



Switched Campus

VLAN Technologies



In This Section

- + Port Types
 - + Access Ports
 - + 802.1q Trunk Ports
 - + Dynamic Ports
- + VLANs
 - + Standard VLANs
 - + Extended VLANs
 - + VLAN Database
- + Trunking
 - + Dynamic Trunking Protocol (DTP)
 - + 802.1q Native VLAN
 - + Trunking Allowed List

Ethernet Port Types

- + Layer 2 Switchports
 - + Access
 - + Trunk
 - + Tunnel
 - + Dynamic
- + Layer 3 Ports
 - + Switched Virtual Interface (SVI)
 - + Native Routed Interfaces

Layer 2 Switchports

- + Access Switchports
 - + One VLAN
- + Trunked Switchports
 - + Multiple VLANs
- + Tunnel Switchports
 - + Transparent Layer 2 VPN
- + Dynamic Switchports
 - + DTP Negotiation

Layer 2 Trunking

- + 802.1q
 - + Open standard
 - + “Native” VLAN sent untagged
- + DTP
 - + Dynamic Trunking Protocol
- + Verified with...
 - + show interface trunk
 - + show interface switchport
 - + show spanning-tree [vlan | interface]

DTP Negotiation

- + Enabled by default on some platforms
- + DTP Desirable mode
 - + Initiates trunking negotiation
 - + switchport mode dynamic desirable
 - + switchport mode trunk
- + DTP Auto mode
 - + Passively listen for trunking negotiation
 - + switchport mode dynamic auto

Disabling DTP Negotiation

- + Can be disabled by...
 - + `switchport nonegotiate`
 - + `switchport mode access`
 - + `switchport mode dot1q-tunnel`
- + Verified with...
 - + `show interface switchport`

VLANs

- + VLANs generally fall into three categories
 - + Standard
 - + Extended
 - + Internal

Standard VLANs

- + Standard VLAN range is 1 – 1005
- + VLAN 1
 - + Default Ethernet Access VLAN & default 802.1q Native VLAN
 - + Cannot be deleted, but can be manually pruned from trunks
 - + Cannot be pruned by VTP
 - + Should not be used for actual port assignments
- + VLANs 1002 – 1005
 - + Default legacy Token Ring / FDDI VLANs
 - + Cannot be deleted, but can be manually pruned from trunks
 - + Cannot be pruned by VTP
 - + Should not be used for actual port assignments

Extended VLANs

- + Extended VLAN range is 1006 - 4094
- + Can normally only be used in one of two cases
 - + VTP is configured in Transparent Mode
 - + VTP Version 3
- + Not all extended VLANs can be used
 - + Some are reserved for “internal” usage

Internal VLANs

- + VLANs reserved for internal applications
 - + E.g. native layer 3 switchports
 - + show vlan internal usage
- + Not all platforms agree on the internal range
 - + For real deployments, check the internal allocations
 - + Some allocate ascending, some descending

Creating VLANs

- + VLANs can be created...
 - + Globally
 - + VLAN database
 - + At the time of assignment
- + Creating a VLAN automatically creates...
 - + Spanning-Tree instance
 - + MAC address table
- + Verified with...
 - + show vlan [brief]
 - + show spanning-tree vlan

Manual Trunk Pruning

- + Trunk's "allowed list" controls what VLANs will forward over the link
 - + All VLANs (1-4094) by default
- + Allowed list can be edited for manual pruning
 - + switchport trunk allowed vlan
- + Verified with...
 - + show interface trunk
 - + show interface switchport

Command Review

- + show vlan [brief]
- + show interface status
- + show interface switchport
- + show interface trunk
- + show spanning-tree [vlan | interface]





Switched Campus

VLAN Trunking Protocol (VTP)



In This Section

- + VTP v1/v2 Modes
- + VTP v1/v2 Authentication
- + VTP v1/v2 Pruning

VLAN Trunking Protocol (VTP)

