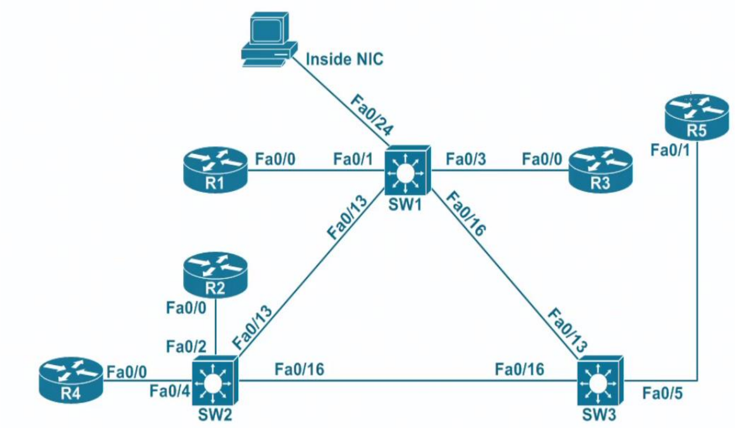
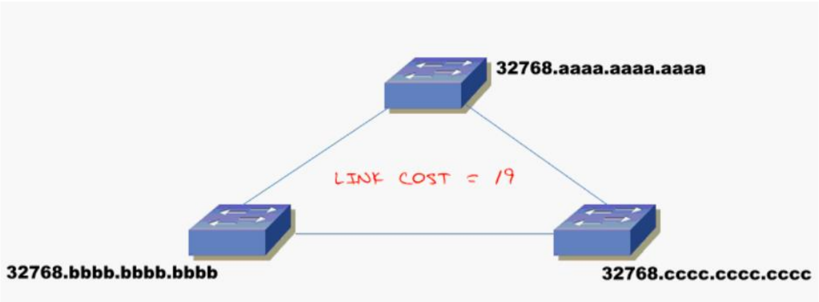
::::::Spanning Tree Protocol/STP(to prevent switching/layer2 loops)(on by deafult):::::::

->When the redundant path exist in the layer 2 network, CAM population logic breaks down and frames are switched out the wrong interfaces
->Looping frames, expecially broadcasts, can quickly overwhelm all links with 100% utilization
->As the network starts to grow redundandy becomes a concern (if the switches are daisy chained and if one cable or switch goes down it shouldn't take down the whole network)
->adding more connections between switches adds redundancy, but can cause problems (frames that follow physical loops in the network can bring the network down)
->STP is desinged to fix this (allows redundant connections but prevent traffic loops)
->layer3 loops are prevented using mechanisms like split horizon
->STP solves the looping problem by 'blocking' redundant paths
(STP tells what interfaces are actively being used to insert mac addresses in the CAM table)
(blocked links cannot forward traffic or use the CAM table)
(same effect as removing or shutting down the link)
(can receive frames but are discarded/only STP control messages are allowed on blocked ports/no mac addresses are associated on that port and thus no forwarding on that port)
->Since STP is dynamic, layer 2 network can reconverge around network failures
->standards based per 802.1d/CST

1.Switches forward broadcast packets out all ports by design
2.redundant connections are necessary in business networks
3.the place of spanning tree: blocks redundant links (until they are needed)
4.original STP (802.1d) was created to prevent loops
5.switches send "probes" into the network called bridge protocol data units (BPDUs)(a multicast packet) to discover loops
6.The BPDU probes also help elect the core switch of the network, called the Root bridge
7.The simplistic view of stp: all switches find the best way to reach the root bridge then block all redundant links
8.ARP(Address resolution protocol)->IP to MAC translation for layer 2 communication can cause loops in the network
9.load balancing can be done on layer 3 to send packets on more than one redundant links but on layer 2 there is no way to do it except for seperate redundant links used for separate vlans(pvst+)



(Three Port Types)(every cisco switch runs stp by default)
1.Root port(Root): used to reach the root bridge (always faces upstream towards the root bridge)(root can never be on root bridge)
2.Designated Port(Desg): forwarding port, one per link (always faces downstream but you can send traffic out)
3.Blocking/Non-Designated Port(BLK/Altn): where the link is blocked (always faces downstream you can't send traffic out)(can't associate mac addresses with this port)
4.Bridge ID=Priority(32768 by default/lower is better)+MAC address
5.All the priorities are tied by fault on all the switched so MAC address is used to break the tie and the oldest MAC addressed switch becomes the root brdige which could slow down the network as the oldest mac address will be on the oldest switch.
6.all ports of the root bridge are designated ports
7.there will be one (designated port) forwarding port per link
8.you cannot change the mac address but can change the priority to change the election of root bridge
9.sys-id-ext is added to the priority of the switch, which is the vlan number in which a particular stp instance is running (e.g. 32768 + sys-id-ext 1 = 32769)(i.e. per vlan spanning-tree)
10.standard stp(802.1d/not used anymore) is really slow (30 - 50 secs downtime)(not suitable for real time application like VoIP)
11.by one side blocking its port it disables the whole link (even tough the other side keeps showing Desg/FWD)
12.when all upstream root ports are elected rest of the upstream ports are blocked, where all downstream ports are designated



(How STP finds the best path?) (Root Bridge(Lowest BID)->Root Port(Lowest cost)(Lowest upstream BID)(Lowest port ID))

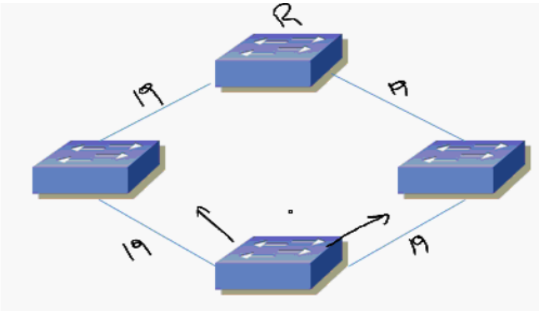
1.Exchanges bridge and link attributes i.e. BDPU(Bridge protocol data units) which has bridge id(Priority+MAC)(lower numerical value is highest priority)(lowest mac address preffered)

1.sent as multicast frames between adjacent bridges 0180.C200.0000

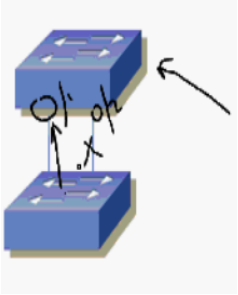
2.(Attributes are Root ID/RootPath Cost/Bridge ID/Port ID/Timers)

2.elect the root bridge: (lowest BID)(lowest priority+lowest mac address)

3.Use lower bridge id on equal cost paths



4.Use lower port to break a tie (same bridge id to we look at port IDs to break the tie)



->Root port election per bridge: switches find lowest cost path to root and checks the lowest bridge id and then lowest port priority

10Mobps=100/100Mbps=19/1Gbps=4/10Gbps=2

->(root bridge has a cost of 0 to reach itself)

5.elect designated ports

(How STP Loop prevention works in detail)

1.All devices agree on a reference point in the network called the root bridge (most upstream device in the topology)

2.Devices directly downstream of the root bridge performs the following:

1.select one upstream facing port to forward traffic towards the root bridge called root port (based on lowest cost path and most bandwidth)

2.all other upstream facing ports are disabled called blocking ports

3.all downstream facing ports are called designated ports

3.Next downstream device performs the same, selecting one upstream facing

root port
4.repeated till loop free tree is built

(Port States)
1.Disabled (e.g. shutdown)
2.Listening(LIS)(exchanging BPDUs with adjacent bridges)
3.Learning(LRN) (Building CAM table)
4.Forwarding(FWD) (Normal Loop free traffic forwarding)
5.Blocking (BLK) (receiving BPDUs but not forwarding)

(Progression between states)
1.Disabled(e.g. shutdown or cable not connected)->listening(15)->learning(15)->forwarding (30 seconds)
2.Blocking(20)->listening(15)->learning(15)->forwarding (50 seconds)

(Port Roles)
1.Root(Root)
2.Designsted(Desg)
3.Blocked (Altn)

(BPDUs/Bridge Protocol Data Units)
1.sent once every 2 seconds.
2.[priority+mac address(i.e. switch mac address)]
3.priority is some value between 0 and 61440 (default is 32768); increments of 4096 - lower is better
4.Superioir BPDU=advertisements coming in to the root(i.e best path to the root)
5.Inferior BPDU=received on designated port or the blocking ports

(Two Types of BPDUs)
1.Configuration BPDUs
2.Topolgy change notification/TCN BPDU (used when there is a state change in the network (up/down), it tells everyone to flush the CAM table and try to recalculate where the mac address is suppose to be located)

(Root Bridge election)((32768+sysID).MAC)
1.lowest bridge id(BID)=Lowest Priority+Lowest Mac address
2.BID is an 8 byte field that contains bridge priority(0-65535)(default 32768) and mac address
3.New standard splits bridge priority into two fields
    1.mac address reduction feature
    2.bridge priority 6 high order bits (0-61440 in increments of 4096)(0 is the most prefered)
    3.system-id extension (12 low order bits (0-4095)/VLANs)
4.Lowest BID in the network becomes everyone's Root ID (RID) in their BPDUs

(Root Port election)
(Lowest cost->Lowest Bridge ID->Lowest Port Priority)
1.Port closest to the root bridge (root is always upstream)
2.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)

3.if tie in cost then chooses the lowest upstream BID->lowest upstream port ID

(Designated Port election)
1.Ports downstream facing away from the root bridge
2.Like root port, elected based on lowest root path cost->lowest BID->lowest port ID
3.all other ports go into 'blocking' mode(receives BDPUs/discards all other traffic/cannot send traffic)
4.blocking ports are the key to the loop free topology

(Spanning-tree transition process)(802.1d/CST(common Spanning-tree)(slow convergence i.e. more forwarding delay (30 - 50 secs) compared to RSTP)(blocking->listening->learning->forwarding)(Problem with CST)

1.Listening: 15 seconds of listening for BPDUs/switch sends and receives BPDUs
2.Learing: 15 seconds of learning MAC addresses/populate switch CAM table
3.Forwarding: port is forwarding traffic
4.Blocking: switch will wait up to 20 seconds (max-age) before moving a blocked port into the listening phase

