Networklessons.com

Search...

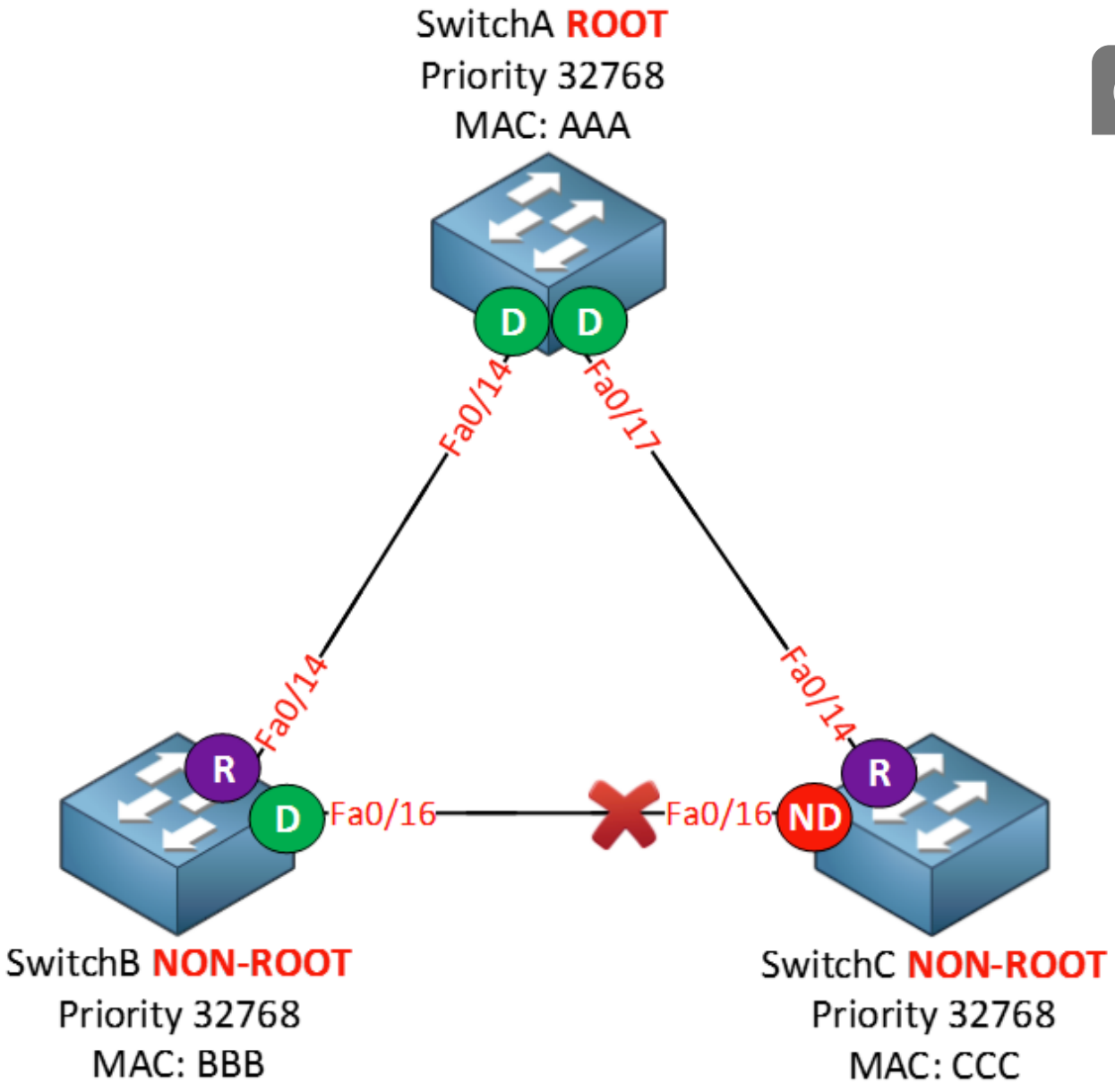
Table of Contents

CCIE Routing & Switching

Spanning-Tree UplinkFast

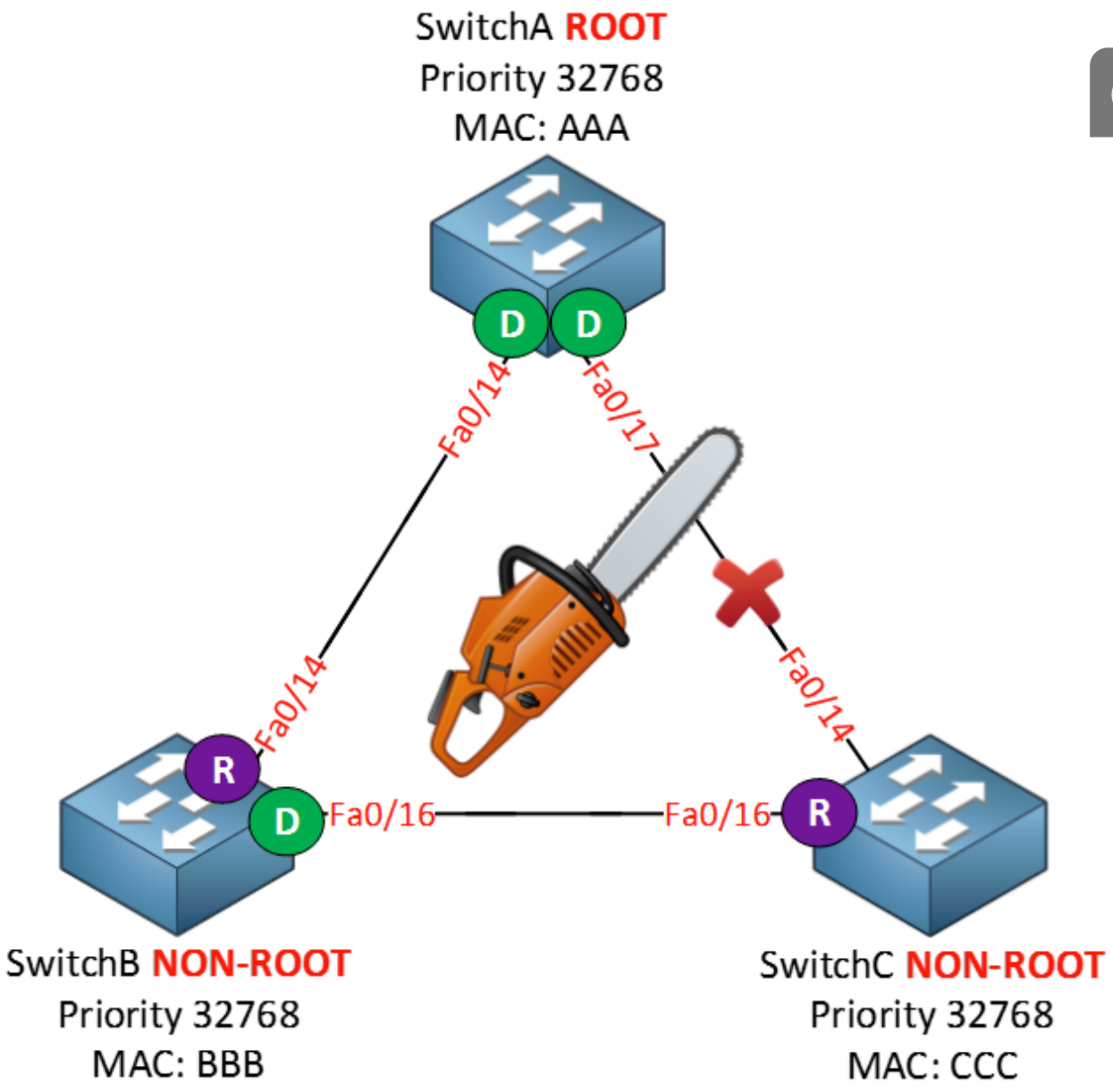
14 votes

Uplinkfast is a spanning-tree feature that was created to improve the convergence time. In this lesson we'll take a look at how uplinkfast works. Here's the topology that I will use to demonstrate it:



Three switches and SwitchA is our root bridge. The fa0/16 interface on SwitchC has been blocked. I'm only using VLAN 1 so nothing fancy here...

