Networklessons .com

Search...

Table of Contents

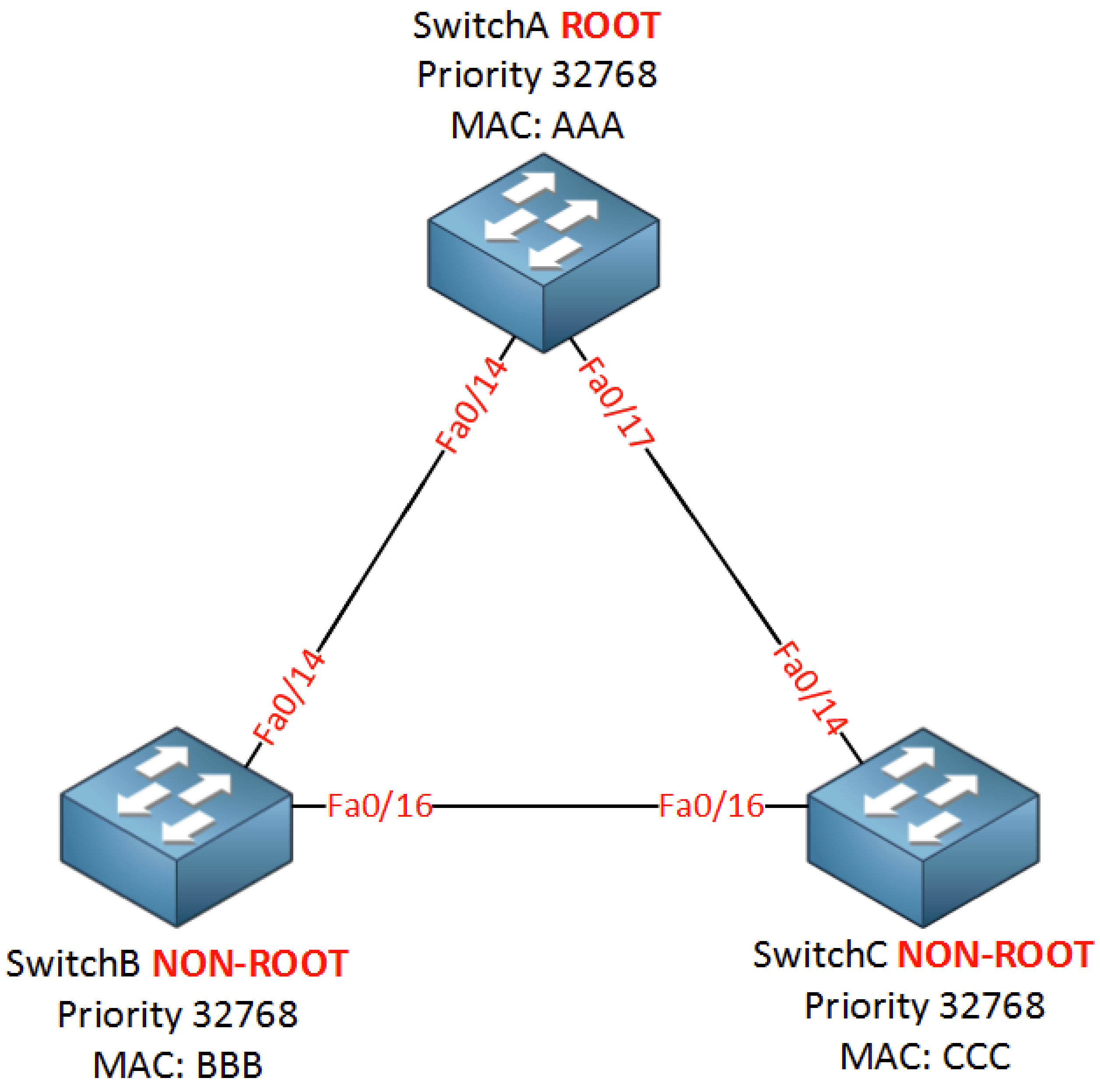
CCIE Routing & Switching

Rapid Spanning-Tree Configuration

10 votes

In a previous lesson I explained the differences between classic and rapid spanning-tree and how rapid spanning-tree works. If you haven't seen it before, I would recommend to look at it first before diving in the configuration.

Having said that, let's look at the configuration. This is the topology that I will use:



This is the topology I'm going to use. SwitchA will be the root bridge in my example. First we have to enable rapid spanning-tree:

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

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

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

That's it…just one command will enable rapid spanning tree on our switches. The implementation of rapid spanning tree is **rapid-pvst**. We are calculating a rapid spanning tree for each VLAN.

First I want to show you the sync mechanism:

```
SwitchA(config)#interface fa0/14
SwitchA(config-if)#shutdown
SwitchA(config)#interface f0/17
SwitchA(config-if)#shutdown
```

I'm going to shut both interfaces on SwitchA to start with.

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

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

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

Second step is to enable debug on all the switches.

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

I'm going to bring the fa0/14 interface back to the land of the living on SwitchA. Here's what we see:

```
SwitchA#
setting bridge id (which=3) prio 4097 prio cfg 4096 sysid 1 (on) id
1001.0011.bb0b.3600
RSTP(1): initializing port Fa0/14
RSTP(1): Fa0/14 is now designated
RSTP(1): transmitting a proposal on Fa0/14
```

The fa0/14 interface on SwitchA will be blocked and it'll send a proposal to SwitchB.

```
SwitchB#
RSTP(1): initializing port Fa0/14
RSTP(1): Fa0/14 is now designated
RSTP(1): transmitting a proposal on Fa0/14
RSTP(1): updt roles, received superior bpdu on Fa0/14
RSTP(1): Fa0/14 is now root port
```

Apparently SwitchB thought it was the root bridge because it says it received a superior BPDU on its fa0/14 interface. It changes its fa0/14 interface to root port.

```
SwitchB# RSTP(1): syncing port Fa0/16
```

The fa0/16 interface on SwitchB will go into sync mode. This is the interface that connects to SwitchC.

```
SwitchB#  RSTP(1): synced Fa0/14
RSTP(1): transmitting an agreement on Fa0/14 as a response to a proposal
```

SwitchB will respond to SwitchA its proposal by sending an agreement.

```
SwitchA# RSTP(1): received an agreement on Fa0/14
%LINK-3-UPDOWN: Interface FastEthernet0/14, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed
state to up
```

SwitchA receives the agreement from SwitchB and interface fa0/14 will go into forwarding.

```
SwitchB# RSTP(1): transmitting a proposal on Fa0/16
```

SwitchB will send a proposal to SwitchC.

```
SwitchC# RSTP(1): transmitting an agreement on Fa0/16 as a response to a
proposal
```

SwitchC will respond to the proposal of SwitchB and send an agreement.

```
SwitchB# RSTP(1): received an agreement on Fa0/16
%LINK-3-UPDOWN: Interface FastEthernet0/14, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed
state to up
```

