

# Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Dépannez](#)

[Racine STP](#)

[Interface STP](#)

[Enquête BPDU avec Ethalyzer](#)

[Convergence STP](#)

[Mappage externe VLAN](#)

[Debugs STP](#)

[Le Nexus 5000 n'a pas traité des BPDU](#)

## Introduction

Ce document décrit de diverses méthodes pour dépanner des problèmes courants liés au Protocole Spanning Tree (STP).

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Système d'exploitation CLI de Nexus
- STP

### [Composants utilisés](#)

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Dépannez

Cette section couvre quelques méthodes pour dépanner des problèmes courants avec STP.

## Racine STP

Afin de dépanner une question STP, il est essentiel de connaître quel commutateur est actuellement la racine. La commande d'afficher la racine STP sur un commutateur de gamme de Nexus 5000 est :

```
Nexus-5000# show spanning-tree vlan 1

VLAN0001
Spanning tree enabled protocol rstp
Root ID Priority 32769
Address c84c.75fa.6000
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
Address c84c.75fa.6000
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

Voici quelques autres commandes appropriées :

```
Nexus-5000# show spanning-tree vlan 1 detail
Nexus-5000# show spanning-tree vlan 1 summary
```

Une fois que vous avez déterminé de qui la racine en cours est, vous pouvez vérifier l'historique d'événement pour voir s'il a changé et d'où les notifications de modification de topologie proviennent.

```
Nexus-5000# show spanning-tree internal event-history tree 1 brief
2012:11:06 13h:44m:20s:528204us T_EV_UP
VLAN0001 [0000.0000.0000.0000 C 0 A 0 R none P none]
2012:11:06 13h:44m:21s:510394us T_UT_SBPDU
VLAN0001 [8001.547f.ee18.e441 C 0 A 0 R none P Po1]
2012:11:06 13h:44m:21s:515129us T_EV_M_FLUSH_L
VLAN0001 [1001.001b.54c2.5a42 C 6 A 5 R Po1 P none]
2012:11:06 13h:44m:23s:544632us T_EV_M_FLUSH_R
VLAN0001 [1001.001b.54c2.5a42 C 6 A 5 R Po1 P Po1]
2012:11:06 13h:44m:24s:510352us T_EV_M_FLUSH_R
VLAN0001 [1001.001b.54c2.5a42 C 6 A 5 R Po1 P Po1]
```

**Conseil :** Voici quelques définitions pour les acronymes qui apparaissent dans la sortie des commandes. **SBPDU** : Bridge Protocol Data Unit supérieur reçu ; **FLUSH\_L** : Annulation de gens du pays ; **FLUSH\_R** : Videz du commutateur distant.

Remarque: Les versions NX-OS avant la version 5.1(3)N1(1) ne se connectent pas plus de 149 événements, et le log ne roule pas.

## Interface STP

Cette commande est utilisée afin d'afficher les événements pour une interface.

```
Nexus-5000# show spanning-tree internal event-history tree 1 interface
ethernet 1/3 brief
2012:11:05 13h:42m:20s:508027us P_EV_UP Eth1/3 [S DIS R Unkw A 0 Inc no]
```

```
2012:11:05 13h:42m:20s:508077us P_STATE Eth1/3 [S BLK R Desg A 0 Inc no]
2012:11:05 13h:42m:20s:508294us P_STATE Eth1/3 [S LRN R Desg A 0 Inc no]
2012:11:05 13h:42m:20s:508326us P_STATE Eth1/3 [S FWD R Desg A 0 Inc no]
```

Cette commande est utilisée afin d'étudier des modifications STP sur une interface. Cette sortie offre beaucoup de détails :

```
Nexus-5000# show spanning-tree internal info tree 1 interface port-channel 11
----- STP Port Info (vdc 1, tree 1, port Po11) -----
dot1d info: port_num=4106, ifi=0x1600000a (port-channell1)
ISSU FALSE non-disr, prop 0, ag 0, flush 0 peer_not_disputed_count 0
if_index          0x1600000a
namestring port-channell1
..... cut to save space .....