5.Problem with PCs: PCs can boot faster than 30 seconds->solution:
S(config-if)#spanning-tree portfast

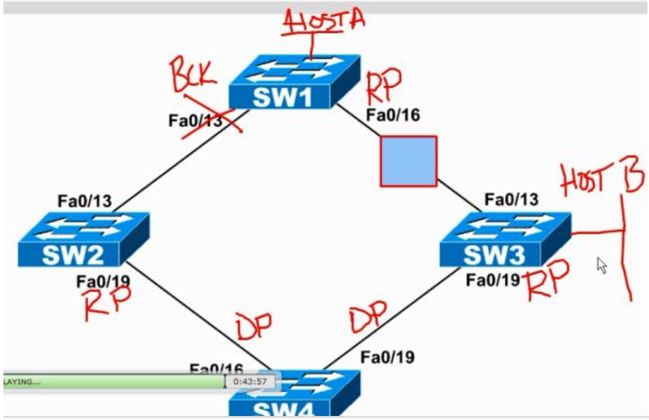
6.Problems with uplink ports: 50 seconds down time causes big problems ->solution: Rapid Spanning-tree

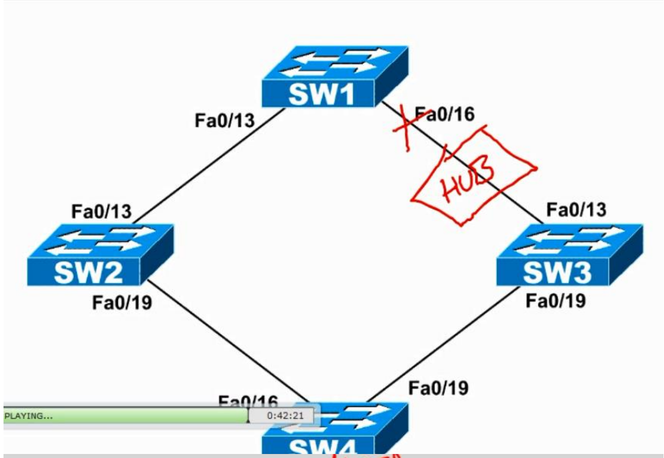
7.uplink fast and backbone fast to resolve CST problem before RSTP was released

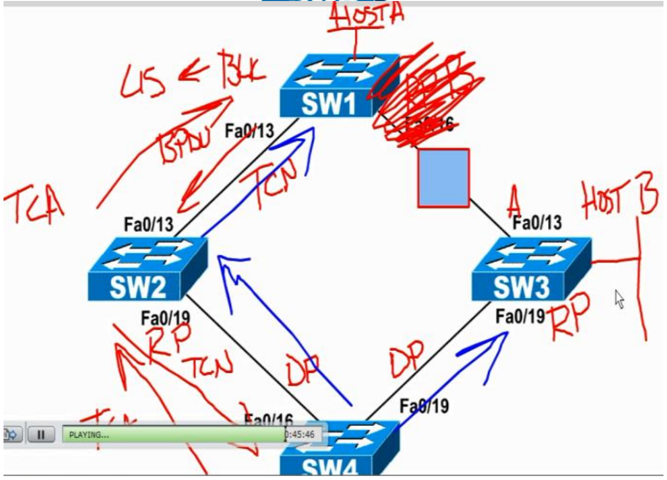
(STP Timers)(timers effect the transition between port states)(set only on the root bridge)
1.Hello timer: (how often configuration BPDUs are sent/default to 2 seconds)
2.MaxAge timer: (how long to wait in blocking state without hearing a BPDU/default to 20 seconds)(more like dead-interval in routing protocols)
3.Forward Delay: (how long to wait in each the listening and learning phases/default to 15 seconds each)
4.setting the timer locally does not effect anything it must be in the root bridge

(STP Reconvergence)(20 secs MaxAge timer)
1.The second BPDU type, TCN(topology change notification), is used to quickly age out the CAM table in case of a port state change (e.g. Forwarding->Down,BLocking->Designated)
2.TCN is sent up to Root Bridge
    1.TCN sent out Root port
    2.Upstream switch sends TCAck in configuration BPDU back
    3.Upstream switch sends TCN out Root port
    4.Next Upstream switch sends TCAck in configuration BPDU back
    5.Next upstream switch sends TCN out Root port

6.Process continues until Root Bridge receives TCN
3.When Root bridge receives TCN, it replies with TCN out all ports
4.Result is that CAM aging time is reduced to forward dealy (deafult of 5 mins reduced to 15 seconds)

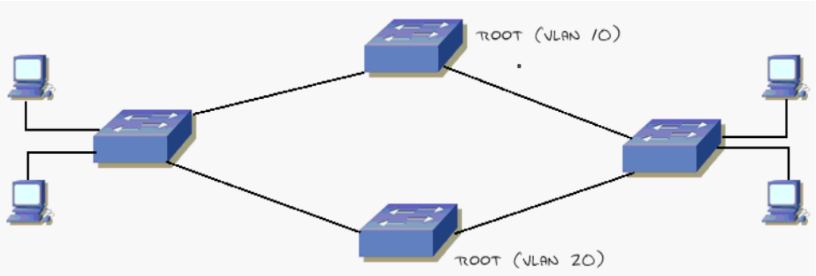


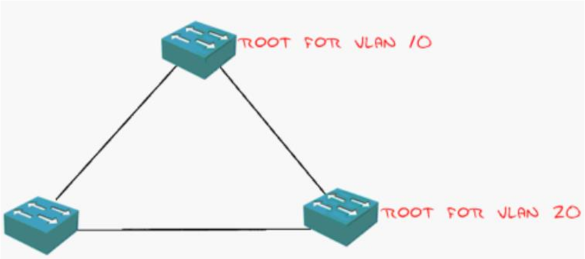


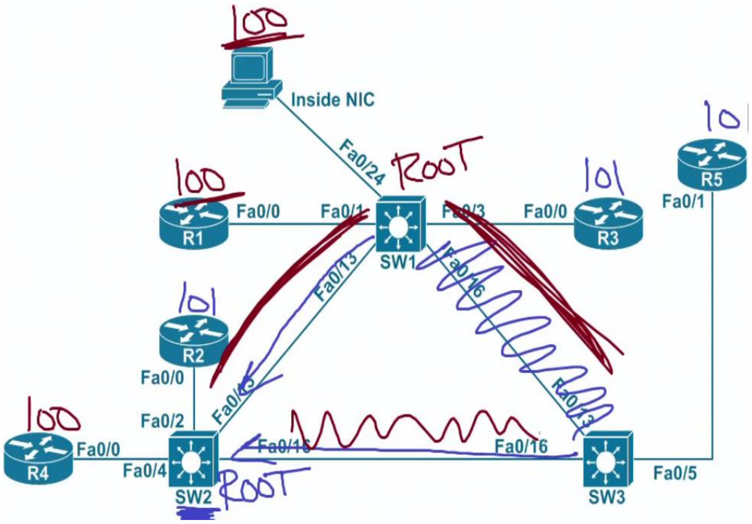


(PVST+)(sort of manual load-balancing per vlan instance)
1.Runs an instance of STP per-VLAN (sort of manual system of loadbalancing)[priority+VLAN id+MAC address](e.g.32768+1+mac address)

2.allows different root bridges per vlan (one root bridge elected for each VLAN)(separate root port and designated port elections per VLAN)(by deafult the path selection and root bridge is the same for all vlan instances as the priority is the same for all)(pvst instance is not created until you create a vlan)
3.default for all cisco switches
4.helps load-balance more effectively

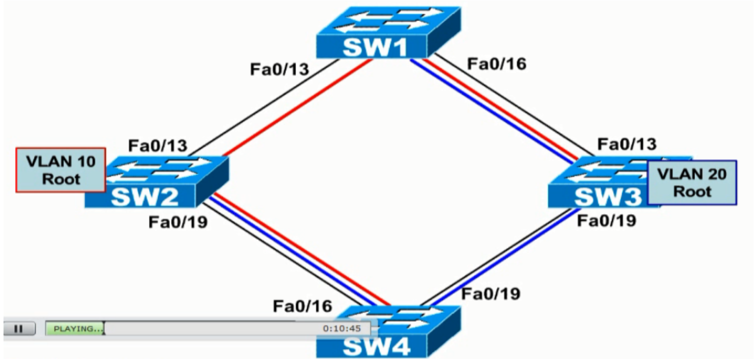






(Per-VLAN Path Selection)(layer 2 traffic engineering/path selection)

1.Changing the bridge priority per-vlan basis
2.Changing inididvidual port costs at the port level per-vlan basis

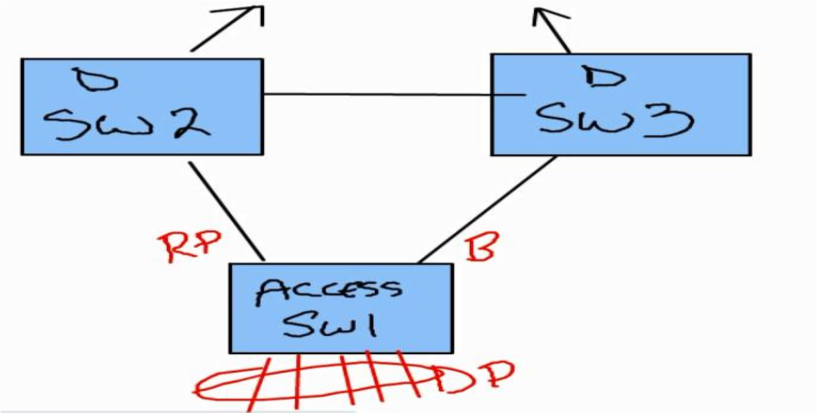


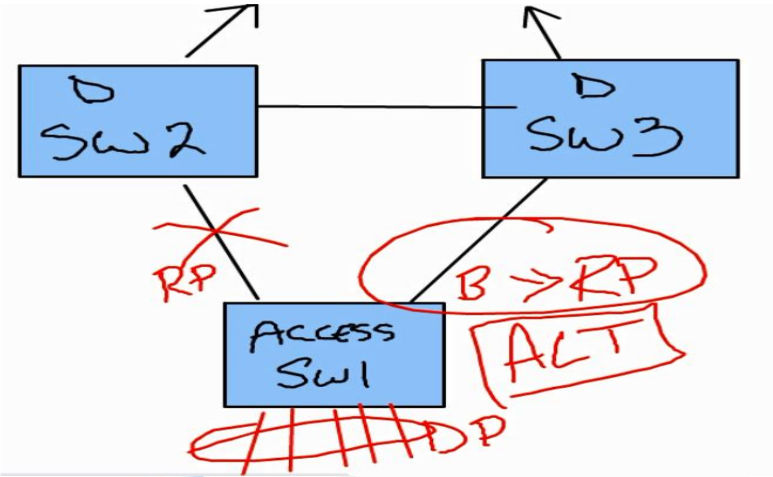
(Cisco's 802.1d Convergence Enhancements)(sub-second convergence)

