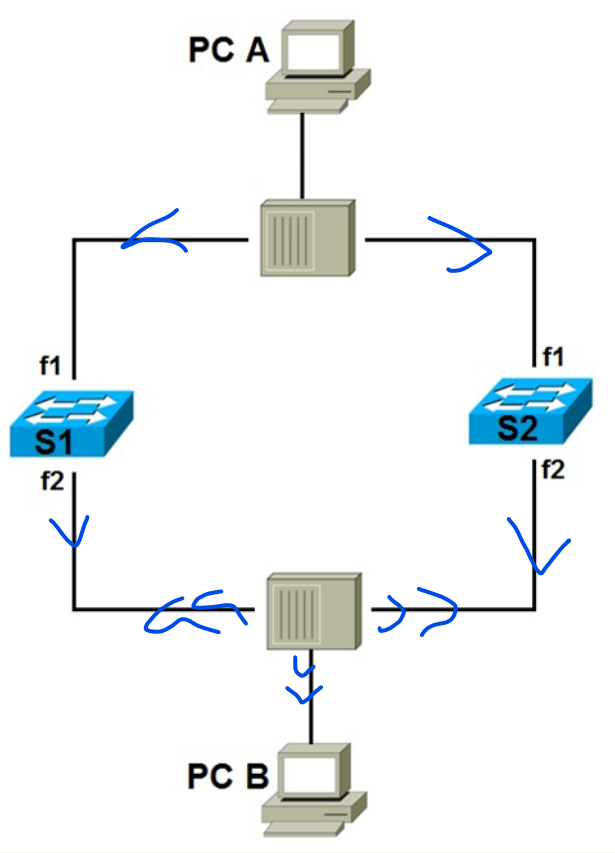
STP
Spanning-Tree Protocol

CyberQuince

The "Bridge Loop"

Let's consider the following scenario:

- **PC A** sends an ARP packet.
- The **upper hub** forwards that packet to all ports, except to the port where the packet came from.
- **S1 and S2** receive the packet at ports **f1** and save **PC A**'s MAC address.
- **S1 and S2** forward the packets through their **f2** ports.
- The **lower hub** receives two packets from two sides. **It sends the packet** it received on the left-side interface (from S1) to it's right side (**to S2**) **+ to PC B**. The same happens for the packet it receives from S2.
- **S1 and S2** receive a packet who's source MAC address is still **PC A**'s MAC address! **S1 and S2** both save this MAC address on their **f2** interfaces.



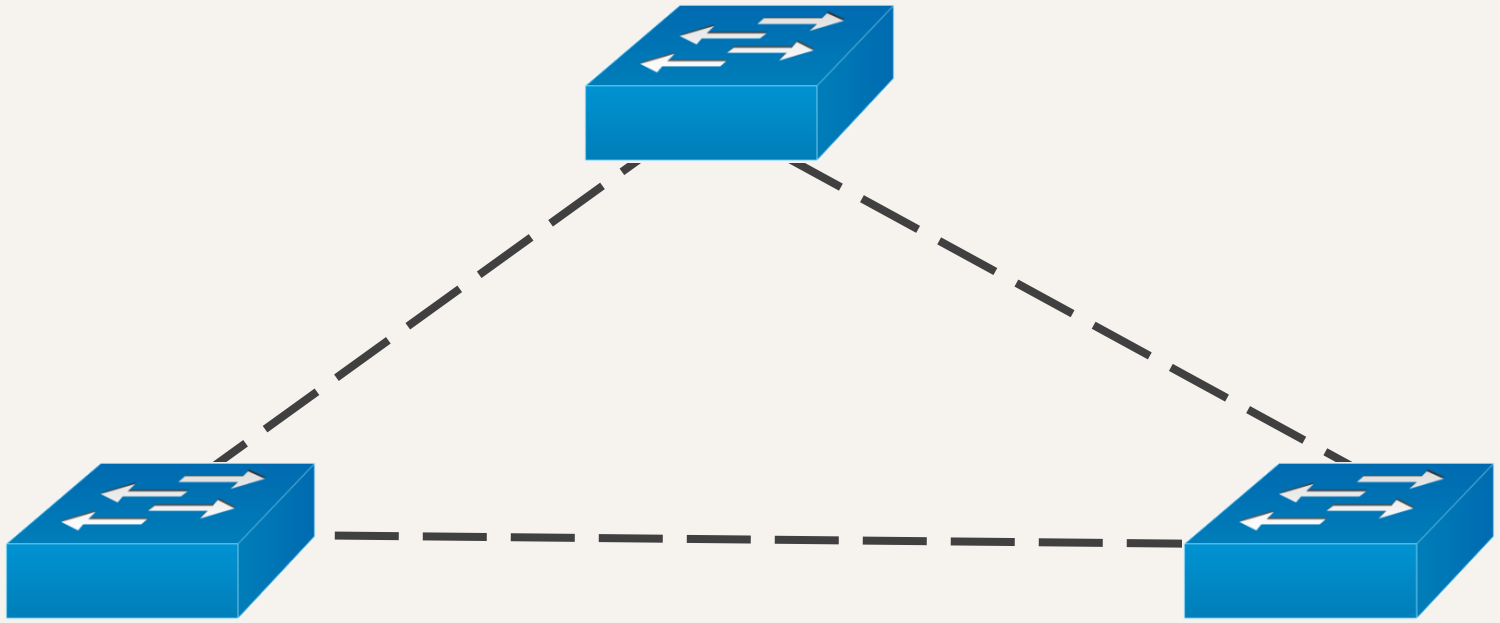
STP — Spanning Tree Protocol

STP (*Spanning Tree Protocol* – IEEE 801.1D) "scans" the network to find all links, making sure that there are no loops – by "switching off" all redundant links.

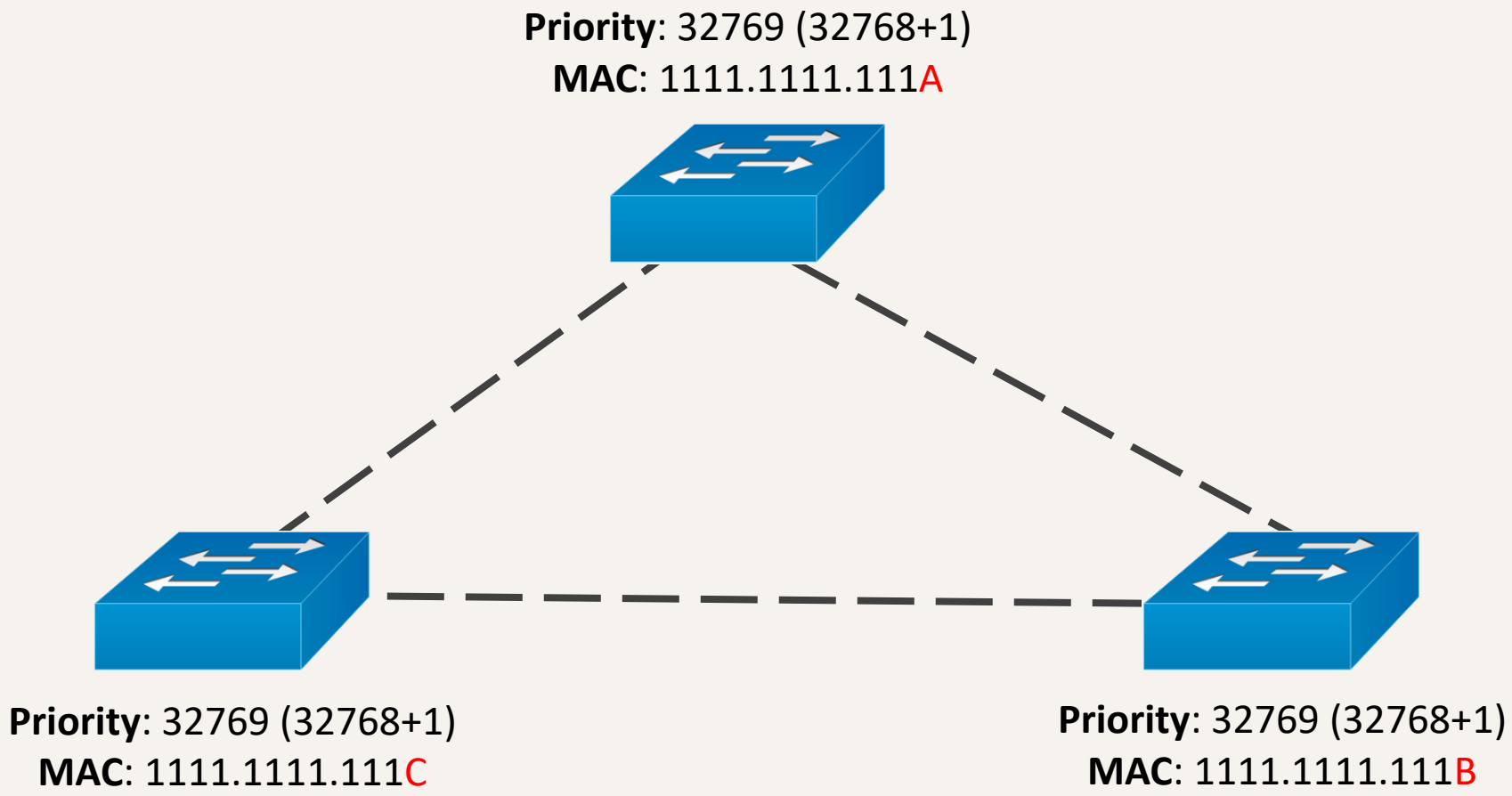
It first uses the **STA** (*Spanning Tree Algorithm*) to figure out the topology, and then identifies the links that form loops. Those links are blocked one by one, removing loops.

In order for this to function properly, all switches must cooperate: one of the switches is chosen to be ***a reference point – the "Root Bridge"*** and it becomes the root of the tree.