SwitchB receives the agreement from SwitchC and the interface will go into forwarding. That's all there to is it…a quick number of handshakes and the interfaces will move to forwarding without the use of any timers. Let's continue!

```
SwitchA(config)#interface fa0/17
SwitchA(config-if)#no shutdown
```

I'm going to enable this interface so that connectivity is fully restored. Let's look at an overview:

```
SwitchA#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     0011.bb0b.3600
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    4097   (priority 4096 sys-id-ext 1)
             Address     0011.bb0b.3600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------------------
--
Fa0/14              Desg FWD 19        128.16   P2p
Fa0/17              Desg FWD 19        128.19   P2p
```

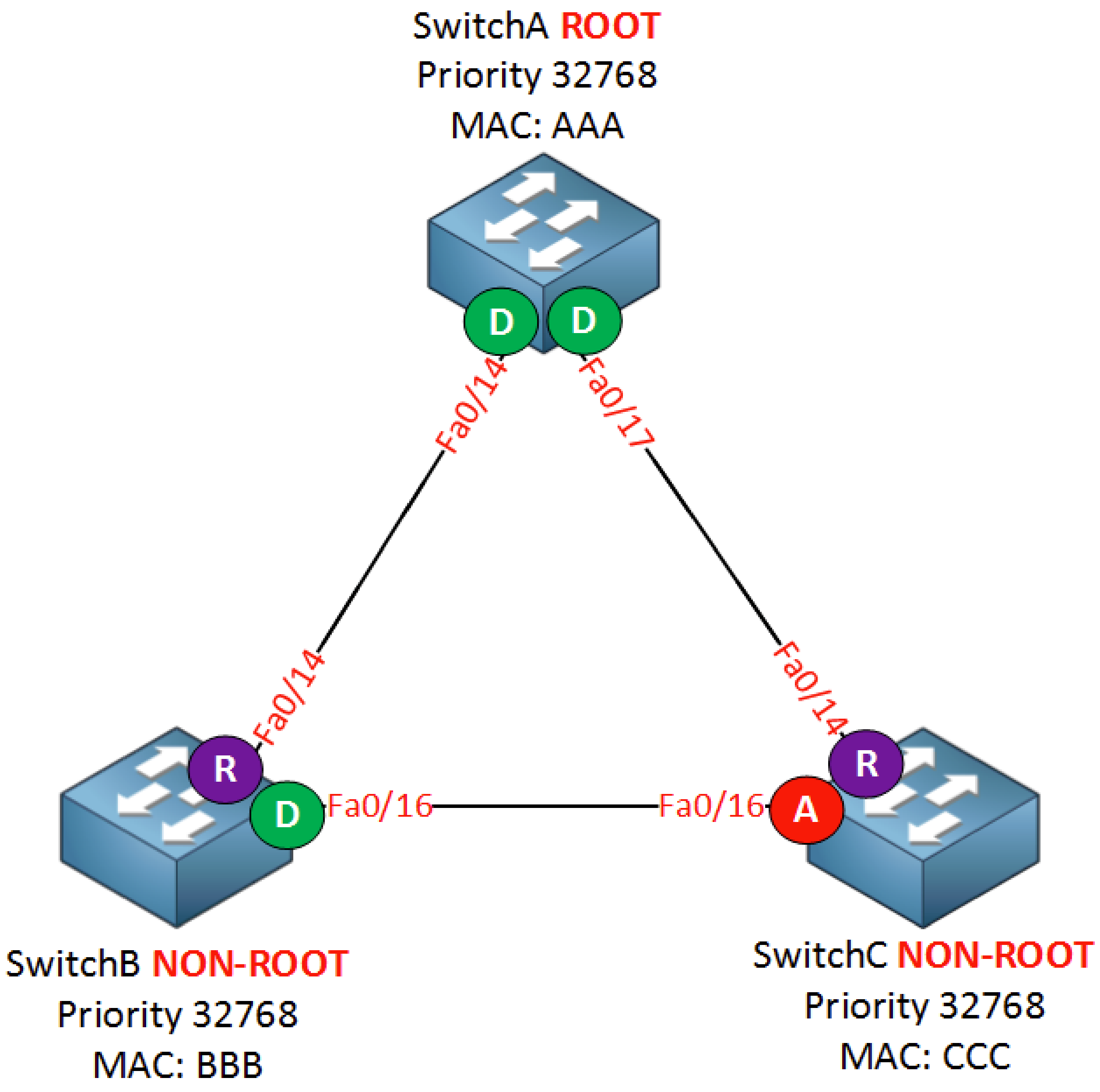
We can verify that SwitchA is the root bridge. This show command also reveals that we are running rapid spanning tree. Note that the link type is **p2p**. This is because my FastEthernet interfaces are in full duplex by default. Let's run the same command on the other two switches:

```
SwitchB#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     0011.bb0b.3600
             Cost        19
             Port        16 (FastEthernet0/14)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8193   (priority 8192 sys-id-ext 1)
             Address     0019.569d.5700
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------------------
--
Fa0/14              Root FWD 19        128.16   P2p
Fa0/16              Desg FWD 19        128.18   P2p
```

