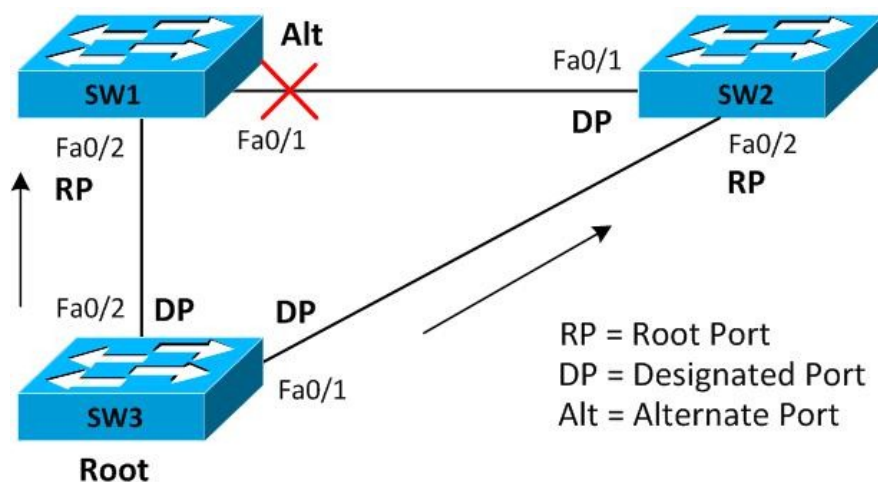


**- TP 4 -**  
**Analyse de trames sur Packet Tracer**  
**Spanning Tree Protocol**  
**Etiquetage VLAN**

*par Édouard Lumet*



## Sommaire

Introduction.....	3
1. Analyse du Spanning Tree Protocol.....	4
1.1. Observation de STP dans une conception redondante.....	4
1.2. Étude de l'algorithme STA : convergence STP.....	9
1.3. Configuration du protocole STP.....	15
2. Analyse de l'étiquetage VLAN.....	19
2.1. Configuration de l'infrastructure réseau.....	19
2.2. Observation des trames étiquetées VLAN.....	23
3. Spanning Tree et multi-VLAN.....	25
3.1. Configuration de l'infrastructure réseau.....	26
3.2. Analyse du processus de Spanning Tree.....	27
3.3. Optimisation de la configuration du Spanning Tree.....	29
Conclusion.....	32

## Introduction

Dans ce dernier TP du module d'ASR3 (réseaux locaux et équipements actifs), nous allons utiliser le logiciel Packet Tracer afin d'observer et de comprendre le protocole Spanning Tree ainsi que l'algorithme Spanning Tree. Nous pourrons ainsi analyser les trames BPDU qui comprennent un en-tête supplémentaire pour l'ajout du numéro de VLAN entre autres.

Au fur et à mesure, l'objectif sera dans un premier temps d'observer le protocole et ses faiblesses dans le cas d'un réseau simple et d'un réseau contenant plusieurs VLAN. Ensuite, nous apprendrons à l'optimiser en le guidant dans son élection du pont racine et la sélection des différents états des ports.

# 1. Analyse du Spanning Tree Protocol

## 1.1. Observation de STP dans une conception redondante

« Spanning Tree Protocol » est un protocole ayant pour objectif de neutraliser les boucles dans un réseau dues à la topologie de celui-ci. Pour obtenir une certaine redondance au sein d'un réseau, il peut être intéressant de relier plusieurs switches formant ainsi une boucle au cas où l'un des liens ou switches serait inopérant. Cependant, cette boucle est alors source de « tempête de broadcast » ou « broadcast storm ». Lors de l'envoi d'un broadcast, la trame reçue sur un port repart alors vers le switch voisin et ainsi de suite. Le protocole Spanning Tree désactive alors un des ports sur un ou plusieurs switches pour éviter les boucles, qui pourra être réactivé en cas de changement de topologie (switch tombant en panne par exemple).

- Afin de comprendre la définition ci-dessus, nous allons étudier un réseau avec une topologie redondante sous Packet Tracer.
- Au démarrage (à l'ouverture du fichier `.pkt`), toutes les liaisons sont orange, elles sont inactives.
- A chaque connexion d'un équipement sur la baie de brassage de la salle ou sur le switch utilisé lors des TP 1, 2 et 3, nous pouvons également observer ce phénomène. Cette attente est due au protocole STP car le branchement d'un équipement par exemple sur un switch est synonyme de modification de la topologie du réseau, et donc la formation possible d'une boucle.
- Il faut attendre environ 30 à 40 secondes avant que les ports soient opérationnels. Sous Packet Tracer, il faut compter 30 secondes.
- Une fois la convergence atteinte, lorsque les ports sont verts cela correspond aux ports désignés ou racines, ils sont alors actifs, et lorsque les ports sont orange ils sont dits alternatifs, ils sont donc non actifs.

Ensuite, nous nous intéressons à l'élection du pont racine afin de déterminer l'arbre recouvrant dans un second temps.

- Un pont racine est un switch élu en tant que tel par l'algorithme STA (Spanning Tree Algorithm). Il servira ensuite de point de référence pour le calcul de STA, notamment lors des calculs de coût. Le plus judicieux est d'avoir un switch important, puissant (coeur de réseau par exemple) comme pont racine. C'est le switch ayant le BID (Bridge Identifier) le plus faible qui sera élu pont racine par l'algorithme STA.
- Le BID est composé principalement d'une valeur de priorité et de l'adresse MAC du switch. La valeur de priorité par défaut est  $8000|_8$  ( $32768|_{10}$ ).

- Pour chaque switch, on indique sa priorité de pont et son adresse MAC utilisées par l'algorithme STA. On utilise pour cela la commande **show spanning-tree**.

Switch	Priorité de pont	Adresse MAC
SC1	32769 (0x8001)	00-01-63-B8-CB-C3
SC2	32769 (0x8001)	00-01-63-52-0C-74
SD1	32769 (0x8001)	00-01-0C-31-32-CA
SD2	32769 (0x8001)	00-06-2A-88-88-EC
SD3	32769 (0x8001)	00-01-43-4B-81-CC
SD4	32769 (0x8001)	00-01-64-D1-00-D5
SA1	32769 (0x8001)	00-30-F2-93-EE-63
SA2	32769 (0x8001)	00-E0-A3-DB-82-D3
SA3	32769 (0x8001)	00-00-0C-31-A9-61
SA4	32769 (0x8001)	00-90-2B-02-4D-59
SA5	32769 (0x8001)	00-E0-F7-09-9C-32
SA6	32769 (0x8001)	00-E0-F7-72-AB-22

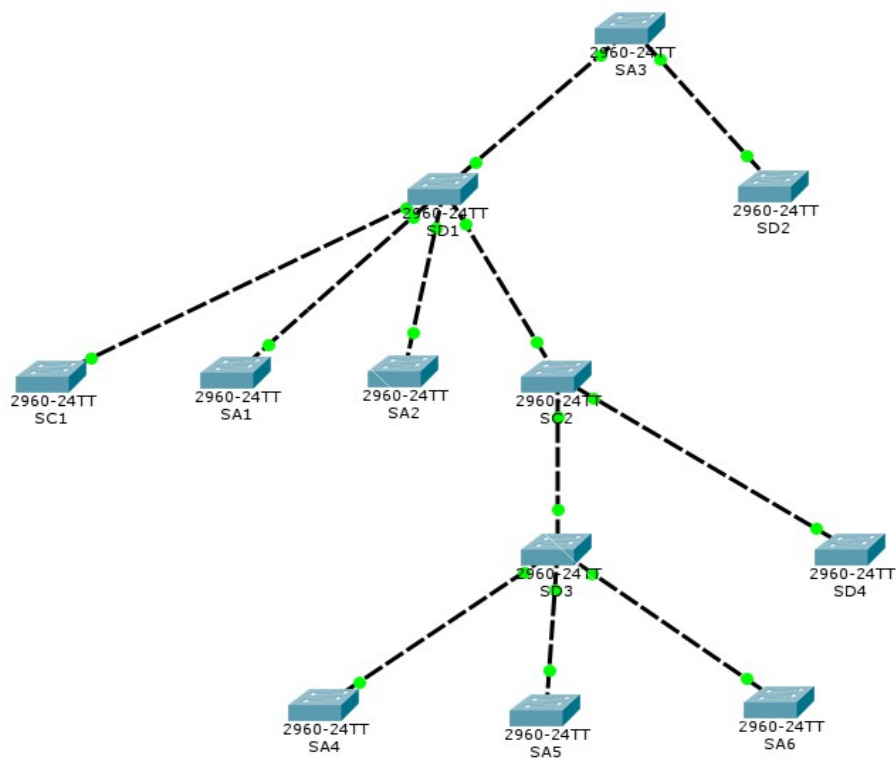
- Ici, tous les switches ont la même valeur de priorité de pont. Le switch élu pont racine sera donc le switch ayant l'adresse MAC la plus faible. Dans notre cas, l'adresse MAC la plus petite est celle de SA3, il sera donc élu pont racine.
- Avec la commande **show spanning-tree**, on peut rapidement identifier le pont racine dans la section « Root ID » où l'on voit l'adresse MAC du pont racine. Lorsque l'on exécute cette commande sur le switch élu pont racine, une phrase est ajoutée dans la section « Root ID » pour indiquer explicitement que ce switch est le pont racine.

```
Switch>en
Switch#show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
            Address     0000.0C31.A961
            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     0000.0C31.A961
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time 20
```

