Rapid Spanning Tree Protocol

Fulvio Risso

Politecnico di Torino

Evolutions of the STP

- Rapid Spanning Tree protocol
  - Fast convergence
  - Appropriate for current network topologies (no hubs, only point-to-point links)
  - 802.1w (2001): later incorporated into 802.1D-2004
- Multiple Spanning Tree
  - Metropolitan area networks
  - Coexistence of STP and RSTP within the same domain
  - 802.1s (2002): later incorporated into 802.1Q-2005

Rapid Spanning Tree (RSTP, 802.1w)

- Modern solution for mission-critical bridged LAN
  - Fast convergence (less than 1 second)
  - Replaces other proprietary solutions with fast convergence introduced by many vendors
- Operates only on point-to-point links
  - Direct connections (no hubs)
- Major improvements
  - Fast topology convergence
  - Fast update of the filtering database
- Defines a set of symbols to be used in network design
- 802.1w and 802.1t-2001 were integrated in IEEE 802.1D-2004

Fast convergence

- Requirements
  - All the switches must be 802.1w
  - Links that may create a mesh between different switches
    - Must be point-to-point (twisted pair or fiber), full duplex
      - STP supported also shared medium (e.g. coax)
    - No links terminated on hubs (although still possible at the edge; often marked as “shared links”)
    - These links guarantee that devices at both sides of the link detect the fault at the same time and that they initiate the convergence process at almost the same time
- Convergence is usually on the order of ~ 10ms
  - Faults are immediately detected at the physical layer
  - New transition rules for ports helps to improve the convergence
  - In general, <1s

Port States and Roles (1)

- STP does not distinguish appropriately between port *roles* and *states*
- RSTP has a better separation of the two concepts
- Port States
  - Possible operational states with respect to data frames
  - STP: Disabled, Blocking, Listening, Learning, Forwarding
    - Which is the difference between Blocking and Listening?
      - It relates to the port status according to the STP topology, but from the operational point of view those ports behave the same (do not forward frames, do not learn addresses)
  - RSTP: only 3 states left (from the 5 defined in STP)
    - Discarding, Learning, Forwarding
    - *Disabled*, *Blocking* and *Listening* merged in the same state

Port States and Roles (2)

- Port Roles
  - Role of the port within the topology calculated by the spanning tree
  - STP: Root, Designated, Blocking
  - RSTP: Root, Designated, Alternate, Backup, Edge

Port states in RSTP

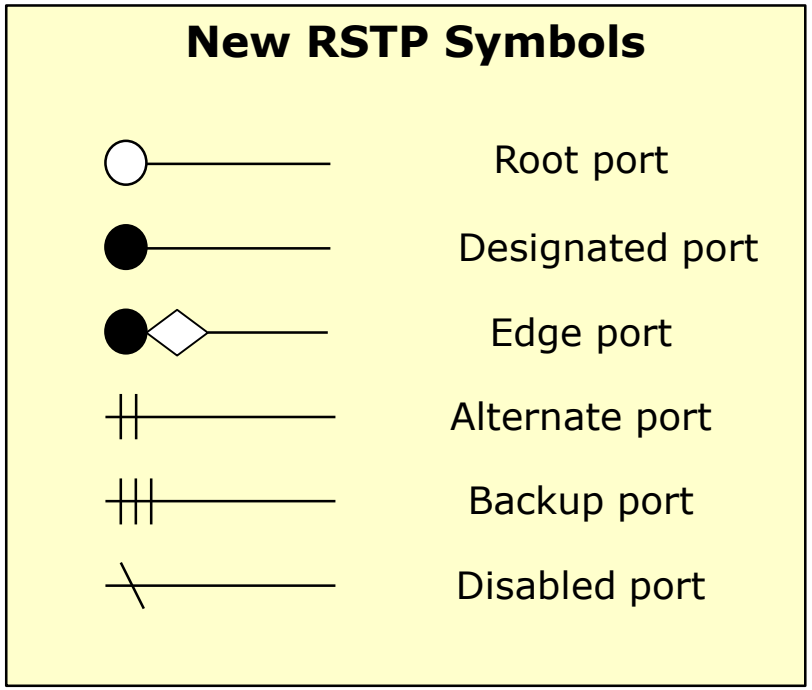
	Update filtering DB	Process frames	Possible RSTP role
Discarding	NO	NO	Alternate or Backup
Learning	YES	NO	The port is on the way to become Root or Designated
Forwarding	YES	YES	Root, Designated or Edge

States in STP

	Receive frames	Forward frames	Process received BPDU	Transmit BPDU	Update filtering DB
Disabled	NO	NO	NO	NO	NO
Blocking	YES	NO	YES	NO	NO
Listening	YES	NO	YES	YES	NO
Learning	YES	NO	YES	YES	YES
Forwarding	YES	YES	YES	YES	YES

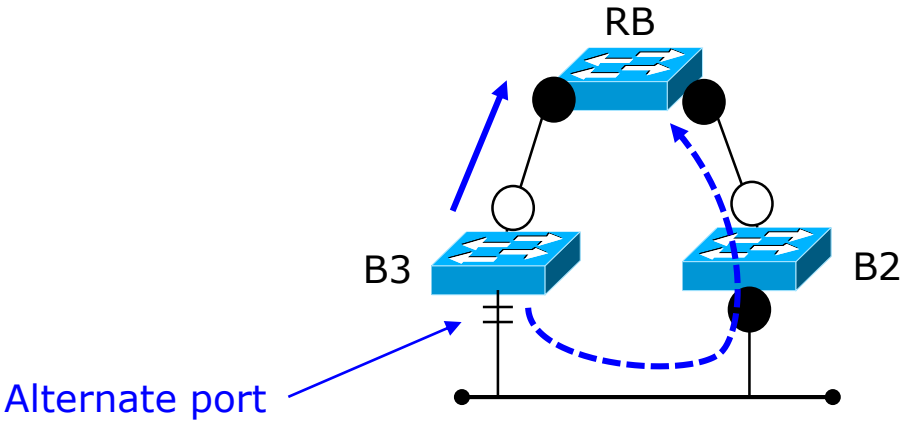
New Port Roles

- Root and Designated port remain the same
- Blocked ports are always “blocked”, but can assume one of the following states
  - Alternate
  - Backup
- New role: Edge ports



