

# Le routage IP multi-chemin

## Objectifs

Comprendre et observer le fonctionnement de l'ECMP

## 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 mettre en œuvre de l'ECMP dans le réseau de l'opérateur du réseau de la figure 1.

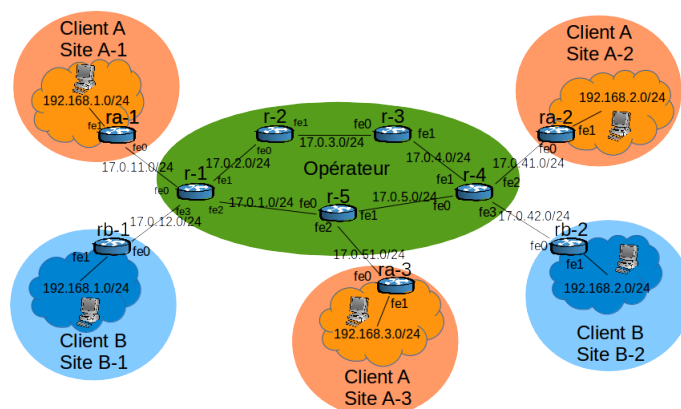


FIGURE 1 – Le réseau de cet exercice

Pour ne pas surcharger davantage le dessin, les octets de poids faible de l'adressage 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.

La configuration de l'adressage est faite dans le réseau fourni, il vous reste donc à mettre en place le routage.

## 3 Mise en place du routage

Afin de mieux observer ce que nous allons mettre en place, nous allons réaliser un routage manuel. Afin de simplifier, nous focaliserons sur le routage entre les réseaux 17.0.11.0/24 et 17.0.41.0/24. Les routeurs des sites clients (*ra-1* et *ra-2* par exemple) utiliseront des routes

par défaut.

▷ **Exercice 1 : Routage simple**

*Mettez en place un routage simple permettant à ra-1 et ra-2 de communiquer.  
On mettra en place (successivement) deux routes que l'on testera.* ■

## 4 Mise en place du routage multi-chemin

Maintenant que nous avons testé et validé deux chemins différents permettant de communiquer entre les deux réseaux, nous allons utiliser l'ECMP pour faire du partage de charge entre ces deux chemins.

Pour cela, sous Linux, la commande `ip route add` accepte plusieurs clauses "nexthop", par exemple

```
# ip route add 192.168.19.0/24 nexthop via 147.127.18.98 nexthop via 147.127.18.99
```

▷ **Exercice 2 : Routage multi-chemin**

*Mettez en place un routage multi-chemin permettant à ra-1 et ra-2 de communiquer.* ■

L'observation des conséquences d'un tel routage peut être un peu plus délicate que pour du routage simple. On utilisera par exemple la commande `traceroute` ou encore l'observation du trafic sur les routeurs intermédiaires avec la commande `tcpdump`.

On pourra utiliser des adresses cibles variées (voire des numéros de port différents) afin de maximiser les chances d'utiliser plusieurs chemins. Le fait que les adresses visées ne soient pas attribuées n'empêche pas `traceroute` de chercher un chemin.

▷ **Exercice 3 : Observation des routes**

*Observez que le trafic utilise effectivement les deux chemins que vous avez configurés.* ■

## 5 Paramétrage de l'ECMP

Dans Linux, l'algorithme utilisé se fonde sur certaines informations contenues dans les paquets et utilisées dans une fonction de hachage. Il effectue ensuite un *round robin* entre les différents chemins disponibles.

▷ **Exercice 4 : Pourquoi ces choix ?**

*À votre avis, quelles sont les raisons de ces deux choix d'implantation ?* ■

Les informations à utiliser sont décrites par la variable système `fib_multipath_hash_policy` (qui peut être consultée et modifiée par la commande `sysctl` par exemple).

La valeur 0 stipule qu'on utilise uniquement des informations de la couche 3 et la valeur 1 que l'on utilise également des informations de la couche 4.

▷ **Exercice 5 : Quelles différences ?**

*À votre avis, quelles sont les avantages et inconvénients de ces deux stratégies ?* ■

Un poids peut également être attribué à chaque chemin configuré avec la commande IP :

```
# ip route add 192.168.19.0/24 nexthop via 147.127.18.98 weight 2  
nexthop via 147.127.18.99 weight 3
```

▷ **Exercice 6 : Observation et modification de la configuration**

*Modifiez les différents paramètres de votre routage et observez les conséquences.* ■