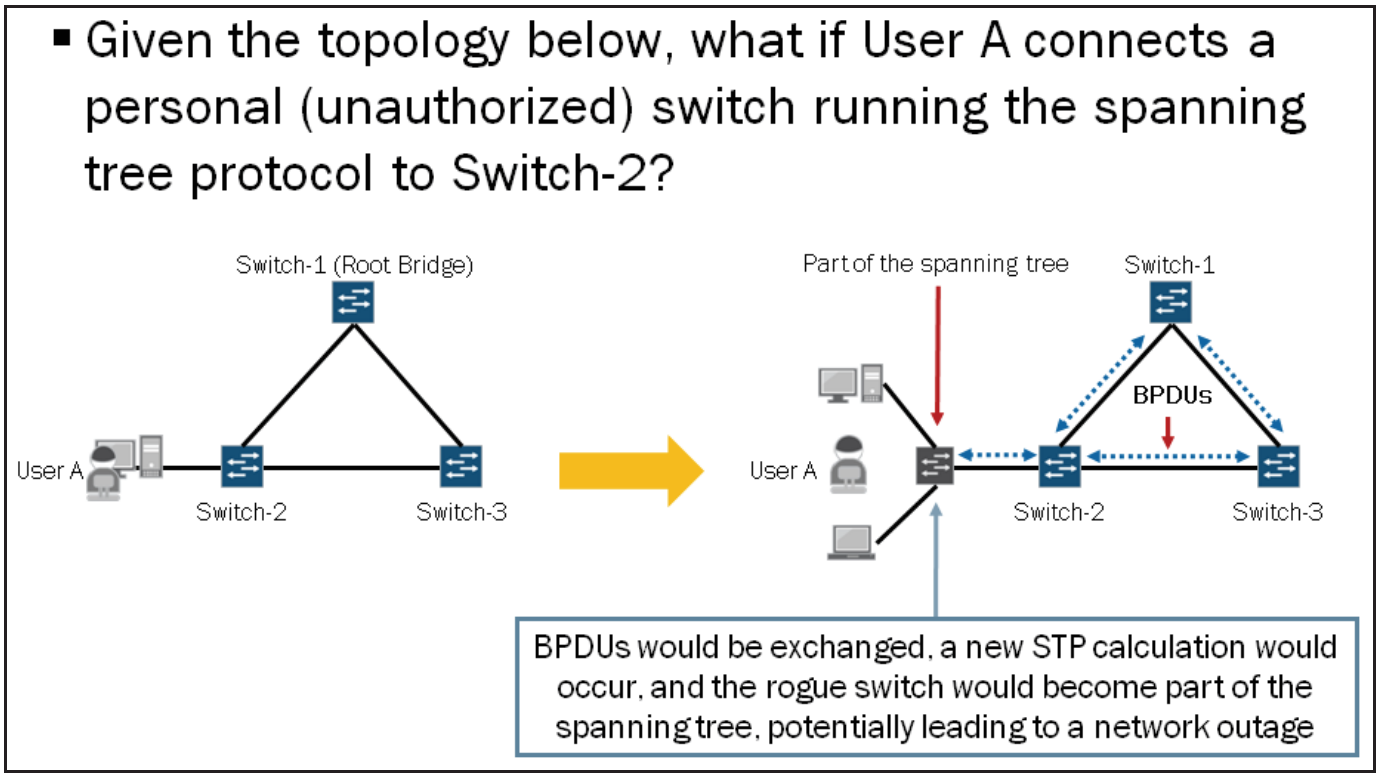
Test Your Knowledge: Part 4



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:

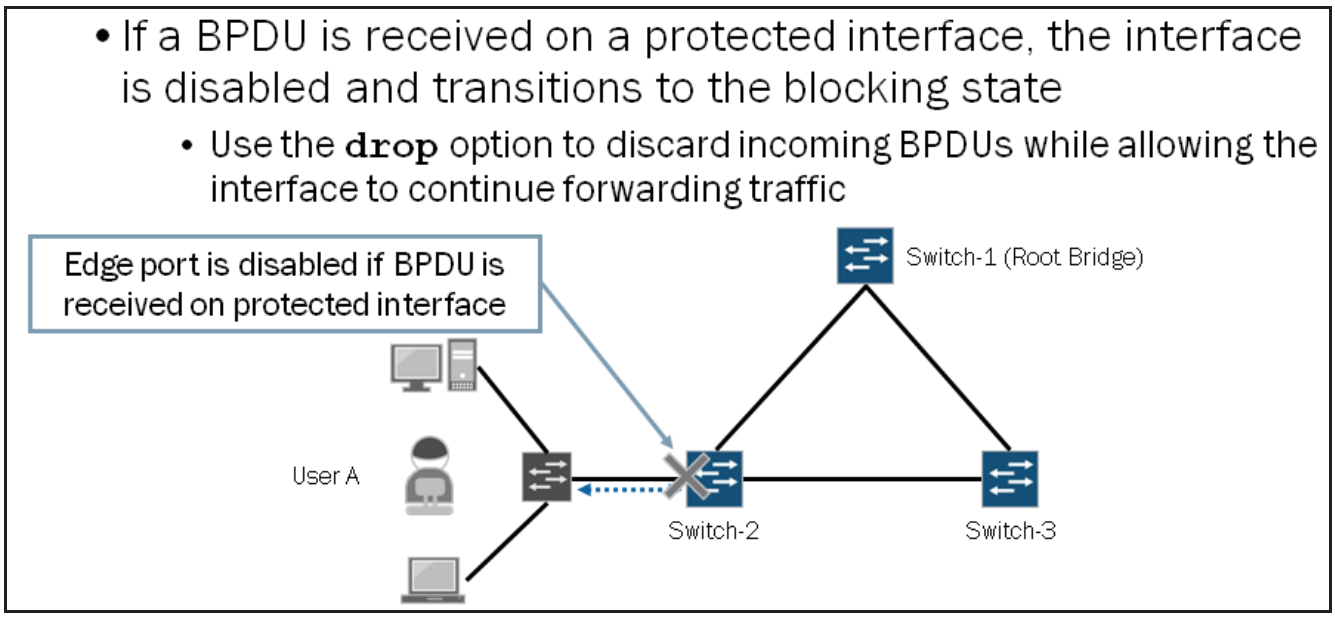
```
user@Switch-4> show spanning-tree interface
Spanning tree interface parameters for instance 0
Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID         bridge ID          Cost
