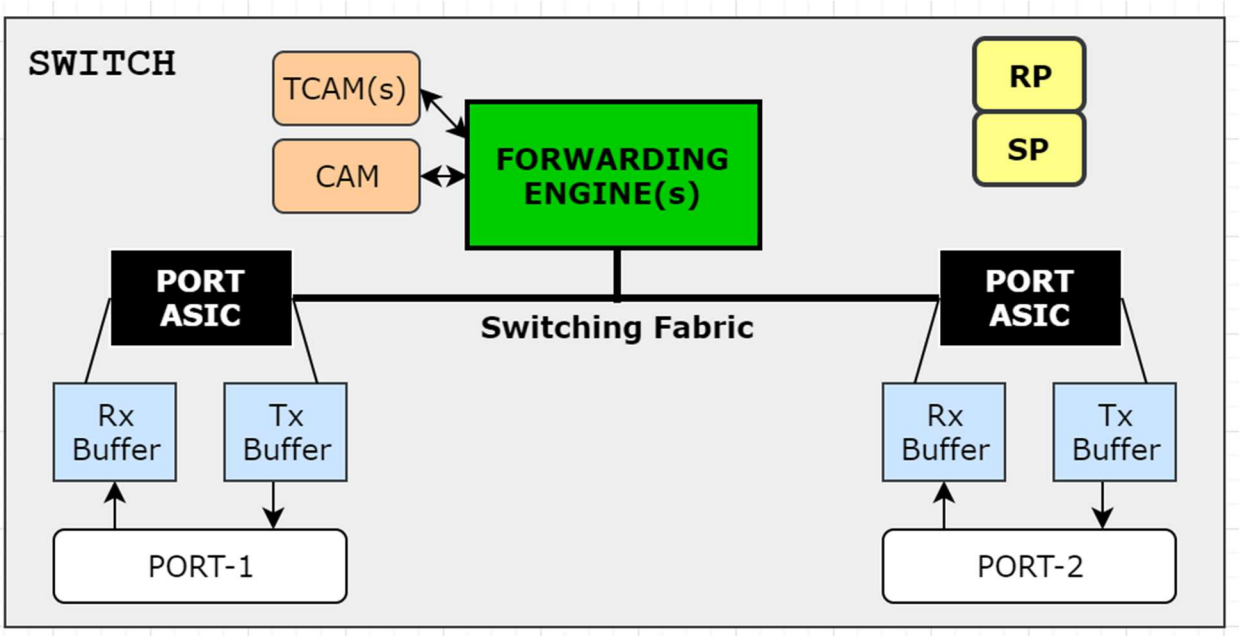
CCNP Switch Master StudyGuide

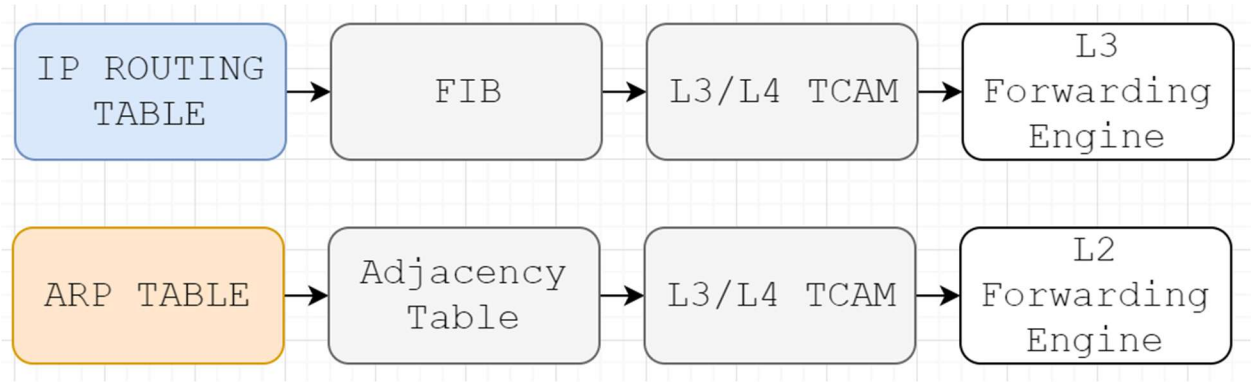


1.0 Layer 2 Technologies

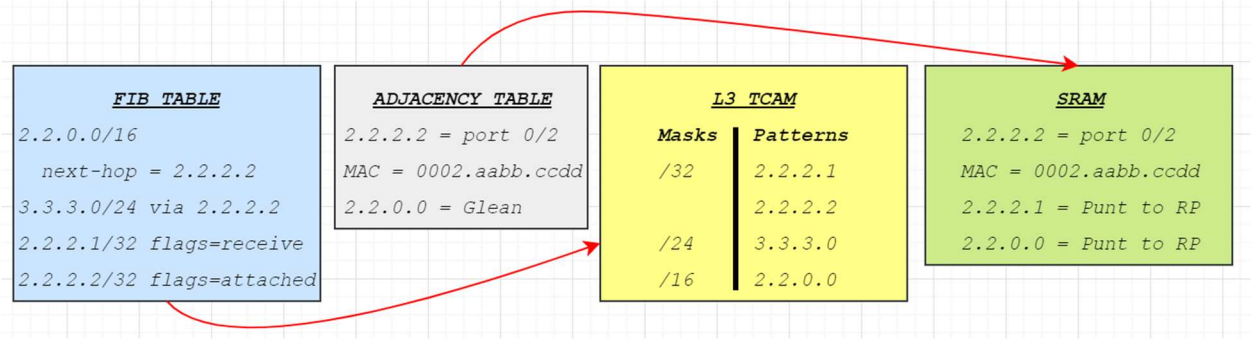
1.1 Configure and verify switch administration

- **THREE PRIMARY SWITCHING METHODS**
  - **PROCESS-BASED SWITCHING**
    - Tasks are switched using CPU
    - This process is called *IP Input*
  - **FAST SWITCHING**
    - First packet would hit CPU
    - Fast Switching Cache would be populated
  - **CISCO EXPRESS FORWARDING**
    - Prepopulated L2 and L3 caches
    - CPU is rarely bothered
    - Tracks the ARP and Routing tables
- **CEF A.K.A. "CISCO EXPRESS FORWARDING"**
  - Known as *switching* within routers and switches
  - Enabled by default on Multilayer Switches
  - Composed two components: **FIB** and **Adjacency Table**

- **FIB A.K.A. "FORWARDING INFORMATION BASE"**
  - Copy of IP Routing Table and tracks information for automatic updates
  - Does not contain everything from the IP Routing Table
  - Commonly referred to as a **Shadow Copy** of the IP Routing Table
  - Only contains the minimum necessary information required to switch data
- **ADJACENCY TABLE**
  - Populated by the L2 tables such as ARP and Frame-Relay Mapping
  - Some lookups cannot be CEF switched and must be dropped or sent to CPU
- **ADJACENCY TYPES**
  - **GLEAN**
    - No specific information about an address in a subnet
    - Requires packet forwarding to CPU to trigger ARP reply
    - Temporary state until address is attached
  - **NULL**
    - Valid packet that must be dropped
    - Destined for the Null0 Interface
  - **DROP**
    - Packets that must be dropped
    - This could be because of bad encapsulation or no route
  - **DISCARD**
    - Packets that must be dropped
    - This could be because of a security policy such as an ACL
  - **PUNT**
    - Requires packet forwarding to CPU
    - This is because the packet is destined for the CPU itself
    - This could be because of a routing update from a neighbor
- **RELATED CISCO COMMANDS**
  - `Switch# show ip cef [detail]`
  - `Switch# show ip cef <ip-address> <mask> detail`
  - `Switch# show adjacency <interface> [summary|detail]`
  - `Switch# show adjacency vlan <vlan-id> detail`
- **ASIC A.K.A. "APPLICATION SPECIFIC INTEGRATED CIRCUIT"**
  - Piece of hardware with millions of transistors
  - Looks up information thousands of times faster than a processor
  - Stores frame information in the TCAM and searches them up using keys
  - CEF is responsible for providing the information that goes into the TCAM