Résultat sur SA3 de **show spanning-tree**

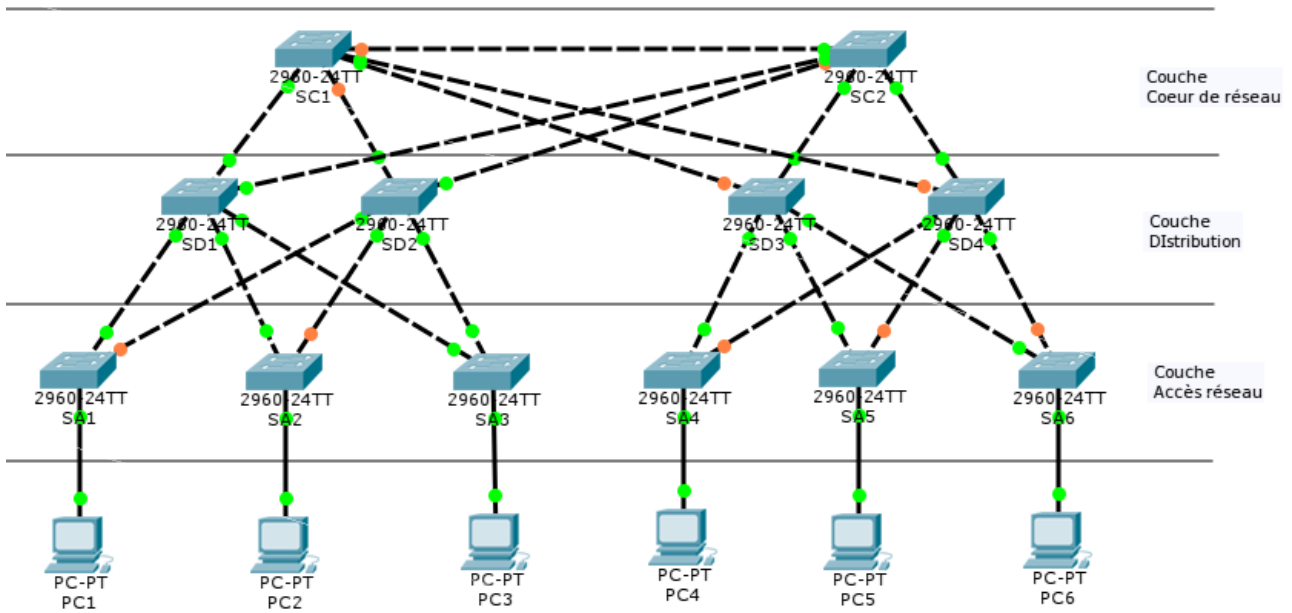
- Le pont racine élu est un switch d'accès réseau. Ce choix n'est pas le plus judicieux comme expliqué dans la définition au début de la partie 1. En effet, la couche d'accès réseau n'offre pas de redondance puisque la plupart des équipements terminaux ne disposent que d'une carte réseau filaire. De plus, les switches d'accès réseau sont les moins puissants, le switch idéal en tant que pont racine serait donc SC1 ou SC2 de cœur de réseau.
- Désormais, nous pouvons représenter l'arbre recouvrant en désactivant les liaisons non fonctionnelles (orange) et en positionnant le pont racine au sommet.



*Arbre recouvrant de la topologie initiale*

Enfin, nous allons observer le comportement lié à la redondance de notre topologie en cas de panne (retrait) d'un switch ou d'une liaison. Cela permet de vérifier le fonctionnement de STP et de son action à chaque changement de topologie.

- Sur la même topologie, on passe en mode simulation avec Packet Tracer et on filtre les messages ICMP uniquement (ping).



Rappel de la topologie initiale (topologie\_1\_1.pkt)

- Ensuite, nous envoyons un ping depuis le PC4 à destination du PC6. Le chemin emprunté par celui-ci est le suivant :

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.008	--	PC4	ICMP	<span style="color: green;">█</span>
	0.009	PC4	SA4	ICMP	<span style="color: green;">█</span>
	0.010	SA4	SD3	ICMP	<span style="color: green;">█</span>
	0.011	SD3	SA6	ICMP	<span style="color: green;">█</span>
	0.012	SA6	PC6	ICMP	<span style="color: green;">█</span>
	0.013	PC6	SA6	ICMP	<span style="color: green;">█</span>
	0.014	SA6	SD3	ICMP	<span style="color: green;">█</span>
	0.015	SD3	SA4	ICMP	<span style="color: green;">█</span>
	0.016	SA4	PC4	ICMP	<span style="color: green;">█</span>

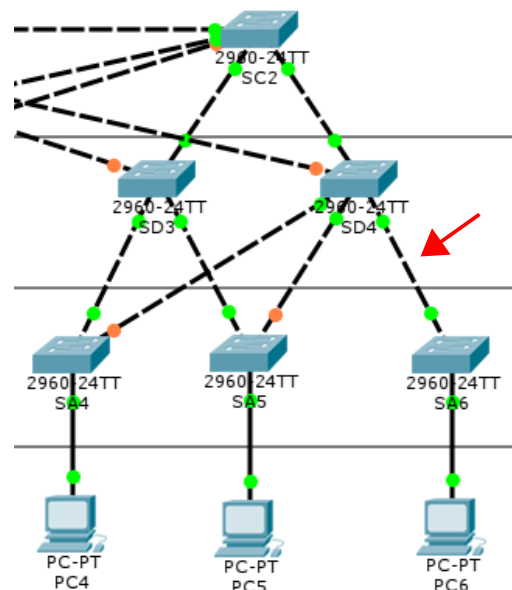
on voit que le chemin est conforme, le message ne passe que par des liaisons actives (vertes).

- Ensuite, on supprime la liaison entre SD3 et SA6 qui était empruntée par le ping précédent. Au bout de quelques secondes, les points de couleurs changent pour donner la topologie suivante :

Ping PC4 → PC6 (request + reply)

- On remarque que la liaison SD4 ↔ SA6 est active. On envoie de nouveau un ping entre PC4 et PC6, le chemin est :

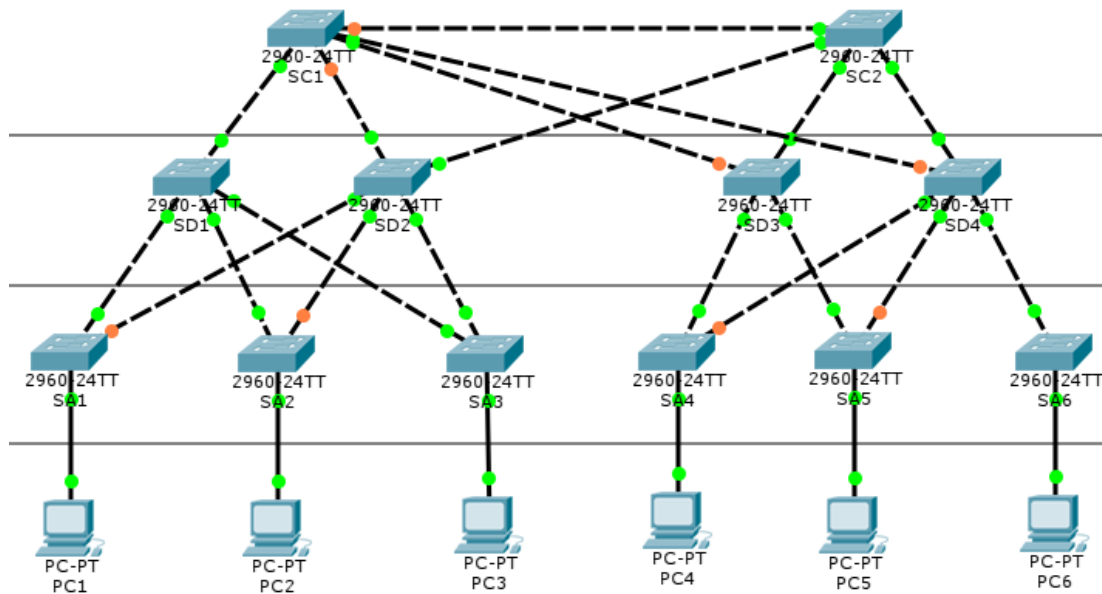
Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC4	ICMP	<span style="color: magenta;">█</span>
	0.001	PC4	SA4	ICMP	<span style="color: magenta;">█</span>
	0.002	SA4	SD3	ICMP	<span style="color: magenta;">█</span>
	0.003	SD3	SC2	ICMP	<span style="color: magenta;">█</span>
	0.004	SC2	SD4	ICMP	<span style="color: magenta;">█</span>
	0.005	SD4	SA6	ICMP	<span style="color: magenta;">█</span>
	0.006	SA6	PC6	ICMP	<span style="color: magenta;">█</span>
	0.007	PC6	SA6	ICMP	<span style="color: magenta;">█</span>



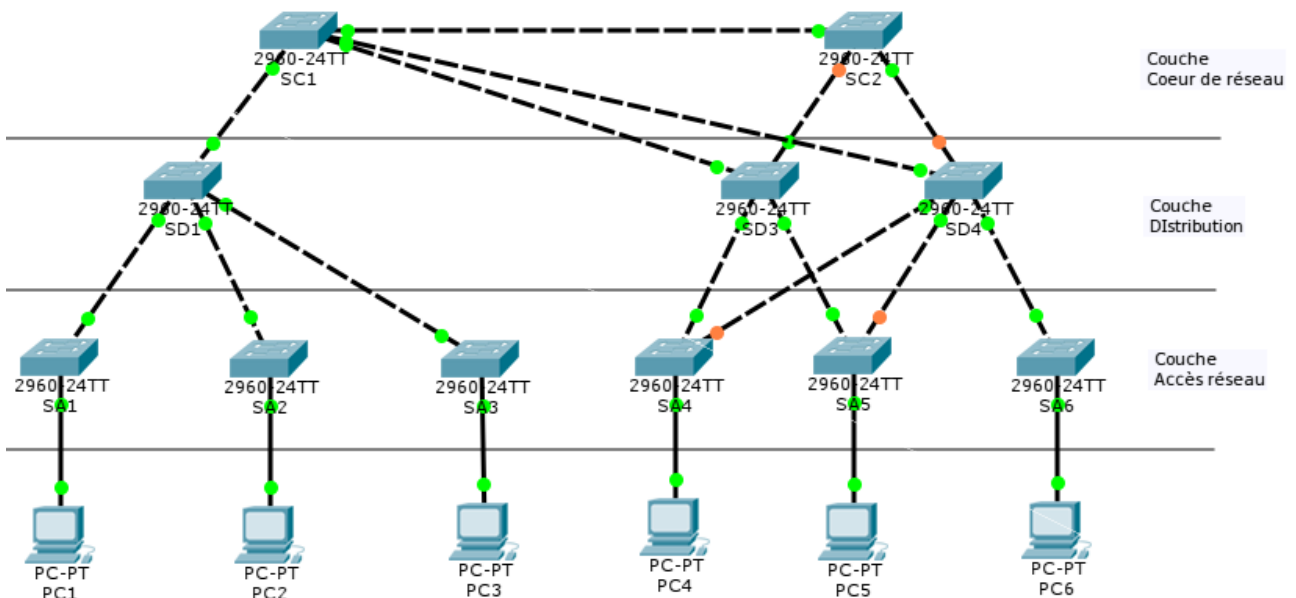
Ping PC4 → PC6 (request seulement)

Le nouveau chemin emprunté est conforme aux couleurs des points. Le message passe donc par SC2 puis par SD4 puisqu'il ne peut plus passer directement de SD3 à SA6. C'est donc pour cela que le lien SD4 > SA6 a été activé.

- On supprime la liaison SD1 ↔ SC2. Cela n'a aucun impact sur le chemin emprunté par le ping PC4 → PC6 puisque cela affecte seulement le lien SD2 ↔ SC2, de l'autre côté du réseau donc.