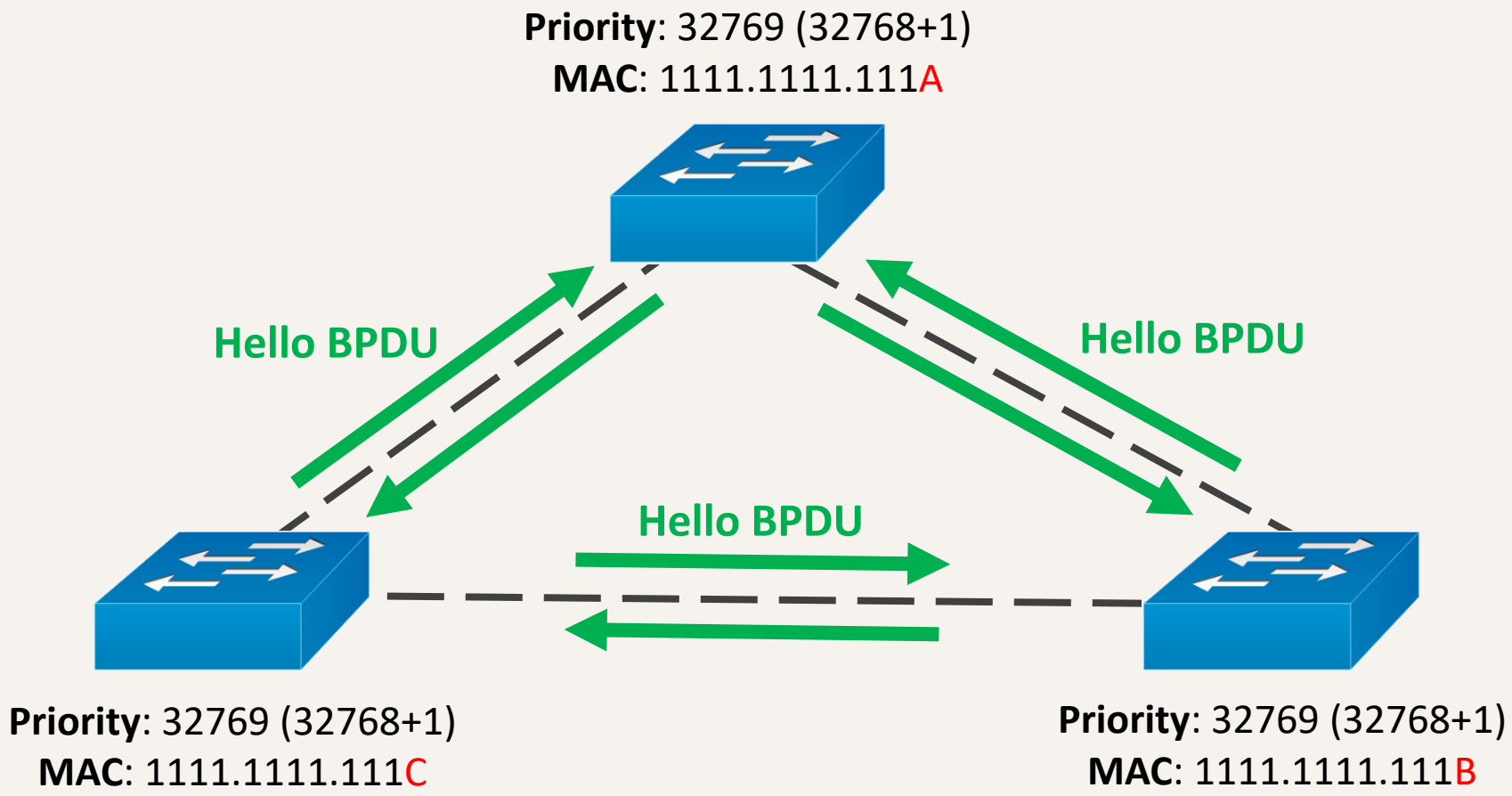
STP — An Example



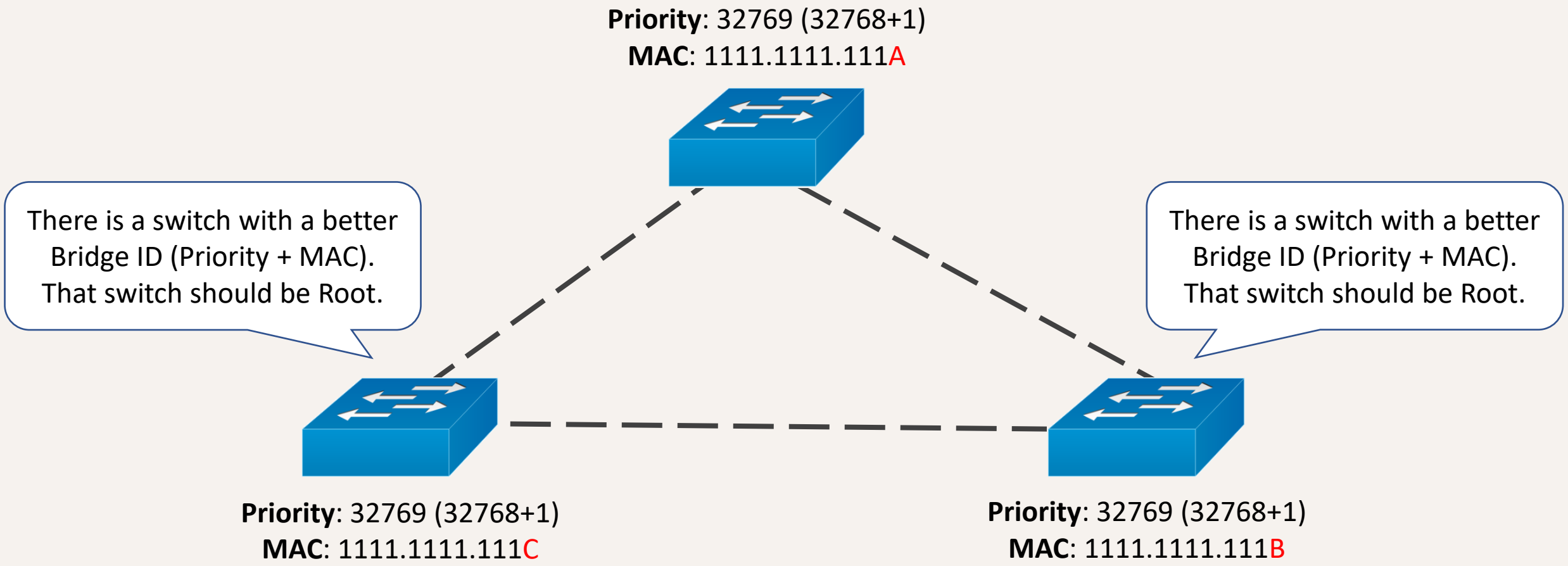
STP — An Example



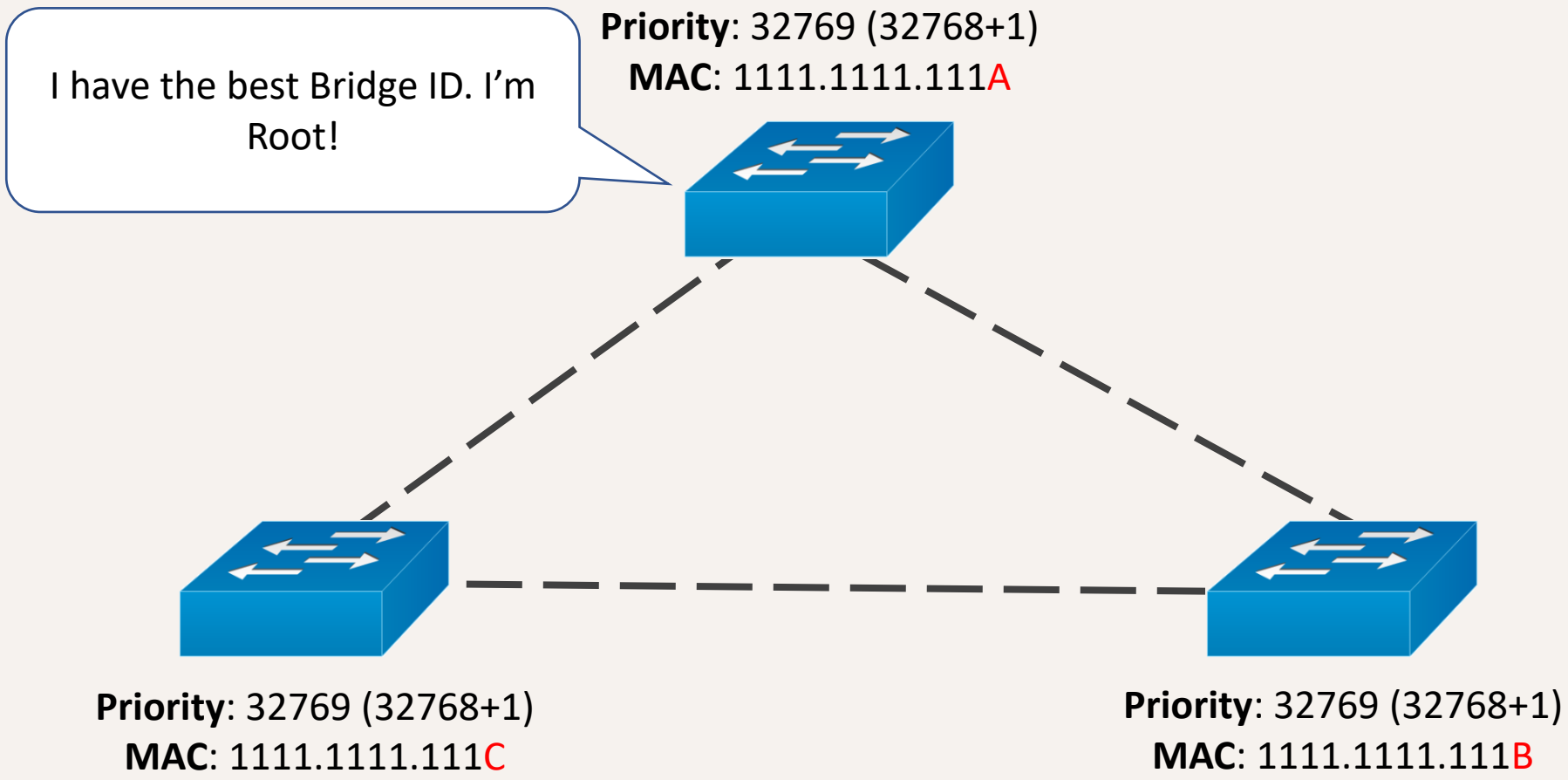
STP — An Example



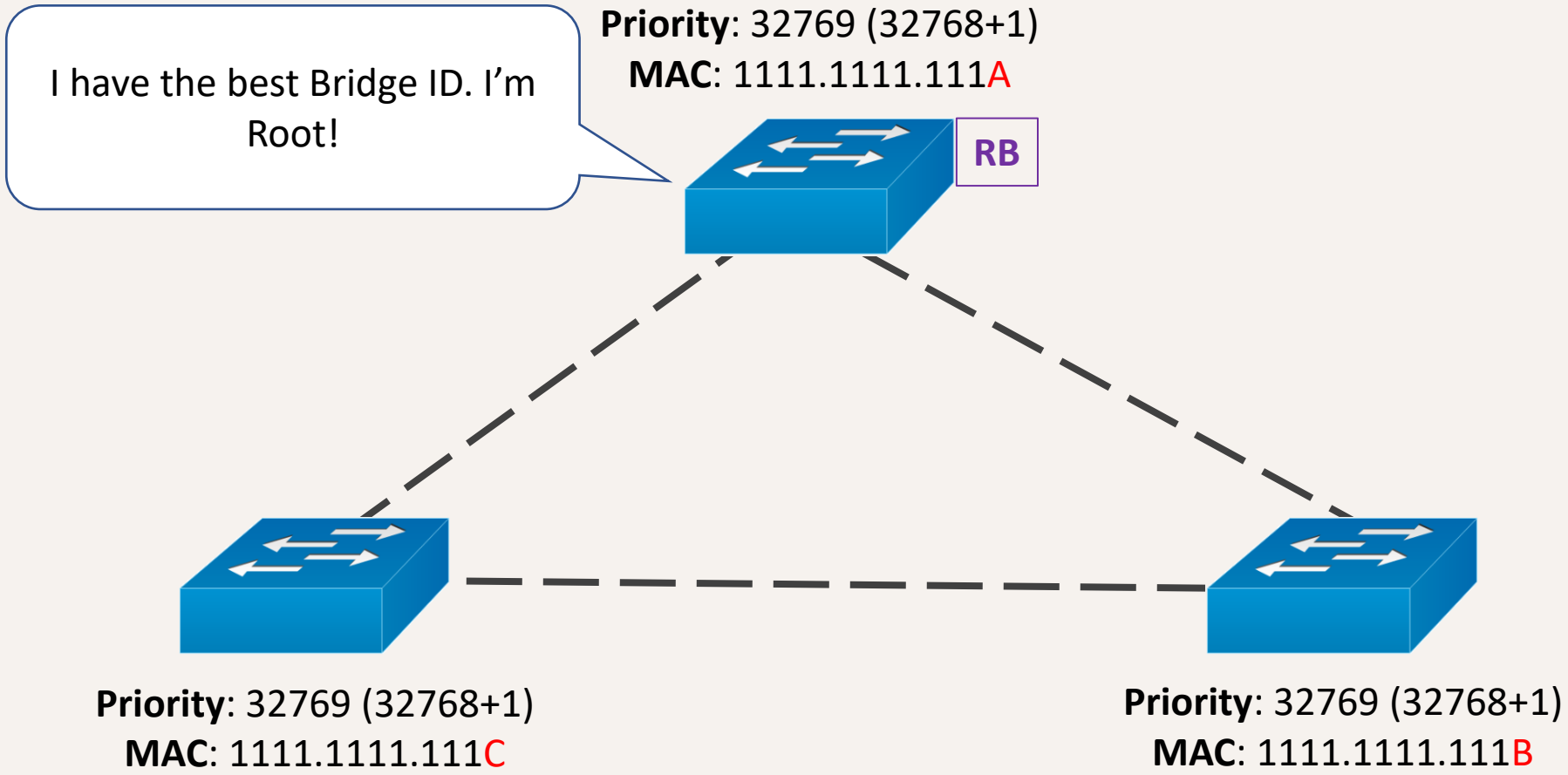
STP — An Example



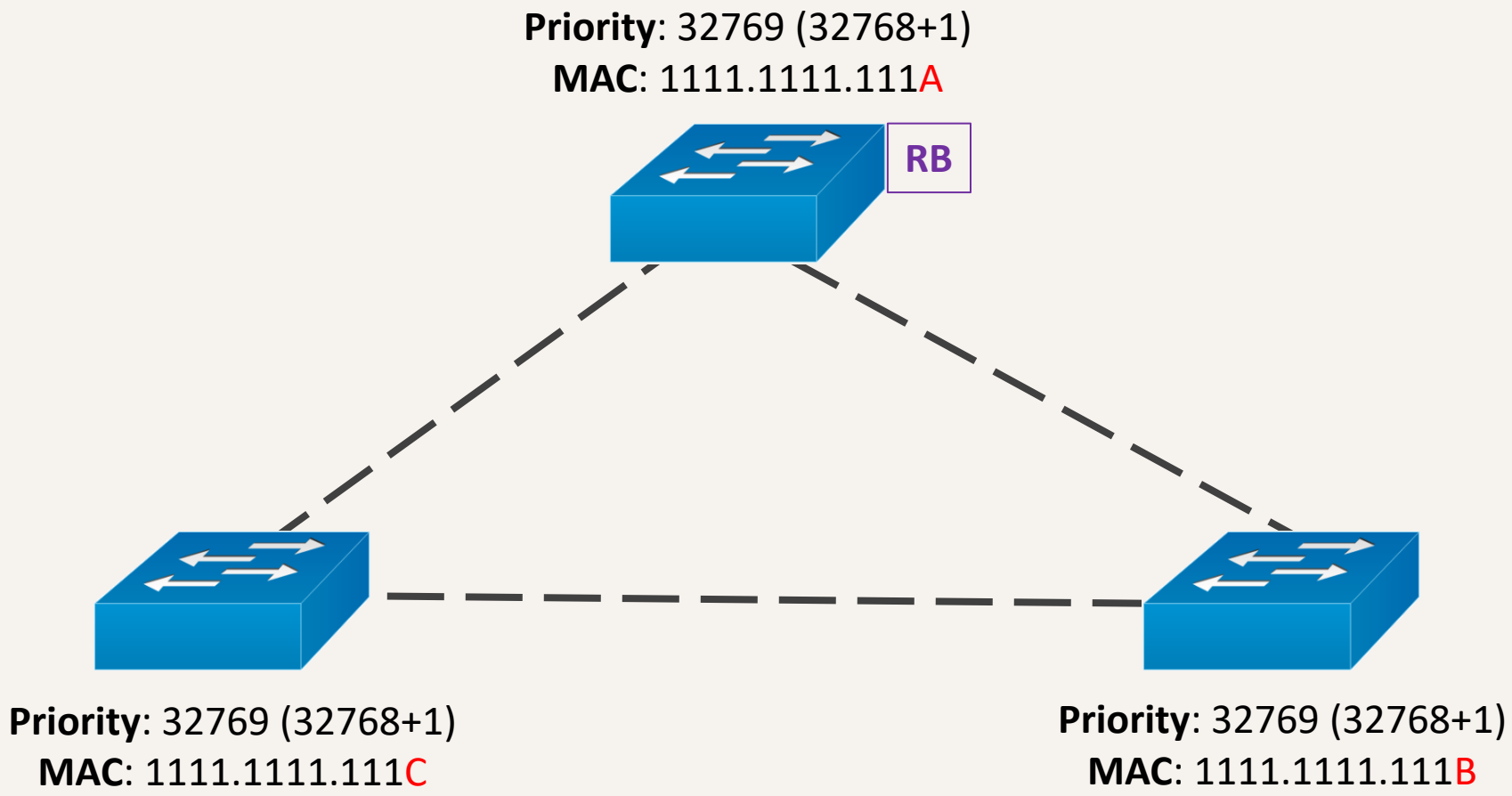
STP — An Example