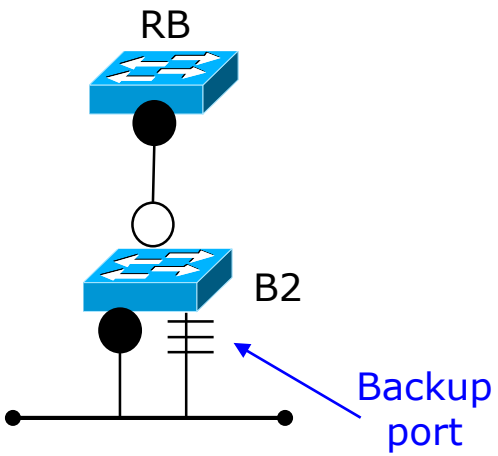
New Port Roles: Alternate

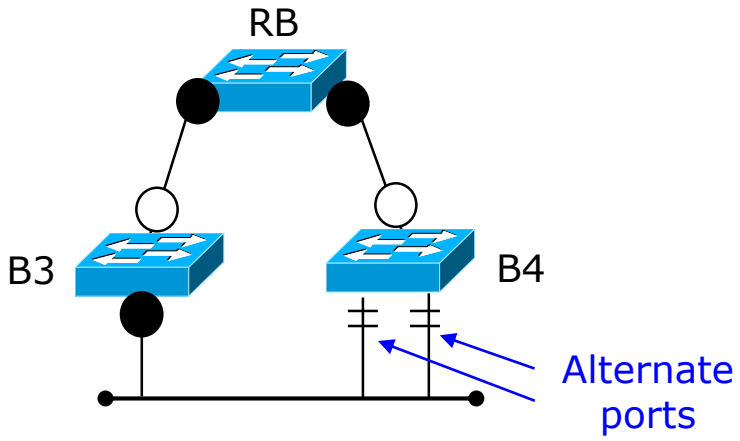
- An alternate port is a blocked port that receives a better BPDU from *another* bridge
- Alternate port provides an *alternate path* to the Root Bridge
  - It can replace the root port if that port fails
- Note: Alternate ports are **alternate for the bridge they are currently installed on**
  - E.g., B2 does not have any *alternate ports*, but this does not mean it does not have *alternate paths*



New Port Roles: Backup

- A Backup port is a blocked port that receives a better BPDU from the *same* bridge
  - It act as "backup link" for that LAN from the current bridge
    - It cannot always guarantee an alternate connectivity to the root bridge (e.g. it fails if the bridge itself fails)
  - Backup Ports exist only where there are two or more connections from the *same* bridge to the *same* LAN and *one is Designated*
    - It does no longer exist on modern networks



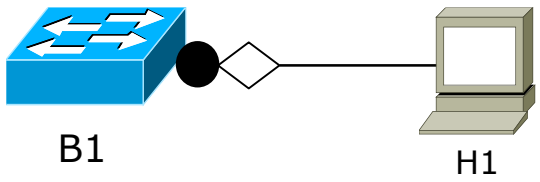


Alternate and Backup in brief

- Alternate is a replacement for the Root port
  - If the root fails, the Alternate represents a fast replacement for it
  - It is a blocked port connected to *another bridge*
- Backup is a replacement for the Designated port
  - If the designated fails, the Backup represents a fast replacement for it
  - It is a blocked port connected to *the same link of the Designated port*

New Port Roles: Edge (1)

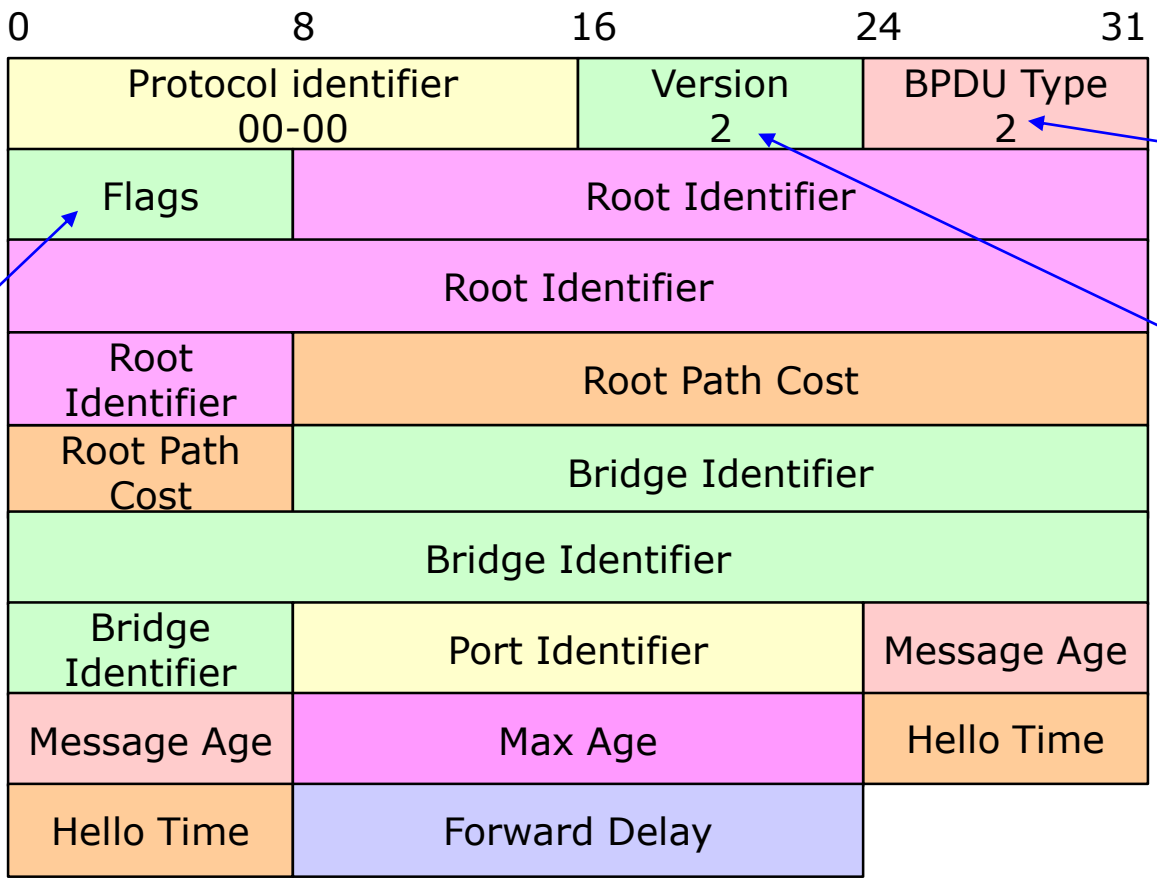
- Port that connects to a single end-station
  - No other switches are served from that point
- Needs an explicit configuration from the network admin
  - The RSTP still monitors the presence of BPDUs on that link
    - Protection against possible loops
  - In case a BPDU is received, the port will become part of the "traditional" RSTP domain and will change its status according to the well-known RSTP rules



New Port Roles: Edge (2)

- Edge ports work differently from other ports
  - As soon they detect the “link up” signal, they move immediately to Forwarding state without transition into the Learning state
    - No longer have to wait 30s (2 * Max Forward Delay) before having the port fully operational
  - Edge ports become immediately “Designated”
  - A port changing state do not triggers the transmission of a Topology Change Notification BPDU through the root port

802.1w BPDU (1)