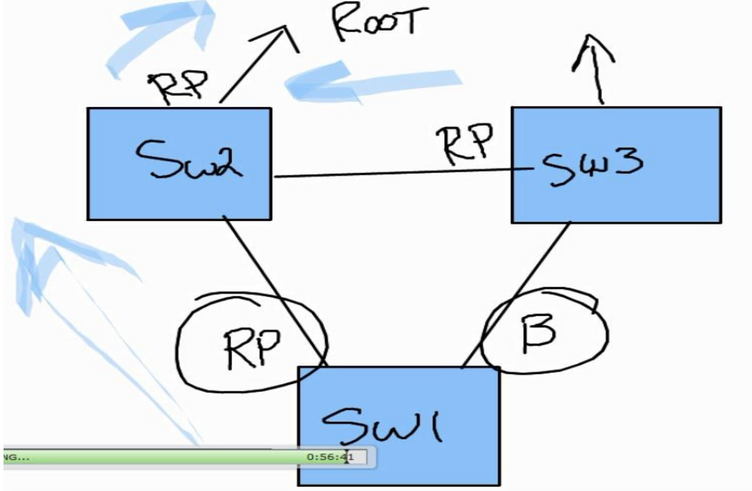
1.Portfats: (saves 30 secs)(port turns from disabled to enable state immidiately)(end hosts need not be subject to forwarding delay)(also effects TCN generation(flushes CAM table/CAM table of the whole broadcast domain ages-out))(if we activate portfast it will not generate TCN)(shows as p2p edge port)

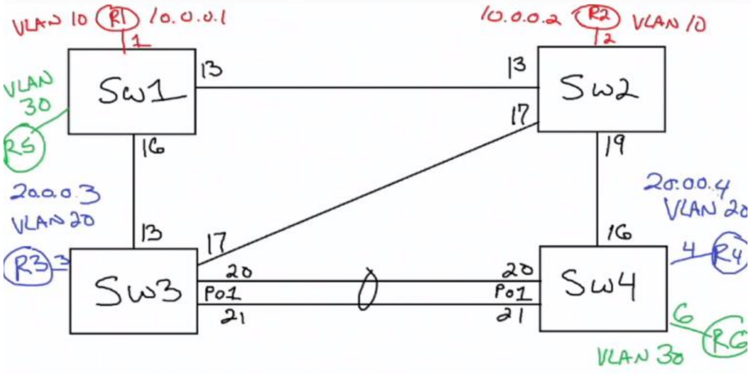
2.uplinkfast: (saves 30 secs)(direct root port failure should reconverge immidiately if alternate port avaiable)(flood the entire CAM table out to update the nei)(saves 30 secs i.e. LRN->LIS->FWD)

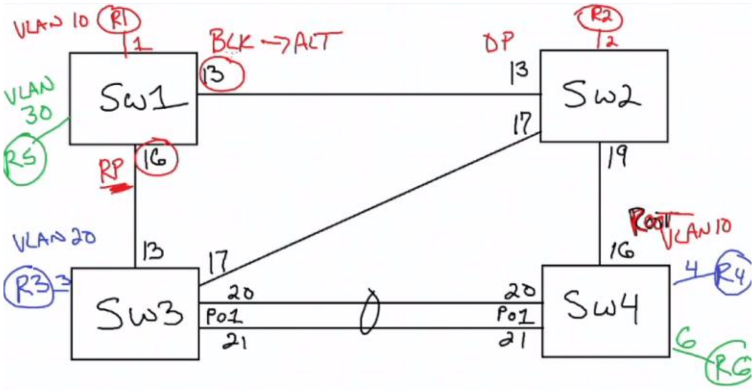
3.Backbonefast: (saves 20 secs)(indirect failures should start recalculating immidiately)(automatically expires max-age timer)(saves 20 secs i.e. BLK->LRN->LIS->FWD)(if an inferior BPDU is received on the root port that means someone upstream had a network failure and should go around and reconverge around it)

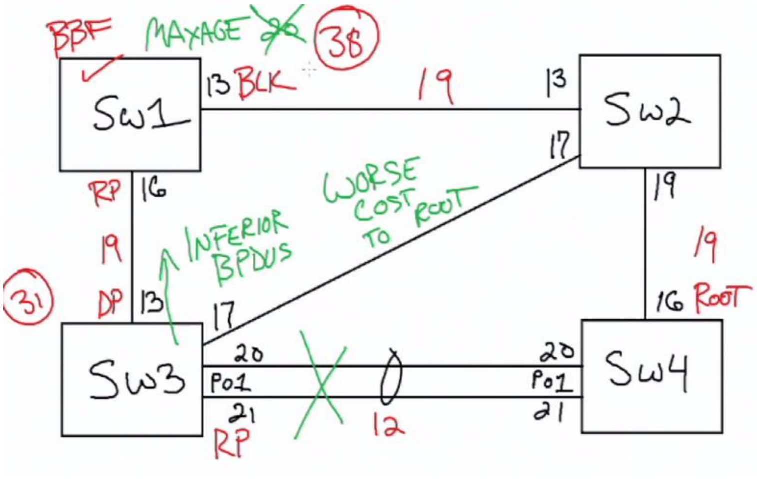



D
Sw2
D
Sw3
RP
B > RP
Access
Sw1
ALT
DP


Root
RP
Sw2
RP
Sw3
RP
B
Sw1
NG...
0:56:41


VLAN 10
R1
10.0.0.1
1
10.0.0.2
R2
VLAN 10
2
VLAN 30
R5
Sw1
13
16
13
Sw2
17
19
20.0.0.3
VLAN 20
13
17
R3
3
Sw3
20
Po1
21
20
Po1
21
Sw4
16
20.00.4
VLAN 20
4
R4
6
R6
VLAN 30





(Other STP Features)

1.BPDU Filter:

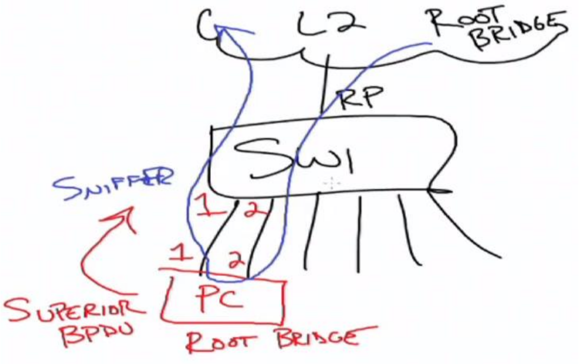
(interface level->filter BPDUs inbound/outbound i.e. do not send or receive BPDUs in the interface by simply terminating the spanning tree domain, sort of an ACL)

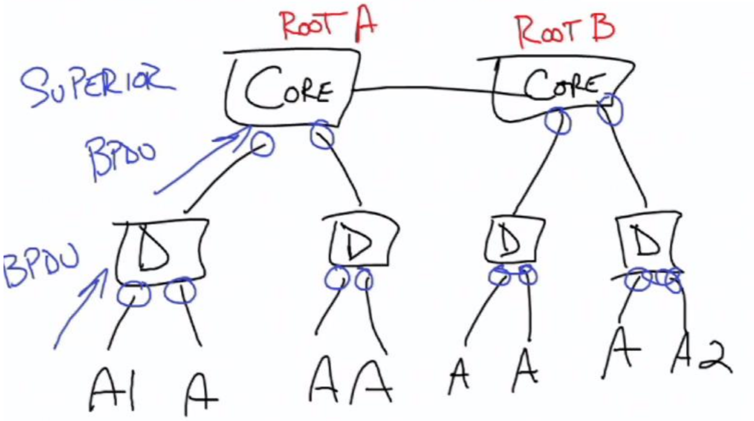
(global->BPDU is received revert out of portfast state/same functionality that rstp has with its edge ports)(so you don't have to worry about configuring portfast individually on each port)(not a security mechanism)(usually filter bpdu going to access layer)

2.BPDU Guard: (If BPDU is received shut port down)(if BPDU received it puts the port into err-disable state)

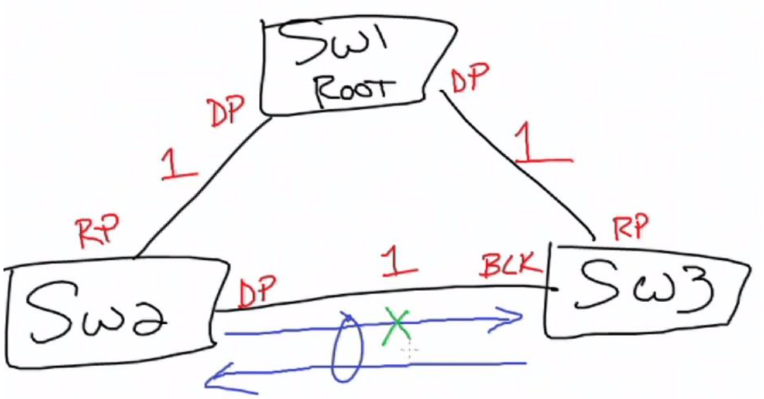
3.Root Guard:(If superior BPDU is received shut port down)(e.g. in metro ethernet and if you are service provider then the CE can never be the root bridge even if their engineers make a mistake)(also used to protect from layer2 man-in-the-middle attack)(security mechanism)