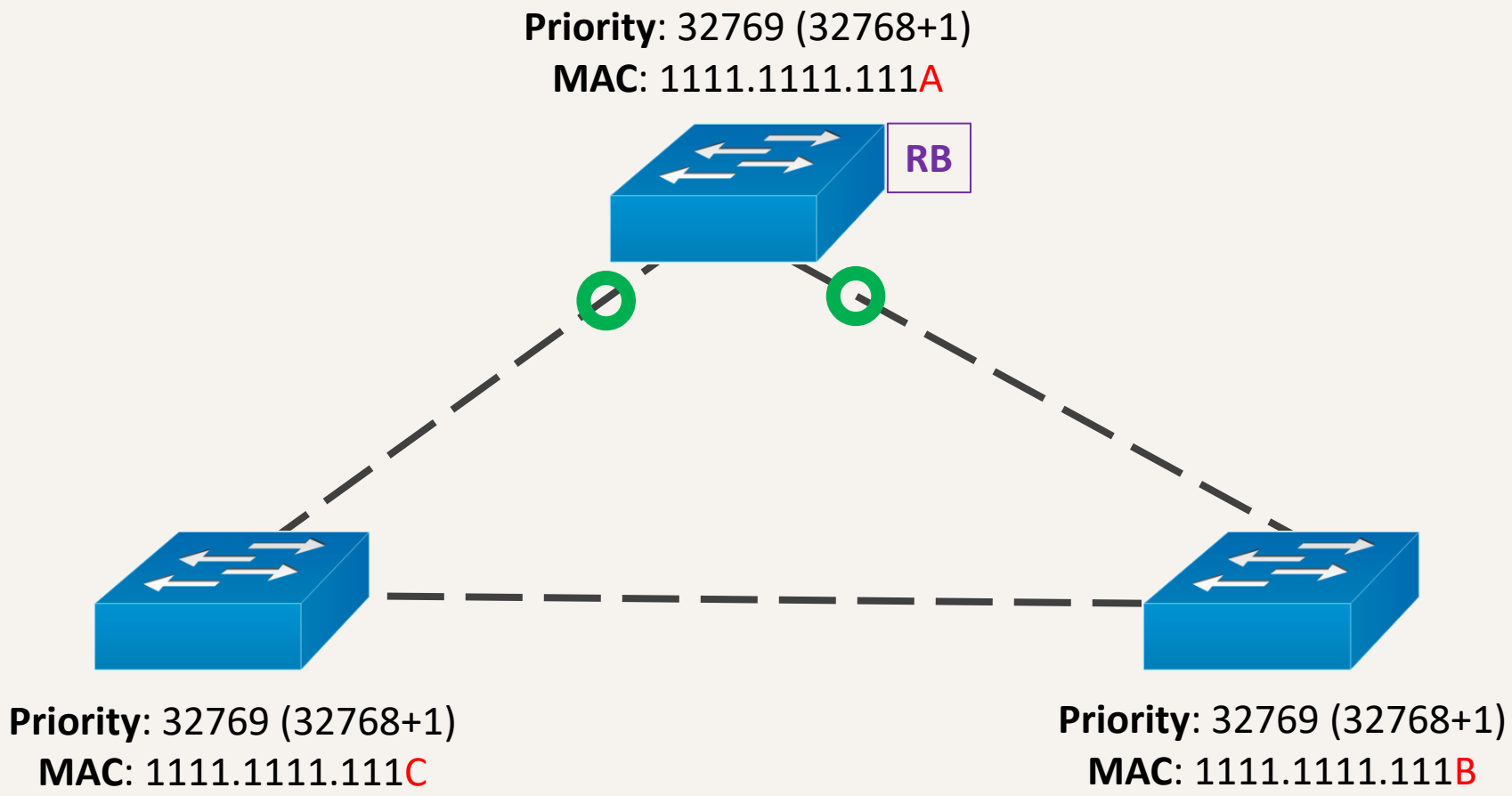
STP — An Example



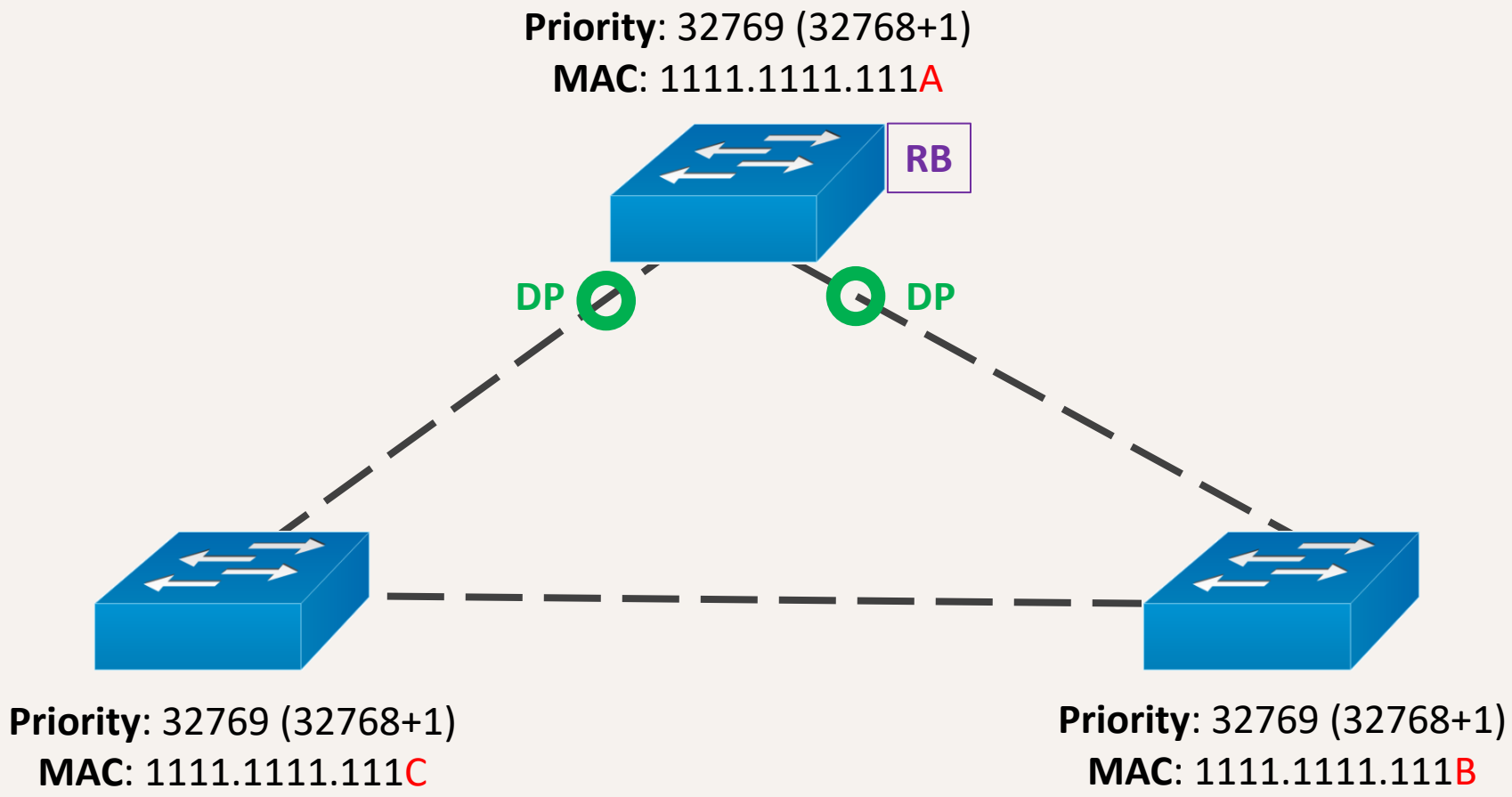
STP — An Example



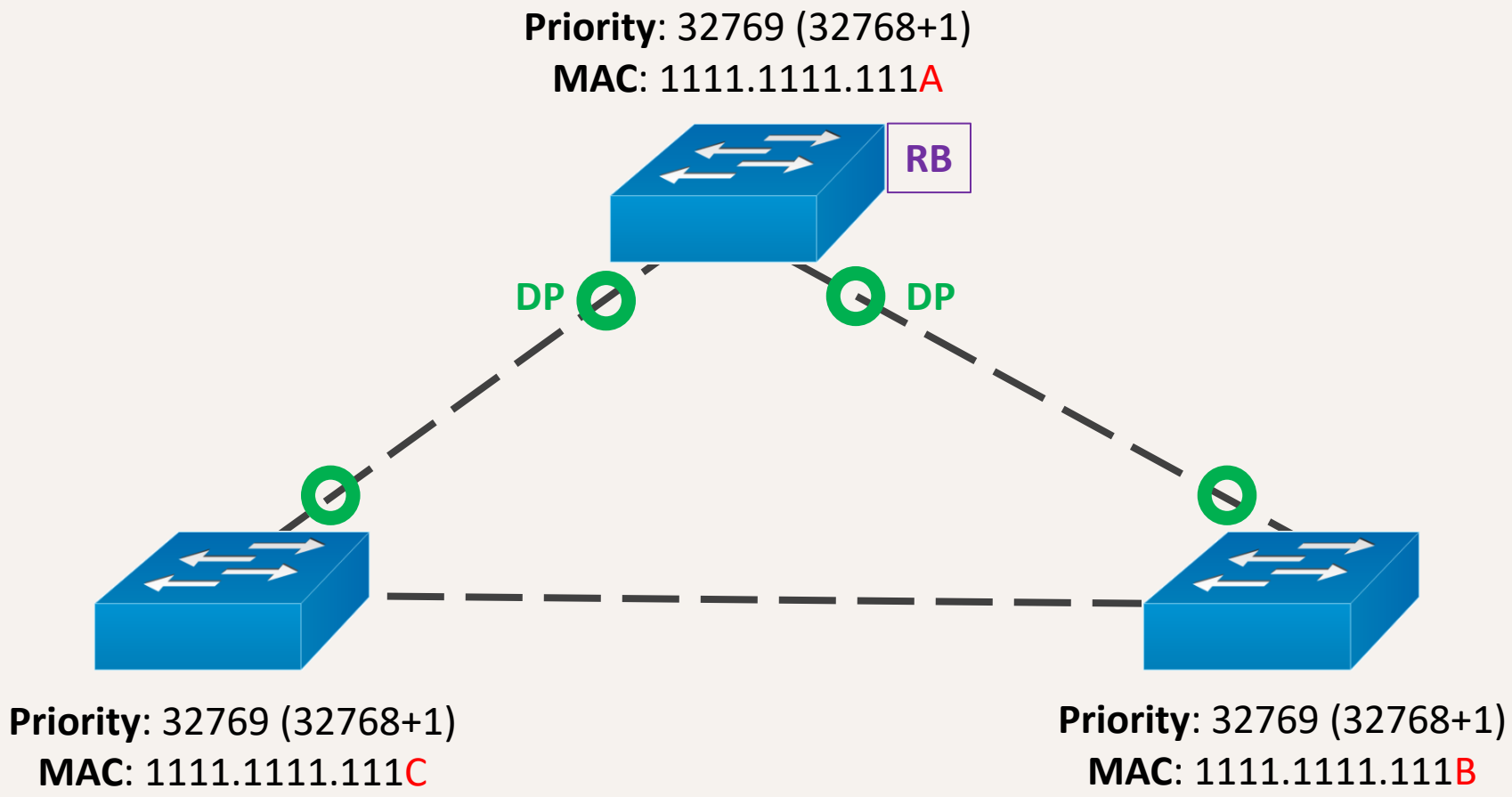
STP — An Example



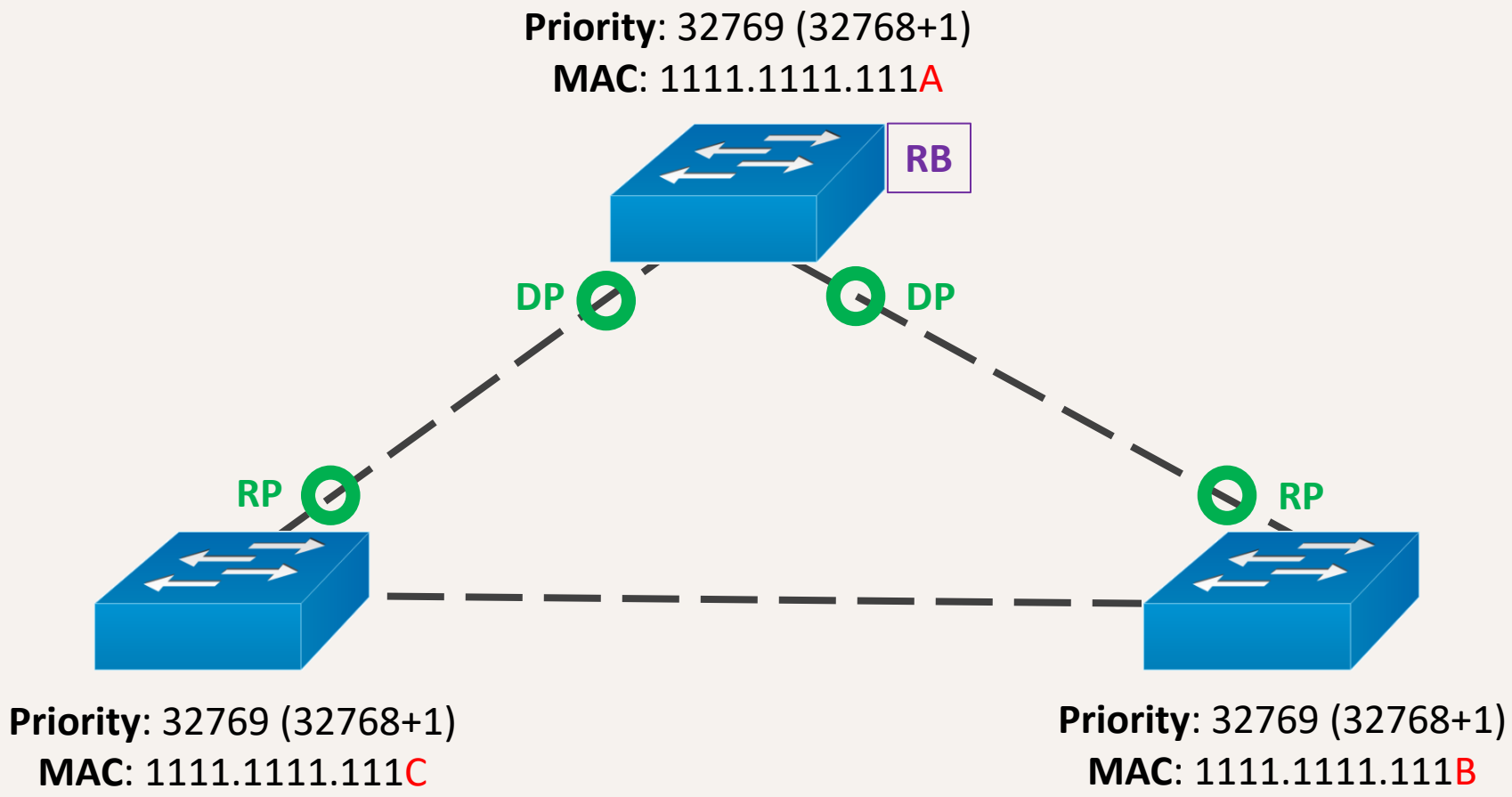
STP — An Example



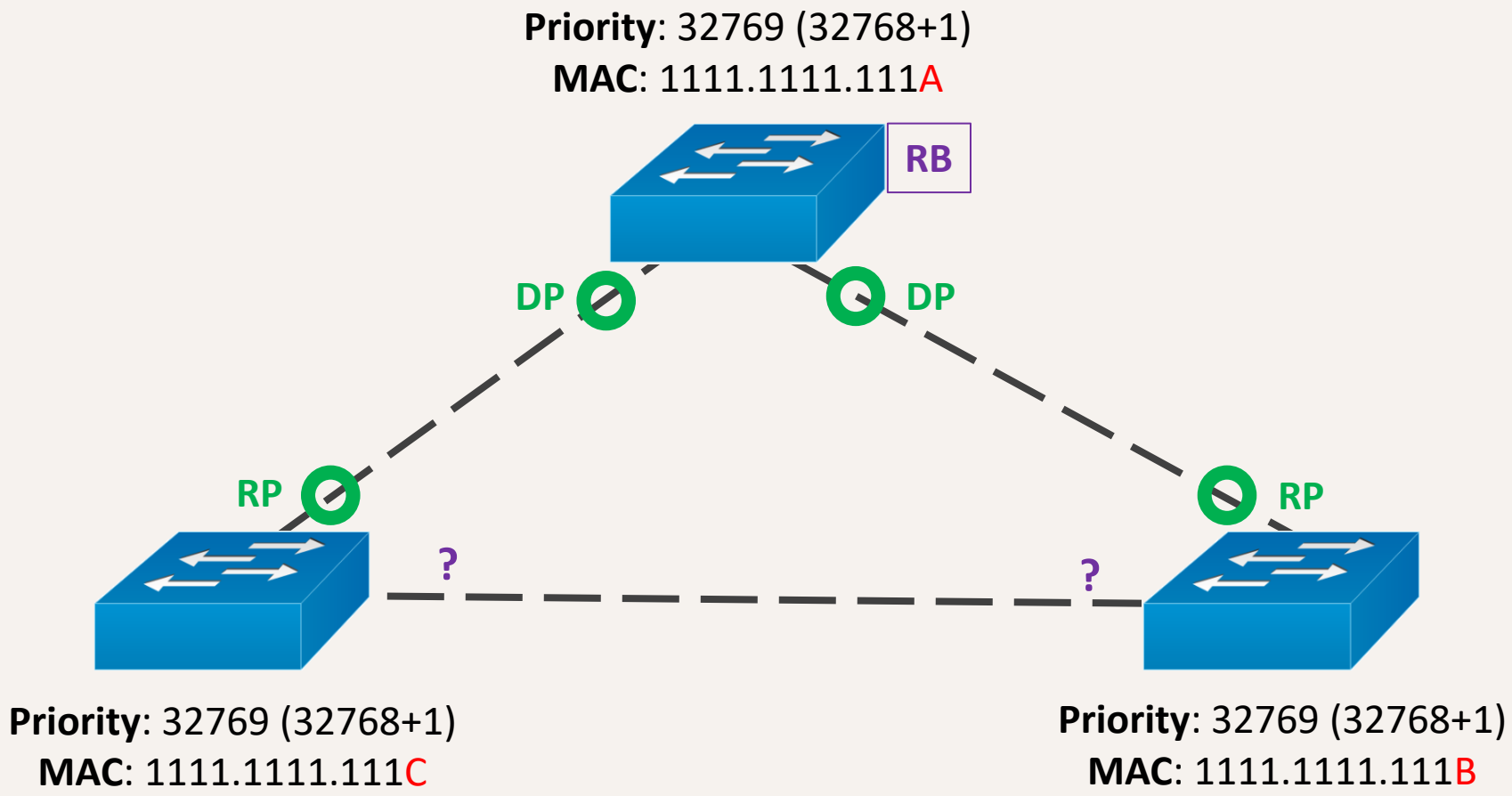