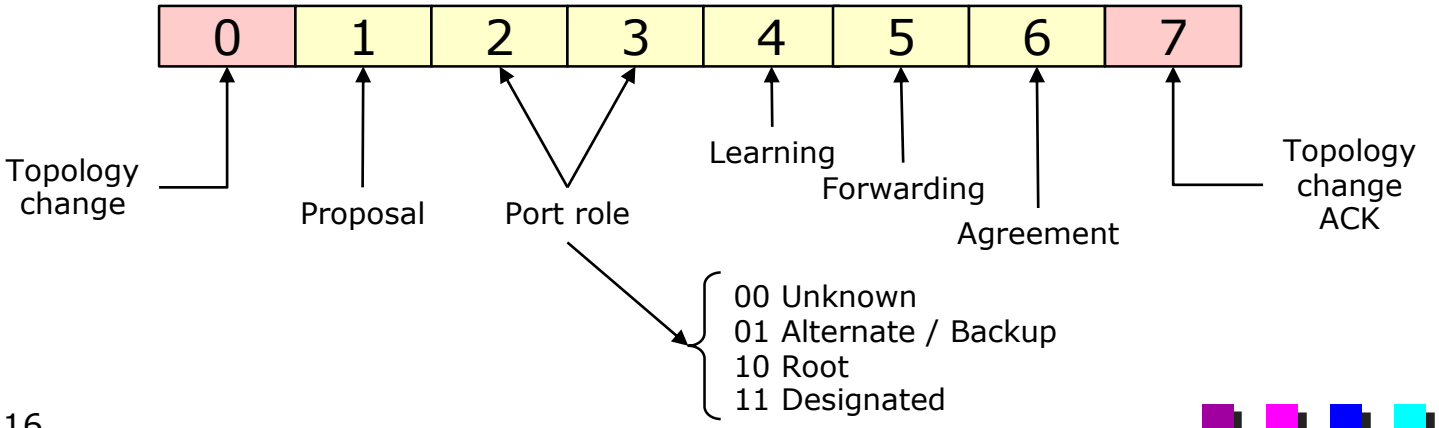
More used flags

Only configuration BPDU (no Topology Change) It was "0" in 802.1D

It was "0" in 802.1D

802.1w BPDU (2)

- Modified
  - BPDU Version (now 2)
    - Old bridges can discard new BPDUs
  - BPDU Type (now 2)
  - Flags
- New flags used for:
  - Encode role and state of the port that generates the BPDU
  - Handle the proposal/agreement mechanism



Principles of the new algorithm

- The tree is created in the same way as STP
  - Root bridge election
  - Root port definition
  - Designated port definition
  - Other ports either Alternate or Backup
- Costs and other parameters are handled the same way
  - Some parameters are no longer needed (e.g., MaxAge, ForwardDelay)
  - Kept for compatibility
- The port not selected as root or designated will become:
  - Alternate if connected to a port on a different bridge
  - Backup if connected to a different port of the same bridge

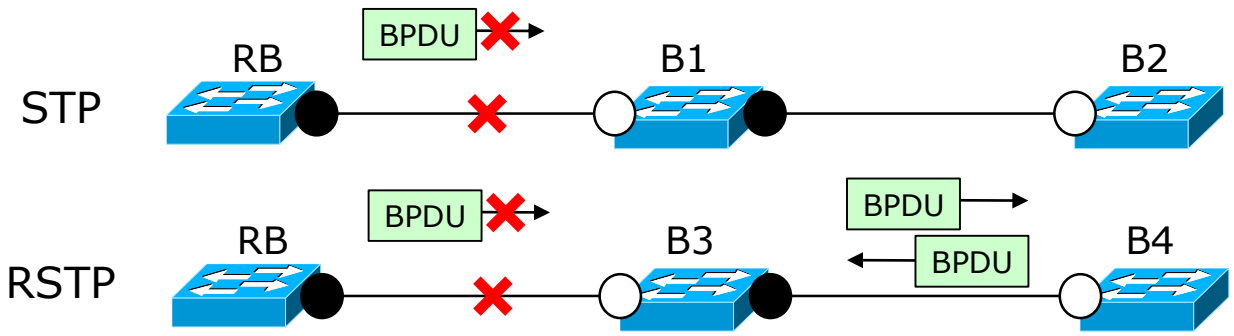
Improvements against STP

- New BPDU Handling
  - 1) BPDU are *sent* (not *relayed*) every Hello-Time
  - 2) Faster aging of information
  - 3) Accept inferior BPDUs

- 4) Rapid Transition to Forwarding State
  - Edge Ports
  - Alternate Ports
  - Backup Ports
  - Proposal / Agreement Sequence in case the “pre-computed” solution (i.e., Alternate/Backup) is not available

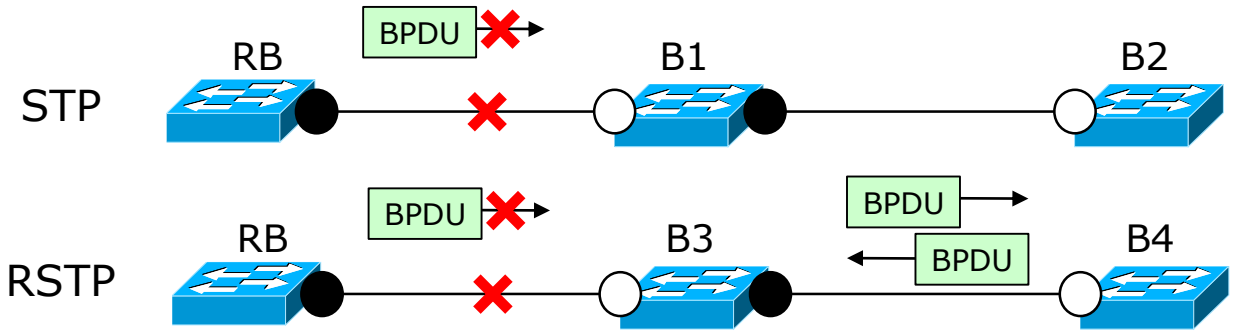
(1) BPDU sent every Hello Time

- BPDU are **sent** every Hello Time on **all active ports**
  - In STP, non-root bridges simply **relay** BPDUs when receive them from the root port
    - If the root bridge dies, nobody generates BPDU till MaxAge expires
  - In RSTP, a bridge always generates its BPDU every Hello Time, even if it does not receive the corresponding BPDU from the root
- BPDU are used as "keep-alive" between bridges
- Hello Time: default 2 sec



(2) Faster Aging of Information (1)

- If BPDU is not received for 3 consecutive times, the current BPDU of the root bridge is declared obsolete
  - No need to wait for MaxAge, although still valid parameter
  - Quick failure detection
- Corollary
  - If a bridge does not receive BPDUs from a neighbor, it can be sure that there is a fault on the link to the neighbor
  - In STP, the problem might have been anywhere from that bridge to the root



(2) Faster Aging of Information (2)

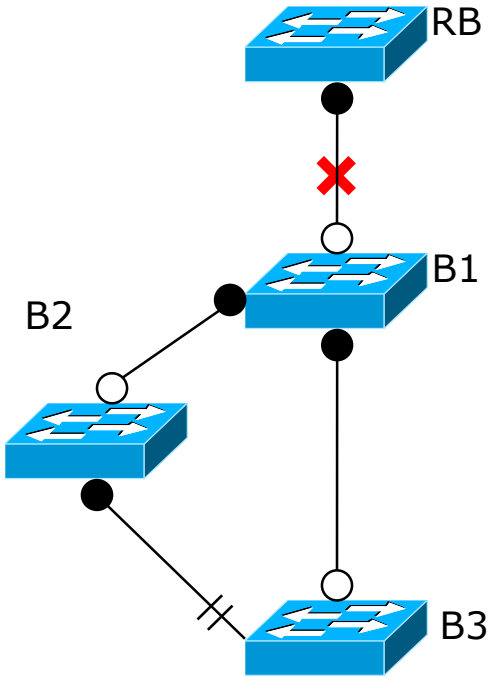
- Not very useful on modern networks
  - Most of current networks are pure switched (and full-duplex) networks
- In that case, convergence in on the order of some ms
  - It exploits the signal coming from the physical layer (e.g. “link down”)
- Faster aging has been defined to improve the convergence in older networks