stats
fwd_transition_count 1          bpdus_in      40861    bpdus_out    40861
config_bpdu_in      0          rstp_bpdu_in 40861    tcn_bpdu_in  0
config_bpdu_out     0          rstp_bpdu_out 40861    tcn_bpdu_out  0
bpdufilter_drop_in  0
bpduguard_drop_in   0
err_dropped_in      0
sw_flood_in         0
..... cut to save space .....
```

## Enquête BPDU avec Ethalyzer

Cette section décrit comment employer Ethalyzer afin de capturer des BPDU :

```
Ethalyzer local interface inbound-hi display-filter "vlan.id == 1 && stp"
```

Example:

```
Nexus-5000# ethalyzer local interface inbound-hi display-filter "vlan.id
== 1 && stp"
```

Capturing on eth4

```
2013-05-11 13:55:39.280951 00:05:73:f5:d6:27 -> 01:00:0c:cc:cc:cd STP RST.
Root = 33768/00:05:73:ce:a9:7c Cost = 1 Port = 0x900a
2013-05-11 13:55:40.372434 00:05:73:ce:a9:46 -> 01:00:0c:cc:cc:cd STP RST.
Root = 33768/00:05:73:ce:a9:7c Cost = 0 Port = 0x900a
2013-05-11 13:55:41.359803 00:05:73:f5:d6:27 -> 01:00:0c:cc:cc:cd STP RST.
Root = 33768/00:05:73:ce:a9:7c Cost = 1 Port = 0x900a
2013-05-11 13:55:42.372405 00:05:73:ce:a9:46 -> 01:00:0c:cc:cc:cd STP RST.
Root = 33768/00:05:73:ce:a9:7c Cost = 0 Port = 0x900a
```

Afin de voir les paquets détaillés, utilisez la commande de détail :

```
Nexus-5000# ethalyzer local interface inbound-hi detail display-filter
"vlan.id == 1 && stp"
```

Capturing on eth4

Frame 7 (68 bytes on wire, 68 bytes captured)

```
Arrival Time: May 11, 2013 13:57:02.382227000
[Time delta from previous captured frame: 0.000084000 seconds]
[Time delta from previous displayed frame: 1368280622.382227000 seconds]
[Time since reference or first frame: 1368280622.382227000 seconds]
Frame Number: 7
Frame Length: 68 bytes
Capture Length: 68 bytes
[Frame is marked: False]
[Protocols in frame: eth:vlan:llc:stp]
```

```
Ethernet II, Src: 00:05:73:ce:a9:46 (00:05:73:ce:a9:46), Dst: 01:00:0c:cc:cc:cd
(01:00:0c:cc:cc:cd)
```

```
Destination: 01:00:0c:cc:cc:cd (01:00:0c:cc:cc:cd)
Address: 01:00:0c:cc:cc:cd (01:00:0c:cc:cc:cd)
```

```

    .... .01 ..... = IG bit: Group address (multicast/broadcast)
    .... .0. .... = LG bit: Globally unique address
(factory default)
Source: 00:05:73:ce:a9:46 (00:05:73:ce:a9:46)
Address: 00:05:73:ce:a9:46 (00:05:73:ce:a9:46)
    .... .0 ..... = IG bit: Individual address (unicast)
    .... .0. .... = LG bit: Globally unique address
(factory default)
Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN
111. .... = Priority: 7
...0 .... = CFI: 0
.... 0000 0000 0001 = ID: 1
Length: 50
Logical-Link Control
DSAP: SNAP (0xaa)
IG Bit: Individual
SSAP: SNAP (0xaa)
CR Bit: Command
Control field: U, func=UI (0x03)
    000. 00.. = Command: Unnumbered Information (0x00)
    .... .11 = Frame type: Unnumbered frame (0x03)
Organization Code: Cisco (0x00000c)
PID: PVSTP+ (0x010b)
Spanning Tree Protocol
Protocol Identifier: Spanning Tree Protocol (0x0000)
Protocol Version Identifier: Rapid Spanning Tree (2)
BPDU Type: Rapid/Multiple Spanning Tree (0x02)
BPDU flags: 0x3c (Forwarding, Learning, Port Role: Designated)
    0... .... = Topology Change Acknowledgment: No
    .0.. .... = Agreement: No
    ..1. .... = Forwarding: Yes
    ...1 .... = Learning: Yes
    .... 11.. = Port Role: Designated (3)
    .... ..0. = Proposal: No
    .... ...0 = Topology Change: No
Root Identifier: 33768 / 00:05:73:ce:a9:7c
Root Path Cost: 0
Bridge Identifier: 33768 / 00:05:73:ce:a9:7c
Port identifier: 0x900a
Message Age: 0
Max Age: 20
Hello Time: 2
Forward Delay: 15
Version 1 Length: 0

```

Afin d'écrire ces informations à un fichier PCAP, utilisez cette commande :

