

# Spanning Tree Protocol (STP)

Ecrivez un compte rendu des résultats de votre expérimentation, et répondez aux questions.

[http://wiki.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-lab\\_spanning-tree.pdf](http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-lab_spanning-tree.pdf)

**Bridge identifiant** (bridge-id) : est composé en concaténant 2 éléments:

**Priority** (priorité, locale dans le bridge, imposée par l'administrateur du réseau). Default value : 80-00

**Le bridge MAC** : c'est à dire l'adresse MAC de la "première" porte du bridge (normalement, celle avec le MAC plus bas). Exemple : 80-00-23-ef-...

Ils existent 2 types de paquets, Bridge PDU, ou *bpdu* (PDU =Packet Data Unit) :

**1) configuration bpdu**, qui contient toutes les informations nécessaires pour l'algorithme de spanning tree, c'est à dire :

- root bridge identifiant (la racine actuelle, "root", du spanning tree)
- root path cost (le cout du chemin vers le root bridge)
- bridge identifiant (identifiant de l'émetteur, ou "sender" de la bpdu)
- port identifiant (port à travers laquelle la bpdu est envoyée)

Exemple:

root-bridge-id	80-00-23-ef-...
root-path-cost	100
bridge-id	80-00-2d-12-...
port-id	6

**2) topology change bpdu** contient seulement les données nécessaires pour identifier le paquet comme un paquet de "topology change"

## Les 4 opérations du protocole Spanning Tree

### 1) Election du "root bridge"

Un seul bridge est sélectionné pour être la racine de l'arbre (spanning tree)

## 2) Identification de la "root port" sur chaque bridge

Chaque bridge qui n'est pas le "root bridge" sélectionne une des portes comme celle qui est la plus proche du "root bridge"

## 3) Détermination des "designated bridges"

- Pour chaque LAN, un bridge est choisi pour être celui qui connecte le LAN au spanning tree
- La porte du "designated bridge" qui connecte le LAN au spanning tree est appelée "designated port"

## 4) Blockage des portes redondantes

Toutes les portes qui ne sont pas "root ports" ni "designated ports" sont mise en état de "block" ("blocking state")

### 1) Election du "root bridge"

- Chaque bridge produit une "configuration bpdu" dans laquelle son propre bridge-id est utilisé comme root-identifier
- Quand un bridge reçoit une "configuration bpdu" avec une valeur de bridge-id plus petite,
  - 1) il arrête de produire des "configuration bpdus"
  - 2) il fait forwarding de cette nouvelle "configuration bpdu" à travers toutes ses portes
- Le root bridge est le seul qui continue à produire des *configuration bpdus* avec son propre bridge-id dans le champ « root-identifier »

### Forwarding des "configuration bpdu"

- Quand une "configuration bpdu" est produite par un bridge, la valeur de son champ root-path-cost est égale à zéro
- Quand la configuration bpdu est forwardé par un bridge qui n'est pas donc le root bridge, ses champs sont modifié comme suit :
  - Le root-path-cost est augmenté en additionnant le cout de la porte du bridge à travers laquelle la configuration bpdu a été reçue
  - Le champ bridge-identifier est mis à la valeur du bridge-id du bridge actuel
  - Le port-identifier va contenir le port number de la porte à travers laquelle la configuration bpdu est envoyée

Voici un exemple (le cout de chacune des 8 portes est égal soit à 10, soit à 100)



## 2) Identification de la “root port”

- Chaque bridge qui n’est pas le root bridge identifie la « meilleure porte » à travers laquelle le root bridge est joignable

- La root port est choisie comme celle qui reçoit les configuration bpdus avec

1. le plus bas root-path-cost (après avoir ajouté le port cost)
2. le plus bas bridge-identifiant
3. le plus bas port-identifiant
4. le plus bas « own port-identifiant »

## 3) Determination des designated bridges

Pour chaque LAN, une *bridge port* est choisie comme *designated port* en se basant sur les *configuration bpdus* qui sont forwardées par cette porte.