4.Loop Guard and UDLD: (unidirectional link detection)(prevent unidirectional links)(usually happens if using fiber optic cable as two different channels, send and receive pair, makes it full-duplex)(if one send or receive channel goes down, then the other side could think that the nei is still up, but still can't send traffic etc.)(so must be sending and receiving BPDUs)

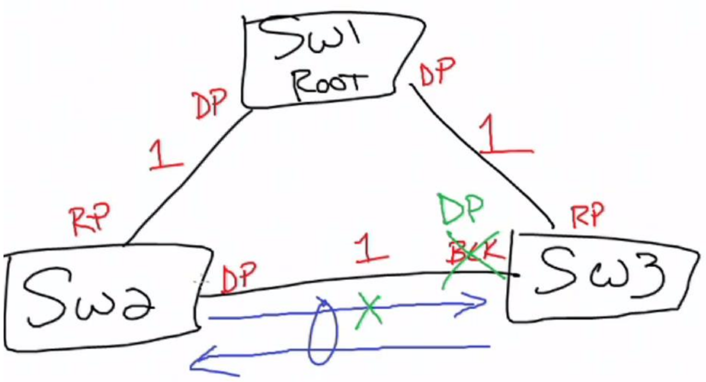


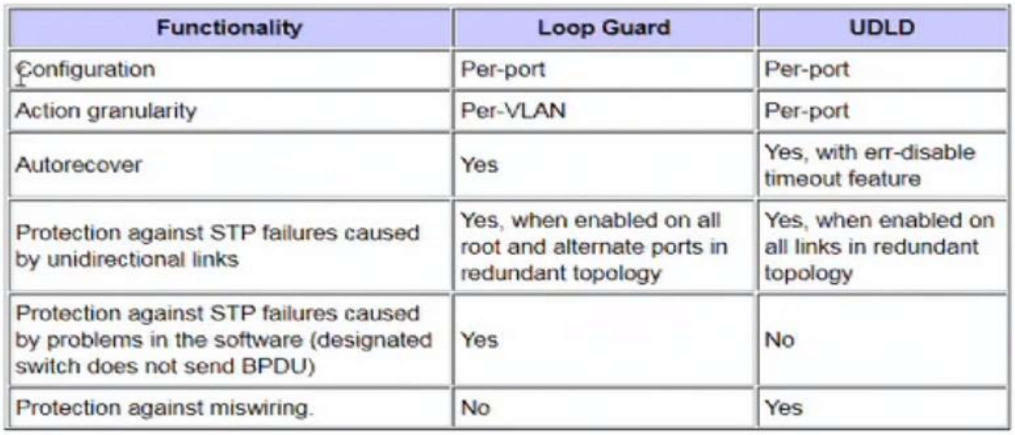


->(root guard is activated on the downstream interfaces on distribution layer switches and the access layer switches)



->(SW3 will wait for the max-age timer to expire and will become desg port after that which will cause loop)(its a layer 1 issue)(solution need some sort of layer 1 keep alive i.e. loop guard and UDLD)





Functionality	Loop Guard	UDLD
Configuration	Per-port	Per-port
Action granularity	Per-VLAN	Per-port
Autorecover	Yes	Yes, with err-disable timeout feature
Protection against STP failures caused by unidirectional links	Yes, when enabled on all root and alternate ports in redundant topology	Yes, when enabled on all links in redundant topology
Protection against STP failures caused by problems in the software (designated switch does not send BPDU)	Yes	No
Protection against miswiring.	No	Yes

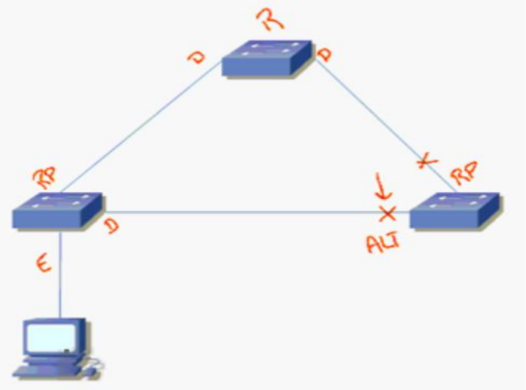
(Rapid STP/RSTP/802.1w) (fast convergence/1-2 secs forwarding delay)(faster convergence than cisco's 802.1d enhancements) (proactive system/redefined port roles)(remembers backup ports and converges faster/alternate port)(simplifies port states and uses handshaking proposal/agreement process for rapid convergence)
1.Root port: used to reach the root bridge
2.Designated port:forwarding port, one per link
3.alternate port: discarding port(blocked port), backup path to root
->host ports need to be in portfast for it to be rapid

(RSTP Port States)
1.Discarding(LIS) (Disabled+Blocking+listening combined)(dropping frames)
2.Learning(LRN) (dropping frames but building the CAM)
3.Forwarding(FWD) (normal forwarding)

(RSTP Port Roles)
1.Root Port(Root)
2.Designated Port(Desg)
3.Alternate Port(Altn)(alternate but less desirable path to the root)(allows the equivalent of uplinkfast)(operates in discarding state)
4.Backup port (Altn) (backup designated port)(could be alternate if both root and designated ports go down)(operates in discarding state)
5.Edge Port (Edge P2p/Desg) (e.g. IP phone, end-host etc.)(requires portfast command)(equivalent of PVST+ portfast enabled ports)(immidiately transitions to forwarding)(maintains edge status as long as no BPDUs are

received)(if BPDU received, remove edge status and generate TCN)

(Advantages of RSTP)
1.it does not forget ports
2.safety timers of STP are eliminated
3.any changes (up/down) to the trunk ports flood through the network to other switches (TC packets)
4.backward compatible with 802.1d standard



->if you try to convert the uplink port to another switch into a portfast port it will not convert due to the BPDUs it detects, as it could cause loops

(RSTP Link Types)(Non-edge ports fall into two types)
1.Point-to-point (full-duplex ports)(only p2p designated ports sue proposal process for rapid convergence)
2.Shared (half-duplex ports)

(RSTP Proposal Process)
1.Root bridge sends proposal out designated ports
2.If downstream switch agrees that root bridge has superioir BPDU (all other non-edge ports blocked)(agreement sent to back out port proposal received on)(port immidiately transitioned to root port)
3.Proposal continues downstream
4.Proposal and agreement process typically happens sub-second
5.If no response to proposal received, revert to Listening and Learning (backward compatibility to 8021.d)

(RSTP Convergence)
1.In 802.1d, BPDUs are only generated by root bridge (all other bridges forward them on)
2.In RSTP, each brdige generates BPDU every hello interval (2 secs by default)
3.If 3 hellos are missed from a nei, reconvergence begins (6 secs vs. 20 secs MaxAge)

(RSTP PVST+)
1.Same as PVST+, but uses RSTP enhancements for rapid convergence
2.Configured as: spanning-tree mode rapid-pvst

(Multiple Spanning-tree Protocol/MST)(802.1s)(groups vlans into single instance compared to cisco's versions)(automatically enables RSTP)

1.IEEE (802.1s) response to PVST/PVST+

2.supports multiple user-defined instances of spanning-tree (PVST+ uses one instance per vlan)(MST uses definable instances)

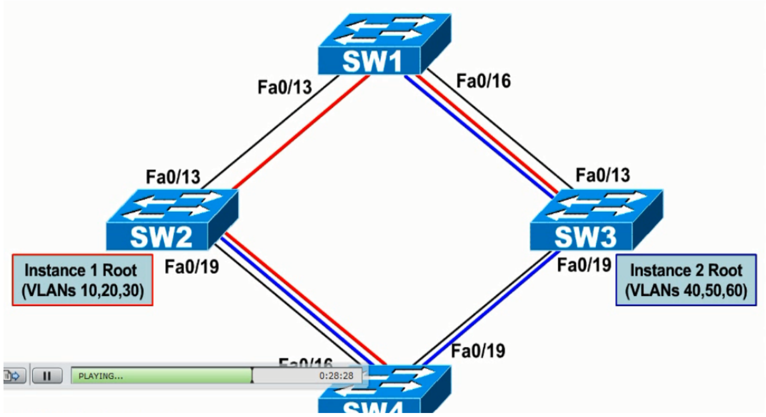
3.not as resource intensive as PVST/PVST+

4.Automatically runs RSTP

5.highly scalable(switches with same instances, configuration revision number, and anme form a 'region')(different regions see each other as virtual bridges)

6.same election process as CST/PVST ((Root Bridge(Lowest BID)->Root Port(Lowest cost)(Lowest upstream BID)(Lowest port ID)))

7.VTP v3 advertises



(MST with Multiple Regions/Inter-Region MTS)

1.MST Region is defined by:

1.VLAN to instance mappings

2.Region name

3.Revision number

2.Inter-Region path selection uses a CST(COmmon Spanning-tree)

1.Intra-region MSTIs(multiple spanning-tree instances) are collpsed into CIST(common internal spanning-tree)

->every region is considered as a cloud (as a single node for inter-region CST/acts as one single bridge for CST) with its own intra-region root bridge(runs MST) and the inter-region root bridge (runs CST)

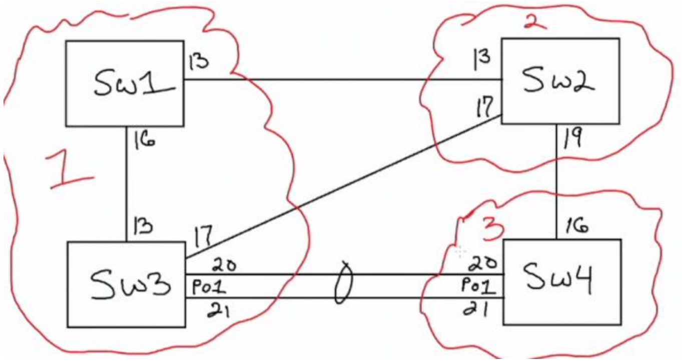
->so regional root bridge (is the intra-region root bridge) and root (is the overall root bridge for inter-region)

(MST Interoperability)

1.MST is backwards compatible with legacy CST(802.1d) and PVST+

2.Behaves like inter-region MST

3.CST Root must be within MST domain



->still have the same vlan numbers overall (nothing to do with the broadcast domain). This is to with path selection for spanning-tree