If we look at SwitchC we see that the fa0/16 interface has been blocked and fa0/14 is the root port. Let's see what happens with this topology when one of the links fails:



When the fa0/14 interface on SwitchC fails we'll have to use fa0/16 to reach the root bridge. How long does it take for SwitchC to make the transition? Let's find out:

```
SwitchC#debug spanning-tree events
Spanning Tree event debugging is on
```

```
SwitchC(config)#interface fa0/14
SwitchC(config-if)#shutdown
```

Now we'll just wait for the magic to happen...

```
SwitchC#STP: VLAN0001 new root port Fa0/16, cost 38
SwitchC#STP: VLAN0001 Fa0/16 -> listening
```

```
SwitchC#STP: VLAN0001 Fa0/16 -> learning
SwitchC#STP: VLAN0001 Fa0/16 -> forwarding
```

BPDUs are originated from the root bridge so if we receive BPDUs on an interface the switch knows it can reach the root bridge on this interface. We have to go through the listening (15 seconds) and learning state (15 seconds) so it takes 30 seconds to end up in the forwarding state.

The good thing is that spanning-tree solves the link failure automatically but it also means that we have a downtime of 30 seconds. If you want you can tune the forward delay timer to speed up this process down to roughly 14 seconds.

```
SwitchC(config)#interface fa0/14
SwitchC(config-if)#no shutdown
```

Let's restore connectivity first.

```
SwitchC(config)#spanning-tree uplinkfast
```

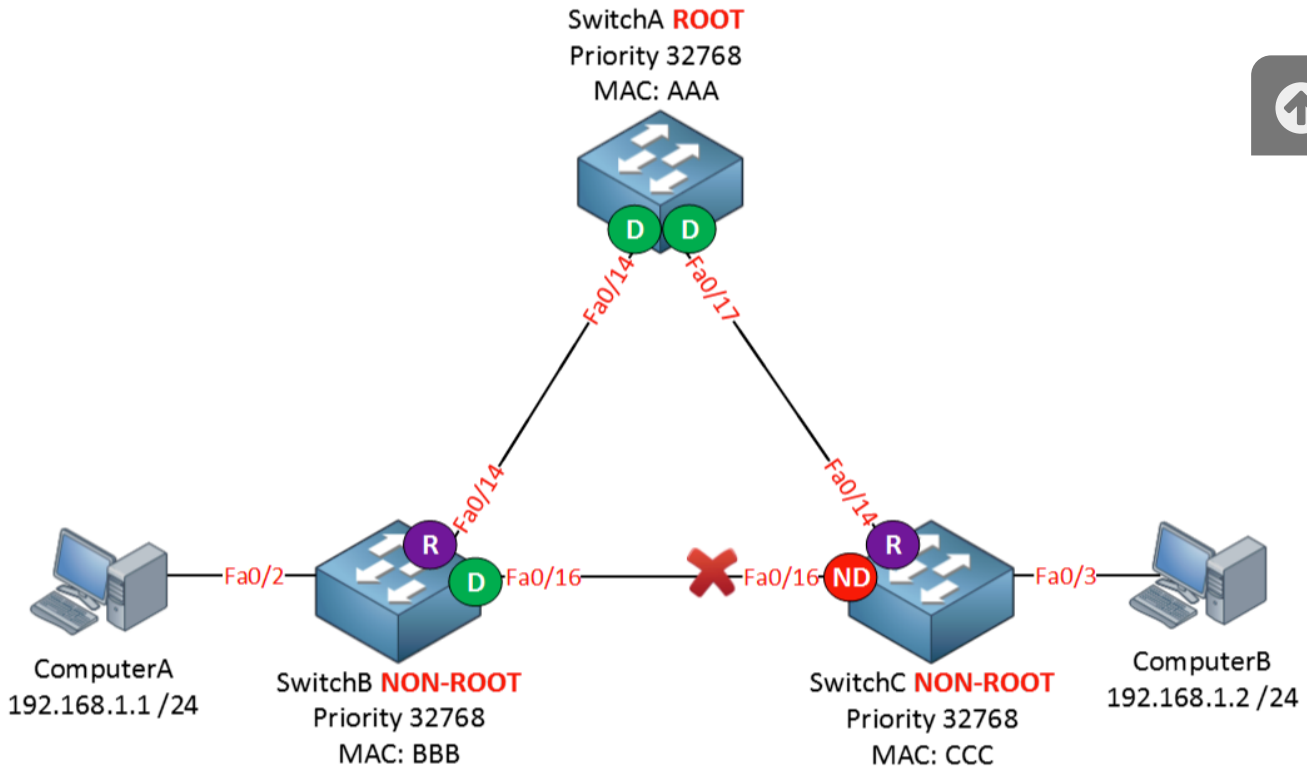
Now I'm going to enable **spanning-tree uplinkfast**. This is a global command, you can't configure it on the interface level.

```
SwitchC(config)#interface fa0/14
SwitchC(config-if)#shutdown
```

```
SwitchC# STP: VLAN0001 new root port Fa0/16, cost 3038
SwitchC# %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLAN0001 FastEthernet0/16 moved to
Forwarding (UplinkFast).
```