STP — An Example



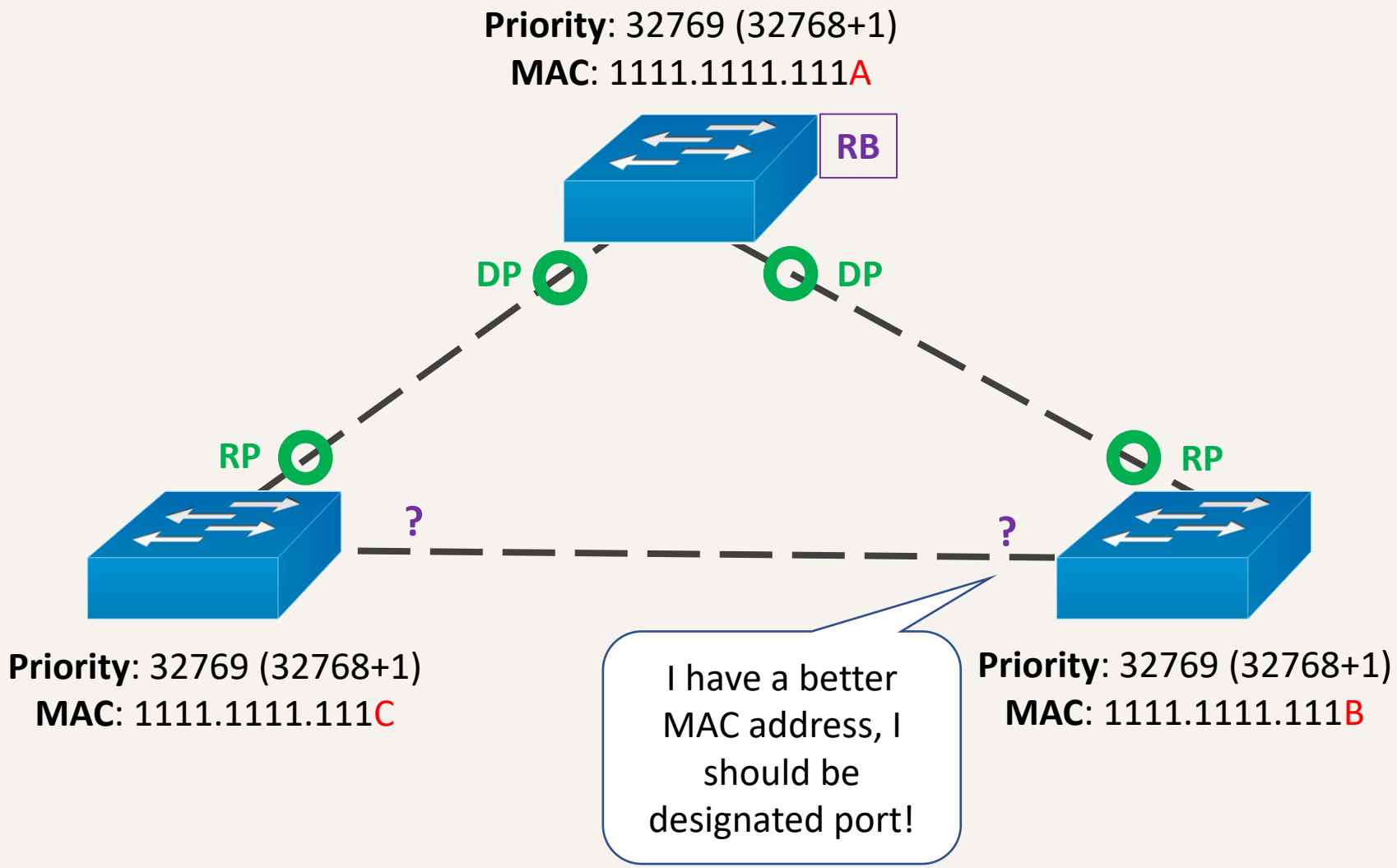
STP — An Example



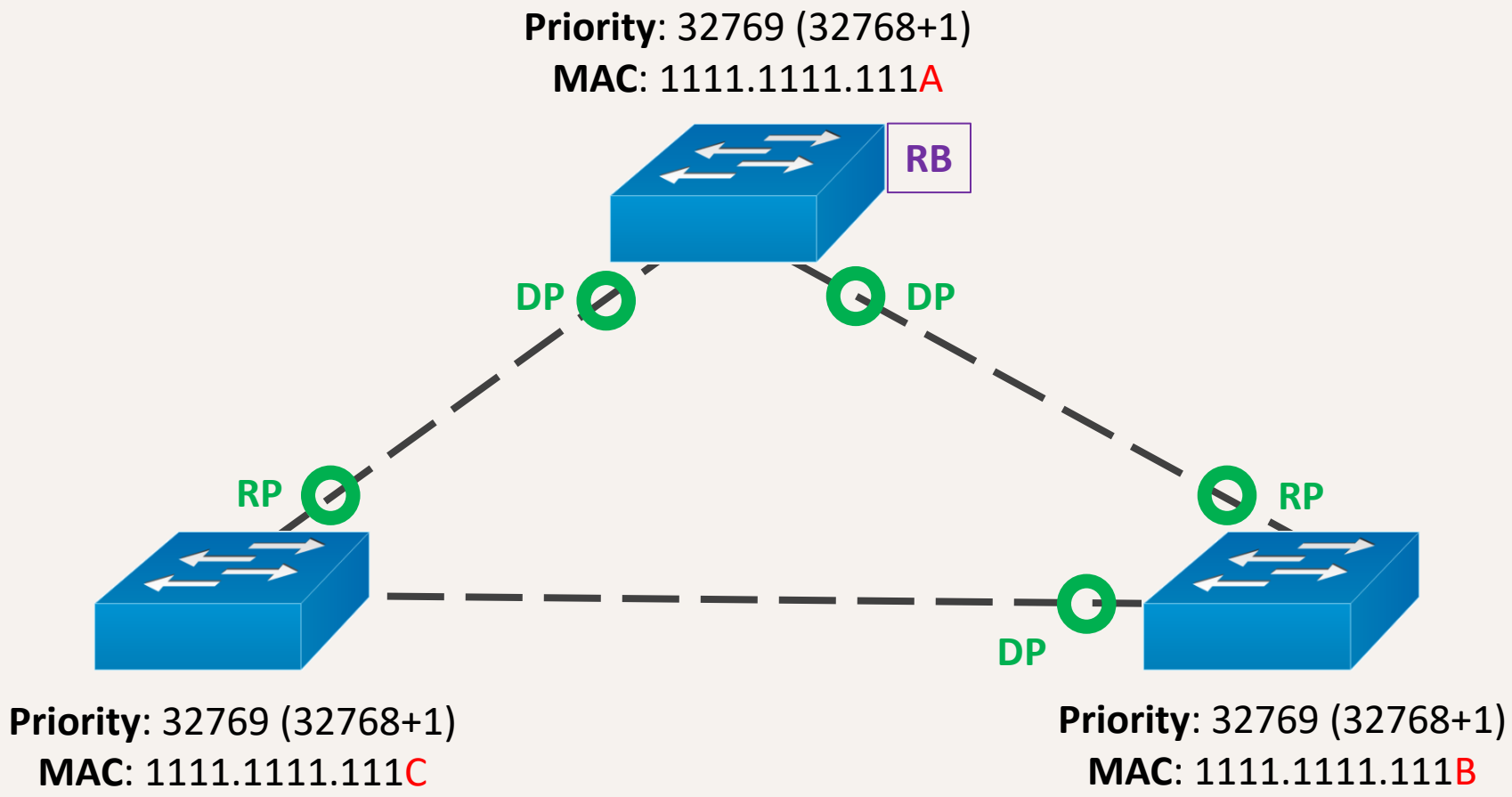
STP — An Example



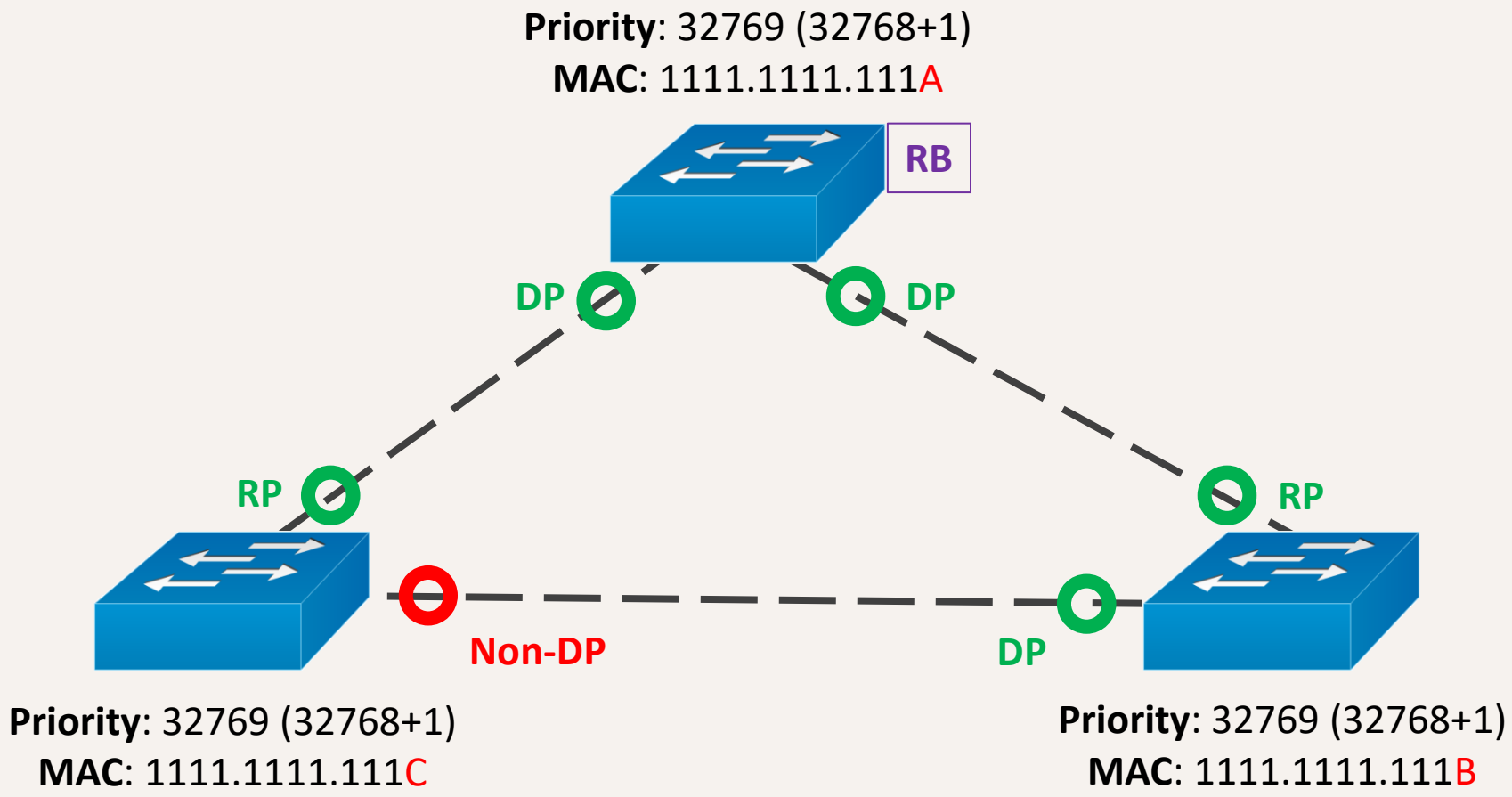
STP — An Example



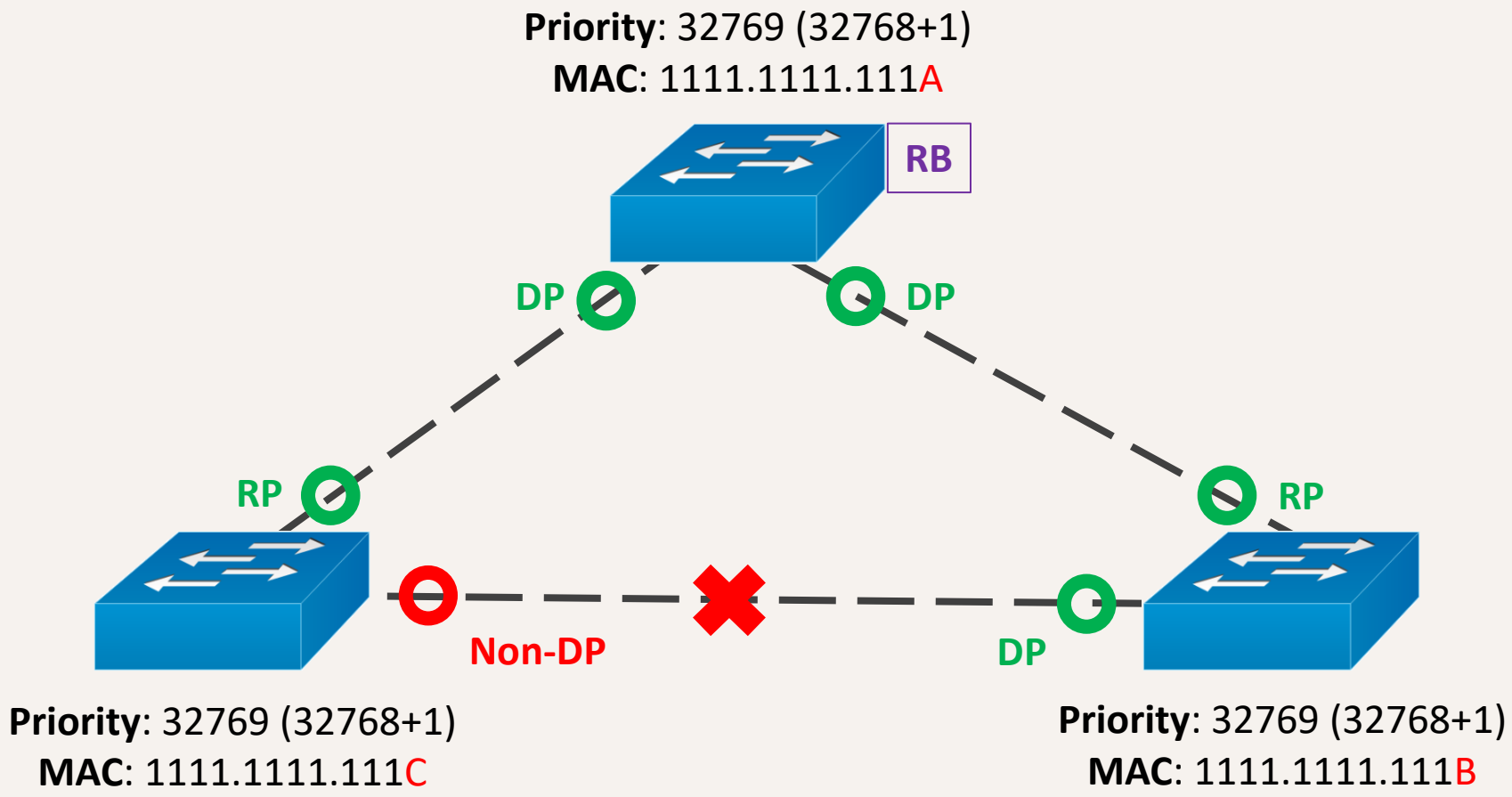
STP — An Example



STP — An Example



STP — An Example



STP — Why Configure Manually?

When you boot a switch, STP is on by default. Still, you should configure it manually. Here's why:

- If you let STP create the "tree" on its own, it might pick an older switch to be Root, which could end up getting much more traffic than the other, newer switches.

- Also, it's unaware of any specific VLAN configurations on a link - if one trunk link can carry all VLANs, but another one can't, and STP blocks the first one – you're in trouble.

As an administrator, you can pick the Root switch (bridge) manually, which will greatly influence the resulting tree.