ge-0/0/8.0      32:521       16:521  32768.002688026b90     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

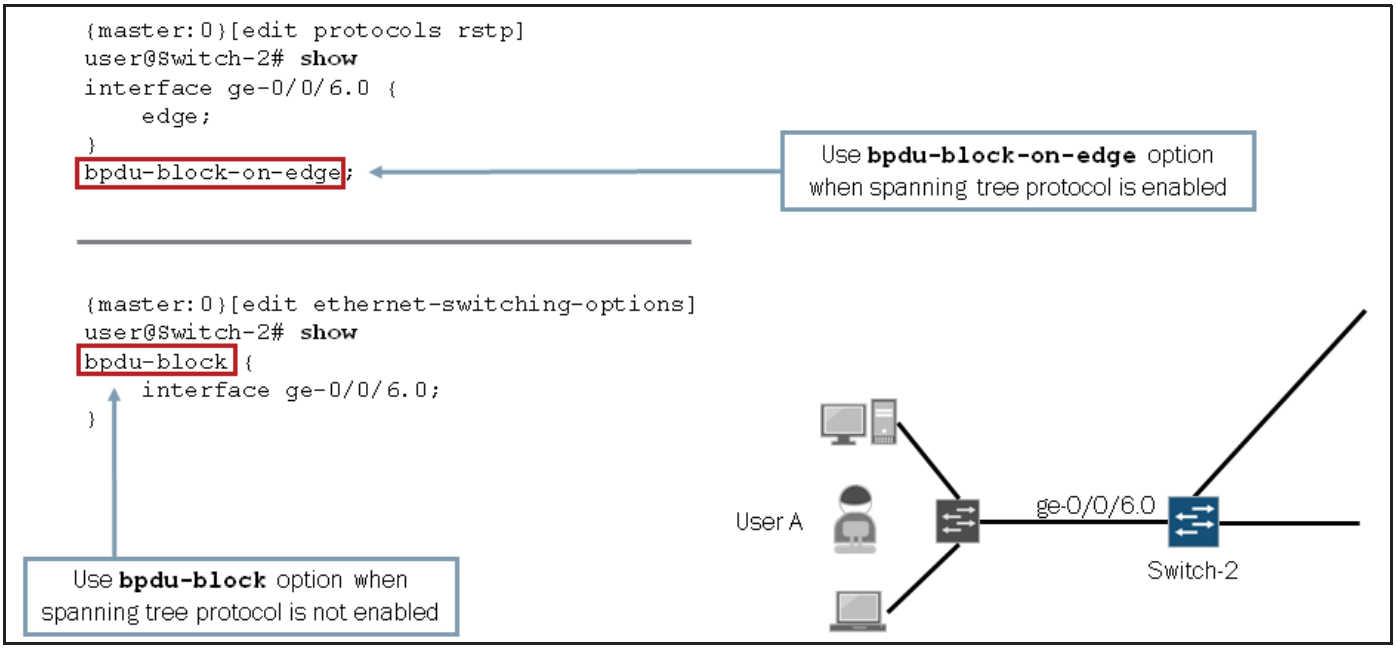


You can enable BPDU protection on switch interfaces on which no BPDUs are expected. If a protected interface receives BPDUs, the switch disables the interface and stops forwarding frames by transitioning the interface