- + What is VTP?
 - + Used to synchronize VLAN creation between switches
 - + E.g. simplifies the management of VLANs
- + What is VTP not?
 - + Not a requirement of Ethernet networks
 - + Does not define the broadcast domain

VTP v1 & v2 Modes

- + Three different modes of operation
 - + Server
 - + Client
 - + Transparent

VTP Server

- + Creates VLANs
- + Advertises VLANs
- + Installs VLANs from other advertisements

VTP Client

- + Cannot create VLANs
- + Advertises VLANs
- + Installs VLANs from other advertisements

VTP Transparent

- + Creates locally significant VLANs
- + Transparently forwards other VTP advertisements
- + Does not install VLANs from other advertisements

VTP Configuration Revision

- + Configuration Revision Number
 - + Sequence number for the database
 - + Highest number wins
 - + Domain is synchronized when revision number matches everywhere
- + Potential problems in VTP
 - + Wrong database with high configuration revision number can overwrite the database
 - + True for both VTP servers and clients
 - + Reason that VTP v1/v2 is rarely used in production

VTP Authentication

- + Used for validation of VTP updates
- + Configuring / Verifying
 - + vtp password
 - + show vtp password
 - + show vtp status
 - + compare MD5 hashes

VTP Pruning

- + Reduces unnecessary replication of...
 - + Broadcasts
 - + Unknown unicasts
 - + Unknown multicast
- + Only supported in server & client mode
- + Configuring / Verifying
 - + vtp pruning
 - + show interface trunk
 - + show interface pruning

VTP Pruning (cont.)

- + VLANs 2 – 1001 are “prune eligible”
 - + i.e. able to be pruned
- + VLANs not in the “prune eligible list” cannot be pruned
 - + i.e. traffic will always be sent/received for them
- + To edit prune eligible list...
 - + switchport trunk pruning vlan
- + To verify...
 - + show interface trunk
 - + show interface pruning
 - + show interface switchport

VTP Pruning Problems

- + What if all devices don't support pruning?
 - + VTP is Cisco proprietary
- + What if devices don't agree on the VTP Revision Number?
- + What if there are VTP Transparent switches?





Switched Campus

VLAN Trunking Protocol Version 3 (VTPv3)



In This Section

- + VTP Version 3 Enhancements
- + VTP Version 3 Configuration

VTP Version 3 Enhancements

- + Security enhancements
 - + Fixes configuration revision overwrite problem
- + New advertisements
 - + Extended VLANs
 - + Private VLANs
 - + MST configuration
- + VTP can now be disabled
 - + Globally
 - + Per Link





Switched Campus

EtherChannel



In This Section

- + What Is EtherChannel?
- + How Does EtherChannel Work?
- + Single vs. Multi-Chassis EtherChannel
- + EtherChannel Negotiation Protocols
- + EtherChannel Load Balancing
- + Layer 2 vs. Layer 3 EtherChannels