Here's the big difference. When uplinkfast is enabled a **non-designated port will go to forwarding state immediately if the root port fails**. Instead of 30 seconds downtime connectivity is restored immediately.

UplinkFast is useful but it will cause a problem with our MAC address tables. In the picture above I added two computers to our topology. Interface fa0/16 on SwitchC is the non-designated port and fa0/14 is the root port. Take a look at the following topology:



Let me show you the MAC address tables for all switches:

```
SwitchA#show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000c.2928.5c6c    DYNAMIC     Fa0/14
   1    000c.29e2.03ba    DYNAMIC     Fa0/17
```

```
SwitchB#show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000c.2928.5c6c    DYNAMIC     Fa0/2
   1    000c.29e2.03ba    DYNAMIC     Fa0/14
```

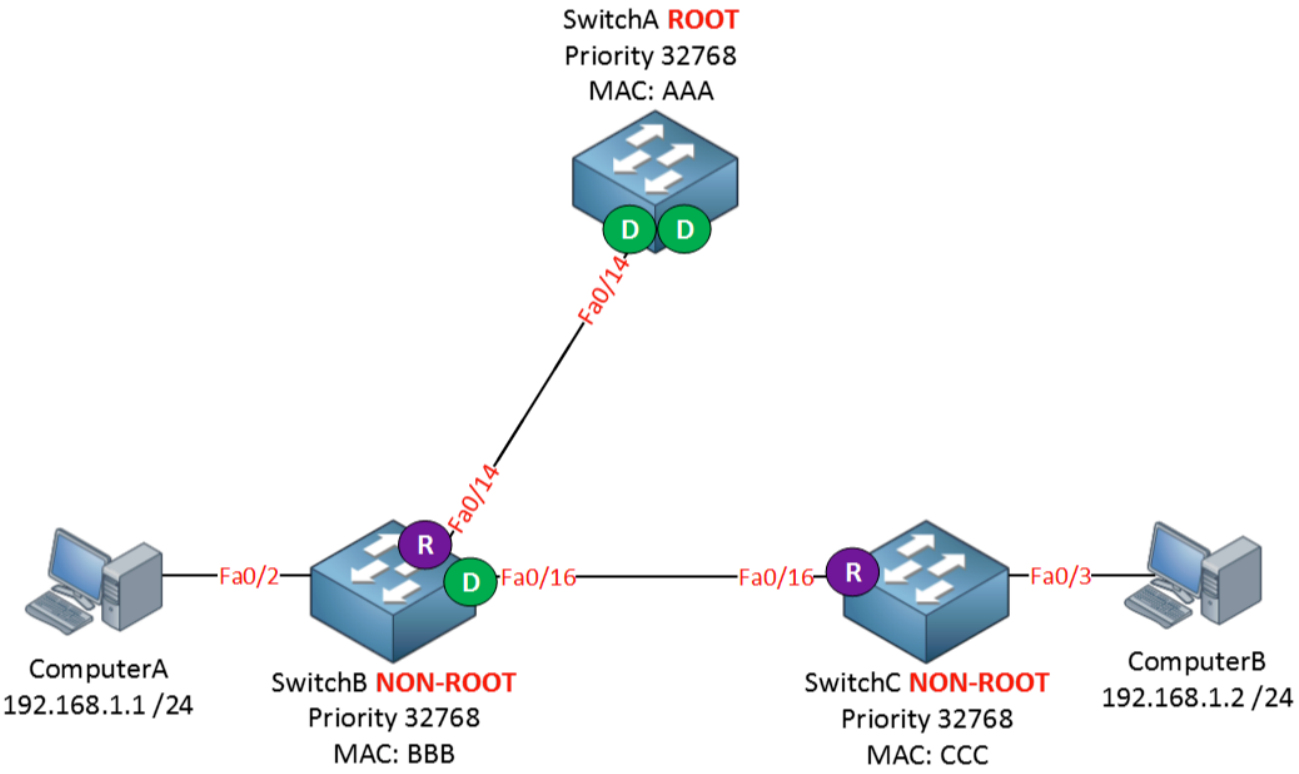
```
SwitchC#show mac address-table dynamic
          Mac Address Table
```

```
-----------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000c.2928.5c6c    DYNAMIC     Fa0/14
   1    000c.29e2.03ba    DYNAMIC     Fa0/3
```