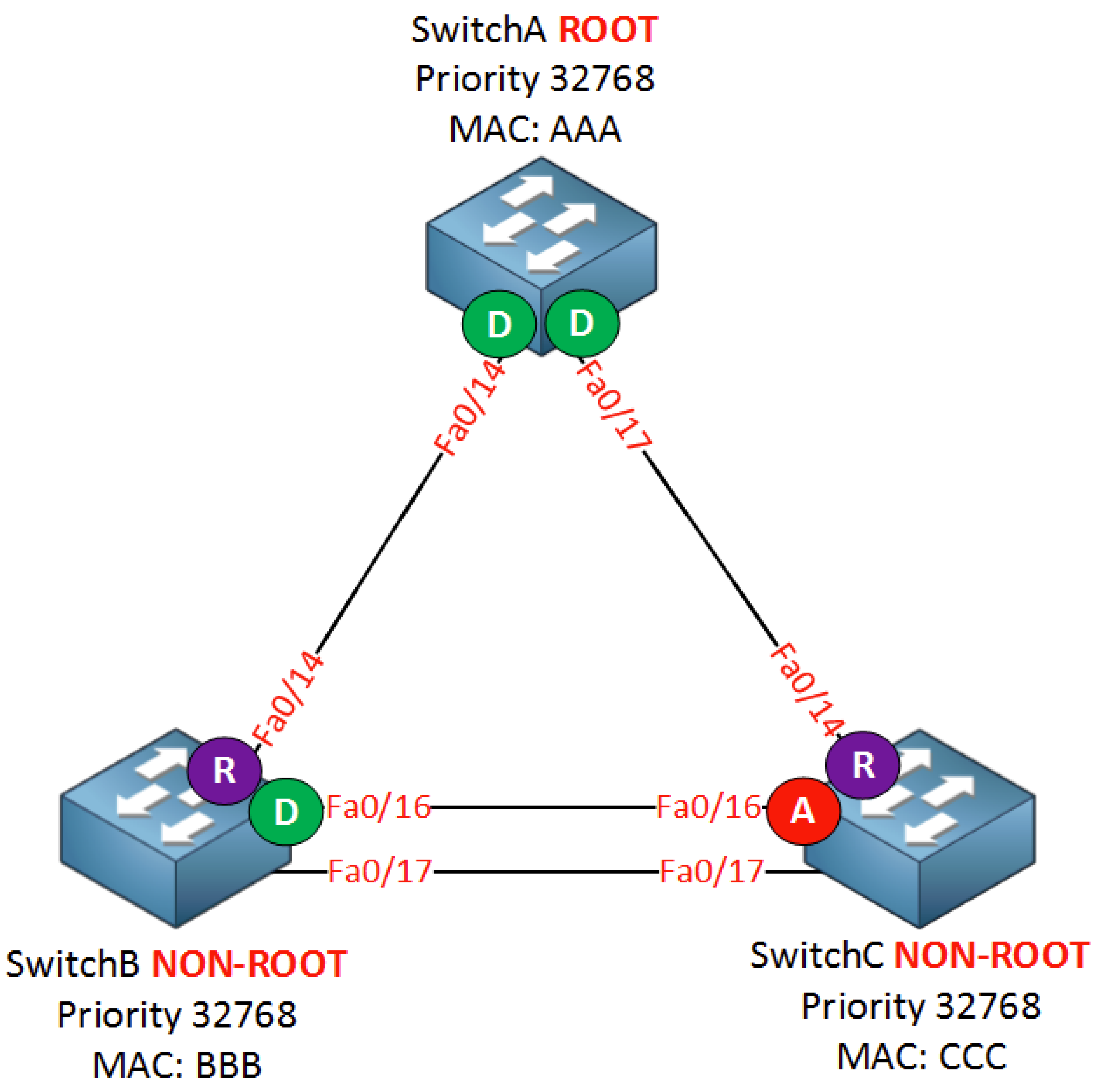
```
SwitchC#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    4097
             Address     0011.bb0b.3600
             Cost        19
             Port        14 (FastEthernet0/14)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000f.34ca.1000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
```

```
Fa0/14              Root FWD 19         128.14    P2p
Fa0/16              Altn BLK 19         128.16    P2p
```

Here are SwitchB and SwitchC. Nothing new here, it's the same information as classic spanning tree. Here's what the topology looks like now:



Let's add another link between SwitchB and SwitchC to see if this influences our topology:

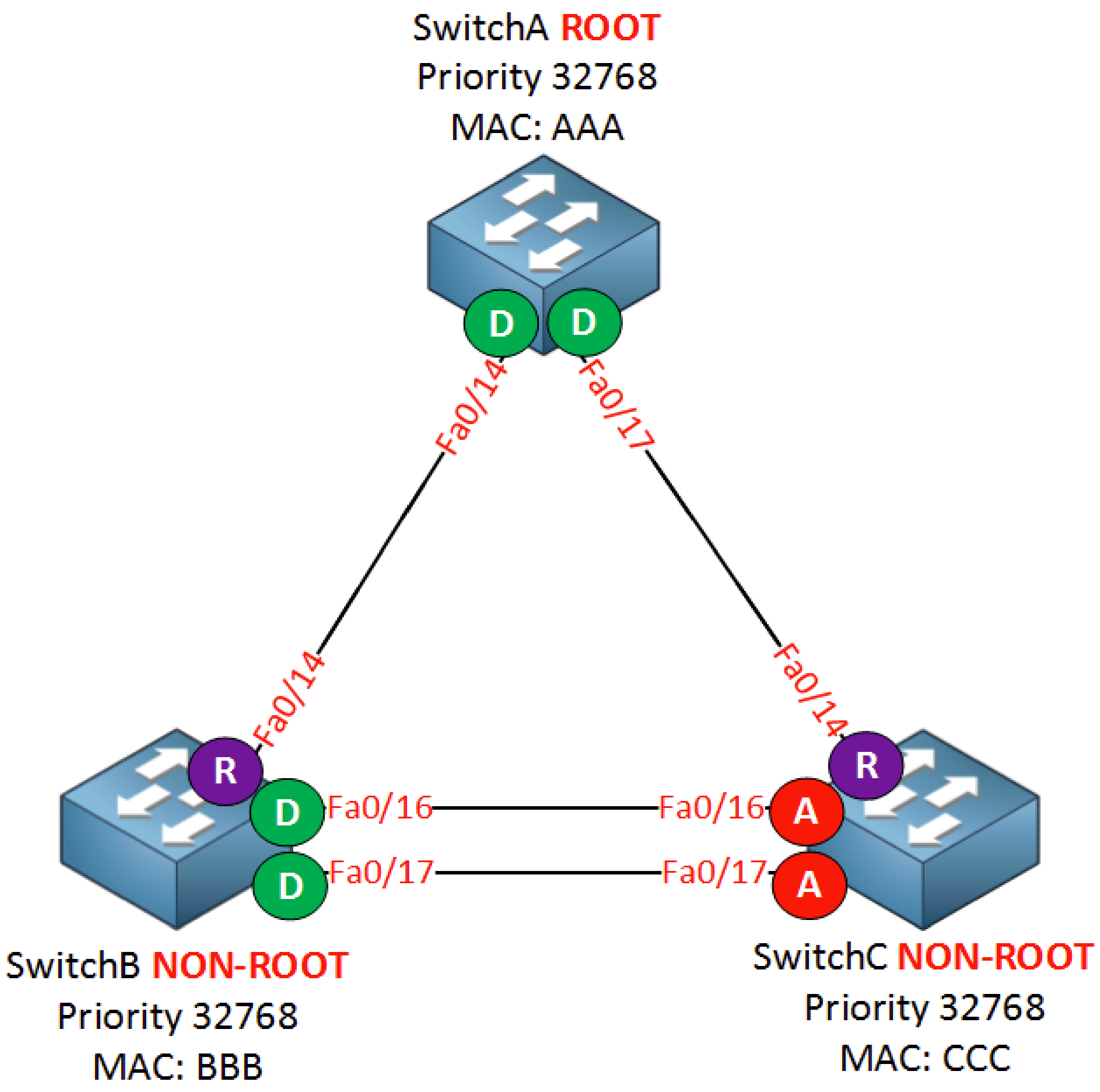


```
SwitchB#show spanning-tree | begin Interface
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------------------
--
Fa0/14              Root FWD 19        128.16   P2p
Fa0/16              Desg FWD 19        128.18   P2p
Fa0/17              Desg FWD 19        128.19   P2p
```