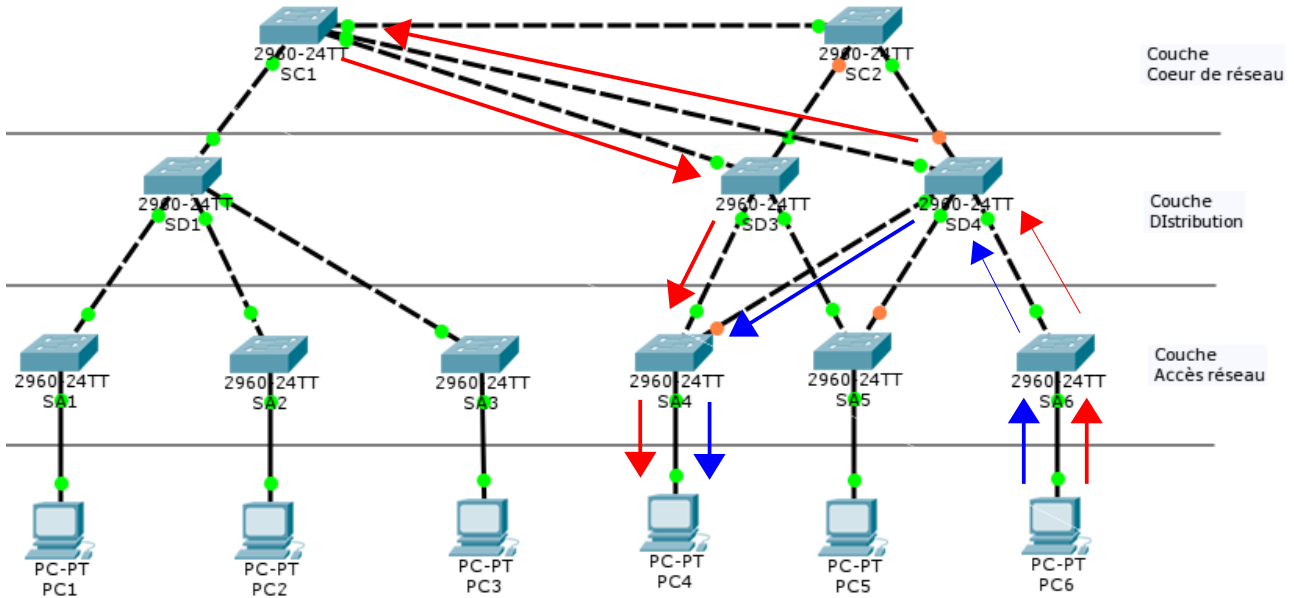


- On supprime maintenant le switch SD2 de la couche distribution. Toutes les liaisons entre les deux arborescences opposées sont rétablies étant donné leur nombre désormais réduit. En revanche, le switch SC2 de cœur de réseau a été exclu du réseau. Ce « choix » n'est pas judicieux.





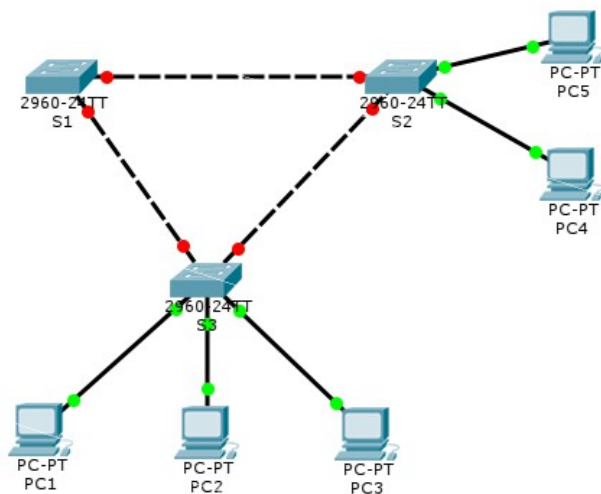
- Tous les ping fonctionnent entre les PC. Cependant, certaines communications doivent emprunter des chemins qui ne sont pas idéaux comme PC6 vers PC4 par exemple. Le chemin emprunté par ce ping est, en rouge :



Pour rappel, le chemin emprunté au début de la partie 1 pour ce même message était celui indiqué en bleu. Les performances sont donc désormais amoindries. C'est pour cela qu'il faut faire en sorte que le pont racine élu soit un switch de la couche cœur de réseau (SC1 ou SC2 par exemple).

### 1.2. Étude de l'algorithme STA : convergence STP

Dans cette deuxième sous-partie, nous allons étudier plus précisément le fonctionnement du protocole STP, notamment de l'élection du pont racine au sein du réseau. Pour cela, nous disposons sous Packet Tracer d'un réseau local simple, composé de trois switches interconnectés et de quelques PC.



- Les trames BPDU (Bridge Protocol Data Unit) sont utilisées dans le protocole STP pour faire converger l'algorithme STA lors de l'élection du pont racine et de la sélection des états des ports. Les switches s'échangent ces trames qui contiennent le BID du switch émetteur, soit : la valeur de priorité + le VLAN ID (optionnel) + l'adresse MAC ; et le coût vers le pont racine.

Topologie simple - analyse trames BPDU

- L'algorithme STA compte 4 étapes, effectuées pour chaque LAN ou VLAN d'un réseau :
  1. élection du pont racine : les switches élisent un pont racine en échangeant des trames BPDU. Le switch ayant le BID le plus faible est élu pont racine ;

*NB : bien que peu démocratique, en modifiant la valeur de priorité sur les switches on peut alors faire en sorte que le switch voulu soit élu...*

  2. détermination des chemins les plus courts vers le pont racine : pour chaque port de chaque switch, un coût est calculé vers le pont racine en fonction des vitesses des liaisons, spécifié par l'IEEE. Plus la vitesse est faible, plus le coût est élevé. Par exemple, si entre un port 1 d'un switch et le pont racine il y a deux liaisons 1 Gbps et entre un port 2 de ce même switch et le pont racine il y n'a qu'une liaison mais à 100 Mbps, le port 1 aura le coût le plus faible ;
  3. sélection des ports racine : pour chaque switch un port est sélectionné comme étant port racine d'après le résultat de l'étape précédente ;
  4. sélection des ports désignés et alternatifs : les ports désignés sont ceux autorisés à acheminer le trafic, et les ports alternatifs sont ceux qui bloquent le trafic afin d'éviter les boucles. Le pont racine configure tous ses ports comme désignés. Les autres switches échangent des trames BPDU sur les liaisons devant être désactivées : le port ayant le BID le plus faible est désigné tandis que l'autre port (à l'autre bout de la liaison) est alternatif.
- A l'ouverture du fichier dans Packet Tracer, les ports reliant les switches entre eux sont administrativement éteints (down).
- Avant toute manipulation, nous devons donc remettre en service les ports interconnectant les switches. Sur chaque switch, on entre les commandes suivantes :

```
>enable
#conf t
#int range fa0/1-2
#no shutdown
```

- Ensuite, comme pour la première sous-partie, on relève les valeurs de priorité et les adresses MAC à l'aide de la commande **show spanning-tree** :

Switch	Priorité de pont	Adresse MAC
S1	32769 (0x8001)	00-90-0C-92-6E-E6
S2	32769 (0x8001)	00-E0-F7-78-91-C4
S3	32769 (0x8001)	00-E0-B0-21-3E-9E

- Les priorités étant les mêmes pour les trois switches, l'adresse MAC la plus petite va déterminer ici le pont racine. C'est donc S1 qui, théoriquement, sera déclaré pont racine.

```
Switch#show span
VLAN0001
Spanning tree enabled protocol ieee
Root ID    Priority    32769
           Address     0090.0C92.6EE6
           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     0090.0C92.6EE6
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 20
```

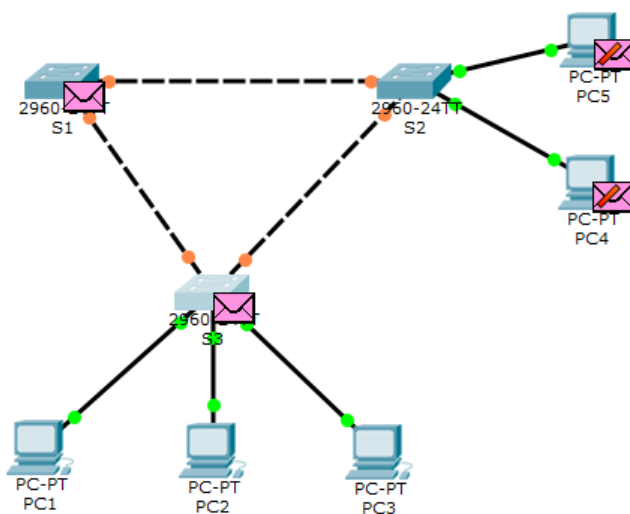
Résultat sur S1 de la commande *show spanning-tree*

On s'intéresse désormais aux échanges de trames BPDU entre nos trois switches, en filtrant le protocole STP uniquement en mode simulation sur Packet Tracer.

- Initialement, avant tout échange de trames BPDU entre les switches, le pont racine estimé pour chaque switch est :

Switch	Adresse MAC du switch	Adresse MAC du pont racine initialement « estimé »
S1	00-90-0C-92-6E-E6	00-90-0C-92-6E-E6
S2	00-E0-F7-78-91-C4	00-E0-F7-78-91-C4
S3	00-E0-B0-21-3E-9E	00-E0-B0-21-3E-9E

- Les premiers envois de message BPDU que l'on peut observer sont 4 trames de S2 à destination de tous les équipements lui étant directement connectés. S2 y indique qui est le pont racine d'après lui et son BID.



Premier échange STP (trames BPDU)

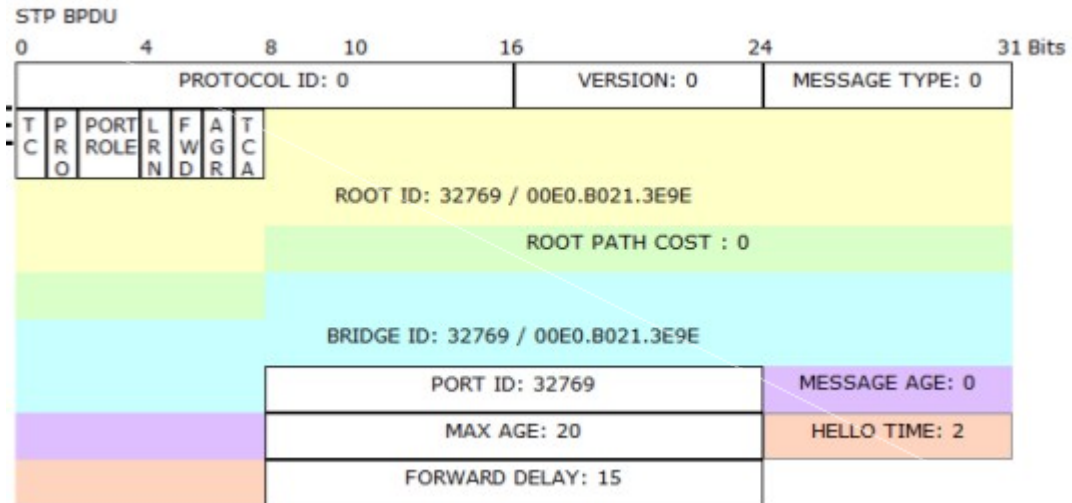
Vis.	Time(sec)	Last Device	At Device	Type	Info
	1.951	--	S2	STP	
	1.952	S2	S3	STP	
	1.952	S2	PC5	STP	
	1.952	S2	PC4	STP	
	1.952	S2	S1	STP	

Premier échange STP (trames BPDU)

Les trames BPDU sont ignorées par les autres équipements que les switches.

