



Table of Contents

CCIE Routing & Switching

► Unit 1: Preparation

▼ Unit 2: Switching

- Static MAC Address Table Entry
- Cisco Switch Virtualization
- Introduction to VLANs (Virtual LAN)
- How to configure VLANs
- 802.1Q Encapsulation
- How to configure a trunk between switches
- Cisco DTP (Dynamic Trunking Protocol) Negotiation
- 802.1Q Tunneling (Q-in-Q)
- Etherchannel over 802.1Q Tunneling
- How to change the Native VLAN
- VTP (VLAN Trunking Protocol)
- VTP Version 3
- Protected Port
- Private VLANs (PVLAN)
- Introduction to Spanning-Tree
- Spanning-Tree Cost Calculation
- PVST (Per VLAN Spanning Tree)
- Spanning-Tree Port States
- Spanning-Tree TCN (Topology Change Notification)
- Spanning-Tree Portfast
- Spanning-Tree UplinkFast

[Spanning-Tree Backbone Fast](#)**[Rapid Spanning-Tree](#)**[Rapid Spanning-Tree Configuration](#)[MST \(Multiple Spanning-Tree\)](#)[Spanning-Tree BPDUGuard](#)[Spanning-Tree BPDUFilter](#)[Spanning-Tree RootGuard](#)[Spanning-Tree LoopGuard and UDLD](#)[FlexLinks](#)[Introduction to Etherchannel](#)[Layer 3 Etherchannel](#)[Cisco IOS SPAN and RSPAN](#)

- ▶ [Unit 3: IP Routing](#)
- ▶ [Unit 4: RIP](#)
- ▶ [Unit 5: EIGRP](#)
- ▶ [Unit 6: OSPF](#)
- ▶ [Unit 7: BGP](#)
- ▶ [Unit 8: Multicast](#)
- ▶ [Unit 9: IPv6](#)
- ▶ [Unit 10: Quality of Service](#)
- ▶ [Unit 11: Security](#)
- ▶ [Unit 12: System Management](#)
- ▶ [Unit 13: Network Services](#)
- ▶ [Unit 14: MPLS](#)

You are here: [Home](#) » [Cisco](#) » [CCIE Routing & Switching](#)

Rapid Spanning-Tree (RSTP)



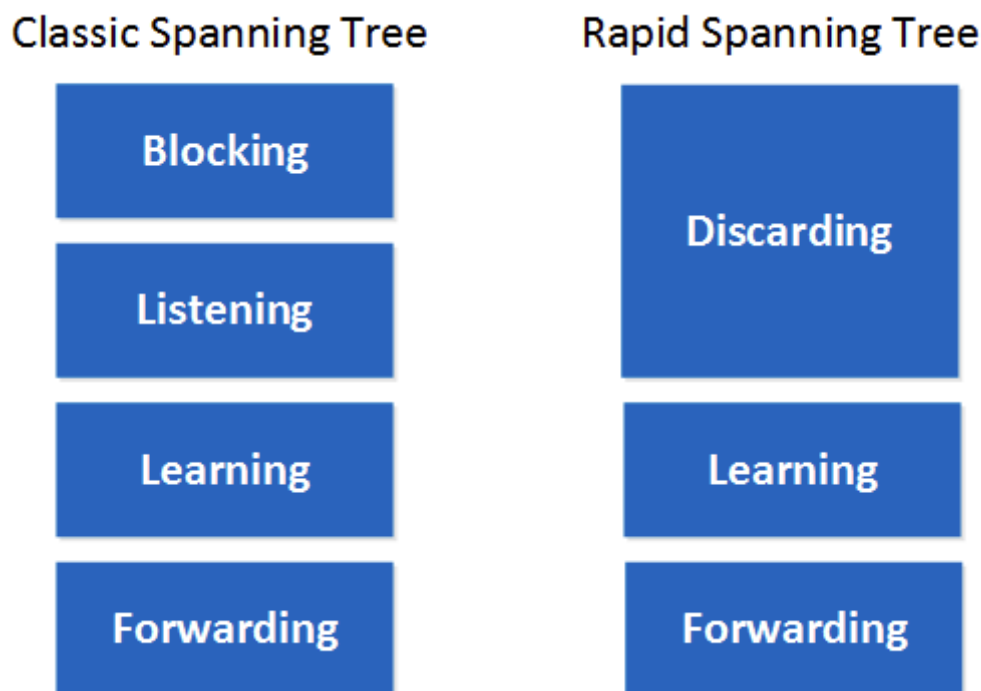
12 votes



Nowadays we see more and more routing in our networks. Routing protocols like OSPF and EIGRP are much faster than spanning-tree when they have to deal with changes in the network. To keep up with the speed of these routing protocols another flavor of spanning-tree was created...**rapid spanning-tree**.

Rapid spanning-tree is not a revolution of the original spanning-tree but an evolution. Behind the scenes some things have been changed to speed up the process, configuration-wise it's the same as what you have seen so far. I will refer to the original spanning-tree as "classic spanning-tree".

Let's dive into rapid spanning-tree and we'll see what the differences are with the classic spanning-tree. Take a look at the picture below:



Remember the port states of spanning-tree? We have a blocking, listening, learning and forwarding port state. This is the first difference between spanning-tree and rapid spanning-tree. Rapid spanning-tree only has three port states:

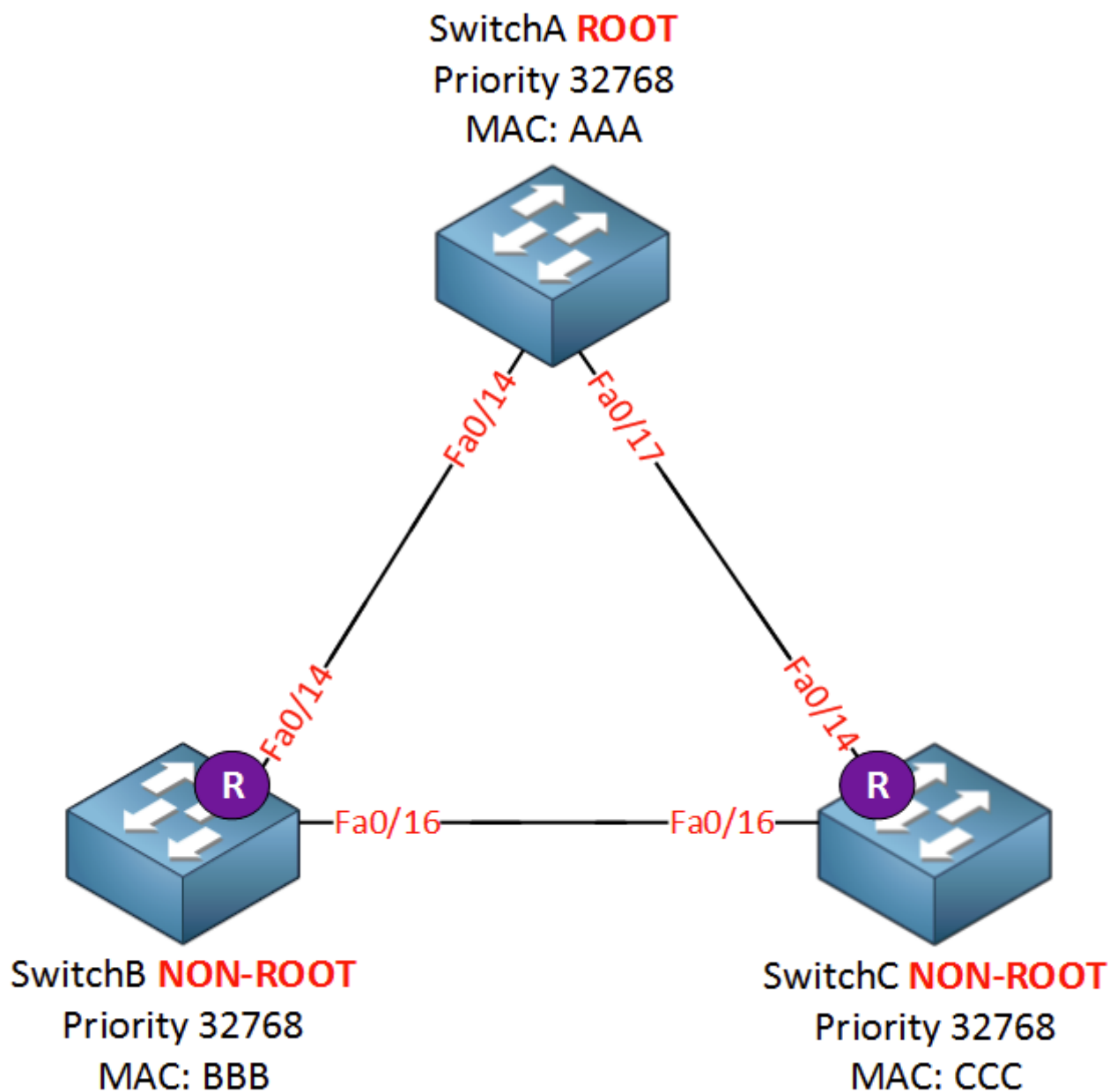
- **Discarding**
- Learning
- Forwarding