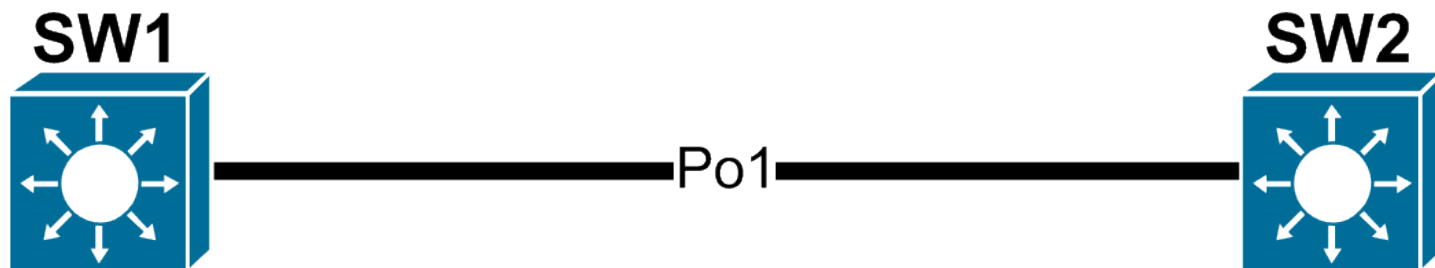
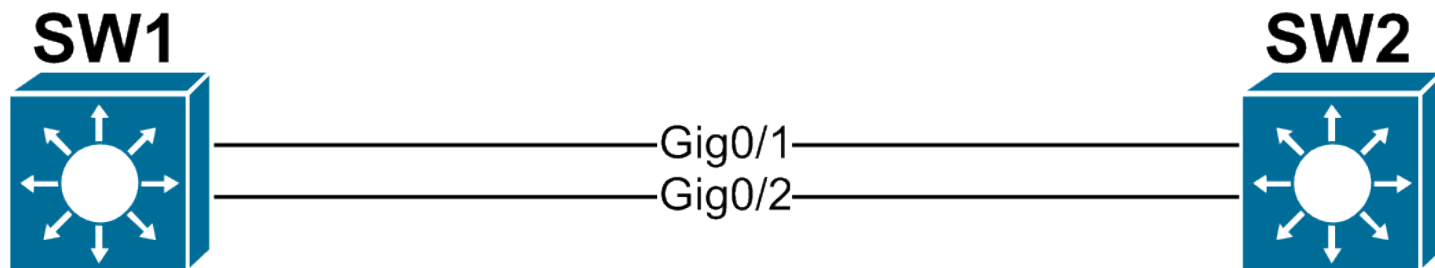
What Is EtherChannel?

- + Technique for aggregating the bandwidth of physical links together
- + Also known as...
 - + Port Channels
 - + Channeling
 - + Link Aggregation (LAG)
 - + NIC Teaming

How EtherChannels Work

- + EtherChannels consists of two parts
 - + Port-Channel interface
 - + Logical interface representing the link bundle
 - + Member interfaces
 - + Physical links that are part of the link bundle
- + LAG goal is to hide the member interfaces from the upper layer protocols
 - + E.g. STP sees one 2Gbps link not 2 x 1 Gbps link
 - + Result is active/active forwarding instead of active/standby with STP