Here are the MAC addresses of the computers:

- ComputerA: 000c.2928.5c6c
- ComputerB: 000c.29e2.03ba



When the link between SwitchA and SwitchC fails, SwitchC will use the fa0/16 interface immediately. However it will take 15 seconds for the topology change mechanism to age out the MAC address table!

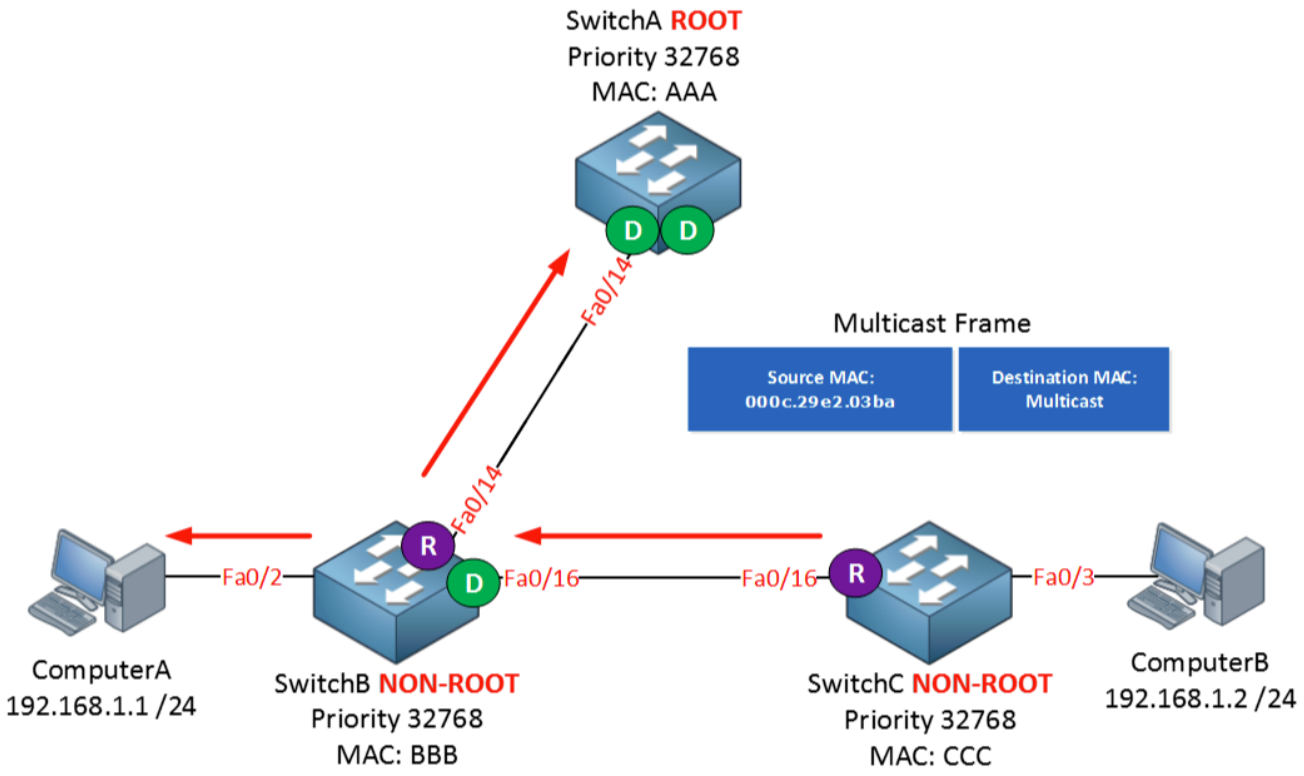
```
SwitchB#show mac address-table dynamic
          Mac Address Table
-----------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000c.2928.5c6c    DYNAMIC     Fa0/2
```

```
   1    000c.29e2.03ba    DYNAMIC     Fa0/14
```

Take a look again at the MAC address table for SwitchB. The MAC address (000c.29e2.03ba) that I highlighted belongs to ComputerB. When SwitchB receives an Ethernet Frame for ComputerB it will be forwarded to SwitchA and it will be dropped! (Well at least for 15 seconds until the topology change mechanism kicks in...).