to a blocking state. In some situations you might not want the interface to become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

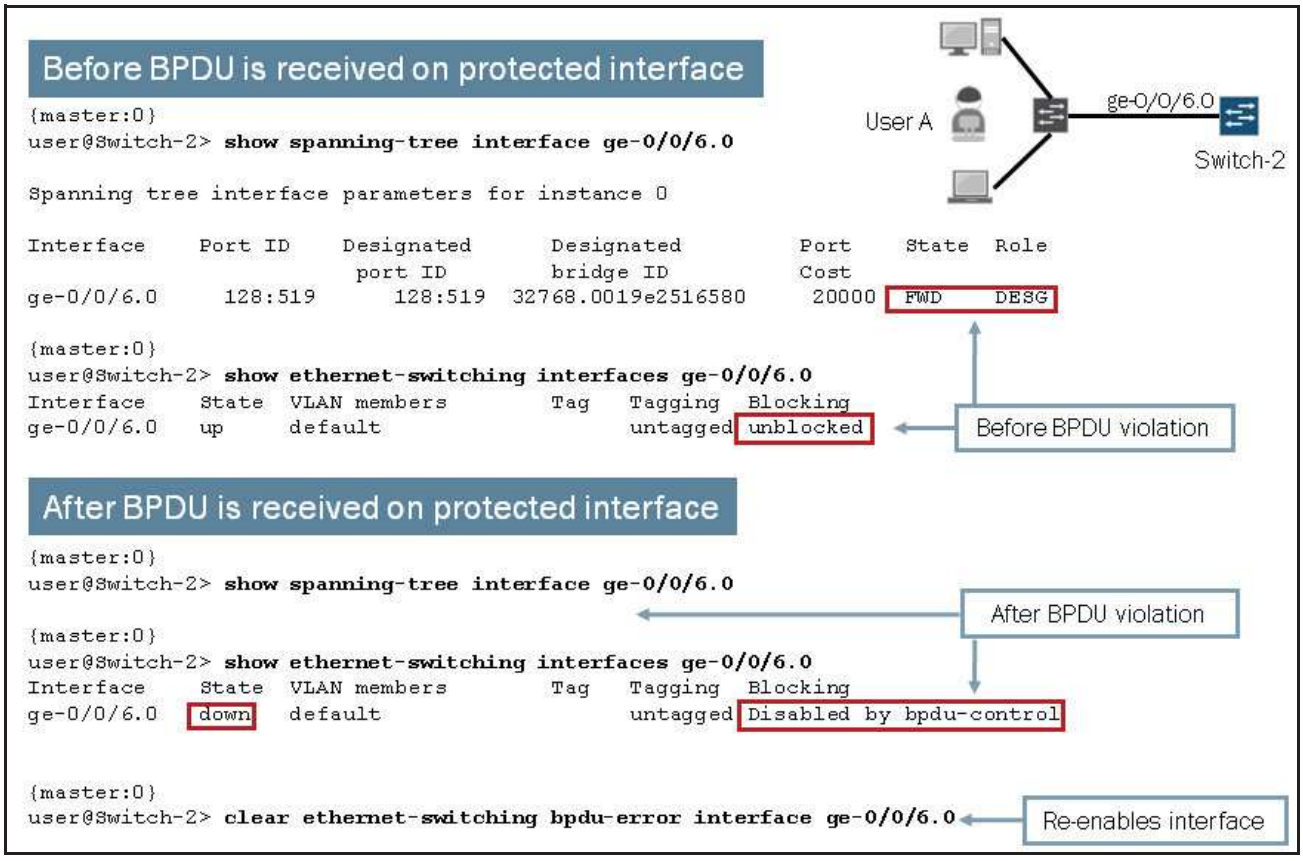
You can configure BPDU protection on a switch with a spanning tree as well as on a switch that is not running STP. We discuss the configuration of BPDU protection in the next section.