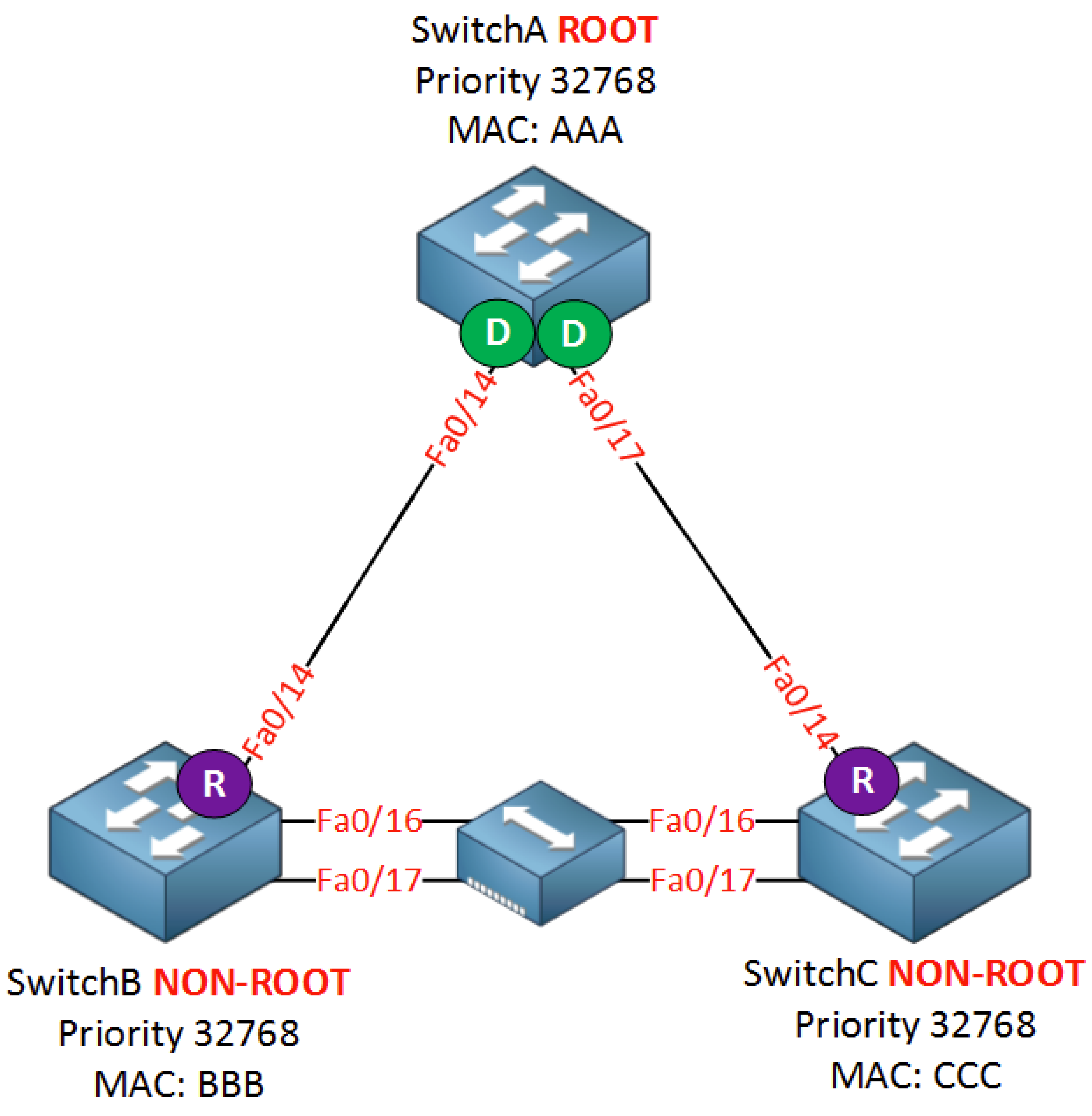
```
SwitchC#show spanning-tree | begin Interface
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/14           Root FWD 19        128.14   P2p
Fa0/16           Altn BLK 19        128.16   P2p
```

```
Fa0/17          Altn BLK 19        128.17   P2p
```

Nothing spectacular, we just have another designated port on SwitchB and another alternate port on SwitchC. Let me add that alternate port to the topology:



So far the topology with rapid spanning-tree looks the same as with classic spanning-tree. Now let me show you something you haven't seen before. I will add a hub between SwitchB and SwitchC:



Now take a look again at the interfaces:

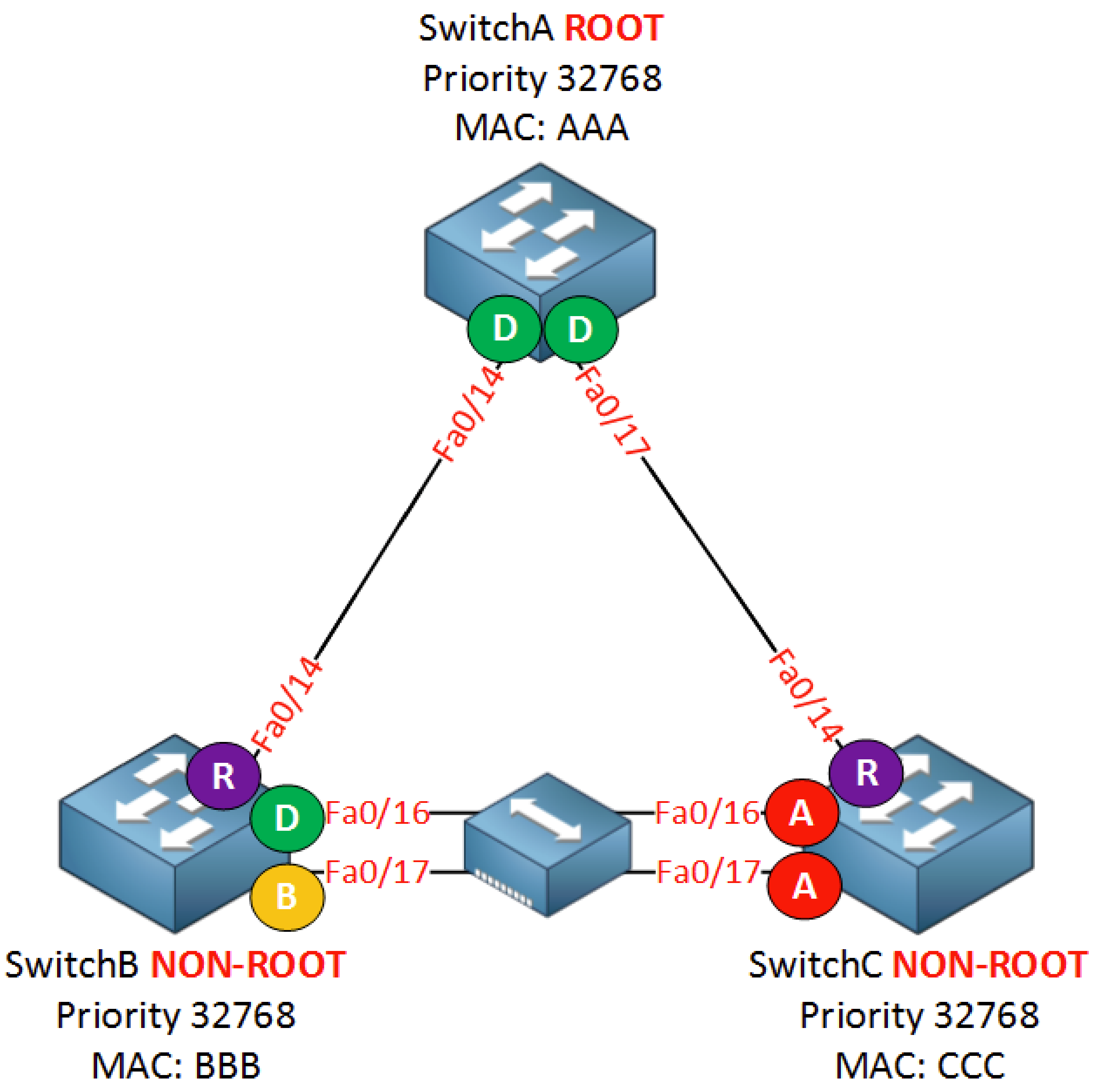
```
SwitchB#show spanning-tree | begin Interface

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------------------
--
Fa0/14               Root FWD 19        128.5    P2p
Fa0/16               Desg FWD 100       128.3    Shr
Fa0/17               Back BLK 100       128.4    Shr
```

```
SwitchC#show spanning-tree | begin Interface