- En cliquant sur les enveloppes, on peut voir le contenu des trames entrantes et sortantes.

- Dans l'exemple ci-dessous, on peut observer une trame reçue par S2 en provenance de S3. S3 y indique :
  - son identité, BID = 32769 / 00-E0-B0-21-3E-9E
  - l'identité du pont racine estimé (qu'il considère au moment de l'envoi), Root ID = 32769 / 00-E0-B0-21-3E-9E



- Nous continuons l'observation des trames BPDV en mode pas-à-pas et nous rapportons le contenu des trames dans le tableau suivant jusqu'à l'élection du pont racine :

Trame		Contenu trame émise			S1	S2	S3
N° séq	Émetteur / Destinataire	@MAC du BID	Root ID	Root Path Cost	Pont racine estimé	Pont racine estimé	Pont racine estimé
<i>État initial</i>					S1	S2	S3
1	De S2 vers S1	00-E0-F7-78-91-C4	00-E0-F7-78-91-C4	0	<b>S1</b>	"	"
	De S2 vers S3	00-E0-F7-78-91-C4	00-E0-F7-78-91-C4	0	"	"	S3
2	De S3 vers S2	00-E0-B0-21-3E-9E	00-E0-B0-21-3E-9E	0	"	S3	"
	De S1 vers S2	00-90-0C-92-6E-E6	00-90-0C-92-6E-E6	0	"	<b>S1</b>	"
3	De S2 vers S1	00-E0-B0-21-3E-9E	00-E0-B0-21-3E-9E	0	"	"	"
	De S2 vers S3	00-E0-F7-78-91-C4	00-E0-B0-21-3E-9E	19	"	"	"
4	De S1 vers S3	00-90-0C-92-6E-E6	00-90-0C-92-6E-E6	0	"	"	<b>S1</b>

C'est donc S1 qui a été élu pont racine. Les trames suivantes ne donnent pas d'autres informations concernant les champs Bridge ID et Root ID mais seulement concernant le coût, c'est le calcul et la sélection des ports racines, désignés et alternatifs.

- Le champ « Root Path Cost » indique le coût vers le pont racine (ou pont racine estimé avant la convergence de STP). Par exemple, sur le premier échange, S2 indique à S1 un coût de 0 vers le pont racine. En effet, à ce moment-là le pont racine estimé de S2 est lui-même. De même vers S3. Sur le troisième échange, S2 indique à S3 un coût de 19 vers le pont racine. A ce moment-là le pont racine pour S2 est S1.

Enfin, on repasse en mode temps réel afin d’observer la sélection des ports racines, désignés et alternatifs.

- Un port racine sur un switch est associé au meilleur chemin vers le pont racine du réseau (ou du VLAN).
- Pour élire les ports racines, un calcul du coût vers le pont racine via tous les switches est fait. Ensuite, le chemin avec le coût le plus faible indique quel est le port racine.
- A l’aide de la commande **show spanning-tree** sur chaque switch, on détermine le port racine de chacun des switches :

Équipement	S1	S2	S3
Port racine	∅	Fa0/2	Fa0/2

Interface	Role	Sts	Cost	Prio.	Nbr	Type	Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/2	Desg	FWD	19	128.2		P2p	Fa0/1	Altn	BLK	19	128.1		P2p
Fa0/1	Desg	FWD	19	128.1		P2p	Fa0/3	Desg	FWD	19	128.3		P2p
							Fa0/4	Desg	FWD	19	128.4		P2p
							Fa0/2	Root	FWD	19	128.2		P2p

Résultat sur S1 de la commande **show spanning-tree**

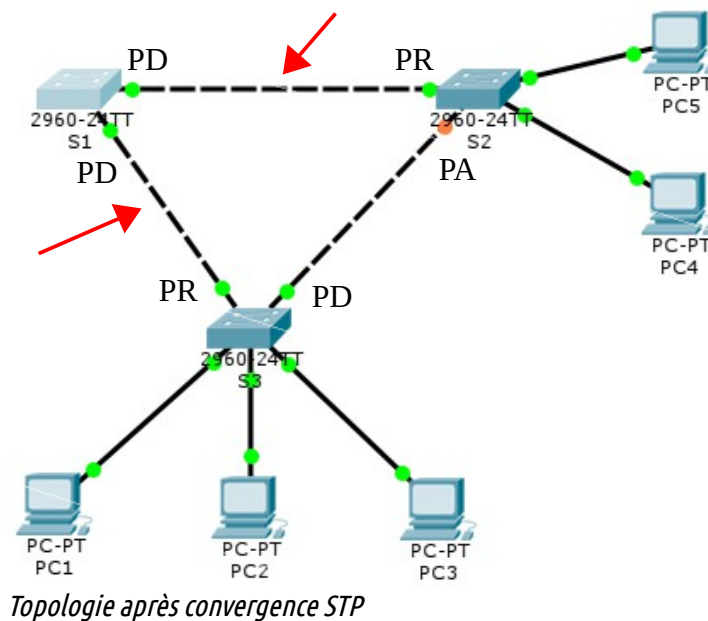
Résultat sur S2 de la commande **show spanning-tree**

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/3	Desg	FWD	19	128.3		P2p
Fa0/2	Root	FWD	19	128.2		P2p
Fa0/1	Desg	FWD	19	128.1		P2p
Fa0/4	Desg	FWD	19	128.4		P2p
Fa0/5	Desg	FWD	19	128.5		P2p

Résultat sur S3 de la commande **show spanning-tree**

- On peut voir deux autres statuts concernant les ports : désigné et alternatif (ou non désigné). Un port désigné laisse passer le trafic alors qu’un port alternatif le bloque pour éviter les boucles. A noter que le pont racine ne possède donc que des ports racines.

- Une seule liaison a été désactivée dans notre réseau local afin d'éviter une boucle, la liaison S3 ↔ S2 comme on peut le voir ci-dessous :



NB : Les flèches rouges indiquent les liens « inter-switches » actifs.

PA = port alternatif ; PD = port désigné ; PR = port racine

- L'arbre recouvrant contient S1 au sommet (car c'est le pont racine) et S2 et S3 en dessous, au même niveau, auxquels sont reliés les équipements terminaux. Concernant cette topologie ce n'est pas vraiment le plus optimal car les communications entre les deux sous-réseaux ne sont pas directes, il y a trois switches à traverser alors que seulement deux switches pourraient être traversés entre PC4 et PC2 par exemple (S2 et S3).
- Pour obtenir un arbre recouvrant plus adapté, il faudrait que S2 ou S3 soient élus pont racine. Ainsi, la liaison inactive sera celle vers S1 qui lui ne relie pas d'équipement terminal. Il faut donc « truquer » l'élection du pont racine !

### 1.3. Configuration du protocole STP

Comme nous l'avons vu dans la sous-partie précédente, l'arbre recouvrant obtenu une fois que la convergence de l'algorithme STA est atteinte n'est pas toujours optimal. Le chemin emprunté par les paquets est parfois rallongé ou le pont racine, élément clé du protocole STP, n'est pas judicieusement choisi. Pour ces différentes raisons, nous allons apprendre à « guider » l'algorithme.

- Dans la [première sous-partie](#), nous avons observé que le pont racine élu était un switch de la couche accès réseau (*topologie\_1\_1.pkt*). Logiquement, le pont racine aurait dû appartenir à la couche cœur de réseau car ce sont des switches ayant une haute disponibilité.
- Dans la [deuxième sous-partie](#), le switch élu pont racine était le switch S1 alors que ce sont plutôt les switches S2 ou S3 qui auraient dû avoir ce rôle (*topologie\_1\_2.pkt*). En effet, l'arbre recouvrant mettait en évidence un rallongement du chemin entre les PC alors que SC1 ne reliait aucun équipement terminal.

Pour rappel, le BID d'un switch contient : une valeur de priorité du pont (0x8000 par défaut), l'extension de l'ID système qui correspond au numéro de VLAN, et l'adresse MAC.

- On reprend le topologie du paragraphe 1.1 afin de favoriser l'élection des deux switches de cœur de réseau (SC1 ou SC2) en tant que pont racine.
- Par défaut, nous avons observé que la valeur de priorité est de 32769 (32768 + 1), soit 0x8001.

```
Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
           Address    0090.0C92.6EE6
```

*#show spanning-tree*

- La valeur 4096 correspond à l'ID système étendu maximum. En effet, codé sur 12 bits, il correspond à l'ID de VLAN qui lui même est compris entre 1 et 4096. La valeur 65535 est la valeur totale de l'ID maximum avec l'ID système étendu minimum de 1, soit  $2^{16} - 1 = 65535$ .
- Pour faire en sorte qu'un switch soit élu pont racine, il faut donc lui affecter une valeur de priorité très petite afin que son BID soit globalement plus faible, élément déterminant de l'élection du pont racine.

Comme nous l'avons vu dans la définition de STP et des étapes de l'algorithme STA, le processus Spanning-Tree s'effectue indépendamment dans chaque VLAN.

- Afin de déterminer un BID le plus faible possible pour le VLAN par défaut pour les deux switches cœur de réseau (SC1 et SC2), on exécute :

- sur SC1 :

```
>enable
#conf t
(config)#spanning-tree vlan 1 priority 0
```

- sur SC2 :

```
>enable
#conf t
(config)#spanning-tree vlan 1 priority 4096
```

SC1 a désormais une valeur de priorité de pont totale de 1. SC2 quant à lui a une valeur de 4097. SC1 aura donc le BID le plus faible, il sera donc élu pont racine et SC2 sera le nouveau pont racine en cas de panne de SC1, son BID étant le deuxième plus faible.

- Sur le switch SA2 par exemple, on vérifie la nouvelle élection du pont racine avec la commande **show spanning-tree**.

