Spanning Tree Protocol

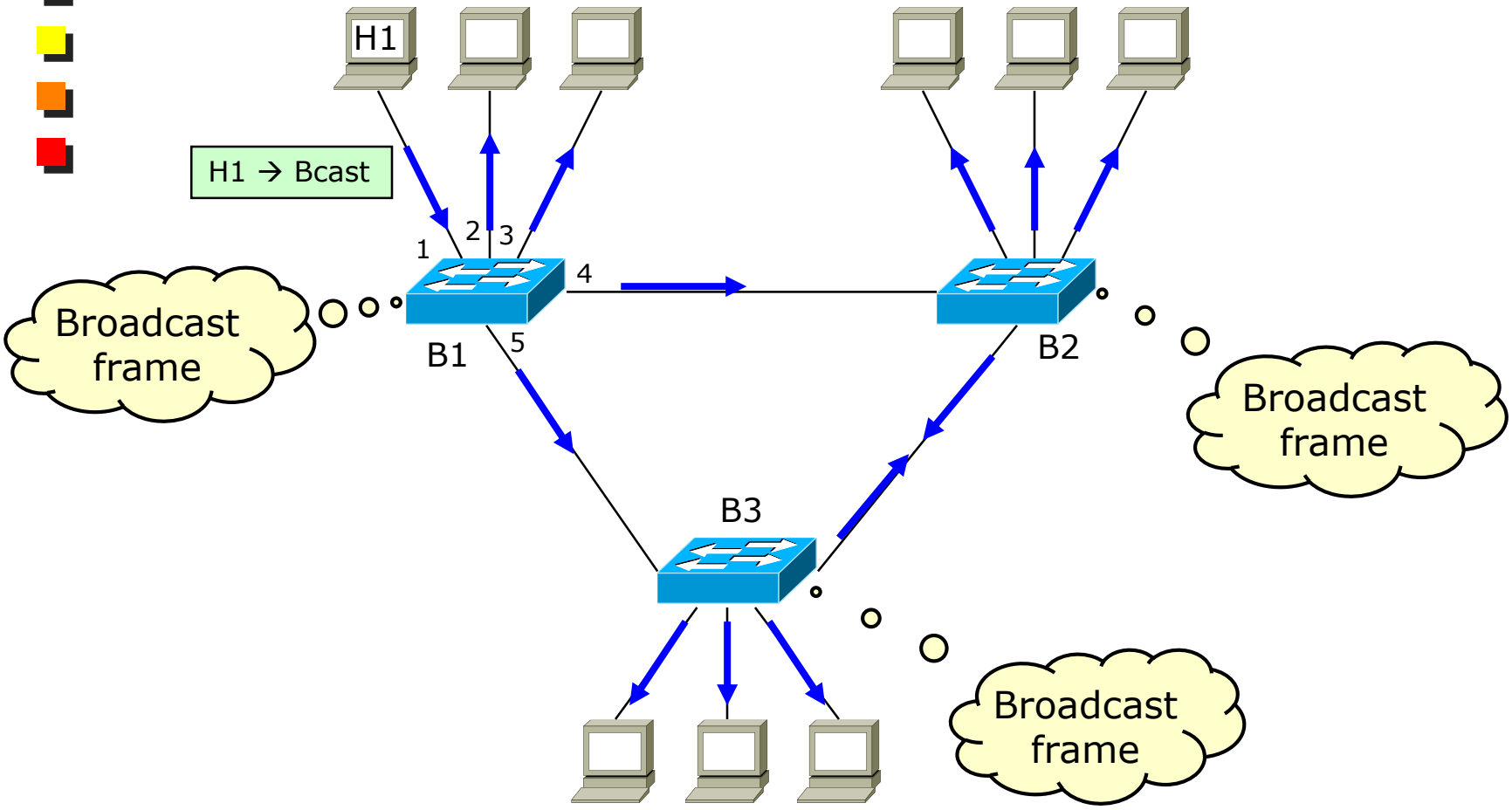
Fulvio Risso

Politecnico di Torino

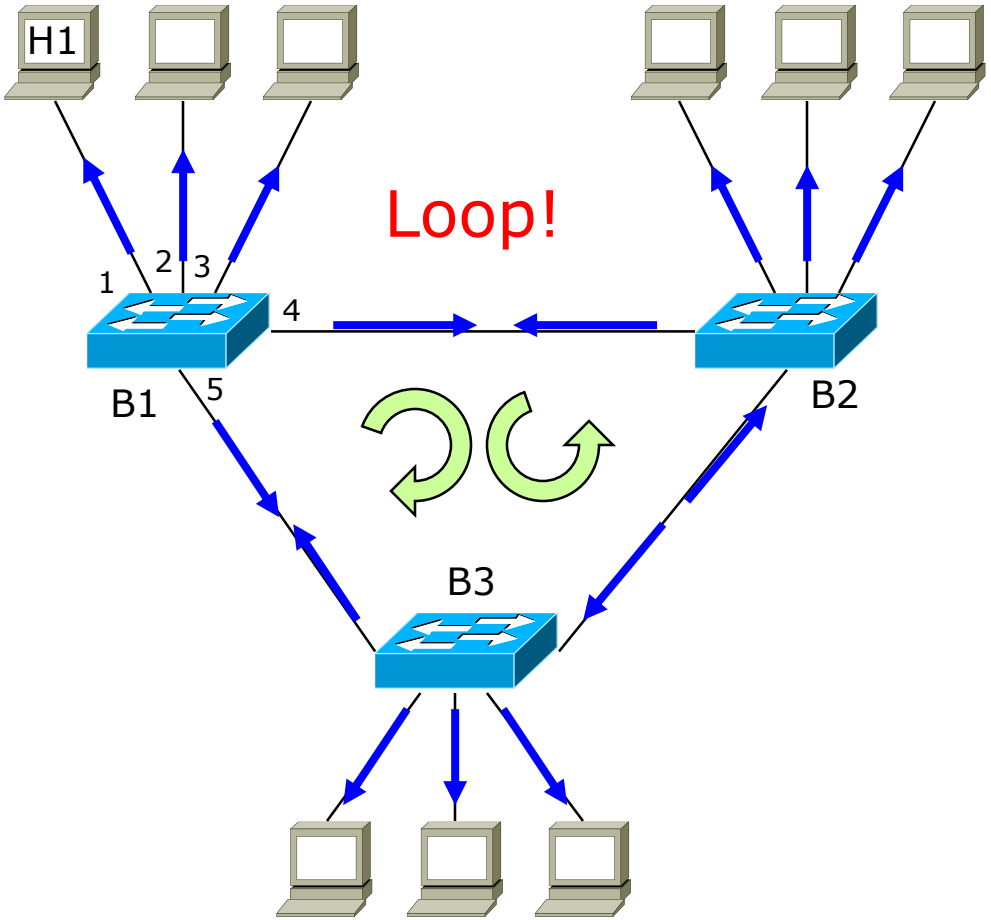
Bridges and meshes

- Two problems
  - Frames can enter in a loop
  - Backward learning no longer able to operate
- It’s now the time to present the third component (i.e. “Spanning Tree”) after the ones we presented earlier
  - “Filtering Database” and “Backward Learning”

Bridges and meshes: the loop problem



Bridges and meshes: the loop problem



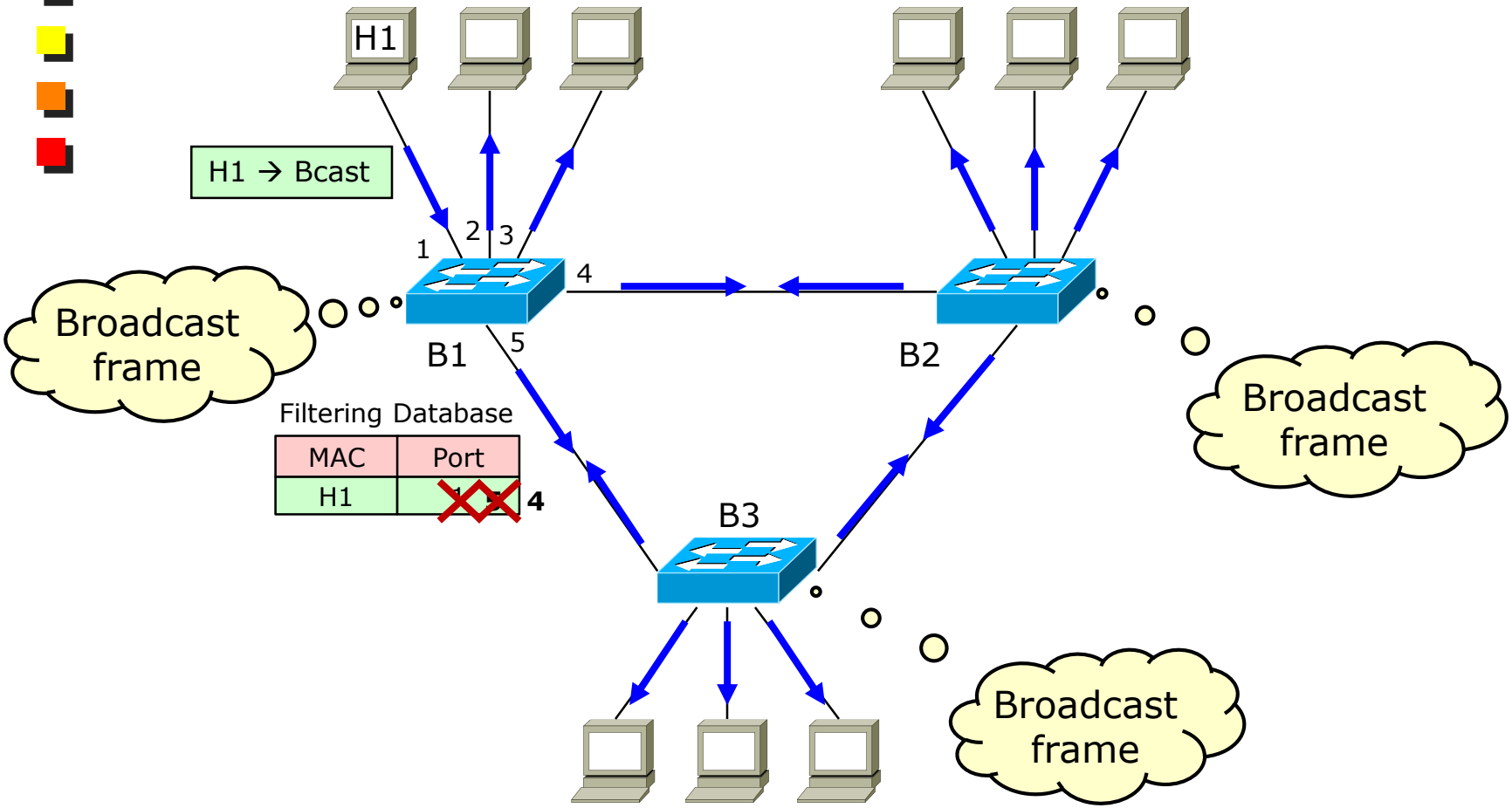
Which frames can generate a loop?

- Multicast/broadcast frames
  - Very common
- Frame to a non-existing station
  - MAC address not present in the filtering DB (e.g. non existing station)
  - Problem that may happen rarely (unless under attack)
    - IP sends an ARP before contacting an L2 station
    - If the station does not exist, the ARP will never get a reply and the destination MAC address is unknown
    - Therefore, no MAC frames will be sent to that station intentionally

The Broadcast Storm

- Massive load due to broadcast/multicast traffic on a LAN
- One of the most dangerous problems at data-link layer
- No solutions, except for disabling (physically) loops
  - E.g., detach a cable from a bridge
- Network operators are almost impotent in such this case
- Due to the lack of a “time-to-live” field in L2 frames
- L3 networks can tolerate transient loops
  - TTL available on L3 packets
- Can be used to create a low-cost traffic generator sending frames at line-rate

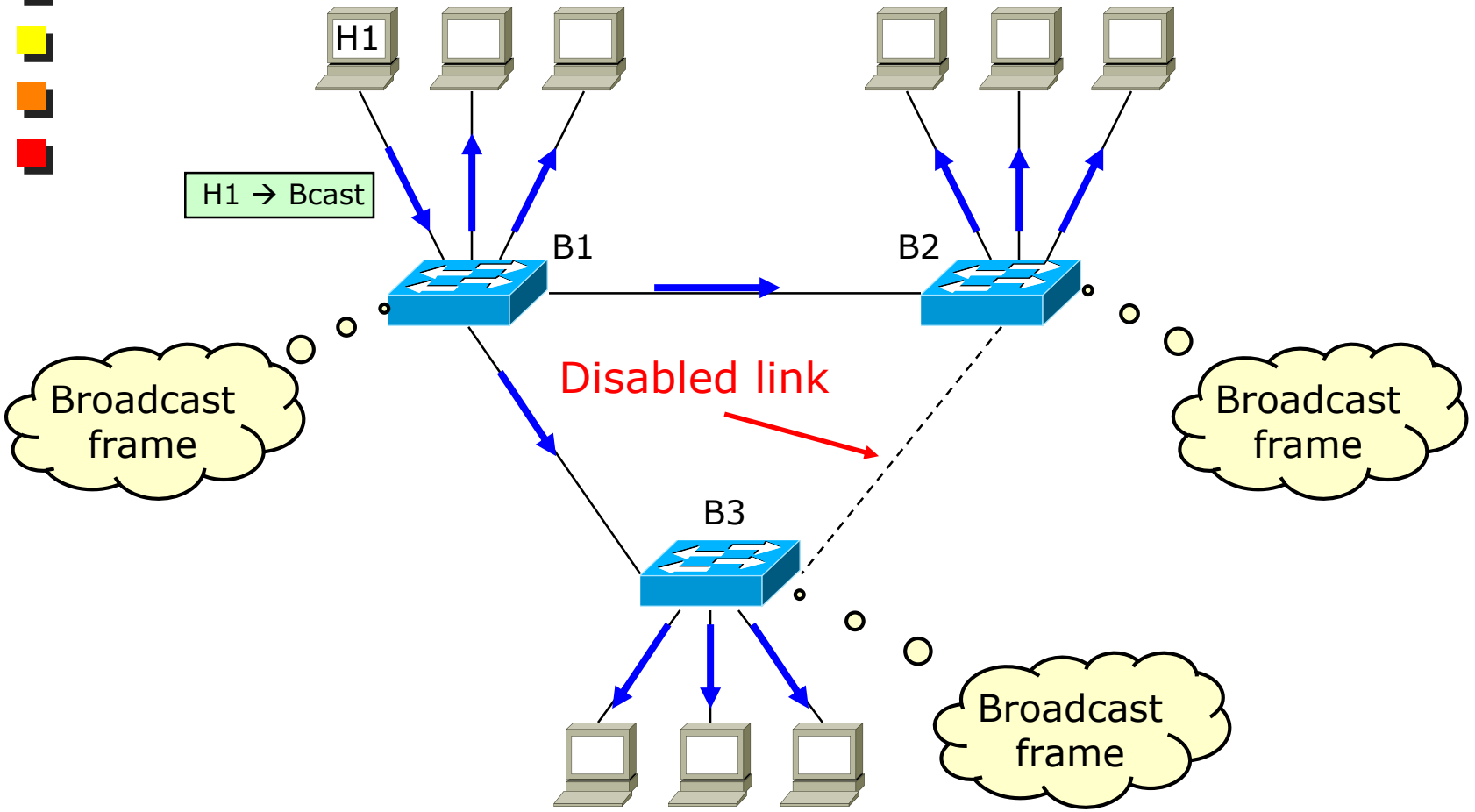
Bridges and meshes: the learning problem (1)



Bridges and meshes: the learning problem (2)

- Backward learning problem
  - Switches may have inconsistent filtering database
  - An entry in the filtering database may change the port indefinitely
    - An entry may not able to reach a stable state
    - Transient loops can be created among back-to-back bridges
      - B1 forwards to B2 that forwards to B1,…
      - Larger (B1-B2-B3-B1) loops may occur as well

The Spanning Tree idea: no loops in the network



Spanning Tree

- In order to avoid troubles, you must avoid loops in the physical network
  - Either create loop-free networks
    - Discouraged; not robust
  - Or define al algorithm that disables (temporarily) loops
- 802.1D
  - Original idea from Radia Perlman, PhD @DEC
- Meshes detected and disabled; the network becomes a tree
  - Unique path between any source and any destination
- Operates periodically (every second)
  - Decides which port set to forwarding state and which port set to blocking state

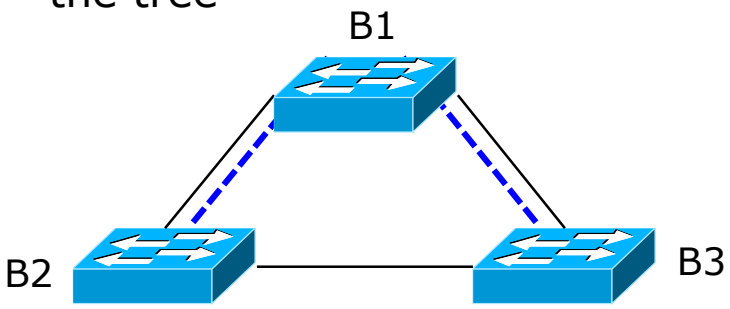
For a better comprehension… a nice poem

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me.
Then bridges find a spanning tree.