- **SWITCH FABRIC CONCEPT**
  - The **ethernet** within the switch
  - There is a bus to get to the forwarding engine
- **SWITCH FABRIC BUS TYPES**
  - **SHARED BUS**
    - All port ASICs share the same line to transfer data to Forwarding Engine
    - Each port ASIC must wait till given permission to forward data
    - The first image in this page represents this switch fabric type
  - **SHARED RING**
    - All port ASICs are Forwarding Engines
    - The ring is how they forward data to each other
    - Only one ASIC controls the ring at a time
    - This control is determined through the pulse that must be captured
- **TCAM A.K.A. "TERNARY CONTENT ADDRESSABLE MEMORY"**
  - Memory holds three possible values: **0** (off), **1** (on), **x** (don't care)
  - Performs longest match lookups specifically for Layer 3
  - The memory is divided into two sections...
    - **Masks** (similar to subnet/wildcard masks)
    - **Patterns** (similar to IP addresses)
  - The matching TCAM entry will point to adjacency entry in SRAM
  - Impossible to see the contents of the TCAM table but SHOULD match CEF
  - **Feature Manager** is responsible for downloading configured features into TCAM



1.1.a Managing MAC address table

- **CAM A.K.A. "CONTENT ADDRESSABLE MEMORY"**

    - Every bit in lookup key must find an exact match
    - The default behavior when no match is found is to flood
    - Specifically used for MAC Address lookups
- **MAC ADDRESS TABLE USAGE**
    - Static MAC Address
    - Dynamic MAC Address
    - MAC Address Age Timer (5 minutes by default)
- **RELATED CISCO COMMANDS**
    - `Switch(config)# mac address-table aging-time <secs>`
    - `Switch(config)# mac address-table static <mac> vlan <id> interface <interface>`
    - `Switch# show mac address-table <dynamic> <address|value>`

1.1.b Troubleshoot Err-disable recovery

- **ERROR DISABLE FEATURE**
    - Many different features can place a port into the error-disabled state
        - BPDUGuard
        - ARP Inspection
        - DTP flap (trunk encapsulation is changing)
        - LACP flap (etherchannel ports no longer identical)
        - And more...
    - A port will remain error-disabled until administratively reactivated
- **RELATED CISCO COMMANDS**
    - `Switch(config)# no errdisable detect cause {all | cause-name}`
    - `Switch(config)# errdisable recovery cause {all | cause-name}`
    - `Switch(config)# errdisable recovery interval <secs> (default 300)`

1.2 Configure and verify Layer 2 protocols

1.2.a CDP, LLDP

- **CDP A.K.A. "CISCO DISCOVERY PROTOCOL"**
    - Allows you to see directly connected cisco devices
    - Cisco proprietary protocol
    - Also used for other communication like POE negotiation
- **LLDP A.K.A. "LINK LAYER DISCOVERY PROTOCOL"**
    - Industry standard and not proprietary
    - Non-Cisco hosts support usage of LLDP
- **RELATED CISCO COMMANDS**
    - `Switch(config-if)# no cdp enable`
    - `Switch(config-if)# no lldp receive`
    - `Switch(config-if)# no lldp transmit`
    - `Switch# show cdp entry <name>`
    - `Switch# show cdp detail`
    - `Switch# show cdp neighbor`

1.2.b UDLD

- **UDLD A.K.A. "UNI-DIRECTIONAL LINK DETECTION"**
  - Layer-2 Protocol, Cisco Proprietary
  - Not dependent on BPDUs to detect problems
  - Originally designed for Fiber-Optic Links
  - Sends CDP-like messages that **MUST** be echoed back
    - Remote side of link adds info about itself before reflection
    - Messages sent every 15-seconds (default)
  - Must be configured on both sides of the link
- **UDLD MODES**
  - **NORMAL**
    - Prints out a syslog message and marks port undetermined
    - Pretty much useless, do not bother using this
  - **AGGRESSIVE**
    - Discovers neighbor when port comes online
    - Error-disable port with three missed messages
- **RELATED CISCO COMMANDS**
  - `Switch(config)# udld {enable | aggressive | message-time-seconds}`
  - `Switch(config-if)# udld {enable | aggressive | disable}`
  - `Switch# show udld neighbors`
  - `Switch# show udld <interface>`
  - *If enabled globally, only applied to fiber-optic links*

1.3 Configure and verify VLANs

1.3.a Access ports

- **VLAN A.K.A. "VIRTUAL LOCAL AREA NETWORK"**
  - Breaks hosts up into different broadcast domains
  - Allows extension of VLANs between switches
  - **Access Ports** are single vlan untagged interfaces
- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# switchport mode access`
  - `Switch(config-if)# switchport access vlan <vlan-id>`
  - `Switch# show vlan <vlan-id>`

1.3.b VLAN database

- **VLAN DATABASE CONCEPT**
  - There are two methods to configure VLANs
  - The legacy method is with the VLAN database
  - The vlan information is stored in the **vlan.dat** file

1.3.c Normal, extended VLAN, voice VLAN

- **NORMAL VLANS**
  - VLAN range is 1 - 4094
  - 1 - 1001 are the usable normal-range VLANs
  - 1002 - 1005 are reserved for token ring
- **EXTENDED VLANS**
  - 1006 - 4094 are extended-range VLANs
  - **MUST** be in VTP Transparent Mode
    - VTPv3 can be used to propagate extended VLANs
    - Most switches do not support VTPv3 however
    - If Server/Client, extended VLANs must be manually deleted
    - This includes manually re-configuring switch ports
  - **NOT** stored in **vlan.dat** file
- **VOICE VLAN**
  - Useful for isolating user data and voice traffic in the same interface
  - Both data and voice VLANs must be explicitly configured
  - Two methods IP Phones can use to learn about the Voice VLAN
    - CDP for Cisco IP Phones
    - DHCP Option 156 for non-Cisco Phones
  - IP Phone works as a switch between PC and switch
    - 802.1Q trunk between switch and IP phone
  - **QoS** is usually implemented to prioritize voice traffic over data
- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# switchport voice vlan <vlan-id | dot1p | untagged | none>`
  - ***vlan-id*** → *CDP: Use VLAN-X for Voice, Data = untagged only.*
  - ***dot1p*** → *CDP: Use VLAN-0 for Voice, Data = allow 802.1p*
  - ***untagged*** → *CDP: Use VLAN-4095 for Voice, Data = untagged only.*
  - ***none*** → *CDP: No Voice VLAN, Data = untagged only.*
  - `Switch# show interface <interface> switchport`
- **VOICE QOS CONCEPT**
  - Voice QoS solves two major issues
    - Predictability for voice traffic
    - Jitter and delay
  - Done by trying to minimize Loss, Delay, and Jitter
  - FYI - **Chronic Congestion** implies a design problem and QoS won't help
  - **Expedited Forwarding** (technically AF46) in QoS matches against Voice traffic

1.4 Configure and verify trunking

1.4.a VTPv1, VTPv2, VTPv3, VTP pruning

- **VTP A.K.A. "VLAN TRUNKING PROTOCOL"**
  - Cisco proprietary protocol
  - Create and propagate VLAN information on all switches in the domain

- VTP passwords are never displayed in running-config
- Higher revision database wins at least with VTPv2 and VTPv1
- VTP Pruning reduces broadcasts thus reducing congestion
- **MINIMUM REQUIREMENTS**
  - All switches in the same VTP Domain
  - Switches connected via VLAN Trunks
  - Same VTP Password
  - Same version of VTP must communicate across a link

- **VTP VERSIONS**
  - **VTPv1**
    - Introduced original concept of VTP
  - **VTPv2**
    - Introduced token ring support
  - **VTPv3**
    - No auto setup (NULL)
    - All VLAN numbers supported
    - 802.1s MST configuration propagation support
    - Enhanced VTP password security with hidden instead of clear-text
    - Primary Server concept
    - Private VLAN support
    - Backward compatible with Version 2 on a per-link basis
- **VTP SWITCH TYPES**
  - **SERVER**
    - Switches which can add or remove VLANs
  - **CLIENT**
    - Switches which can only receive VLAN updates
  - **TRANSPARENT**
    - Switches that hear VTP messages, but only passes it through
  - **OFF (VTPv3 ONLY)**
    - Same as transparent, but rejects VTP messages instead of passing it
- **VTP VERSION 3 SERVERS**
  - **SECONDARY SERVER (DEFAULT)**
    - Similar to VTP Client, does not allow manual addition/deletion of VLANs
    - Not allowed to update VLAN database of other devices
  - **PRIMARY SERVER**
    - Only one per VTP Domain
    - Only device in Domain allowed to update VLAN Database
    - Only device upon which VLANs may be added or removed manually
    - To promote a new device as primary server, must demote existing
- **RELATED CISCO COMMANDS**
  - `Switch# vtp version 2 (change version)`
  - `Switch(config)# no vtp (VTPv3 only)`
  - `Switch(config-if)# no vtp (VTPv3 only)`
  - `Switch# vtp password <password> (clear-text)`
  - `Switch(config)# vtp password <password> (clear-text)`