EtherChannel Physical vs. Logical View



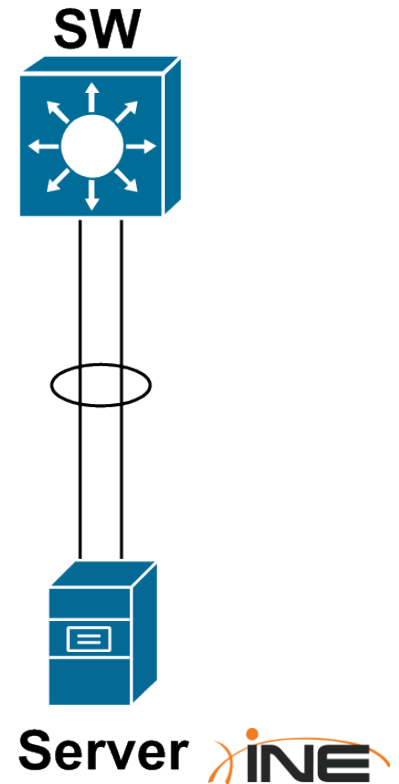
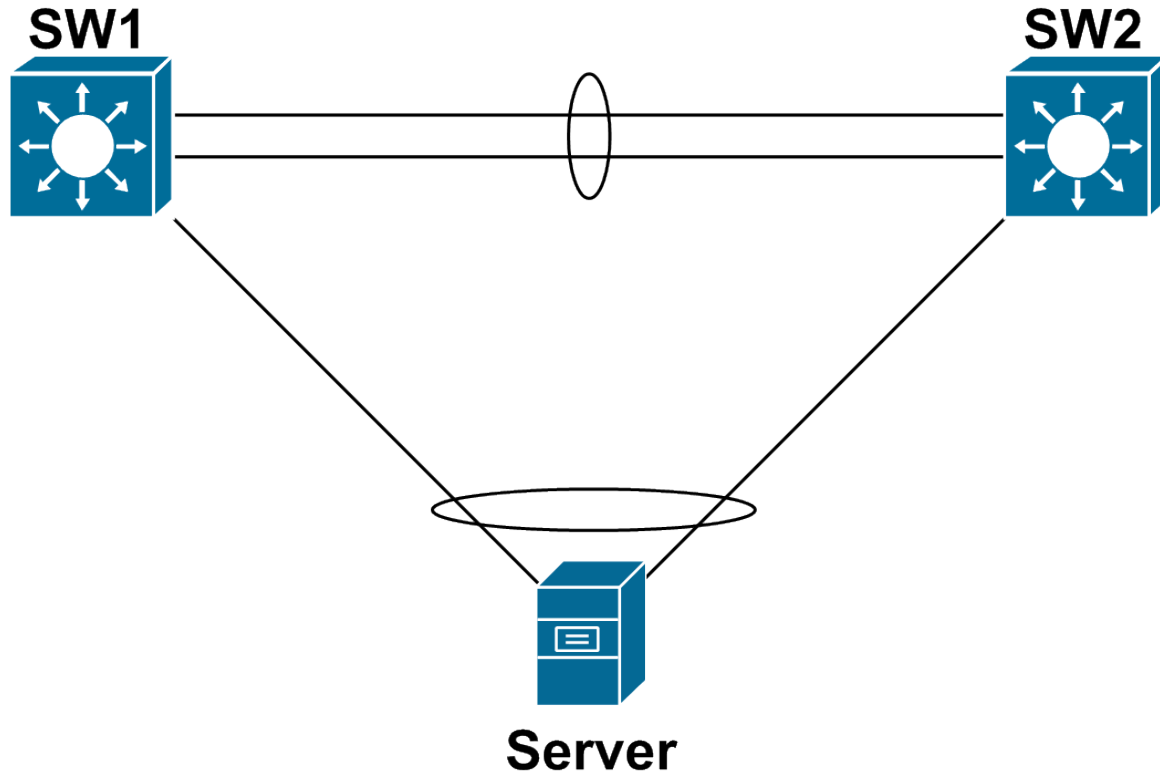
EtherChannel Pros & Cons

- + Pro – cheap incremental upgrade solution
 - + E.g. 2 x 10GigE likely cheaper than moving to 40GigE or 100GigE
- + Pro – adds link layer redundancy
 - + E.g. 4 x 10GigE likely has better resiliency than 1 x 40GigE
- + Con – flows cannot exceed the bandwidth of an individual link
 - + E.g. 2Gbps LAG is not a 2Gbps pipe, but 2 x 1Gbps pipes
 - + LAG adds lanes to the highway but doesn't increase the speed limit
- + Con – flows can get polarized to one member of the LAG
 - + LAG supports load distribution but not load balancing
 - + E.g. 1 x 40GigE gives better throughput than 4 x 10GigE

Single vs. MultiChassis EtherChannel

- + EtherChannels cause fate sharing at the access layer
 - + E.g. servers may have redundant power, redundant NICs, but what happens if the access switch goes down?
- + One solution - use two access layer switches
 - + Problem is that NICs usually only support active/standby outside of LAG
 - + I.e. 50% of bandwidth is lost unless the application supports load distribution at upper layers
- + Better solution – trick the server into running LAG
 - + Form a logical chassis between two physical switches
 - + Result is that the server thinks it has two connections to the same switch
 - + 100% of bandwidth can be used, and more resiliency is added