You already know about learning and forwarding but **discarding** is a new port state. Basically it combines the blocking and listening port state. Here's a nice overview:

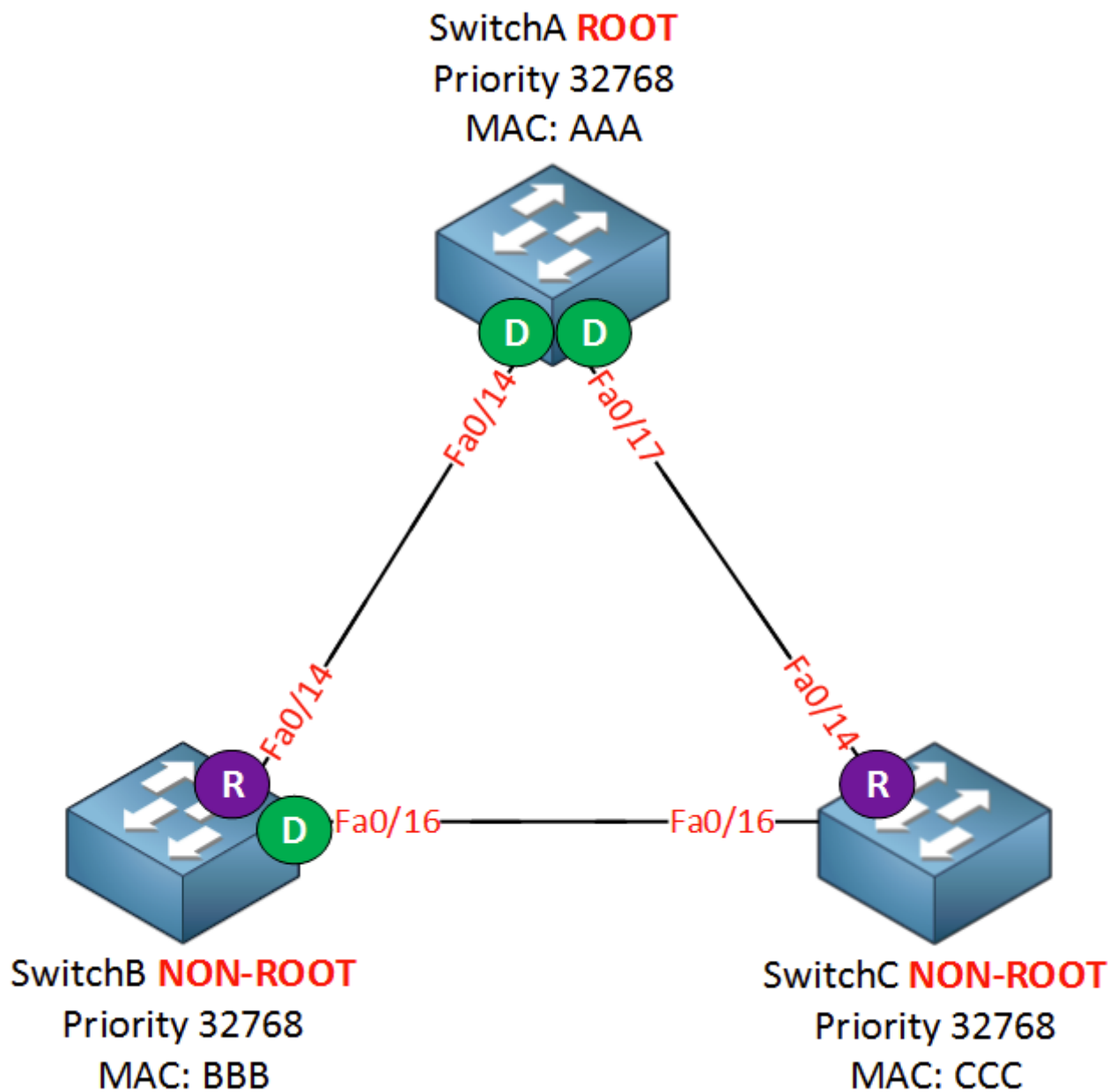
| Classic Spanning-Tree | Rapid Spanning-Tree | Port active in topology? | Learns MAC addresses? |
|-----------------------|---------------------|--------------------------|-----------------------|
| Blocking | Discarding | No | No |
| Listening | Discarding | Yes | No |

| | | | |
|------------|------------|-----|-----|
| Learning | Learning | Yes | Yes |
| Forwarding | Forwarding | Yes | Yes |

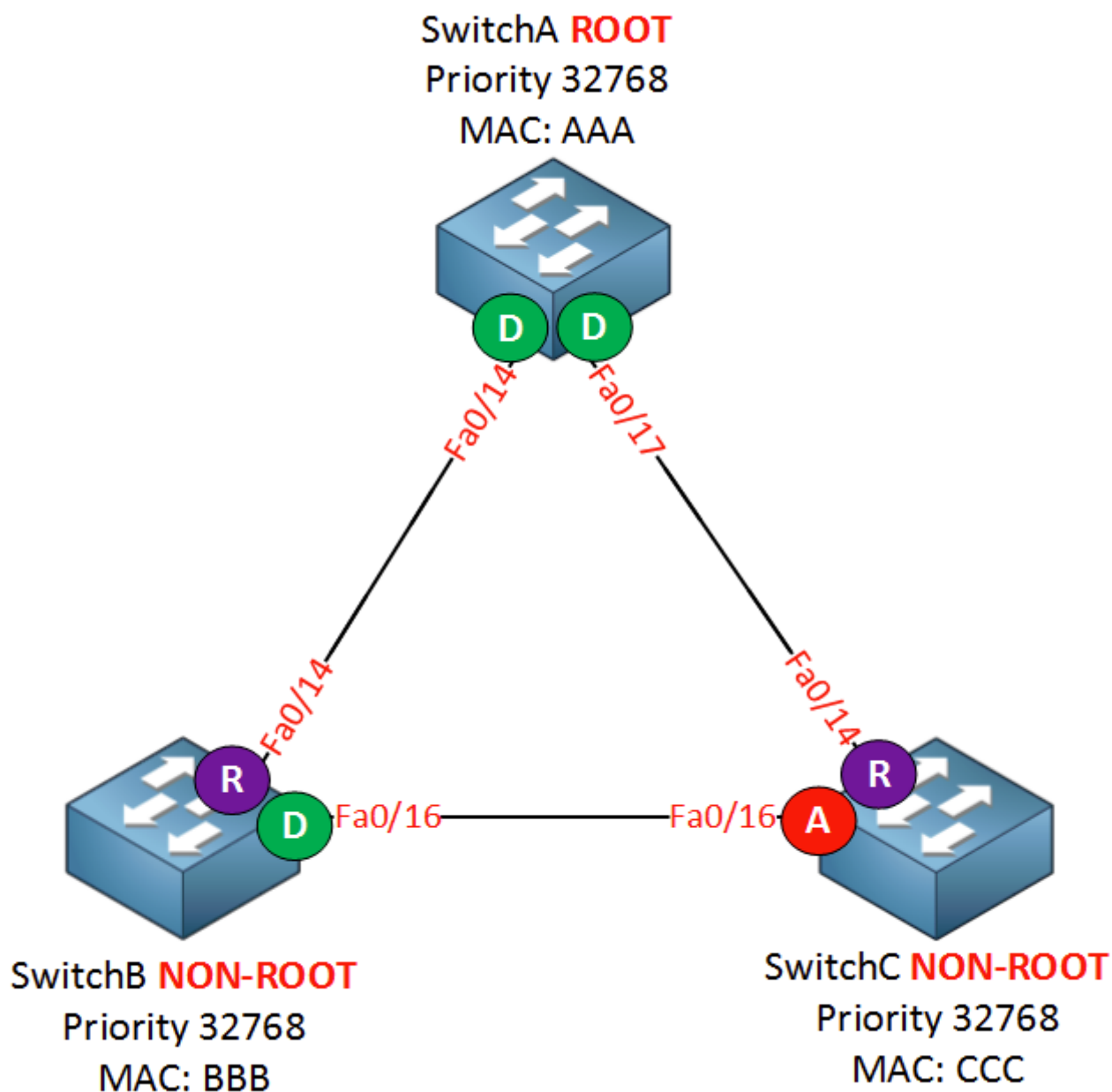
Do you remember all the other port roles that spanning-tree has? Let's do a little review and I'll show you what is different for rapid spanning-tree:



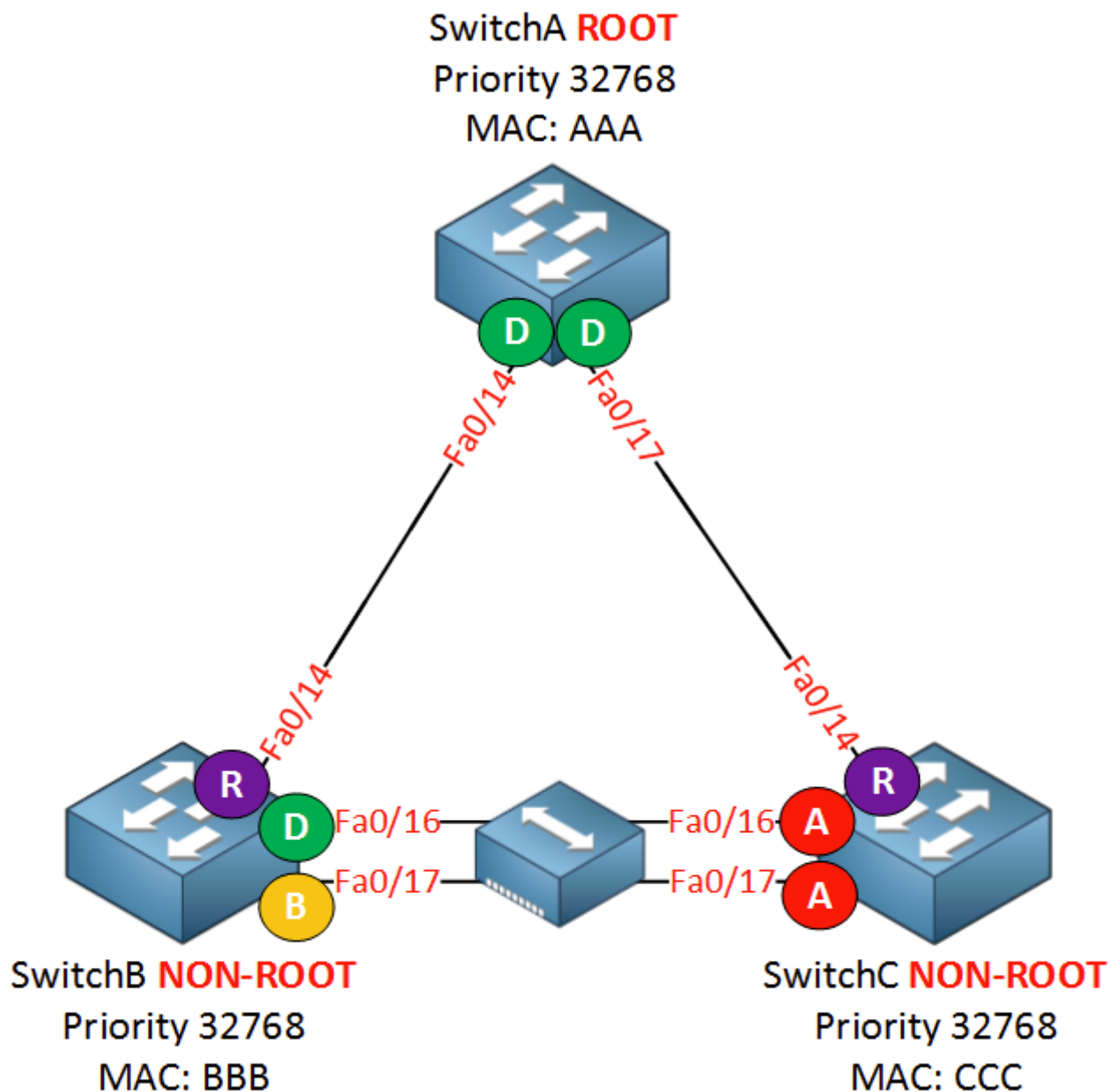
The switch with the best bridge ID (priority + MAC address) becomes the root bridge. The other switches (non-root) have to find the shortest cost path to the root bridge. This is the root port. There's nothing new here, this works exactly the same for rapid spanning-tree. The next step is to select the designated ports:



On each segment there can be only one designated port or we'll end up with a loop. The port will become the designated port if it can send the best BPDU. SwitchA as a root bridge will always have the best ports so all of interfaces will be designated. The fa0/16 interface on SwitchB will be the designated port in my example because it has a better bridge ID than SwitchC. There's still nothing new here compared to the classic spanning-tree. The interfaces that are left will be blocked:



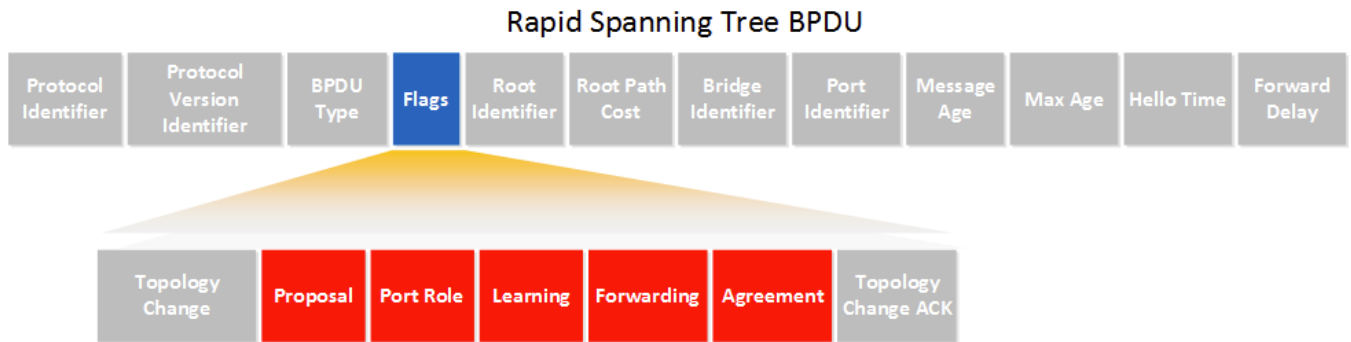
SwitchC receives better BPDUs on its fa0/16 interface from SwitchB and thus it will be blocked. This is the alternate port and it's still the same thing for rapid spanning-tree. Let me show you a new example with a port state that is new for rapid spanning-tree:



Here is a new port for you, take a look at the fa0/17 interface of SwitchB. It's called a **backup port** and it's new for rapid spanning-tree. You are very unlikely to see this port on a production network though. Between SwitchB and SwitchC I've added a hub. Normally (without the hub in between) both fa0/16 and fa0/17 would be designated ports.

Because of the hub the fa0/16 and fa0/17 interface on SwitchB are now in the **same collision domain**. Fa0/16 will be elected as the designated port and fa0/17 will become the **backup port** for the fa0/16 interface. The reason that SwitchB sees the fa0/17 interface as a backup port is because it receives its own BPDUs on the fa0/16 and fa0/17 interfaces and understands that it has two connections to the same segment. If you remove the hub the fa0/16 and fa0/17 will both be designated ports just like the classic spanning-tree.

Something else that is different is the BPDU, take a look:



The BPDU is different for rapid spanning-tree. In the classic spanning-tree the **flags** field only had two bits in use:

- Topology change.
- Topology change acknowledgment.

All bits of the flag field are now used. The role of the port that originates the BPDU will be added by using the **port role** field, it has the following options:

- Unknown
- Alternate / Backup port.
- Root port.
- Designated port.

This new BPDU is called a **version 2 BPDU**. Switches running the old version of spanning-tree will drop this new BPDU version. In case you are wondering...rapid spanning-tree and the old spanning **are compatible!** Rapid spanning-tree has a way of dealing with switches running the older spanning-tree version.

Let's walk through the other stuff that has been changed:

BPDUs are now sent **every hello time**. Only the root bridge generated BPDUs in the classic spanning-tree and those were relayed by the non-root switches if they received it on their root port. Rapid spanning-tree works differently...all switches generate BPDUs **every two seconds (hello time)**. This is the default hello time but you can change it.

The classic spanning-tree uses a max age timer (20 seconds) for BPDUs before they are discarded. Rapid spanning-tree works differently! BPDUs are now used as a **keepalive mechanism** similar to what routing protocols like OSPF or EIGRP use. If a switch **misses three BPDUs** from a neighbor switch it will assume connectivity to this switch has been lost and it will remove all MAC addresses immediately.

Rapid spanning tree will **accept inferior BPDUs**. The classic spanning tree ignores them. Does this ring a bell? This is pretty much the backbone fast feature of classic spanning-tree.

Transition speed (convergence time) is the most important feature of rapid spanning tree. The classic spanning tree had to walk through the listening and learning state before it would move an interface to the forwarding state, this took 30 seconds with the default timers. The classic spanning tree was based on **timers**.

Rapid spanning **doesn't use timers** to decide whether an interface can move to the forwarding state or not. It will use a **negotiation mechanism** for this. I'll show you how this works in a bit.