(Spanning-tree Protocol)
::::::::**Commands/Config**:::::::::::

```
S1(config)#spanning-tree vlan 1,100,101 root primary (will decrease the
priority automatically)(rrot bridge macro/sets local priority based on
current root bridge)
or
S1(config)#spanning-tree vlan 1,100,101 root secondary  (this is set if
you want a switch to act as a backup for the primary root bridge)
or
S1(config)#spanning-tree vlan 1 priority <0-61440>  (should be increment
of 4096) (to set a specific priority value/0 being the best priority)
->change the mac addresses on router interfaces for the purposes of lab
R(config-if)#mac-address 0000.0000.0001

S(config)#spa portfast default (to enable portfast on all the ports in
access mode)
S(config)#int range fa 0/2 - 24
S(config-if)#spanning-tree portfast (saves 30 seconds)

S1(congfig-if)#spa portfast trunk (forcing portfast even if the interface
is in trunking mode as the other end could be a server)

S1(config)#spa uplinkfast (saves 30 seconds)
S1(config)#spa backbonefast (saves 20 seconds)

S1(config)#int fa 0/0
S1(config-if)#spa bpdufilter enable (filters out bpdu/won't send or
receive/inbound and outbound filter)

S1(config)#spa portfast default
S1(config)#spa portfast bpdufilter default (any interface where the
portfast is enabled we will not be sending bpdu out)(only outbound
filter)(will not send bpdu, but will receive them)(when it receives it in
then it disabled portfast and makes it a non-edge port and starts sending
bpdu)(security concern)
```

```
S1(config)#int fa 0/0
S1(config-if)#spa bpduguard enable (if receives bpdu it goes to err-disable
state)(some platforms have recovery timer on)(you can set a recovery timer
on as for example if someone plug hosts on all ther access ports and send
spoofed bpdu all the ports will go into err-disable state)
```

```
S1(config)#errdisable recovery interval 30
S1(config)#errdisable recovery casue bpduguard (if still receiving bpdu
frames it will go back to shutdown)
```

```
S1(config)#spa portfast default
S1(config)#spa portfast bpduguard default
```

```
S1(config)#int fa0/1
S1(config-if)#spa guard root
```

```
S1(config)#int fa0/1
S1(config-if)#spa guard loop
->(UDLD definitely configured on both neighbors)
```

```
S(config-if)#spa vlan 1 cost 9999 (to modify port's cost for a specific
vlan number/to change the local port path cost)(Layer 2 traffic
engineering/path selection)
S(config-if)#bandwidth <bps>
```

```
S(config-if)#spa vlan 1 port-priority <0-240>(modify the port id to change
path selection)(must be increments of 16)(in spa detail command it shows
as designated port id 128.20, where 128 is the port priority and 20 is the
port number)(so we can change the port priority)(layer 2 traffic
engineering/path selection)
```

```
S1(config)#service timestamps debug datetime msec
S1(config)#service timestamps debug uptime
S1(config)#service timestamps log uptime
```

```
S1(config)#mac address-table aging-time 300 (i.e. 5 mins)
```

```
S1(config)#spa vlan 1 hello-time <1-10>  (in seconds)
S1(config)#spa vlan 1 forward-time <4-30> (in seconds)
S1(config)#spa vlan 1 max-age <6-40> (in seconds)
(timers are set on root bridge and only then replicated to other bridges,
in case of STP/PVST/PVST+, whereas in case of RSTP and MST it could be
anywhere)(fastest you can converge with normal stp is hello 1 sec,
foward-delay 4sec and max-age 6sec, but we need sub-second convergence)
```

```
R1(config)#brdige 1 protocol ieee  (to turn stp,vtp,cdp etc. on the router)
R1(config)#bridge 1 priority 65535  (highest priority least likely to be
the root bridge)(if you set priority 0 it will try to become root bridge
depending on if the root guard featue it set)
R1(config)#int fa0/0
R1(config-if)#bridge-group
```

::::::Verification/Show commands::::::

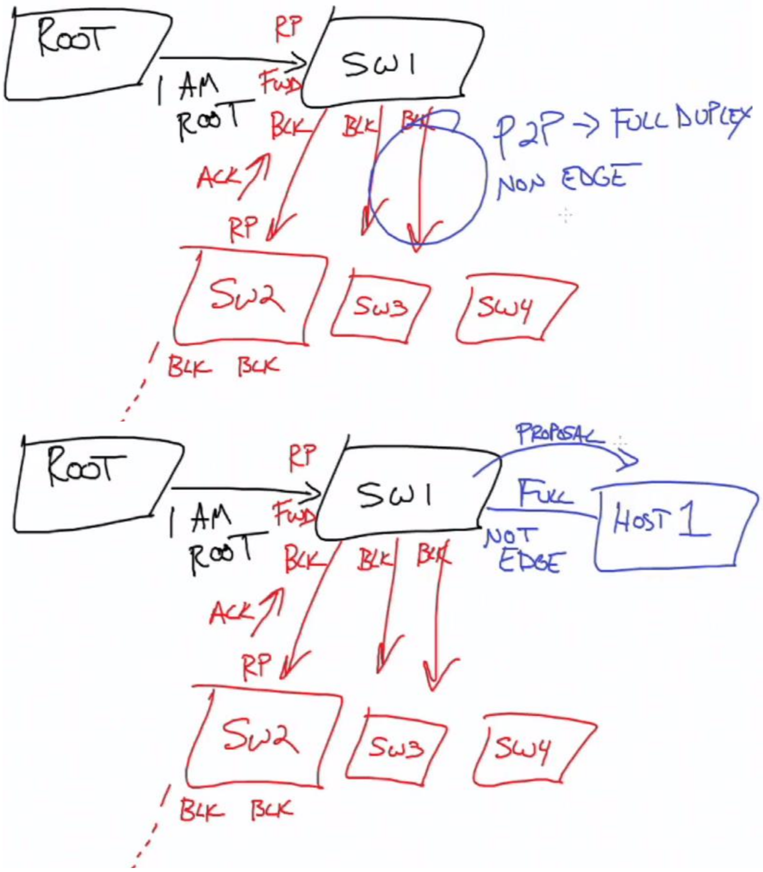
```
sh spa
(Root ID/Bridge ID  ->  priority/address/cost/port/hello timer)

sh spa vlan 1  (to see spa of specific vlan instance/shows timers)

sh spa detail
->(shows BID/sys-is/mac/cost/port(root/desg(FWD)/Blk) for both current
bridge and root)
->(designated path cost=remote neighbors cost to reach the root advertised
to us/0 means it directly connects to the root)(path cost of root bridge
send to the neighbor is always 0)
->(port path cost/root path cost=cost to reach the nei who is advertising
designated path cost)
->to check if sending and receiving BPDUs

sh spa vlan 1 detail (separate topology databases per vlan)

sh spa root (shows the cumulative path from end to end/instead of looking
at sh spa detail we can see the path cost from here)

sh spa int fa0/13 detail

sh spa int fa0/13 detail | in bpdu (shows if send/receiving BPDUs)
sh spa int fa0/1 portfast (shows if portfast enabled for all the instances)

sh interface status
sh run int fa0/16
sh cdp nei
sh mac address-table dynamic vlan 1  (blocking port won't show up as traffic
is being forwarded the other ports)
sh arp
arp -a (PC command)
sh processes cpu (cpu utilization)
sh processes cpu history   (shows graph of cpu history over time)
ping 192.168.1.1 repeat 10000

sh int trunk
sh run | in spanning
sh int fa0/1 switchport

sh errdisable recovery

clear arp  (clearing arp cache)
sh mac address-table aging-time

clear spa counter  (clears the bpdu counters in sh detail command)

debug spa
debug spa events  (shows progression to different phases)
debug spa uplinkfast (shows if receives inferior bpdu)
debug spa backbonefast
```

(Rapid Spanning-tree Protocol/RSTP-PVST)(smilar commands as MST/CST)
::::::::Commands/Config:::::::::::

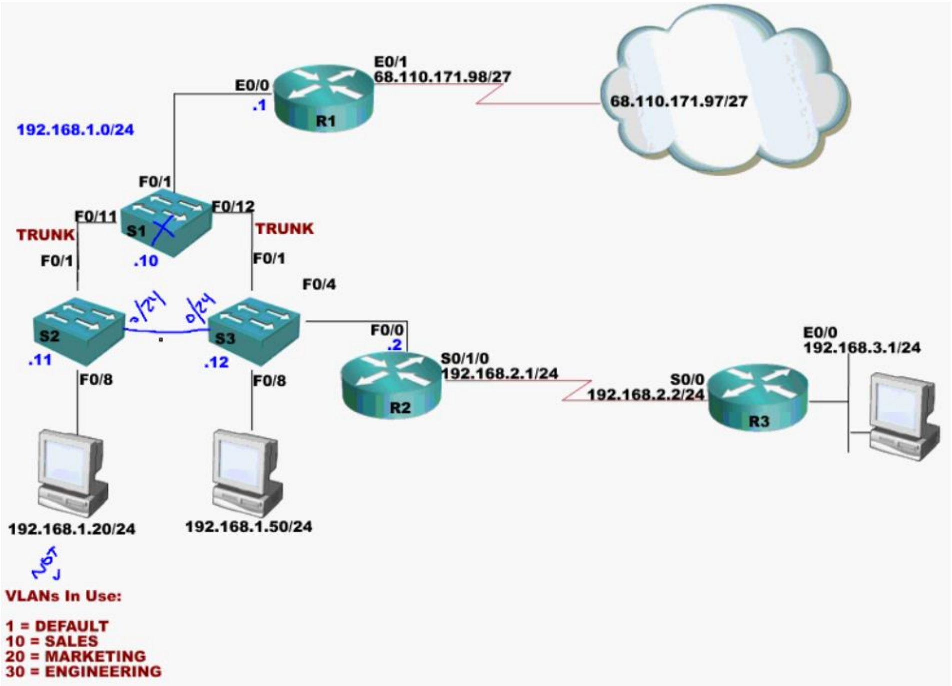
```
S1(config)#spanning-tree mode rapid-pvst
```