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.

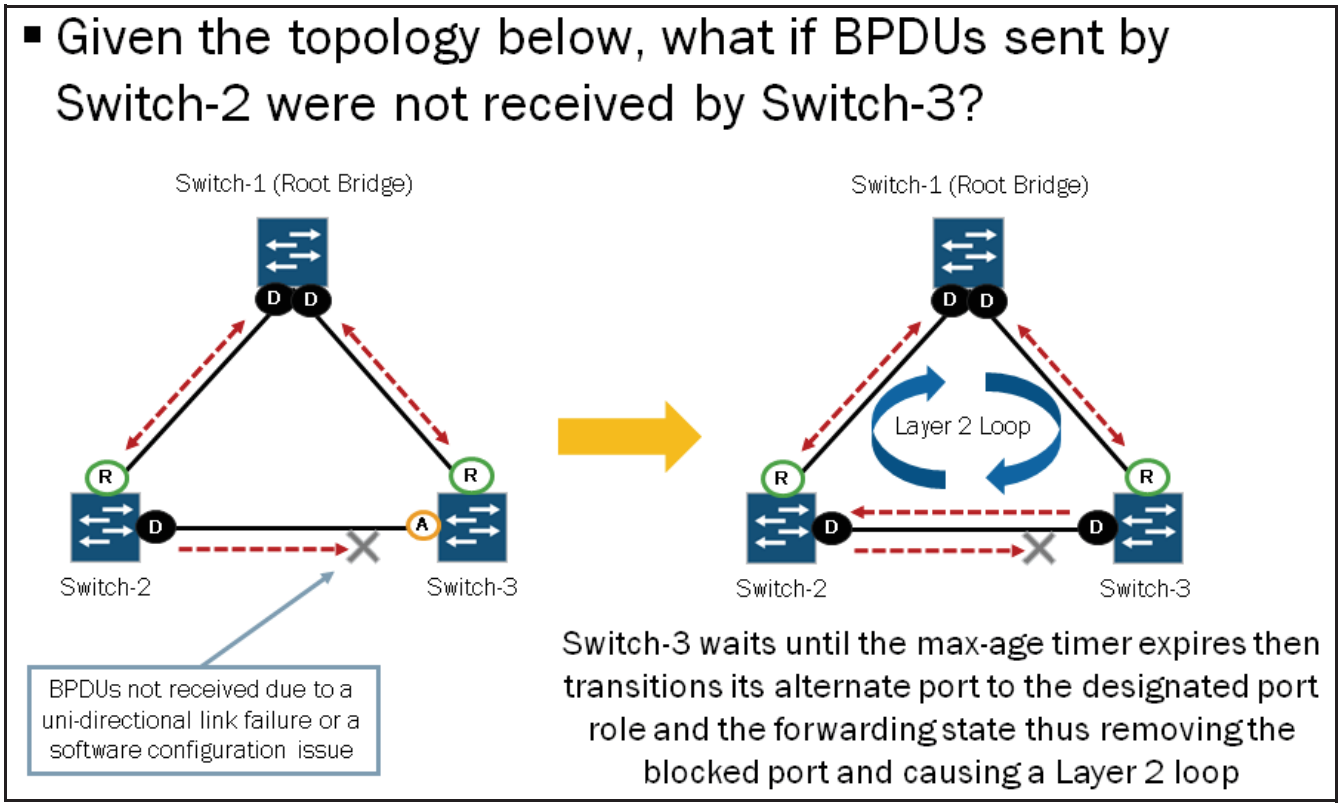
These commands provide the information on the state and role changes on the protected interfaces. Specifically, once the BPDUs are sent from an offending device to the protected interface, the interface transitions to the `DIS` role, meaning that it becomes a BPDU