(3) Accepts Inferior BPDUs (1)

- If a bridge receives *inferior* BPDU from its Root port, it accepts the new BPDU immediately and replaces the one previously stored
  - The current bridge accepts that information without having to wait for MaxAge
    - The new BPDU will replace the one previously stored
    - Much quicker convergence
  - Rational: if somebody closer to the rood bridge changed its parameters and moved to a worse path, there should be a reason (see next slide)
- Inferior BPDU
  - BPDU with a worse RootID *(or)*
  - BPDU with a worse root path cost

(3) Accepts Inferior BPDUs (2)

- Rational
  - If a bridge receives inferior BPDU from its Root port (i.e., from its Designated or Root Bridge), it means that something bad happened on its path toward the root bridge
    - E.g., a link failure that increased the root path cost
    - E.g., a loss of connectivity toward the old root bridge, and another bridge elected itself as root
  - Being received on its Root port, the information is trustworthy



(3) Accepts Inferior BPDUs (4)

- Accepting an inferior BPDU does not mean that the new (inferior) information will become the one used by the bridge
  - The new info is “accepted” and this triggers a new re-computation of the RSTP without having to wait for the timeout
    - In STP, we had to wait MaxAge in order to accept the new RootID
  - E.g., if the new RootID received is worse than the CurrentBridgeID, that bridge:
    - (1) understands that the past root bridge may no longer available
    - (2) starts re-computing the RSTP
    - (3) since it detects that it will be a better root bridge than the one received, it will start propagating itself as root

(4) Rapid Transition to Forwarding State (1)

- Edge ports connect end users and are not part of the ST topology
  - Edge ports transition immediately in Forwarding State
    - No listening/ learning stage
    - No delayed connectivity for the end user
  - No Topology Change Notifications

- Alternate Ports are immediately promoted Root when the Root port fails
  - If a Root Port fails and no Alternate Ports are available, the bridge start proposing itself as Root bridge
    - Rational: apparently, the bridge is no longer connected to the root

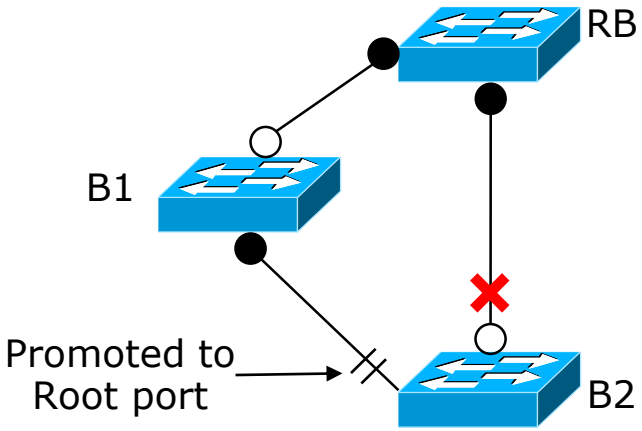
(4) Rapid Transition to Forwarding State (2)

- Backup Ports: immediately promoted Designated if the Designated fails

- Proposal/Agreement Sequence used when a new link goes up
  - Allows deciding which end of the point-to-point link has to become Designated Port
    - Those ports are moved up in a very short time
  - Available only on Full Duplex links
    - Full-duplex links are always point-to-point
  - A port in Half Duplex mode is considered as “shared port” by default

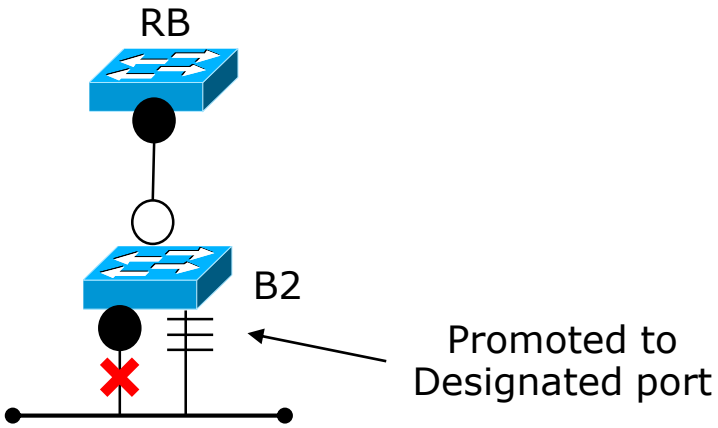
Alternate ports promoted Root

- Acts when a fault is detected on the Root port
- If an Alternate port exist on the same bridge, it is promoted to Root
  - In case more than one Alternate exist, the best one is promoted
  - This path will become the new path toward the root bridge



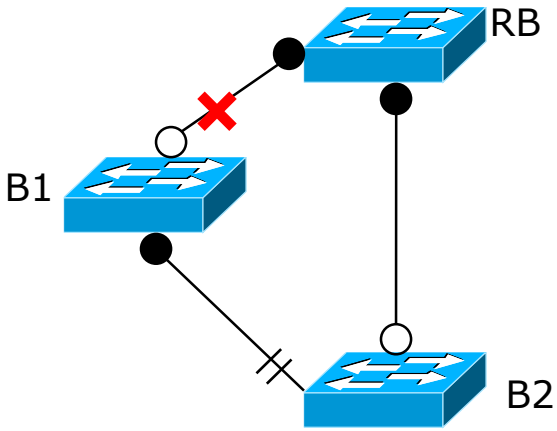
Backup ports promoted Designated

- Acts when a fault is detected on the Designated port
- If a Backup port exist on the same bridge, it is promoted to Designated
  - This path will become the new path from the link toward the root bridge