MultiChassis EtherChannel Physical vs. Logical



How MultiChassis EtherChannel (MCEC) Works

- + Goal is to turn physical triangle into logical P2P link
 - + Downstream box thinks upstream boxes are one
 - + This is what allows active/active
- + MCEC/MLAG relies on synchronized control plane between switches
 - + Forward the data plane out member ports of both chassis
 - + Synchronize the control plane on the backend transparently
 - + Synchronization is proprietary and not inter-operable

Cisco's MCEC Implementations

- + StackWise Cross-Stack EtherChannel
 - + Access platforms, e.g. Catalyst 3750/3850
 - + Control plane sync over dedicated stacking cables
 - + Stack cable creates a bidirectional closed-loop
 - + One control plane is shared among stack members
 - + One management plane is shared among stack members

Cisco's MCEC Implementations (cont.)

- + Virtual Switching System (VSS)
 - + Aggregation platforms, e.g. Catalyst 4500/6500/6800
 - + Control plane synch over Virtual Switch Link (VSL)
 - + Typically 2 x 10GigE LAG
 - + One control plane is shared among VSS
 - + One management plane is shared among VSS
 - + Usually 1 active supervisor and 3 standby supervisors

Cisco's MCEC Implementations (cont.)

- + Virtual Port Channel (vPC)
 - + Data Center platforms, e.g. Nexus 5000/7000/9000
 - + Control plane synchronized over a vPC Peer Link
 - + Typically 2 x 10GigE LAG
 - + ***Two independent control planes*** in the vPC
 - + ***Two independent management planes*** in the vPC
 - + Usually 2 active supervisors and 2 standby supervisors

StackWise vs. VSS vs. vPC

- + StackWise can have more than 2 members, up to the stack limit
 - + Stack limit depends on platform, e.g. 9 on 3750-X
- + VSS and vPC are always a pair of switches
- + Logical result of all three is the same
 - + Turns a physical triangle into a logical P2P link

EtherChannel Negotiation Protocols

- + LAG goal is to trick STP into forwarding active/active
 - + Risk is now an infinite loop in the data plane
 - + Negotiation protocols help to mitigate this risk
- + LAG negotiation comes in two forms
 - + Port Aggregation Protocol (PAgP)
 - + Link Aggregation Control Protocol (LACP)
- + Static LAG is supported but not recommended
 - + Failure to LAG can cause an STP loop
 - + EtherChannel Guard can help mitigate this

PAgP vs. LACP

- + PAgP
 - + Cisco Proprietary
 - + Essentially pre-standard LACP
- + LACP
 - + Standard per IEEE 802.3ad Link Aggregation (LAG)
- + Functionality is the same
 - + Think ISL vs. 802.1q

EtherChannel Negotiation Modes

- + LAG is configured as...
 - + channel-group [number] mode [mode]
- + Mode determines how negotiation occurs
 - + On
 - + No negotiation
 - + Desirable & Auto
 - + Initiate or listen for PAgP
 - + Active & Passive
 - + Initiate or listen for LACP

EtherChannel Mode Compatibility

- + On – On
- + Desirable – Desirable
- + Desirable – Auto
- + Active – Active
- + Active – Passive

EtherChannel Load Balancing

- + LAG driver is responsible for load distribution amongst members
 - + I.e. STP & CAM table see LAG as outgoing interface, not members
- + Available load balancing methods are per-platform, such as...
 - + Source & Destination MAC address
 - + Source & Destination IP address
 - + Source & Destination Layer 4
- + Balancing method is locally significant and outbound
 - + E.g. mismatching on either end is okay
 - + Adjustments should be made based on traffic patterns to prevent traffic polarization

Layer 2 vs. Layer 3 EtherChannel

- + LAG is independent of the port mode
 - + E.g. LAG can be access, trunk, tunnel, layer 3, etc.
- + LAG commonly suffers from order of operations issues
 - + Members and LAG interface must agree on parameters
 - + E.g. both members and LAG are layer 2 or layer 3, but no mix





