Post 3

Thursday, January 24, 2019 11:19 PM

LAN REDUNDANCY P2

7 varieties of spanning tree protocols have emerged since IEEE 802.1D:

STP	**Original IEEE 802.1D version (1998/earlier):** • Loop-free topology in network w/redundant links **CST: Common Spanning Tree:** • Assumes 1 s/tree instance for entire bridged network • Regardless of # of VLANs
PVST+	**Cisco enhancement of STP:** • Provides separate 802.1D spanning tree instance for each VLAN config in network Separate instance supports: • PortFast • UplinkFast • BackboneFast • BPDU guard • BPDU filter • Root guard • Loop guard.
802.1D-2004	**Updated ver of STP standard:** • Incorporates IEEE 802.1w
RSTP or IEE 802.1w	**RSTP: Rapid Spanning Tree Protocol or IEEE 802.1w:** • Evolution of STP that provides faster convergence
Rapid PVST+	**Cisco enhancement of RSTP that uses PVST+:** • Rapid PVST+ provides separate instance of 802.1w per VLAN Separate instance supports: • PortFast • BPDU guard • BPDU filter • Root guard • Loop guard
MSTP	**MSTP: Multiple Spanning Tree Protocol:** • IEEE standard inspired by earlier Cisco **MISTP: Multiple Instance STP** implementation • Maps multiple VLANs into same spanning tree instance **Cisco implementation of MSTP is MST:** • Provides up to 16 instances of RSTP • Combines many VLANs w/same physical/logical topology into common RSTP instance Each instance supports: • PortFast • BPDU guard • BPDU filter • Root guard • Loop guard

Characteristics of the Spanning Tree Protocols

STP	**Assumes 1 IEEE 802.1D spanning tree instance for entire bridged network: Regardless of # of VLANs** • B/C 1 instance: ○ CPU/Mem reqs lower

	○ Only 1 root bridge/1 tree • Traffic for all VLANs flows over same path: Can lead to suboptimal traffic flows • Limitations 802.1D: Slow to converge
PVST+	**Cisco enhancement of STP:** • Separate instance of Cisco implementation of 802.1D for each VLAN config in network • Speed of convergence: Similar to STP Separate instance supports: • PortFast/UplinkFast/BackboneFast/BPDU guard/BPDU filter/root guard/loop guard • Port roles defined: Same as w/RSTP • Creating instance for each VLAN increases CPU/Mem reqs: *Allows per-VLAN root bridges* • Design allows spanning tree to be optimized for traffic of each VLAN • Convergence of ver similar to 802.1D • Convergence per-VLAN
RSTP (IEEE 802.1w)	**Evolution of spanning tree:** • Provides faster convergence than 802.1D • Addresses many convergence issues • Still single instance of STP: Doesn't address suboptimal traffic flow To support faster convergence: • CPU/Mem reqs of this ver higher than of CST: Less than Rapid PVST+
Rapid PVST+	**Cisco enhancement of RSTP that uses PVST+:** • Provides separate instance of 802.1w per VLAN Separate instance supports: • PortFast/BPDU guard/BPDU filter/root guard/loop guard • Ver addresses convergence issues/suboptimal traffic flow • Highest CPU/Mem reqs
MSTP	**IEEE 802.1s standard:** • Inspired by earlier Cisco proprietary MISTP implementation To reduce # of required STP instances: • MSTP maps multiple VLANs • Same traffic flow reqs into same spanning tree instance
MST	**Cisco implementation of MSTP:** • Up to 16 instances of RSTP (802.1w) • Combines many VLANs w/same physical/logical topology into common RSTP instance Each instance supports: • PortFast/BPDU guard/BPDU filter/root guard/loop guard • CPU/Mem reqs this ver: Less than Rapid PVST+ **||** More than RSTP

Default spanning tree mode for Cisco Catalyst switches is PVST+

- Enabled on all ports/Much slower convergence after topology change than Rapid PVST+

PVST+

A network running CST has these characteristics:

- No load sharing possible
- 1 uplink must block for all VLANs
- CPU is spared: Only 1 instance of spanning tree must be computed
- Developed so network can run independent instance of implementation of IEEE 802.1D for each VLAN in network
- Possible for 1 trunk port on switch to block for VLAN while fwding for other VLANs
- Can be used to implement L2 load balancing
- Switches need greater CPU process/BPDU BW consumption than CST B/C each VLAN runs separate instance of STP

Spanning tree params can be tuned so half of VLANs fwd on each uplink trunk

- Accomplished by config 1 switch to be elected root bridge for 1/2 of the VLANs in network
- Second switch to be elected root bridge for other 1/2 of VLANs
- Multiple STP root bridges per VLAN increases redundancy

Networks running PVST+ chars:

- Optimum load balancing can result
- Considerable waste of CPU cycles for all switches in network (in addition to BW used for each

instance to send BPDU)
- Only problematic if large number of VLANs config

Port States/PVST+ Operation
- STP facilitates logical loop-free path throughout broadcast domain
- Spanning tree is determined through info learned by exchange of BPDU frames bet interconnected switches
- Learning of logical spanning tree: Each switch port transitions through 5 possible states/3 BPDU timers
- Spanning tree determined immediately after switch finished booting
- If port transitions directly from blocking to fwding state w/out info about full topology during transition
- Port can temp create a data loop
- For this reason: STP introduces 5 port states
- PVST+ uses same 5 port states

Port states during creation of logical spanning tree:

Blocking	Port is an alternate port: Doesn't participate in frame fwding • Port receives BPDU frames to determine location/root ID of root bridge switch • And which port roles each port should assume in final active STP topology
Listening	Listens for path to root • STP has determined port can participate in frame fwding according to BPDU frames that switch received • Port receives BPDU frames: Transmits its own BPDU frames: • Informs adjacent switches port is preparing to participate in active topology
Learning	Learns MAC addresses: • Port prepares to participate in frame fwding/begins to populate MAC table
Fwding	Port considered part of active topology: Fwds data frames/sends/receives BPDU frames
Disabled	L2 port doesn't participate in spanning tree: Doesn't fwd frames • Set when port is admin disabled

# of ports in each of various states can be displayed w/**show spanning-tree summary**
For each VLAN in switched network: PVST+ performs 4 steps to provide loop-free logical topology:

- **Elects 1 root bridge: Only 1 switch can act as root bridge (for given VLAN)**

    1. RB (root bridge) is switch w/lowest BID
    2. RB: All ports designated (no root ports)

- **Selects root port on each non-root bridge:**

    1. Establishes 1 root port on each non-root bridge for each VLAN
    2. RP (Root port) is lowest-cost path from non-root bridge to RB (indicates direction of best path to RB)
    3. RP normally in fwding state

- **Selects designated port on each segment:**

    1. On each link: Establishes 1 designated port for each VLAN
    2. Designated port selected on switch that has lowest-cost path to RB
    3. Designated ports normally in fwding state/Fwding traffic for segment

- **Remaining ports in switched network are alternates:**

    1. Normally in blocking state, to break loop topology
    2. When port in blocking state: Doesn't fwd traffic/Can still process received BPDU msgs

Extended System ID and PVST+ Operation
PVST+ environment: Extended sys ID ensures each switch has a unique BID for each VLAN
Overview of Rapid PVST+

- Evolution of 802.1D standard: Incorporated into IEEE 802.1D-2004
- 802.1w STP terminology remains same as original
- Most params left unchanged
- Cisco: RSTP on per-VLAN basis: Independent instance of RSTP runs for each VLAN
- RSTP:
    - Doesn't have blocking port state
    - Defines port states as discarding/learning/fwding

RSTP speeds recalc of spanning tree when L2 topology changes:

- Much faster convergence
- Redefines type of ports/state
- If port is config to be alternate/backup: Can immediately change to fwding state w/out waiting for network to converge

RSTP characteristics:

- Preferred protocol for preventing L2 loops in switched network env
- Many diff established by Cisco enhancements to 802.1D

These enhancements:

- BPDUs carrying/sending info about port roles only to neighboring switches: Req no additional config
- Perform better than earlier Cisco vers
- Transparent/integrated into protocol's op
    - UplinkFast/BackboneFast: NOT compatible w/RSTP

RSTP (802.1w): Supersedes 802.1D while retaining backward compatibility

- Much of 802.1D term remains/most params unchanged
- 802.1w: Capable of reverting back to legacy 802.1D to interoperate w/legacy switches on per-port basis
- Keeps same BPDU fmt as 802.1D:
    - Except ver field is set to 2 to indicate RSTP: Flags field uses all 8 bits
- Able to actively confirm port can safely transition to fwding state w/out having to rely on timer config

RSTP BPDUs

RSTP uses type 2, ver 2 BPDUs:

- Original 802.1D STP uses type 0, version 0 BPDUs
- Switch running RSTP can comm w/switch running 802.1D STP

RSTP sends BPDUs/populates flag byte in slightly diff manner than 802.1D:

- Protocol info can be immediately aged on port if Hello packets not received for 3 consecutive Hello times
    - 6 seconds/max age timer expires (still 2 second intervals)
- **BPDUs:** Used as keep-alive mechanism
- 3 consecutively missed BPDUs indicate lost connectivity between bridge/neighboring root /designated bridge
- Fast aging of info allows failures to be detected quickly

Ver 2 BPDU:

Bits 0/7	Used for topology change/acknowledgment • In 802.1D
Bits 1/6	Used for Proposal Agreement process (rapid convergence)
Bits 2/5	Encode role/state of port
Bits 4/5	Used to encode port role using 2-bit code

Edge Ports

RSTP Edge Port: Switch port that never intended to be connected to another switch

- Immed transitions to fwding state when enabled
- Corresponds to PVST+ PortFast
- **Edge port:** Directly connected to end station/assumes no switch device is connected to it
- Skips the time-consuming 802.1D listening/learning states because of the fwding transition

Cisco RSTP: Rapid PVST+: Maintains PortFast keyword:

spanning-tree portfast Edge port config

Config edge port to attach to other switch NOT recommended

- Negative implications for RSTP B/C temp loop may result: Possibly delays convergence of RSTP

Link Types Provides categorization for each port participating in RSTP: Uses duplex mode on the port

Depending on what's attached to each port: 2 diff link types can be identified:

• **Point-to-Point**	**Port operating in full-duplex mode:** • Connects a switch to switch • Candidate for rapid transition to fwding state
• **Shared**	**Port operating in half-duplex mode:** • Connects switch to hub that attaches multiple devices

Link type can determine:

- Whether port can immed transition to fwding state: Assuming conditions are met
- Conditions are diff for edge ports/non-edge ports

Non-edge ports are categorized into 2 link types:

1. Point-to-point
2. Shared

Link type is auto determined: Can be overridden w/explicit port config using **spanning-tree link-type**

Characteristics of port roles w/regard to link types include:

- Edge port and point-to-point connections: Candidates for rapid transition to fwding state
    - Before link-type param considered: RSTP must determine port role
- Root ports don't use link-type param: RP's are able to make rapid transition to fwding state as soon as port in sync
- Alternate/backup ports don't use link-type param in most cases
- Designated ports make most use of link-type param
- Rapid transition to fwding state for designated port occurs only if link-type param is set to point-to-point