Le bridge qui possède cette *designated port* est appelé *designated bridge*

La *designated port* est celle choisie pour envoyer des configuration bpdus avec :

1. le plus bas root-path-cost
2. le plus bas bridge-identifiant
3. le plus bas port-identifiant

#### 4) blocking

Toutes les portes qui ne sont pas root ports ou designated ports sont placées en « *blocking state* »

Toutes les root ports et designated ports sont placées en revanche en « *forwarding state* »

#### Port states

Durant l'exécution/détermination du spanning tree, l'état d'une porte peut donc être l'un de suivants :

	receives frames	forwards frames	updates database	receives bpdus	transmits bpdus
blocking	✗	✗	✗	✓	✗
listening	✗	✗	✗	✓	✓
learning	✗	✗	✓	✓	✓
forwarding	✓	✓	✓	✓	✓
disabled	✗	✗	✗	✗	✗

#### Timers

Il y a aussi plusieurs *timers* qui sont utilisé dans l'exécution du protocole spanning tree, notamment :

- *hello time* [valeur par default = 2s]

Intervalle de temps entre la génération de bpdu par n'importe quel bridge

- *max age* [20s]

Temps maximal durant lequel le bridge va attendre de recevoir une bpdu. Une fois ce timer expiré, un message de « topology change notification » va être envoyé

- *forward delay* [15s]

Temps durant lequel le bridge va rester dans les états de *listening* et de *learning port*.

## Laboratoire 1 : élection du *root bridge*

Nous allons utiliser le laboratoire disponible ici :

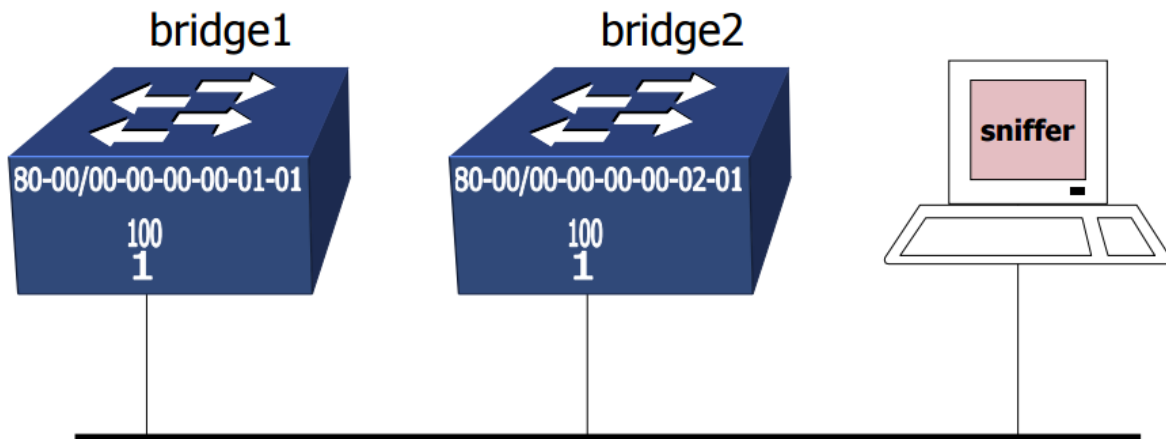
[http://wiki.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-labs\\_spanning-tree.tar.gz](http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-labs_spanning-tree.tar.gz)

Sur la machine réelle utiliser les commandes :

```
cd netkit-lab_stp-root-election
```

```
lstart
```

Topologie du réseau considéré :



Ce laboratoire est configuré pour:

- Faire démarrer les 2 bridges
- Faire démarrer une machine virtuelle avec un packet sniffer qui écoute tout le trafic généré durant l'exécution du protocole spanning tree.
  - o Après avoir capturé 20 paquets, la machine virtuelle "sniffer" est arrêté automatiquement ...
  - o ... et un fichier nommé "sniffer.cap" est créé dans le dossier du lab, pour être ensuite ouvert avec des logiciels tels que wireshark, tshark.

**Q1** : Vérifiez qui a été élu « root bridge », à l'aide de la commande (à exécuter, par exemple, dans « bridge2 »)

```
brctl showstp br0
```

Regardez en particulier la valeur des champs "bridge id" et "designated root"

**Q2** : Ensuite, dans la machine réelle analysez l'échange des 20 paquets capturés auparavant dans le fichier sniffer.cap, en décrivant ce que peut être déduit dans ces paquets concernant notamment l'élection du root bridge. Cela peut être réalisé par exemple à l'aide de la commande