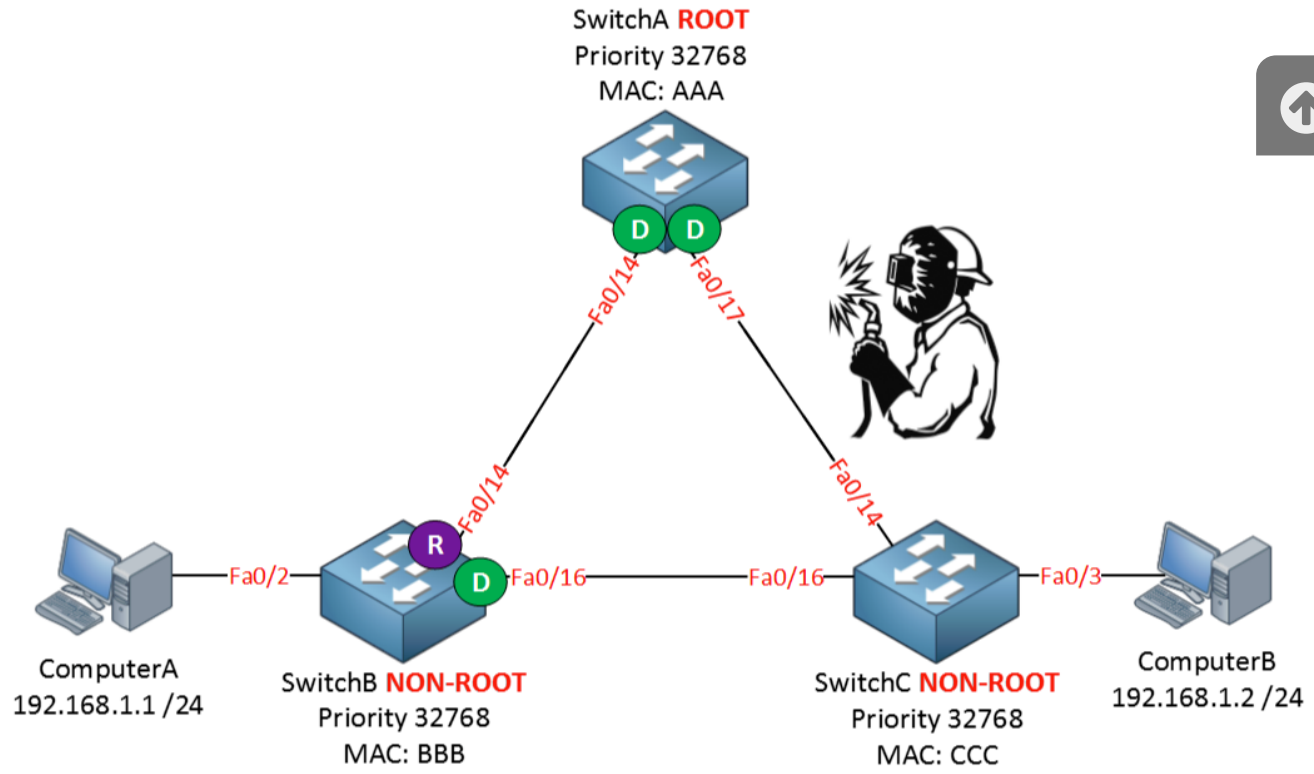
Of course we have a solution to speed this up, here's what we will do:



Once SwitchC switches over to use its non-designated port it will create a **dummy multicast frame.** The source MAC address of this Ethernet Frame will be **all the MAC addresses that it has in its MAC address table**. In my example above this is only the MAC address of ComputerB. The destination multicast address is a proprietary Cisco MAC address.

