

Mise en place d'un routage intradomaine

1 Contexte du laboratoire

Le but de ce laboratoire est de vous permettre de vous familiariser avec la configuration de routeurs en amont du projet Mini-Internet. Vous allez devoir configurer une petite topologie de réseau internet avec des routeurs et des hôtes. Au cours de ce laboratoire, vous verrez comment configurer des adresses IP statiques, configurer des routes statiques et vous familiariser avec un protocole de routage dynamique (OSPF) et sa configuration à travers la suite FRRouting.

Description de la topologie

La topologie utilisée dans ce laboratoire est représentée dans la figure 1. Elle est composée de 4 routeurs et 2 hôtes. Les interfaces sur lesquelles les routeurs sont connectées sont indiquées dans la figure. Pour cet exercice vous devrez vous-même configurer les différentes adresses IP des routeurs. Le plan d'adressage que vous devrez respecter est représenté dans la figure. Dans les adresses, il faut remplacer X par les **deux premiers chiffres** de votre NOMA.

2 Kathara

Pour ce laboratoire vous allez utiliser le logiciel Kathara. Il permet de créer des réseaux virtuels grâce à Docker.

Installation

Vous devez tout d'abord installer Docker. Pour cela, rendez-vous à l'adresse suivante et suivez les instructions correspondantes à votre système d'exploitation : <https://docs.docker.com/engine/install/>.

Une fois Docker installé, rendez-vous à l'adresse suivante et suivez les instructions correspondantes à votre système d'exploitation pour installer Kathara : <https://www.kathara.org/download.html>.

Composition d'un laboratoire

Dans cette section, nous allons voir comment est composé un laboratoire Kathara. Allez dans le dossier `lab_routage_intradomaine`. Vous y trouverez un fichier `lab.conf` qui contient la description de la topologie. Il décrit notamment à quoi sont connectées les différentes interfaces des machines. Par exemple : `r1[2]="A"` signifie que l'interface eth2 du routeur r1 est connectée au domaine de collision "A", et donc à toute autre interface connectée à ce domaine.

Chaque routeur/hôte dispose d'un fichier `".startup"`. N'importe quelle commande linux entrée dans ce fichier sera exécutée automatiquement au démarrage du laboratoire.

Enfin chaque routeur dispose d'un dossier contenant différents fichiers de configuration pour FRRouting (nous reviendrons sur ce dernier plus tard dans ce laboratoire). Par exemple pour r1, le dossier est `r1/etc/frr`.

Lancement d'un laboratoire

Tout d'abord veillez à ce que docker soit lancé.

Depuis le répertoire où se situe le fichier `lab.conf`, lancez la commande suivante :

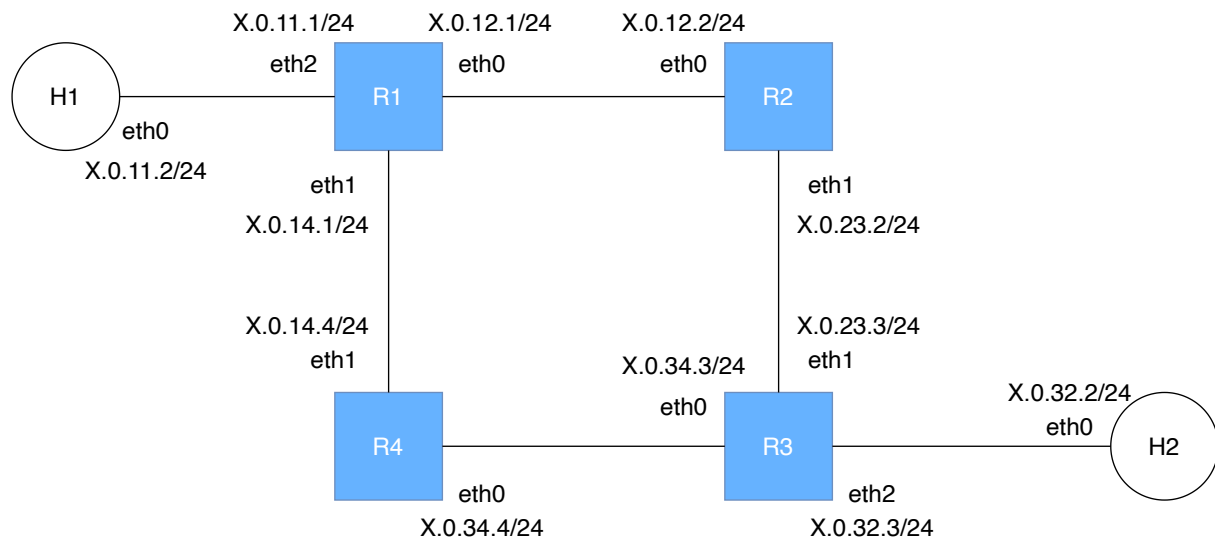


FIGURE 1 – Topologie du premier laboratoire

```
$ kathara lstart
```

et attendez que le laboratoire démarre. Si vous obtenez le message d'erreur suivant :

```
(DockerException) Error while fetching server API version:
('Connection aborted.', FileNotFoundError(2, 'No such file or directory'))
```

il est probable que docker ne soit pas lancé.