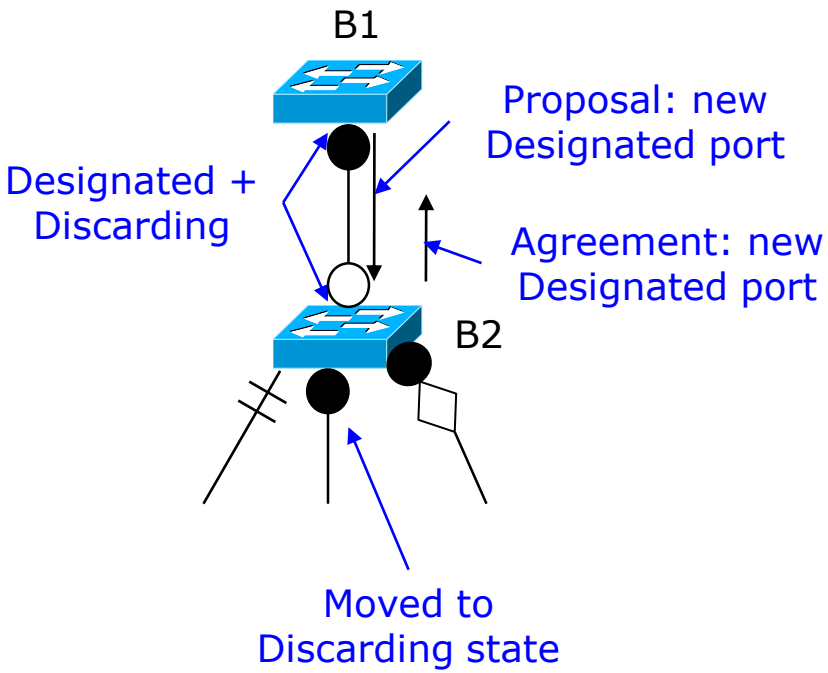
Other cases

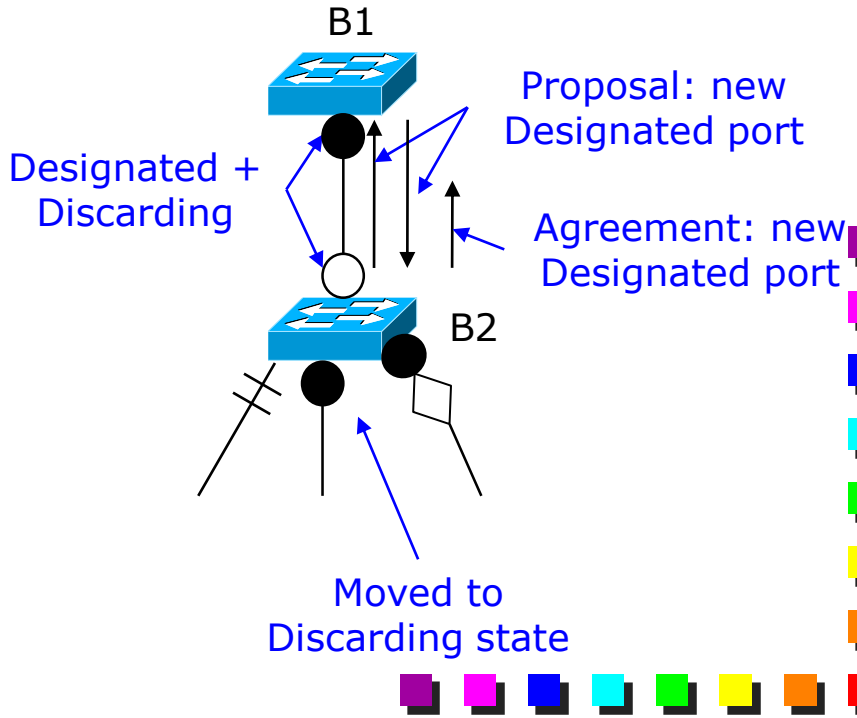
- No alternate port exist: the Bridge promotes itself as Root
  - E.g., B1 temporary promotes itself as root
  - Obviously, in this case all the ports go Designated
  - Still rather fast, but not as fast as the previous case
  - It requires the bridge to start a Proposal/Agreement sequence (see later)
- New link up
  - Port promoted as Designated and start a Proposal/Agreement sequence (see later)



Proposal/Agreement Sequence (1)

- Algorithm for fast synchronization on the port role between two switches
- General idea: we want to re-create the proper tree starting from the root and propagate the new topology down toward the edge





Proposal/Agreement Sequence (2)

- Very fast, and **does not rely on timers**
- If a designated discarding port does not receive an agreement after it sends a proposal, it slowly transitions to the forwarding state, and falls back to the traditional 802.1D listening-learning sequence

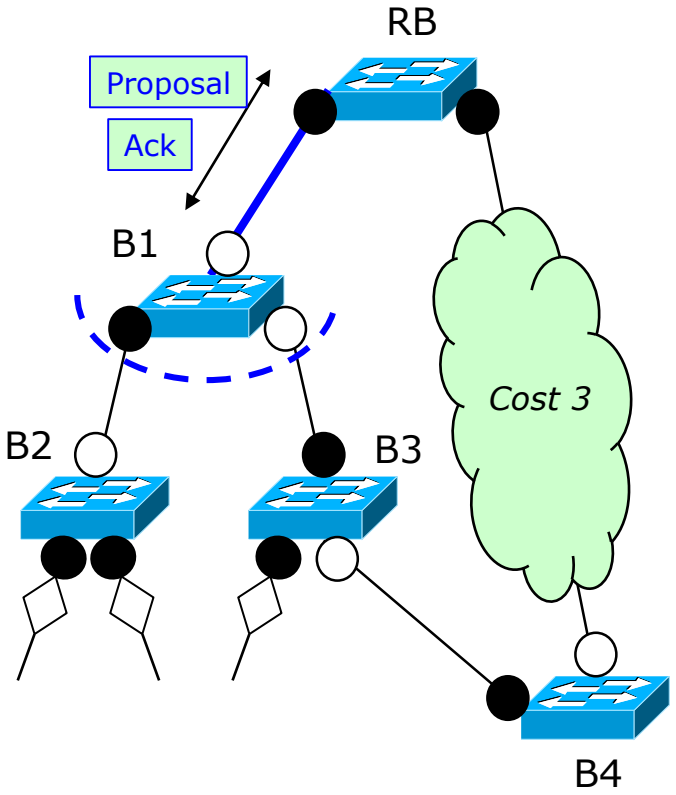
- Algorithm
  - When a bridge comes up it puts its ports in a Designated Discarding state
  - A port in Designated Discarding (or Designated Learning) sends a new BPDU to the other party proposing itself as Designated

Proposal/Agreement Sequence (3)

- Algorithm (cont)
  - If the bridge **accepts** the proposal, it will sync its ports
    - The SYNC process aims to verify that all of its ports are in-sync with this new information, and that no loops can occur
    - The SYNC process occurs only if the bridge detects that the incoming BPDU contains better information
    - The SYNC process will block all the active ports
      - Edge, Alternate and Backup ports are kept unchanged
      - Root and Designated ports are moved into Discarding state
  - The bridge will acknowledge the Proposal BPDU and configures its port in the appropriate state
    - Root, Alternate, Backup
  - If the bridge does not accept the proposal, it replies with its own BPDU with the Proposal bit set
    - The rest is the same

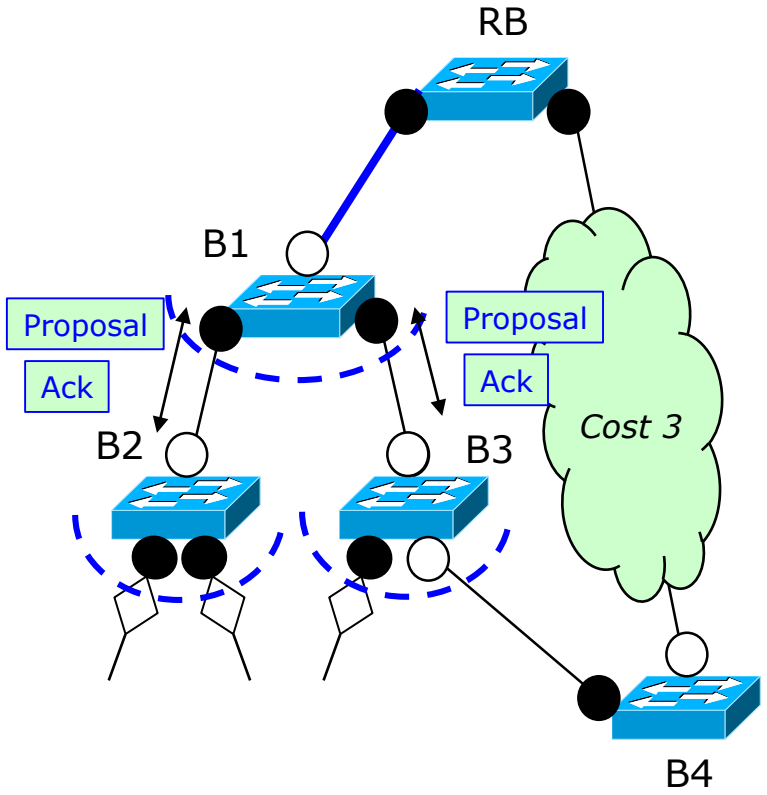
Convergence with RSTP: example (1)