- `Switch(config)# vtp password <password> hidden (generates 32-hex password)`
- `Switch(config)# vtp password <32-Hex password> secret`
- *Last option can be used if copying generated hidden password from other devices*
- `Switch# show vtp password`
- `Switch(config)# vtp mode transparent`
- `Switch# vtp primary vlan (set up primary server, VTPv3 only)`
- `Switch(config)# vtp pruning (enable VTP pruning)`
- `Switch# show vtp status`

1.4.b dot1Q

- **TRUNKING CONCEPT**
  - Tags traffic with VLAN information to segregate across physical links
  - **802.1q**: 32-Bit Tag Field and Internet Standard
  - Manual control of which VLANs are allowed on a trunk is possible
- **DTP A.K.A. "DYNAMIC TRUNKING PROTOCOL"**
  - Cisco proprietary protocol
  - Automatically configures trunks between switches
  - **AUTO**
    - Passively convert to trunk if neighbor wishes to
  - **DESIRABLE**
    - Actively attempt to convert link to trunk
- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# switchport mode <auto | desirable | trunk>`
  - `Switch(config-if)# switchport no negotiate`
  - `Switch(config-if)# switchport trunk allowed vlan { vlans | all | add | remove}`

1.4.c Native VLAN

- **NATIVE VLAN CONCEPT**
  - Generally, on a trunk link, VLANs are tagged
  - However, there is always one VLAN on that trunk that is not tagged
  - This is called the native VLAN and by default is VLAN 1
- **RELATED CISCO COMMANDS**
  - `Switch# show interface trunk`
  - `Switch(config-if)# switchport trunk native <vlan-id>`

1.4.d Manual pruning

- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# switchport trunk allowed vlan remove <vlan-list>`
  - *Instead of relying on VTP, you can also manually remove VLANs from a trunk*
  - *Note that this option will only take effect if VTP Pruning is disabled*
  - `Switch(config-if)# switchport trunk pruning vlan remove <vlan-list>`
  - *This removes VLANs **from** being pruned via VTP Pruning (so a double negative!)*

1.5 Configure and verify EtherChannels

1.5.a LACP, PAgP, manual

- **PAgP A.K.A. "PORT AGGREGATION PROTOCOL"**
  - Cisco Proprietary
  - The modes are **auto** and **desirable**
  - Works on half-duplex, max of 8 active ports
- **LACP A.K.A. "LINK AGGREGATION CONTROL PROTOCOL"**
  - Internet Standard
  - The modes are **passive** and **active**
  - Works on full-duplex, max of 8 active ports
  - Allows additional ports on standby
  - Concept of system priorities and port priorities, same as STP
    - Default priority is 32768
    - Lower number is better
    - This is used to determine LACP Master
- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# channel-group <#> mode {auto|desirable|passive|active|on}`
  - `The "on" option configures a manual etherchannel; does not send control packets`

1.5.b Layer 2, Layer 3

- **DIFFERENCES BETWEEN LAYER 2 AND LAYER 3**
  - Main benefit of etherchannel is redundancy
  - With Layer 2 etherchannels, there is just one broadcast domain
  - This means there is a chance of it becoming clogged up
  - Suggest: New broadcast domain by providing etherchannel with separate vlan
    - Spanning-tree convergence is still an issue
    - Losing one link will result in a new cost for the logical link
    - There is still an issue with data flooding
  - With a Layer 3 etherchannel, these issues do not exist
    - Separate broadcast domain
    - No STP
- **RELATED CISCO COMMANDS**
  - `LAYER 2 CONFIGURATION`
  - `Switch(config)# interface port-channel <#>`
  - `Switch(config-if)# switchport mode trunk`
  - `Switch(config-if)# switchport trunk encapsulation dot1q`
  - `Switch(config)# interface range Eth1/1 - 4`
  - `Switch(config-if)# channel-group <#> mode desirable`
  - `LAYER 3 CONFIGURATION`
  - `Switch(config)# interface port-channel <#>`
  - `Switch(config-if)# no switchport`
  - `Switch(config-if)# ip address <ip-address> <subnet>`
  - `Switch(config)# interface range Eth1/1 - 4`
  - `Switch(config-if)# channel-group <#> mode desirable`

1.5.c Load balancing

- **GENERAL CONCEPT**
  - Etherchannels generally load balance on a per-flow basis round-robin style
  - This mean if a single flow exceeds a single port's bandwidth, there will be drops
  - In other words, etherchannel does not necessarily give you additional bandwidth
  - There is a hash algorithm that determines load per port on a port-channel
  - Generally recommended to have **2^x** number of ports in a port-channel
  - Otherwise, some ports will be tasked with more load than the others
  - The exact algorithm used to differentiate flows can be adjusted
- **EXACT HASH LOAD**
  - **8 PORTS**
    - `1:1:1:1:1:1:1:1`
  - **7 PORTS**
    - `2:1:1:1:1:1:1`
  - **6 PORTS**
    - `2:2:1:1:1:1`
  - **5 PORTS**
    - `2:2:2:1:1`
  - **4 PORTS**
    - `2:2:2:2`
  - **3 PORTS**
    - `3:3:2`
  - **2 PORTS**
    - `4:4`
- **RELATED CISCO COMMANDS**
  - **CONFIGURING LOAD BALANCING ALGORITHM**
  - `Switch(config)# port-channel load-balance {src-mac | dst-mac | src-dst-mac...}`
  - **VERIFICATION**
  - `Switch# show etherchannel load-balance`

1.5.d EtherChannel misconfiguration guard

- **BACKGROUND INFORMATION**
  - Ports bundled into an Etherchannel share a single MAC address
  - All BPDUs transmitted in an etherchannel have the **SAME** Sending-Port-ID
  - **Etherchannel Misconfiguration Guard** can take advantage of this
- **ETHERCHANNEL MISCONFIGURATION GUARD**
  - On by default
  - Detects if remote end of Etherchannel is misconfigured by BPDU Port-ID
  - Places ports into error-disabled mode if misconfiguration detected
- **RELATED CISCO COMMANDS**
  - **CONFIGURING ETHERCHANNEL MISCONFIGURATION GUARD**
  - `Switch(config)# spanning-tree etherchannel guard misconfig`
  - *Should display syslog message about error*
  - **VERIFICATION**