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    1
            Address    0001.63B8.CBC3 ← @MAC de SC1 (pont racine)
            Cost      38
            Port      2 (FastEthernet0/2)
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
            Address    00E0.A3DB.82D3 ← Commande exécutée sur SA2
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time 20
```

Résultat sur SA2 de la commande **show spanning-tree**

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#spann
Switch(config)#
Switch(config)#interface FastEthernet0/1
Switch(config-if)#spann
Switch(config-if)#spanning-tree vlan 1 priori
Switch(config-if)#exit
Switch(config)#spanning
Switch(config)#spanning-tree vlan 1 priority 0
```

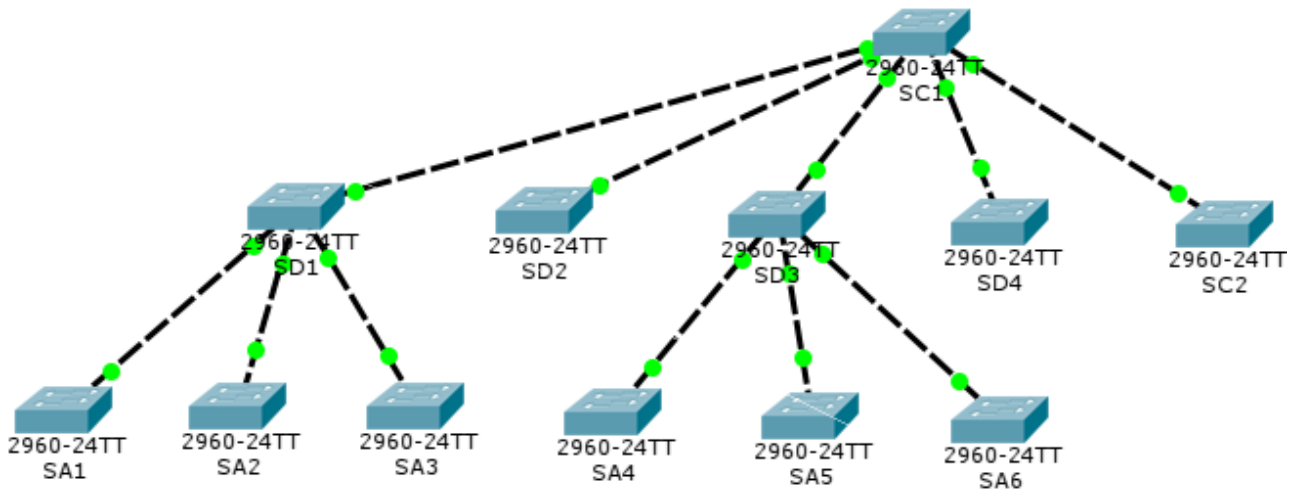
*Assignation de la valeur de priorité sur SC1*

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#spann
Switch(config)#spanning-tree vlan 1 priority 4096
```

*Assignation de la valeur de priorité sur SC2*



- Le nouvel arbre recouvrant est alors le suivant :



Arbre recouvrant topologie\_1\_3.pkt

- On supprime le switch SC1 pour tester la perte du pont racine et la correcte affectation du nouveau pont racine. Sur le switch SA3 par exemple, on vérifie quel est le nouveau pont racine et c'est effectivement SC2 :

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    4097
             Address     0001.6352.0C74
             Cost        38
             Port        2 (FastEthernet0/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
             Address     0000.0C31.A961
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
```

Résultat sur SA3 de la commande `show spanning-tree`

Pour finir, on rétablit la topologie initiale et on configure les ports reliés aux équipements terminaux afin d'optimiser STP.

- En effet, un port relié à un équipement terminal n'a pas besoin d'être inclus dans le processus STP puisqu'il ne peut pas y avoir de boucle entre un PC et... lui-même sur la liaison le reliant au switch ! Dans la topologie utilisée précédemment, ces ports sont les ports Fa0/3 des switches d'accès réseau.

- Lors de l'entrée de la commande qui permet de passer un port en mode « portfast » (l'excluant du processus STP), un message d'alerte est renvoyé par l'OS. En effet, en faisant cela on exclut totalement le port du processus STP et ce jusqu'à ce qu'on le rétablisse en mode normal. Ce qui veut dire que si l'on relie un switch sur un tel port en créant une boucle et sans aucune autre action, une tempête de broadcast s'opérera rapidement sans que STP puisse l'éviter. Le réseau tombera alors entièrement.
- On configure alors ces ports en mode « portfast » comme ceci :

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa0/3
Switch(config-if)#spann
Switch(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 FastEthernet0/3 but will only
have effect when the interface is in a non-trunking mode.
Configuration en mode "portfast" sur SA2 d'un port terminal (Fa0/3)
```

- Enfin, on vérifie cette configuration et on observe ce qui est indiqué pour le port ainsi configuré à l'aide de la commande **show running-config** sur un des switches d'accès réseau :

On peut voir la mention « **spanning-tree portfast** » sous l'interface FastEthernet0/3 que nous venons entre autres de configurer.

Nous avons donc pu observer et comprendre le processus complet STP de l'élection du pont racine à la sélection des ports racine, désigné et alternatif. En observant le comportement par défaut, nous avons pu ensuite déduire l'enjeu d'optimisation du processus en « guidant » l'algorithme STA à prendre en compte les bons ports et à élire le bon switch en tant que pont racine.

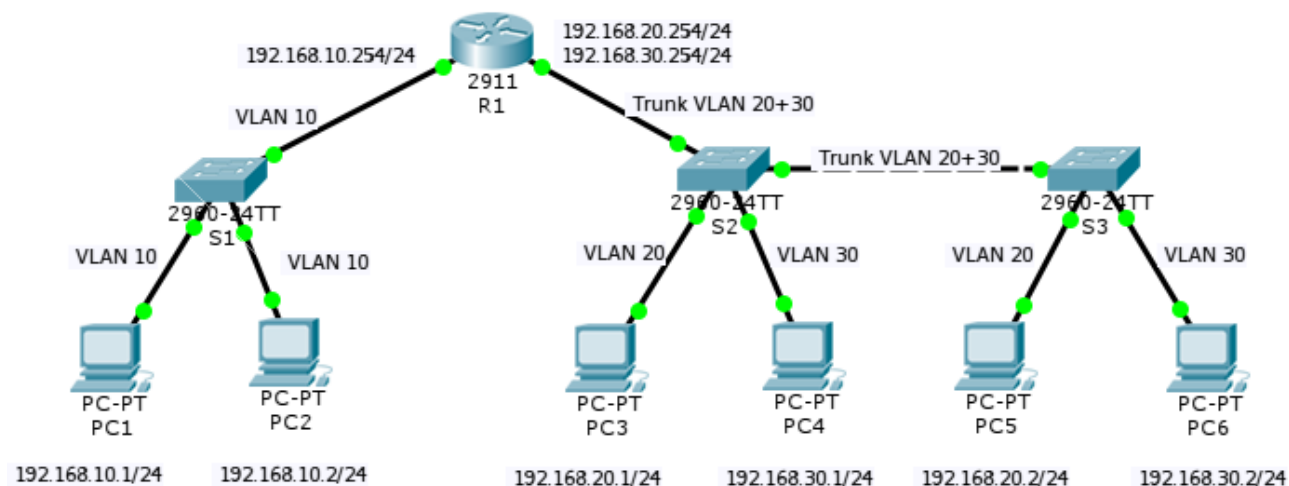
---

```
Switch#show running-config
Building configuration...

Current configuration : 1067 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Switch
!
!
!
!
!
spanning-tree mode pvst
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
    spanning-tree portfast
!
Configuration en cours sur SA2
```

## 2. Analyse de l'étiquetage VLAN

Dans cette deuxième partie, nous nous intéressons à l'étiquetage VLAN des trames transitant sur le réseau dont la topologie est la suivante :



### 2.1. Configuration de l'infrastructure réseau

Dans un premier temps, nous devons configurer l'ensemble des machines du réseau afin qu'elles puissent communiquer entre elles. On affecte aux PC les adresses IP et masques indiqués sur la topologie ainsi que la bonne gateway. Concernant les switches, on crée les VLAN indiqués.

- On configure les PC comme suit :

Nom d'équipement	Adresse IP	Masque	Gateway
PC1	192.168.10.1	255.255.255.0	192.168.10.254
PC2	192.168.10.2	255.255.255.0	192.168.10.254
PC3	192.168.20.1	255.255.255.0	192.168.20.254
PC4	192.168.30.1	255.255.255.0	192.168.30.254
PC5	192.168.20.2	255.255.255.0	192.168.20.254
PC6	192.168.30.2	255.255.255.0	192.168.30.254

- Ensuite, on configure le switch S1 en créant un VLAN 10 nommé « Admin » auquel on y affecte tous les ports. Les ports Fa0/1 et Fa0/2 seront en mode portfast.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int vlan 10
Switch(config-if)#descr
Switch(config-if)#description Admin
Switch(config-if)#int range gi0/1,fa0/1-2
Switch(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
Switch(config-if-range)#
%LINK-5-CHANGED: Interface Vlan10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up

Switch(config-if-range)#int range fa0/1-2
Switch(config-if-range)#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 will be configured in 2 interfaces due to the range command
but will only have effect when the interfaces are in a non-trunking mode.
Switch(config-if-range)#end
```

#### *Configuration du switch S1*

- On configure maintenant les switches S2 et S3 en affectant les ports reliés aux PC3 et 5 au VLAN 20 nommé « RT » et ceux reliés aux PC4 et 6 au VLAN 30 nommé « Info ». Le lien entre S2 et S3 est un trunk-VLAN agrégeant les VLAN 20 et 30 et le VLAN natif 666.

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int vlan 20
Switch(config-if)#description RT
Switch(config-if)#int fa0/1
Switch(config-if)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up

Switch(config-if)#spanning-tree portfast
```

#### *Configuration des switches S2 et S3*

```
Switch(config-if)#int vlan 30
Switch(config-if)#description Info
Switch(config-if)#int fa0/2
Switch(config-if)#switchport access vlan 30
% Access VLAN does not exist. Creating vlan 30
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed state to up

Switch(config-if)#spanning-tree portfast
```

#### *Configuration des switches S2 et S3 - suite*

```
Switch(config-if)#int gi0/1
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
to up

Switch(config-if)#switchport trunk allowed vlan 20,30
Switch(config-if)#switchport trunk native vlan 666
Switch(config-if)#end
```

*Configuration des switches S2 et S3 - fin*

- Enfin, on configure le routeur R1 d'un côté pour le sous-réseau 192.168.10.0/24 et de l'autre côté en mode « router-on-a-stick » pour les sous-réseaux 192.168.20.0/24 et 192.168.30.0/24.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int gi0/1
Router(config-if)#ip address 192.168.10.254 255.255.255.0
Router(config-if)#int gi0/2.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2.20, changed state to up
ip address 192.168.20.254 255.255.255.0

% Configuring IP routing on a LAN subinterface is only allowed if that
subinterface is already configured as part of an IEEE 802.10, IEEE 802.1Q,
or ISL vLAN.

Router(config-subif)#encapsulation dot1q
% Incomplete command.
Router(config-subif)#encapsulation dot1q ?
 <1-1005> IEEE 802.1Q VLAN ID
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.254 255.255.255.0
Router(config-subif)#int gi0/2.30
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2.30, changed state to up
enca
Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip address 192.168.30.254 255.255.255.0
```

*Configuration du routeur R1*

On configure l'interface Gi0/1 en lui affectant les paramètres IP 192.168.10.254/24.