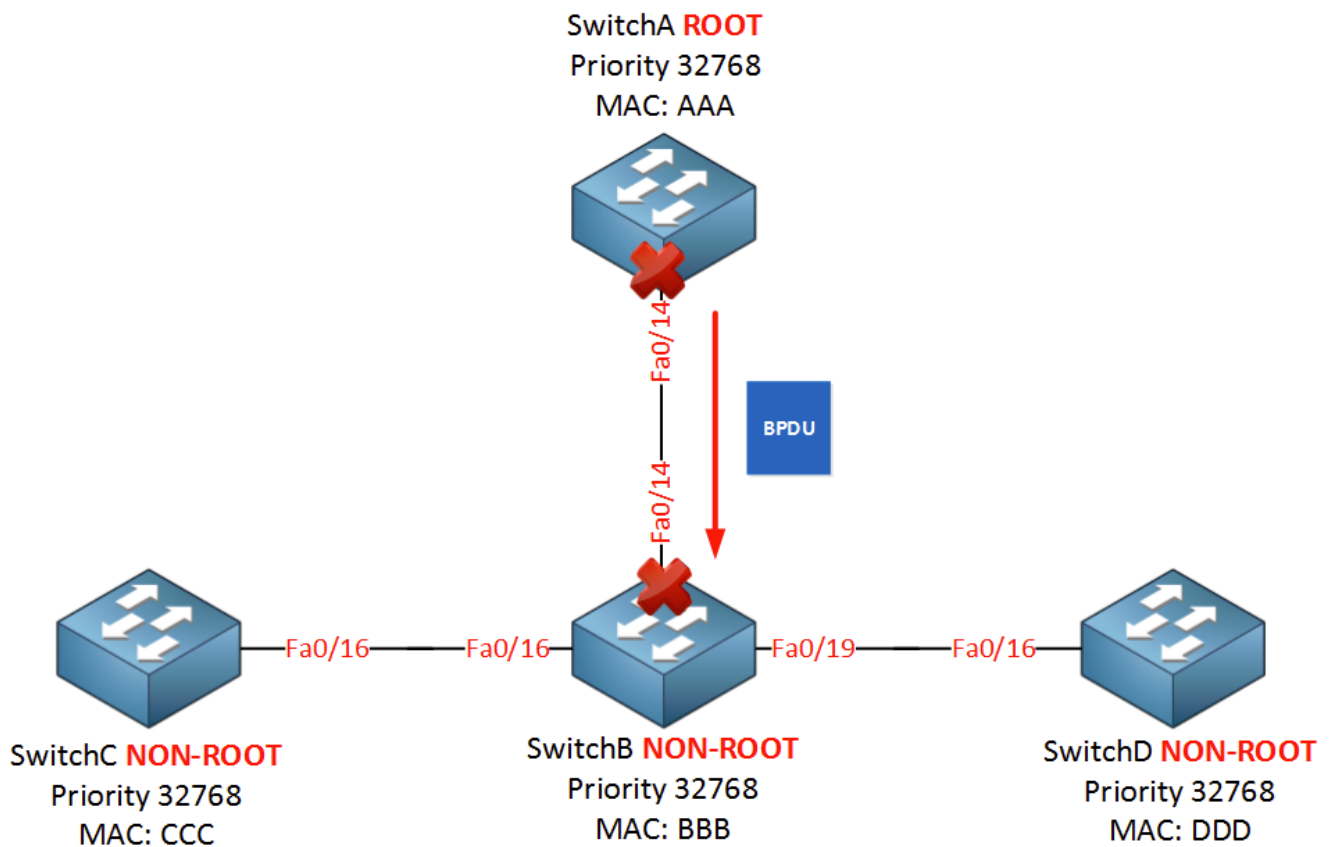
Do you remember portfast? If we enable portfast while running the classic spanning tree it will skip the listening and learning state and put the interface in forwarding state right away. Besides moving the interface to the forwarding state it will also **not generate topology changes** when the interface goes up or down. We still use portfast for rapid spanning tree but it's now referred to as an **edge port**.

Rapid spanning tree can only put interfaces in the forwarding state really fast on **edge ports (portfast)** or **point-to-point interfaces**. It will take a look at the **link type** and there are only two link types:

- Point-to-point (full duplex)
- Shared (half duplex)

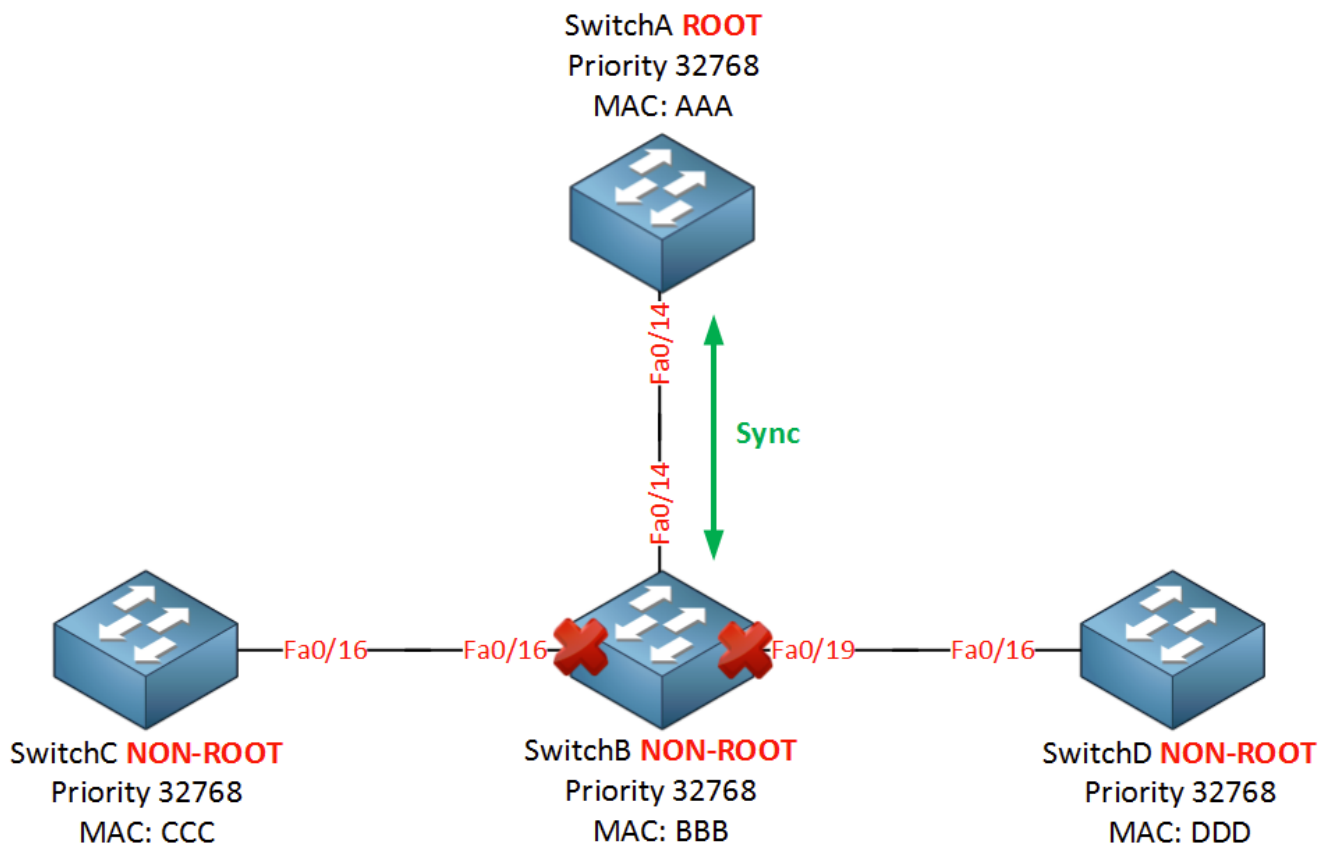
Normally we are using switches and all our interfaces are configured as full duplex, rapid spanning tree sees these interfaces as point-to-point. If we introduce a hub to our network we'll have half duplex which is seen as a shared interface to rapid spanning-tree.