Switched Campus

Spanning-Tree Protocol



In This Section

- + Understanding STP Root Bridge Election
- + Understanding STP Root Port Election
- + Modifying STP Path Selection
- + Understanding STP Timers

How STP Works

- + Elect one Root Bridge
- + Elect one Root Port per bridge
- + Elect Designated Ports

Root Bridge Election

- + Switch with lowest Bridge ID in the network becomes Root Bridge
- + Bridge ID contains...
 - + Bridge Priority
 - + 0 - 61440 in increments of 4096
 - + System ID Extension
 - + 0 - 4095
 - + MAC Address

Changing the Root Bridge Election

- + Manually change BID priority
 - + `spanning-tree vlan [vlan] priority`
 - + Lower is better
- + Use root bridge macro
 - + `spanning-tree vlan [vlan] root [primary | secondary]`
 - + Sets local priority based on current Root Bridge
- + Verification
 - + `show spanning-tree vlan [vlan]`
 - + `show spanning-tree root`

Root Port Election

- + RP is upstream facing towards Root Bridge
- + Elected based on lowest Root Path Cost
 - + Cumulative cost of all links to get to the root
- + Cost based on inverse bandwidth
 - + i.e. higher bandwidth, lower cost
 - + Not linear
- + If tie in cost...
 - + Choose lowest upstream BID
 - + Choose lowest upstream Port ID

Designated Port Election

- + DPs are downstream facing away from Root Bridge
- + Like Root Port, elected based on...
 - + Lowest Root Path Cost
 - + Lowest BID
 - + Lowest Port ID
- + All other ports go into blocking mode
 - + Receive BPDUs
 - + Discard all other traffic
 - + Cannot send traffic

Changing the Port's Role

- + Modify the port's cost
 - + spanning-tree [vlan] cost
 - + bandwidth [bps]
- + Modify the Bridge ID
 - + spanning-tree vlan [vlan] priority
- + Modify the Port ID
 - + spanning-tree vlan [vlan] port-priority
- + Verification
 - + show spanning-tree interface [int] detail
 - + show spanning-tree vlan [vlan] detail

STP Timers

- + Timers affect the transition between port states
 - + Set only on the Root Bridge
- + Hello
 - + How often configuration BPDUs are sent
 - + Defaults to 2 seconds
- + MaxAge
 - + How long to wait in blocking state without hearing a BPDU
 - + Defaults to 20 seconds
- + Forward Delay
 - + How long to wait in each the listening and learning phases
 - + Defaults to 15 seconds

Changing STP Timers

- + Configuration
 - + spanning-tree vlan [vlan] hello-time
 - + spanning-tree vlan [vlan] forward-time
 - + spanning-tree vlan [vlan] max-age
- + Verification
 - + show spanning-tree vlan [vlan]





Switched Campus

Optional Spanning-Tree Protocol Features



In This Section

- + STP Convergence Optimizations
 - + PortFast
 - + UplinkFast
 - + BackboneFast
- + STP Filters
 - + BPDU Filter
 - + BPDU Guard
 - + Root Guard
- + STP Loop Prevention
 - + Loop Guard
 - + UDLD

Legacy STP Convergence Optimizations

- + PortFast
 - + Edge ports shouldn't be subject to Forwarding Delay
 - + Also effects TCN generation
- + UplinkFast
 - + Direct Root Port failure should reconverge immediately if Alternate Port available
- + BackboneFast
 - + Indirect failures should start recalculating immediately

STP Filters

- + BPDU Filter
 - + Filter BPDUs in and out
- + BPDU Guard
 - + If BPDU is received shut port down
- + Root Guard
 - + If superior BPDU is received shut port down
- + Filters can be configured globally in conjunction with PortFast

Loop Prevention

- + STP Loop Guard
 - + Prevent unidirectional links by using BPDU keepalives
- + Unidirectional Link Detection (UDLD)
 - + Prevent unidirectional links by using UDLD keepalives





Switched Campus

Rapid Spanning-Tree Protocol (RSTP)



In This Section

- + What is RSTP?
- + How does RSTP Work?
- + RSTP Configuration

What is RSTP?