On configure l'interface Gi0/2 en créant deux sous interfaces avec les paramètres IP 192.168.20.254/24 et 192.168.30.254/24. On y active respectivement l'encapsulation pour les VLAN 20 et 30.

- On teste maintenant à l'aide de pings la communication entre les PC, nous permettant de vérifier notre configuration.

		Émetteur					
		PC1	PC2	PC3	PC4	PC5	PC6
Destinataire	PC1		OK	OK	OK	OK	OK
	PC2	OK		OK	OK	OK	OK
	PC3	NOK	NOK		OK	OK	OK
	PC4	NOK	NOK	OK		OK	OK
	PC5	NOK	NOK	OK	OK		OK
	PC6	NOK	NOK	OK	OK	OK	

On s'aperçoit que la communication entre les PC connectés à l'interface physique Gi0/1 vers ceux connectés à l'interface physique Gi0/2 est unilatéralement impossible. Cependant, depuis les PC1 et 2 on peut pinger avec succès les deux sous-interfaces virtuelles de Gi0/2. En passant en mode simulation en filtrant les messages ICMP, on voit que au niveau du switch S2 Packet Tracer indique gi0/2 a reçu le message mais que son en-tête est de type 802.1q alors que l'interface est en mode accès simple... nous avons oublié de configurer gi0/2...

On configure alors l'interface gi0/2 en mode trunk agrégeant les VLAN 20 et 30 et pour VLAN natif le VLAN 666 :

```
Switch(config)#int gi0/2
Switch(config-int)#switchport mode trunk
Switch(config-int)#switchport trunk allowed vlan 20,30
Switch(config-int)#switchport trunk native vlan 666
Switch(config-int)#end
```

At Device: S2  
Source: PC1  
Destination: PC3

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer3	Layer3
Layer2: Dot1q Header 0090.2BBE.2603 >> 000D.BDAE.5184	Layer2
Layer 1: Port GigabitEthernet0/2	Layer1

En effet, notre paquet est taggé 20 et donc appartenant au VLAN 20 pour aller vers une machine du sous réseau 192.168.20.0/24. Or, sans configuration, le port gi0/2 est par défaut en mode accès et appartenant au VLAN 1, il ne peut donc pas faire transiter de paquet taggé 20.

- The receiving port is an access port but the receiving frame is an IEEE 802.1q frame.
- The Switch drops the invalid frame.

Trame (ping) arrivant sur S2 à destination de PC3 depuis PC1 (modèle OSI)

## 2.2. Observation des trames étiquetées VLAN

Dans cette seconde sous-partie, nous observons l'étiquetage des trames sur un lien trunk-VLAN. Pour rappel, ce type de lien permet d'acheminer entre deux switches par exemple, du trafic issu de plusieurs VLAN sans avoir besoin de mettre un lien par VLAN entre les deux équipements.

- Concernant l'étiquetage des trames sur un trunk-VLAN, les trames sont encapsulées dans un PDU particulier qui ajoute une en-tête avec des informations sur le VLAN dont le numéro de VLAN. Ceci est défini par la norme IEEE 802.1Q (dot1Q).
- Un champ VLAN ID codé sur 12 bits est rajouté aux trames afin d'identifier le VLAN auquel appartient le message transporté par le PDU.
- Pour visualiser la différence entre une trame Ethernet classique (802.3) et une trame taguée (802.1Q), on reprend la [topologie de la sous-partie précédente](#) et on filtre uniquement les messages ICMP (ping) en mode simulation.
- On envoie un ping de PC3 vers PC5 en mode pas à pas pour observer le contenu de la trame en tout point du réseau (arrivée sur le switch, sortie du switch, arrivée sur le PC, ...)
- Premièrement, la trame émise par PC3 vers le switch S2 est la suivante :

At Device: S2  
Source: PC3  
Destination: PC5

### In Layers

Layer7
Layer6
Layer5
Layer4
Layer3
Layer 2: Ethernet II Header 000D.BDAE.5184 >> 00D0.58A0.D2C7
Layer 1: Port FastEthernet0/1

Trame émise par PC3 (modèle OSI)

En couche 2, le PDU est une trame Ethernet II dont l'en-tête contient l'adresse destination de PC5 (00-D0-58-A0-D2-C7) et l'adresse source de PC3 (00-0D-BD-AE-51-84).

- Entre S2 et S3, la trame a désormais une en-tête 802.1Q avec le numéro de VLAN correspondant au PC3, soit le VLAN 20 (= appartenance du port Fa0/1 de S2). La trame est donc plus longue et la valeur d'étiquetage correspond.

Ethernet 802.1q

0		4		7		8		14		19	
PREAMBLE: 1010 1010				S F D	DEST ADDR: 00D0.58A0.D2C7			SRC ADDR: 000D.BDAE.5184			
TPID: 0x81		TCI: 0x14		TYPE: 0x1		DATA (VARIABLE LENGTH)				FCS: 0x0	

En-tête 802.1Q de la trame émise par S2

At Device: S3  
Source: PC3  
Destination: PC5

**In Layers**

Layer7
Layer6
Layer5
Layer4
Layer3
Layer 2: Dot1q Header 000D.BDAE.5184 >> 00D0.58A0.D2C7
Layer 1: Port GigabitEthernet0/1

Trame émise par S2 (modèle OSI)

La valeur du VLAN ID de l'en-tête 802.1Q est indiquée comme étant 0x14. Cela correspond à  $14|_{16} = (1*16 + 4)|_{10} = 20|_{10}$ . On retrouve donc bien le bon numéro de VLAN auquel appartient le port Fa0/1 du switch S2 et donc le PC3. Ainsi, la trame est correctement véhiculée sur le trunk-VLAN.

- Pour finir, on vérifie le contenu de la trame émise de S3 vers PC5. Le contenu est le suivant :

Ethernet II

0		4		8		14		19		
PREAMBLE: 101010...1011				DEST MAC: 00D0.58A0.D2C7			SRC MAC: 000D.BDAE.5184			
TYPE: 0x800		DATA (VARIABLE LENGTH)				FCS: 0x0				

En-tête Ethernet II de la trame émise par S3 vers PC5

**Out Layers**

Layer7
Layer6
Layer5
Layer4
Layer3
Layer 2: Ethernet II Header 000D.BDAE.5184 >> 00D0.58A0.D2C7
Layer 1: Port(s): FastEthernet0/1

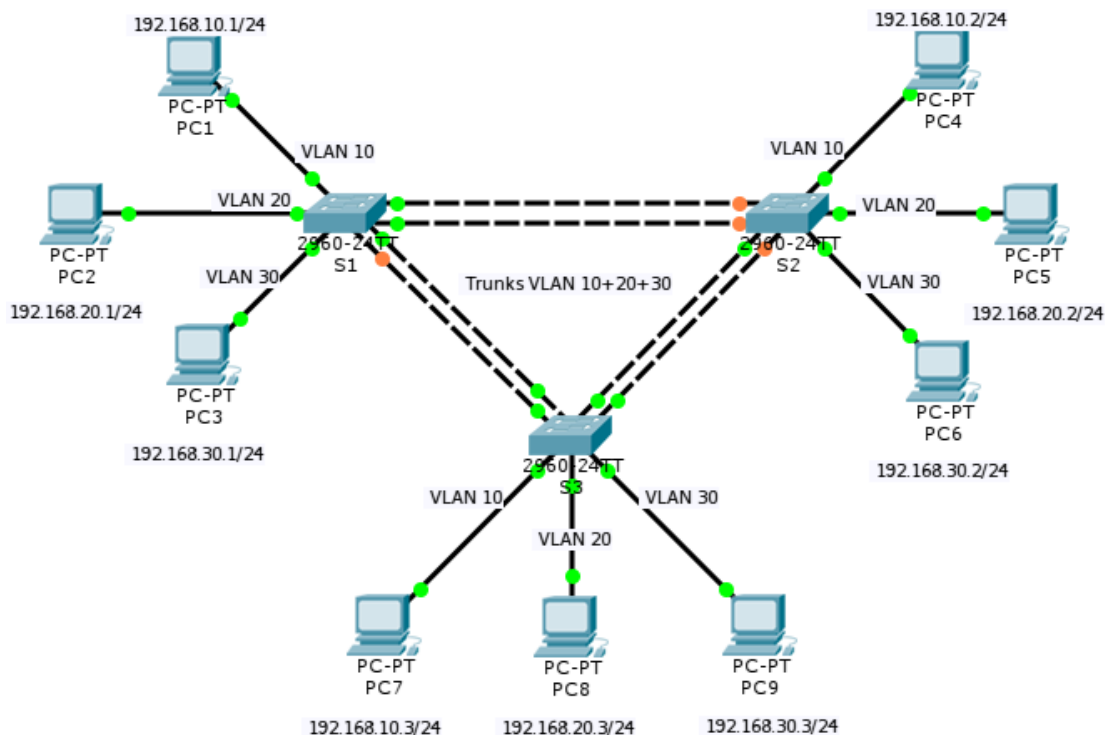
Trame émise par S3 (modèle OSI)

On voit effectivement que l'en-tête 802.1Q avec le tag VLAN a été supprimé avant de retransmettre sur le lien vers PC5. On en conclut donc que ce sont bien les switches qui taguent les trames en les encapsulant dans un en-tête 802.1Q sur les liens appropriés.