inconsistent state. The BPDU inconsistent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

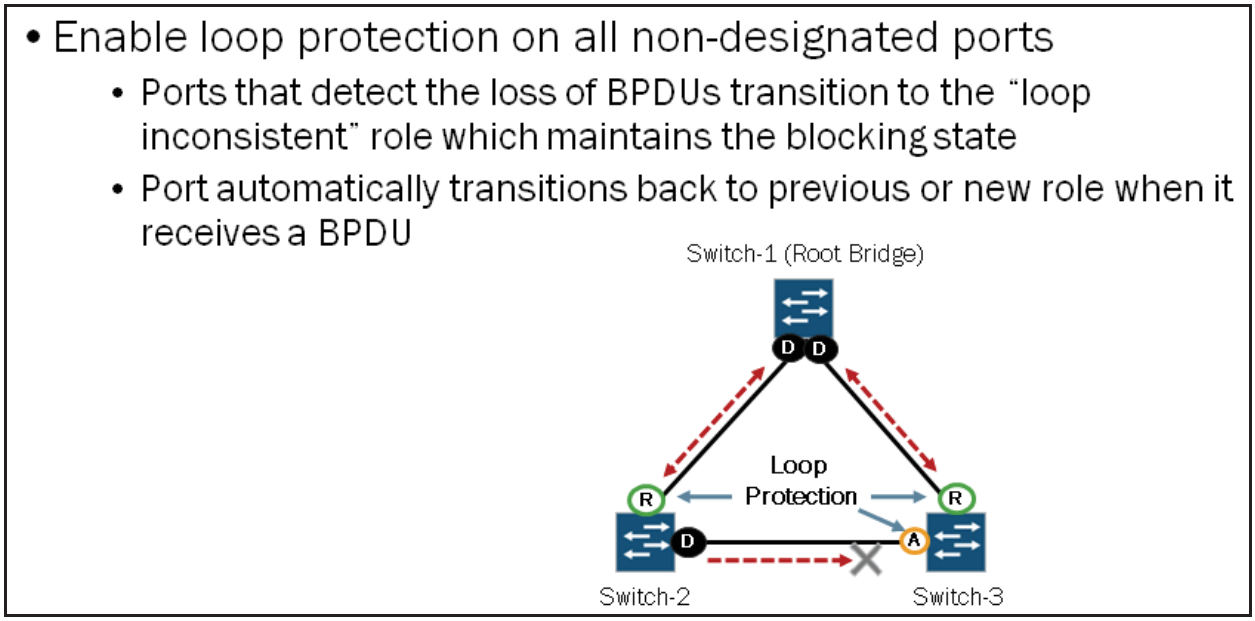
Disabling the BPDU protection configuration for an interface does not unblock the interface. You must clear the violation using the **clear ethernet-switching bpdu-error** command or wait for the configured timer to expire.

