

# Le protocole MPLS

## Objectifs

Configurer, observer, analyser et comprendre le plan de données de MPLS

## 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 notre expérimentation

Le but de cette séance est de construire les LSP MPLS nécessaires pour acheminer le trafic entre les sites des différents clients de l'opérateur du réseau de la figure 1.



FIGURE 1 – Le réseau de cet exercice

Pour ne pas surcharger davantage le dessin, les octets de poids faible ne sont pas précisés. Ils ont les valeurs suivantes :

- pour r-1 à r-5 : le numéro du routeur (1 à 5);
- pour l'interface "extérieure" (ou publique) de ra-1, rb-1, ..., c'est 10;
- pour leur interface interne, c'est 1;
- pour chaque machine interne des clients (une seule par réseau) c'est 10.

Toute la configuration de base du réseau (adressage et un routage minimal) vous est fournie.

## 3 Configuration de base

Pour le bon fonctionnement de cette séance, un minimum de routage est nécessaire. Les routeurs et machines des clients doivent avoir un routage correctement configuré (c'est essentiellement une route par défaut).

Sur les routeurs de l'opérateur, et pour ce qui est de cette séance, seules les routes directes sont nécessaires. Inutile que chaque routeur soit capable de contacter ceux situés à plus d'un saut.

▷ **Exercice 1 : Pas de routage?!**

*Pourquoi le routage interne à l'opérateur est-il, ici, inutile?* ■

Le routage décrit ci-dessus est pré-configuré dans le réseau qui vous est fourni.  
Sur chaque routeur concerné par MPLS, il est nécessaire d'activer cette fonctionnalité.  
Il faut en particulier modifier le nombre de labels, avec une commande du type

```
# sysctl -w net.mpls.platform_labels=4096
```

Il est, de plus, nécessaire de stipuler les interfaces par lesquelles du trafic MPLS peut être reçu grâce à la commande (ici pour l'interface v1)

```
# sysctl -w net.mpls.conf.v1.input=1
```

Enfin, sur les *Egress Label Edge Routers*, il est nécessaire de veiller, sous Linux, à ce que le *Reverse Path Filtering* soit désactivé. Cela se fait par la commande suivante (ici pour l'interface v1)

```
# sysctl -w net.ipv4.conf.v1.rp_filter=0
```

Toutes ces opérations ont été réalisées dans le réseau qui vous est fourni, mais rien ne vous empêche de vérifier!

## 4 Construction des LSP

Nous allons maintenant construire les LSP. Il en faut deux (un dans chaque sens) pour chaque couple de sites de chaque client.

Pour suivre le bon fonctionnement des opérations, lancez un `ping` depuis un site vers un autre et laissez-le tourner. Vous observerez alors l'avancée des paquets au grès de vos configurations.

▷ **Exercice 2 : Définition des LSP**

*Définissez les LSP que vous allez construire, par exemple l'un depuis le site 1 du client A vers le site 2 du client A et un depuis le site A du client B vers le site 2 du client B.*

*Pour cela, représentez ces LSP sur la figure 1. Notez également sur chaque section le numéro de label utilisé.* ■

Nous allons maintenant construire progressivement ces LSP à la main. Pour vérifier votre bonne progression, utilisez la commande `tcpdump` dans les terminaux des différents routeurs ou/et utilisez `wireshark` avec les options proposées par le script.

### 4.1 Sur un *Ingress Label Edge Router*

La première étape consiste à définir la FEC, ce qui peut se faire sous linux avec les commandes suivantes

```
# ip rule add iif v0 table client1
# ip route add 192.168.60.0/24 encap mpls 100 via inet 192.168.20.1
  table client1
```

La première de ces commandes précise que le trafic entrant par l'interface v0 doit être routé grâce à la table de routage `client1`.

La seconde installe une route dans cette même table de routage. Cette route utilise une encapsulation MPLS.

▷ **Exercice 3 : Configuration des E-LEs**

*En adaptant les commandes précédentes à votre configuration, configurez vos Ingress Label Edge Routers.*

*Assurez-vous du bon résultat en utilisant la commande `ping` et `wireshark`.* ■

## 4.2 Sur un Label Switch Router

Sur chaque LSP, il faut maintenant configurer la commutation MPLS. Sous Linux, la commande suivante

```
# ip -f mpls route add 100 as 200 via inet 192.168.30.3 dev v0
```

permet de commuter le trafic MPLS entrant avec le label 100 en lui attribuant le label 200 et en l'envoyant *via* l'interface `v0` vers l'hôte dont l'adresse IP (ou `inet`) est donnée.

▷ **Exercice 4 : Configuration des LSRs**

*En adaptant la commande précédente à votre configuration, configurez vos Label Switch Routers.*

*Assurez-vous du bon résultat en utilisant la commande `ping` et `wireshark`.*

*À quoi sert l'adresse IP ? La voyez-vous dans le trafic observé ?* ■

## 4.3 Sur un Egress Label Edge Router

La dernière étape consiste à extraire le trafic du LSP et à l'injecter dans le réseau (ici dans la pile IP puisqu'il s'agit d'IPv4). Sous Linux, on peut réaliser cette opération grâce à la commande suivante

```
# ip -f mpls route add 200 via inet 192.168.10.23
```

Cette commande stipule que le trafic arrivant avec le label 200 est transmis via le destinataire désigné. Il est également possible de donner uniquement une interface (par exemple `lo`) de sorte que le trafic soit traité (routé ici) par le E-LE.

▷ **Exercice 5 : Configuration des E-LEs**

*En adaptant la commande précédente à votre configuration, configurez vos Egress Label Edge Routers.*

*Assurez-vous du bon résultat en utilisant la commande `ping` et `wireshark`.*

*À quoi sert l'adresse IP ? La voyez-vous dans le trafic observé ?* ■