### 3. Spanning Tree et multi-VLAN

Dans cette dernière partie, l'objectif est de comprendre le comportement de Spanning Tree dans une topologie comprenant plusieurs VLAN. Nous savons déjà que l'algorithme STA est appliqué au sein de chaque LAN et de chaque VLAN le cas échéant. Pour étudier cela, nous allons utiliser la topologie suivante :



- On remarque que l'administrateur réseau a mis deux liens Ethernet pour interconnecter les switches. Cela permet d'augmenter la bande passante en configurant un trunk ou une agrégation de liens ( $\neq$  agrégation de VLAN ou trunk-VLAN). Les deux liens physiques d'interconnexion sont en réalité perçus par le switch comme un seul lien logique. Cependant, ici ces liens ne sont pas configurés de la sorte mais ce sont des trunk-VLAN afin de faire transiter le trafic des trois VLAN différents au sein du réseau.
- Le protocole Spanning Tree va donc devoir empêcher les boucles situées entre les deux switches du fait de la double interconnexion et, en plus, il va devoir empêcher la formation de boucle entre les trois switches due à la topologie en anneau de ce réseau. On peut dès à présent remarquer cela sur l'image précédente, où l'on voit que « l'anneau extérieur » est totalement bloqué et que sur « l'anneau intérieur », seuls les liens entre S1 et S3 et S3 et S2 sont fonctionnels... on peut garder cela en mémoire pour la suite...

### 3.1. Configuration de l'infrastructure réseau

Comme à chaque fois, dans une première sous-partie nous configurons dans Packet Tracer tous les équipements afin d'effectuer par la suite nos observations.

- On configure les PC de la façon suivante :

Nom d'équipement	Adresse IP	Masque
PC1	192.168.10.1	255.255.255.0
PC2	192.168.20.1	255.255.255.0
PC3	192.168.30.1	255.255.255.0
PC4	192.168.10.2	255.255.255.0
PC5	192.168.20.2	255.255.255.0
PC6	192.168.30.2	255.255.255.0
PC7	192.168.10.3	255.255.255.0
PC8	192.168.20.3	255.255.255.0
PC9	192.168.30.3	255.255.255.0

- Sur les switches S1, S2 et S3 nous créons trois VLAN auxquels nous y affectons certains ports comme suit :

Numéro VLAN	Nom VLAN	Ports affectés
10	Admin	Fa0/1
20	RT	Fa0/2
30	Info	Fa0/3

De plus, les ports que nous venons d'affecter seront en mode portfast. Les commandes à entrer sur les trois switches sont :

```
>enable
#conf t
(config)#int vlan 10
(config-vlan)#description Admin
(config-vlan)#int vlan 20
(config-vlan)#description RT
(config-vlan)#int vlan 30
(config-vlan)#description Info
(config-vlan)#int fa0/1
(config-if)#switchport access vlan 10
(config-if)#int fa0/2
(config-if)#switchport access vlan 20
(config-if)#int fa0/3
(config-if)#switchport access vlan 30
(config-if)#int range fa0/1-3
(config-if)#spanning-tree portfast
```

- Pour finir la configuration du réseau, on configure les trunk-VLAN entre les trois switches agrégeant les VLAN 10, 20 et 30 et de VLAN natif 666. Pour cela, on entre les commandes suivantes sur les trois switches :

```
(config)#int range fa0/21-24
(config-if)#switchport mode trunk
(config-if)#switchport trunk allowed vlan 10,20,30
(config-if)#switchport trunk native vlan 666
```

La configuration est désormais terminée. Nous pouvons la tester à l'aide de ping pour voir si celle-ci est correcte. A noter que nous pouvons seulement communiquer entre PC de même VLAN car sans routeur, la communication inter-VLAN est impossible. De plus, dans cette topologie les équipements appartiennent à des sous-réseaux différents. Les équipements du sous réseau 192.168.10.0/24 appartiennent au VLAN 10, ceux du sous-réseau 192.168.20.0/24 au VLAN 20 et ceux du sous-réseau 192.168.30.0/24 au VLAN30. Par conséquent, pour la même raison que pour la communication inter-VLAN, la communication inter-réseau n'est pas possible.

- Tous les pings sont OK, signifiant que notre configuration est correct.
- Comme expliqué précédemment, la communication entre PC de VLAN différents aussi appelé communication inter-VLAN est impossible. Ces pings ne peuvent donc pas fonctionner.

### 3.2. Analyse du processus de Spanning Tree

On peut désormais observer le processus Spanning Tree au sein d'un réseau avec plusieurs VLAN. Nous savons déjà que ce processus s'effectue indépendamment pour chaque VLAN, effectivement, nous allons voir que plusieurs instances de l'algorithme STA (une par VLAN) exécutent [les étapes du processus](#).

- Sur les trois switches, on exécute la commande **show spanning-tree** :

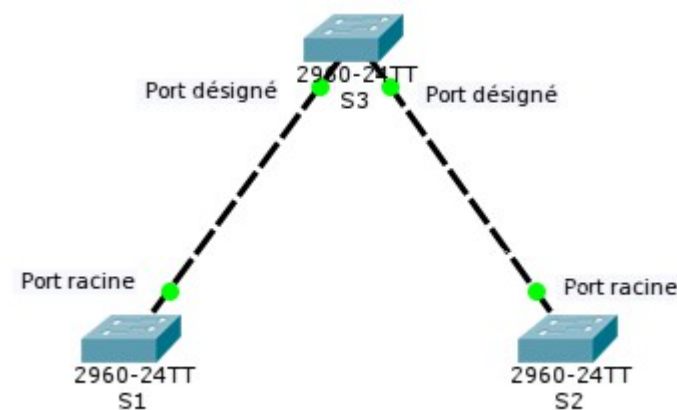
Switch	@MAC	Priorité de pont	Priorité de pont en système étendu		
			VLAN 10	VLAN 20	VLAN 30
S1	00-01-96-2D-66-E8	32769	32778	32788	32798
S2	00-60-47-1C-9D-B6	32769	32778	32788	32798
S3	00-01-42-5A-7A-43	32769	32778	32788	32798

ID VLAN	@MAC pont racine	Switch élu pont racine
10	00-01-42-5A-7A-43	S3
20	00-01-42-5A-7A-43	S3
30	00-01-42-5A-7A-43	S3

- Ensuite, on fait de même avec la même commande pour identifier le rôle des ports des trois switches pour chaque VLAN :

Switch	Port	État du port		
		VLAN 10	VLAN 20	VLAN 30
S1	Fa0/21	Désigné	Désigné	Désigné
	Fa0/22	Désigné	Désigné	Désigné
	Fa0/23	Racine	Racine	Racine
	Fa0/24	Alternatif	Alternatif	Alternatif
S2	Fa0/21	Alternatif	Alternatif	Alternatif
	Fa0/22	Alternatif	Alternatif	Alternatif
	Fa0/23	Racine	Racine	Racine
	Fa0/24	Alternatif	Alternatif	Alternatif
S3	Fa0/21	Désigné	Désigné	Désigné
	Fa0/22	Désigné	Désigné	Désigné
	Fa0/23	Désigné	Désigné	Désigné
	Fa0/24	Désigné	Désigné	Désigné

- L'élection du pont racine a donné le même résultat pour chaque instance. De plus, l'état des ports est aussi le même pour chaque VLAN. Les trois arbres recouvrant sont par conséquent les mêmes :



*Arbre recouvrant des VLAN 10, 20 et 30*

- En mettant en relation le tableau précédent et la topologie, on voit très facilement que seuls 2 liens sont actifs : un lien entre S1 (port Fa0/23) et S3 et un autre entre S2 (port Fa0/23) et S3. 4 liens sont donc inactifs sur les 6. Cela correspond à la couleur des points car les résultats des différentes instances de STA sont identiques, ce qui ne serait pas le cas si les résultats étaient différents. Pour ne pas faire d'erreur le cas échéant, on s'habitue donc à ne pas se fier aux couleurs lorsqu'il y a plusieurs VLAN.

- Chaque lien actif transporte donc le trafic de 3 VLAN puisque ce sont des trunk-VLAN autorisant cela et qu'avec seulement deux liens actifs, un seul chemin existe pour communiquer entre les machines au sein de ce réseau.
- L'ajout d'un lien supplémentaire pour interconnecter les switches n'a donc pas permis de doubler la bande passante. Comme précisé dans l'introduction de la partie 3, pour éviter les boucles dues à la topologie en anneau ET à l'utilisation de deux liens non agrégés, STP bloque un maximum de lien, ne laissant que deux liens actifs et un chemin comme nous venons de le voir. Il aurait fallu agréger les liens (EtherChannel chez Cisco) pour que les doubles liens d'interconnexion soient pris en compte comme un seul lien logique afin de doubler la bande passante. STP aurait seulement pris en compte la boucle possible due à la topologie en anneau puisque les switches auraient en réalité qu'un seul lien logique d'interconnexion (par exemple ils ne feraient pas la différence entre le trafic arrivant sur le port Fa0/21 ou Fa0/22 et donc ne renverrait pas le trafic provenant d'un de ces deux liens vers un de ces deux mêmes liens).

### 3.3. Optimisation de la configuration du Spanning Tree

À travers cette ultime étape, nous allons apprendre à optimiser la configuration de Spanning Tree en « truquant » le processus, l'élection du pont racine et la sélection des ports. D'une part, cela va permettre d'élire un switch pont racine le plus judicieux et d'autre part, comme dans la topologie précédente, cela va permettre d'équilibrer le trafic en optimisant les doubles liens d'interconnexion des switches.

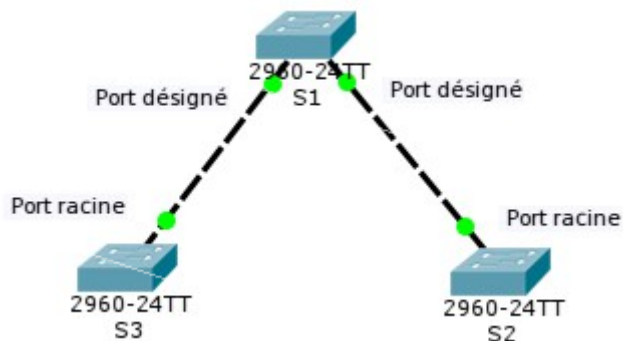
En supposant que les trois VLAN acheminent un trafic égal, chaque VLAN devrait donc avoir un pont racine différent :

ID VLAN	Pont racine
10	S1
20	S2
30	S3

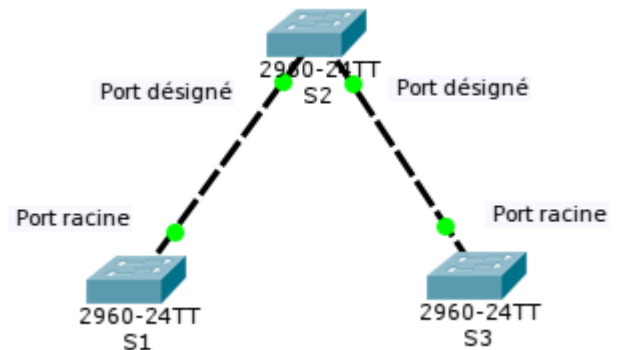
- On modifie alors les valeurs de priorité sur les switches par VLAN. Comme vu dans la partie 1.3, on assigne la plus petite valeur de priorité au switch à élire pont racine pour le VLAN voulu. On exécute alors les commandes suivantes :

```
S1(config)#spanning-tree vlan 10 priority 0
S2(config)#spanning-tree vlan 20 priority 0
S3(config)#spanning-tree vlan 30 priority 0
```