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



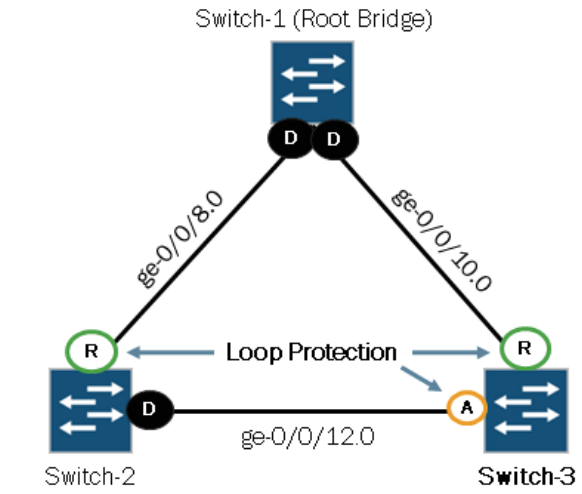
When loop protection is enabled, the spanning-tree topology detects root ports and blocked ports, and ensures that both are receiving BPDUs. If an interface with the loop protection feature enabled stops receiving BPDUs from its designated port, it reacts as it would react to a problem with the physical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

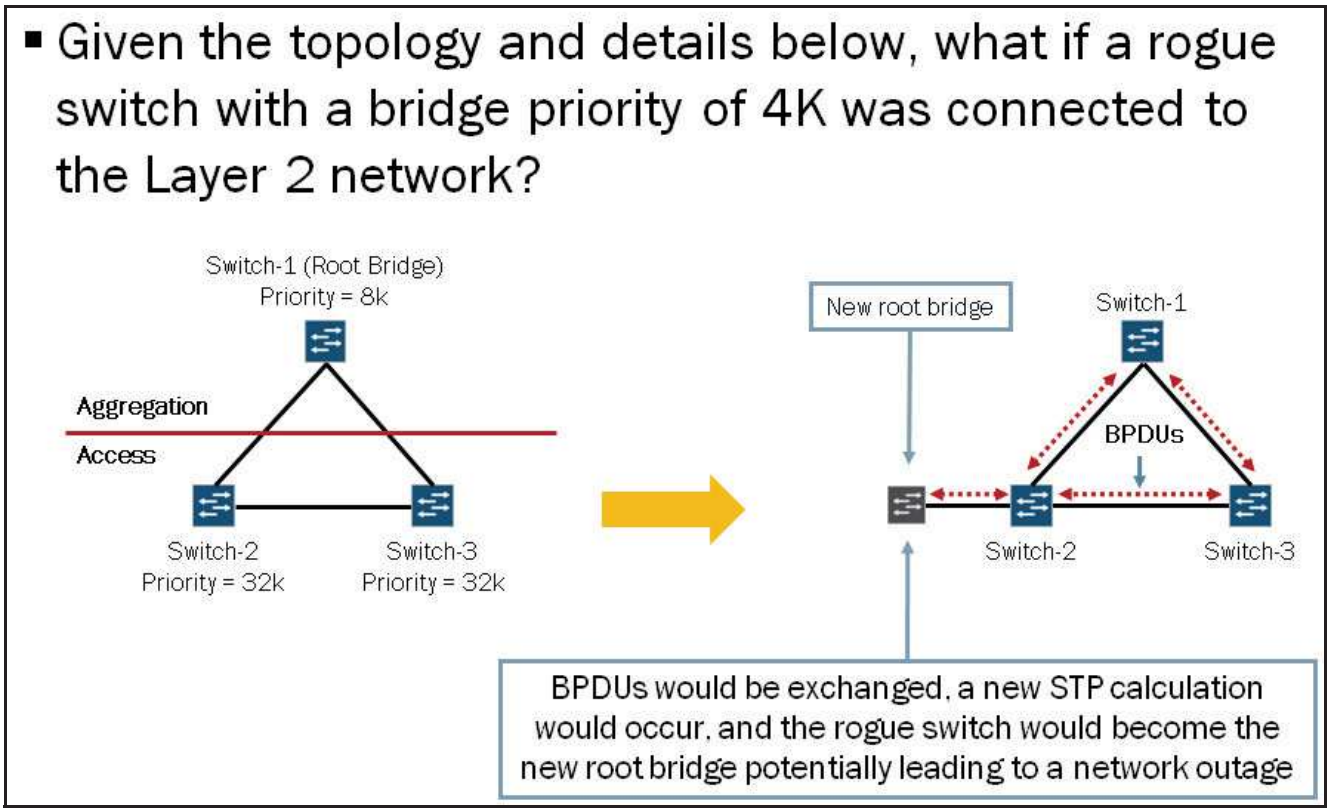
To confirm that the configuration is working properly on the STP-running switch, use the **show spanning-tree interface** operational mode command prior to configuring loop protection. This command provides information for the interface's spanning-tree state, which should be blocking (BLK).