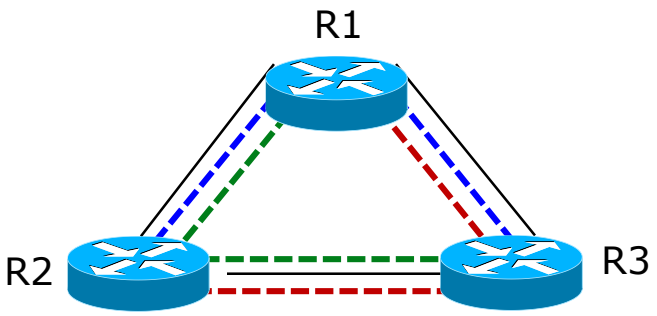
[Radia Perlman]

Spanning tree and paths

- Single tree for the entire network
  - Root Bridge is the root of the tree
  - Unique path from any source to any destination
  - Although also L3 routing uses the "spanning tree" idea…
    - Here the tree is unique in the entire network
    - At L3, each source calculates its own tree
- Paths are not optimized
  - In general, paths are optimized only with respect to the root of the tree



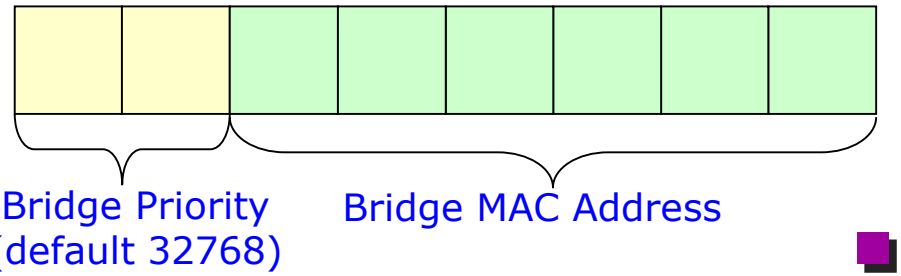
Single spanning tree



Multiple spanning trees

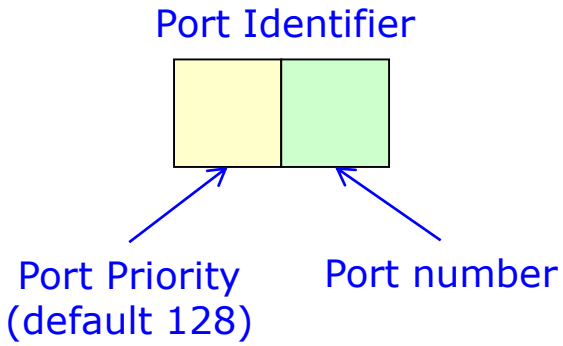
Parameters of the Spanning Tree Algorithm (1)

- Bridge Identifier
  - In general, each port of the bridge has its own MAC address
  - One of them is chosen to create the Bridge Identifier
    - Vendor-specific algorithm
  - A priority is assigned to each bridge
    - A default value (32768) guarantees "plug&play" operation
  - 8 bytes (2 for Priority and 6 for the MAC address)
    - New 802.1t will reserve only the first 4 bits to priority
    - I.e. better to set the priority in 4096 multiples
  - The MAC address cannot be modified, while the Priority can be set by management



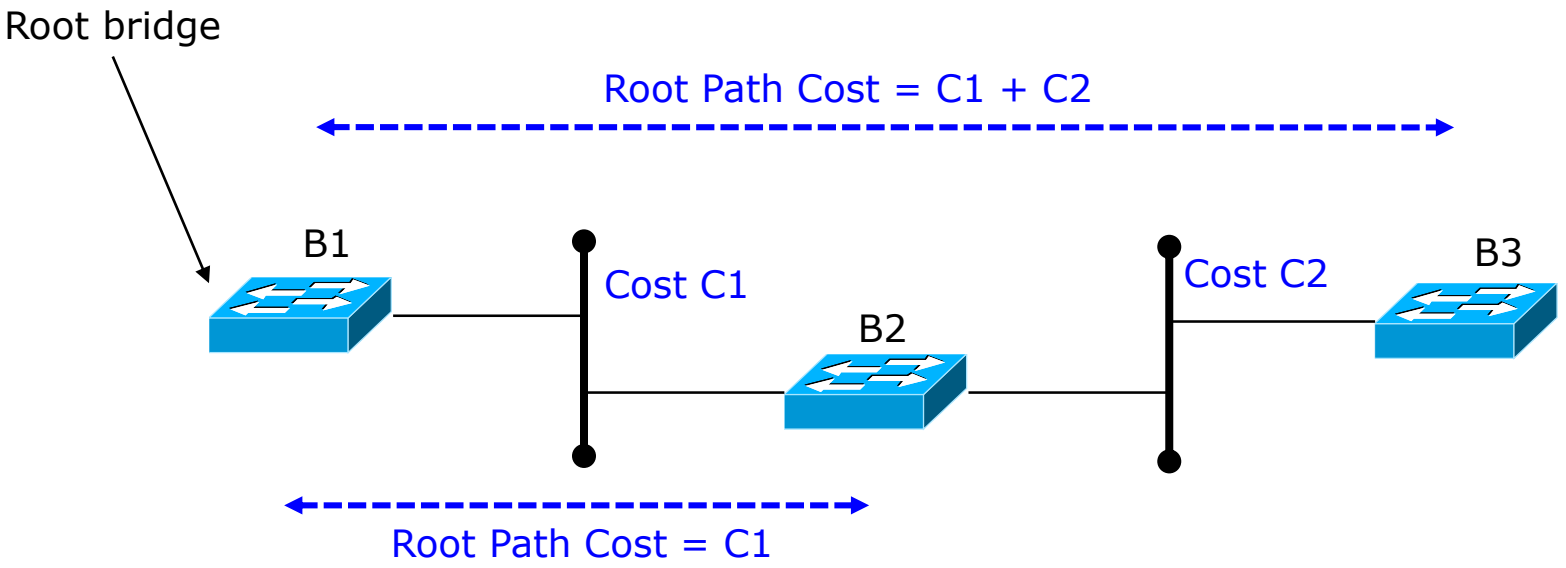
Parameters of the Spanning Tree Algorithm (2)

- Port Identifier
  - Identifier associated to each port
    - Port priority (1 byte)
      - Default: 128
    - Port number (1 byte)
  - In theory, no more than 256 ports per bridge
    - In practice, we can use also the Port Priority field if needed



Parameters of the Spanning Tree Algorithm (3)

- Root path cost
  - Cost for reaching the root bridge
  - Sum of the costs of all the links traversed

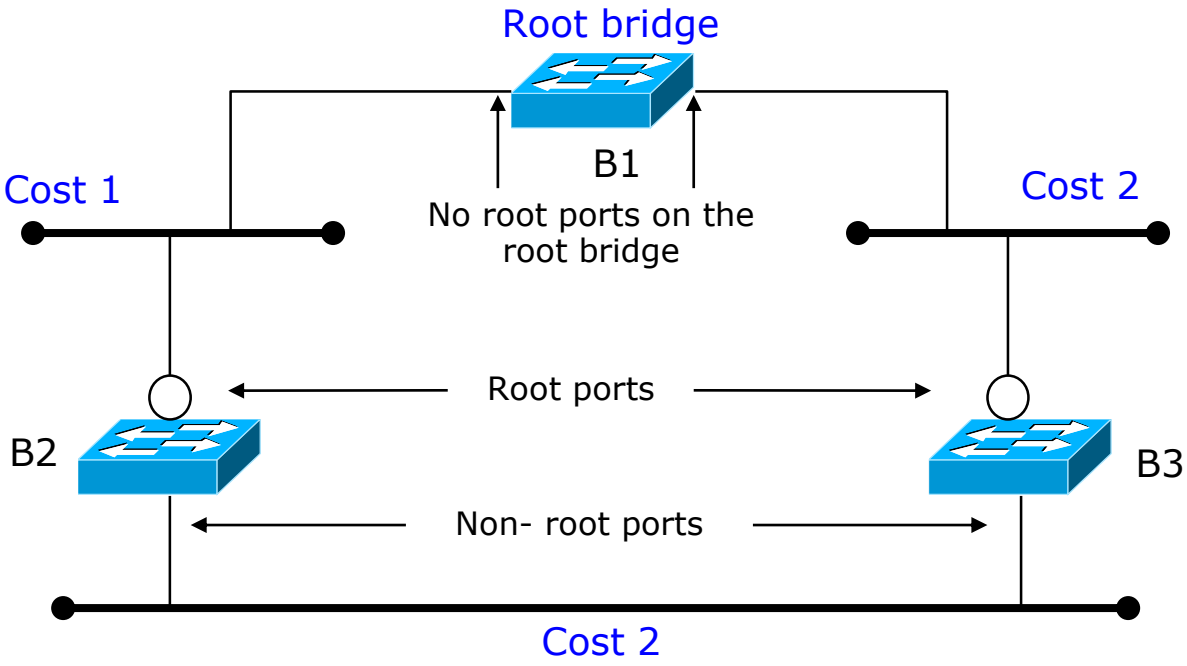


Spanning tree and ports (1)

- The algorithm ends up having ports on each bridge labeled as
  - Root Port
  - Designated Port
  - Blocked Port

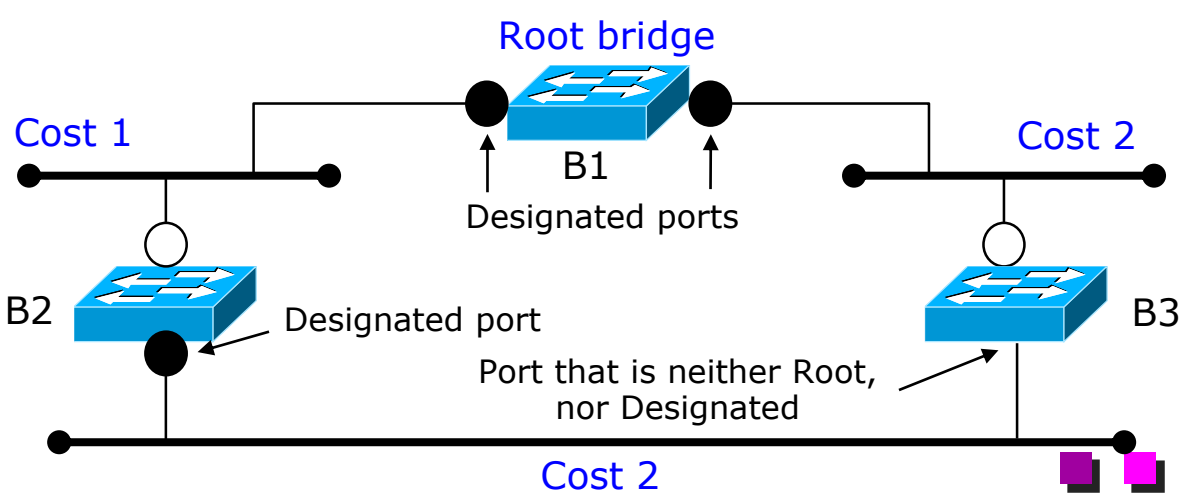
Spanning tree and ports (2)

- Root Port
  - Port of the bridge with the best path toward the root bridge
  - Each bridge can have only one Root port
    - Except for the root bridge, which does not have any root port



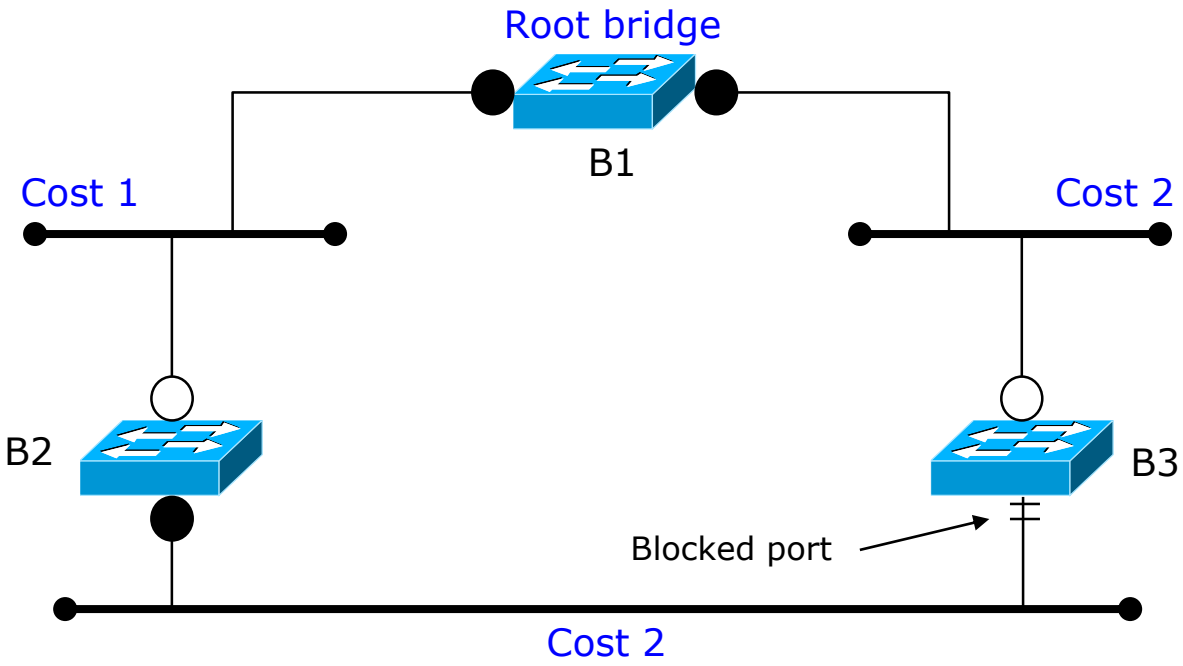
Spanning tree and ports (3)

- Designated Port
  - For any given link, the unlabelled port that has the best cost toward the rood bridge
    - Unlabelled → remaining ports, i.e., that are not root
  - Each link must have *one and only one* Designated port
    - Having multiple Designated Ports will lead to circular paths
  - Each bridge may have multiple Designated Ports
    - Usually, all the ports of a root bridge are Designated



Spanning tree and ports (4)

- Blocked Port
  - Ports that are neither Root, nor Designated



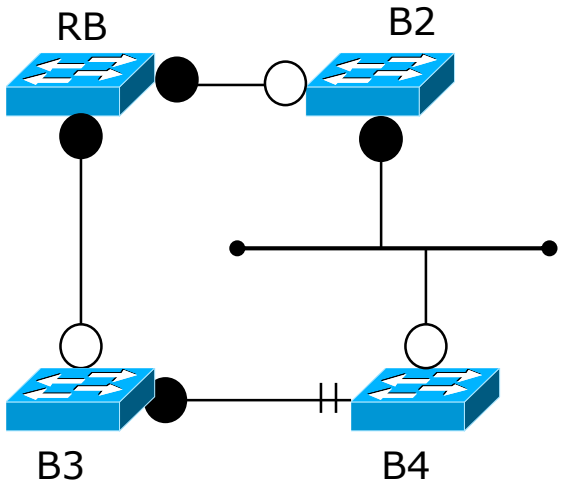
Spanning tree and ports (5)

- The root bridge has
  - Usually, all designated ports
  - May have some blocked ports
- The other bridges
  - One port is always the root port
  - The other ports will be either Designated or Blocked
- All the ports must have a status associated to them!

- Each link has:
  - One Designated Port
  - All the other ports are either Root or Blocked

Spanning tree and ports (6)

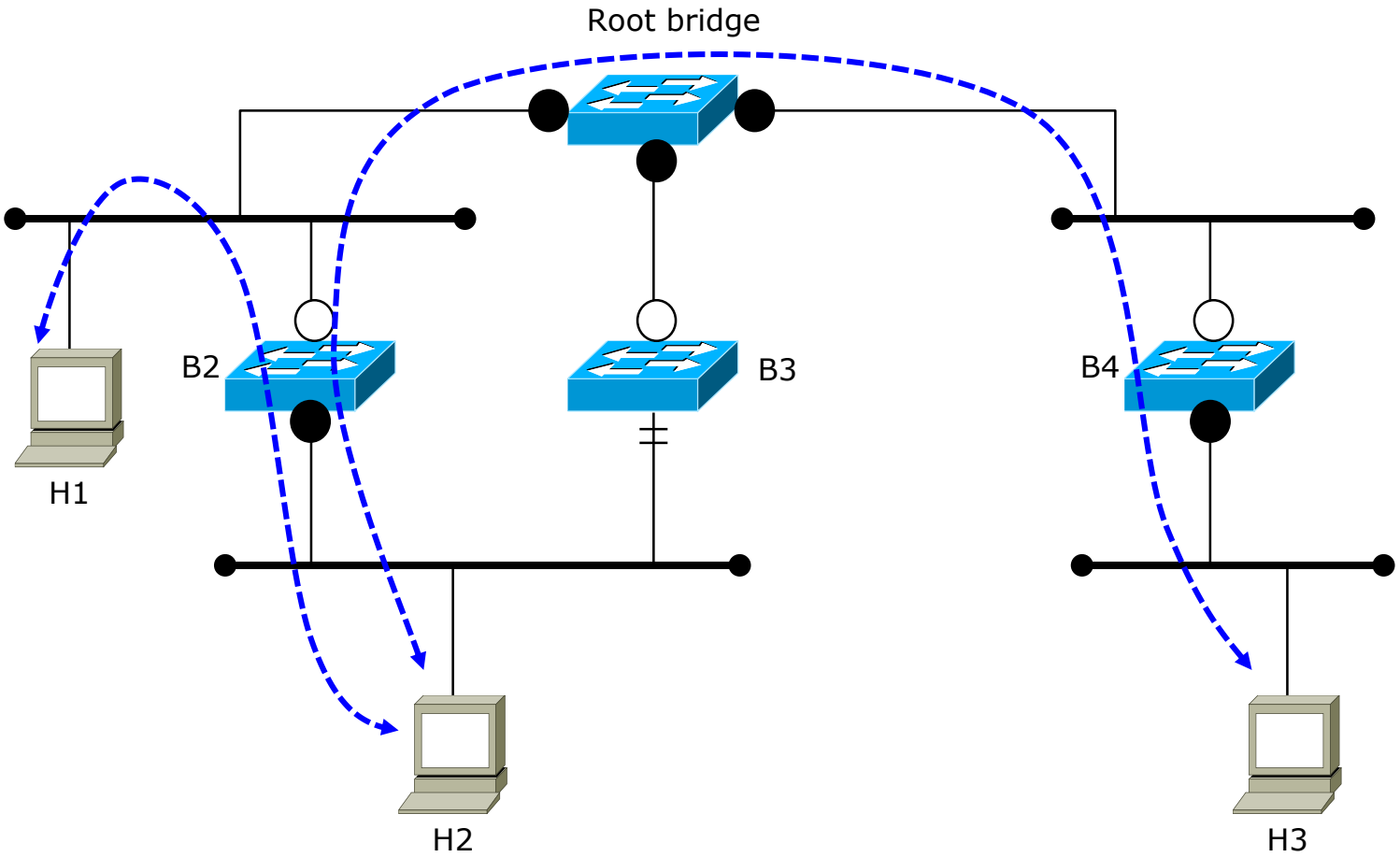
- On a point-to-point link we may have one designated and one blocking port
- Question: Is this useful?
- Answer: how can we guarantee that the link is a point to point link?