**tshark -V -r sniffer.cap**

(l'option « V » permet d'avoir les détails sur les paquets capturés, et « r » de lire ce qui a été capturé dans un fichier .cap)

```
⊞ IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
  Length: 38
⊞ Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x00
    0... .... = Topology Change Acknowledgment: No
    .... ...0 = Topology Change: No
  Root Identifier: 32768 / 00:00:00:00:02:01
  Root Path Cost: 0
  Bridge Identifier: 32768 / 00:00:00:00:02:01
  Port identifier: 0x8001
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
```

multicast address  
(all the bridges  
on the lan)

this is a  
configuration  
bpdu

at the beginning the  
bridge claims to be  
the root bridge

## Laboratoire 2 : changement dans la topologie du réseau

Nous allons utiliser le laboratoire disponible ici :

[http://wiki.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-labs\\_spanning-tree.tar.gz](http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-labs_spanning-tree.tar.gz)

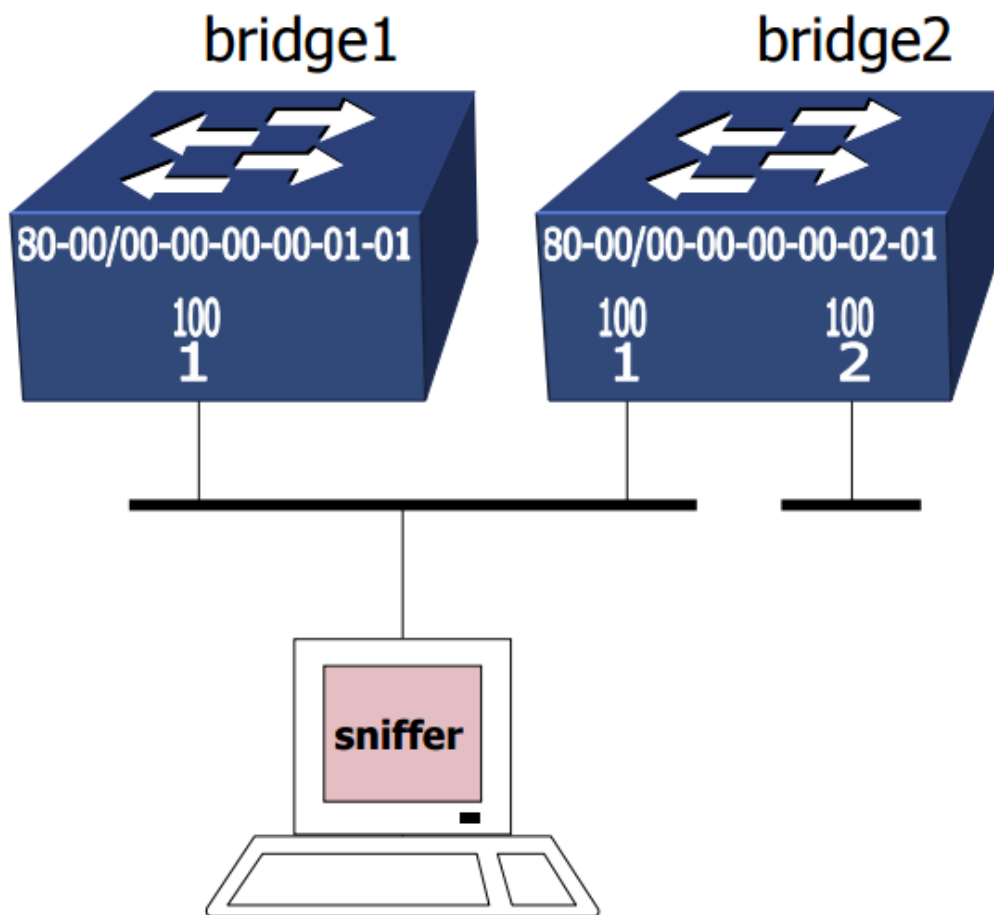
Sur la machine réelle utiliser les commandes :

**cd netkit-lab\_stp-topology-change**

**Istart**

Ce laboratoire est configuré pour:

- Faire démarrer les 2 bridges
- Faire démarrer une machine virtuelle avec un packet sniffer qui écoute tout le trafic généré durant l'exécution du protocole spanning tree.
  - o Mais cette fois, la machine virtuelle "sniffer" n'est pas arrêtée automatiquement
  - o Un fichier nommé "sniffer.cap" est créé dans le dossier du lab, pour être ensuite ouvert avec des logiciels tels que wireshark, tshark