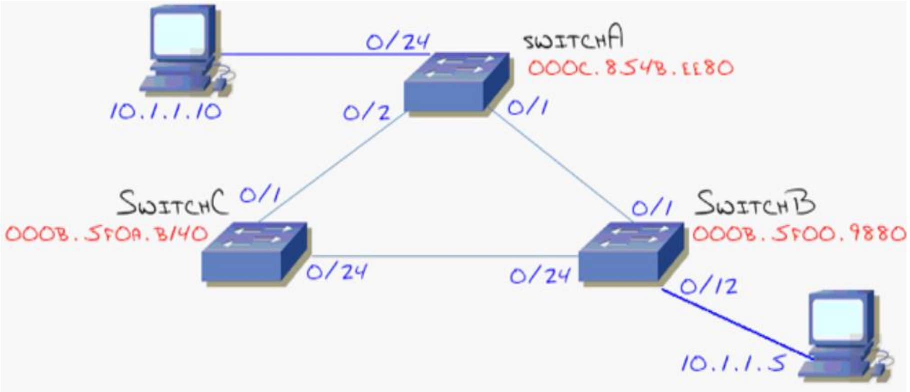
->no need to configure uplinkfast or backbonefast but you can configure portfast for fast convergence.
->rest of the mechanisms like path selection and root election are the same as CST except for the fast convergence
:::::Verification/Show commands:::::

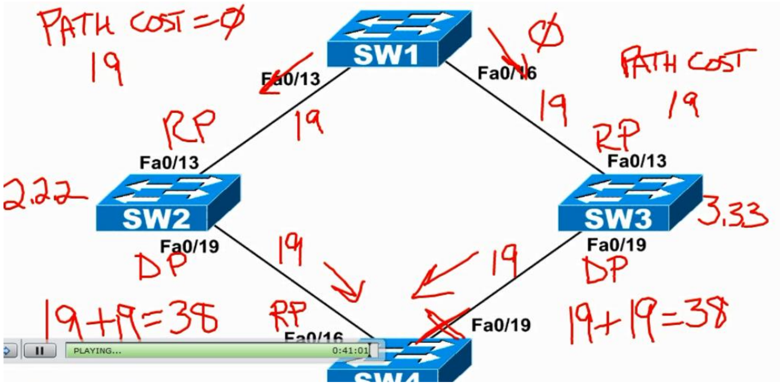
```
sh spa
```

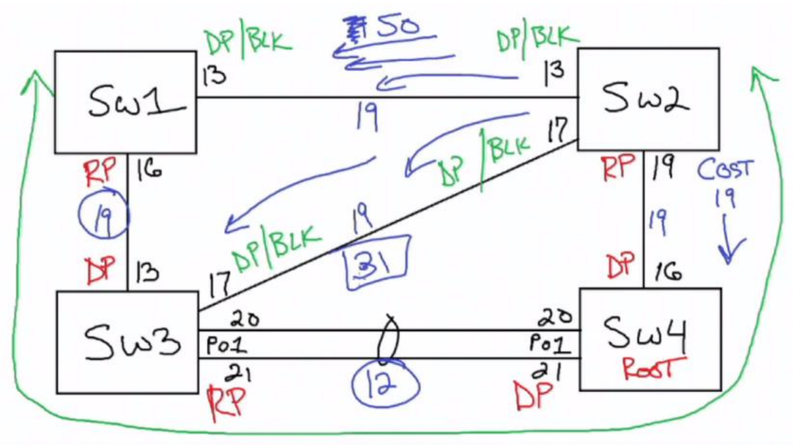


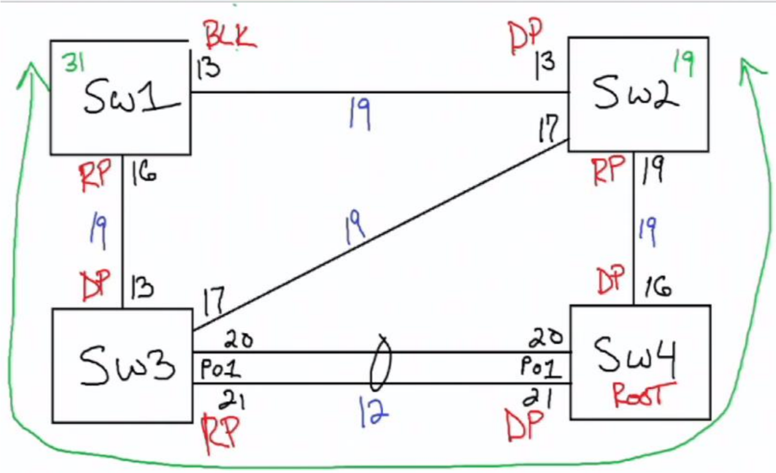
->(every end host port must be edge port for rapid convergence and can be done by portfast command)

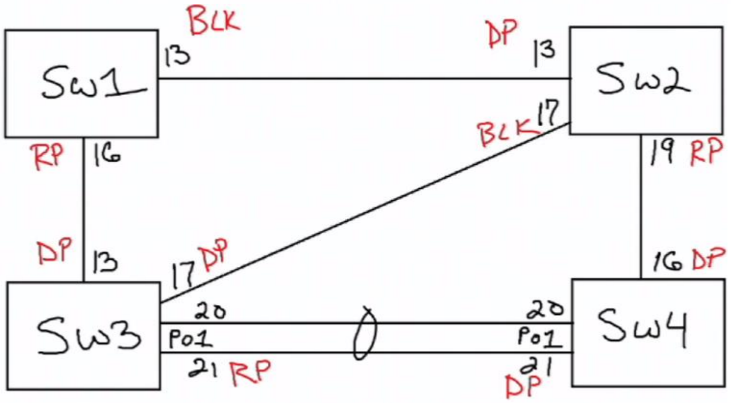

E0/1
68.110.171.98/27
E0/0
.1
R1
68.110.171.97/27
192.168.1.0/24
F0/1
F0/11
F0/12
S1
TRUNK
TRUNK
.10
F0/1
F0/1
F0/4
0/24
0/24
S2
S3
.11
.12
F0/0
.2
S0/1/0
192.168.2.1/24
S0/0
192.168.2.2/24
E0/0
192.168.3.1/24
R2
R3
F0/8
F0/8
192.168.1.20/24
192.168.1.50/24
NOT
VLANs In Use:
1 = DEFAULT
10 = SALES
20 = MARKETING
30 = ENGINEERING

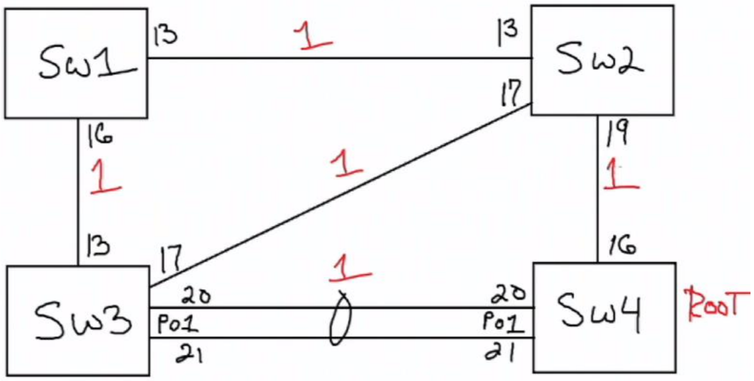

0/24
SWITCHA
000C.854B.EE80
10.1.1.10
0/2
0/1
SWITCHC
0/1
000B.5F0A.B140
0/1
SWITCHB
000B.5F00.9880
0/24
0/24
0/12
10.1.1.5

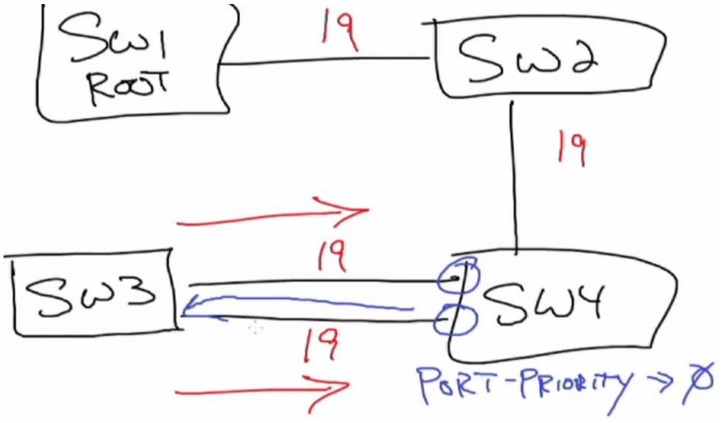

PATH COST = Ø
19
SW1
Ø
Fa0/13
Fa0/16
PATH COST
19
RP
19
19
RP
19
Fa0/13
Fa0/13
2.22
SW2
SW3
3.33
Fa0/19
19
19
Fa0/19
DP
DP
19+19=38
RP
19+19=38
Fa0/16
Fa0/19
PLAYING...
0:41:01
SW4


Sw1
Sw2
Sw3
Sw4
ROOT
DP/BLK
50
DP/BLK
13
13
19
17
DP/BLK
RP
16
19
COST
19
19
19
DP/BLK
31
DP
13
17
DP
16
20
Po1
21
12
20
Po1
21
RP
DP

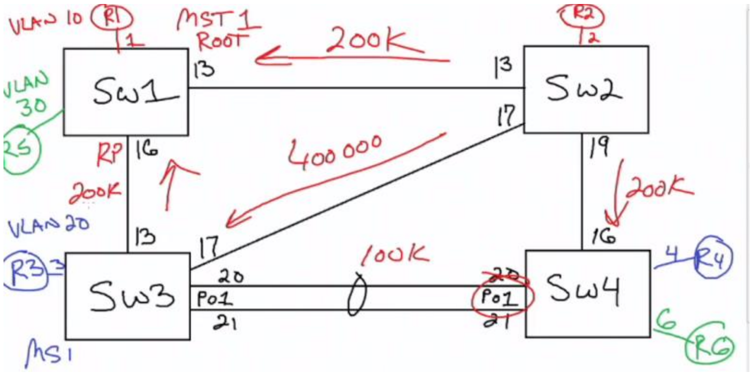

BLK
DP
31
Sw1
13
19
13
Sw2
19
17
RP
16
RP
19
19
19
19
DP
13
DP
16
17
Sw3
20
Po1
21
RP
12
Sw4
ROOT
20
Po1
21
DP


BLK
DP
Sw1
13
13
Sw2
BLK
17
RP
16
19
RP
DP
13
17
DP
16
DP
Sw3
20
Po1
21
RP
Sw4
20
Po1
21
DP


Sw1
13
1
13
Sw2
17
16
1
1
19
1
13
17
16
1
Sw3
20
Po1
21
Sw4
20
Po1
21
ROOT



->in case of etherchannel/port-channel cumulative bandwidth of member interface/individual ports is considered for stp path cost
->blocked links are only used for BPDUs