Ports and data traffic (1)

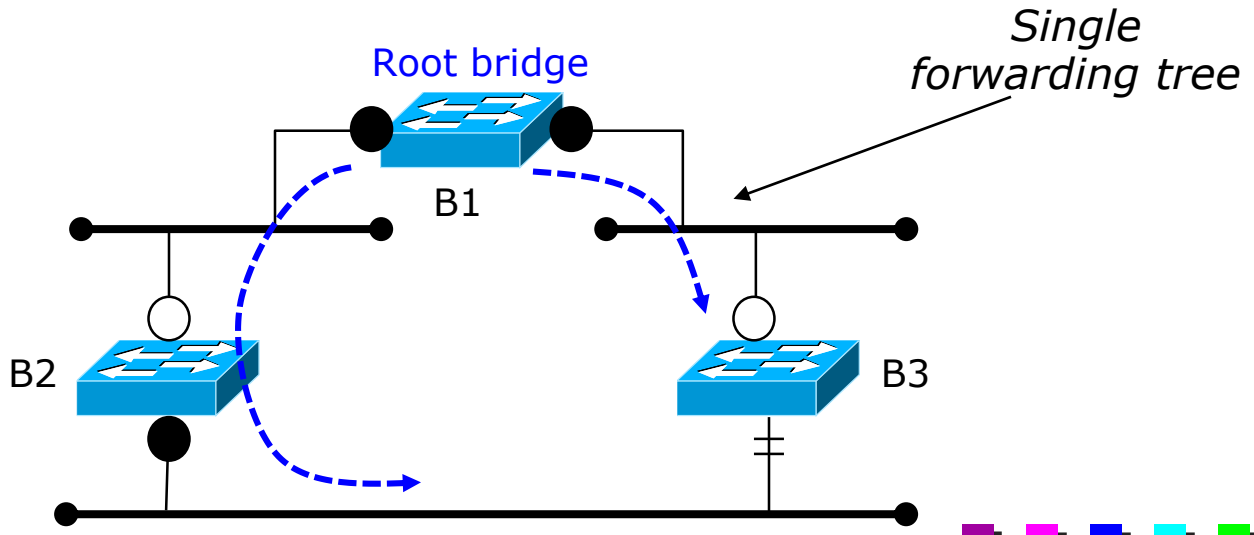
- Active ports (Designated and Root) are the ones in charge of sending/receiving data traffic
  - Designated Port
    - Port that connects the link to the root bridge
    - In charge of serving the link (and, possibly) the portion of the network that is reachable through that link
  - Root port
    - Port that connects the bridge to the root
  - Bridges forward traffic (up, toward the root bridge, or down, toward the leaves) based on the content of the filtering database, only through active ports
- Blocked ports never send data on their link
  - And discard all the data received (except for BPDU configuration messages; see later)

Ports and data traffic (2)



Spanning Tree Algorithm

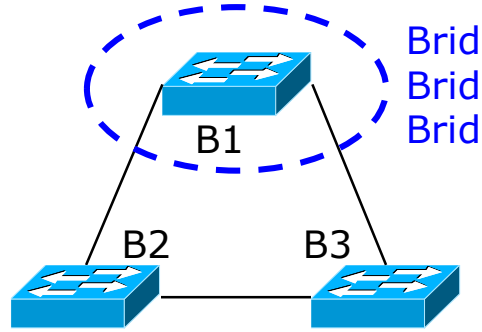
- 1) Root Bridge election (smaller bridge-ID)
- 2) Selection of the Root Port for each bridge
  - The port with smaller cost to the Root Bridge
- 3) Selection of the Designated Port for each link
  - The port on that link with the smallest cost toward the RB
- 4) Disable all ports that are neither Root nor Designed



Root Bridge selection (1)

- Choose the bridge with the lowest BridgeID as root
  - The bridge with the lowest priority becomes root bridge
  - If Bridge Priority is the same in every bridge, the lowest MAC address determines the root bridge

Root Bridge Selection (2)

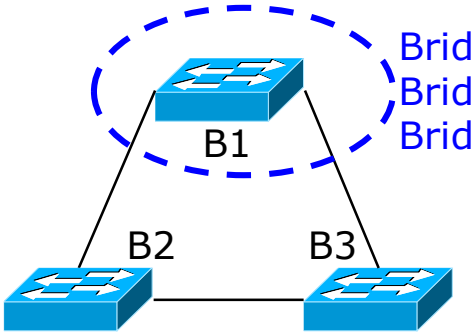


Bridge ID:
Bridge Priority: 28672
Bridge MAC: 00-08-00-11-22-33

Bridge ID:
Bridge Priority: 32768
Bridge MAC: 00-08-00-22-33-44

Bridge ID:
Bridge Priority: 32768
Bridge MAC: 00-08-00-33-44-55

1) Better Bridge Priority



Bridge ID:
Bridge Priority: 32768
Bridge MAC: 00-08-00-11-22-33

Bridge ID:
Bridge Priority: 32768
Bridge MAC: 00-08-00-22-33-44

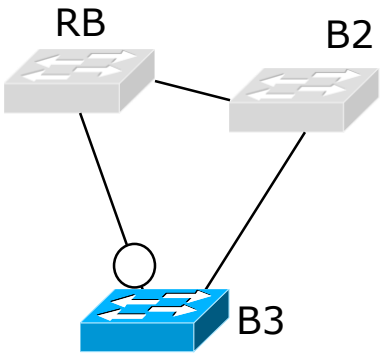
Bridge ID:
Bridge Priority: 32768
Bridge MAC: 00-08-00-33-44-55

2) Better MAC address

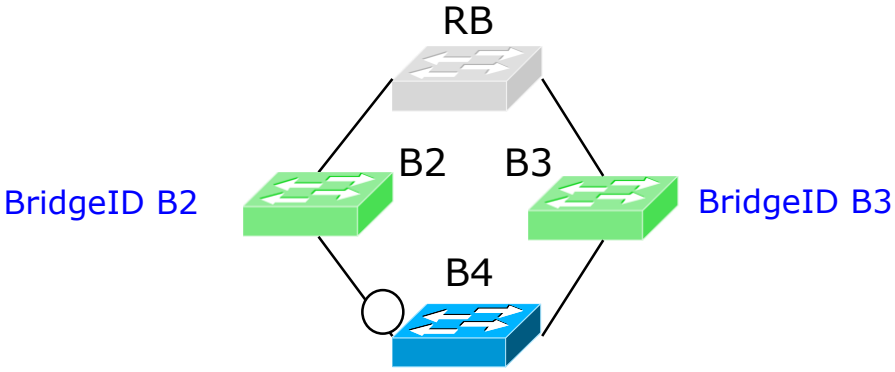
Root Port Selection (1)

- Selection, in each non-root bridge, of the port that has the best path toward the root bridge
  - We have to calculate the *root path cost* for all the ports of the bridge
  - The bridge can have multiple ports; only one will become root
- Among the ports present in a bridge, the root port will be the one that (in decreasing order of priority):
  - Has the smallest path cost toward the root bridge
  - Connects to a bridge (on the path toward the root bridge) that has the smallest Bridge Identifier
  - Connects to a port (on the path toward the root bridge) that has the smallest Port Identifier
  - Has the smallest Port Identifier

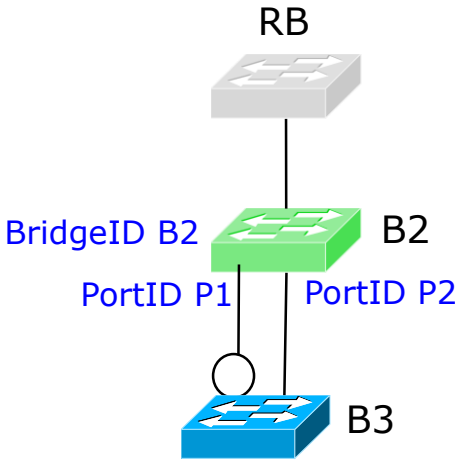
Root Port Selection (2)



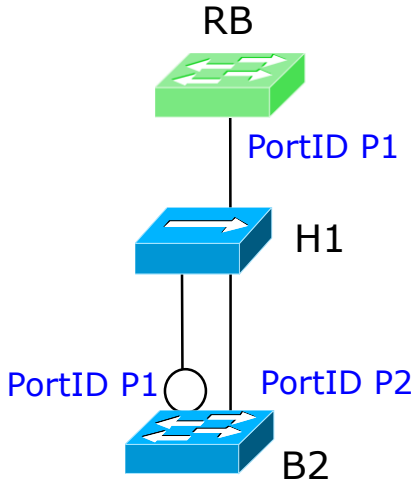
1) Smallest path cost



2) Equal cost
Smallest BridgeID



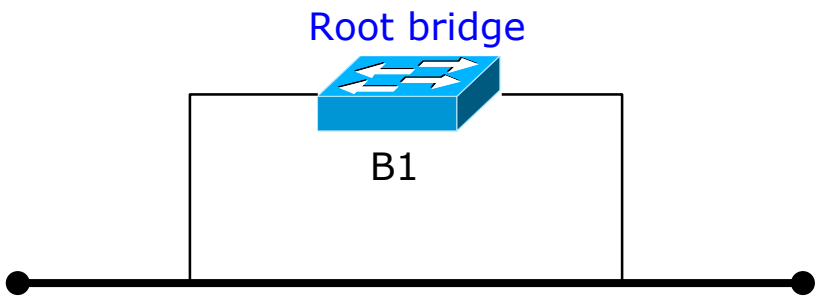
3) Equal cost
Equal BridgeID
Smallest remote PortID



4) Equal cost
Equal BridgeID
Equal PortID
Smallest incoming PortID

Link cost: 1

Question: which is the root port?

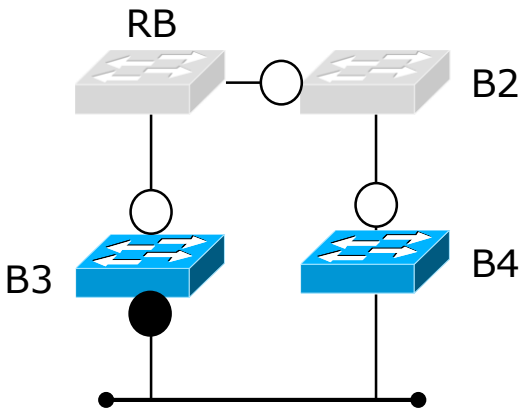


Answer: the root bridge does not have root ports!

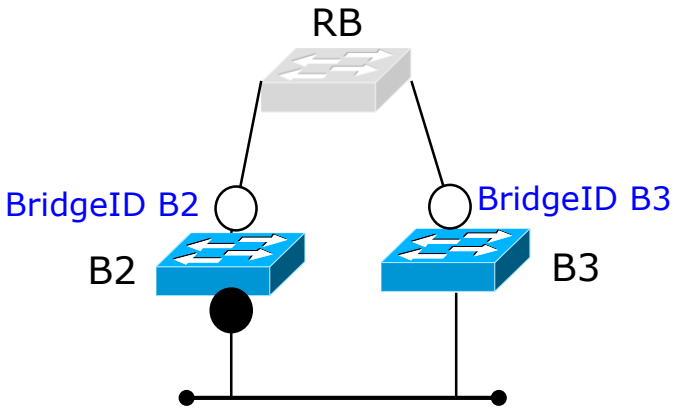
Designated Port Selection (1)

- Selection, for each link, of the port that will be the "master"
  - A link can have only one "master" interface on it (i.e., the designated port)
  - Other ports can be either Root (if the port is the best port of that bridge toward the root) or Blocked
- The Designated Port is the port with the smallest cost, i.e. the one that (in decreasing order of priority):
  - Has the smallest path cost toward the root bridge
  - Is connected to the bridge that has the smallest Bridge Identifier
  - Has the smallest Port Identifier

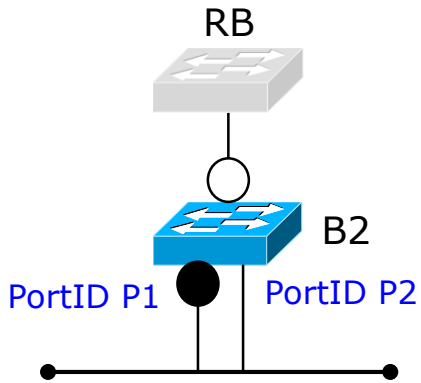
Designated Port Selection (2)



1) Smaller path cost



2) Equal Path Cost
Smaller BridgeID

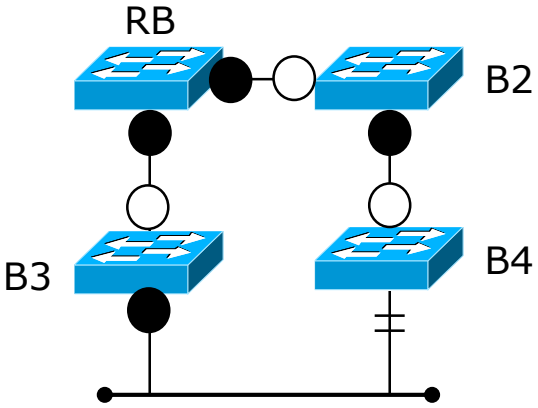


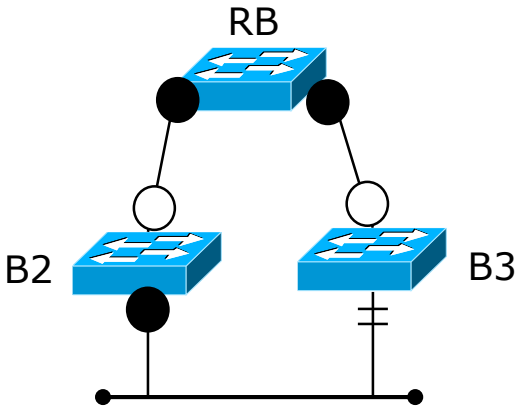
3) Equal Path Cost
Equal BridgeID
Smaller PortID

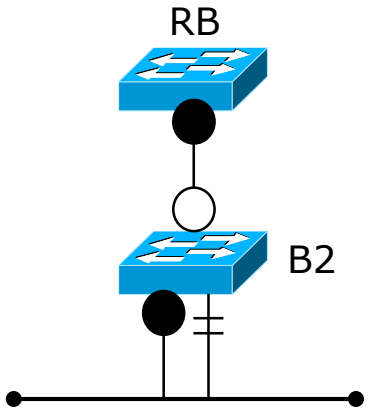
Note: this process is repeated for all the links, although it is omitted in this slide for the sake of clarity

Blocked Ports

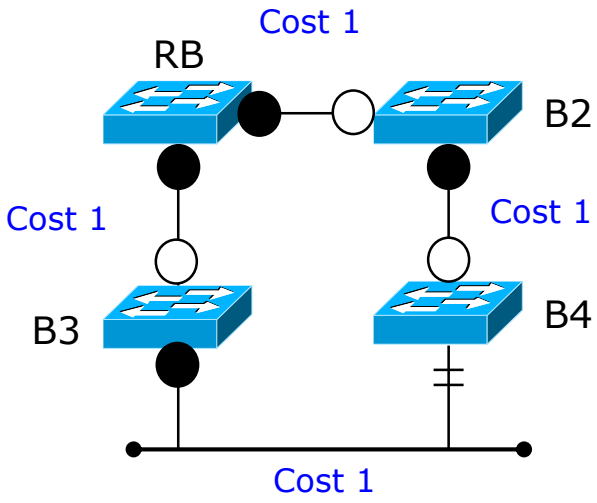
- All the ports that are neither root, neither designated are moved in blocking state
- Some examples (from the previous pictures)

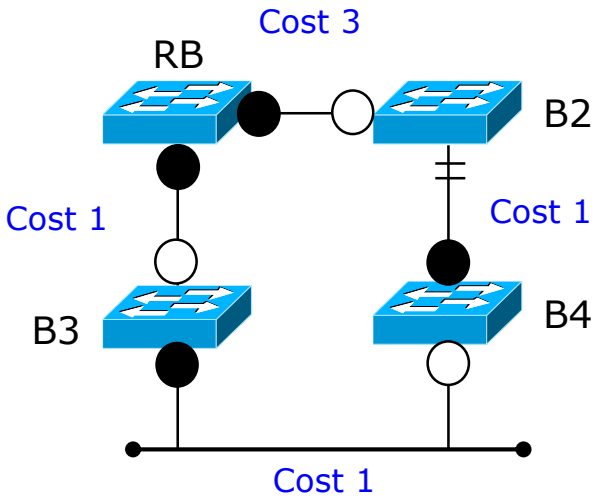


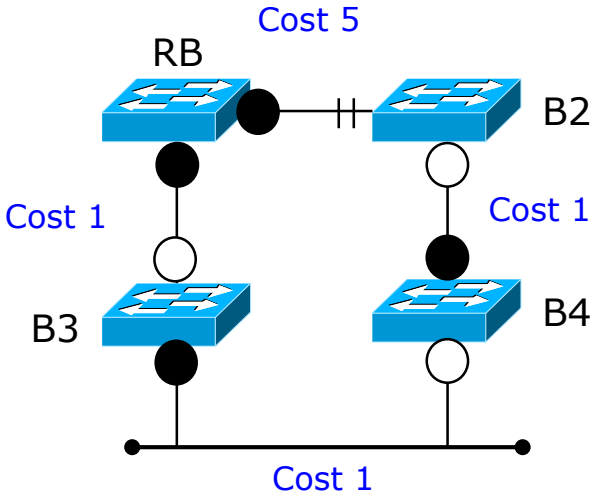




Pay attention to the link cost!