- `Switch# show spanning-tree summary`

1.6 Configure and verify spanning tree

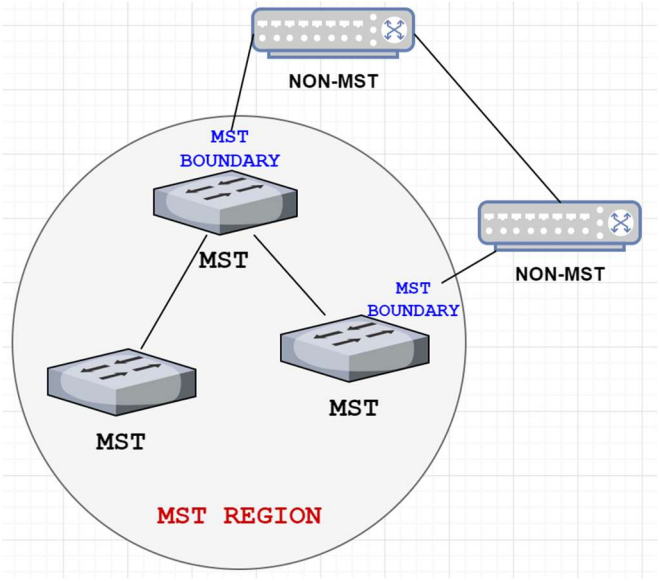
1.6.a PVST+, RPVST+, MST

- **PVST+ A.K.A. "PER VLAN SPANNING TREE"**
  - Per vlan version of 802.1d
  - Goal is to prevent bridging loops
  - Convergence time is 30 to 50 seconds
  - BPDU A.K.A. "Bridge Protocol Data Unit"
    - Provides Root ID
    - Provides Bridge ID
    - Provides cost to Root Bridge
  - Technically, two main types of BPDUs…
    - Configuration BPDUs which are **ONLY** generated by Root Bridge
    - Other bridges just relay the BPDU when it receives it from root port
    - TCN BPDU can be generated by all switches
  - Root Bridge Election Process
    - Lowest Bridge ID wins
    - This is either via priority or mac address
  - Topology change is triggered by…
    - Learning → Forwarding
    - Learning → Blocking
    - Port becomes disabled
    - *TCN is NOT triggered by a port configured with PortFast*
  - Triggers creation of TCN A.K.A. "Topology Change Notification"
    - Transmitted on Root Port until acknowledged
    - No indication in the TCN of what exactly changed
  - What does the Root Bridge do when receiving a TCN?
    - Send a TCA A.K.A. "Topology Change Acknowledgement" to TCN bridge
    - Sets "TC-Flag" when transmitting BPDUs to flood into that VLAN
    - Those BPDUs are now called Topology Change BPDUs
    - Continue transmitting TC-BPDUs for...
      - `Forwarding_Delay + Max_Age (35 sec)`
    - Reduces own CAM Aging time to Forwarding Delay for the affected VLAN
  - What do Non-Root Bridges do when receiving TC-BPDU?
    - Reduce CAM Aging Time = Forwarding-Delay
    - Any unheard of MAC Addresses during this time are flushed
  - Missing 10 BPDUs is equivalent to a dead neighbor
  - Uses Short Cost (See **1.6.b**)
- **RPVST+ A.K.A. "RAPID PER VLAN SPANNING TREE"**
  - Per vlan version of 802.1w
  - State changes were too slow in legacy STP

- Designed to speed up convergence
- Convergence time is by default 6 seconds or within a few milliseconds if P2P
- Link type is derived from duplex mode
    - **Full duplex link** is considered as P2P
    - **Half duplex link** is considered to be shared
- Originally, only Root Bridge sent BPDUs, now every switch creates them
    - Act as keepalives now
    - Goes both ways in a link, as even blocking state ports send BPDUs
- Missing 3 BPDUs is equivalent to a dead neighbor
- Default uses Short Cost, can be changed to Long Cost (See **1.6.b**)

- **RSTP PORT ROLES**
    - **ROOT PORT**
        - Port that has the best root path cost to the root
    - **DESIGNATED PORT**
        - Downstream port that has the best root path cost to the root
    - **ALTERNATE PORT**
        - Port that has an alternate path to the root
        - Can only listen to the BPDUs
    - **BACKUP PORT**
        - Considered as a backup designated port
        - Leads to the same collision domain as the other designated port
- **RSTP PORT STATES**
    - **DISCARDING**
        - Combines the 802.1d disabled, blocking, and listening states
        - No MAC addresses are learned, and incoming frames are dropped
    - **LEARNING**
        - Can’t send or receive data
        - MAC addresses are learned
    - **FORWARDING**
        - Can send and receive data
- **RSTP ROOT BRIDGE ELECTION PROCESS**
    - STP uses flags for topology change acknowledgement and notification
    - RSTP adds new flags: Proposal and Agreement
    - The P and A flags skip the whole forwarding and listening states
    - **PROPOSAL FLAG**
        - “I would like to be the root”
    - **AGREEMENT FLAG**
        - “I agree, you would be a better root than me”
- **MST A.K.A. “MULTIPLE SPANNING TREE”**
    - IEEE’s answer to Cisco’s Per-Vlan STP implementations
    - With PVST and RPVST, there is an instance of STP running per VLAN
    - With MST, there is one instance of STP running per MSTI...
        - One-or-more VLANs bundled into each MSTI
        - VLANs mapped to MSTI Instance-0 by default

        - Maximum of 16 instances supported
        - Each MSTI should represent a different **STP topology**
    - Like VTP, Bridges running MST must have certain compatible parameters...
        - MST Region Name
        - MST Revision Number (not dynamically set)
        - VLAN-to-Instance Mapping Configuration Digest
    - Creating separate STP topologies involve tuning STP variables per instance
    - With MST, there is just ONE BPDU, no matter how many instances
        - Instance-0 is in every single BPDU
        - Mrecord is all the information about an MSTI within an MST BPDU
        - Instance-0 does not count as an mrecord
    - Default uses Long Cost (See **1.6.b**)
- **MST AND INSTANCE 0**
    - There is an important reason why Instance-0 is always in the BPDU
    - Even if it is an access port for a different instance, Instance-0 will still be in BPDU
    - Instance-0 has all timer information
    - **Boundary Ports** = Ports unable to do MST with peer
        - Boundary Ports Tx/Rx must be non-MST BPDUs
        - BPDU information sent on Boundary Ports = Instance-0
        - Other instance VLANs take on Instance-0 parameters
        - Instance-0 is used to do backwards compatibility with CST and R/PVST+
    - **MST Internal Ports** = Ports that can communicate using MST BPDUs
    - Instance-0 is called the IST A.K.A. "Internal Spanning-Tree"
    - IST replicates Non-MST BPDU on MST Boundary
        - This includes PVST+ which has a BPDU per VLAN



- **MST ROOT ELECTION PROCESS**
    - **TWO ROOT BRIDGES**
        - Main Root Bridge (Root for the entire CST and Instance-0)
        - CIST Regional Root (Also called IST Master)

        - MST switches select their IST Root Port based on location of IST Master
    - **IST MASTER ELECTION**
        - If CST Root Bridge is elected inside MST Region…
            - CST Root = IST Master
        - If CST Root Bridge is located on non-MST switch…
            - IST Master = MST switch with lowest cost to Root
    - **MAIN ROOT BRIDGE ELECTION**
        - Same as regular STP Root Bridge Election with Priority
        - Lowest Bridge Priority wins
        - If Non-MST mixed with MST, then always use Bridge Priorities of IST
- **RELATED CISCO COMMANDS**
    - **MST CONFIGURATION**
    - `Switch(config)# spanning-tree mode mst`
    - `Switch(config)# spanning-tree mode configuration`
    - `Switch(config-mst)# revision <revision-number>`
    - `Switch(config-mst)# name <region-name>`
    - `Switch(config-mst)# instance <instance-#> vlan <vlan-list>`
    - `Switch(config)# spanning-tree mst <instance-#> {cost | port-priority}`
    - `Switch(config-if)# spanning-tree mst <instance-#> {cost | port-priority}`
    - **STP/RSTP CONFIGURATION**
    - `Switch(config)# spanning-tree vlan <#> priority <#>`
    - `Switch(config)# spanning-tree vlan <#> root primary`
    - `Switch# show spanning-tree details`
    - `Switch# show spanning-tree summary`
    - **RSTP CONFIGURATION**
    - `Switch(config)# spanning-tree mode rapid-pvst`
    - `Switch(config-if)# spanning-tree link-type point-to-point (enable P/A on half)`

1.6.b Switch priority, port priority, path cost, STP timers

- **SWITCH PRIORITY**
    - In every flavor of STP, **Lowest Bridge ID becomes ROOT**
    - The Bridge ID is a combination of the Priority and Mac Address
    - Default Bridge Priority is 32768
- **PORT PRIORITY**
    - Generally, the port priority is used to determine **Root Port**
    - IF there are two-or-more equal cost paths to Root Bridge…
        - Lowest Bridge ID of the neighbor wins
        - If same Bridge ID, lowest port priority wins
        - If same port priority, lowest port number wins
    - IF there are two-or-more bridges on same segment with equal-cost paths to Root
        - Lowest bridge ID will become the designated port in segment
        - The other port will go into blocking state
- **PATH COSTS**
    - Lowest path cost to Root Bridge becomes Root Port
    - This is determined by the bandwidth along the path, see table below...