(Multiple Spanning-tree Protocol/MST)

:::::::::**Commands/Config**::::::::::::

```
S1(config)#spanning-tree mst configuration
S1(config-mst)#name MST_REGION1
S1(config-mst)#revision 1
S1(config-mst)#instance 1 vlan 10, 20, 30
S1(config-mst)#instance 2 vlan 40, 50, 60
S1(config)#spanning-tree mode mst (it will come up with a message if the other nei switches are still on stp/rstp and blocks the links/so need pretty big maintenance windows if done in the production network)
->(vlans not assigned to any instances will fall back to deafault instance 0)

S1(config)#spa mst 1 root primary (1 is an instance)
S1(config)#spa mst 1 root secondary (1 is an instance)

S1(config)#spanning-tree mst 1 priority 4096 (1 is the instance)
S1(config)#int fa0/13
S1(config-f)#spanning-tree mst 1 cost 50000 (path selection/layer2 traffic engineering)
```

```
S1(config-if)#spa link-type point-to-point
S1(config-if)#spa link-type shared
```

```
S1(config)#service timestamps debug datetime msec
```

::::::**Verification/Show commands**::::::

```
sh spa mst
sh spa
sh spa mst 1  (cost values in mst are larger)(1 is the instance)
sh spa mst 1
```

```
sh spa mst config (shows vlans corresponding to instances)
```

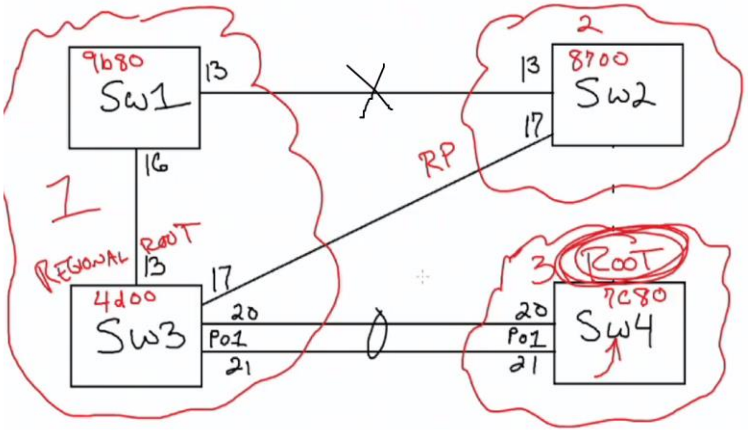
```
sh run | inc spa
sh spa root (if the root port shows is empty for an MST instance that means
the current bridge is the root for that particular instance)
```

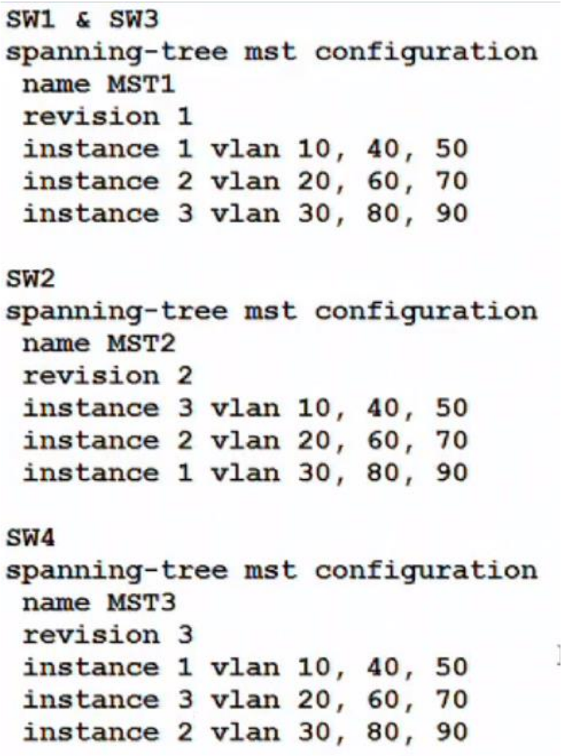
```
sh vlan bri (does not match with sh spa mst config)(vlans that we create
are advertised by vtp)(so vtp will work for vlans showing here)
```

```
debug spa mst states
debug spa mst proposals
debug spa mst roles
```

(MST with Multiple Regions)

:::::::::**Commands/Config**:::::::::::





```
SW1 & SW3
spanning-tree mst configuration
 name MST1
 revision 1
 instance 1 vlan 10, 40, 50
 instance 2 vlan 20, 60, 70
 instance 3 vlan 30, 80, 90

SW2
spanning-tree mst configuration
 name MST2
 revision 2
 instance 3 vlan 10, 40, 50
 instance 2 vlan 20, 60, 70
 instance 1 vlan 30, 80, 90

SW4
spanning-tree mst configuration
 name MST3
 revision 3
 instance 1 vlan 10, 40, 50
 instance 3 vlan 20, 60, 70
 instance 2 vlan 30, 80, 90
```

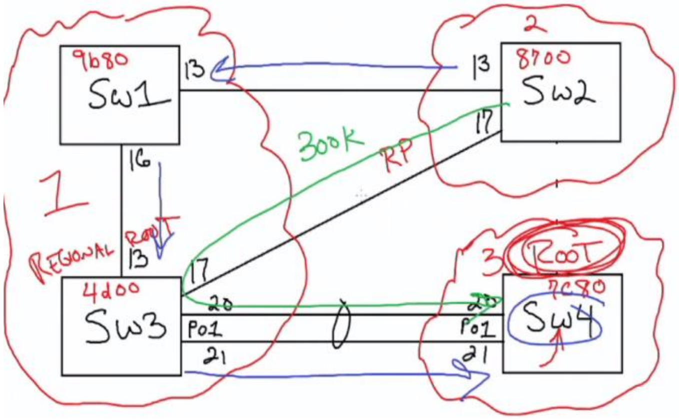
```
SW1(config)#spa mode mst
SW2(config)#spa mode mst
SW3(config)#spa mode mst
SW4(config)#spa mode mst
```

```
S(config-if)#spa mst 0 cost 200000000 (for path selection between regions)
S(config)#spa mode pvst (using pvst in one of the regions)
```

::::::**Verification/Show commands**::::::

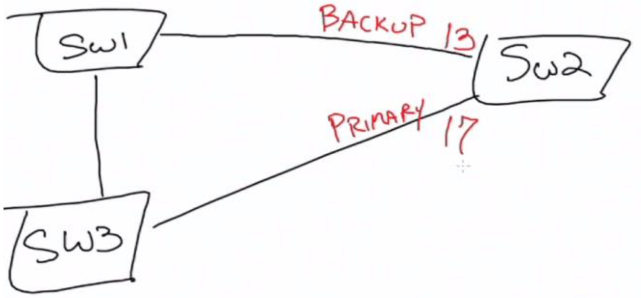
sh spa mst (bridge info/root bridge/regional root bridge info)(shows boundry links to other RSTP domains i.e. p2p bound(RSTP) or p2p bound(PVST))(shows what sort of interoperability are you running)

```
sh spa mst detail
sh spa root
sh mac address-table dynamic vlan 10
sh spa
sh run | in spa
sh int fa0/1
sh int trunk
```



->region 3 is still the root for the regions
->CST Root must be within MST domain

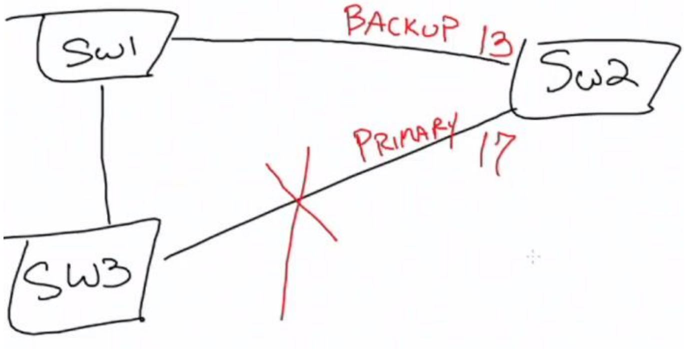
(Flex Links)(used as opposed to STP)
->allows link redundancy without STP (primary link forwards while backup link blocks)
->MAC-Address Move Update allows for faster convergence of CAM



```
SW2(config)#int fa0/17
SW2(config-if)#switchport backup int fa0/13 mmu primary vlan 1
SW2(config-if)#mac address-table move update transmit
```

```
SW1(config-if)#mac address-table move update receive
```

```
SW2(config)#int fa0/17
SW2(config-if)#switchport backup int fa0/13 preemption mode forced
SW2(config-if)#switchport backup int fa0/13 preemption delay 5
```
->preemption value can be set once the primary is back up again