Spanning Tree Protocol

- The algorithm operates in a “static” network
  - No transient
  - Nice pictures on a paper
- What about the real world?

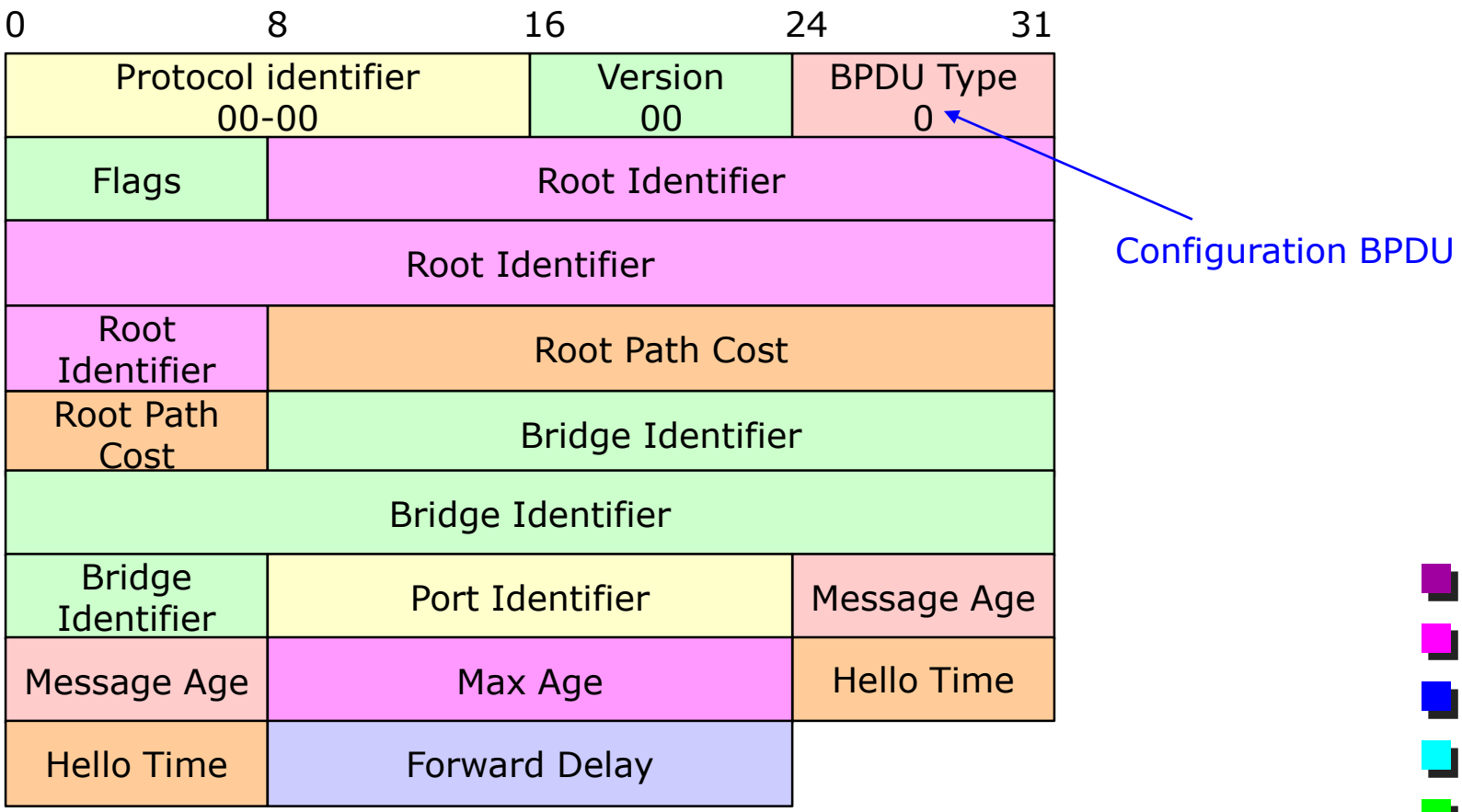
- A proper protocol (802.1D) has been defined

Bridge Protocol Data Unit (BPDU)

- Frame used for STP and network reconfiguration
- Two frames
  - Configuration BPDU
  - Topology Change Notification BPDU
- Source address: MAC address of the bridge that is actually sending the BPDU on the network
  - Changes at every hop

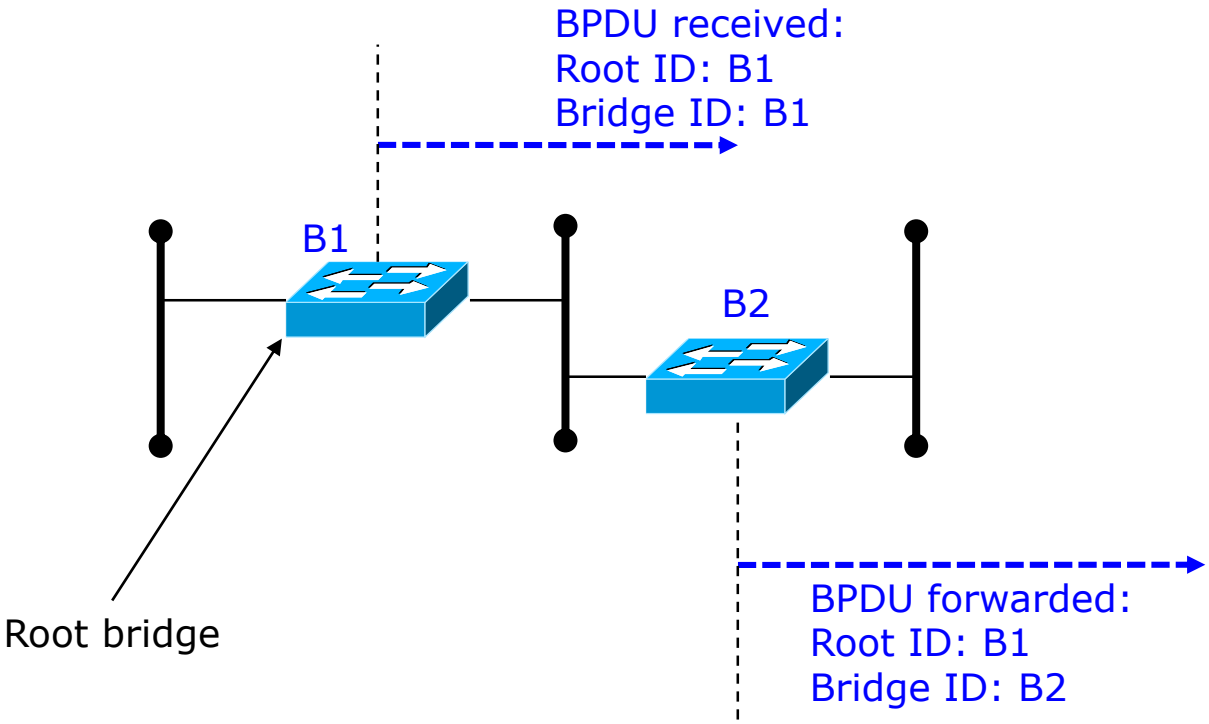
MAC Dest.	MAC Src	Length	DSAP	SSAP	Control	BPDU	FCS
01-00-C2 00-00-00 (multicast)	Bridge Address (unicast)	xx	0x42	0x42	0x03	Configuration BPDU (or) Topology Change Notification BPDU	xx

Configuration BPDU (1)



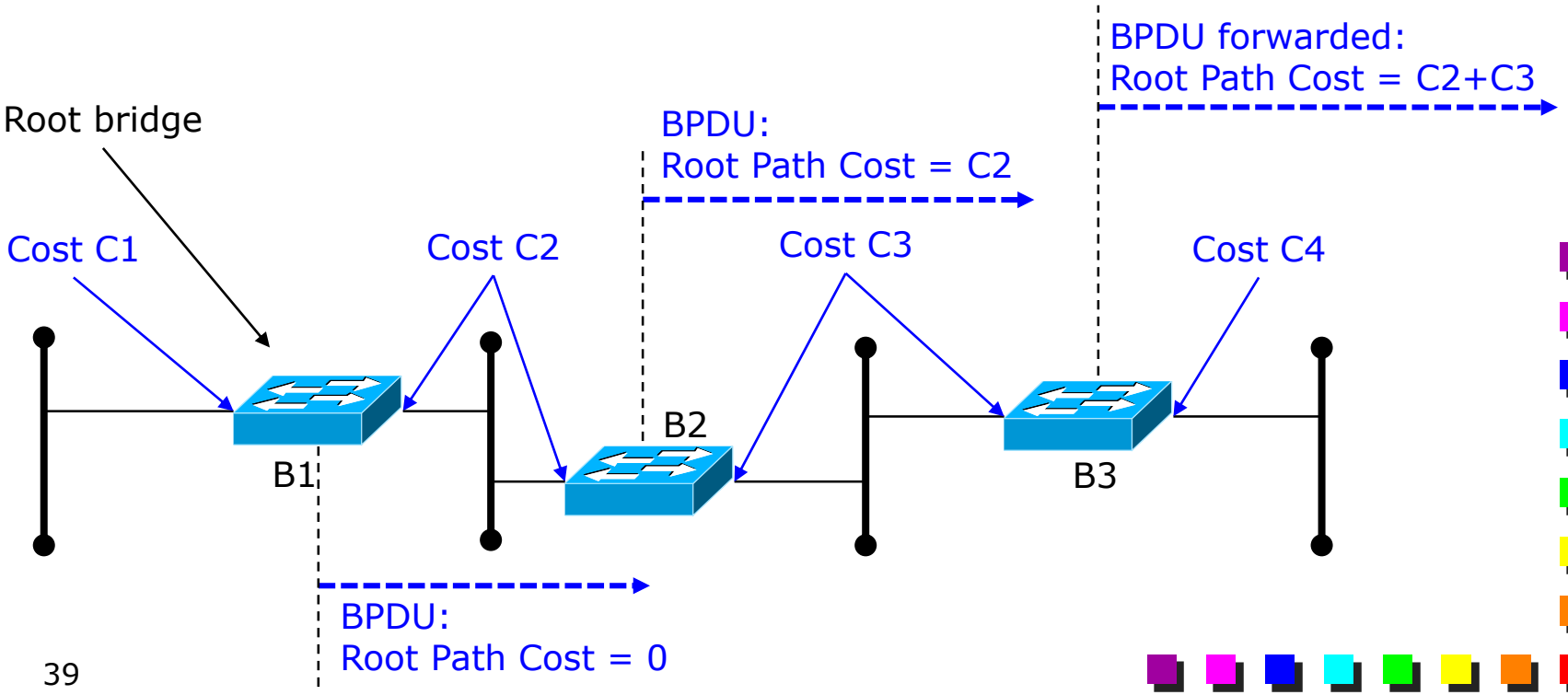
Configuration BPDU (2)

- Root ID: Bridge ID of the root bridge
- Bridge ID: ID of the bridge that is propagating the BPDU



Configuration BPDU (3)

- Root path cost
  - Cost for reaching the root bridge on the path used by the message (as seen by the bridge that forwards the BPDU)
  - Sum of the costs of all the link traversed (but the last link)



Configuration BPDU (4)

- Port Identifier
  - Identifier of the port that forwarded the current BPDU on the LAN
- Message Age, Max_Age
  - Avoids info that lasts forever (zombies)
  - Root Bridge generates BPDU with Age=0
  - A BPDU received by a bridge is valid till Max_Age
    - If the BPDU reaches Max_Age, the information in there is discarded
      - E.g., a new root bridge may be elected
      - E.g., a new port may become Designated
  - In units of /256 seconds (i.e., about 4 ms)

Configuration BPDU (5)

- Hello Time
  - Interval between two consecutive BPDU generated by the root bridge
  - In units of /256 seconds (i.e., about 4 ms)
  - Should be 2*Transit_Delay
    - Or, better… the TransitDelay will be set by the bridge equal to HelloTime/2
    - Not clear what happens if the actual transit delay (i.e. the time required by the bridge to move the BPDU on its port) is bigger than this value

Configuration BPDU (6)

- Forward Delay
  - Time required to force a transition of a port to another state
  - Max duration of filtering database entries when a topology change is detected
  - In units of /256 seconds (i.e., about 4 ms)

- Timers are set by the root bridge for the entire network (are sent on the configuration BPDU)
  - Max Age, Hello Time, Forward Delay

Configuration BPDU (7)

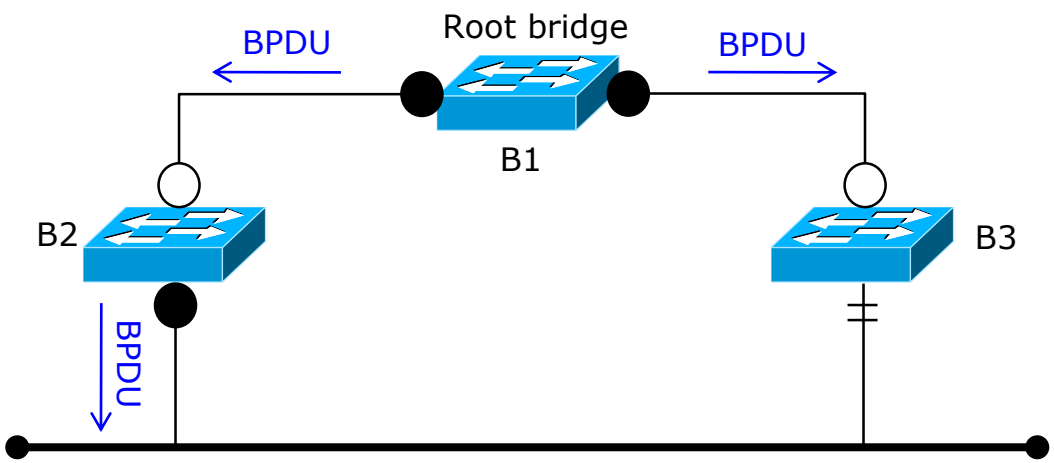
- Flags
  - Only two bits are used
  - Meaningful when a topology change is detected in the network
  - Topology Change: set by the root bridge to inform all the bridges that a change occurred in the network
  - Topology Change Acknowledgement: set by an upstream bridge to inform the downstream node that its Topology Change Notification BPDU has been received
  - More details in the slides that present the topology change phase

Generation of the Configuration BPDU

- Only the Root Bridge can generate the BPDU
  - The root bridge places its Bridge Identifier in the “Root Identifier” and in the BridgeID fields
  - The Configuration BPDUs is sent every “hello time”
  - The Age field in the message is set to “zero”
- All the other bridges simply *propagate* the received BPDU on all their *Designated* ports
  - Blocked ports do not forward any BPDU
  - Designated ports inject the BPDU on the link they are serving
  - Some BPDU fields are updated (more details later)

Propagation of the Configuration BPDU

- Root Ports are the ones that receive the *best* BPDU
  - BPDUs can be received also on Blocked ports
  - The BPDU is propagated only when it is received on a Root port
- Blocked ports never *send* BPDU but *listen* to BPDU sent on that link
  - In case no BPDUs are received for a while, they can become Designated
    - More details later



STP dynamic behavior (1)

- The beginning
  - Each bridge assumes to be root
  - All its ports assume to be the best ports on their links and therefore they become Designated
  - Each bridge will then generate its own BPDU in flooding on all its ports

STP dynamic behavior (2)

- When a BPDU is received with RootID < CurrentRootID
  - The bridge recognizes there is a better bridge on the network
    - It may be either that the current bridge was the root, or the "new" root is better than the "old" root
  - It stops the generation of its own BPDU
  - It begins repeating the new BPDU, with some updated info
    - BridgeID (contained in the BPDU) is replaced with its BridgeID
    - PortID (contained in the BPDU) is replaced with the ID of the port that is going to propagate the BPDU
    - Root Path Cost is set to the previous cost (contained in the BPDU) plus the cost of the incoming link
    - Message Age is updated
      - Increased by the time needed to forward the BPDU (TransitDelay)
      - TransitDelay= HelloTime/2

STP dynamic behavior (3)

- When a BPDU is received with RootID > CurrentRootID
  - Ignored

- Important: the BPDU is repeated **only** when it is received on the root port
  - A non-root bridge never generates its own BPDU, nor propagates it on the network autonomously (without having received it from the root bridge)

STP dynamic behavior (4)

- When a BPDU is received with RootPathCost < CurrentRootPathCost
  - The port that received the BPDU will become Root Port
  - Better Root Path Cost:
    - Smaller Root Path Cost contained in the BPDU plus the cost of the incoming link (or, if equivalent)
    - Smaller BridgeID contained in the BPDU (or, if equivalent)
    - Smaller Port ID contained in the BPDU (or, if equivalent)
    - Smaller local PortID

STP dynamic behavior (5)

- When a BPDU is received with RootPathCost > CurrentRootPathCost
  - Ignored (with respect to the Root Port Selection)
  - May be generated by another bridge that share a link with the current bridge

STP dynamic behavior (6)

- When a BPDU is received with Priority < CurrentPriority
  - The port that received the BPDU will become Blocked
  - Better Priority:
    - Small Root Path Cost contained in the BPDU (or, if equivalent)
    - Smaller BridgeID contained in the BPDU (or, if equivalent)
    - Smaller Port ID contained in the BPDU