This multicast frame will be flooded to all other switches so they can update their MAC address tables right away.

Spanning-tree has saved the day again...anything else you need to know? What do you think will happen if I re-enable the fa0/14 interface on SwitchC again (the original root port)?




```
SwitchC(config)#interface fa0/14
SwitchC(config-if)#no shutdown
```

Let's bring the interface back up. In my lab it's not as exciting as in the picture...

```
SwitchC# STP: VLAN0001 Fa0/14 -> listening
SwitchC# STP: VLAN0001 Fa0/14: root port delay timer active
SwitchC# STP: VLAN0001 Fa0/14 -> blocking
SwitchC# STP: VLAN0001 new root port Fa0/14, cost 3019
SwitchC# STP: VLAN0001 Fa0/16 -> blocking (uplinkfast)
```

You can see we don't immediately switch back to interface fa0/14. There's no reason to switch back to this interface ASAP because we have a working root port. Even if we would switch back to interface fa0/14 right away we'd still have to wait because the fa0/17 interface on SwitchA will have to go through the listening and learning state (which takes 30 seconds).

That's all there is about spanning-tree uplinkfast. I hope this lesson has been useful!

Rate this Lesson:

« Previous Lesson
Spanning-Tree Portfast

Next Lesson
Spanning-Tree Backbone Fast »

Home › Forums › Spanning-Tree UplinkFast

This topic contains 24 replies, has 11 voices, and was last updated by Rene Molenaar 2 weeks, 4 days ago.

Viewing 15 posts - 1 through 15 (of 24 total)
1 2 →

- Author
  Posts | Subscribe
- February 16, 2015 at 22:23 #11362 Reply

Humberto A
Member
Hi Rene, excelent post.
However, I have some doubts about this topic. I read that using RSTP we have some improvements similar to Uplink and Backbone Fast. So, we don´t have to use this features on RSTP. Is that correct?

This output was taken from a Cat3750: “UplinkFast is enabled but inactive in rapid-pvst mode” Can you help me with this??

Whatever, your example is fantastic.

February 20, 2015 at 11:49 #11365 Reply

Rene Molenaar
Keymaster
Hi Humberto,

A long time ago, Cisco created Uplink Fast and Backbone Fast to “enhance” the old classic spanning-tree version (802.1D) . These two features were proprietary.

RSTP (802.1w) has something similar to Uplink fast and Backbone fast integrated in it so there's no need to enable Uplink / Backbone fast. This was only nice to use for the old 802.1D version of spanning-tree.

Does this help?

Rene

February 21, 2015 at 02:45 #11368 Reply

Humberto A
Member
Of course.
Thank you very much!!!

Humberto

August 31, 2015 at 00:44 #11369 Reply

Thomas K
Participant
Rene,
Hi. Can you please further clarify the following:

“BPDUs are originated from the root bridge so if we receive BPDUs on an interface the switch knows it can reach the root bridge on this interface. We have to go through the blocking (20 seconds), listening (15 seconds) and learning state (15 seconds) so it takes 50 seconds to end up in the forwarding state.

The good thing is that spanning-tree solves the link failure automatically but it also means that we have a downtime of 30 seconds. If you want you can tune the forward delay timer to speed up this process down to roughly 14 seconds.”

In the first part of the article when the link between A and C is severed what is the convergence time, 30 seconds or 50 seconds? I believe you mention that it will automatically switch over to the new root port, and then just wait the listening/learning (30 sec)?

Many thanks,

Thomas

Many thanks.

September 6, 2015 at 21:49 #11371 Reply

Rene Molenaar
Keymaster
Hi Thomas,

I just fixed this, it's 30 seconds. It has to go through the listening (15) and learning (15) states so it's 30 seconds in total.

Rene

October 11, 2015 at 13:34 #17989 Reply

Frades
Participant
great lessons so far.

i have a question, regarding the original root port that being brought up, once the original root port is up, will it also send a dummy multicast frame to switch A through F0/14? the traffic will still flow on (Fa0/16 of Switch C) because of the mac address table right?

October 13, 2015 at 18:03 #18835 Reply

Rene Molenaar
Keymaster
Hi John,

Good question, I think not. We don't switch back immediately to the "old" root port. Even if we would, interface fa0/17 on SwitchA still has to go through the listening and learning state which takes 30 seconds.

