

# La commutation Ethernet

## Objectifs

**Pont Ethernet :** Nous allons mettre en place et configurer un réseau ethernet constitué de plusieurs segments interconnectés par des ponts afin de comprendre le fonctionnement de ces derniers.

## 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 utilisé

Nous allons mettre en place le réseau de la figure 1.

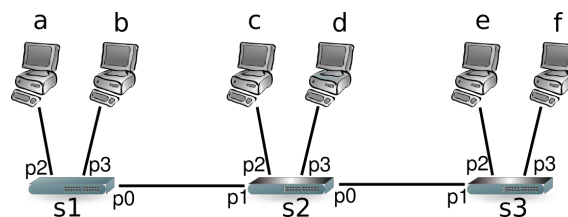


FIGURE 1 – Un réseau ponté simple

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).

Nous allons faire ensemble cette configuration progressivement.

## 3 Mise en place d'un réseau ponté

Nous allons donc activer les fonctions de pontage sur les équipements *s1*, *s2* et *s3*.

### 3.1 Création d'un pont

Une machine Linux peut être utilisée comme un pont (et même un ensemble de ponts) Ethernet (de niveau liaison donc). Pour cela, un pont est créé grâce à la commande suivante (ici le pont aura le nom *br0*) :

```
# ip link add name br0 type bridge
```

Le pont ainsi créé se nomme `br0` ; il pourra être détruit grâce à l'option `del` de la commande `ip link`.

▷ **Exercice 1 : Mise en place des ponts**

*Configurer les ponts de la figure 1. Pour cela, on créera un pont sur chacune des machines `s1`, `s2` et `s3`.* ■

Comme toute interface, un pont doit être activé (`ip link set dev br0 up`) afin de pouvoir être utilisé.

### 3.2 Ajout d'interfaces au pont

L'intégration d'une interface dans un pont est réalisée par la commande suivante

```
# ip link set dev v0 master br
```

Ici, c'est donc l'interface `v0` qui est ajoutée dans le pont `br`. Attention, cette interface doit être active, mais ne doit pas être configurée au niveau réseau (elle ne doit pas avoir d'adresse IP). Elle sera donc par exemple dans un premier temps activée par

```
# ip link set dev v0 up
```

Une interface pourra être supprimée du pont grâce à l'option `nomaster` de la commande `ip link`.

▷ **Exercice 2 : Configuration du pont**

*Réalisez la configuration logicielle du pont ethernet.* ■

### 3.3 Activation d'un pont

C'est la même commande qui permet d'activer un pont et une interface IP :

```
# ip set dev br0 up
```

À partir de cet instant, deux machines situées de part et d'autre du pont peuvent commencer à échanger des paquets. Le pont peut évidemment être désactivé grâce à l'option `down` de la commande `ip`.

▷ **Exercice 3 : Activation du pont**

*Activez votre pont et testez-en le bon fonctionnement. Pour cela, vous devrez configurer les adresses IP des machines `a`, `b`, ...* ■

### 3.4 Fonctionnement du pont

Le système de pontage de Linux offre, par le biais d'options de la commande `ip`, la possibilité d'observer le fonctionnement du pont. Le contenu des tables d'adresses associées à chaque interface est donné par la commande suivante :

```
# bridge fdb show br br0
```

Des informations sur les interfaces configurées dans le pont sont obtenues par la commande suivante

```
# ip link show master br0
```

▷ **Exercice 4 : Observer notre pont**

*Observez l'état du pont avant toute communication. Configurez le réseau sur l'ensemble des machines reliées au pont puis démarrez un outil d'observation du réseau sur chacune d'entre elles.*

*Échangez alors des paquets IP entre ces machines et observez le trafic induit ainsi que les évolutions des tables du pont.* ■

### 3.5 Congestion sur un pont ethernet

En tant qu'outil d'interconnexion en mode paquet, le pont Ethernet peut être soumis à des phénomènes de congestion que nous allons tâcher d'illustrer ici.

Un premier type de congestion classique est dû à la capacité des mécanismes *Store and Forward* d'interconnecter des réseaux aux débits différents. Pour le mettre en évidence, nous allons relier par un pont deux réseaux aux débits différents.

La commande `ethtool` permet, par exemple de la façon suivante, de modifier le débit d'une interface ethernet :

```
# ethtool -s eth1 autoneg off speed 10
```

▷ **Exercice 5 : Utilisation de débits différents**

*Utilisez la commande `ethtool` pour configurer un lien à 10 Mbit/s. Utilisez ensuite un outil tel que `iperf` pour observer le comportement du pont.* ■