- The Root Bridge adds a new link toward B1
  - Both ports put in Designated Discarding
  - BPDU + Proposal flag sent on the new link
  - Sync on the lower ports of B1
  - Ack from B1 allows RB to move its port in Forwarding State
  - B1 puts that link in forwarding state too
  - No loops can occur, since downstream ports of bridge A are still blocked



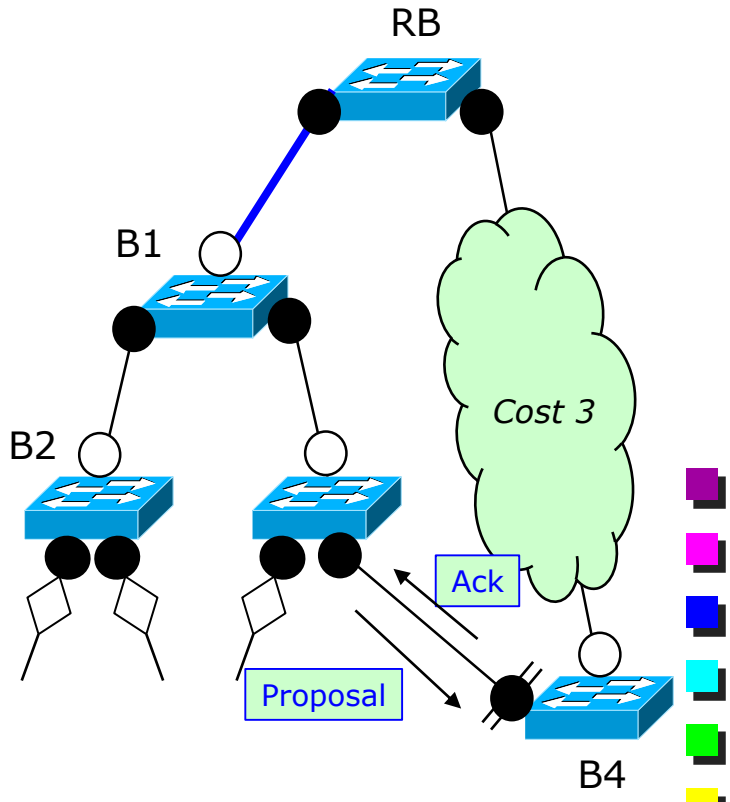
Convergence with RSTP: example (2)

- The process is repeated on B1-B2 and B1-B3
  - B1 will send a BPDU with the Proposal bit
    - Please note that upstream ports of B2 and B3 were still keeping their role until the BPDU is received
  - B2 and B3 put the other ports in SYNC state
    - Note that B2 does not have to block any port, while B3 will block only the link toward B4
  - They will accept the proposal
  - Port roles on B1-B2 will stay unchanged, while they will change on B1-B3



Convergence with RSTP: example (3)

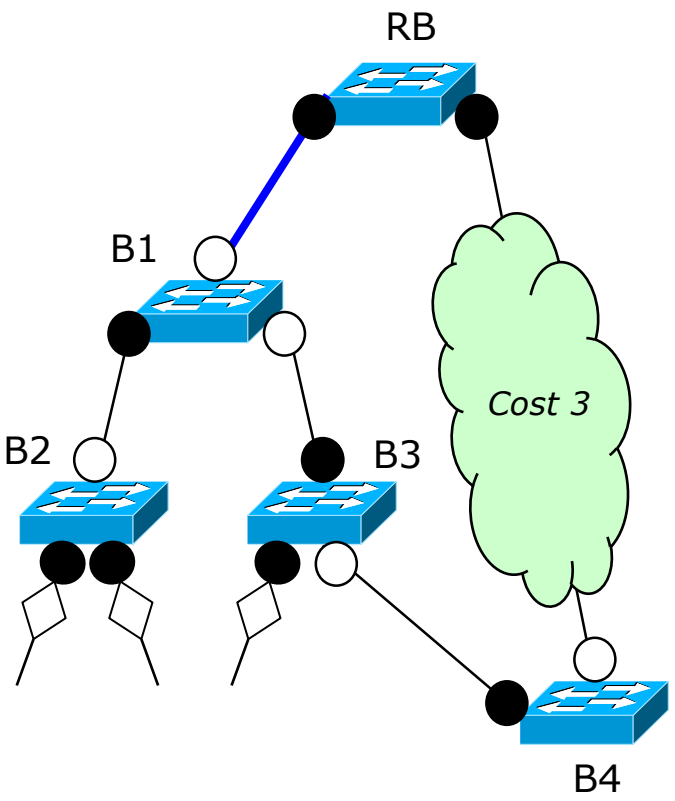
- … and now the synchronization reaches B4
  - B3 will send a BPDU with the Proposal bit
  - However, B4 has a better path toward the root bridge[1]
  - So, it sends a BPDU with the Proposal bit back
    - No SYNC occurs on B4, since its ports do no change role
  - B3 will ack the proposal
  - B3 will set its port as Alternate



[1]*The path of B4 on the right is better than the one on the left, because costs are the same, but the remote bridge ID is better.*

The same example with STP

- RB adds a new link toward B1
  - Link goes into Listening (no data, no loops)
  - BPDU generated by root propagated down to B1, B2, B3
  - B1, B2, B3 update their STP Topology (e.g. Root Port)
    - B1, B2 and B3 were previously reachable through B4
    - All links involved still in blocking
    - B3 will move its lower port in Blocking state
  - B1, B2, B3 and their leaves unreachable for 2 * ForwardDelay
    - Link RB-B1 in Forwarding state after 2* Forward Delay