Let's take a close look at the negotiation mechanism that I described earlier:

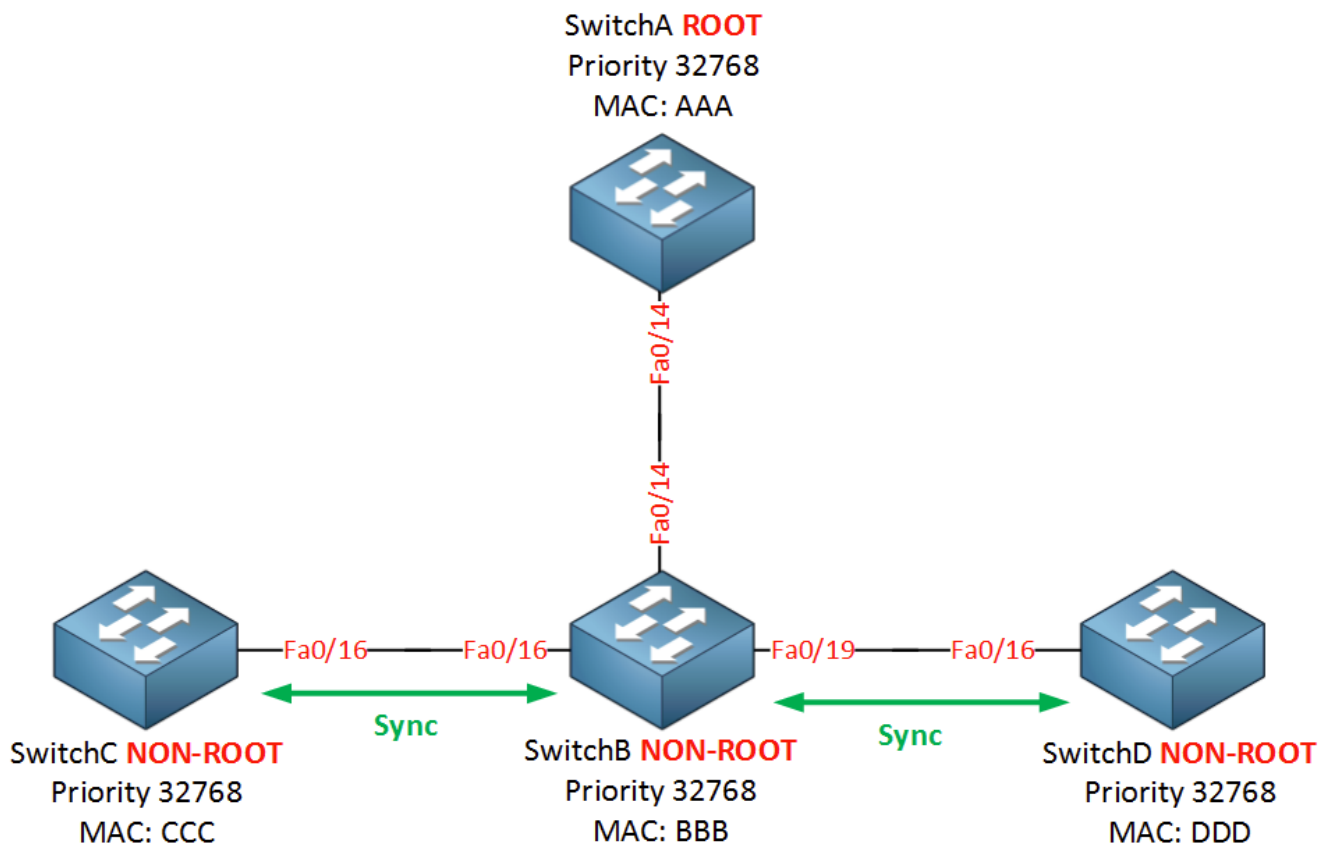


Let me describe the rapid spanning tree synchronization mechanism by using the picture above. SwitchA on top is the root bridge. SwitchB, SwitchC and SwitchD are non-root bridges.

As soon as the link between SwitchA and SwitchB comes up their interfaces will be in blocking mode. SwitchB will receive a BPDU from SwitchA and now a **negotiation** will take place called **sync**:

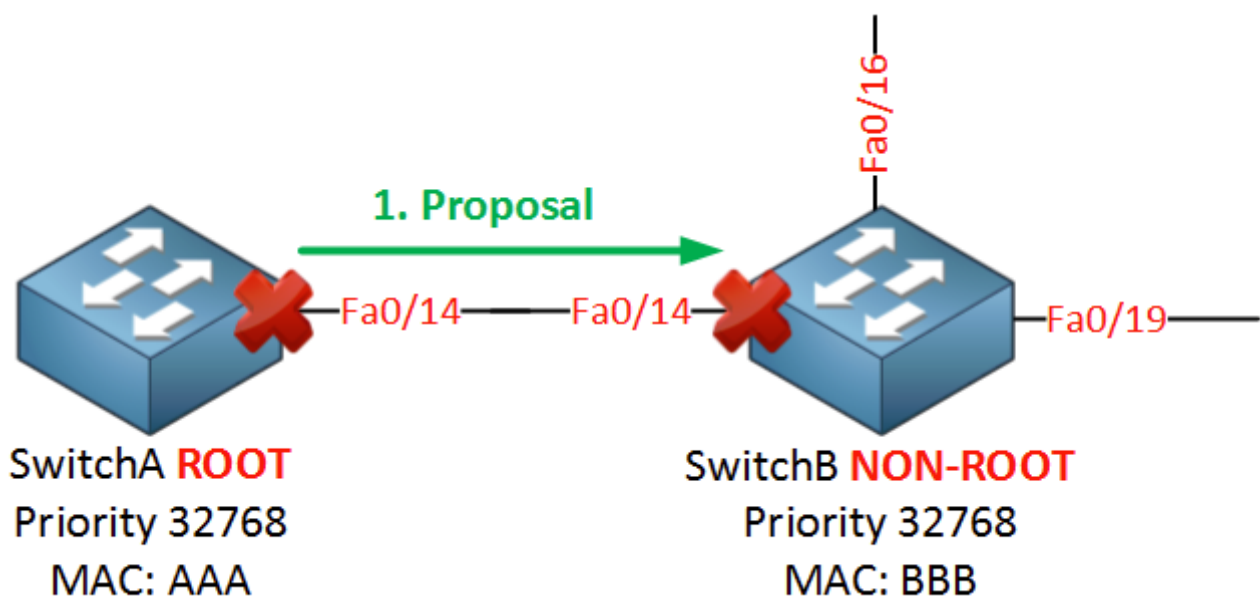


After SwitchB received the BPDU from the root bridge it **immediately blocks all its non-edge designated ports**. Non-edge ports are the interfaces that connect to other switches while edge ports are the interfaces that have portfast configured. As soon as SwitchB blocks its non-edge ports the link between SwitchA and SwitchB will go into forwarding state. SwitchB will now do the following:

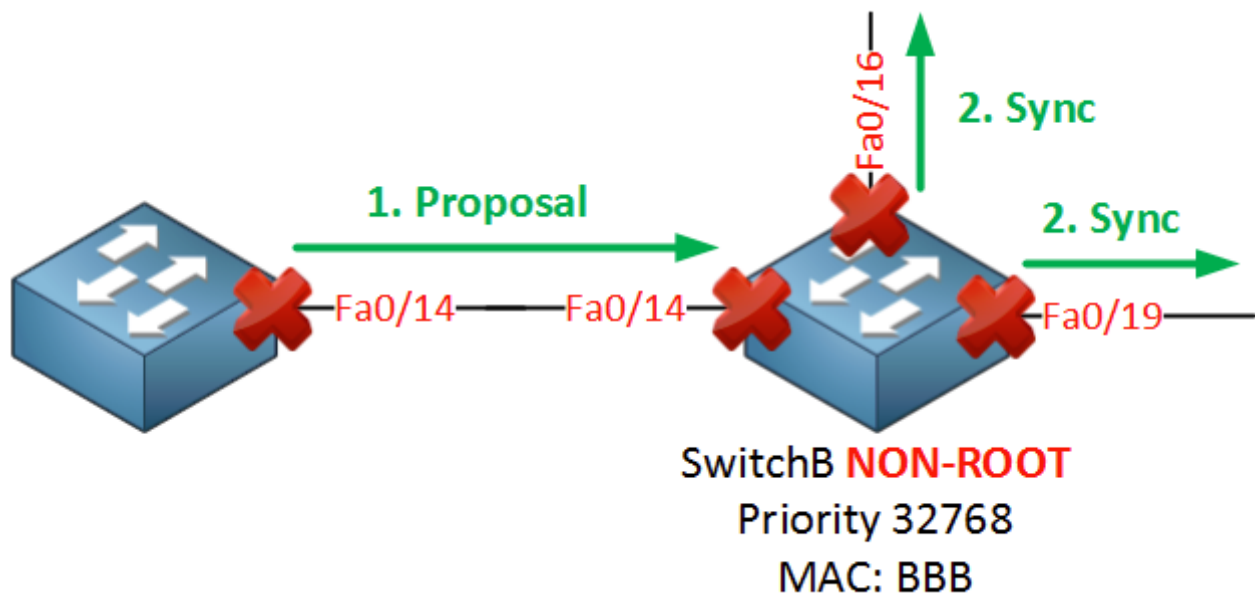


SwitchB will also perform a sync operation with both SwitchC and SwitchD so they can quickly move to the forwarding state.