- + Rapid Spanning-Tree Protocol
 - + New standard originally defined in IEEE 802.1w
 - + Now incorporated as IEEE 802.1D-2004
- + Changes vs. legacy STP
 - + Simplifies port states
 - + Additional port roles
 - + Rapid convergence based on synchronization process
 - + Path calculation remains the same

RSTP Port States

+ Legacy STP uses...

- + Disabled
- + Blocking
- + Listening
- + Learning
- + Forwarding

+ RSTP simplifies to...

- + Discarding
 - + Dropping frames
- + Learning
 - + Dropping frames but building the CAM
- + Forwarding
 - + Normal forwarding

RSTP Port Roles

- + Port Roles are decoupled from port states
- + Root Port & Designated Port
 - + Same as before
- + New roles
 - + Alternate
 - + Backup
 - + Edge

RSTP Alternate Ports

- + Alternate but less desirable path to the root
- + Allows the equivalent of UplinkFast
 - + I.e. fast root path recovery
 - + Automatic, does not require uplinkfast command
- + Operates in discarding state

RSTP Backup Ports

- + Backup Designated Port
- + Activates if the primary Designated Port fails
- + Operates in discarding state

RSTP Edge Ports

- + Equivalent of PVST+ PortFast enabled ports
 - + Immediately transitions to forwarding
 - + Do not generate TCN for state change
 - + Configured with **spanning-tree portfast** command for backwards compatibility
- + Maintains edge status as long as no BPDUs are received
 - + If BPDU received, remove edge status and generate TCN

RSTP Link Types

- + Non-edge ports fall into two types
- + Point-to-point
 - + Full-Duplex ports
- + Shared
 - + Half-Duplex ports
- + Only point-to-point Designated Ports use the sync process for rapid convergence

RSTP Sync Process

- + Goal is for a bridge to synchronize its root port with the rest of the topology
- + When a bridge elects a root port it assumes all non-edge ports to be designated
 - + All non-edge ports are discarding at this moment
- + Bridge sends proposals out all designated ports
 - + Proposal has port role set to designated
 - + Proposal contains root bridge info (priority, cost, etc.)
- + Downstream bridges review this information
 - + If they don't have better paths to the root they agree
 - + If they do have it they announce their information

RSTP Sync Process (cont.)

- + When designated port receives agreement, it is unblocked
- + If downstream bridge sends better root information, local bridge changes root port
- + If downstream bridge agrees to upstream proposal, then it
 - + Elects a local root port
 - + Blocks all non-edge designated ports
 - + Starts sync process on all designated ports
- + Port blocking is essential in preventing transient loops
- + Sync process ensures all bridges agree on the same root bridge

RSTP Fault Detection

- + In Legacy STP, BPDUs are only generated by Root Bridge
 - + All other bridges forward them on
- + In RSTP, each bridge generates BPDU every hello interval
 - + 2 seconds by default
- + If 3 hellos are missed from a neighbor, reconvergence begins
 - + 6 seconds vs. 20 seconds MaxAge

RSTP Fault Detection (cont.)

- + MaxAge is used as hop count
 - + Every bridge sends BPDUs on its own
 - + Age incremented by every bridge
 - + MaxAge also used on shared ports for legacy STP backwards compatibility
- + Faults can be detected faster by means of physical layer signaling

RSTP Convergence

- + RSTP needs to re-converge when Root port is lost
- + If there is an Alternate port, it is selected in place of old Root port
 - + New Root port is then synchronized with downstream bridges
- + If there are no Alternate ports and no better info
 - + Declare itself as root
 - + Synchronize this decision
 - + Possibly adapt to better information

RSTP Convergence (cont.)