The dummy multicast is required because when the root port fails, we switch to the alternate port immediately.

Rene

October 24, 2015 at 06:43 #19166 Reply

Victor R
Participant
Hi,

I have a question regarding timers in a few pplaces I have seen that it can take up to 50 seconds before a port goes to forwarding. But in this case it only took 30 seconds. When does the 50 seconds come into play?

October 24, 2015 at 12:34 #19173 Reply

Rene Molenaar
Keymaster
Hi Victor,

It will take 50 seconds if the interface is currently in blocking mode (20 seconds).

Rene

February 21, 2016 at 16:05 #22264 Reply

Robert S
Participant
If I understand correctly, a non root switch ports can be either in designated mode (forwarding state) or in non-designated mode (blocking state). For a non-designated port it takes 50 seconds to become a designated port so in which scenario a non-designated port becomes a designated port (forwarding state) in 30 seconds.

Also, you mentioned that all ports start in blocking mode, do you mean during boot time?

Part that confuses me is the one where you said that when you plug a PC to a port on a swi takes for this port 30 seconds to start forwarding (listening plus learning) so in what mode this port was in before plugging PC into it?

February 22, 2016 at 11:41 #22282 Reply

Rene Molenaar
Keymaster
Hi Robert,

When you plug in a PC or anything then the interface only has to go through the listening and learning state, this will take 30 seconds in total before the interface is in forwarding mode.

Only a non-designated interface which is currently in blocking mode has to go through the blocking > listening > learning state. This takes 50 seconds in total. Where did you see the “all ports start in blocking mode” sentence? If I posted that somewhere then I’ll have to remove it, that’s confusing 🙂

Rene

March 16, 2016 at 18:08 #22760 Reply

Oscar S
Participant
In the first part of this article you says:
Now we’ll just wait for the magic to happen…
SwitchC#STP: VLAN0001 new root port Fa0/16, cost 38
SwitchC#STP: VLAN0001 Fa0/16 -> listening
SwitchC#STP: VLAN0001 Fa0/16 -> learning
SwitchC#STP: VLAN0001 Fa0/16 -> forwarding

BPDUs are originated from the root bridge so if we receive BPDUs on an interface the switch knows it can reach the root bridge on this interface. We have to go through the listening (15 seconds) and learning state (15 seconds) so it takes 30 seconds to end up in the forwarding state.

In this case, the port f0/16 is in blocking mode but you mentioned that the time is 30s and in the last answer you mentioned 50s, what is the difference?

March 16, 2016 at 20:38 #22762 Reply

Andrew P
Moderator
The difference is the MaxAge timer which is a default of 20 seconds. Only after the MaxAge expires will the listening/learning begin. If you add the maximum possible MaxAge to Listening/Learning, you wind up 50 seconds as the longest it would take.

April 21, 2016 at 20:31 #23541 Reply

Abdool Y
Participant
Thank you for this lesson in Spanning Tree UplinkFast, i wasn't clear if we could still use uplinkfast when using PVST+ until now.

April 22, 2016 at 16:26 #23583 Reply

Rene Molenaar
Keymaster
Hi Abdool,

You are welcome. You can still use it in PVST.

There's no need to use this in Rapid-PVST though. Rapid-PVST has similar mechanisms for backbonefast and uploadfast included "out of the box".

Rene

- Author
  Posts

Viewing 15 posts - 1 through 15 (of 24 total)
1 2 →
Reply To: Spanning-Tree UplinkFast

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:
Выберите файл Файл не выбран
Add another file

Submit

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!

Read my story

Social Fans

14,267
FANS

7,929
FOLLOWERS

1,589
SUBSCRIBERS

Highest Rated Lessons

MPLS Layer 3 VPN Configuration

(25 votes)

VRF Lite Configuration on Cisco IOS
(23 votes)

Cisco Portfast Configuration
(20 votes)

IPv6 Address Types
(18 votes)

EIGRP Stub Explained
(17 votes)

New Lessons

Introduction to Cisco IOS XE
ERSPAN Configuration on Cisco IOS XE
IGMP Filter
IGMP Snooping without Router
Cisco Group Management Protocol (CGMP)

Spanning-Tree UplinkFast written by Rene Molenaar average rating 4.6/5 - 14 user ratings