BANDWIDTH	LONG COST 32-Bit	SHORT COST 16-Bit

10MB	2000000	100
100MB	200000	19
1GB	20000	4
10GB	2000	2

- **STP TIMERS**
  - **HELLO**
    - Time between each BPDU that is sent on a port
    - By default, this is 2 seconds, but this can be tuned
  - **FORWARD DELAY**
    - Time spent in the listening and learning state
    - By default, this is 15 seconds, but this can be tuned
  - **MAX AGE**
    - Maximum length of time before a bridge port saves its BPDU information
    - By default, this is 20 seconds, but this can be tuned

1.6.c PortFast, BPDUguard, BPDUfilter

- **PORTFAST**
  - Bypasses most of the STP states so the end hosts can turn up immediately
  - This should NOT be used between switches as it effectively disables STP
  - If a switchport receives a BPDU on a portfast link, portfast state is revoked
- **BPDUGUARD**
  - Blocks against any BPDU packets that may be received on a port
  - When a BPDU is received, the port goes into error-disabled state
  - This is an enhancement of Portfast
- **BPDUFILTER**
  - Prevents interfaces from sending or receiving BPDUs
  - Can be enabled either globally or per-interface
  - If a BPDU is received, portfast state is revoked and filter is disabled (global-only)
  - Enabling BPDU Filter is the same as disabling STP on the port (interface-only)
- **RELATED CISCO COMMANDS**
  - `Switch(config-if)# spanning-tree portfast`
  - `Switch(config)# spanning-tree portfast default`
  - `Switch(config-if)# switchport host (macro for portfast and other features)`
  - `Switch(config)# spanning-tree portfast bpduguard default`
  - `Switch(config-if)# spanning-tree bpduguard enable`
  - `Switch(config)# spanning-tree portfast bpdufilter default <- global`
  - `Switch(config-if)# spanning-tree bpdufilter enable <- interface`

1.6.d Loopguard and Rootguard

- **LOOPGUARD**
  - Generally, on a blocked state port, BPDUs are received
  - If BPDUs stop being received, the port normally goes into designated

    - This is because STP assumes there is no loop due to no BPDU
    - Loopguard prevents the blocked state port from going designated
    - Instead of going designated, it goes into **loop-inconsistent blocking state**
- **ROOTGUARD**
    - Provides a means to control placement of the Root Bridge
    - If a bridge receives a superior STP BPDU on a root guard enabled port…
        - The port is placed into root-inconsistent STP state
        - This is the same as the listening state where there is no forwarding
    - Any port that has root guard enabled will never become a root port
    - You should enable root guard on all ports where root bridge should not appear
- **RELATED CISCO COMMANDS**
    - `Switch(config)# spanning-tree loopguard default`
    - `Switch(config-if)# spanning-tree guard loop`
    - `Switch(config-if)# spanning-tree guard root`

1.7 Configure and verify other LAN switching technologies

1.7.a SPAN, RSPAN

- **SPAN A.K.A. "SWITCHPORT ANALYZER"**
    - Allows you to copy ingress traffic from port or VLAN
    - The copied frames are then sent to an egress port for observation
    - Also known as Local SPAN or Port SPAN
    - Source and Destination Ports must be on the same switch
    - **SOURCE PORTS**
        - One or more ports
        - Select direction (RX, TX, or Both), default is Both
        - One or more VLANs
        - Traffic to/from Switch CPU
    - **DESTINATION PORTS**
        - One or more ports
        - A port identified as SPAN Destination is in monitoring state
        - This means all other features are disabled on that port
        - By default, SPAN Destination Ports cannot receive any ingress frames
        - This behavior can be modified via configuration
- **RSPAN A.K.A. "REMOTE SWITCHPORT ANALYZER"**
    - Allows you to capture traffic on one switch and send it over to a remote switch
    - This involves utilizing a "Remote VLAN"
    - The source and destination switches must be connected together via trunk
    - When setting up a destination span across a remote VLAN…
        - Must set a random unused port as a reflector-port
        - You essentially steal the port's ASIC to handle the forwarding logic
        - Not required on newer platforms (credit to /u/vista_df)
- **RELATED CISCO COMMANDS**

- **CONFIGURING LOCAL SPAN**
- `Switch(config)# monitor session <#> source {interface..|vlan..} {both|rx|tx}`
- `Switch(config)# monitor session <#> destination {interface..} [ingress]`
- *The ingress keyword allows the port to operate normally too*
- **CONFIGURATION SOURCE SWITCH REMOTE SPAN**
- `Sw1(config)# vlan 200`
- `Sw1(config-vlan)# remote-span`
- `Sw1(config)# monitor session 1 source interface Eth1/1`
- `Sw1(config)# monitor session 1 destination remote vlan 200 reflector-port Eth1/2`
- **CONFIGURATION DESTINATION SWITCH REMOTE SPAN**
- `Sw2(config)# vlan 200`
- `Sw2(config-vlan)# remote-span`
- `Sw2(config)# monitor session 1 source remote vlan 200`
- `Sw2(config)# monitor session 1 destination Eth1/3`

1.8 Describe chassis virtualization and aggregation technologies

1.8.? VSS

- **VSS A.K.A. "VIRTUAL SWITCHING SYSTEM"**
  - Combines two 6500 series switches into a single, virtual switch
  - **BENEFITS**
    - Same general benefits as Stackwise
    - Single point of management
    - Multichassis Etherchannel
  - **DIFFERENCES**
    - Stackwise required all members of stack to be co-located
    - This is because they had to be connected via a special short cable
    - VSS Switches are connected via standard 10GB Ethernet
    - With fiber, this means switches can be 40 KM apart!
    - The interlink can also support Etherchannel
- **VSS ACTIVE AND STANDBY CHASSIS**
  - **VSS ACTIVE CHASSIS**
    - Runs Layer-2 and Layer-3 control protocols
    - Provides console interface
    - Provides other management functions
  - Both chassis perform packet forwarding for locally-hosted interfaces
  - Standby sends all control traffic to Active Chassis
- **VSL A.K.A. "VIRTUAL SWITCH LINK"**
  - Utilizes the VSLP A.K.A. "Virtual Switch Link Protocol"
  - The VSLP performs communication between the VSS members across VSL
  - VSLP uses the following two sub-protocols:
    - LMP A.K.A. "Link Management Protocol"
    - The LMP provides domain identity parameters and capabilities
    - RRP A.K.A. "Role Resolution Protocol"
    - The RRP is used to determine which chassis becomes VSS active

- VSL carries both data and control traffic between both chassis
- However, if the VSS recognizes traffic is reachable via one chassis…
    - The traffic will be forwarded through the local switch
    - This avoids congestion on the VSL

2.0 Infrastructure Security

2.1 Configure and verify switch security features

2.1.a DHCP snooping

- **DHCP SNOOPING CONCEPT**
    - Generally, the way DHCP works, is a host sends out a DHCP Discover
    - As DHCP Discover is broadcast, a malicious actor can read this information
    - Thus, the malicious actor can respond to the DHCP request
    - Hosts accept the first offer that they get
    - For instance, the default gateway can be set as the malicious actor's IP
    - That way all of the host's traffic will be sent the malicious actor!
    - DHCP Snooping resolves this issue
- **DHCP SNOOPING TERMINOLOGY**
    - **UNTRUSTED PORTS**
        - All ports on a switch is untrusted if DHCP Snooping is enabled
    - **TRUSTED PORTS**
        - Manually have to specify an interface as a trusted port
    - **DHCP SNOOPING BINDING DATABASE**
        - If a DHCP discover come in from an untrusted port, start monitoring
        - Then forward that request to ONLY trusted ports
        - This information would contain the Client's MAC, IP, and Lease Time
        - Dynamic ARP Inspection relies on this specific feature to operate
- **DHCP SNOOPING OPERATION**
    - **UNTRUSTED → TRUSTED**
        - DHCP Discover
        - DHCP Request/Inform
        - DHCP Decline
        - DHCP Release
    - **TRUSTED → UNTRUSTED**
        - DHCP Offer
        - DHCP ACK
        - DHCP NACK
    - **UNTRUSTED → REMOVE DHCP**
        - DHCP Snooping is only designed to be enabled on access layer
        - DHCP relay causes the GiAddr field to be non-zero
        - If DHCP Snooping was enabled upstream and noticed a relay request…

        - The upstream switch will kill the request immediately
        - So be careful in your design implementation
- **RELATED CISCO COMMANDS**
    - **CONFIGURING DHCP SNOOPING**
    - Switch(config)# **ip dhcp snooping**
    - Switch(config)# **ip dhcp snooping vlan** <vlan-id>
    - **CONFIGURING RATE LIMIT TO PREVENT DOS ATTACKS**
    - Switch(config-if)# **ip dhcp snooping limit rate** <1 - 2048>
    - **CONFIGURING ADDITIONAL PARAMETERS**
    - Switch(config-if)# **ip dhcp snooping trust**
    - Switch(config-if)# **ip dhcp snooping information option**
    - *Generally, DHCP Snooping adds information to Option 82 in DHCP*
    - *Recommended to turn off otherwise it may cause issues*
    - **VERIFICATION**
    - Switch# **show ip dhcp snooping**
    - Switch# **show ip dhcp snooping binding**