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:

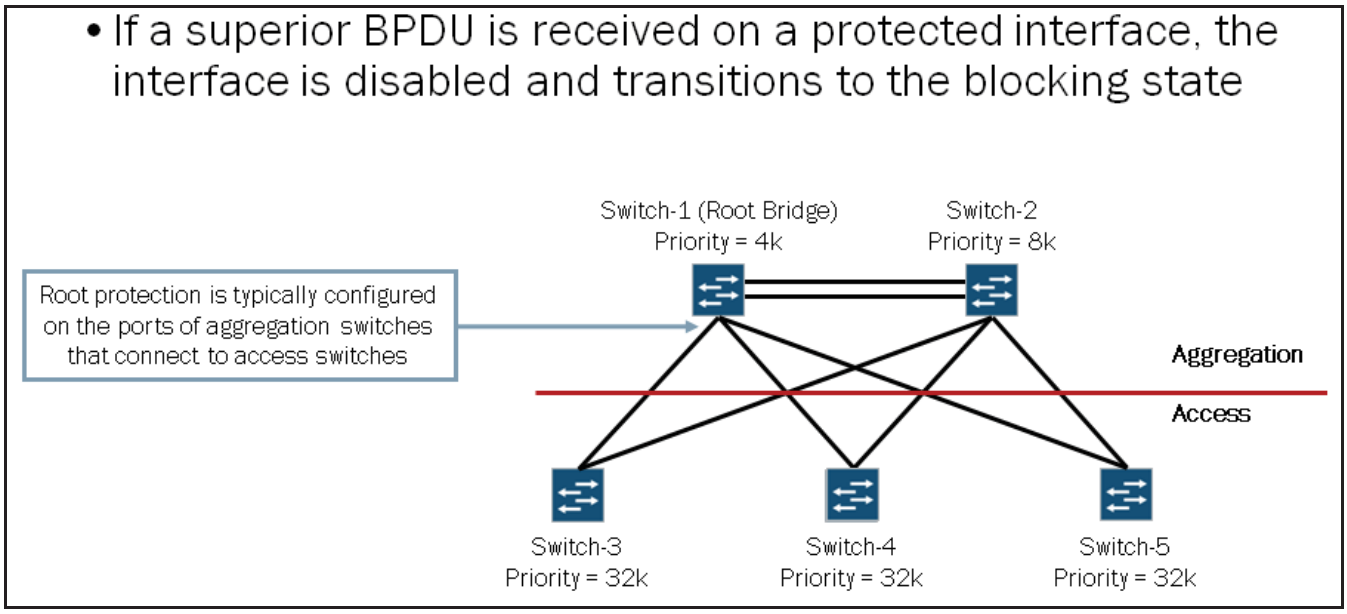
```
{master:0}
user@Switch-3> show log messages | match "loop|protect"
Apr 27 20:04:49  Switch-3 eswd[40744]: Loop_Protect: Port ge-0/0/12.0: Received
information expired on Loop Protect enabled port
Apr 27 20:04:49  Switch-3 eswd[40744]: ESWD_STP_LOOP_PROTECT_IN_EFFECT: ge-0/0/12.0:
loop protect in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