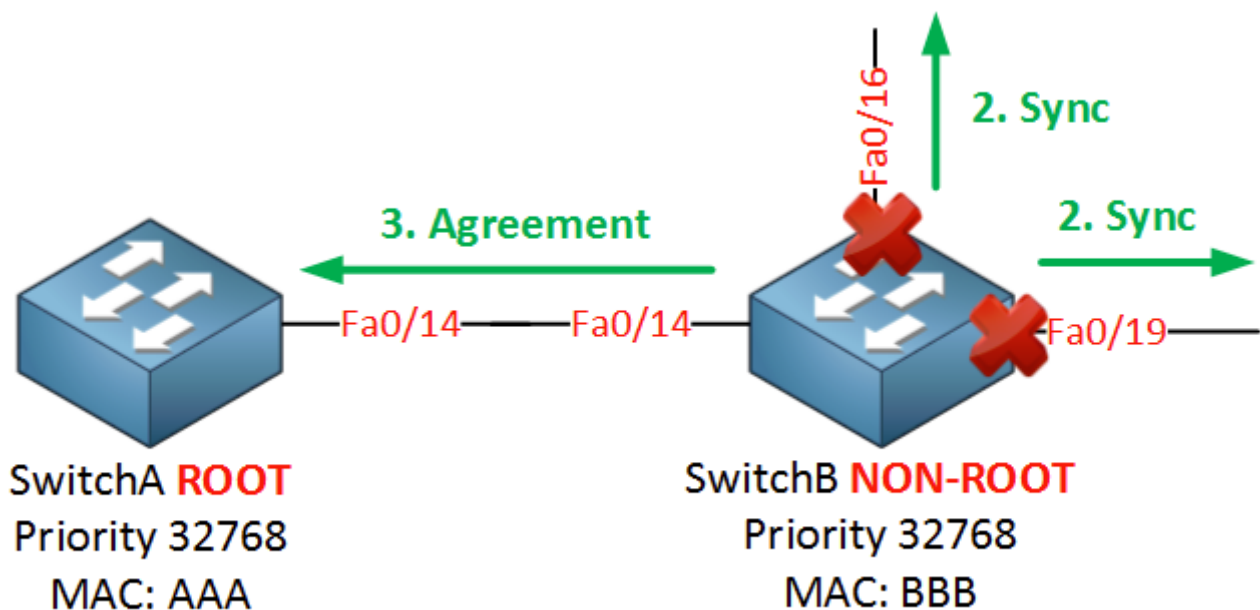
Are you following me so far? The lesson to learn here is that rapid spanning tree uses this **sync mechanism instead of the “timer-based” mechanism** that the classic spanning tree uses (listening > learning > forwarding). I’m going to show you what this looks like on real switches in a bit. Let’s take a closer look at the sync mechanism, let’s look at what happens exactly between SwitchA and SwitchB:



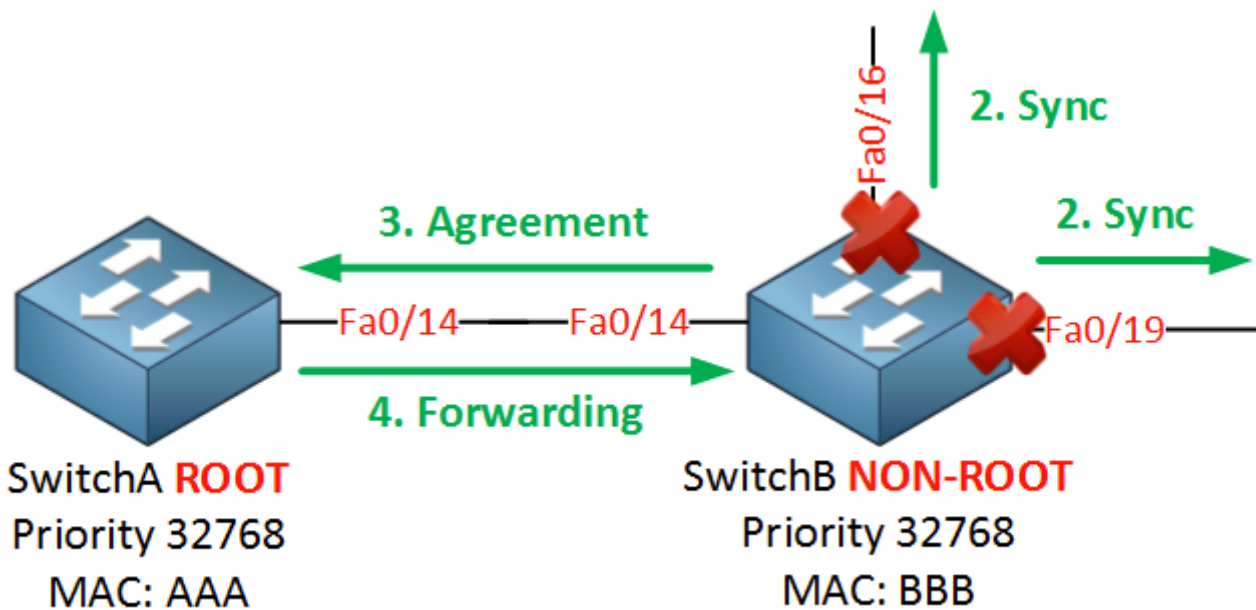
At first the interfaces will be blocked until they receive a BPDU from each other. At this moment SwitchB will figure out that SwitchA is the root bridge because it has the best BPDU information. The sync mechanism will start because SwitchA will set the **proposal bit** in the flag field of the BPDU. When SwitchB receives the proposal it has to do something with it:



SwitchB will block all its non-edge interfaces and will start the synchronization towards SwitchC and SwitchD, once this is done SwitchB will let SwitchA know about this:



Once SwitchB has its interfaces in sync mode it will let SwitchA know about this by sending an **agreement**. This agreement is a **copy of the proposal BPDU** where the proposal bit has been switched off and the agreement bit is switched on. The fa0/14 interface on SwitchB will now go into forwarding mode. When SwitchA receives the agreement here's what happens:



Once SwitchA receives the agreement from SwitchB it will put its fa0/14 interface in forwarding mode immediately.

What about the fa0/16 and fa0/19 interface on SwitchB?

SwitchC NON-ROOT

Priority 32768

MAC: CCC

Fa0/16
Fa0/16

5. Proposal

**SwitchB NON-ROOT**

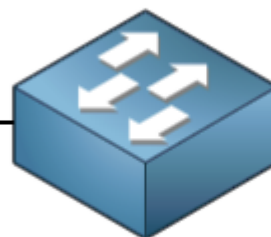
Priority 32768

MAC: BBB

5. Proposal

Fa0/19

Fa0/16

**SwitchD NON-ROOT**

Priority 32768

MAC: DDD

The exact same sync mechanism will take place now on these interfaces. SwitchB will send a proposal on its fa0/16 and fa0/19 interfaces towards SwitchC and SwitchD. SwitchC and SwitchD will send an agreement:

SwitchC NON-ROOT

Priority 32768

MAC: CCC

Fa0/16
Fa0/16

5. Proposal

**SwitchB NON-ROOT**

Priority 32768

MAC: BBB

5. Proposal

Fa0/19

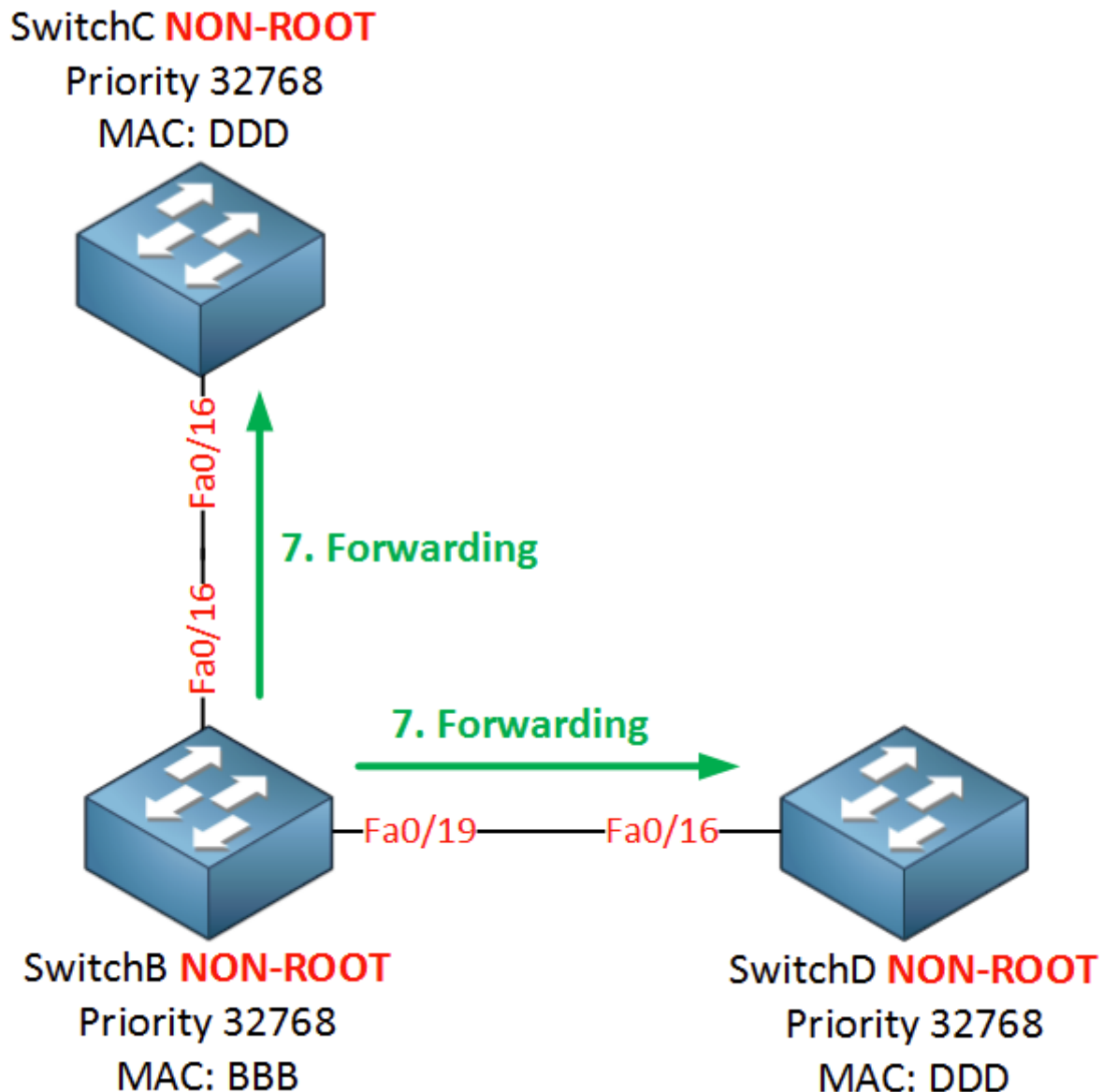
Fa0/16

**SwitchD NON-ROOT**

Priority 32768

MAC: DDD

SwitchC and SwitchD don't have any other interfaces so they will send an agreement back to SwitchB:



SwitchB will place its fa0/16 and fa0/19 interface in forwarding and we are done. This sync mechanism is just a couple of messages flying back and forth and very fast, it's much faster than the timer-based mechanism of the classic spanning tree!

What else is new with rapid spanning tree? There are three more things I want to show you:

- UplinkFast
- Topology change mechanism.
- Compatibility with classic spanning tree.

When you configure the classic spanning tree you have to enable UplinkFast yourself. Rapid spanning tree uses UpLinkFast by default, you don't have to configure it yourself. When a switch loses its root port it will put its alternate port in forwarding immediately.

The difference is that the classic spanning tree needed multicast frames to update the MAC address tables of all switches.

We don't need this anymore because the topology change mechanism for rapid spanning tree is different. So what's different about the topology change mechanism?

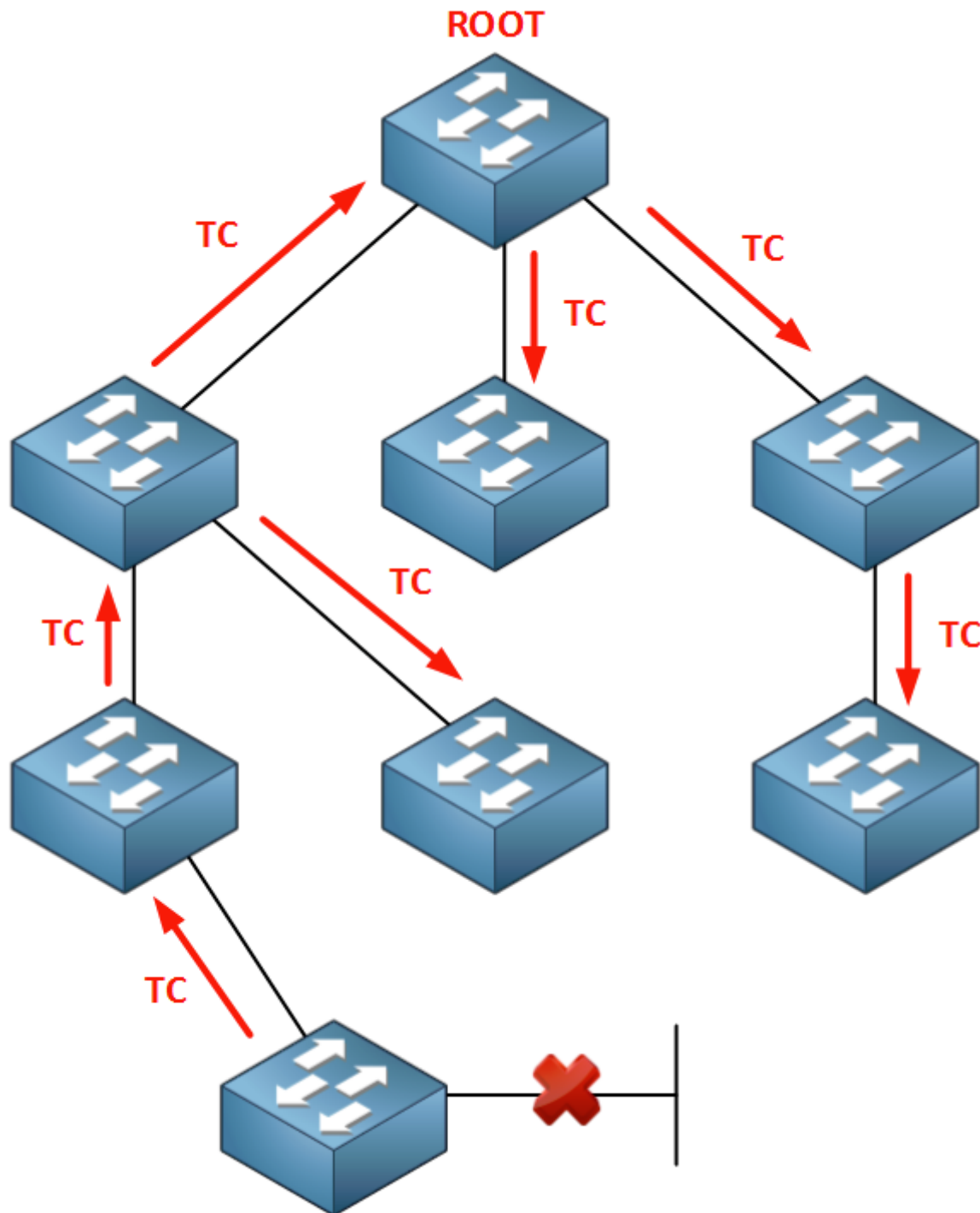
With the classic spanning tree a link failure would trigger a topology change. Using rapid spanning tree a **link failure is not considered as a topology change**. Only non-edge interfaces (leading to other switches) that move to the forwarding state are considered as a topology change. Once a switch detects a topology change this will happen:

- It will start a **topology change while timer** with a value that is twice the hello time. This will be done for all non-edge designated and root ports.
- It will **flush the MAC addresses** that are learned on these ports.
- As long as the topology change while timer is active it will set the topology change bit on BPDUs that are sent out these ports. BPDUs will also be sent out of its root port.

When a neighbor switch receives this BPDU with the topology change bit set this will happen:

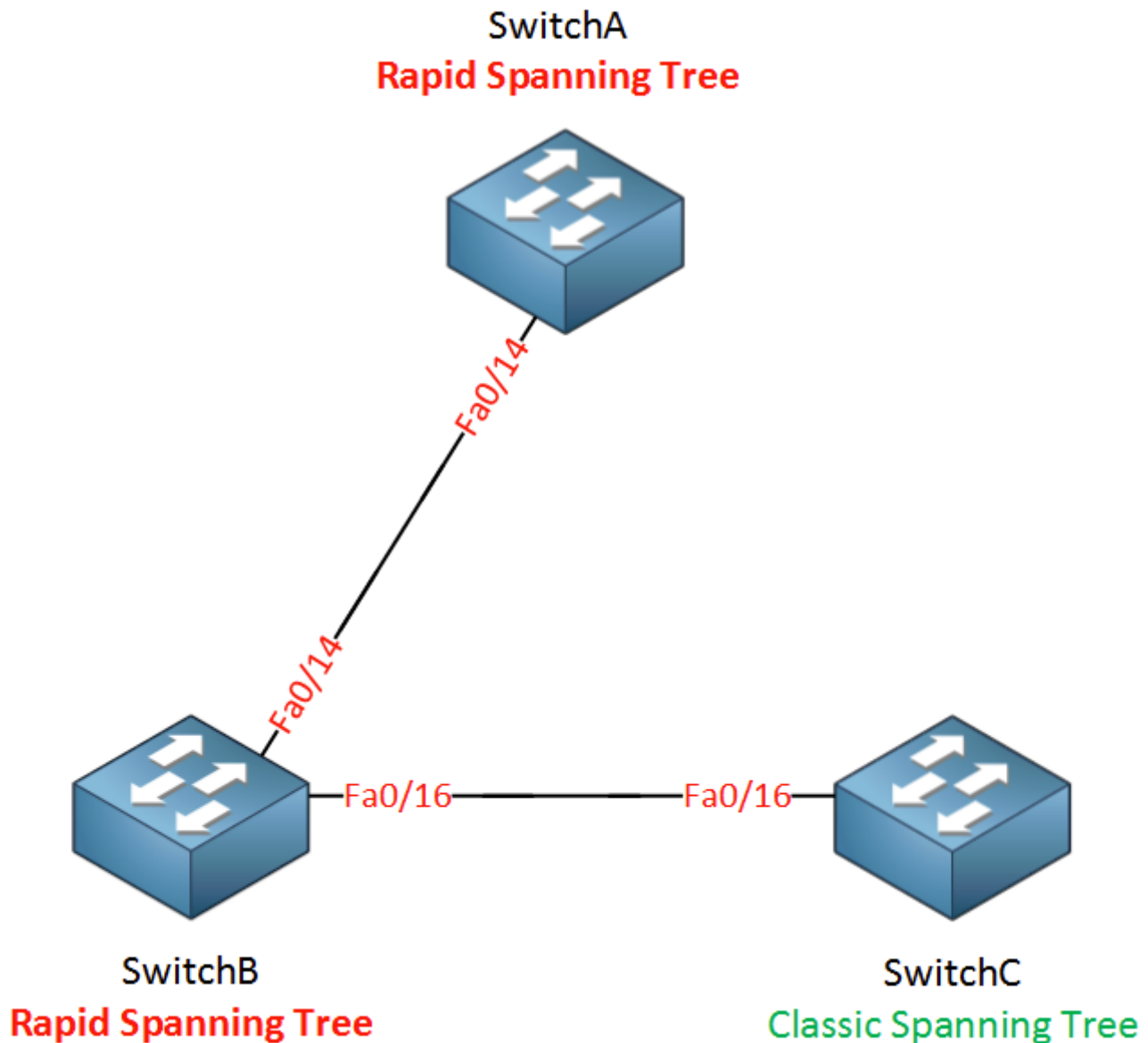
- It will clear all its MAC addresses on all interfaces except the one where it received the BPDU with the topology change on.
- It will start a topology change while timer itself and send BPDUs on all designated ports and the root port, setting the topology change bit.





