Networklessons.com

Search...

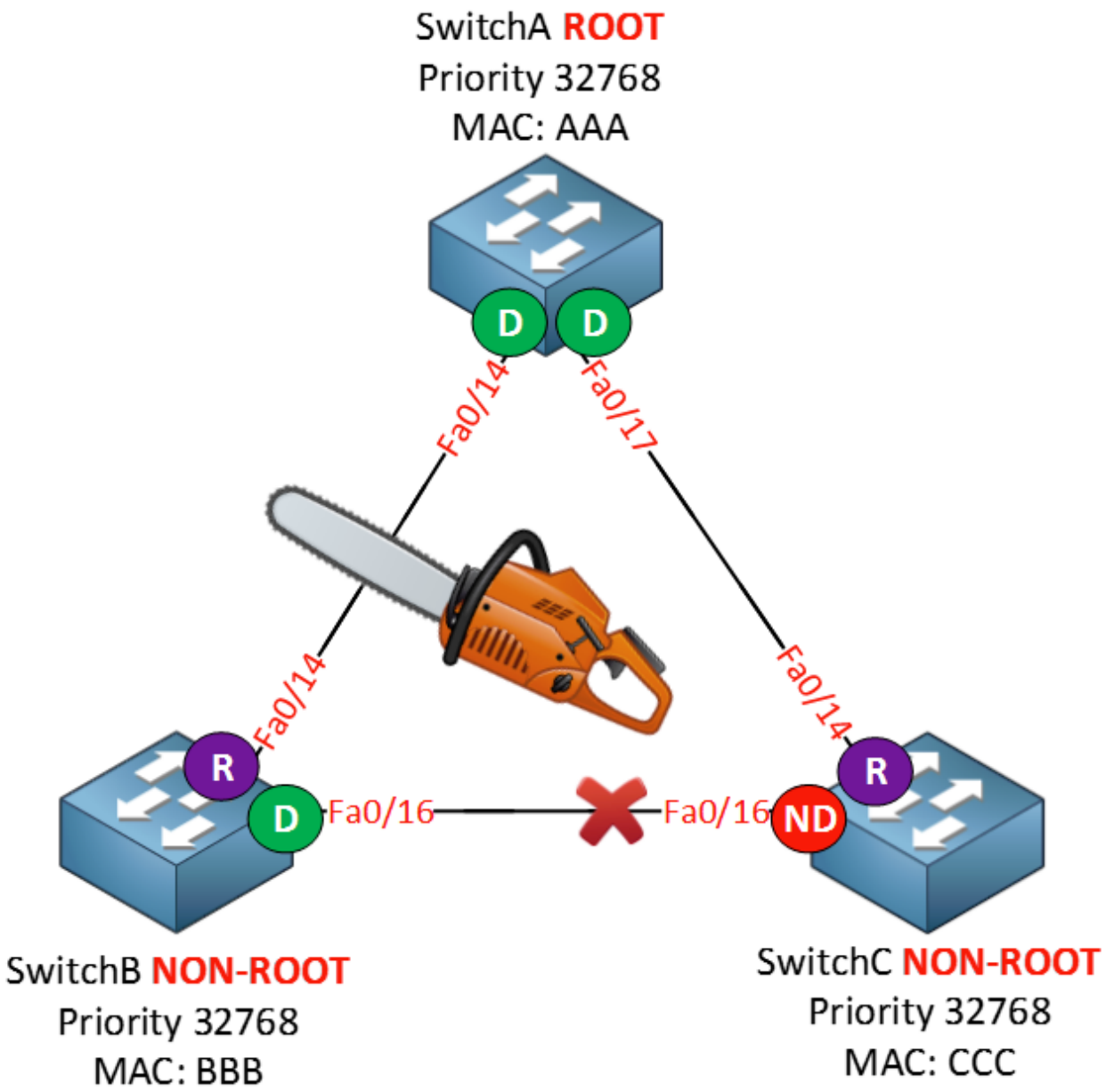
Table of Contents

CCIE Routing & Switching

Spanning-Tree Backbone Fast

9 votes

Backbone Fast is used to recover from an indirect link failure. What does that mean? Let me show you an example of an indirect link failure and how spanning-tree deals with it:



Take a look at the picture above. SwitchA is the root bridge and the fa0/16 interface on SwitchC has been blocked. Suddenly the link between SwitchA and SwitchB fails. From SwitchC's perspective this is **an indirect link failure**.

This is what will happen:

1. SwitchB will detect this link failure immediately since it's a directly connected link. Since it doesn't receive any BPDUs from the root anymore it assumes it is now the new root bridge and will send BPDUs towards SwitchC claiming to be the new root.
2. SwitchC will receive these BPDUs from SwitchB but it will realize that this new BPDU is **inferior** compared to the old one it has currently stored on its fa0/16 interface and will **ignore this new BPDU**. When a switch receives an inferior BPDU it means that the neighbor switch has lost its connection to the root bridge.
3. After 20 seconds (default timer) the max age timer will expire for the old BPDU on the fa0/16 interface of SwitchC. The interface will go from blocking to the listening state and will send

BPDUs towards SwitchB.