```

```
Interface           Role Sts Cost      Prio.Nbr Type
--------- -------- --------------------------------
Fa0/14               Root FWD 19        128.5    P2p
Fa0/16               Altn BLK 100       128.3    Shr
Fa0/17               Altn BLK 100       128.4    Shr
```

Here's something new. SwitchB has a backup port. Because of the hub in the middle SwitchB and SwitchC will hear their own BPDUs.

You can also see that the link type is **shr (shared)**. That's because the hub causes these switches to switch their interfaces to half duplex. Here's the topology picture again:



You probably won't ever see the backup port on a production network since hubs are scarce now but if you see it, you'll know why…

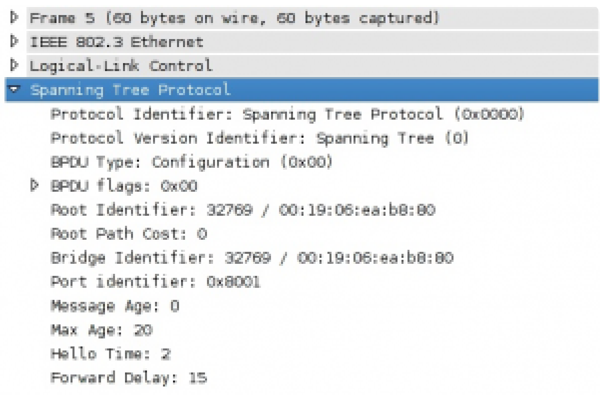
BPDUs are sent every two seconds (hello time) and if you want to prove this you can take a look at a debug:

```
SwitchB#debug spanning-tree bpdu
```

You can use the debug spanning-tree bpdu command to view BPDUs are are sent or received.

```
SwitchB#
STP: VLAN0001 rx BPDU: config protocol = rstp, packet from FastEthernet0/14  ,
linktype IEEE_SPANNING , enctype 2, encsize 17
STP: enc 01 80 C2 00 00 00 00 11 BB 0B 36 10 00 27 42 42 03
STP: Data
000002023C10010011BB0B360000000000010010011BB0B36008010000014000200 0F00
STP: VLAN0001 Fa0/14:0000 02 02 3C 10010011BB0B3600 00000000 10010011BB0B3600
8010 0000 1400 0200 0F00
RSTP(1): Fa0/14 repeated msg
RSTP(1): Fa0/14 rcvd info remaining 6
RSTP(1): sending BPDU out Fa0/16
RSTP(1): sending BPDU out Fa0/17
STP: VLAN0001 rx BPDU: config protocol = rstp, packet f
```

You will see the contents of the BPDU like above. It's not very useful if you want to see the content of the BPDU but it does show us that SwitchB is receiving BPDUs and sending them on its interfaces.