Note: the STP topology refers to the one before adding the link

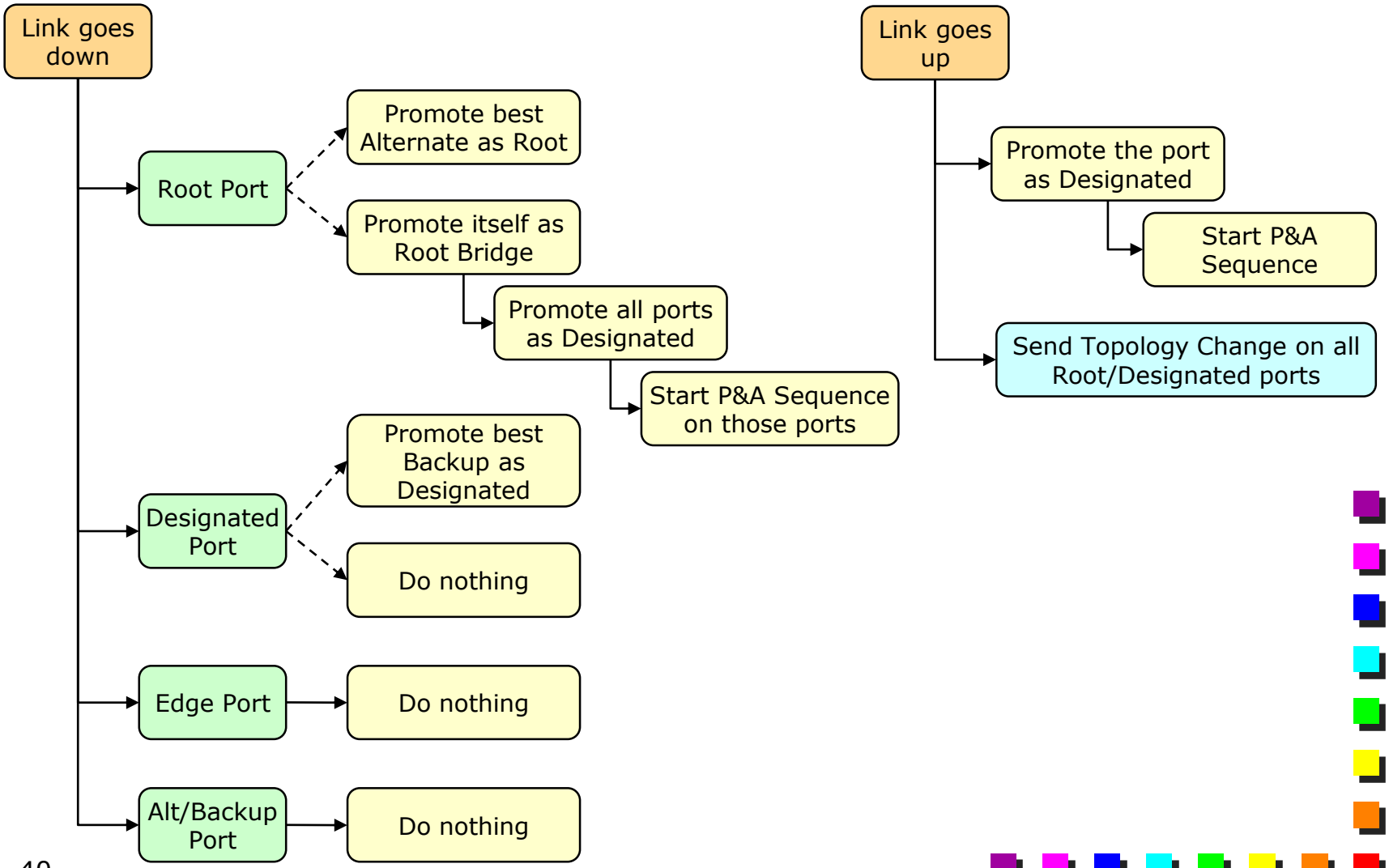
Convergence with RSTP: some notes (1)

- An alternate path may be created in 10 – 20 ms
- Fast convergence relies upon:
  - Ability to detect a failure in an reliable way
  - Ability to quickly detect a failure
    - Recovery based physical layer
- For this purpose the physical layer used is definitely relevant
  - High stability due to reliable hardware parts
    - Intermittent failures may create stability issues
  - Transceiver able to locate a local or remote failure over point-to-point links

Convergence with RSTP: some notes (2)

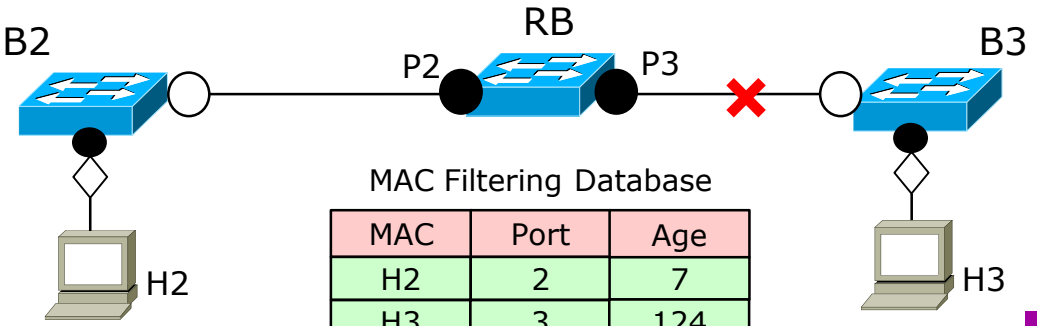
- The explicit authorization sent in the SYNC process replaces the 2*Forward Delay of STP
- The final topology is exactly the one calculated by the STP
  - I.e., the blocked port will be exactly in the same place as before
  - Only the steps to obtain this topology have changed
- Explicit negotiation is possible only when bridges are connected by point-to-point links
  - I.e., full-duplex links unless explicit port configuration

RTSP Link Events: Recap



Topology Change Detection (1)

- A port that goes down does not generate a Topology Change
  - A loss of connectivity (a port moving to Blocking) is no longer considered a Topology Change
  - The rest of the network may have useless entries in the Filtering Database but this does not represent a problem
    - E.g., MAC addresses associated to hosts that disappeared
  - Obviously the bridge clears the Filtering Database entries associated to that port
- Only *non-edge* ports that move to Forwarding state cause a Topology Change



MAC Filtering Database

MAC	Port	Age
H2	2	7
H3	3	124

Topology Change Detection (2)

- An RTSP bridge that detects a topology change (i.e., a port that goes in Forwarding State):
  - Starts the TC While timer (2*Hello Time) for all its non-edge active ports (i.e., Designated and Roots)
    - It flushes all MAC addresses associated to these ports

    Note 1: it does not flush MAC addresses associated to Edge ports

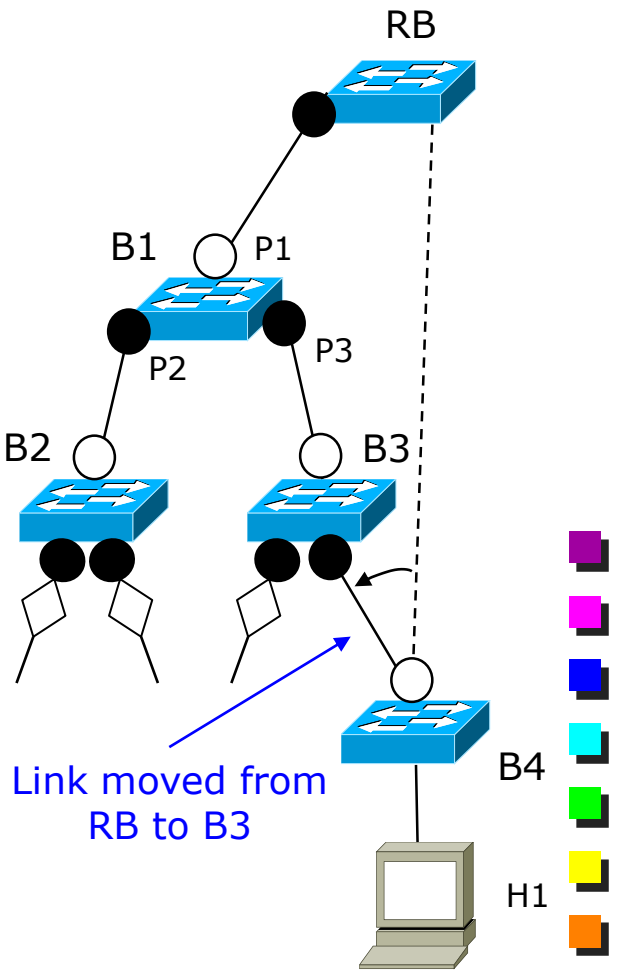
    Note 2: the TC While timer is much shorter than in STP
  - Propagates the BPDU with the TC bit on all the ports where the TC While timer is active, until it expires
    - RSTP no longer uses the specific TCN BPDU, unless a legacy bridge needs to be notified
      - The TCP BPDU, in fact, does no longer exist