```

Nexus-5000# ethanalyzer local interface inbound-hi display-filter
"vlan.id == 1 && stp" write bootflash:bpdu.pcap
Capturing on eth4
3 << Lists how many packets were captured.

```

Sur des captures BPDU, l'adresse MAC source est l'adresse MAC d'interface du périphérique final.

Sur la capture d'Ethanalyzer, le port apparaît dans un format hexadécimal. Afin d'identifier le numéro de port, vous devez convertir d'abord le nombre en hexadécimal :

0x900a (du suivi précédent) = 36874

C'est la commande qui décode ce nombre à un port :

```
Nexus-5000# show spanning-tree internal info all |
grep -b 50 "port_id          36874" | grep "Port Info"
----- STP Port Info (vdc 1, tree 1, port Po11) -----
----- STP Port Info (vdc 1, tree 300, port Po11) -----
----- STP Port Info (vdc 1, tree 800, port Po11) -----
----- STP Port Info (vdc 1, tree 801, port Po11) -----
----- STP Port Info (vdc 1, tree 802, port Po11) -----
----- STP Port Info (vdc 1, tree 803, port Po11) -----
----- STP Port Info (vdc 1, tree 999, port Po11) -----
```

Dans ce cas, c'est le Port canalisé 11.

## Convergence STP

Si vous devez étudier la convergence STP, utilisez la commande **interne d'interactions de show spanning-tree**. Cette commande fournit la vue dans quels événements ont déclenché les modifications STP. Il est important de recueillir ces informations dès que la question se produira, parce que les logs sont grands, et ils s'enveloppent au fil du temps.

```
Nexus-5000#show spanning-tree internal interactions
- Event:(null), length:123, at 81332 usecs after Sat May 11 12:01:47 2013
Success: pixm_send_set_mult_cbl_vlans_for_multiple_ports, num ports 1
VDC 1, state FWD, rr_token 0x21b9c3 msg_size 584
- Event:(null), length:140, at 81209 usecs after Sat May 11 12:01:47 2013
vb_vlan_shim_set_vlans_multi_port_state(2733): Req (type=12) to PIXM
vdc 1, inst 0, num ports 1, state FWD
[Po17 v 800-803,999-1000]
- Event:(null), length:123, at 779644 usecs after Sat May 11 12:01:46 2013
Success: pixm_send_set_mult_cbl_vlans_for_multiple_ports, num ports 1
VDC 1, state FWD, rr_token 0x21b99a msg_size 544<
- Event:(null), length:127, at 779511 usecs after Sat May 11 12:01:46 2013
vb_vlan_shim_set_vlans_multi_port_state(2733): Req (type=12) to PIXM
vdc 1, inst 0, num ports 1, state FWD
[Po17 v 300]
- Event:(null), length:123, at 159142 usecs after Sat May 11 12:01:32 2013
Success: pixm_send_set_mult_cbl_vlans_for_multiple_ports, num ports 1
VDC 1, state LRN, rr_token 0x21b832 msg_size 584
- Event:(null), length:140, at 159023 usecs after Sat May 11 12:01:32 2013
vb_vlan_shim_set_vlans_multi_port_state(2733): Req (type=12) to PIXM
vdc 1, inst 0, num ports 1, state LRN
[Po17 v 800-803,999-1000]
- Event:(null), length:123, at 858895 usecs after Sat May 11 12:01:31 2013
Success: pixm_send_set_mult_cbl_vlans_for_multiple_ports, num ports 1
VDC 1, state LRN, rr_token 0x21b80b msg_size 544
- Event:(null), length:127, at 858772 usecs after Sat May 11 12:01:31 2013
vb_vlan_shim_set_vlans_multi_port_state(2733): Req (type=12) to PIXM
vdc 1, inst 0, num ports 1, state LRN
[Po17 v 300]
..... cut to save space .....
```

## Mappage externe VLAN

Les Commutateurs de gamme de Nexus 5000 emploient des VLAN internes afin de tracer aux nombres externes VLAN pour la transmission. Parfois l'ID DE VLAN est l'identification VLAN interne afin d'obtenir le mappage à un VLAN externe, entrent :