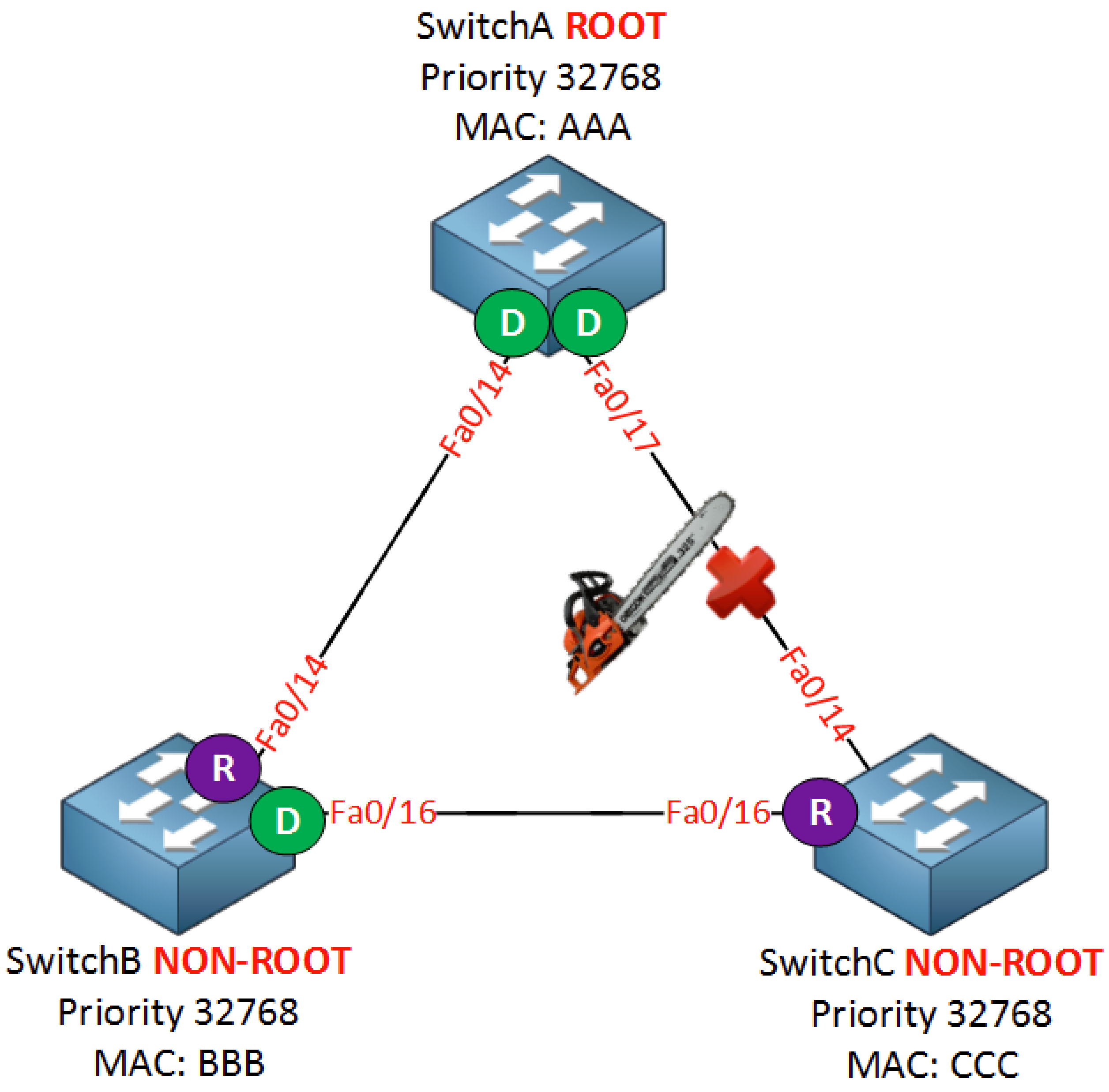


```
▷ Frame 5 (60 bytes on wire, 60 bytes captured)
▷ IEEE 802.3 Ethernet
▷ Logical-Link Control
▽ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
  ▷ BPDU flags: 0x00
    Root Identifier: 32769 / 00:19:06:ea:b8:80
    Root Path Cost: 0
    Bridge Identifier: 32769 / 00:19:06:ea:b8:80
    Port identifier: 0x8001
    Message Age: 0
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
```

If you do want to look at the contents of a BPDU I recommend you to use wireshark. It shows everything in a nice structured way.

You don't have to capture a BPDU yourself if you don't feel like. The wireshark website has many pre-recorded packet captures.

Let’s get rid of the hub and do something else…I’m going to simulate a link failure between SwitchA and SwitchC to see how rapid spanning tree deals with this.



```
SwitchA(config)#interface fa0/17
SwitchA(config-if)#shutdown
```

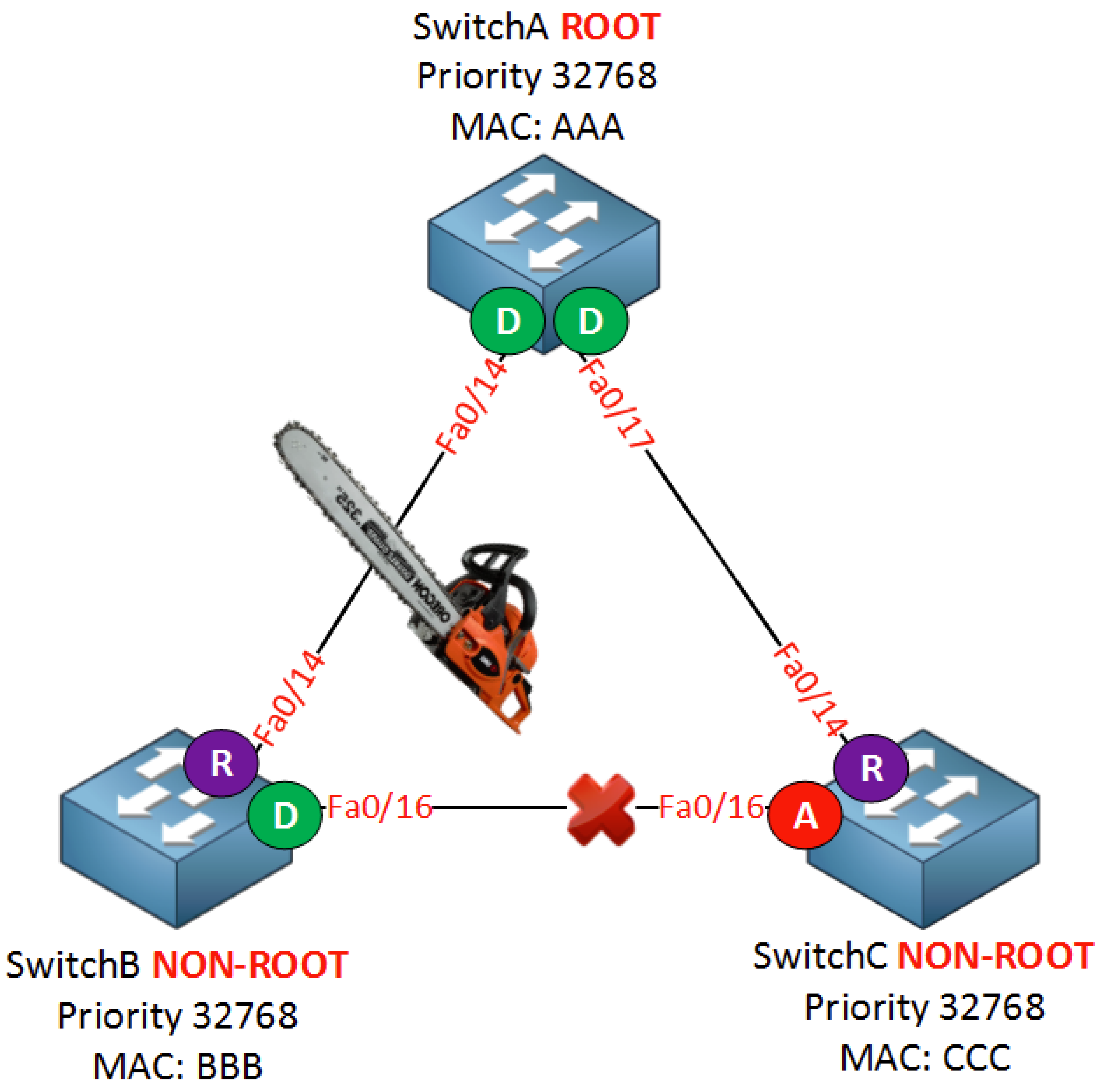
First I’m going to shut the fa0/17 interface on SwitchA.

```
SwitchC#
RSTP(1): updt rolesroot port Fa0/14 is going down
RSTP(1): Fa0/16 is now root port
```

SwitchC realized something is wrong with the root port almost immediately and will change the fa0/16 interface from alternate port to root port. This is the equivalent of UplinkFast for classic spanning tree but it's enabled by default for rapid spanning tree.

```
SwitchA(config)#interface fa0/17
SwitchA(config-if)#no shutdown
```

Let's restore connectivity before we continue.