Topology Change Propagation (1)

- When a bridge receives a BPDU with the TC bit set:
  - It starts the TC While timer on all its Designated and Root ports but the one on which it was received
  - It sends BPDUs with TC set on all the ports where the TC While timer is active
  - It clears the MAC addresses learned on all the ports where the TC While timer is active
    - I.e., it does not clear the entries on the port it received the TC bit
- TCN floods very quickly across the entire network
  - No longer needed to notify the Root Bridge (such as in the STP) in order to generate BPDUs with the TC set
  - No longer keep the "reduced" filtering database for <MaxAge + ForwardDelay>

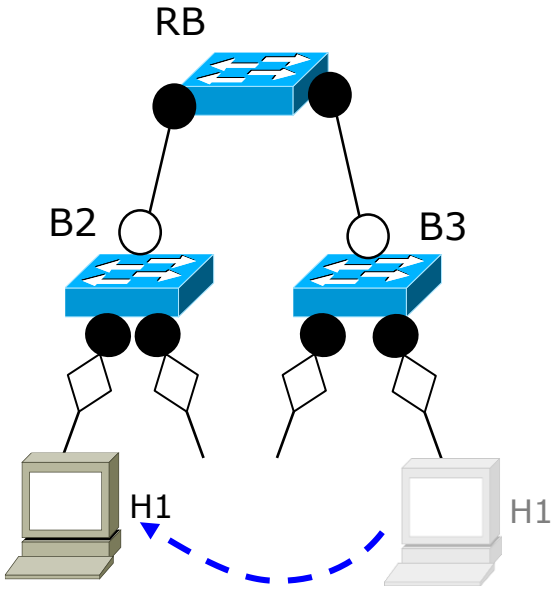
Topology Change Propagation (2)

- A bridge clears the MAC addresses learned on all its ports, **except the one that receives the topology change**
- Rational
  - The TC is generated only *when a port on a remote bridge goes in Forwarding state*
  - This means that we may have *more* MAC addresses associated to the local port compared to the ones we had previously
  - So, we do not need to clear the entries already associated to that port, which will be still reachable on that port in the future
  - The problems is on the other local ports … MAC addresses associated to them may no longer be valid



Topology Change Propagation and Edge ports

- TCN is not generated when an Edge port goes up
  - E.g., host H1 moving from bridge B3 to B2
- Host H1 is required to send a broadcast frame to update the Filtering Database of the switches



RSTP: compatibility with STP (1)

- RSTP bridges may be configured to operate in STP mode
  - Vital if there is a repeater between bridges
- RSTP bridges switch automatically in STP mode when they detect one or more bridges operating in 802.1D
  - BPDUs received with the protocol version identifier field set to 0 are handled in a different way
- Migration delay timer (3 sec) starts when a port comes up
  - During this time, the current STP or RSTP mode associated to the port is locked
  - When the migration delay expires, the port adapts to the mode that corresponds **to the next BPDU it receives**
  - If the port changes its mode of operation as a result of a BPDU received, the migration delay restarts

Problems when mixing STP and RSTP (1)

- Problem 1: fast convergence is lost when operating in 802.1D
  - Even if we have all Full Duplex links
- Problem 2: an 802.1w port that starts operating in 802.1D compatibility mode may not be able to not turn back in 802.1w unless an explicit configuration is taken
  - In 802.1D BPDUs flows from the root to the edge
  - Even if the 802.1D edge goes down, there is no way to inform the upstream bridge that it should revert back in 802.1w
    - *Look at the examples in the next slides*

Problems when mixing STP and RSTP (2)

- Example 1
  - B1 has the Designated Port over the LAN
  - B1 receives a 802.1D BPDU from B2
    - Please note that B3 advertises itself as Designated Port, since it cannot understand the 802.1w BPDU generated by B1
  - B1 changes in compatibility mode and starts generating 802.1D BPDUs
  - B1 has the best BridgeID, hence B2 will put its port in Blocking State
  - Since B2 will not generate BPDUs on its port, B1 will stay in 802.1D mode forever since it has no way to detect the (possible) death of B2
    - Manual intervention is required to restart B1 (so that it will run in 802.1w mode) in this case

Problems when mixing STP and RSTP (3)

- Example 2
  - B2 has its port in Blocking state
    - B1 is the Designated Bridge for the LAN
  - The Designated Bridge has to be re-elected when B1 dies
  - B2 brings its port to up and waits for the Migration Timer
  - No other BPDUs are received, so it will generate 802.1w BPDUs

Problems when mixing STP and RSTP (4)

- Problem 3: network instability (e.g., loops)
  - A switch may send 802.1D BPDUs on some ports, and 802.1w BPDUs on some other ports
    - An RTSP bridge sends 802.1D BPDUs only on the port a 802.1D BPDU was received
  - The RTSP portion of the network will enable the forwarding of data frames in a matter of seconds (usually < 3 seconds)
  - The STP portion of the network is still well away from converging
    - Possible packets duplication
    - Possible receipt of *out-of-sequence* packets
    - Some (transient) loops can occur
- Be careful when using protocols that assume a L2 "traditional" connectivity!
  - They cannot manage out of order and duplicated packets
  - Better to disable RTSP mode

Default Path Costs

- 802.1w path costs are the ones already defined in 802.1t

Port speed	Recommended Value	Recommended range values
<= 100Kb/s	200.000.000	20.000.000 – 200.000.000
1 Mb/s	20.000.000	2.000.000 – 20.000.000
10 Mb/s	2.000.000	200.000 – 2.000.000
100 Mb/s	200.000	20.000 – 200.000
1 Gb/s	20000	2.000 – 20.000
10 Gb/s	2000	200 – 20000
100 Gb/s	200	20-2000
1 Tb/s	20	2-200
10Tb/s	2	1-20

Problems due to the fast RSTP reactivity

- RTSP works perfectly when the physical layer is reliable
- When this condition does not hold, some problems may appear
- Example
  - A link goes up and down frequently because of a dirty connector
  - RTSP reconfigures the network at each change of status of the link
  - The network will stay in a transient state most of the time
- Possible solution: enable a proprietary “antiflapping” mechanism on the link
  - E.g. Cisco put ports in “error disable” state when detects link flapping
  - The port must be re-enabled manually from management
- Lesson learned: fast reactivity is not always a good thing

Conclusions

- Efficient
- Fast convergence (often < 1s)
- Can replace many proprietary protocols
- Possibility to mix devices from different vendors
- Interoperability problems with STP
  - Mostly used with Multiple Spanning Tree (MST)