- + Non-deterministic, depends on topology
 - + Meshy and large topologies converge slow
- + Root bridge failures may cause slow convergence
 - + “Understanding and Mitigating the Effects of Count to Infinity in Ethernet Networks”
 - + <http://www.cs.rice.edu/~eugeneng/papers/TON-Ethernet.pdf>
- + To ensure fast convergence
 - + Keep topology small and avoid excessive redundancy
 - + Rely on physical layer failure detection not the Hello BPDUs

RSTP Topology Change

- + Generated when link becomes forwarding
 - + Originated by the switch that detected the event
 - + Uses special BPDU bit to signal topology change
 - + Flooded by all switches using reverse path forwarding
- + Flushes MAC address tables
 - + Causes temporary excessive unicast traffic flooding
 - + Use Edge Ports as much as possible

RSTP Configuration

- + Enable RSTP
 - + **spanning-tree mode rapid-pvst**
 - + Automatically backwards compatible with legacy STP
- + Sync only occurs on P2P non-edge ports
 - + Implies link-type must be accurate
 - + **spanning-tree link-type [point-to-point|shared]**
 - + **spanning-tree portfast [trunk]**
- + Path selection remains unchanged
 - + Root bridge election
 - + Root port & designated port elections





Switched Campus

Multiple Spanning-Tree Protocol



In This Section

- + What is MST?
- + How Does MST Work?
- + Intra-Region vs. Inter-Region MST
- + MST Interoperability
- + MST Configuration

What is MST?

- + Multiple Spanning-Tree Protocol
 - + Started as Cisco's MISTP
 - + Originally standard defined in IEEE 802.1s
 - + Now standard per IEEE 802.1Q-2005

How Does MST Work?

- + MST works by decoupling VLAN & STP Instance
 - + STP Instance to VLAN mapping is user defined
 - + Topology calculation done by RSTP
- + Result is higher scalability
 - + (Rapid) PVST+ uses one instance per VLAN
 - + As VLANs scale, control plane dies
 - + PVST is inefficient because there are typically only 3 possible trees anyways

MST Regions

- + MST defines a Region as bridges that agree upon...
 - + Instance name
 - + Revision number
 - + VLAN to STP instance mappings

Intra vs Inter Region

+ Intra Region

- + Details of the region are known within the region
- + VLAN to STPIs are manually defined
- + Undefined VLANs fall into CIST (MST 0)

+ Inter Region

- + Details between regions are not known
- + Different regions see each other as virtual bridges
- + Result is simplified Inter-Region calculation
- + Intra-region MSTIs are collapsed into CIST

MST Interoperability

- + MST is backwards compatible with legacy CST and PVST+
- + Behaves like Inter-Region MST
- + CST Root must be within MST domain

MST Configuration

- + Define the following in MST configuration mode
 - + Region name
 - + Revision number
 - + VLAN to instance mappings
- + Enable MST globally
 - + Real deployment must start at Root and work out

MST Path Selection

- + Same election process as CST/PVST
 - + Root bridge
 - + Lowest BID
 - + Root port
 - + Lowest cost
 - + Lowest upstream BID
 - + Lowest port ID

Changing MST Root Bridge Election

- + Manually change BID priority
 - + `spanning-tree mst [instance] priority`
 - + Lower is better
- + Use root bridge macro
 - + `spanning-tree mst [instance] root [primary | secondary]`
 - + Sets local priority based on current Root Bridge
- + Verification
 - + `show spanning-tree mst [instance]`
 - + `show spanning-tree root`

Changing an MST Port's Role

- + Modify the port's cost
 - + `spanning-tree mst [instance] cost`
 - + `bandwidth [bps]`
- + Modify the Bridge ID
 - + `spanning-tree mst [instance] priority`
- + Modify the Port ID
 - + `spanning-tree mst [instance] port-priority`
- + Verification
 - + `show spanning-tree interface [int] detail`
 - + `show spanning-tree mst [instance] detail`