Let's simulate an indirect link failure. The classic spanning-tree has backbone fast and a similar mechanism is enabled by default for rapid spanning tree.

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

Shutting down this interface will simulate an indirect link failure for SwitchC.

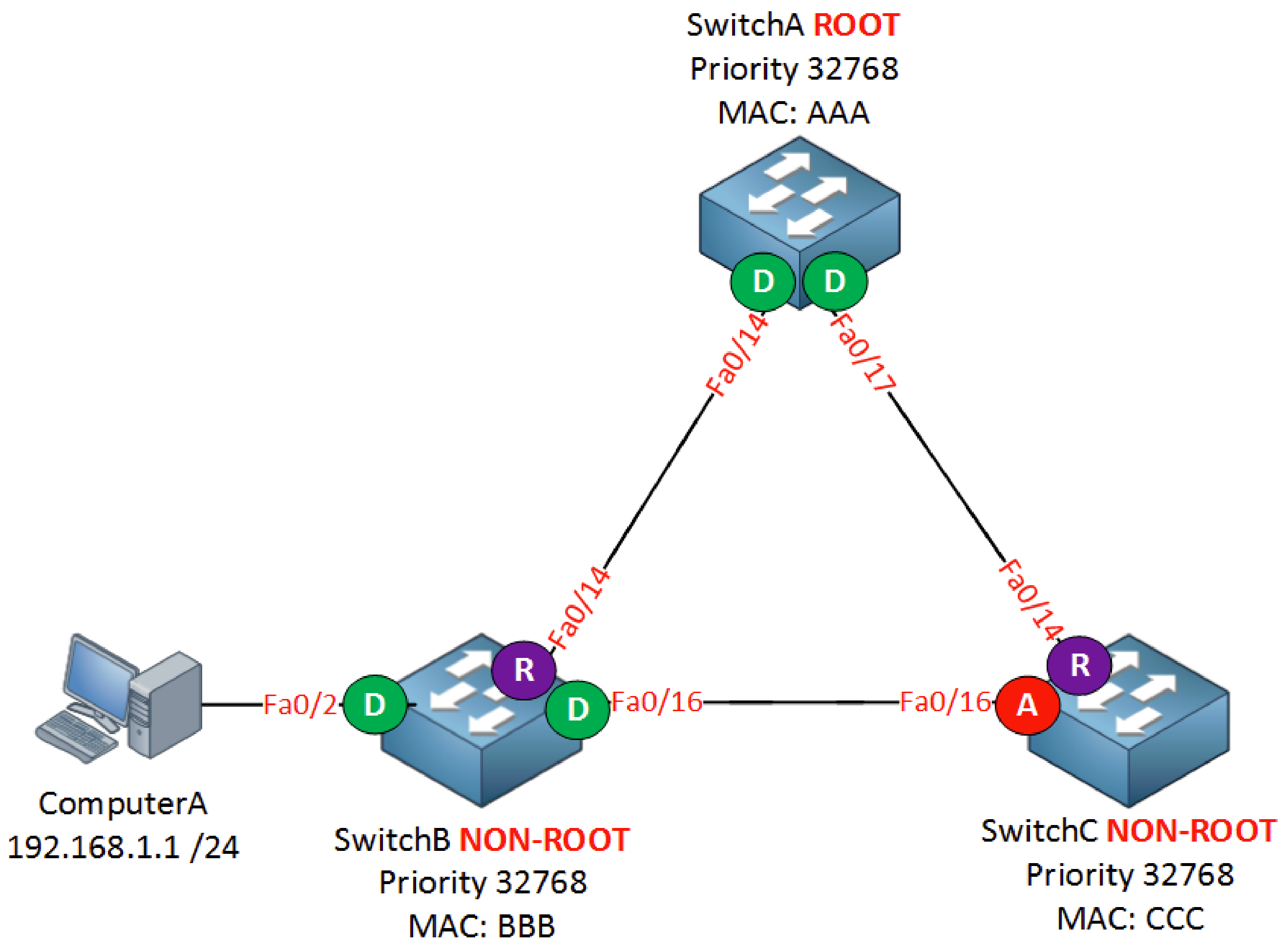
```
SwitchB#
RSTP(1): updt roles, root port Fa0/14 going down
RSTP(1): we become the root bridge
RSTP(1): updt roles, received superior bpdu on Fa0/16
RSTP(1): Fa0/16 is now root port
```

```
SwitchC#
03:41:29: RSTP(1): updt rolessuperior bpdu on Fa0/16 (synced=0)
03:41:29: RSTP(1): Fa0/16 is now designated
```

SwitchB believes it’s the root bridge until it receives a superior BPDU from SwitchC. This happens within the blink of an eye.

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

Let’s get rid of the shutdown command and continue...let's look at the edge ports:



I added ComputerA and it's connected to the fa0/2 interface of SwitchB. Let's see how rapid spanning tree deals with interfaces connected to other devices:

```
SwitchB(config)#interface fa0/2
SwitchB(config-if)#no shutdown
RSTP(1): initializing port Fa0/2
RSTP(1): Fa0/2 is now designated
RSTP(1): transmitting a proposal on Fa0/2
%LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
RSTP(1): transmitting a proposal on Fa0/2
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state
to up
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): Fa0/2 fdwhile Expired
```

You see that it sends a bunch of proposals from the sync mechanism towards the computer. After a while they will expire. The port will end up in forwarding state anyway but it takes a while.

```
SwitchB(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on FastEthernet0/2 but will only
 have effect when the interface is in a non-trunking mode.
```

You have to tell rapid spanning tree that the interface connecting the computer is an edge port. The word “edge” makes sense; it’s the border of our spanning tree topology. Enable portfast and you are ready to go.

```
SwitchB(config)#interface fa0/2
SwitchB(config-if)#shutdown
SwitchB(config-if)#no shutdown
```

I’ll bring the interface up and down.

```
SwitchB#
RSTP(1): initializing port Fa0/2
RSTP(1): Fa0/2 is now designated
*Mar  1 04:08:32.931: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state
to up
```

The interface will go to forwarding immediately. Our switch knows that this is the edge of the spanning tree and we don’t have to send proposals to it. The last thing we have to look at is compatibility…

I’m going to change SwitchB to PVST mode. SwitchA and SwitchC will remain at rapid-PVST:

```
SwitchB(config)#spanning-tree mode pvst
```