2.1.b IP Source Guard

- **IP SOURCE GUARD CONCEPT**
    - Protects against a malicious host from impersonating a legitimate host
    - This feature is used in conjunction with DHCP Snooping and static bindings
    - Initially, all IP traffic on the protected port is **BLOCKED** except for DHCP packets
    - Traffic is only permitted after client receives…
        - Static IP Source Binding from Admin
        - Dynamically Assigned IP from DHCP Server
    - Prevents malicious actor assuming neighbor's IP address!
- **RELATED CISCO COMMANDS**
    - **CONFIGURING IP SOURCE GUARD**
    - Sw1(config-if)# **ip verify source vlan dhcp-snooping**
    - Sw1(config)# **ip source binding** <mac> **vlan** <#> <ip-addr> **interface** <interface>
    - *The above command assigns an IP address statically*
    - **VERIFICATION**
    - Sw1# **show ip verify source** [**interface** interface]

2.1.c Dynamic ARP inspection

- **DAI A.K.A. "DYNAMIC ARP INSPECTION"**
    - A malicious actor can read ARP requests from regular host easily
    - The malicious actor can then spoof its own IP address with the destination IP
    - By sending this unsolicited ARP Response, ARP cache of the host is poisoned
    - The host then redirects all traffic flow to that destination to the malicious actor
    - The actor can then replay the data to the real end host
    - This results in the malicious actor becoming a MITM A.K.A. "Man In The Middle"
    - DAI verifies ARP replies by inspecting them against Snooping Binding Database
    - If there is no match, drop the ARP and generate a syslog message
    - Can match against static ARP ACL entries for devices with static IP addresses
- **RELATED CISCO COMMANDS**

- **CONFIGURING DAI**
- Sw1(config)# **ip arp inspection vlan** <vlan-id>
- Sw1(config-if)# **ip arp inspection trust**
- **CONFIGURING STATICALLY BINDING ADDRESS**
- Sw1(config)# **arp access-list** <name>
- Sw1(config-acl)# **permit ip host** <ip-addr> **mac host** <mac-addr>
- Sw1(config)# **ip arp inspection filter** <arp-acl-name> **vlan** <vlan-id> [**static**]
- *The keyword "static" will force DAI to NOT check binding database at all*
- **CONFIGURING OPTIONS**
- Sw1(config)# **ip arp inspection validate** {[**src-mac**][**dst-mac**][**ip**]}
- *Checks additional parameters besides the default Sender MAC/IP/VLAN*
- **VERIFICATION**
- Sw1# **show ip arp inspection**
- Sw1# **show ip arp inspection interface** <interface>

2.1.d Port security

- **VACL A.K.A. "VLAN ACCESS LIST"**
  - Used for bridged or routed traffic
  - Applied to VLANs
  - Configured similar to Route Maps
  - Can match IP, IPX, and Layer-2 MAC traffic
  - MAC ACLs **ONLY** work if Ethertype field does not indicate IP/X in payload
- **VACL DETAILS**
  - **FORWARD**
    - Traffic is permitted and allowed to forward
  - **DROP**
    - Traffic is dropped
  - **REDIRECT**
    - Traffic is redirected out of a different destination interface
  - **CAPTURE**
    - Enhancement of forward
    - Allows capture of the forwarded traffic, similar to SPAN
- **PORT SECURITY**
  - Generally, anyone can access unsecure network resources by connecting
  - Port Security attempts to give the Admin more control over access
  - See cisco commands section for specific details
- **PORT SECURITY VIOLATION MODES**
  - **SHUTDOWN**
    - Places interface into error-disabled state
  - **PROTECT**
    - Unfamiliar frames are dropped, legitimate traffic permitted
  - **RESTRICT**
    - Unfamiliar frames are dropped, legitimate traffic permitted
    - Increases violation counter
    - Generates a syslog message
- **RELATED CISCO COMMANDS**

- **VACL CONFIGURATION**
- Sw1(config)# **vlan access-map** <map-name> [seq-num]
- Sw1(config-access-map)# **match ip address** {acl-number | acl-name}
- Sw1(config-access-map)# **match mac address** {acl-number | acl-name}
- Sw1(config-access-map)# **action** {**drop** | **forward** [**capture**] | **redirect** x/x}
- Sw1(config)# **vlan filter** <map-name> **vlan-list** <vlan-ids>
- Sw1# **show vlan access-map** <map-name>
- Sw1# **show vlan filter**
- **PORT SECURITY CONFIGURATION**
- Sw1(config-if)# **switchport port-security** (enables port security on interface)
- Sw1(config-if)# **switchport port-security violation** {shutdown|restrict|protect}
- Sw1(config-if)# **switchport port-security maximum** <# of macs>
- *By default, port security limits ingress MAC address count to one*
- Sw1(config-if)# **switchport port-security maximum** <# of macs> **vlan access**
- Sw1(config-if)# **switchport port-security maximum** <# of macs> **vlan voice**
- *You can also specify voice and access vlans separately*
- Sw1(config-if)# **switchport port-security mac-address** aa.aa.aa **vlan access**
- *You can also statically configure allowed MAC address per interface*
- Sw1(config-if)# **switchport port-security mac-address sticky**
- *This allows you to learn and set the ingress MAC address dynamically*
- Sw1(config-if)# **switchport port-security aging time** <#>
- Sw1(config-if)# **switchport port-security aging type inactivity**
- *Allows you to specify aging parameters for learned MAC addresses*

2.1.e Private VLAN

- **PRIVATE VLAN MOTIVATION**
  - If you have customers on a VLAN, you **DO NOT** want them talking to each other
  - However, they should be able to communicate with their own servers
  - Using a separate VLAN for each customer may not be feasible due to limitations
  - Possible solution is setting up VACLs on VLAN but that becomes unmanageable
  - We want the same subnet, but also want security without VACLs
  - This is exactly where Private VLANs can help us with
- **PRIVATE VLAN CONCEPT**
  - PVLAN is just a combination of two VLANs working together
    - **Primary Vlan** → Controls IP Subnet reachability
    - **Secondary VLANs** → Controls Security Policy within Primary VLAN
  - Secondary VLANs come in two types
    - **Community** → Devices can talk to each other in same community
    - **Isolated** → Devices cannot talk to each other with no exceptions
- **PRIVATE VLANS SECONDARY TYPES**
  - **COMMUNITY**
    - Reside in the same IP subnet as Primary VLAN
    - Reside in same L2 broadcast domain
    - Cannot access members of other secondary VLANs
  - **ISOLATED**
    - Reside in the same IP subnet as Primary VLAN
    - Cannot access members of the same isolated VLANs

        - Cannot access members of any other secondary VLANs
- **PRIVATE VLAN PROMISCUOUS**
    - A configured Promiscuous Port allows PVLAN hosts to reach default gateway
    - This allows outside routing even if PVLAN members cannot talk to each other
    - Promiscuous Port is either the physical interface leading to router or the SVI
- **PRIVATE VLAN RESTRICTIONS**
    - Switches MUST be VTP Transparent Mode (exception is VTPv3)
    - Must select unused VLANs for Primary and Secondary assignment
    - PVLAN configuration must be consistent across all switches to trunk properly
    - This is because only the secondary vlan is transmitted across link
    - Etherchannel must not have any PVLAN configuration
- **RELATED CISCO COMMANDS**
    - **CONFIGURE SECONDARY PVLAN**
    - Sw1(config)# **vlan** <vlan-id>
    - Sw1(config-vlan)# **private-vlan** <**community** | **isolated**>
    - **CONFIGURE PRIMARY PVLAN**
    - Sw1(config)# **vlan** <vlan-id>
    - Sw1(config-vlan)# **private-vlan primary**
    - Sw1(config-vlan)# **private-association** <vlan-ids | **add** | **remove**>
    - *This ties secondary vlans with primary*
    - **CONFIGURE HOST PORTS**
    - Sw1(config-if)# **switchport mode private-vlan host**
    - Sw1(config-if)# **switchport private-vlan host associatio** <prim-pvlan> <sec-pvlan>
    - **CONFIGURE PROMISCUOUS PORT**
    - Sw1(config-if)# **switchport mode private-vlan promiscuous** (only on physical)
    - Sw1(config-if)# **switchport private-vlan mapping** <prim> <sec> (only on physical)
    - Sw1(config-if)# **private-vlan mapping** <sec> (SVI)
    - **VERIFICATION**
    - Sw1# **show vlan private-vlan**
    - Sw1# **show interface vlan** <#> **private-vlan mapping**