4. SwitchB will receive this BPDU from SwitchC and discovers that he isn't the root bridge. It won't send BPDUs anymore towards SwitchC.
5. The fa0/16 interface on SwitchC will continue from the listening state (15 seconds) to the learning state (15 seconds) and ends up in the forwarding state.

Connectivity is now restored but it took 20 seconds for the max age timer to expire, 15 seconds for the listening state and another 15 seconds for the learning state before we go to the forwarding state. That's a total of 50 seconds downtime. Let's take a look at this situation on our switches:

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

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

Let's enable our debugging.

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

I will shut this interface to simulate an indirect link failure.

```
SwitchB# STP: VLAN0001 we are the spanning tree root
```

SwitchB believes it is the root bridge.

```
SwitchC# STP: VLAN0001 heard root  8193-0019.569d.5700 on Fa0/16
```

SwitchC receives the BPDUs from SwitchB who claims to be the root bridge.

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

After the max age timer expires (20 seconds) for the old BPDU from SwitchB the fa0/16 interface on SwitchC will go to the listening and learning state and ends up in forwarding state.

```
SwitchB# STP: VLAN0001 heard root 4097-0011.bb0b.3600 on Fa0/16
SwitchB# STP: VLAN0001 new root is 4097, 0011.bb0b.3600 on port Fa0/16, cost
38
```

The identity crisis of SwitchB comes to an end. It now hears the BPDUs from the root bridge through SwitchC and understands that it's not the root bridge.

Without backbone fast, spanning-tree will discard the inferior BPDUs that SwitchC receives on its fa0/16 interface and it will have to wait till the max age timer expires (20 seconds).

If we enable backbone fast it will skip the max age timer so we can save 20 seconds of time.

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

Let's enable the fa0/14 interface on SwitchA first.

```
SwitchA(config)#spanning-tree backbonefast
```

```
SwitchB(config)#spanning-tree backbonefast
```

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

Let's enable backbone fast on all switches. This is a global command (spanning-tree backbonefast).

```
SwitchA#debug spanning-tree backbonefast detail
Spanning Tree backbonefast detail debugging is on
```

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

```
SwitchC#debug spanning-tree backbonefast detail
Spanning Tree backbonefast detail debugging is on
```

Use the debug spanning-tree backbonefast detail command to see real-time information on backbone fast.

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

Let's simulate the indirect link failure again...

```
SwitchB# STP: VLAN0001 we are the spanning tree root
```

SwitchB loses its connection to the root bridge and assumes he is now the new root bridge. Nothing new so far...

```
SwitchA# STP FAST: VLAN0001 FastEthernet0/17: sending requested RLQ response
PDU
```

SwitchA receives a new packet called a (RLQ) Root Link Query from SwitchC. As soon as SwitchC receives an inferior BPDU it will send a root link query on its root port and non-designated ports to check if the root bridge is still available.

```
SwitchC# STP FAST: received RLQ response PDU was expected on VLAN0001
FastEthernet0/14 - resp root id 4097-0011.bb0b.3600
```

SwitchC receives a reply to its root link query on the fa0/14 interface to SwitchA.

```
SwitchC# STP FAST: received_rlq_bpdu on VLAN0001 FastEthernet0/16 - making
FastEthernet0/16 a designated port
```

Because SwitchC received a response from the root bridge on its fa0/14 interface it can now skip the max age timer on its f0/16 interface and the interface goes to the listening and learning state right away. We effectively save 20 seconds (max age timer).

Take a good look at the last debug output from SwitchC. It doesn't say that it received something on the fa0/16 interface, it means that it received an answer to the root link query that it did because it received an inferior BPDU on the fa0/16 interface. That's all there is to backbone fast, I hope you enjoyed this lesson.

Rate this Lesson:

« Previous Lesson
Spanning-Tree UplinkFast

Next Lesson
Rapid Spanning-Tree »

Home › Forums › Spanning-Tree Backbone Fast

This topic contains 10 replies, has 6 voices, and was last updated by Andrew P 1 month, 1 week ago.

Viewing 10 posts - 1 through 10 (of 10 total)

- Author
  Posts | Subscribe
- June 21, 2015 at 09:18 #11356 Reply

Srinivasan C
Participant
Hi Rene,
In my understanding , ND port can only receive BPDU . With Backbone fast enabled,
How does switch C send RLQ on both the Root port and ND port ? RLQ is not a BPDU frame ?
Please explain RLQ.

Thanks,
Srini

June 21, 2015 at 16:15 #11357 Reply

Rene Molenaar
Keymaster
Hi Srini,

The RLQ is not send on the ND port, only the root port. The last message is a bit cryptic, it means that SwitchC received a reply on its root port that it sent because it received an inferior BPDU on its ND port.

Rene

June 21, 2015 at 16:49 #11358 Reply

Srinivasan C
Participant
Hi Rene,
Thanks for explainig RLQ in RSTP.

Srini.

July 3, 2015 at 23:31 #11359 Reply

Atif S
Participant
Uplinkfast and Backbone fast are they used together in conjunction?

July 5, 2015 at 18:49 #11360 Reply

Rene Molenaar
Keymaster
In the past yes, these were used for PVST.