STP — How to Configure?

```
Switch(config)# spanning-tree vlan VLAN_ID root primary
```

STP — How to Configure?

```
Switch(config)# spanning-tree vlan VLAN_ID root primary

Switch(config)# spanning-tree vlan VLAN_ID root secondary
```

STP — How to Configure?

```
Switch(config)# spanning-tree vlan VLAN_ID root primary

Switch(config)# spanning-tree vlan VLAN_ID root secondary

Switch(config)# spanning-tree vlan VLAN_ID priority PRIORITY
```

STP — How to Configure?

```
Switch(config)# spanning-tree vlan VLAN_ID root primary

Switch(config)# spanning-tree vlan VLAN_ID root secondary

Switch(config)# spanning-tree vlan VLAN_ID priority PRIORITY
```

** Default priority value is 32.768, so it's enough to set any value lower than that for given switch to become Root. Priority value must be an increment of 4096. Highest possible value is 61440.*

scenario: in case you changed all the priorities to 20,480, and You ran the first command on a random switch, it will know the least priorities because of PBDUs and will make it 16,xxx

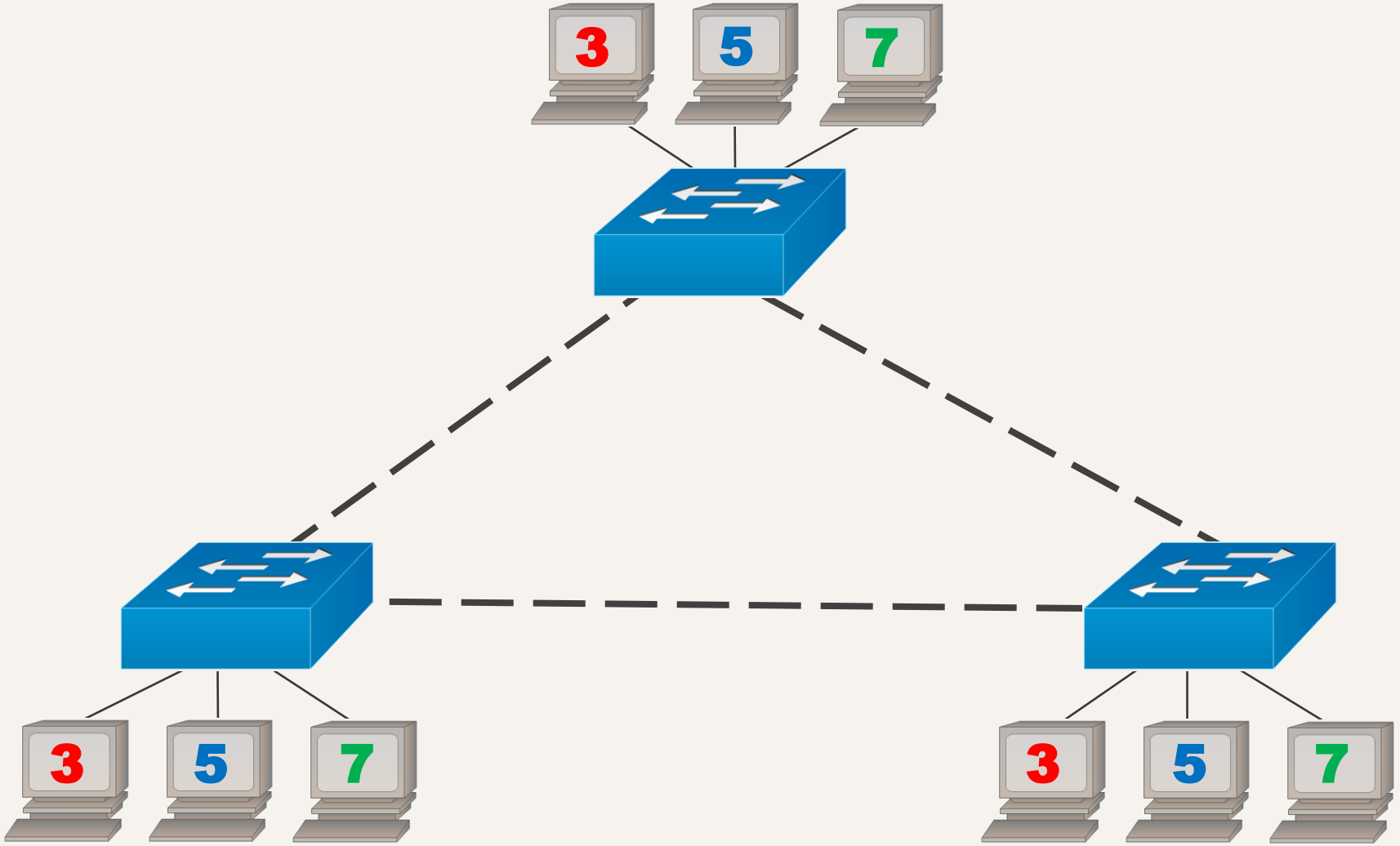
STP — Multiple VLANs

Spanning Tree Protocol was initially designed to work with bridges, and support only one LAN (or one VLAN).

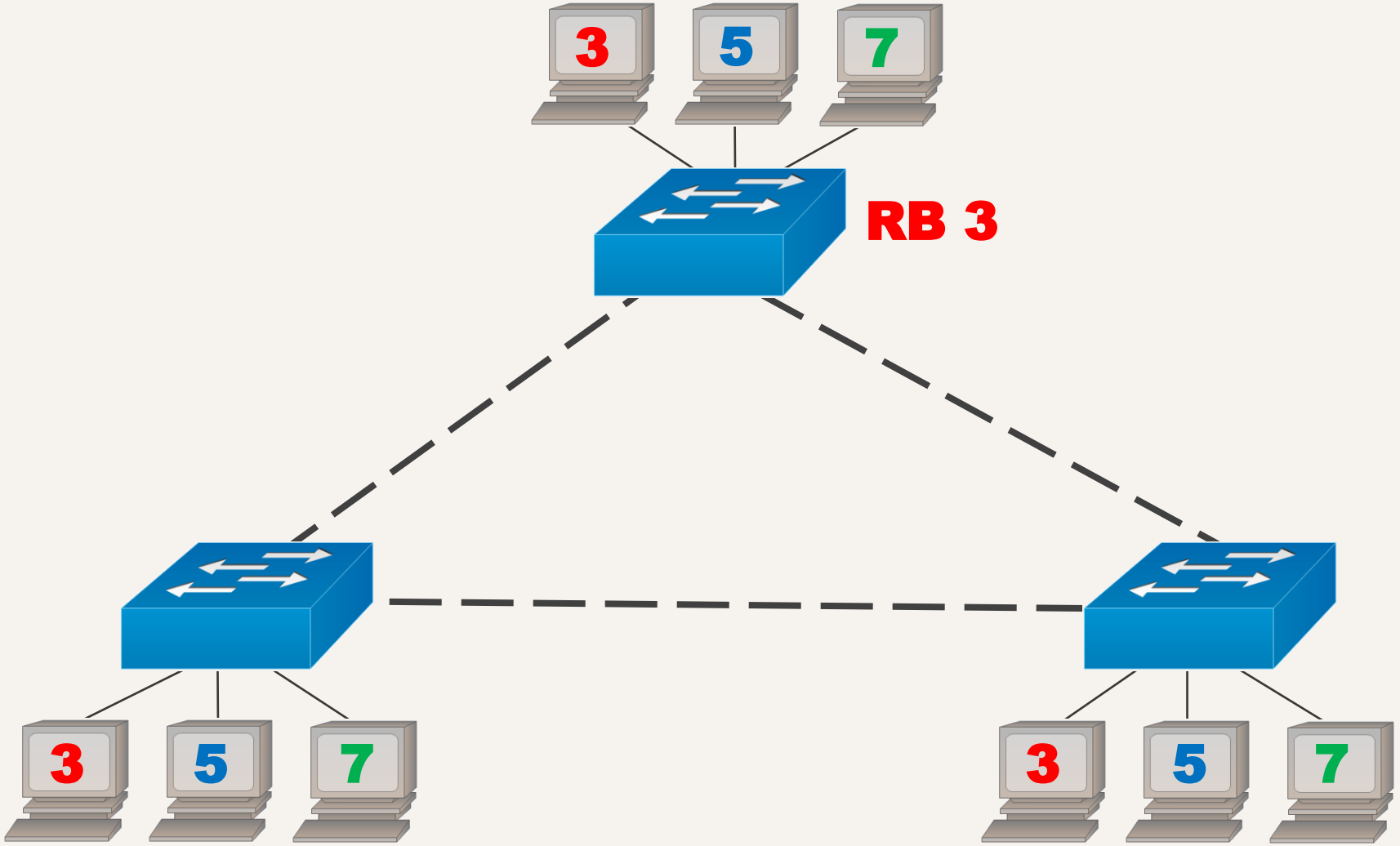
Even if you have multiple VLANs in your network, you can still only have a single instance of STP – all VLANs will share the same tree. But to best utilize your gear, you should have a different tree for each VLAN.

This is achieved by running a separate instance of STA per VLAN. It's called **PVST** – ***Per VLAN Spanning Tree***.