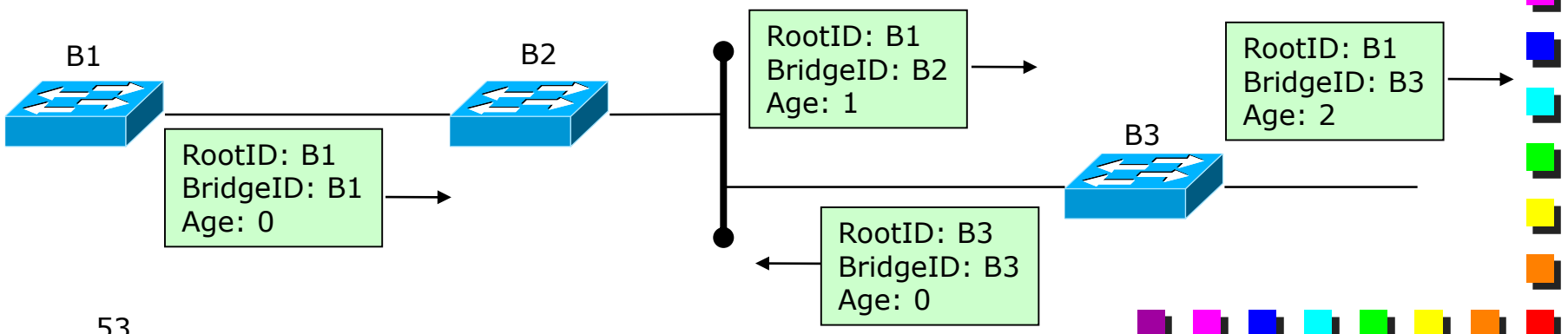
- When a BPDU is received with Priority > CurrentPriority
  - Ignored (with respect to the Designated Port Selection)
  - May be generated by another bridge is not yet part of the network
    - Actually, two Designated Ports exist on the link, but one is going to become Blocked soon

STP dynamic behavior (7)

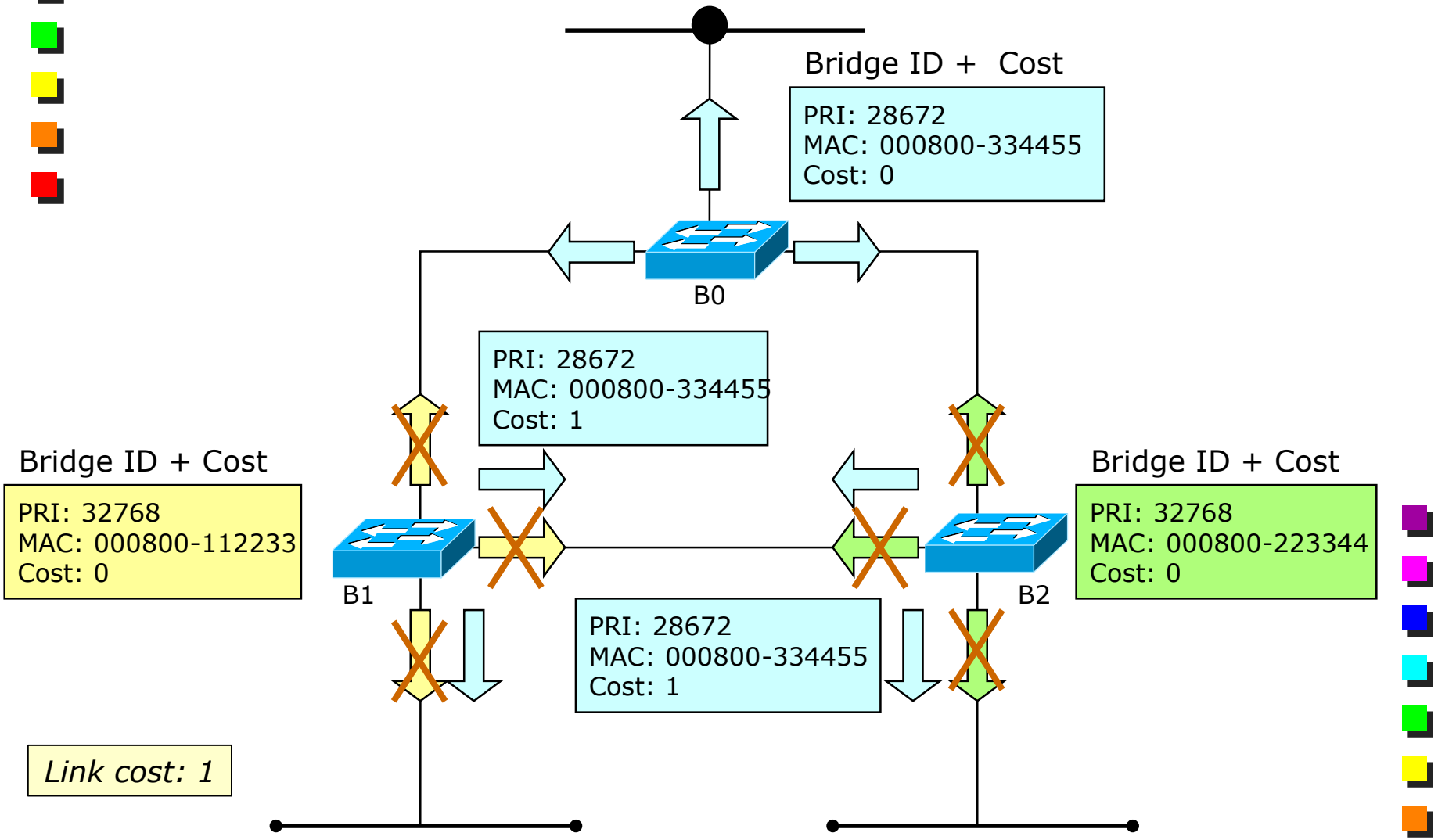
- At the end, only the Root Bridge generates the BPDU on the network
- All the Designated Ports propagate the BPDU toward the leafs
- All the Blocked ports stay silent (but can receive BPDU propagated by the designated port on the link)
- All the Root ports are *silent* as well (they receive the BPDU from the root bridge)
  - At least for what concerns the STP
  - Of course, root ports propagate the *data* traffic

STP dynamic behavior: example

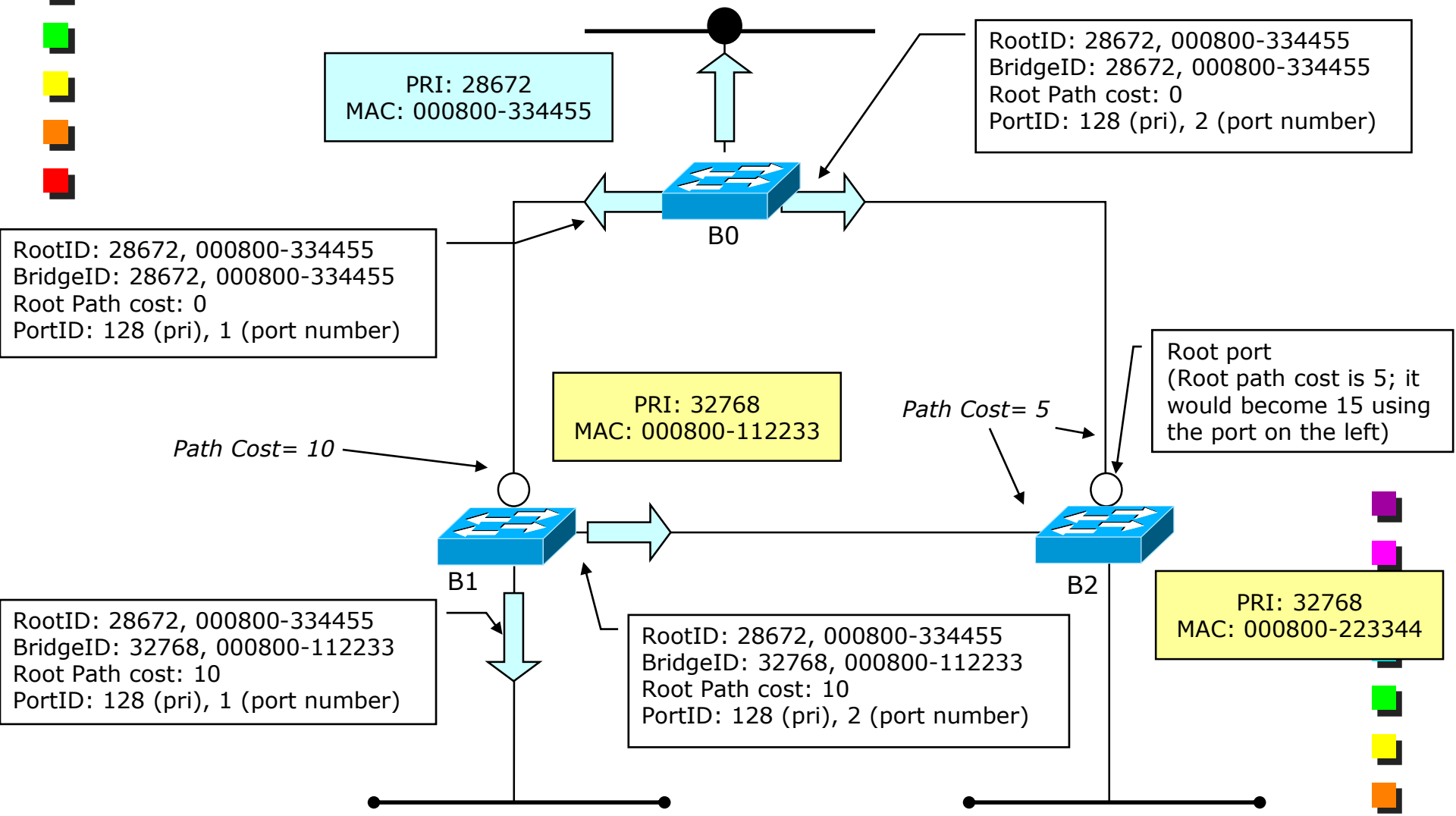
- Let's image another bridge appears on the network
  - At the beginning it will send a BPDU announcing itself as a root (although it does not have the right to be root)
    - It does not know anything about the rest of the network (yet)
  - The Designated bridge for that link will ignore the BPDU
  - The new bridge will known the actual root bridge as soon as the bridge B2 will forward a BPDU to it
    - This happens as soon as B2 will receive a new BPDU from its upstream bridge