Here's what we see:

```
SwitchB(config)#
RSTP(1): updt roles, non-tracked event
setting bridge id (which=3) prio 8193 prio cfg 8192 sysid 1 (on) id
2001.0019.569d.5700
set portid: VLAN0001 Fa0/2: new port id 8004
STP: VLAN0001 Fa0/2 ->jump to forwarding from blocking
set portid: VLAN0001 Fa0/14: new port id 8010
STP: VLAN0001 Fa0/14 -> listening
set portid: VLAN0001 Fa0/16: new port id 8012
STP: VLAN0001 Fa0/16 -> listening^Z
STP: VLAN0001 heard root  4097-0011.bb0b.3600 on Fa0/16 supersedes  8193-
0019.569d.5700
STP: VLAN0001 new root is 4097, 0011.bb0b.3600 on port Fa0/16, cost 38
STP: VLAN0001 new root port Fa0/14, cost 19
STP: VLAN0001 Fa0/14 -> learning
STP: VLAN0001 Fa0/16 -> learning
STP: VLAN0001 sent Topology Change Notice on Fa0/14
STP: VLAN0001 Fa0/14 -> forwarding
STP: VLAN0001 Fa0/16 -> forwarding
```

SwitchB will throw some information at you. You can see that it receives BPDUs from the root bridge and that the interfaces will have to go through the listening and learning state. When switches that are talking rapid spanning tree receive a BPDU from the classic spanning tree will generate classic spanning tree BPDUs themselves so everything keeps working.

```
SwitchA#show spanning-tree | begin Interface
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
--
Fa0/14              Desg FWD 19        128.16   P2p Peer(STP)
Fa0/17              Desg FWD 19        128.19   P2p

SwitchB#show spanning-tree | begin Interface
Interface           Role Sts Cost      Prio.Nbr Type
```

```
------------------- ---- --- --------- -------- --------------------------------
--
Fa0/2               Desg FWD 19        128.4    P2p Edge
Fa0/14              Root FWD 19        128.16   P2p
Fa0/16              Desg FWD 19        128.18   P2p
```

```
SwitchC#show spanning-tree | begin Interface
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/14           Root FWD 19        128.14   P2p
Fa0/16           Altn BLK 19        128.16   P2p Peer(STP)
```

We can verify this by looking at the interfaces again. All switches still agree on the port states and everything will function as it should be!

That's all there is about rapid spanning-tree. The configuration is pretty simple but I hope the debugs and show commands helped to understand exactly how everything works.

Rate this Lesson:

★★★★★

« Previous Lesson
Rapid Spanning-Tree

Next Lesson »
MST (Multiple Spanning-Tree)

Home › Forums › Rapid Spanning-Tree Configuration

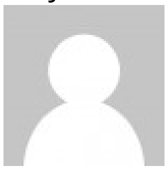
This topic contains 8 replies, has 5 voices, and was last updated by Andrew P 1 month, 2 weeks ago.

Viewing 8 posts - 1 through 8 (of 8 total)

- Author

Posts | Subscribe

- May 12, 2015 at 15:58 #11373 Reply



Betar R
Member
Nice explanation, Rene.

You nicely described in STP, how the affected switch updated CAM table.

When a link fails (Uplink/Backbone) in RSTP environment, how the affected switch will collect MAC?

May 14, 2015 at 17:15 #11374 Reply

Rene Molenaar
Keymaster
Hi Betar,

Here’s an example how the MAC addresses are updated for uplink failures:

https://networklessons.com/spanning-tree/spanning-tree-uplinkfast/

Uplinkfast can be enabled for “classic” STP but a similar mechanism is built-in RSTP.

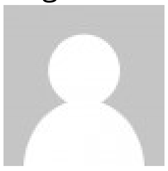
Backbone fast is also a good read:

http://networklessons.com/cisco/spanning-tree-backbone-fast/

A similar mechanism is built in for RSTP.

Rene

August 31, 2015 at 01:05 #11375 Reply



Thomas K
Participant

Rene,
Hi. In the last part of the lesson you convert switch B to PVST mode – what is the impact to the network – does computer A lose connectivity to the rest of the network outside of switch B for 30 seconds? What about if there was a host connected off of switch A and another host connected off switch B – they would still continue to communicate throughout the change of switch B to PVST mode?

Many thanks,
Thomas

September 6, 2015 at 22:02 #11376 Reply

Rene Molenaar
Keymaster
Hi Thomas,

I just tried this, 3 switches and two hosts (one on SW1 and SW2). If you switch to any of the STP mode, you have donwtime. When I change one switch from RPVST to PVST+ then you see the interfaces going through the listening and learning states again.

When all switches are running PVST+ and I turn one of them into RPVST, you see it sending and waiting for proposals. It's all compatible but expect your interfaces to be blocked for a moment 🙂

Rene

May 4, 2016 at 19:57 #23851 Reply

SHANMUGASIVA K
Participant
Hi Rene,

Pasting the debug msgs:

```
SwitchA#
setting bridge id (which=3) prio 4097 prio cfg 4096 sysid 1 (on) id
1001.0011.bb0b.3600
```

what does it mean ? I'm not able to understand.

May 4, 2016 at 21:24 #23852 Reply

Andrew P
Moderator
Sorry, Shanmugasiva,
I am not understanding what you are asking. Could you ask this another way?

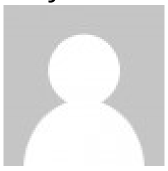
If you are asking what this message means, I can try to help.
```
setting bridge id (which=3) prio 4097 prio cfg 4096 sysid 1 (on) id
1001.0011.bb0b.3600
```

"cfg 4096"
This means you set (either via spanning-tree vlan 1 4096, or spanning-tree vlan 1 root primary). This is what it means when it says "prio cfg 4096"–the vlan was configured to be priority 4096.

"prio 4097"
However, the actual priority is the configured priority + vlan number. So, in this case, the actual priority is 4097 (4096 + 1, for vlan 1). This is what it means for "prio 4097"

"(on) id 1001.0011.bb0b.3600"
This is the MAC address of the switch where this command was issued.

May 5, 2016 at 03:50 #23855 Reply



SHANMUGASIVA K
Participant
Thanks Andrew!

I meant to ask about the priority 4096 and 4097.

From the picture, Switch A has the priority 32768. Why do we represent 4096 here ?

May 5, 2016 at 18:07 #23872 Reply

Andrew P
Moderator

Hi Shanmugasiva,
I suspect this simply might be a case of label in the picture not matching the configured IOS. In one of the outputs from Switch A it says "This bridge is the root" and it lists the Root ID of 4097.

Notice, the same thing is true for Switch B. It says in the text output it says its priority is 8193, but the label in the picture shows 32768.

Good catch!

–Andrew

- Author
  Posts

Viewing 8 posts - 1 through 8 (of 8 total)
Reply To: Rapid Spanning-Tree Configuration

b i 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)