RPVST has similar built-in mechanisms so you don't have to enable these nowadays.

October 11, 2015 at 14:04 #17993 Reply

Frades
Participant
let me clarify this, correct me if im wrong. this is what i understand on this lesson

- Link between Sw A and Sw B is down. so Switch B will not receive any BPDU's from root bridge, claiming he's the root bridge.
- since the link between A and B is down, Switch B's BPDU will send through Switch C.
- Now switch C receives this BPDU and identifies it as inferior BPDU.
- Since it receives an inferior BPDU, Switch C will send a Root Link Query on its Root Port to check if the root bridge is still available (this is the purpose of RLQ? to check if root bridge is available? because Switch B is claiming he's the new root bridge, so Switch C just want to verify if root bridge is still alive)
- now, if Switch C receives a response on his RLQ, it means the root bridge is alive right? then on Switch C's block port, it will skip the max age timer (20secs) and it will go straight listening mode then learning.

is this right? if this is right, then i have a question,

what if Switch C sends a RLQ and didnt receive a response? what will happened? Switch B will be the new root bridge? what is the process?

or i think it will not go through this long process, because not receiving a RLQ response means the RootBridge is dead right? it means directly connected port of Switch C to RootBridge will also detect a link failure, also Switch B will detect a link failure.

October 11, 2015 at 14:27 #17997 Reply

Frades
Participant
i think i get it now, i reread it. so without backbonefast, Switch C will drop the inferior BPDU produced by Switch B. since the link A-B is down, Switch C f0/16 is not receiving the old BPDU so it will wait for the max age time to expire (20secs), then it will now go to listening and learning state.

when backbonefast is enabled, switch C will not drop/ignore the inferior BPDU, C will send a RLQ to the root bridge, then if C receives a response, it means root bridge is alive, then it will skip the max age time on F0/16 then will go to listening and learning, then in forwarding state, C will inform B that Root bridge is alive.

October 13, 2015 at 18:14 #18836 Reply

Rene Molenaar
Keymaster
Hi John,

All steps that you described are correct, that's how it works.

SwitchB doesn't have a link to the root bridge so it claims to be the new root bridge. SwitchC which is still connected to the root doesn't agree so it drops the inferior BPDU from SwitchB.

We want to get rid of the max age timer, that's 20 seconds so instead we let SwitchC do a quick "check" to see if the root bridge is still reachable by sending a RLQ. When it receives a RLQ reply, it knows the root is still there and it can age out the interface immediately.

In case you don't get a RLQ reply, we know the root bridge is gone and we'll have to recompute STP.

Rene

May 10, 2016 at 09:27 #24000 Reply

Mohammad Hasanuz Z
Participant
Hlw Rene,

SwitchC will receive these BPDUs from SwitchB but it will realize that this new BPDU is inferior compared to the old one it has currently stored on its fa0/16 interface.

From above the line of you.....

Switch C will receive inferior BPDU (Best Bridge ID) on port Fa0/16 ?? If so then How ?? In My understanding, Switch A Originate BPDU and send to Down side and down side switch will carry the BPDU to other by using its Designated port . So Switch C will get Root Originated BPDU on its ports Fa0/14 and Fa0/16 and stored on port .

When SWA to SWB link down then SWB originated BPDU and send to SWC .If SWB originate BPDU then How its Lowest Bridge ID than Old one on port SWC fa0/16 ??

Please correct me if my understanding is wrong .Thanks

br//
zaman

May 10, 2016 at 18:37 #24009 Reply

Andrew P
Moderator
Mohammad,
Switch C receives an inferior BPDU (meaning a worse BPDU) on Fa0/16. This worse BPDU was created by Switch B because after Switch B's link to Switch A went down, Switch B now believes it is the Root Bridge. Since it is the job of the Root Bridge to create BPDUs, Switch B is doing what it thinks is the proper activity. However, Switch C knows that Switch B is mistaken. It knows this because Switch C is still able to receive superior BPDUs (meaning better) from the true Root Bridge, Switch A.

Bridge ID is determined by a combination of Priority and MAC Address. The lower the Bridge ID, the better–as far as being elected as the Root Bridge. Since in this example all switches have the same priority, it is the MAC address (the lowest MAC) which determines the Root Bridge. Switch C sees BPDUs coming from "BBB" and from "AAA" both claiming to be root. Switch B knows that AAA is lower than BBB, so it ignores BBB.

After 20 seconds of not receiving any more BPDUs from AAA on its port Fa0/16, Switch C transitions from Blocking to Listening to Learning to Forwarding. Only after all that happens (50 seconds, maximum), will Switch C send an "AAA" BPDU to Switch B. It is at this point that Switch B realizes its mistake (that it shouldn't have claimed to be the Root Bridge). During this entire 50 second process, user data cannot leave switch B, so there would be an outage. Backbone-Fast is meant to minimize this outage.

- Author
  Posts

Viewing 10 posts - 1 through 10 (of 10 total)
Reply To: Spanning-Tree Backbone Fast

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)

Spanning-Tree Backbone Fast written by Rene Molenaar average rating 5/5 - 9 user ratings