2.1.f Storm control

- **STORM CONTROL MOTIVATION**
    - Broadcast, multicast, or unknown unicast are flooded on all ports in VLAN
    - These storms can increase the CPU utilization on receiving hosts
    - Storm Control can help with this
- **STORM CONTROL SUPPORT**
    - **ALL PLATFORM SUPPORT**
        - Monitor bandwidth as a percentage of total available bandwidth of port
    - **SOME PLATFORM SUPPORT**
        - Monitor traffic rate in packets per second on RX
        - Monitor traffic in bits per seconds on RX
        - Monitor traffic rate in packets per second or for small frames
- **STORM CONTROL CONCEPT**
    - When traffic exceeds the “Rising Threshold”...
        - Drop all frames during the **NEXT** timeslot
        - Each timeslot represents a one-second interval

- Frames are not forwarded until…
    - An **ENTIRE** timeslot is beneath the “Falling Threshold”
    - If “Falling Threshold” is not specified, equal to “Rising Threshold”
- **RELATED CISCO COMMANDS**
    - **STORM CONTROL CONFIGURATION**
    - `Sw1(config-if)# storm-control broadcast level 75.5 (percentage)`
    - `Sw1(config-if)# storm-control multicast level pps 2k 1k (rising and falling)`
    - `Sw1(config-if)# storm-control action shutdown (error-disable)`
    - **VERIFICATION**
    - `Sw1# show storm-control`

2.2 Describe device security using Cisco IOS AAA with TACACS+ and RADIUS

2.2.a AAA with TACACS+ and RADIUS

- **AAA A.K.A. “AUTHENTICATION, AUTHORIZATION, ACCOUNTING”**
    - Client → NAS A.K.A. “Network Access Server” → Server
    - Typically used when…
        - Client wants CLI access to network device **or**
        - Client wants network access (802.1x)
    - **DOT1X**
        - Mechanism that will **block** or **unblock** an interface
        - Provides security at Layer 2
        - Known as **port-based control**
        - All traffic besides **EAPoL** is dropped
        - Once authentication, network access is granted and traffic resumes
    - **AUTHENTICATION**
        - Verifying credentials of client
        - Variety of methods to facilitate authentication
        - E.G. Username/Password, Digital Certs, MAC Address
    - **AUTHORIZATION**
        - Determines privileges of authenticated clients
        - Determines WHAT the client is allowed to do/not do
        - Many different features that can be authorized
        - E.G. Basic network access, CLI, VLAN assignment, QoS, ACLs
    - **ACCOUNTING**
        - Gathering of statistics
        - Typically a separate process aside from Authentication/Authorization
        - Information gathered may be identity of users, services, etc.
- **TACACS+**
    - 802.1x is the negotiation between the client and NAS
    - TACACS+ and RADIUS is the negotiation between NAS and AAA Server
    - TACACS+ is Cisco Proprietary

    - TCP Port 49
    - Considers Authentication, Authorization, and Accounting as separate processes
- **RADIUS**
    - IETF Standard Protocol
    - Unlike TACACS+, bundles AUthentication/Authorization
    - Official UDP port 1812 and 1813
- **EAPoL A.K.A. "EXTENSIBLE AUTHENTICATION PROTOCOL OVER LAN"**
    - Authentication framework frequently used in wireless and point-to-point networks
    - In 802.1X, EAP and RADIUS is combined to provide the authentication to users
    - EAP is used to communicate with the client at the network perimeter
    - RADIUS is used to relay authentication details to the server inside the network
- **RELATED CISCO COMMANDS**
    - **AAA CONFIGURATION**
    - Sw1(config)# **aaa new-model**
    - Sw1(config)# **aaa authentication login default group** <**radius** | **tacacs+**>
    - **CONFIGURATION WITH SERVER SPECIFIC INFO**
    - Sw1(config)# **tacacs-server host** <ip-addr> **key** <key>
    - Sw1(config)# **radius-server host** <ip-addr> **key** <key>
    - **CONFIGURATION WITH GLOBAL PARAMETERS**
    - Sw1(config)# **radius-server host** <ip-addr>
    - Sw1(config)# **radius-server key** <key>
    - **ALTERNATIVE CONFIGURATION STYLE**
    - Sw1(config)# **radius server** <name>
    - Sw1(config-radius-server)# **address ipv4** <ip-addr> **auth-port** 1812 **acct-port** 1813
    - Sw1(config-radius-server)# **key** <key>
    - **DOT1X CONFIGURATION**
    - Sw1(config)# **dot1x system-auth-control** (enables dot1x globally)
    - Sw1(config)# **aaa new-model**
    - Sw1(config)# **aaa authentication dot1x default group radius**
    - Sw1(config-if)# **switchport mode access** (switchport must be access)
    - Sw1(config-if)# **dot1x port-control** <**auto**|**force-authorized**|**force-unauthorized**>
    - ***auto*** *- Normal 802.1X authentication*
    - ***force-authorized*** *- No 802.1X authentication is used, default setting*
    - ***force-unauthorized*** *- Port is always unauthorized*
    - **VERIFICATION**
    - Sw1# **show aaa authentication**
    - Sw1# **show aaa accounting**
    - Sw1# **show dot1x interface** <interface>

2.2.b Local privilege authorization fallback

- **RELATED CISCO COMMANDS**
    - **LOCAL PRIVILEGE FALLBACK**
    - Sw1(config)# **aaa authentication login default group** <**radius** | **tacacs+**> **local**
    - Sw1(config)# **username** <username> **password** <password>

3.0 Infrastructure Services

3.1 Configure and verify first-hop redundancy protocols

3.1.a HSRP

- **HSRP A.K.A. "HOT STANDBY ROUTER PROTOCOL"**
  - Provides gateway redundancy for hosts in a specific subnet
  - This is done by having a routers provide a virtual ip to hosts
  - The virtual ip is a standalone ip address that must be configured
  - Cisco Proprietary
  - Uses UDP port 1985 and multicast address 224.0.0.2
  - Two roles: Active and Standby
  - HSRP router with highest priority is considered Active
  - Default priority for switches is 100
  - MAC Address format for HSRP is 000.0c07.acxx
  - The xx refers to the group number in hexadecimal
  - Preemption is disabled by default
  - Both HSRP/VRRP has no load sharing feature
  - Default Hello/Dead is 3/10
- **HSRP FEATURES**
  - **MHSRP A.K.A. "MULTI-GROUP HSRP"**
    - Considered an extension of HSRP
    - Have one virtual IP as active, the other standby on same interface
    - Have DHCP server offer one gateway over the other to hosts
    - End goal is to have half the hosts with one gateway
    - Provides semi-load balancing as half of the hosts will go to one router
  - **OBJECT TRACKING**
    - HSRP can track objects, typically interfaces
    - If tracked object fails, HSRP priority is reduced by a configurable amount
    - The default amount is 10
  - **AUTHENTICATION**
    - Plain text and MD5 authentication supported
    - Can also obtain current/active MD5 key from a key-chain
- **HSRP STATE MACHINE**
  - **DISABLED**
    - Not doing anything
  - **INITIAL (INIT)**
    - HSRP process is not running yet as interface just came up
  - **LEARN**
    - If no virtual IP was provided manually…
    - Router attempts to learn it from other HSRP routers
  - **LISTEN**