- Quand un bridge détecte un changement dans la topologie, il envoie des « topology change notification » bpdus à travers sa root port
- Il continue à envoyer des notifications jusqu'à ce que le designated bridge sur le LAN attaché à la root port les acquitte (en envoyant des acknowledgments)
- Le designated bridge, à son tour, va envoyer le "topology change notification" bpdus à travers sa root port, jusqu'à ce qu'elle soit reçue par le root bridge

- Le root bridge commence à mettre à 1 (true) le "topology change flag" dans ses « configuration bpdus »
- Tous les bridge, en apprenant par l'écoute de ces bpdus qu'un changement de topologie est en train d'avoir lieu, vont utiliser le "forward delay time" (normalement = 15 seconds) en lieu du « filtering database ageing time » (normalement = 5 minutes) pour faire devenir vieilles (et donc faire sortir du tableau) les lignes correspondantes des « source address tables ».

Pour forcer un changement de topologie, et donc la génération de « topology change notifications » bpdus, il suffit d'utiliser la commande suivante :

Dans bridge2 :

**brctl addif br0 eth1**

La notification est capturée correctement par le sniffer après avoir observé les lignes suivantes :

```

sniffer
.....
12:22:24.362414 802.1d tcn
12:22:24.685188 802.1d config TOP_CHANGE TOP_CHANGE_ACK
8000.00:00:00:00:01:01.8001 root 8000.00:00:00:00:01:01 pathcost 0 age 0
max 20 hello 2 fdelay 15
.....

```

```

IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
  Length: 7
Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Topology Change Notification (0x80)

```

bridge2 generates a topology change notification

this is a topology change notification (tcn)



```

IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:01:01 (00:00:00:00:01:01)
  Length: 38
Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x81 (Topology Change Acknowledgment, Topology Change)
    1... .... = Topology Change Acknowledgment: Yes
    .... ...1 = Topology Change: Yes
  Root Identifier: 32768 / 00:00:00:00:01:01
  Root Path Cost: 0
  Bridge Identifier: 32768 / 00:00:00:00:01:01
  Port identifier: 0x8001
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15

```

the root bridge acknowledges the notification

the topology change is further propagated

this configuration bpdv carries a topology change notification acknowledgment

### Laboratoire 3 : changement de la topologie à distance

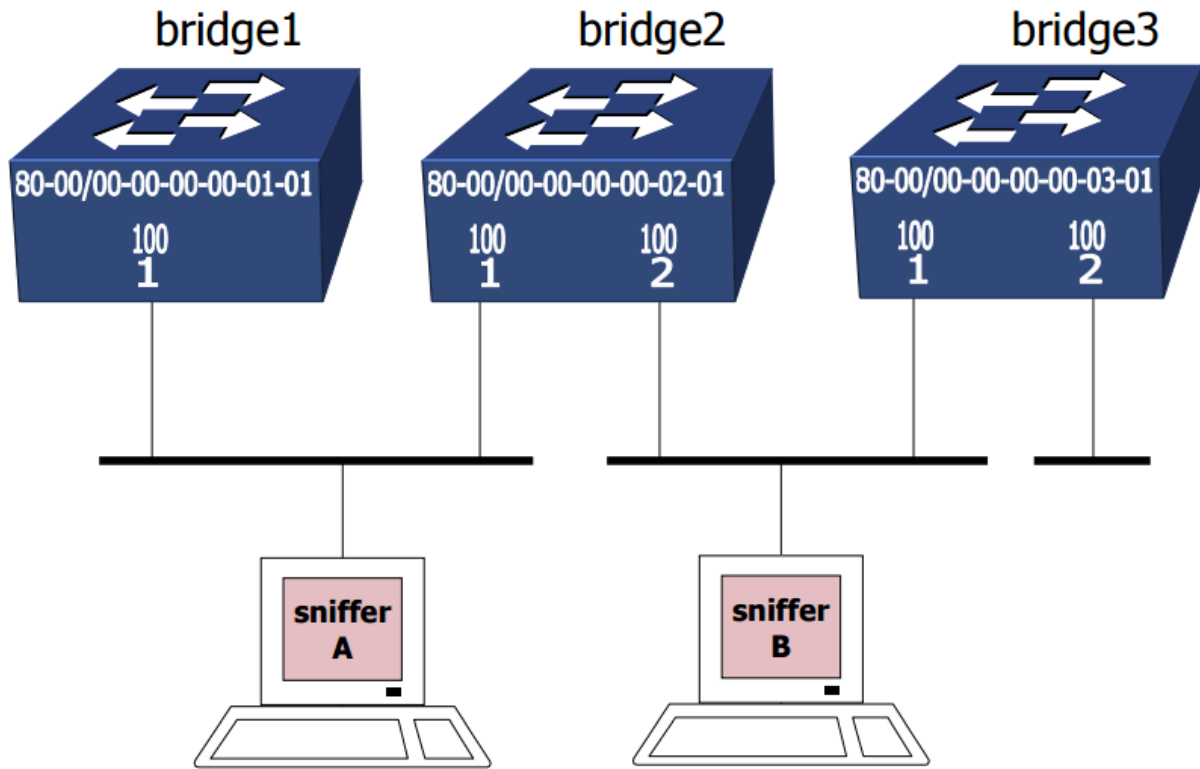
Nous allons utiliser le laboratoire disponible ici :