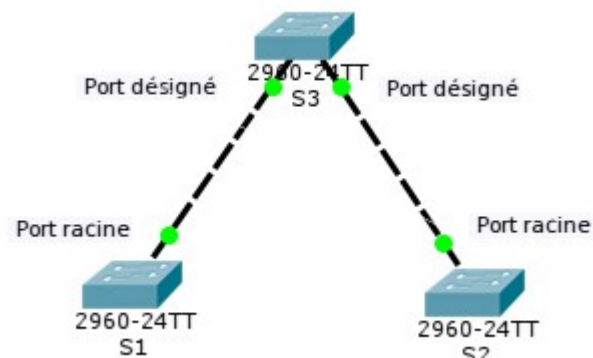
- Après 30 secondes, on vérifie à l'aide de la commande **show spanning-tree** sur chacun des switches si l'élection est réussie. Sur S1, dans la section VLAN0010 on voit sous « Root ID » que « This bridge is the root » (= ce switch est le pont racine). Sur S2, dans la section VLAN0020 on voit sous « Root ID » que c'est le pont racine. Enfin, de même pour S3 dans la section VLAN0030.
- Les arbres recouvrants sont ainsi les suivants :



*Arbre recouvrant VLAN 10 (topologie priorisée)*



*Arbre recouvrant VLAN 20 (topologie priorisée)*



*Arbre recouvrant VLAN 30 (topologie priorisée)*

- En observant les trois arbres recouvrants, on voit que trois liens sont actifs : S1 ↔ S3, S1 ↔ S2 et S2 ↔ S3. Ce qui fait donc trois liens actifs contre trois liens inactifs. Cela apporte une petite amélioration d'un peu moins de 17% étant donné que seulement 2 liens étaient actifs avant.
- Toujours en observant les trois arbres, nous pouvons affirmer que chaque lien transporte le trafic de 2 VLAN puisque chaque lien actif apparaît dans deux arbres différents seulement.
- Cette amélioration n'est donc pas totalement suffisante, la charge n'étant pas répartie de façon équitable. Il faudrait 3 liens actifs avec chacun transportant le trafic de 1 seul VLAN.

Pour finir, nous conservons le pont racine comme précédemment mais nous allons essayer de faire en sorte que les liens actifs acheminent le trafic des trois VLAN.

- Ainsi, le trafic est complètement partagé par rapport à la configuration initiale puisque tous les liens restent actifs. Les liens utilisés sont cependant différents selon le VLAN puisque sinon, il y aurait des boucles. C'est pour cela qu'il est important d'utiliser des trunk-VLAN car ils peuvent acheminer indifféremment le trafic des différents VLAN selon nos besoins. Par exemple, le VLAN10 transite par un des liens entre S1 et S2 tandis que le VLAN20 transite par l'autre lien entre S1 et S2. Les 6 liens restent donc tous actifs d'un point de vue global mais d'un point de vue VLAN, deux sont utilisés par VLAN.

Une autre étape de l'algorithme STA est ensuite de calculer le coût des chemins vers le pont racine à travers les différents ports du switch afin de déterminer le meilleur chemin vers le pont racine.

- Sur nos switches, par défaut le coût associé aux port que nous utilisons est de 19. En effet, tous les ports sont de type FastEthernet soit une bande passante de 100 Mbps. Ces coûts sont définis par l'IEEE.

Pour obtenir les arbres recouvrants voulus, nous devons « truquer » également l'élection du port racine sur les switches autres que le pont racine (pour rappel, un pont racine n'a que des ports désignés).

- Toujours avec la commande **show spanning-tree**, on consulte la valeur de priorité des ports. Par défaut, cette valeur est de 128.

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/1	Desg	FWD	19	128	1	P2p
Fa0/21	Desg	LRN	19	128	21	P2p
Fa0/22	Desg	LRN	19	128	22	P2p
Fa0/23	Root	LRN	19	128	23	P2p
Fa0/24	Altn	BLK	19	128	24	P2p

*Interfaces prises en compte par STP sur un switch*

- Pour qu'un port soit sélectionné port racine, il faut que sa valeur de priorité soit la plus faible possible. En effet, STA procède de la même façon que lors de l'élection du pont racine pour sélectionner les statuts des ports.
- Pour le VLAN 10 : sur S1, on priorise donc les ports Fa0/21 et Fa0/23 vers S2 et S3 pour qu'ils soient choisis comme étant actifs plutôt que Fa0/22 et Fa0/24 (qui eux serviront pour un autre VLAN).

```
S1(config)#int range fa0/21,fa0/23
S1(config-if-range)#spanning-tree vlan 10 port-priority 112
```

- On vérifie alors la configuration des ports faite précédemment :

Fa0/21	Desg LSN 19	112
Fa0/22	Desg FWD 19	128
Fa0/23	Desg LSN 19	112
Fa0/24	Desg FWD 19	128

Priorité modifiée

*Priorisation des ports Fa0/21 et Fa0/23 sur S1 pour VLAN10 (section VLAN0010)*

- Pour le VLAN 20 : sur le switch S2 et d'après le même principe, on configure la priorité des ports Fa0/22 et Fa0/23. On exécute alors les commandes suivantes et on vérifie la configuration :

```
S2(config)#int range fa0/22-23
S2(config-int-range)#spanning-tree vlan 20 port-priority 112
```

Fa0/21	Desg FWD 19	128
Fa0/22	Desg LSN 19	112
Fa0/23	Desg LSN 19	112
Fa0/24	Desg FWD 19	128

*Priorisation des ports Fa0/22 et Fa0/23 sur S2 pour VLAN20 (section VLAN0020)*

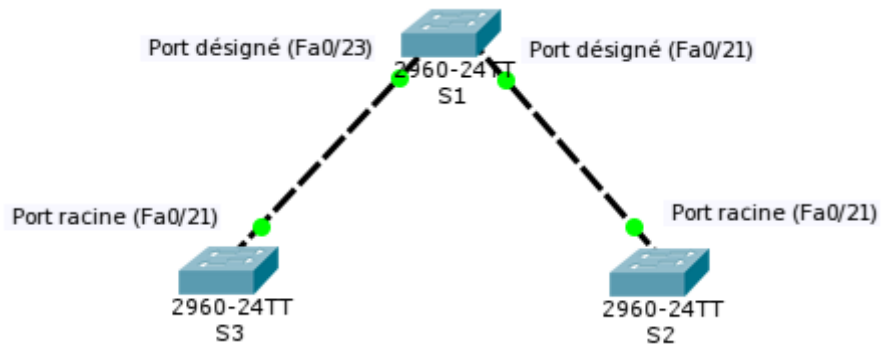
- Enfin, pour le VLAN 30 : sur le switch S3 et toujours d'après le même principe, on configure la priorité des ports Fa0/22 et Fa0/24 puis on exécute les commandes suivantes avant de vérifier la configuration :

```
S3(config)#int range fa0/22,fa0/24
S3(config-int-range)#spanning-tree vlan 30 port-priority 112
```

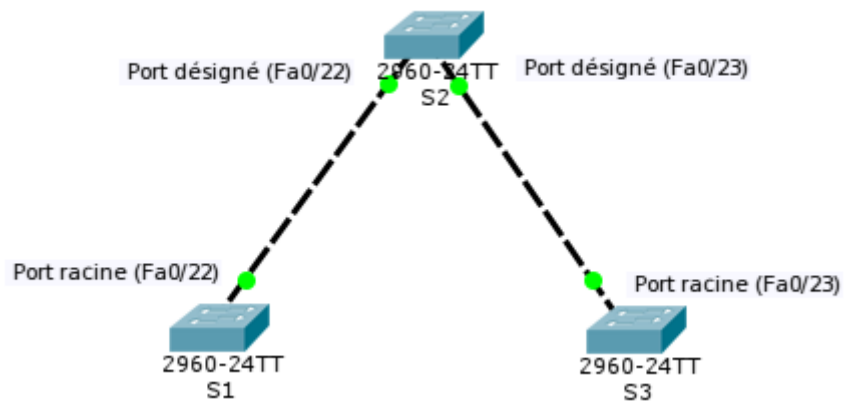
Fa0/21	Desg FWD 19	128
Fa0/22	Desg LSN 19	112
Fa0/23	Desg FWD 19	128
Fa0/24	Desg LSN 19	112

*Priorisation des ports Fa0/22 et Fa0/24 sur S3 pour VLAN30 (section VLAN0030)*

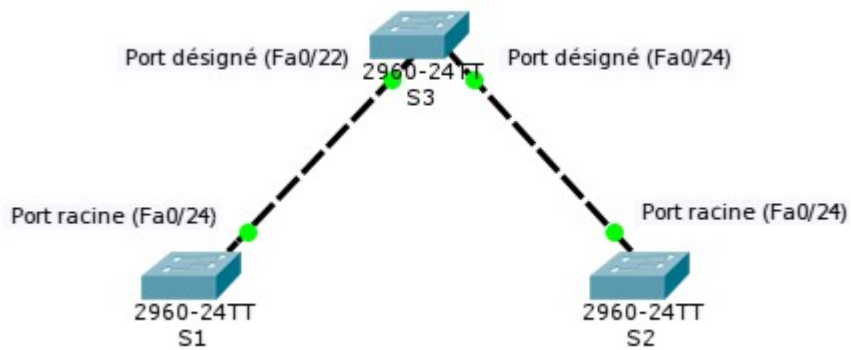




*Arbre recouvrant priorisé du VLAN 10*



*Arbre recouvrant priorisé du VLAN 20*



*Arbre recouvrant priorisé du VLAN 30*

## Conclusion

Dans ce TP, nous avons pu observer en détail le fonctionnement du processus de STP et les étapes de son algorithme STA. En effet, Packet Tracer nous permet de visualiser les échanges STP en mode pas à pas et de consulter le contenu des trames BPDU entre les switches. Le processus STP s'effectue en quatre étapes dont voici les grandes lignes :

1. élection du pont racine, qui sera le switch référent pour STP dans la topologie,
2. détermination des meilleurs chemins vers le pont racine depuis chacun des switch (calcul de coût avec spécification IEEE en fonction de la bande passante des liens),
3. sélection des ports racines sur chaque switch, via lesquels on accède au meilleur chemin vers le pont racine,
4. sélection des ports désignés et alternatifs, pour éviter les boucles, les ports alternatifs bloquant le trafic.

Pour rappel, cet algorithme s'exécute dans chaque VLAN de façon indépendante. En effet, le numéro de VLAN est compris dans le BID (Bridge Identifier).