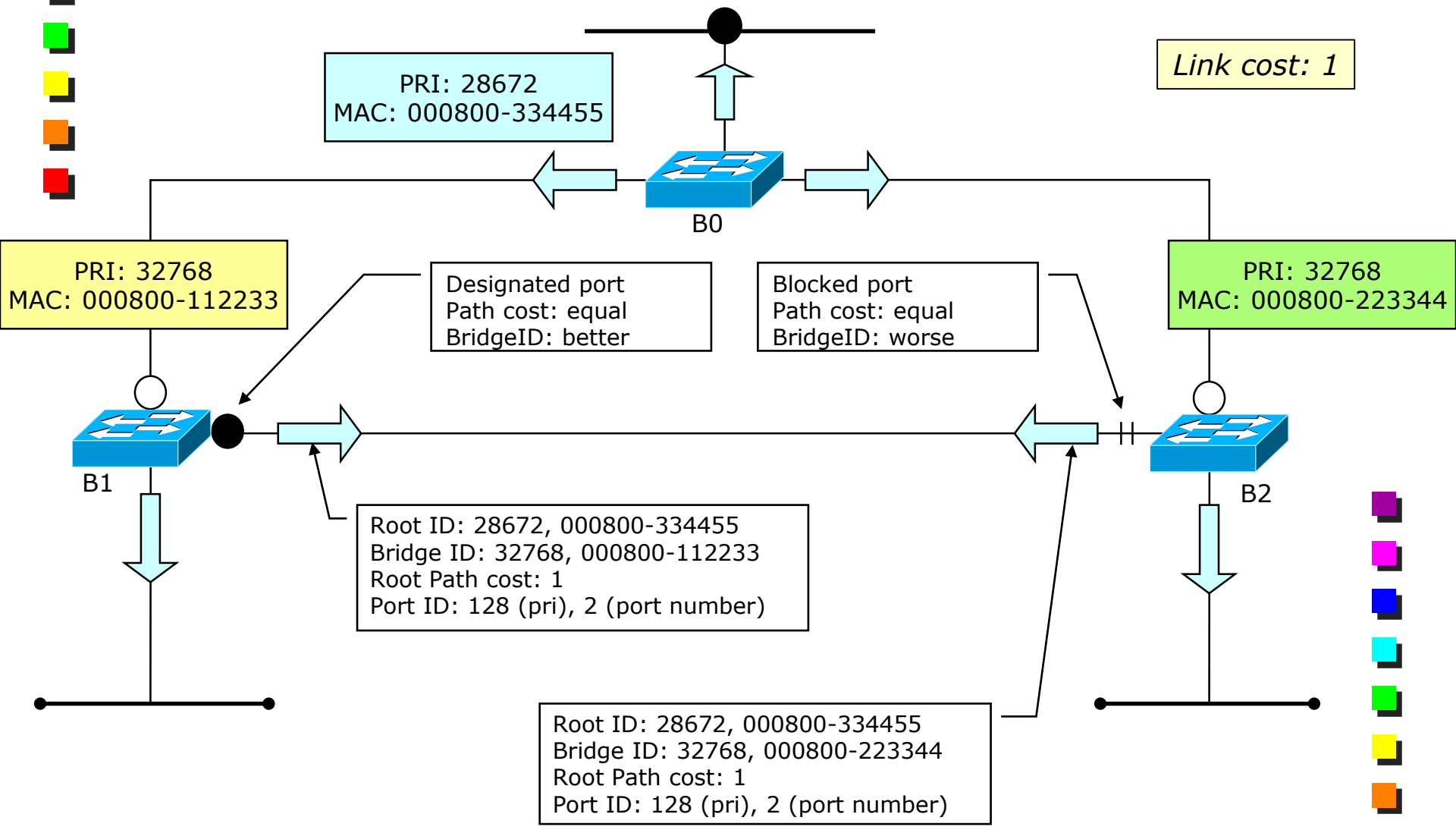
Root Bridge Selection: example



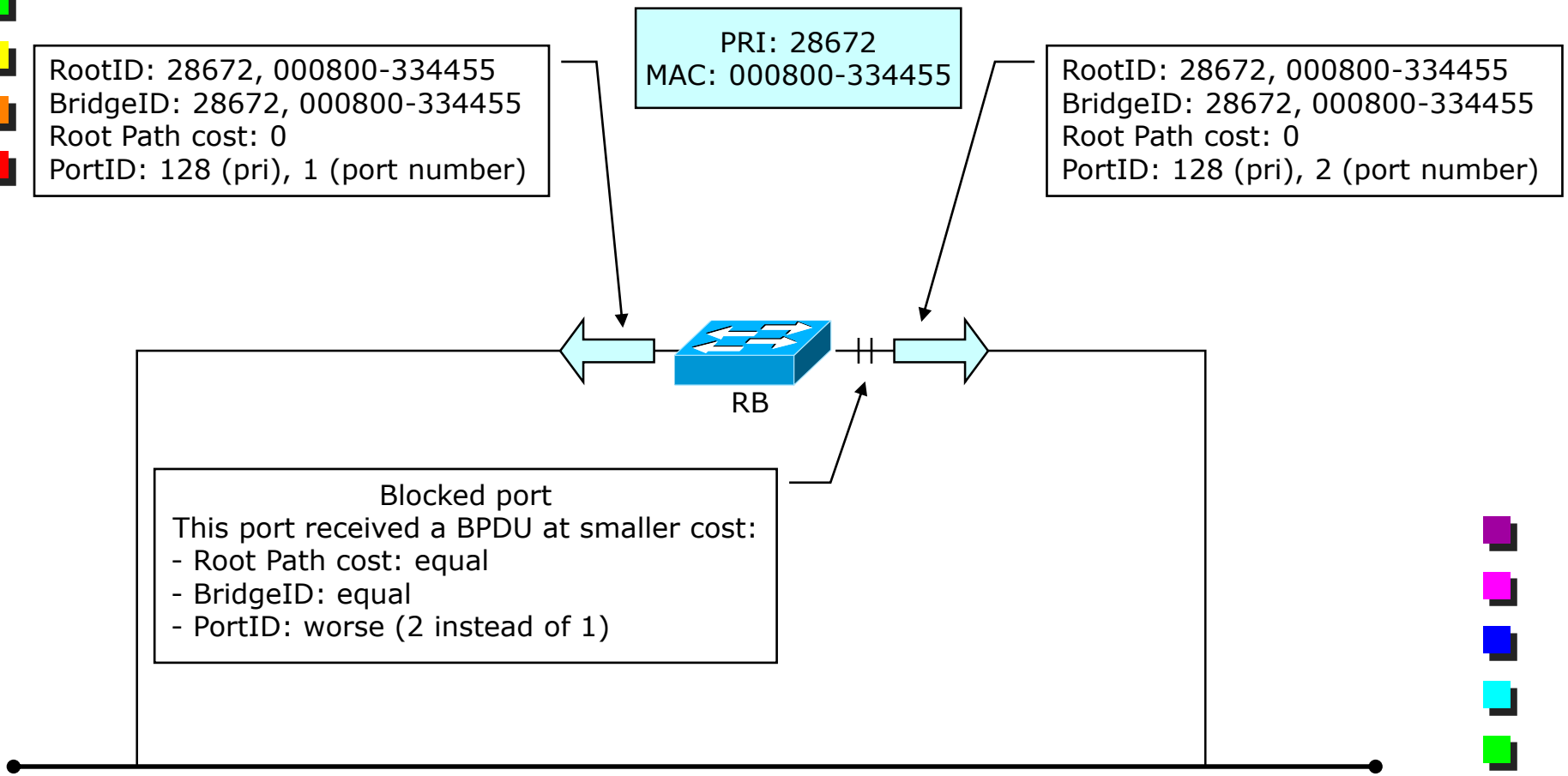
Root Port Selection: example



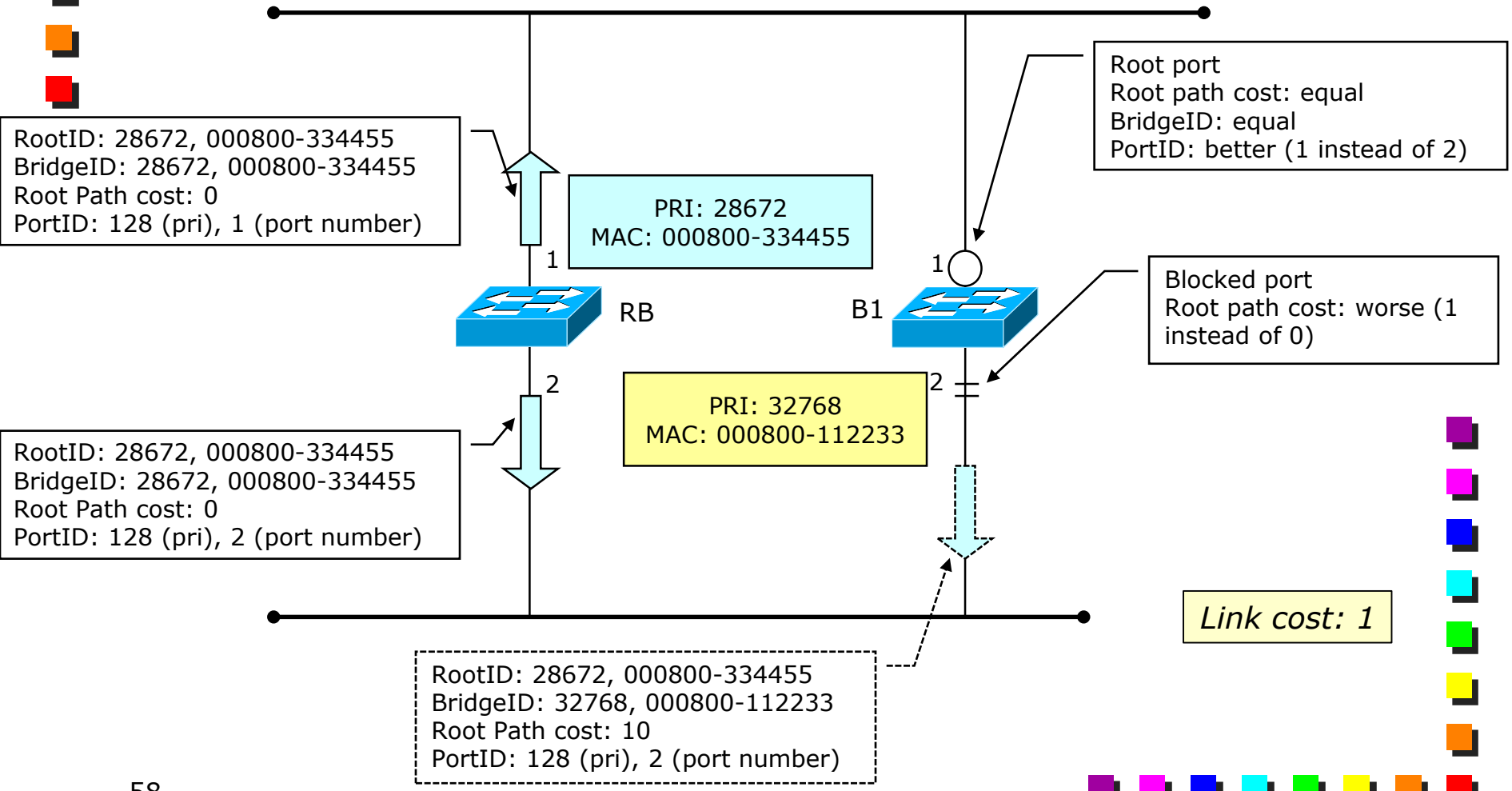
Designated Port selection: Example



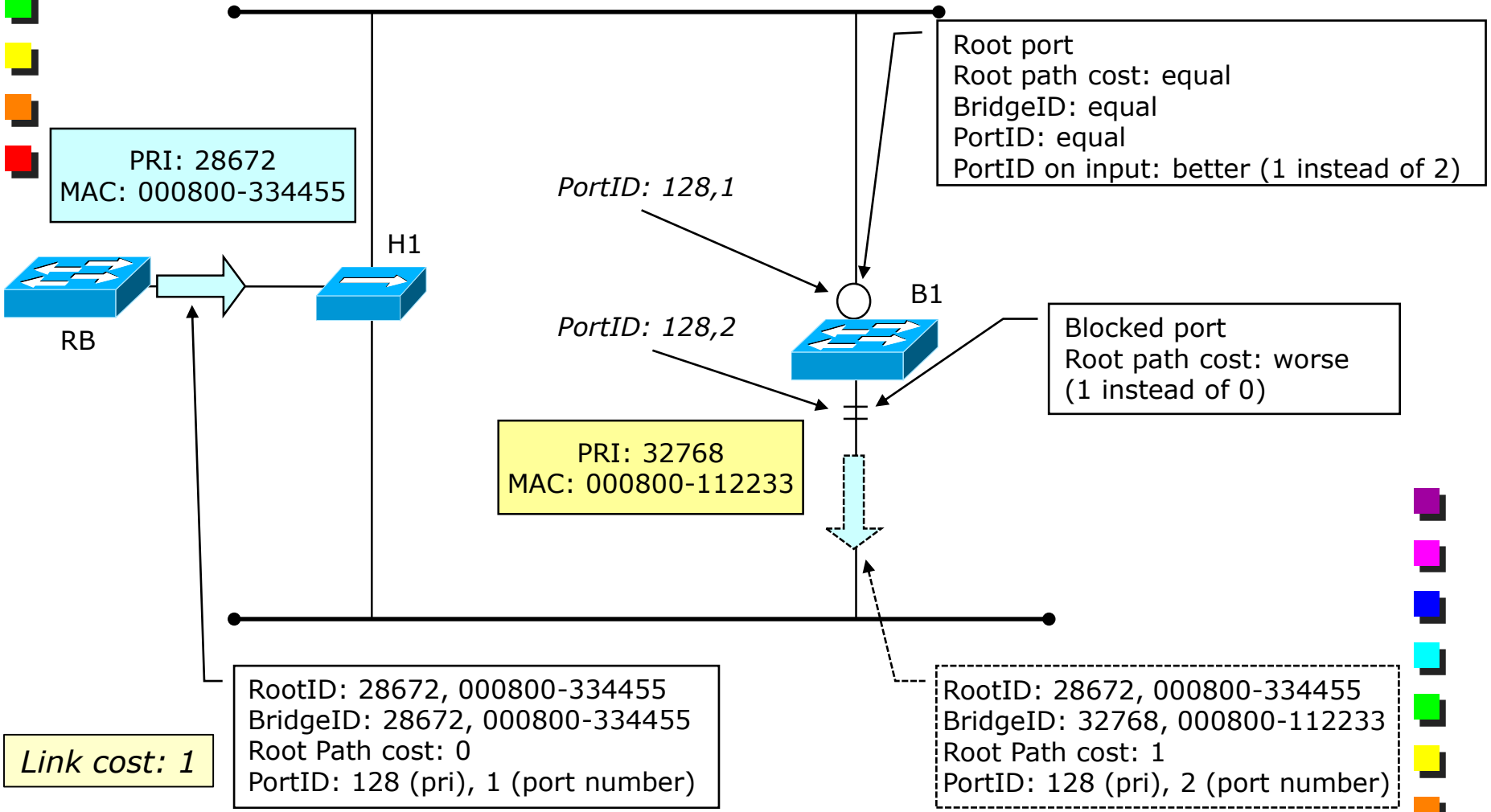
Spanning Tree Example: one bridge



Spanning Tree Example: two bridges



Spanning Tree Example: two bridges and one hub



Updating the current STP topology

- The current topology (Spanning Tree) is re-computed when
  - A bridge receives a better BPDU
    - Root Port updated if another port (on the same bridge) has a better Root Path Cost
    - Designated Port updated if another port on another bridge (on that link) is better
    - Blocked Port updated if it turns out to be the best port on the link
  - The BPDU of the Root Bridge expires due to Max_Age
    - Corollary: STP requires MaxAge seconds to discover a fault toward the root bridge

STP Configuration parameters

- Bridges are plug&play
  - They may work with default configuration parameters
- Bridge priority
  - Range: 0 - 61440
  - Default/recommended: 32768
  - Suggested increment (IEEE 802.1t): 4096
- Port priority
  - Range: 0 - 240
  - Default/recommended: 128
  - Suggested increment (IEEE 802.1t): 16

Path cost recommended by IEEE 802.1D (1998)

- Port Path cost
  - Depends on the bandwidth of the interface
  - Range: 0 - 65535
  - Recommended (IEEE 802.1D): 1000/(Speed in Mb/s)

Port speed	Recommended Value	Recommended range values
4 Mb/s	250	100 – 1000
10 Mb/s	100	50 – 600
16 Mb/s	62	40 – 400
100 Mb/s	19	10 – 60
1 Gb/s	4	3 – 10
10 Gb/s	2	1 - 5

Changing Path Costs

- Modern network devices allow to change the path cost of the interface
- Be careful to change the cost on both sides of the link, otherwise loops may occur

STP Configuration parameters: Timers

- Hello time
  - Range: 1 - 10 seconds – Recommended: 2 seconds
- Forward delay timer
  - Range: 4 - 30 seconds – Recommended: 15 seconds
- Max age
  - Range: 6 - 40 seconds – Recommended: 20 seconds

STP and network topology

- The STP does not propagate the network topology
  - All the decisions are kept local, and the BPDU is used to synchronize all the network
  - Bridges know their relative position with respect to the root bridge, but they do not know the network topology
  - Radical departure from other algorithms
    - Link State routing
  - In fact, the Spanning Tree algorithm looks more closely to the Distance Vector routing
    - Both cooperate to calculate the best path, while the global network topology is unknown

STP and port state transitions (1)

- When the topology changes, some bridges will already have the new topology, while other are still with the old one
  - In this case, loops are possible
- How can we avoid loops in the transient?

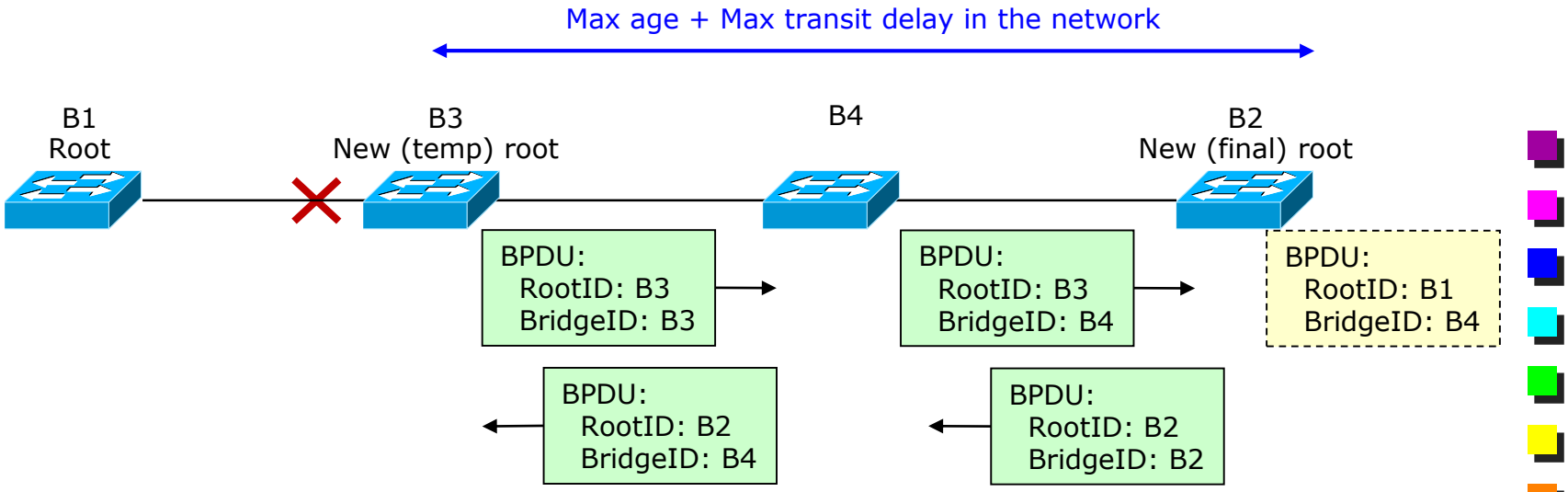
- The idea
  - “Deny always, allow only when sure”

STP and port state transitions (2)

- The idea in details
  - 1) If the old topology was fine, let's keep that until we are sure every bridge knows the new topology
    - We may have loss of connectivity while keeping the old topology, but for sure we do not have loops
  - 2) Ports can go immediately in Blocked state (i.e., new links pruned), but they have to wait more to enter into a Root/Designated state
    - We may prune additional links first, and enable new paths only at a later time
- How much do we have to wait before activating a link?
  - We need to compute the maximum time needed for all the bridges on the network to get informed about the change and prepare for the new topology

STP and port state transitions (3)

- Max time to have the network aligned with the new topology
  - In this example, Max Age + max Transit Delay in the network
  - In general, 2 times the time required by a BPDU to transit from one side of the network to another
    - 14 seconds with standard parameters



STP and port state transitions (4)

- After that time, all the bridges should have the new topology
  - They can activate their ports and start forwarding data frames
  - Obviously, ports are not activated at the same time, but it does not matter
    - The network will become more and more operational as soon as new links are being added
- During transient, some stations may not be reachable
  - Black holes are always preferred to broadcast storms

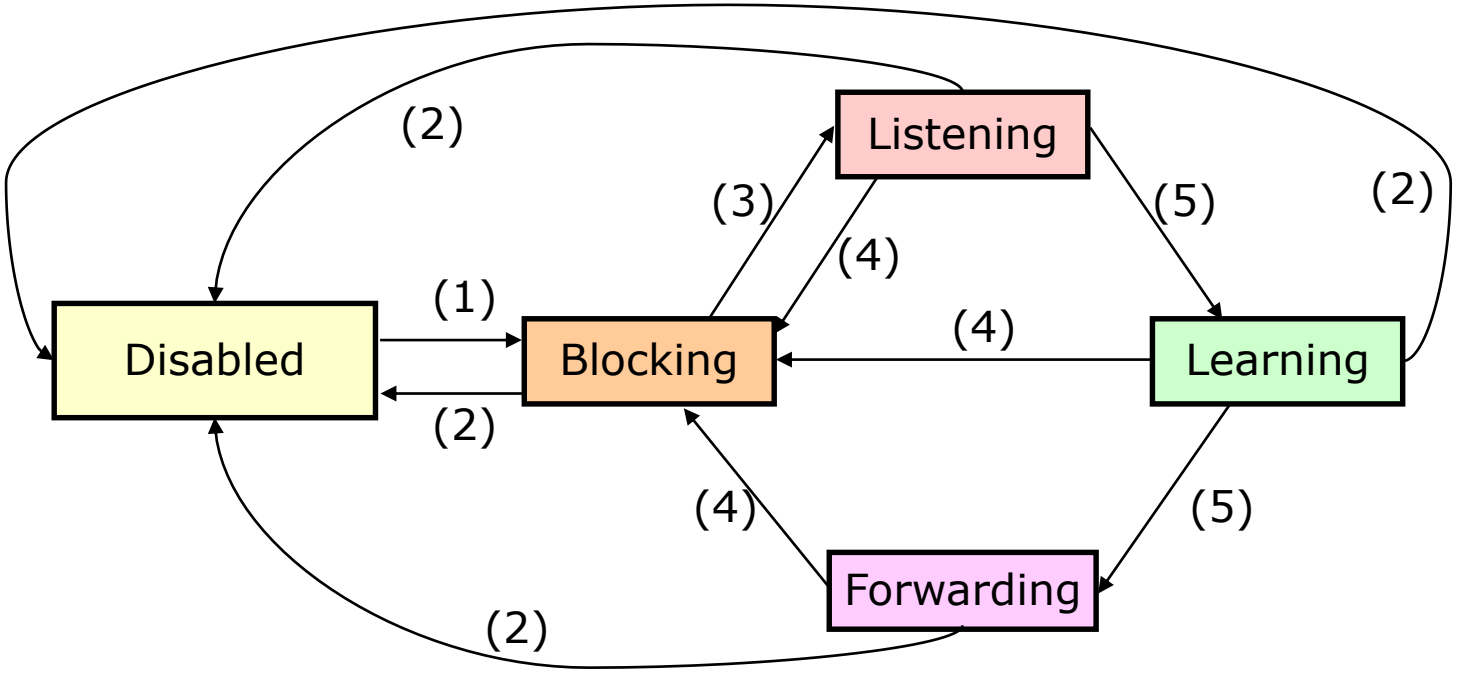
STP and port state transitions (5)

- Let’s recap port status, focusing on data frames (not BPDU)
- In principle, ports can either forward packets or be in a blocking state
  - Forwarding
    - Ports currently forwarding packets
    - Root and Designated ports
  - Blocking
    - Ports that ignore received packets, which are not forwarded
    - All the remaining ports

STP and port state transitions (6)

- In order to cope with transient phases, more states have been defined before going into a forwarding state
  - Listening and Learning
- In principle, one is enough
  - It was called “preforwarding” in the original STP proposal by Perlman
  - The standardization committee decided to add the Learning phase in order to limit flooding on the network

STP and port state transitions (7)



(1) Port enabled by management or initialization (e.g., at boot)
(2) Port disabled by management or failure (e.g., "Error disable" status, disconnected cable)
(3) Port selected as Designated or Root Port
(4) Port selected as Blocking Port
(5) Protocol timer expiry (Forwarding Timer)

STP and port state transitions (8)

- Ports require 2*ForwardDelay to become fully operational
  - 30 seconds with standard values
- Ports transition to the blocking state immediately (if required)
- The idea
  - 2*ForwardDelay should guarantee that the in the worst case the BDPU can traverse the entire network two times
    - I.e., from bridge A to bridge Z and back, which are the outmost bridges on the network
- Algorithm implemented on all ports of a bridge
  - A PC that connects to the network is isolated for 2*ForwardDelay
  - STP does not make any difference between backbone ports and ports that connect to a PC

STP and port state transitions (9)

	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

STP and port state transitions (10)

- Listening and Learning
  - **The port is already active from the STP point of view**
    - BPDU sent/received
  - **The port is blocked from the point of view of data forwarding**
    - We do not want loops!
- Learning
  - Useful to populate the filtering database
  - Remember: if a MAC entry is not in the table, frames directed to that MAC address are flooded
    - Poor performance
  - At the end of the learning phase we should have most of the MAC entries already in, hence limiting the flooding

Spanning Tree and convergence (1)

- Two key parameters
  - Max Age (default: 20 sec): validity of a received BPDU
  - Forward Delay timeout (default: 15 sec): waiting time before changing the status of a port
- Convergence in approximately 50 sec
  - MaxAge + 2 * ForwardDelay
  - Max time needed to detect a fault, plus the time required to turn the port on (Listening → Learning → Forwarding)

Spanning Tree and convergence (2)

- MaxAge
  - Required to detect that the root bridge is expired (or)
  - The current path toward the root bridge is no longer available (and another root port must be elected)
    - Can be avoided if the bridge detects the fault on its Root Port at the physical layer
    - In this case the bridge will reconfigure the topology in 2*ForwardDelay
    - If the fault is on the path toward the root bridge, but not on the bridge root port, we need to wait MaxAge
- ForwardDelay
    - Required to change the state of the bridge ports and turn them on (the one needed in the new topology)

Customization of STP parameters (1)

- The recommended timers guarantees a STP convergence of 50 seconds
- Changing the timers may:
  - Increase or decrease the STP convergence
  - Increase or decrease the maximum bridge diameter
- Not easy to do
  - Suboptimal values can:
    - Reduce network reactivity to topology changes
    - Impair network functionality (loops!!!)