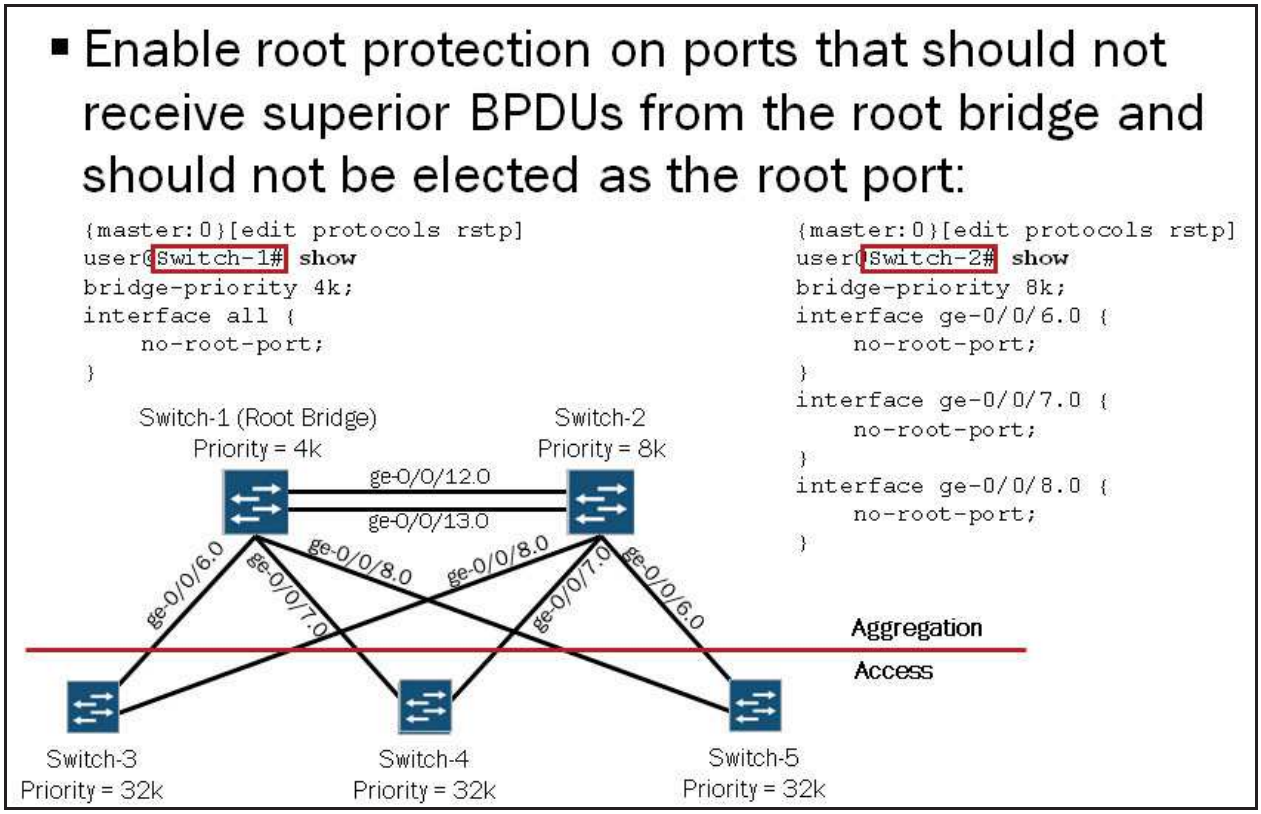


Enable root protection on interfaces that should not receive superior BPDUs and should not be elected as the root port. These interfaces become designated ports. If the bridge receives superior BPDUs on a port that has root protection enabled, that port transitions to an inconsistency state, blocking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