        - Know what virtual IP address is
        - Become neither the active or standby, just OTHER router
    - **SPEAK**
        - When router first comes up, tells others it wants to become active
        - This also occurs when preempt is enabled
    - **STANDBY / ACTIVE**
        - Active has the highest priority/ip
        - Second best is standby
        - Everyone else is in Listen
- **HSRP VERSION DIFFERENCES**
    - HSRPv2 introduces advertisement of millisecond timer values
    - This ensures stability of all HSRP groups in all cases
    - HSRPv2 expands group range from 0 to 4095 compared to 255
    - HSRPv2 includes a 6-Byte identifier for the originating router's MAC
    - Originally, the only information given was the virtual MAC
    - This allows for easier troubleshooting when sniffing data
    - HSRPv2 uses the multicast address 224.0.0.102
    - This is so that the hello packets do not conflict with CGMP leave process
    - CGMP A.K.A. "Cisco Group Management Protocol"
        - Legacy method of dealing with IGMP at layer 2
        - Nowadays, supplanted with IGMP Snooping
- **RELATED CISCO COMMANDS**
    - **CONFIGURING HSRP**
    - Sw1(config-if)# **standby** <group-id> **ip** <virtual-ip>
    - Sw1(config-if)# **standby** <group-id> **priority** <priority>
    - **CONFIGURING PREEMPT**
    - Sw1(config-if)# **standby** <group-id> **preempt**
    - Sw1(config-if)# **standby** <group-id> **preempt delay minimum** <time>
    - **CONFIGURING MHSRP**
    - Sw1(config-if)# **standby** 1 **ip** 10.1.1.1
    - Sw1(config-if)# **standby** 2 **ip** 10.1.1.2
    - Sw1(config-if)# **standby** 1 **priority** 200
    - **CONFIGURING HELLO TIMERS**
    - Sw1(config-if)# **standby** 1 **timers** hello-time-in-sec hold-time-in-sec
    - Sw1(config-if)# **standby** 1 **timers msec** hello-time-in-msec hold-time-in-msec
    - If the msec timer is configured, must ensure this is configured everywhere
    - This is because the values are not advertised in HSRPv1 due to bit limit
    - **CONFIGURING AUTHENTICATION**
    - Sw1(config-if)# **standby** <group-id> **authentication** <password> (plain-text)
    - Sw1(config-if)# **standby** <group-id> **authentication md5 key-string** [0|7] <string>
    - **CONFIGURING AUTHENTICATION WITH KEYCHAIN**
    - Sw1(config)# **key chain** <name>
    - Sw1(config-keychain)# **key** <#>
    - Sw1(config-keychain-key)# **key-string** <string>
    - Sw1(config-if)# **standby** <group-id> **authentication md5 key-chain** <name>
    - **CONFIGURE HSRP OBJECT TRACKING**
    - Sw1(config)# **track** <track-number> **interface** <interface>
    - Sw1(config-if)# **standby** <group-id> **track** <track-number> [**decrement-value**]
    - **VERIFICATION**

- `Sw1# show standby`
- `Sw1# show standby brief`

3.1.b VRRP

- **VRRP A.K.A. "VIRTUAL ROUTER REDUNDANCY PROTOCOL"**
  - Internet Standard: RFC 3768, 5798
  - Custom transport protocol: 112
  - Multicast address: 224.0.0.18
  - Master router replies to ARP requests for virtual IP address
  - The virtual IP address for VRRP is the physical interface IP of the Master
  - Default priority at 100
  - MAC Address format for VRRP is 000.5e00.01xx
  - The xx refers to the group number in hexadecimal
  - Preemption is enabled by default
  - Both HSRP/VRRP has no load sharing feature
  - Different instance of VRRP can provide semi-load-sharing
  - Default Hello/Dead is 1/3.6 seconds
  - VRRP has no standby router to take over active spot
- **VRRP TIMERS**
  - All routers in VRRP group must share the same hello timer
  - If you change one router, the other routers will not adjust their hold timers
  - This behavior can be adjusted via configuration
  - VRRP hello packet cannot advertise millisecond timers
  - Configuring timers greater than default risks passive routers becoming Master
  - For this reason, it is recommended to have passive routers "learn" the new timer
  - If you do not, **MULTIPLE** routers may actually become master
- **VRRP MASTER ELECTION PROCESS**
  - When the master goes down, all routers fight to become new Master
  - This means it takes longer for VRRP to converge compared to HSRP
  - The backup router assigned the highest priority for each VRID becomes master
  - If two backup routers were assigned same priority, highest IP address wins
  - Master is the only one that sends hellos
- **VRRPv2 VS VRRPv3**
  - **VRRPv2**
    - Defined in RFC 3768
    - Supports IPv4 only
    - Timers in seconds
    - 224.0.0.18 for Multicast Address
    - Node with the same priority value but higher IP would cause preemption
    - Configured on a per interface basis
  - **VRRPv3**
    - Defined in RFC 5798
    - Supports both IPv4 and IPv6
    - Timers in milliseconds

        - 224.0.0.18 and FF02::12 for Multicast Address
        - Node with the higher priority would cause preemption ONLY
        - Configured globally
- **RELATED CISCO COMMANDS**
    - **CONFIGURING VRRP**
    - Sw1(config-if)# **vrrp** <group-id> **ip** <virtual ip>
    - Sw1(config-if)# **vrrp** <group-id> **priority** <priority>
    - **CONFIGURING VRRP AUTHENTICATION**
    - Sw1(config-if)# **vrrp** <group-id> **authentication** <password> (plain-text)
    - Sw1(config-if)# **vrrp** <group-id> **authentication md5 key-string** <password>
    - Sw1(config-if)# **vrrp** <group-id> **authentication md5 key-chain** <key-chain-name>
    - *To see keychain configuration, refer to the HSRP section*
    - **ADJUST TIMERS**
    - Sw1(config-if)# **vrrp** <group-id> **timers advertise msec** <value>
    - Sw1(config-if)# **vrrp** <group-id> **timers learn** (use when increasing timers)
    - **VERIFICATION**
    - Sw1# **show vrrp**
    - Sw1# **show vrrp brief**

3.1.c GLBP

- **GLBP A.K.A. "GATEWAY LOAD BALANCING PROTOCOL"**
    - Cisco Proprietary
    - Provides gateway redundancy AND per-host load-balancing
    - This is **TRUE** load-balancing unlike MHSRP
    - UDP port 3222, multicast address 224.0.0.102
- **GLBP TERMINOLOGY**
    - **AVG A.K.A. "ACTIVE VIRTUAL GATEWAY"**
        - Highest priority in group becomes AVG
        - If equal priorities, highest IP becomes AVG
        - Preemption for role of AVG is enabled by default
        - AVG replies to ARP requests sent to the virtual IP
        - AVG determines host-to-gateway allocations
        - AVG is also an AVF
    - **AVF A.K.A. "ACTIVE VIRTUAL FORWARDER"**
        - Maximum of 4 AVFs per group
        - Each AVF assigned a virtual MAC: 0007.b4xx.xxyy
        - Where xx.xx is GLBP Group # and yy is AVF #
        - AVFs request their AVF # and virtual MAC from AVG
        - AVG and AVFs all send hello packets, default 3 seconds
- **GLBP AVF FAILURE SCENARIO**
    - When an AVF fails, AVG assigns the failed router's MAC to an existing AVF
    - The AVF takes on both its own load and the failed router's load
    - The AVG will keep handling out the old MAC during the redirect interval
    - The AVG will start redistributing the load equally after this
    - The AVF will keep servicing the old MAC until the timeout interval passes
- **GLBP LOAD BALANCING OPTIONS**

- Round-Robin (default)
- Host dependent where the mac address the host receives is always the same
- Weighted where weights are given on a per router basis

- **GLBP OBJECT TRACKING**
  - Every router has a default AVF weight of 100 (maximum value)
  - When lowering weight, the router can no longer participate as AVF
  - Object tracking can be used to dynamically decrement weight value
- **RELATED CISCO COMMANDS**
  - **CONFIGURING GLBP**
  - Sw1(config-if)# **glbp** <group-id> **ip** <virtual ip>
  - Sw1(config-if)# **glbp** <group-id> **priority** <priority>
  - Sw1(config-if)# **no glbp** <group-id> **preempt**
  - **CONFIGURING GLBP LOAD BALANCING**
  - Sw1(config-if)# **glbp** <group-id> **load-balancing** <**weighted** | **round** | **host**>
  - **CONFIGURING GLBP WEIGHTED LOAD BALANCING**
  - AVG(config-if)# **glbp** <group-id> **load-balancing weighted**
  - AVF(config-if)# **glbp** <group-id> **weighting** <value> **lower** <value> **upper** <value>
  - Going below lower value, relinquishes AVF
  - Going above upper value will allow router to become an AVF again
  - **CONFIGURING AVF OBJECT TRACKING**
  - AVF(config)# **track** <track-num> **interface** <interface>
  - AVF(config-if)# **glbp** <group-id> **weighting track** <track-num> **decrement** <value>
  - **TWEAKING GLBP INTERVALS**
  - Sw1(config-if)# **glbp** 1 **timers** <hello> <hold>
  - Sw1(config-if)# **glbp** 1 **timers redirect** <redirect interval> <timeout interval>
  - **VERIFICATION**
  - Sw1# **show glbp**
  - Sw1# **show glbp brief**