Customization of STP parameters (2)

- Hard to manage
  - Values are propagated by the Root Bridge to the entire network, but…
  - If the root bridge changes, the new root bridge must advertise the same values
    - Just in case, we should update those parameters on all the bridges
    - If we forget one and this becomes root bridge, we may have broadcast storms

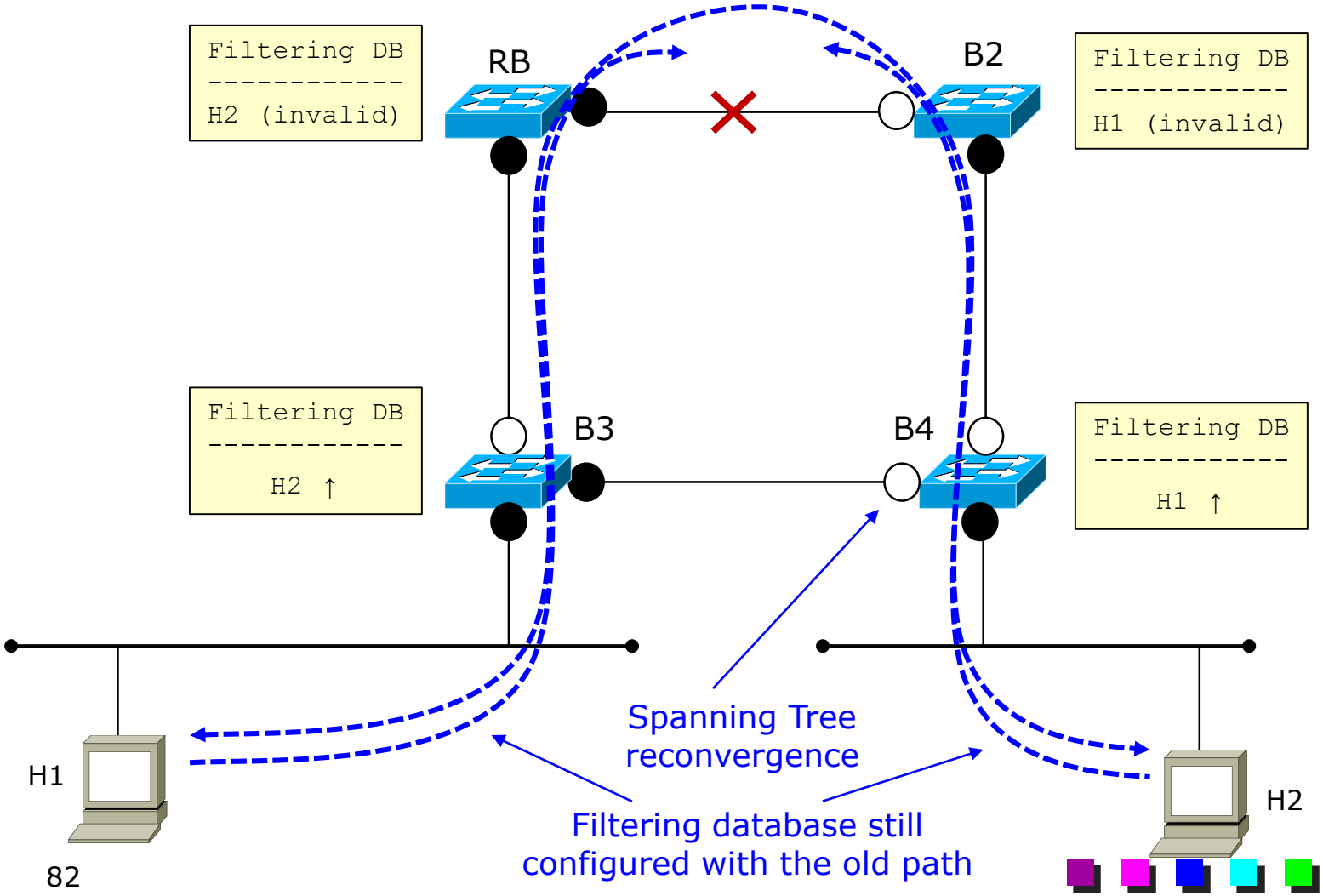
Customization of STP parameters (3)

- IEEE 802.1D shows how to get optimal parameter values starting from:
  - *max bridge diameter* (default: 7 bridges): maximum number of bridges between two end-systems
  - *maximum bridge transit delay* (default: 1 s): maximum time needed by a BPDU to cross a bridge
    - It starts from the arrival to the departure, including processing
- The recommended timers ensure a correct STP behavior with a maximum bridge diameter up 7 to bridges
- The results for all the timers must be computed
  - The hello time is usually twice the maximum bridge transit delay
  - The hello time is often set to 1s for optimization purposes

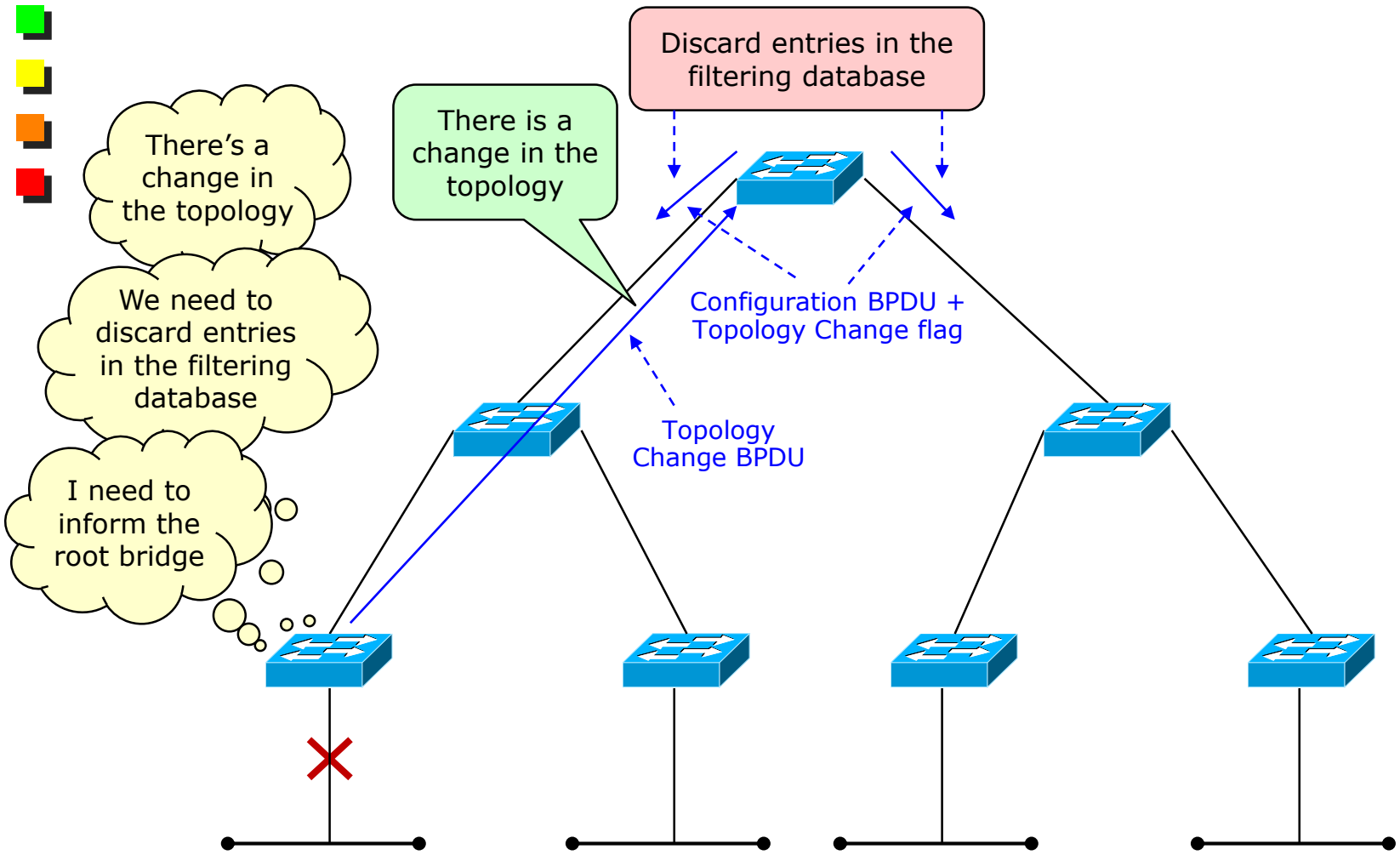
Filtering Database: purging entries (1)

- Objective: speeding up the convergence of the network with respect to the filtering database when a network topology change is detected
  - Entries in the filtering database are not affected by the topology update, driven by the STP, and therefore these entries can be out of date
  - In the worst case, an host can loose connectivity for Aging Time (e.g., 5 mins)
  - Please note that is value is much higher than the STP convergence time (50 seconds)
    - The fault, as experienced by the user, will last 5 minutes, not 50 sec!
- STP defines a new message for this purpose

Filtering Database: purging entries (2)

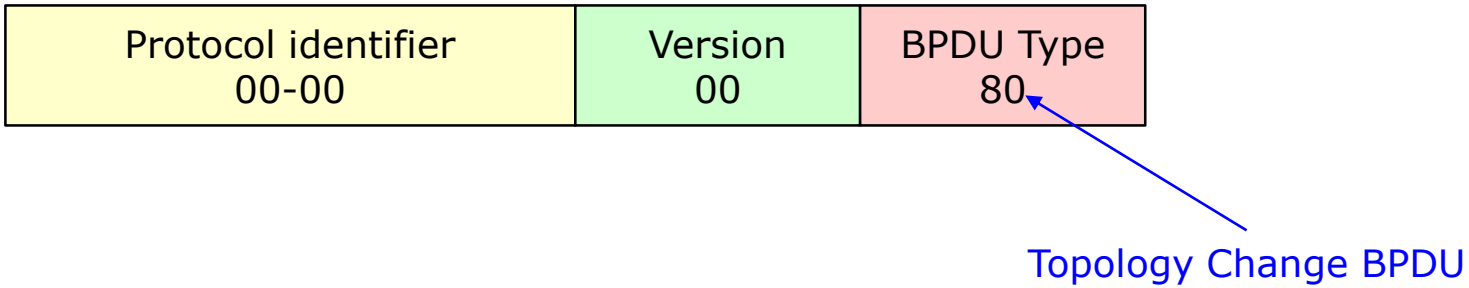


Topology Change process: the idea

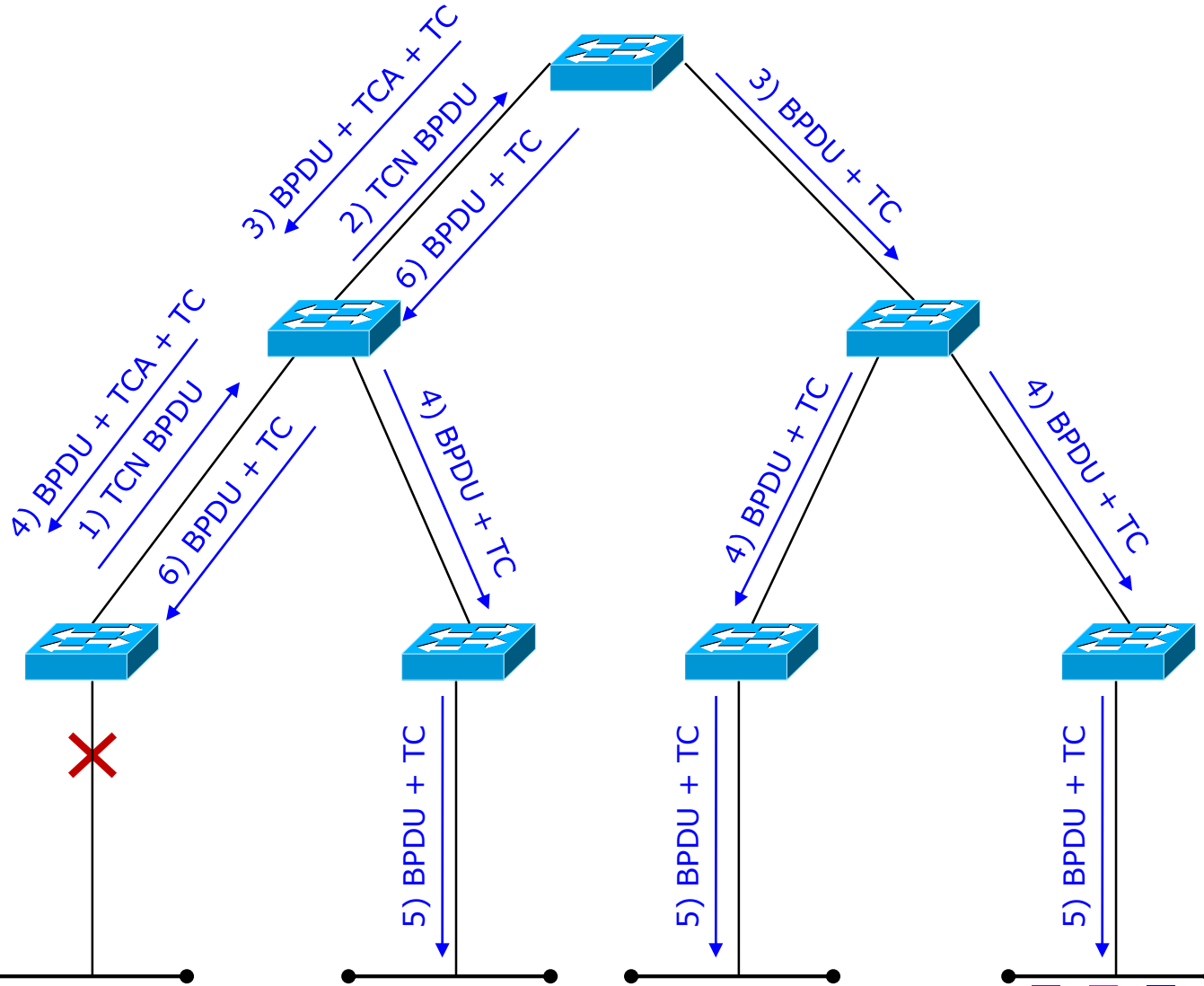


Topology Change BPDU

- Informs the network that a topology change has occurred
  - Entries are dropped in order to force bridges to propagate frames directed to unknown MACs in flooding
    - All the bridges learn the new direction needed for reaching the station
- Generated by the bridge that detected the fault and sent *toward* the root bridge



Algorithm for Topology Change (1)



Algorithm for Topology Change (2)

- The bridge that detects a topology change sends a Topology Change Notification BPDU through the root port toward the Root Bridge
- When a bridge receives a TCN BPDU
  - It forwards immediately the frame through the root port (toward the root bridge)
  - It acknowledges the TCN message with a Configuration BPDU with the TCA bit set on the port it received the TCN
    - The bridge has to wait for a BPDU from the root bridge in order to ack the TCN
- Each bridge repeats the message on its root port every Hello Time until its upstream bridge sends the acknowledge

Algorithm for Topology Change (3)

- When the Root Bridge receives a TCN BPDU
  - It acknowledges the previous message as usual
  - It generates a Configuration BPDU with bit Topology Change set
- The BPDU is propagated down in the network to the periphery, as usual but the TC bit is kept set
  - "A topology change has been detected in the network"
- Root bridge generates C-BPDUs with the TC flag for MaxAge + ForwardDelay
  - Default 20 + 15s

Algorithm for Topology Change (4)

- When a BPDU with the TC flag is received, the bridge sets the MaxAge equal to the ForwardDelay timer
  - Entries in the filtering database will expire much sooner
  - Bridges that detect a topology change are both the bridge that detected the fault and the bridges that received the BPDU with the TC flag
  - By reducing the aging time, we do not clear the active hosts from the table
    - Flooding applies only to hosts that were inactive at that time

Topology change detection

- Events that trigger a Topology Change
  - When a port in forwarding state is going down (blocking, or shut down, a fault at the physical layer)
  - When a port moves in forwarding state and the bridge has a designated port
    - This means that the bridge is not standalone
- Some examples
  - When the MaxAge expires and the bridge updates its tree
  - When the bridge receives a BPDU with a better Root Bridge
  - When the bridge receives a BPDU that changes the Designated port on a link
  - When the bridge receives a BPDU with a better cost to the Root bridge

Topology changes and large networks

- A TCN can be sent each time a PC disconnects from its bridge/switch
- Can cause a sub-optimal behavior of the network
  - Filtering database continuously flushed, i.e. large amount of flooded traffic
- STP can be disabled on "edge" ports
  - Dangerous, as we may have loops in the network
  - Some vendors do not allow this (e.g. Cisco)
    - However, Cisco has the PortFast mechanism, which achieves the same objective
      - It turns also the port on without Listening + Learning states
    - TCN are not sent each time a PC disconnect from the network, therefore avoiding continuous filtering database flushing