```
Nexus-5000# show platform afm info global
Gatos Hardware version 0
Hardware instance mapping
```

```

-----
Hardware instance: 0 asic id: 0 slot num: 0
----- cut to save space -----
Hardware instance: 12 asic id: 1 slot num: 3
AFM Internal Status
-----
[unknown label ]: 324
[no free statistics counter ]: 2
[number of verify ]: 70
[number of commit ]: 70
[number of request ]: 785
[tcam stats full ]: 2

Vlan mapping table
-----
Ext-vlan: 1 - Int-vlan: 65

```

## Debugs STP

Une autre manière de déboguer des questions STP est de l'utiliser met au point. Cependant, l'utilisation de STP met au point pourrait faire clouer l'utilisation du CPU, qui entraîne des soucis dans quelques environnements. Afin de réduire rigoureusement l'utilisation du CPU tout en s'exécutant met au point, utilisez un debug-filtre, et connectez-vous l'activité à un fichier journal.

1. Créez le fichier journal, qui est enregistré sous le log de répertoire.

```

Nexus-5000#debug logfile spanning-tree.txt
Nexus-5548P-L3# dir log:
31 Nov 06 12:46:35 2012 dmesg
----- cut to save space-----
7626 Nov 08 22:41:58 2012 messages
0 Nov 08 23:05:40 2012 spanning-tree.txt
4194304 Nov 08 22:39:05 2012 startupdebug

```

2. Exécutez le débogage.

```

Nexus-5000# debug spanning-tree bpdu_rx interface e1/30
<<<setup your spanning-tree for bpdus
Nexus-5000# copy log:spanning-tree.txt bootflash:

```

Exemple à partir du fichier journal :

```

Nexus-5000# debug spanning-tree bpdu_rx interface e1/30
<<<setup your spanning-tree for bpdus
Nexus-5000# copy log:spanning-tree.txt bootflash:

```

## Le Nexus 5000 n'a pas traité des BPDU

Afin de déboguer ce problème, vérifiez l'historique d'événement pour déterminer si la gamme de Nexus 5000 commute la racine assumée. Le Nexus 5000 assume la racine s'il ou ne traitait pas des BPDU ou il ne les recevait pas. Afin d'étudier qui est la cause, vous devriez déterminer s'il y a d'autres Commutateurs reliés au pont désigné qui a eu ce problème aussi bien. Si aucune autre passerelle n'avait le problème, il est le plus susceptible que le Nexus 5000 n'ait pas traité les BPDU. Si d'autres passerelles avaient le problème, il est le plus susceptible que la passerelle n'ait pas envoyé les BPDU.

Remarque: Choses à maintenir dans le pour le débogage STP d'esprit et le Port canalisé virtuel (vpc). Seulement le vpc primaire envoie des BPDU. Quand le vpc secondaire est la racine STP, le primaire envoie toujours les BPDU. Si la racine est connectée par l'intermédiaire d'un vpc, seulement les compteurs primaires de Rx BPDU d'incrément,

même lorsque le secondaire les reçoit.