[http://wiki.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-labs\\_spanning-tree.tar.gz](http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-labs_spanning-tree.tar.gz)

Sur la machine réelle utiliser les commandes :

**Cd netkit-lab\_stp lab\_stp lab\_stp-remote-change**

**Istart**



Ce laboratoire est configuré pour faire démarrer tout le réseau et créer sur le host les fichiers de capture suivants :

- sniffer\_A.cap
- sniffer\_B.cap

La notification du changement de topologie peut être enclenché en utilisant la commande suivante :

Dans bridge3:

**brctl addif br0 eth1**

La notification est capturée correctement par le sniffer après avoir observé les lignes suivantes :

sniffer_A	sniffer_B
<pre> ..... 13:01:22.366772 802.1d tcn 13:01:22.367400 802.1d config TOP_CHANGE TOP_CHANGE_ACK 8000.00:00:00:00:01:01.8001 root 8000.00:00:00:00:01:01 pathcost 0 age 0 max 20 hello 2 fdelay 15 ..... </pre>	<pre> ..... 13:01:22.365668 802.1d tcn 13:01:22.366464 802.1d config TOP_CHANGE_ACK 8000.00:00:00:00:02:01.8002 root 8000.00:00:00:00:01:01 pathcost 100 age 1 max 20 hello 2 fdelay 15 ..... </pre>

Fichier de capture : dans B

```
IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:03:01 (00:00:00:00:03:01)
  Length: 7
Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Topology Change Notification (0x80)
```

bridge3  
port 1

step 1: bridge3 generates a topology change notification

```
IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:02:02 (00:00:00:00:02:02)
  Length: 38
Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x80 (Topology Change Acknowledgment)
    1... .. = Topology Change Acknowledgment: Yes
    .... ..0 = Topology Change: No
  Root Identifier: 32768 / 00:00:00:00:01:01
  Root Path Cost: 100
  Bridge Identifier:
  Port Identifier:
  Message Age:
  Max Age: 2
  Hello Time:
  Forward Delay: 15
```

bridge2  
port 2

step 2: bridge2 acknowledges  
(hence, bridge3 stops sending notifications)

this is just an acknowledgment

Dans A :

```
⊞ IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:02:01 (00:00:00:00:02:01)
  Length: 7
⊞ Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Topology Change Notification (0x80)
```

bridge2  
port 1

step 3: bridge2 propagates  
the topology change notification  
through its root port

```
⊞ IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: 00:00:00_00:01:01 (00:00:00:00:01:01)
  Length: 38
⊞ Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  SSAP: Spanning Tree BPDU (0x42)
⊞ Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x81 (Topology Change Acknowledgment, Topology Change)
    1... .. = Topology Change Acknowledgment: Yes
    .... ..1 = Topology Change: Yes
  Root Identifier: 32768 / 00:00:00:00:01:01
  Root Path Cost: 0
  Bridge
  Port i
  Messag
  Max Ag
  Hello
  Forward Delay: 15
```

bridge1  
port 1

the topology  
change flag  
is set

step 4: bridge1 acknowlegdes  
(hence, bridge2 stops sending notifications)  
and sets the topology change flag

## Laboratoire 4 : un scenario plus compliqué

Nous allons utiliser le laboratoire disponible ici :