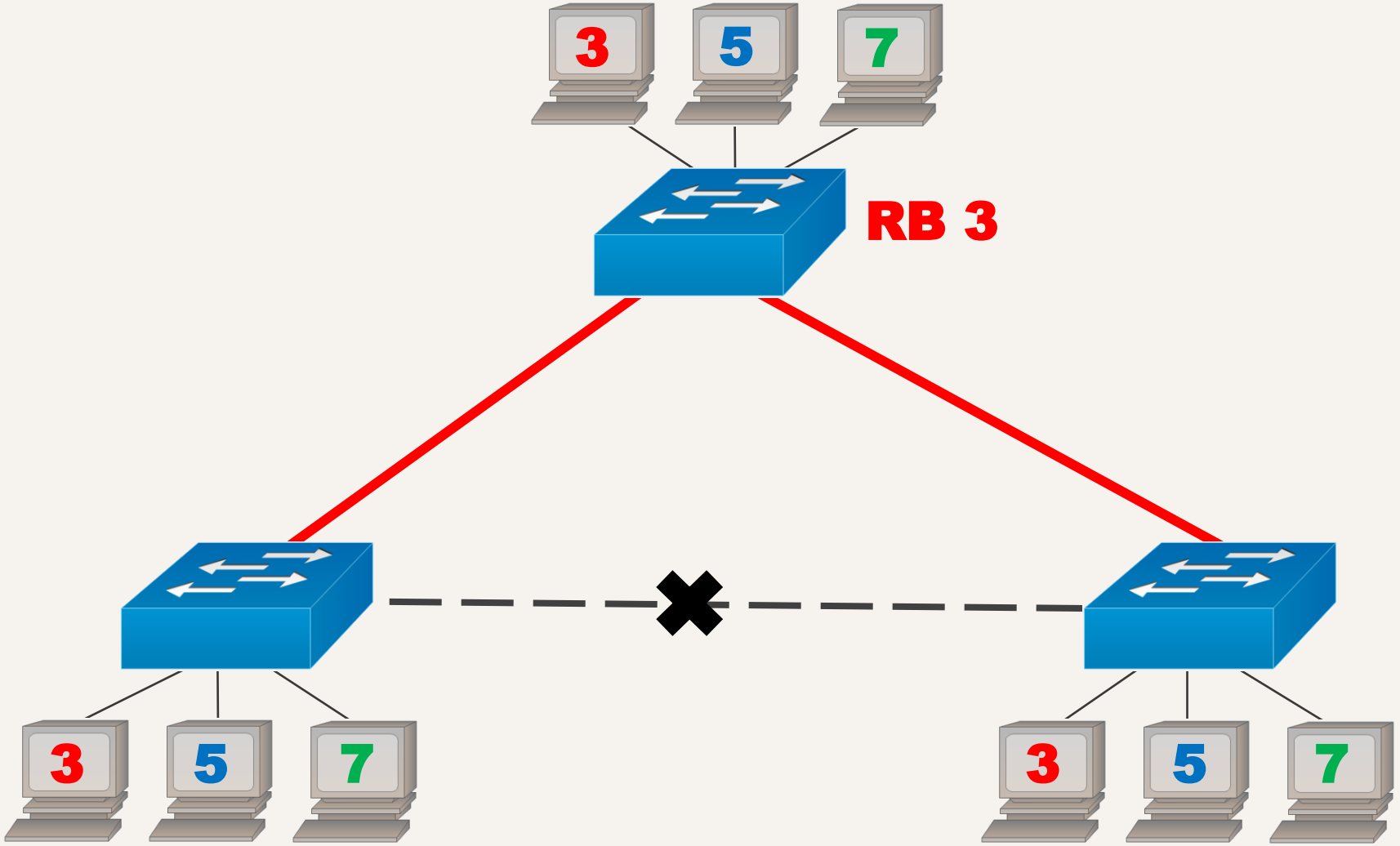
STP — Multiple VLANs



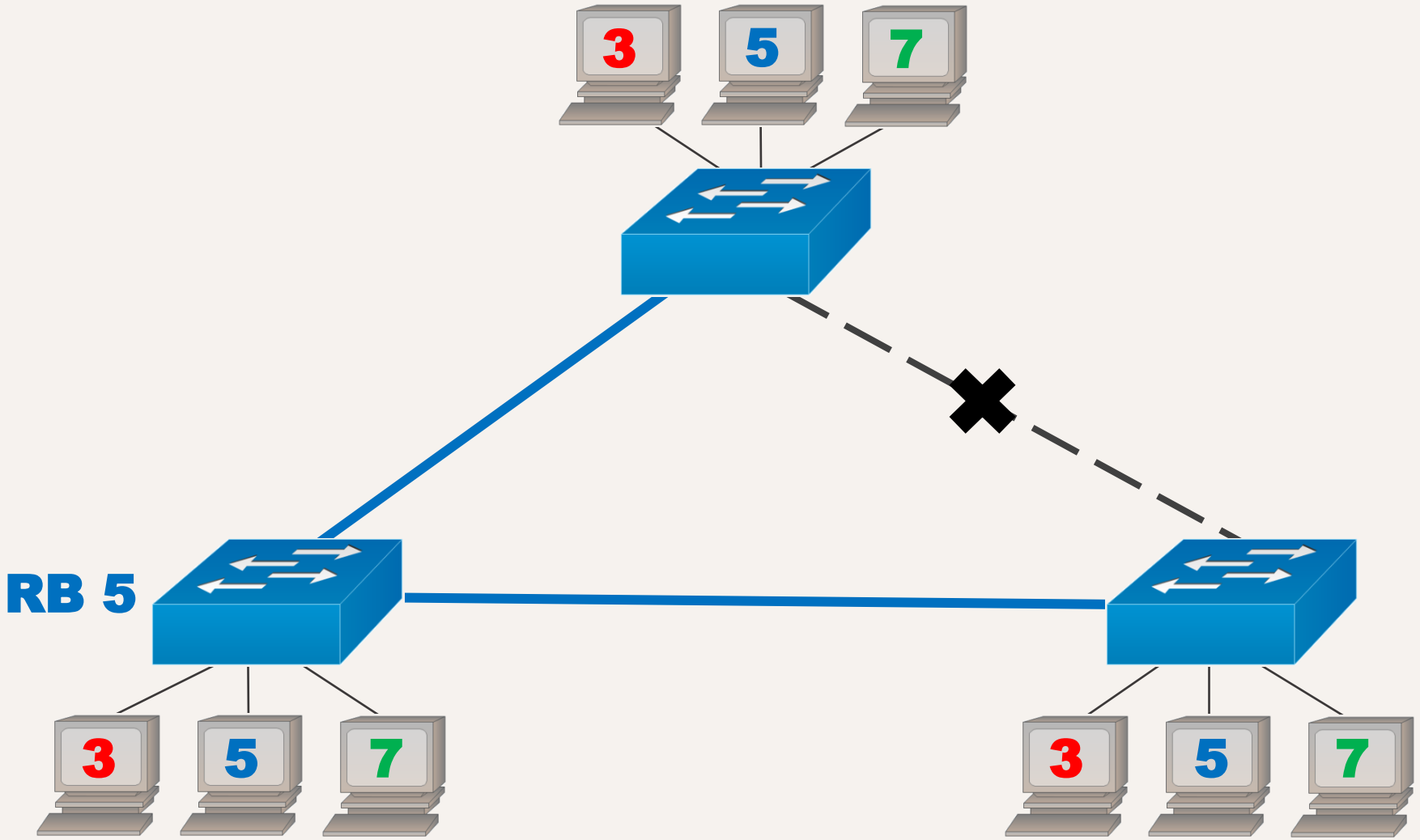
STP — Multiple VLANs



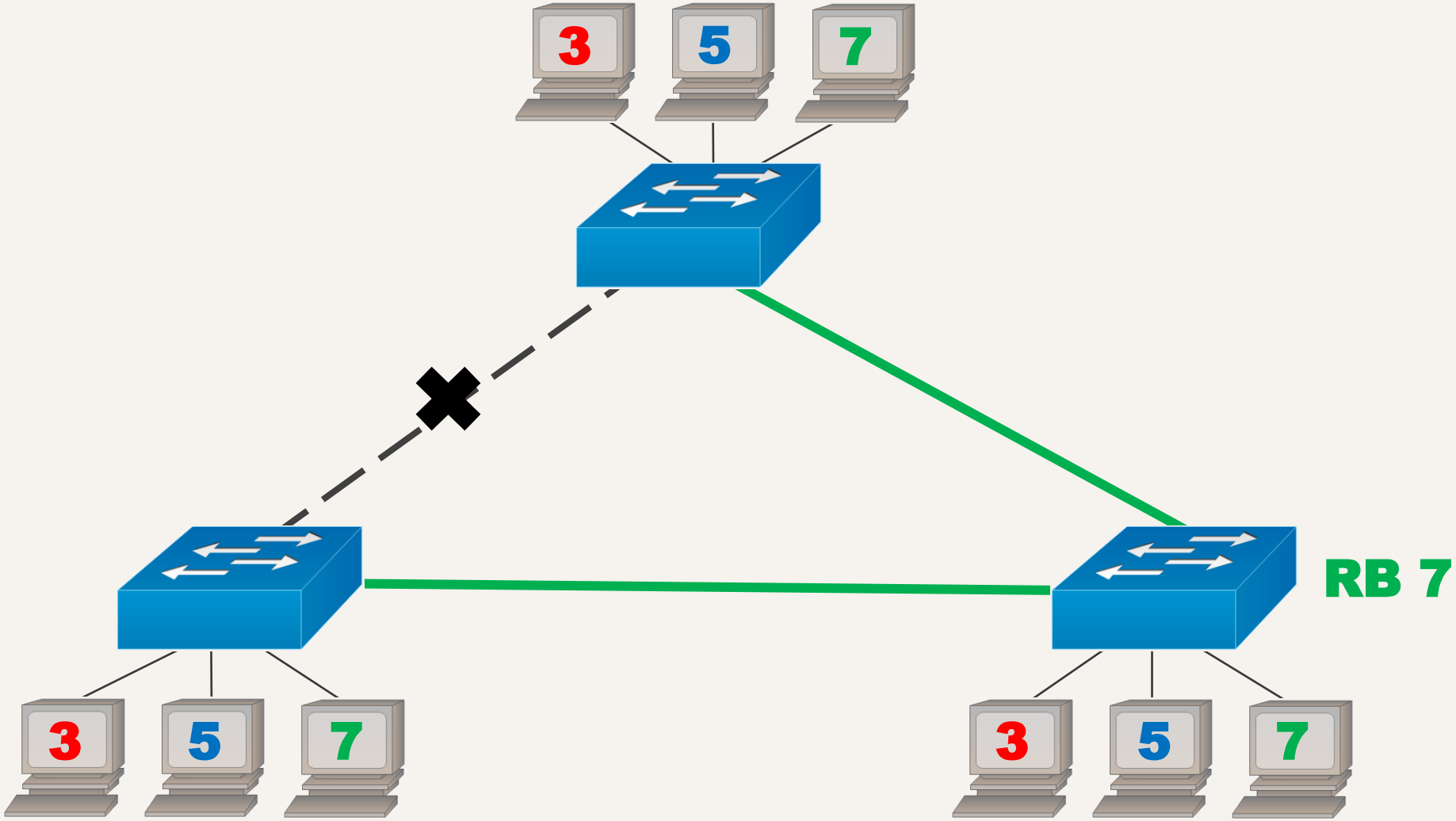
STP — Multiple VLANs



STP — Multiple VLANs



STP — Multiple VLANs



STP — PVST, Faster

After a change is made, the network takes 30-50 seconds to *converge*. In order to improve this, a “rapid” version of STP was introduced by IEEE.

A network that uses **RSTP** (*Rapid Spanning Tree Protocol*) will converge faster, but it still depends on the size of the network. For smaller networks, it’s a matter of seconds.

STP — PVST, Faster

After a change is made, the network takes 30-50 seconds to *converge*. In order to improve this, a “rapid” version of STP was introduced by IEEE.

A network that uses **RSTP** (*Rapid Spanning Tree Protocol*) will converge faster, but it still depends on the size of the network. For smaller networks, it’s a matter of seconds.

per vlan spanning tree

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

Portfast

In case the administrator knows that a port will for sure not be in a loop, they can configure the port to converge faster – STA takes its time, but if it knows that a loop is not possible on a given interface, it can simply skip the procedure and send the port into ***forwarding*** state right away.

```
Switch(config-if)#spanning-tree portfast
```

STP
Spanning-Tree Protocol

CyberQuince