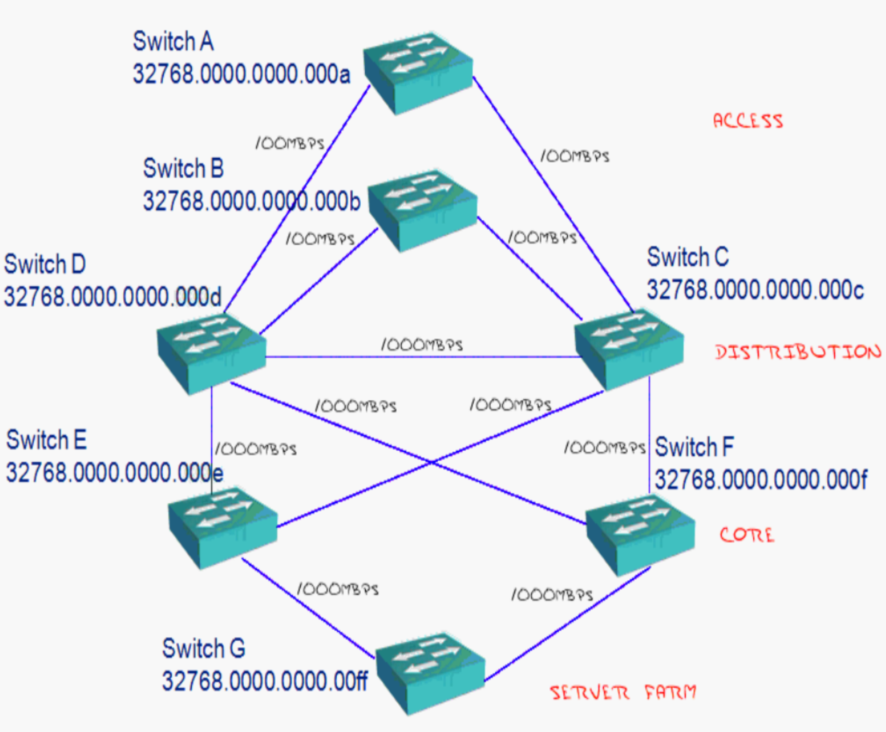


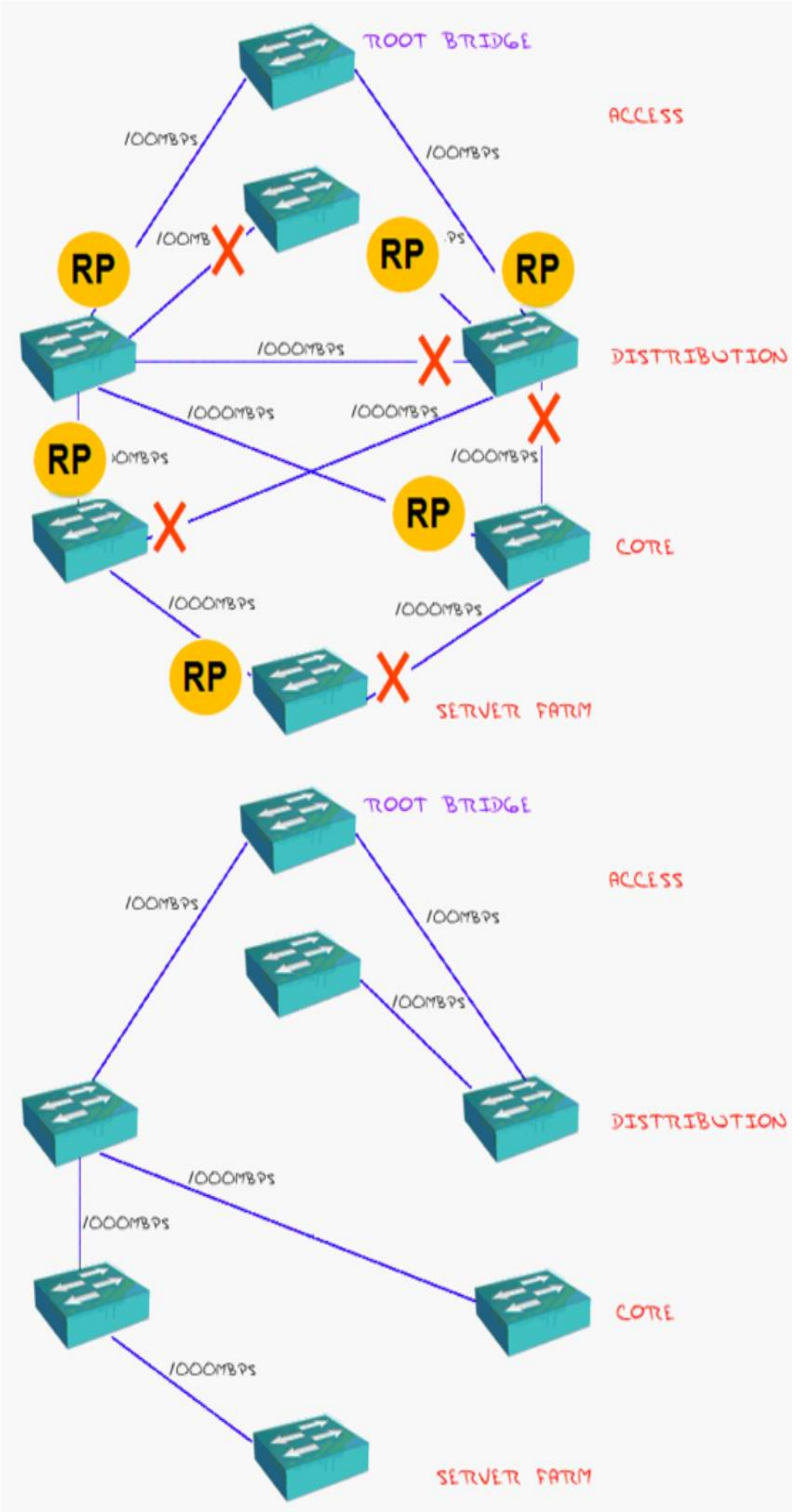
```
S(config-if)#switchport backup interface fa0/13 prefer vlan 1-10  (for
```

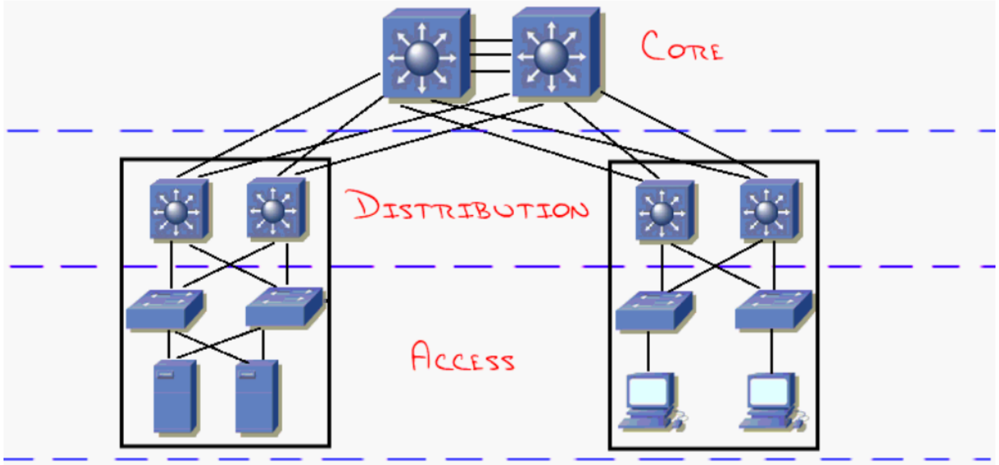
load-balancing specific vlans per port/its not per packet or per frame load-balancing rather vlan specific load-balancing)

```
sh mac address-table move update
sh interface fa0/17 switchport backup (shows the active and backup ports status)
sh mac address-table dynamic vlan 1
sh int switchport backup detail
```

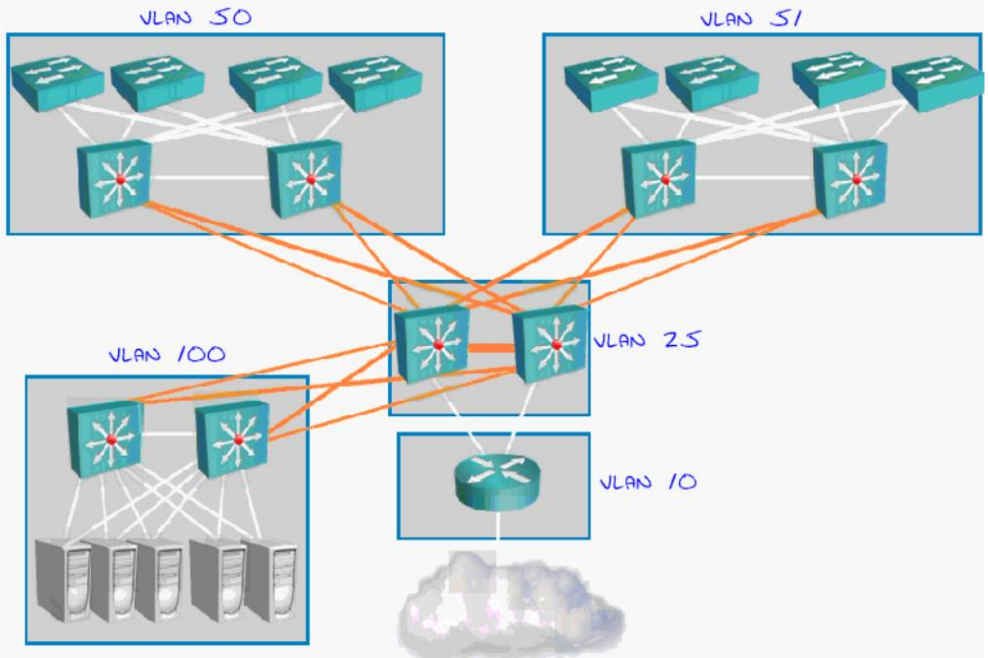
->All the priorities are tied by fault on all the switched so MAC address is used to break the tie and the oldest MAC addressed switch becomes the root brdige which could slow down the network as the oldest mac address will be on the oldest switch.(e.g. access layer switch end up becoming a root bridge, which slows down the network completely)
->STP root should be in the core of the network and if you are using three tierd desgin the STP root should be in the distribution layer




ROOT BRIDGE
ACCESS
100MBPS
100MBPS
100MB
PS
RP
RP
RP
1000MBPS
DISTRIBUTION
1000MBPS
1000MBPS
RP
0MBPS
1000MBPS
RP
CORE
1000MBPS
1000MBPS
RP
SERVER FARM
ROOT BRIDGE
ACCESS
100MBPS
100MBPS
100MBPS
DISTRIBUTION
1000MBPS
1000MBPS
CORE
1000MBPS
SERVER FARM



- Layered approach allows for easy, managable growth
- Etherchannel can provide more bandwidth on key links
- Redundant connections eliminate a single point failure



Topology Diagrams:
http://ccieccie.wordpress.com/category/ine-volume-1-lab-diagrams/