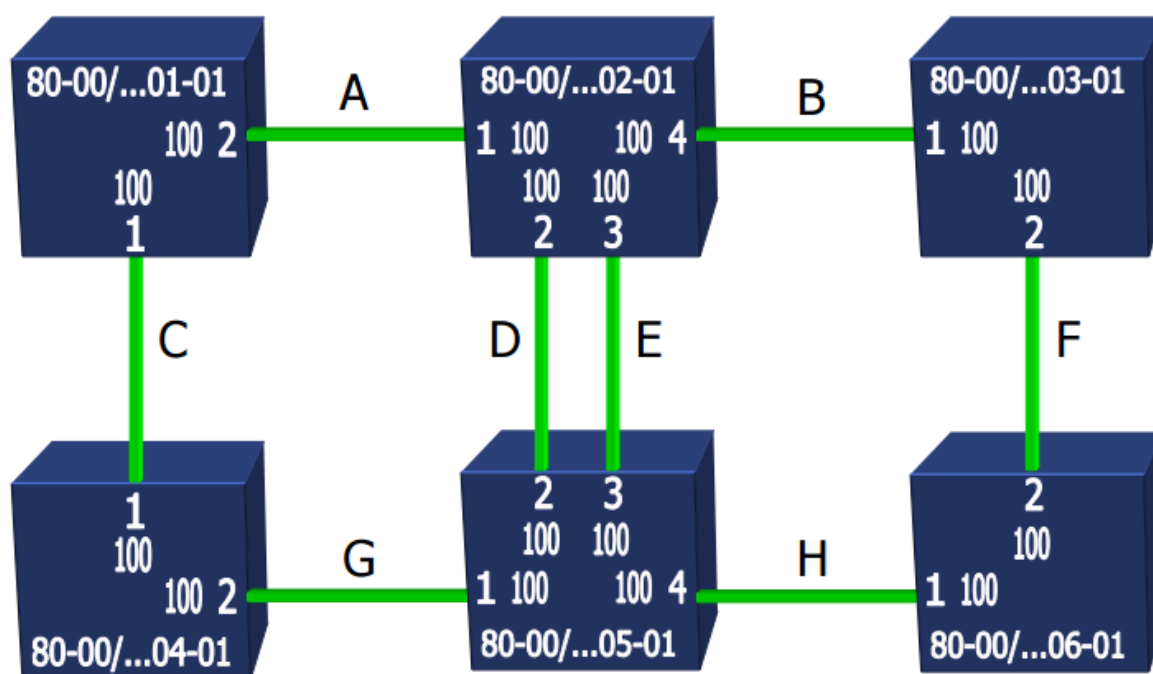
[http://wiki.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-labs\\_spanning-tree.tar.gz](http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-labs_spanning-tree.tar.gz)

Sur la machine réelle utiliser les commandes :

```
cd netkit-lab_stp lab_stp lab_stp-complex
```

```
Istart
```

Le laboratoire est configuré pour faire démarrer les 6 bridges qui utilisent le Spanning Tree Protocol pour élire le *root bridge*,



Q3 : qui est élu *root bridge* ?

(Utiliser la commande **brctl showstp br0** sur chacun des 6 bridges et observer l'état du bridge par rapport au STP. En particulier, le *designated root*)

Q4 : quelles sont (A) les *root ports*, (B) les *designated ports/bridges*, (C) le *path cost* et (D) les portes en état de « blocking » pour chacun des 6 bridges (les représenter graphiquement sur le schéma de la topologie ci-dessus) ? A noter que, dans cette opération, bridge5 et bridge6, pour identifier les *root ports*, doivent « rompre la parité » (break the tie) en utilisant le plus bas bridge-id et aussi le plus bas port-id.

(Utiliser la commande **brctl showstp br0** sur chacun des 6 bridges et observer l'état du bridge par rapport au STP.)

Q5: essayez de faire changer le root bridge dans le réseau en changeant les *bridge priorities*

(commandes utilisables : il suffit d'écrire « brctl » dans un bridge. En particulier il y a la commande setbridgeprio <bridge> <prio>)

Q6 : essayez d'utiliser toutes les deux ports d'un lien spécifique en changeant les *ports priorities*. Pourquoi est-il compliqué ?

(commandes utilisables : il suffit d'écrire « brctl » dans un bridge. En particulier il y a la commande setportprio <bridge> <port> <prio>)