

# Le protocole MPLS

Avril 2014

## Objectifs

**IPv4 sur MPLS** : nous allons utiliser le protocole MPLS pour acheminer des trafics IPv4 au travers d'un réseau IP selon des chemins spécifiques.

## 1 IPv4 sur MPLS

### 1.1 Mise en place d'un réseau

Nous allons observer le fonctionnement de MPLS sur un réseau relativement simple. Il est donc nécessaire, dans un premier temps, de mettre en place ce réseau. Cette configuration se réalise de façon "classique" et ne fait en aucun cas intervenir de commandes spécifiques à MPLS.

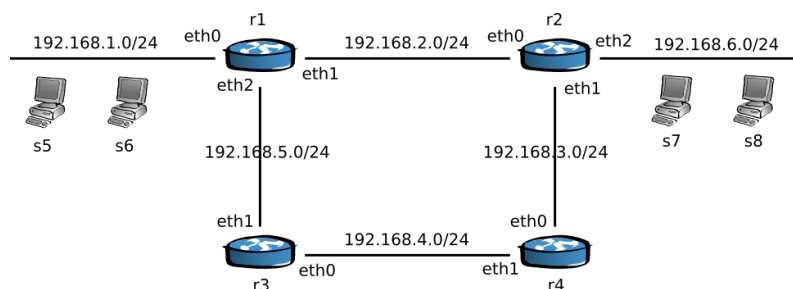


FIGURE 1 – Le réseau de base de notre expérience

#### ▷ Exercice 1 : Mise en place du réseau

Mettez en place le réseau de la figure 1 et vérifiez son bon fonctionnement. Une archive construisant directement un tel réseau dans l'environnement NetKit est disponible à l'adresse suivante [www.enseeiht.fr/~chapat/exercices/mpls-observation.bz2](http://www.enseeiht.fr/~chapat/exercices/mpls-observation.bz2) ■

### 1.2 Configuration d'un LER

Les routeurs situés à l'extrémité d'un LSP sont appelés des *Label Edge Routers* (LER). Ce sont eux qui ont la charge d'insérer le trafic IP dans le LSP et de l'en extraire.

#### ▷ Exercice 2 : Mise en place d'un premier LSP

Établir un premier chemin MPLS au travers du réseau. Les routeurs r1 et r2 seront les extrémités de ce LSP ; il permettra aux machines s5 et s6 d'envoyer des paquets vers les machines s7 et s8. ■

La mise en œuvre du LSP entre les LER utilise une encapsulation particulière qui peut être observée.

Attention cependant, dans l'environnement *NetKit* tel qu'il est fourni ici, l'outil *Wireshark* n'est pas installé. Il est nécessaire d'utiliser `tcpdump` (par exemple avec l'option `-w fichier`) pour envoyer le résultat de l'observation dans un fichier (on utilisera par exemple le répertoire `/hosthome`). On pourra ensuite lire le contenu de ce fichier avec un outil classique sur la machine hôte.

▷ **Exercice 3 : Observation de MPLS**

*Utiliser `wireshark` ou `tcpdump` pour observer le trafic entre les deux routeurs.* ■

Du point de vue de IP, le service fourni par MPLS est celui d'un lien point à point. La MTU du lien ainsi établi est affectée par l'encapsulation. Le mécanisme de PMTUD s'en trouve également perturbé.

▷ **Exercice 4 : Observation de la MTU**

*Observer les conséquences de l'encapsulation sur des paquets de "grande taille". On utilisera par exemple la commande `ping`* ■

### 1.3 Configuration d'un LSR

La configuration d'un routeur situé au cœur d'un LSP (c'est-à-dire d'un LSR ou *Label Switch Router*) est différente de celle d'un routeur d'extrémité.

▷ **Exercice 5 : Configuration d'un LSR**

*Configurer les routeurs `r3` et `r4` en tant que LSR. On construira en l'occurrence un LSP menant du réseau de `s7` et `s8` à celui de `s5` et `s6` au travers de `r2`, `r4`, `r3` et `r1`.* ■

Du point de vue de IP, le service fourni par MPLS est toujours celui d'un lien point à point ; le chemin qu'il emprunte est donc complètement masqué.

▷ **Exercice 6 : Observation du chemin**

*Que donne la commande `traceroute` utilisée entre des machines des réseaux `N1` et `N2` ?* ■

## 2 Utilisation du filtrage

Les manipulations réalisées précédemment permettent donc d'acheminer du trafic au travers d'un LSP. L'utilisation conjointe des fonctionnalités de routage et de filtrage de Linux permet une utilisation bien plus souple des LSP mis en œuvre.

La commande `iptables`, déjà utilisée dans d'autres séances, peut être mise à contribution ici pour sélectionner le trafic qui doit être acheminé ou non dans un LSP.