STP and dynamic networks

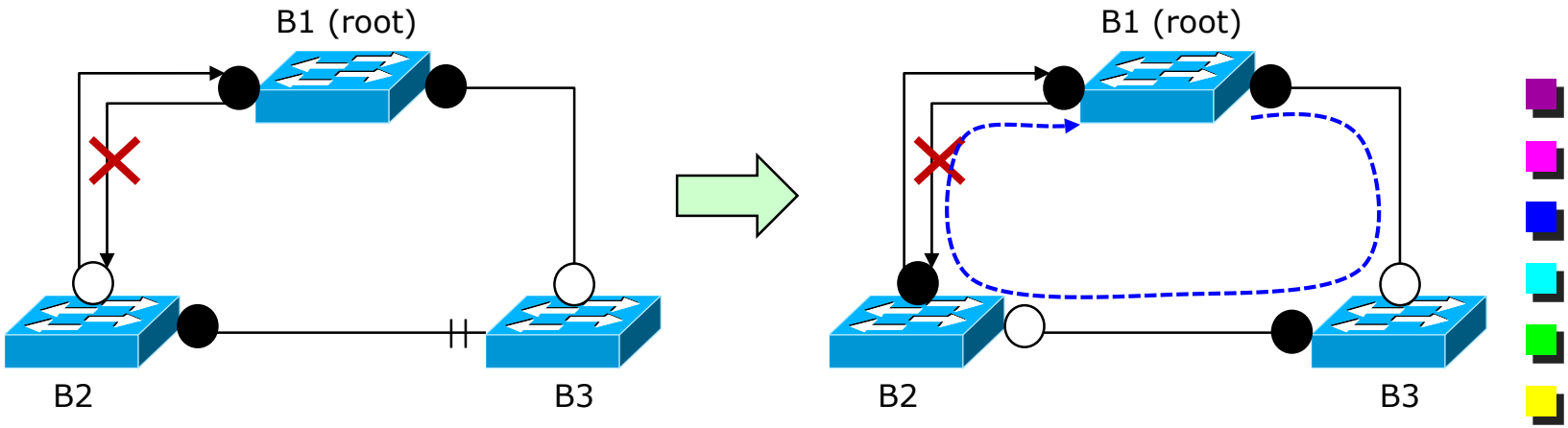
- STP operates when the network is reasonable stable
  - Convergence time rather high (50s)
  - Please note that 50s of network fault may trigger some other nasty problem
    - E.g. a database does not get properly realigned, requiring a recovery procedure (maybe several hours long)
- In case of frequent changes in the STP topology
  - Frequent loss of connectivity
  - Limited age for filtering database entries

STP and corporate / metropolitan networks

- STP does not have security mechanisms built-in
- A user can connect a bridge to the network and force this to become the new root bridge
- Manufacturers offer the possibility to disable the “edge” ports
  - Cisco does not this explicitly, but the “Bridge BPDU Guard” features rejects BPDU when received on a given port (and the port is moved in the “error disable” state)
  - Ports that have the BPDU Guard active still send BPDUs

STP and unidirectional paths

- The BPDUs are forwarded from the root to the leaves
  - The propagation is unidirectional
- If the "direct" propagation path breaks, the other endpoint assumes that the port is a designated one
  - The upstream bridge does not detect the failure
  - A loop is created and broadcast storms are possible

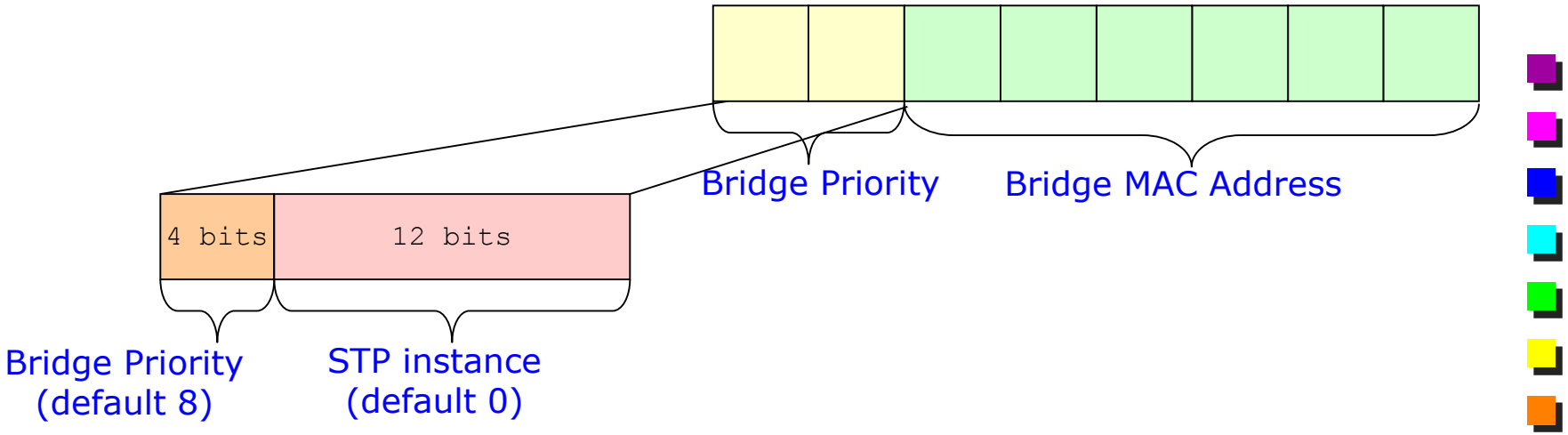


IEEE 802.1t

- 802.1t (2001): Technical and Editorial corrections for 802.1D-1998
- Defines new STP parameters adopted by 802.1w and 802.1s standards
  - Changed the format of Bridge Identifier
  - Extended Port Path Cost in a range from 1 to 200.000.000

IEEE 802.1t: new BridgeID

- New Bridge Identifier
  - Bridge priority partitioned in Bridge Priority + STP instance
  - Necessity to configure multiple STP (e.g. PVST+ and MST) within the same physical network (e.g., for different VLANs)
  - Offer the possibility to create different root bridges within each STP instance
    - Multiple trees for load balancing purposes



IEEE 802.1t: new Path Costs

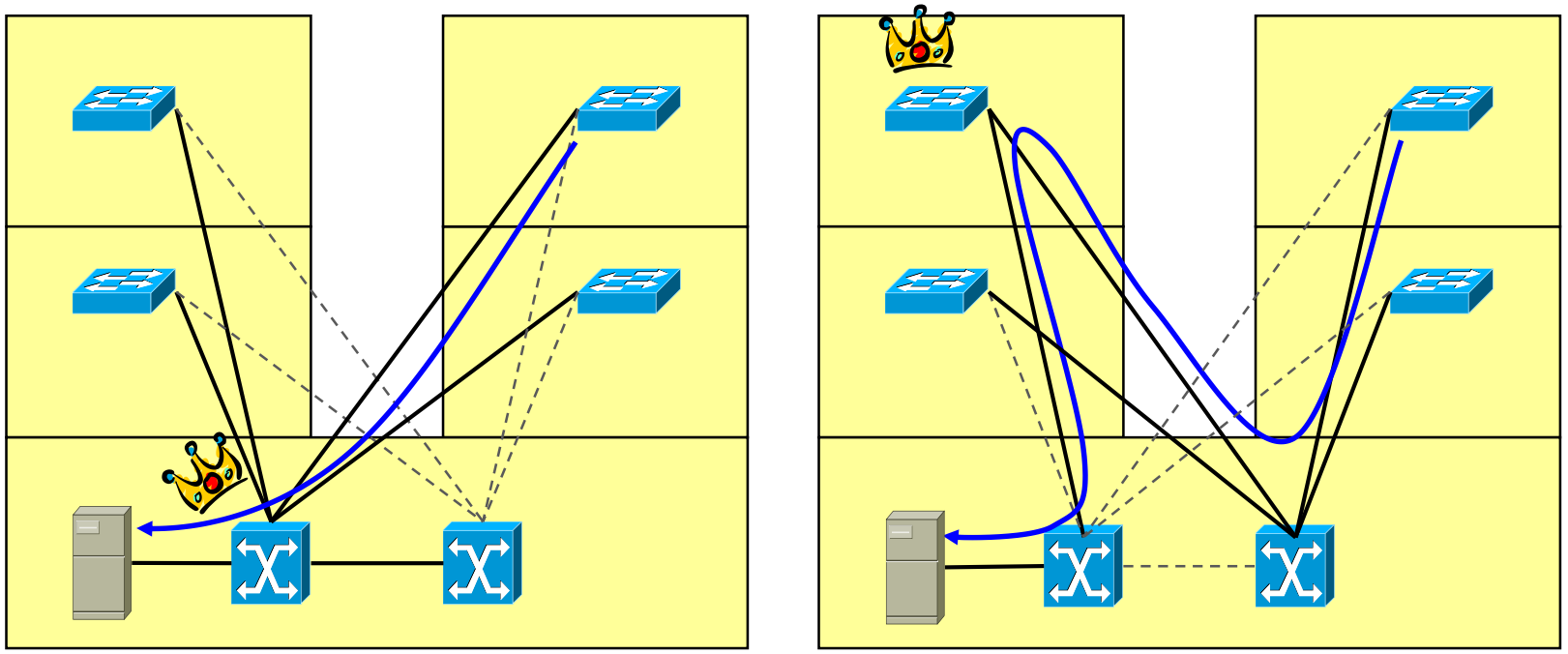
Port speed	Recommended Value	Recommended range values
<= 100Kb/s	200.000.000	20.000.000 – 200.000.000
1 Mb/s	20.000.000	2.000.000 – 20.000.000
10 Mb/s	2.000.000	200.000 – 2.000.000
100 Mb/s	200.000	20.000 – 200.000
1 Gb/s	20000	2.000 – 20.000
10 Gb/s	2000	200 – 20000
100 Gb/s	200	20-2000
1 Tb/s	20	2-200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

Pay attention to path costs!

- Old and new bridges may have different costs for the same link
- This may cause loops in the network

→ *check carefully the link costs in your network!*

Designing an STP network: BridgeID (1)



Note: dashed lines are present for better comprehension. However, in practice we disable ports, not links.

Designing an STP network: BridgeID (2)

- Lesson learned: in practice, bridges are not equivalent when serving as root bridge
  - The location of the root bridge has heavy impact on the network operations
  - The root bridge will probably receive a huge amount of data traffic
    - Traffic from one side of the network to the other side has to cross the Root Bridge

Designing an STP network: BridgeID (3)

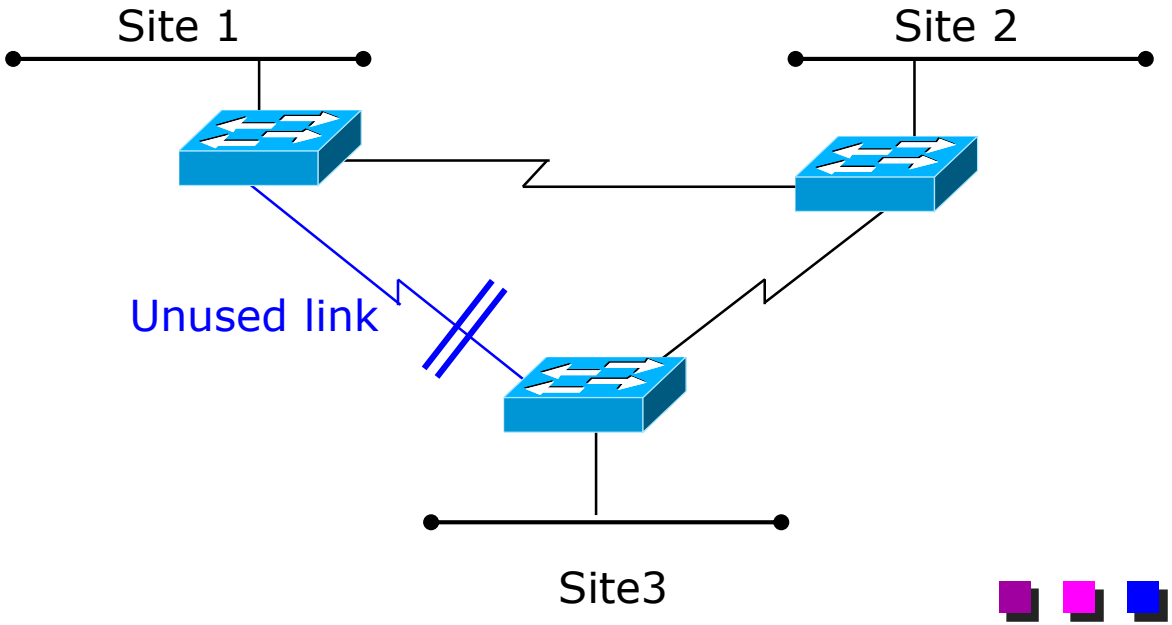
- Suggestions:
  - Customize the Bridge Priority field in order to force a specific bridge to become Root Bridge
  - Get prepared for any trouble the Root Bridge may have, and define which should be the next root bridge in case the first one fails
    - Backup Root Bridge
  - Customize the Bridge Priority field of the "backup root bridge" in order to force that bridge to become Root Bridge in case the first one fails
- You may have also to change some cost links in order to force STP to select the paths you want

Designing an STP network: dimensioning

- Key points
  - Root bridge must be powerful enough (aggregate throughput)
  - Bandwidth of the Root Bridge ports must be appropriate
  - Network around Root Bridge must be stable (in order to avoid a complete recalculation of the STP)
  - Servers / Data Centers should be placed near Root Bridge in order to reduce the latency of data communication

Spanning Tree over WAN links

- Problem: a single tree for the entire network
  - Links not used (while we paid for them)
  - We need to resize the other link in order to sustain the other cross-site traffic
- Solution: use a different tree for each sender
  - Requires an L3 device or VLANs with multiple spanning trees

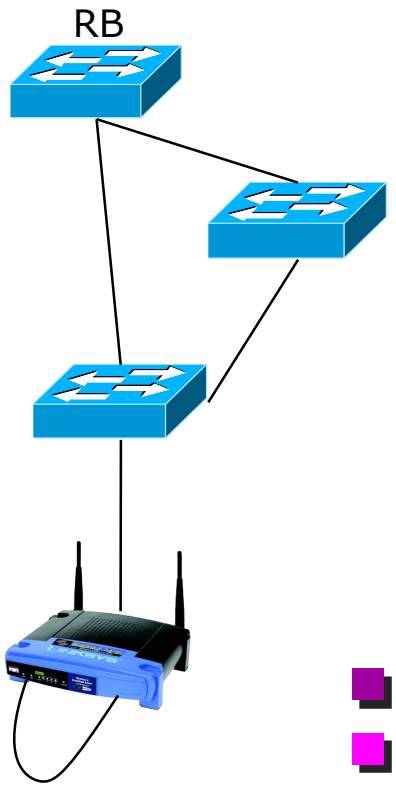


Switches and broadcast storm control

- Almost all the professional switches have some form of broadcast storm control
  - Rate limit on broadcast storm
    - Limit the amount of broadcast you have into the network
  - Usually a bad idea
    - Some broadcast will be dropped by the switch
      - We cannot distinguish between the frames that are currently looping and legitimate broadcast frames sent by stations
    - Result: the network has intermittent problems
      - Some traffic arrives to the destination, some other does not
      - The network administrator has not glue of what is going on, because the traffic appears to be "regular"
      - With "standard" broadcast storm, the effect is clear and the network administrator can immediately diagnose a problem on the ST algorithm

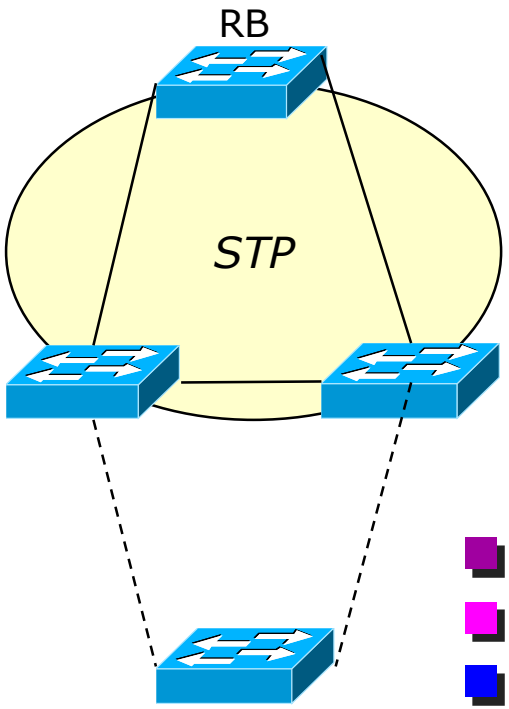
Spanning Tree and home switches

- Example of a real case
  - STP correctly configured and working on the corporate network
  - New home switch added in the access side
  - A port is connected back to the same switch
- Result: broadcast storm on the entire network!
- Lesson learned: a single switch without STP could break the entire corporate network



Spanning Tree and switches without STP

- Example of a real case
  - STP correctly configured and working on the corporate network
  - New switch (with STP disabled) added to the network, with redundant connection.
- Result: Loop!
- Lesson learned again: a single switch without STP can break the entire corporate network



Protecting the STP network at the edge

- BPDU Guard
  - Enabled on edge ports: if a BPDU is received on that port, the port goes in “error disable” state
  - Protects from
    - Accidental connection of a switch on edge ports (which may claim to be the root)
    - Loops on edge ports (the BPDU sent by port A is received on port B, which is protected by Port Guard and hence it will be disabled)
      - See example on previous slide
- BPDU filter
  - Enabled on ports that connect two different STP domains
    - E.g., two providers with their own STP in the same data center
  - It disables sending/receiving BPDUs on that port
  - The port does not transition in “error disable” state if a BPDU is received on that port

Conclusions

- Old, but still effective protocol
- Available (and compatible) on most switches
- Limitations
  - Convergence time (what about a network that transports VoIP traffic?)
    - New protocols in case a faster convergence is needed
  - No support for multipath