Instead of sending a topology change all the way up to the root bridge like the classic spanning tree does, the topology change is now quickly flooded throughout the network. If you want to know exactly how this topology change mechanism works then take a look at [this lesson](#).

Last but not least, let's talk about compatibility. The short answer is that rapid spanning tree and classic spanning tree **are compatible**. However when a switch running rapid spanning tree communicates with a switch running classic spanning tree all the Speedy Gonzales features won't work!



Last but not least, let's talk about compatibility. The short answer is that rapid spanning tree and classic spanning tree **are compatible**. However when a switch running rapid spanning tree communicates with a switch running classic spanning tree all the Speedy Gonzales features won't work!


In the example above I have my three switches. Between SwitchA and SwitchB we will run rapid spanning tree. Between SwitchB and SwitchC we will fall back to the classic spanning tree.

Seen enough theory? In the [next lesson](#) I will show you the [configuration and the debugs](#) of everything that you have learned so far.

Rate this Lesson:



[Previous Lesson](#)[Spanning-Tree Backbone Fast](#)[Next Lesson](#)[Rapid Spanning-Tree Configuration](#)[Home](#) › [Forums](#) › [Rapid Spanning-Tree \(RSTP\)](#)

This topic contains 19 replies, has 9 voices, and was last updated by  Andrew P [3 weeks, 4 days ago](#).

Viewing 15 posts - 1 through 15 (of 19 total)

1 2 →

- Author
Posts | [Subscribe](#)
- December 30, 2014 at 08:28 [#11378 Reply](#)



Raed A

Participant

Thank you very much

December 30, 2014 at 08:40 [#11379 Reply](#)



Rene Molenaar

Keymaster

You are welcome Raed!

January 25, 2015 at 15:41 [#11380 Reply](#)



Michael M

Participant

Wow. Excellent lesson. Very clear and to the point. I am definitely enjoying this site and learning. Excellent refresher course !

Please keep up the great work. BTW...Yes I am recommending this site.



January 26, 2015 at 21:20 [#11381 Reply](#)



Rene Molenaar

Keymaster

Glad to hear you like it!

June 13, 2015 at 23:11 [#11382 Reply](#)



Ricardo V

Participant

Excellent explanation! 😊

June 21, 2015 at 09:52 [#11383 Reply](#)



Srinivasan C

Participant

Hi Rene,

Very Good explanation on RSTP.

I think BPDU message flags are reversed in the diagram.

Thanks,

Srini



June 21, 2015 at 12:11 [#11384 Reply](#)



Srinivasan C

Participant

Hi Rene,

what is the characteristic of blocking mode in RSTP? Can send and receive BPDUs?

In most of the scenario Alternate / Backup port directly transition in to forwarding ?

On what condition the port will be in Learning stat?

Thanks,

Srini

June 21, 2015 at 16:05 [#11385 Reply](#)



Rene Molenaar
Keymaster
Hi Srini,

The port in blocking mode will only receive BPDUs, not send them. The same thing applies to the root port. Only the designated port sends BPDUs since this is downstream (to other switches).

The alternate port can go into forwarding when the switches loses its root port, the same thing occurs when you use uplinkfast with PVST.

Each interface has to go through the blocking, listening and learning mode unless you enable portfast:

<https://networklessons.com/cisco/spanning-tree-port-states/>

Rene

June 21, 2015 at 16:10 #11386 Reply



Rene Molenaar
Keymaster
Hi Srini,



Which one do you mean?

Rene

June 21, 2015 at 16:21 #11387 Reply



Srinivasan C
Participant
Hi Rene,

Can you please explain bit more on TC while timer and Fd While Timer in RSTP?
Tc while timer $2 \times \text{hello time} = 4 \text{ sec}$, only 4 sec for flushing the Mac address learned through topology change action ports. Please correct my understanding on Tc while timer.

Thanks,
Srini

June 21, 2015 at 16:29 [#11388 Reply](#)



Srinivasan C

Participant

Hi Rene,

I am seeing BPDU Message Flags in reverse order.

1 TCN

2 Proposol

.

.

.

7 Agreement

8. TCA.

Please correct me If my understanding is wrong.

Thanks,
Srini

June 21, 2015 at 16:34 [#11389 Reply](#)



Srinivasan C

Participant

Hi Rene,

You are correct . In RSTP a special case , designated port in the non-edge port of a switch will go in to blocking state and block all non-BPDU traffic until it gets agreement from the neighbour switch.

Thanks,
Srini

June 25, 2015 at 16:26 [#11390 Reply](#)



Rene Molenaar



Keymaster
Hi Srini,

When a non-edge or root port moves into the forwarding state then this triggers as topology change in RSTP. When this happens it will start the TC while timer which 2x hello (4 seconds in total) and it will flush the MAC addresses that were learned on all non-edge interfaces.

When the TC while timer is active, all BPDUs that the switch will send will have the TC bit set. Other switches that receive this BPDU will start their own TC while timer and they will flush all MAC addresses except the interface where they received the BPDU with the TC bit.

I hope this helps.

Rene

October 12, 2015 at 03:49 [#18005 Reply](#)



Frades
Participant
awesome lesson! very detailed!

i have a question. you said that RSTP uses uplinkfast by default, so no need to configure it.

done some research on the internet, they are saying that RSTP doesnt use the BBfast and ULfast, but it has a similar feature like them which is set on by default. meaning, when you issue the "show span summary" you'll see the Uplinkfast and BackboneFast disabled, because RSTP has a built-in feature like that. is this true? thanks!



October 12, 2015 at 09:46 [#18012 Reply](#)



Rene Molenaar
Keymaster
Hi John,

That's correct, RSTP has features that are similar to uplinkfast and backbonefast, there's no need to activate these.

Rene

- Author
Posts

Viewing 15 posts - 1 through 15 (of 19 total)

1 2 →

Reply To: Rapid Spanning-Tree (RSTP)

b **/** **link** **b-quote** **del** **img** **ul** **ol** **li** **code** **close tags**

Please put code in between `backticks` or use the CODE button.

To place inline images, please use any image share service (such as [TinyPic](#) or [Imgur](#)) and use the IMG button!

☐ Notify me of follow-up replies via email

Maximum file size allowed is 2048 KB.

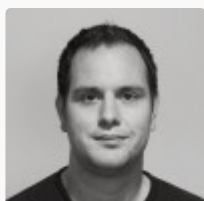
Attachments:

Файл не выбран

[Add another file](#)



About NetworkLessons.com



Hello There! I'm René Molenaar (CCIE #41726), Your Personal Instructor of Networklessons.com. I'd like to teach you everything about Cisco, Wireless and Security. I am here to Help You Master Networking!

Social Fans



14,267

FANS



7,929

FOLLOWERS



1,589

SUBSCRIBERS

Highest Rated Lessons

MPLS Layer 3 VPN Configuration



VRF Lite Configuration on Cisco IOS



Cisco Portfast Configuration



IPv6 Address Types



EIGRP Stub Explained



New Lessons

Introduction to Cisco IOS XE

ERSPAN Configuration on Cisco IOS XE

IGMP Filter

IGMP Snooping without Router

Cisco Group Management Protocol (CGMP)

Disclaimer

Privacy Policy

Support

Rapid Spanning-Tree (RSTP) written by Rene Molenaar average rating 4.8/5 - 12 user ratings

