

# Le protocole STP

## Objectifs

Nous allons observer le fonctionnement du protocole STP sur un réseau simple.

## 1 Utilisation de l'outil de simulation du réseau

Ce réseau utilisé dans cette séance est "simulé" à l'aide d'un outil d'isolation fourni par le noyau Linux (les *Namespaces* réseau). Une description plus complète est disponible à l'adresse suivante

<http://chaput.perso.enseeiht.fr/teaching/ressources/tp-reseaux-virtualises>

Si vous avez déjà utilisé ces outils, vous pouvez passer à la section suivante.

### 1.1 Installation des fichiers

La page web citée plus haut décrit les différentes façons d'obtenir les fichiers vous permettant de démarrer la séance.

Grâce à l'aide de cette page et/ou de votre enseignant-e, vous pouvez donc maintenant démarrer un *shell* dans le répertoire contenant les fichiers de la séance voulue et dans lequel vous prendrez l'identité de l'administrateur (nécessaire pour la suite des opérations) :

```
$ cd le-dossier-de-mon-tp
$ sudo su
#
```

vous pouvez alors démarrer la séance.

### 1.2 Démarrage et arrêt du simulateur

Le lancement du simulateur se fait de la façon suivante

```
# ./creerReseau
```

Un terminal est alors ouvert sur les machines principales du réseau.

Une liste d'options utilisables pour Wireshark vous est également fournie. Vous pourrez l'utiliser pour observer le trafic sur les interfaces des différentes machines.

Pour cela, vous lancerez, par exemple, dans le même terminal

```
# wireshark -i /tmp/nssi/host1/v0
```

Ce qui vous permettra d'observer le trafic sur l'interface v0 de la machine host1.

Notez que vous pouvez bien sûr lancer la commande `tcpdump` directement dans le terminal de la machine correspondante!

Vous pourrez arrêter la simulation ainsi :

```
# ./destruireReseau
```

**Attention**, lorsque vous arrêtez le réseau, toutes les manipulations faites sur les machines sont définitivement perdues!

### 1.3 Lancement d'une commande ou d'un terminal dans une machine

Si vous avez malencontreusement fermé le terminal d'une machine, vous pouvez le relancer de la façon suivante

```
# ./creerReseau -r machine
```

où *machine* est le nom de la machine.

Vous pouvez également lancer une commande sur une machine :

```
# ./creerReseau -r machine "commande et options"
```

Les guillemets sont nécessaires, par exemple

```
# ./creerReseau -r m1 "ip link show"
```

## 2 Le réseau de cette séance

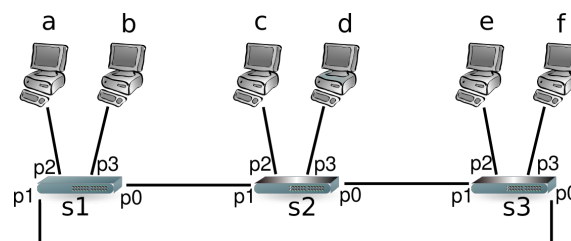


FIGURE 1 – Un réseau ponté présentant un cycle

Dans le réseau qui vous est fourni, les machines (hôtes et *switch Ethernet*) ont été créées et reliées entre elles. Vous n'avez donc aucun "branchement" à faire, uniquement la configuration logicielle (configuration et activation des liens).

Les interfaces sont actives, sauf l'interface p1 de s1 et l'interface p0 de s3.

Nous allons faire ensemble cette configuration progressivement.

## 3 Le Spanning Tree

Nous allons dans cette section mettre en place le protocole de *Spanning Tree* entre différents ponts interconnectés.

### 3.1 Ethernet et la diffusion

Afin de comprendre l'intérêt du protocole STP, intéressons-nous à ce qui peut se produire sans lui dans un réseau tel que le notre.

#### ▷ Exercice 1 : Interconnexion des réseaux Ethernet

*Essayez de faire communiquer des machines au travers d'un tel réseau; qu'observez-vous?*

*Recommençons maintenant après avoir activé les interfaces p1 de s1 et p0 de s3.*

*Que se passe-t-il alors? Comment cela s'explique-t-il?* ■

On arrêtera le réseau avant de le relancer pour la suite de la séance.

## 3.2 Activation du protocole STP

Sur un pont Linux, le Spanning Tree est activé grâce à la commande suivante

```
# brctl stp br0 on
```

(il sera désactivé par l'option `off`). L'état du pont relatif au Spanning Tree peut être observé grâce à la commande

```
# brctl showstp br0
```

### ▷ Exercice 2 : Mise en place du Spanning Tree

*Après avoir lancé des outils d'observation du réseau, activez le Spanning Tree sur vos ponts. Observez les BPDUs échangées et l'évolution de l'état de chacun des ponts.*

*On activera ensuite les interfaces `p1` de `s1` et `p0` de `s3`.* ■

## 3.3 Basculement en cas de défaillance

Un autre intérêt du STP est la possibilité de changer de chemin en cas de modification de la topologie, par exemple en cas de défaillance d'un lien.

### ▷ Exercice 3 : Simulation de la défaillance d'un lien

*Désactivez une des interfaces (du cycle) utilisées entre deux machines et observez comment le trafic bascule sur d'autres liens disponibles.* ■