▷ **Exercice 7 : Classification du trafic**

*Mettez en place un routage simple permettant aux machines d'extrémité de communiquer entre elles sans passer par le LSP établi. On utilisera un chemin différent de celui utilisé par le LSP.*

*Réalisez alors un filtrage du trafic de sorte qu'une partie seulement soit acheminée dans le LSP. On distinguera le trafic en fonction de l'application par exemple.* ■

## A Le fonctionnement de MPLS sous Linux

La mise en œuvre de MPLS n'est pas très complexe, mais elle fait intervenir diverses composantes système qui peuvent la faire paraître plus délicate qu'elle ne l'est réellement.

Lorsqu'un routeur IP reçoit un paquet, il doit utiliser sa table de routage pour déterminer par quelle interface le transmettre.

Lorsqu'un routeur MPLS reçoit un paquet MPLS il doit consulter sa table ILM (*Incoming label mapping*) pour déterminer ce qu'il faut faire du paquet.

Un LER MPLS est donc voué à mener ces deux activités puisqu'un paquet IP peut, suite à son routage, être acheminé au travers d'un LSP MPLS et vice-versa.

### A.1 La commande `mpls`

Sous Linux, la mise en place du protocole MPLS est assurée par la commande `mpls`. C'est cette commande qui permet la configuration spécifique à MPLS, mais elle ne saurait y suffire, pour les raisons évoquées précédemment.

#### A.1.1 Création d'une NHLFE

La notion de NHLFE (pour *Next Hop Label Forwarding Entry*) est un élément important dans la commutation MPLS. C'est en effet une NHLFE qui définit le comportement à adopter lors de la réception d'un paquet.

La création d'une nouvelle NHLFE permettant d'ajouter un label à un paquet entrant puis de l'acheminer vers un prochain routeur est réalisée par exemple de la façon suivante

```
# mpls nhlfe add key 0 instructions push gen 1000 nexthop eth0 ipv4 192.168.31.2
```

Analysons les paramètres de cette commande

`mpls nhlfe add` stipule que l'on souhaite ajouter une nouvelle entrée dans la table ;

`key n` est l'identifiant de l'entrée que l'on souhaite manipuler (ici, comme l'entrée n'existe pas encore on utilise la valeur nulle, la valeur attribuée par le système sera donnée en retour) ;

`instructions` introduit la liste des instructions associées à cette entrée ;

`push gen 1000` stipule que l'on empile le label 1000 au travers d'une encapsulation générique ;

`nexthop eth0 ipv4 192.168.31.2` précise que le paquet doit alors être acheminé par l'interface `eth0` au travers du prochain routeur IPv4 dont l'adresse est précisée.

#### A.1.2 Observation des NHLFE

```
# mpls nhlfe
```

#### A.1.3 Modification d'une NHLFE

Il est possible de modifier une NHLFE de la façon suivante

```
# mpls nhlfe change key 0x5 ...
```

La clef est évidemment l'identifiant de l'entrée à modifier et les autres paramètres acceptés sont les mêmes que pour la création.

#### A.1.4 Suppression d'une NHLFE

```
# mpls nhlfe del key 0x5
```

#### A.1.5 Routage au travers d'une NHLFE

L'ajout d'une route au travers d'un LSP est réalisé de la façon suivante

```
# ip route add 192.168.2.0/24 via 192.168.1.2 mpls 0x2
```

La différence par rapport à un ajout de route "classique" est l'ajout de la clef de la NHLFE à utiliser (introduite par le mot-clef `mpls`).

### A.2 Définition d'un espace de label entrant

Si une interface doit être utilisée pour recevoir des paquets MPLS, un espace de nommage doit lui être attribué de la façon suivante

```
# mpls labelspace set dev eth0 labelspace 0
```

### A.3 Mise en place de la commutation MPLS

La configuration de la commutation MPLS effective est réalisée par la sous-commande `xc` de la commande `mpls` :

```
# mpls xc add ilm_label gen 1000 ilm_labelspace 0 nhlfe_key 0x3
```

### A.4 L'utilisation du routage et du filtrage

Les commandes évoquées ci-dessus permettent donc de construire des chemins à la main et d'y acheminer le trafic IP. Les commandes suivantes permettent de réaliser des traitements plus fins.

On peut également utiliser la commande `IP` pour router le trafic en fonction, par exemple, de l'adresse source. Une telle fonctionnalité est mise en œuvre au travers de la définition de nouvelles tables de routage.

#### A.4.1 Utilisation de plusieurs tables de routage

Chaque table de routage est identifiée par un numéro et manipulée avec la commande `ip` de façon classique en ajoutant `table <n>`, par exemple

```
# ip route add default via 192.168.10.1 table 10
```

#### A.4.2 Sélection de la table en fonction de l'adresse source

C'est encore par la commande `IP`, mais à l'aide de la sous-commande `rule`, que la table de routage utilisée peut être choisie en fonction de l'adresse source :

```
# ip rule add from 192.168.20.0/24 table 10
```

D'autres critères de sélection existent, l'un de ces critères est le résultat d'un marquage associé au paquet :

```
# ip rule add fwmark 10 table 10
```

Ce marquage peut être réalisé par le système *Netfilter*, ce qui ouvre un éventail de possibilités extrêmement vaste.