Normalement, une fenêtre terminal devrait s'ouvrir pour chaque routeur/hôte dans la topologie. Si ce n'est pas le cas, vous pouvez lister les différents *devices* avec la commande suivante :

```
$ kathara list
```

Un tableau s'affichera, listant les différents *devices* du scénario avec différentes informations, dont leur nom.

Ensuite, connectez-vous à un *device* avec la commande suivante :

```
$ kathara connect <DEVICE>
```

avec *DEVICE* le nom du *device*, trouvé dans la colonne *name* du tableau décrit. Le terminal sera alors celui du routeur en question et vous pourrez alors lui envoyer des commandes.

Ouvrez une fenêtre terminal pour chaque *device*.

Quand vous aurez terminé le laboratoire, stoppez-le avec la commande suivante :

```
$ kathara lclean
```

3 Configuration du réseau

Configuration des adresses IP et routage statique

La première étape est d'attribuer une adresse IP aux interfaces des routeurs et des hôtes selon le plan d'adressage donné. Ceci se fait avec la commande `ip`¹. Pour attribuer l'adresse X.0.11.1/24 à l'interface eth2 de r1, vous pouvez utiliser la commande suivante :

1. ip man page : <https://man7.org/linux/man-pages/man8/ip.8.html>

```
root@r1:/# ip address add X.0.11.1/24 dev eth2
```

Faites de même pour les autres interfaces des routeurs et des hôtes. Sur chaque hôte ou routeur, vérifiez que les adresses IP ont bien été attribuées aux bonnes interfaces avec la commande suivante :

```
root@r1:/# ip address show
```

Une fois que toutes les adresses IP sont configurées, depuis h1 essayez de ping l'interface X.0.11.1 de r1. Est-ce que cela fonctionne ? Et si on essaye sur une autre de ses interfaces (par exemple X.0.12.1) ?

À ce stade, h1 ne sait qu'atteindre les adresses du sous-réseau X.0.11.0/24. Pour résoudre ça, il faut indiquer à h1 d'envoyer tous les paquets à destination d'adresses qui lui sont inconnues à r1. Il faut donc ajouter une route par défaut à h1 pointant vers X.0.11.1.

```
root@h1:/# ip route add default via X.0.11.1
```

Vous devriez maintenant être capable de pinger l'interface X.0.12.1 de r1 depuis h1. Affichez la table de routage de h1 avec la commande suivante :

```
root@h1:/# ip route show
```

L'output devrait ressembler à ceci :

```
default via X.0.11.1 dev eth0  
X.0.11.0/24 dev eth0 proto kernel scope link src X.0.11.2
```

Faites la même chose pour h2 avec l'interface X.0.32.3 de r3. Si vous pingez l'interface X.0.12.2 de r2 depuis h1, cela fonctionne-t-il ? Pourquoi ? (Indice : essayez de pinger dans l'autre sens).

OSPF

Pour les routeurs, vous n'allez pas configurer leur table de routage manuellement. Les tables de routages seront configurées de manière dynamique grâce au protocole OSPF. OSPF est un protocole de routage interne (à donc utiliser au sein d'un même AS), de type *link-state*. Il permet de calculer les chemins les plus courts (selon les poids associés aux liens) entre les routeurs d'un même AS.

Pour cela, vous allez utiliser la suite FRRouting pour configurer les routeurs. FRRouting est une suite de protocoles de routages open-source². Elle implémente divers protocoles de routage très utilisés, dont OSPF³.

Interaction avec FRRouting

Au démarrage de chaque routeur, FRRouting est lancé automatiquement, grâce à la commande "systemctl start fr" présente dans le fichier ".startup" de chaque routeur. Pour interagir avec le daemon FRRouting, vous devez utiliser le terminal *vttysh* en utilisant la commande :

```
root@r1:/# vtysh
```

Votre terminal devrait vous afficher ceci.

```
Hello , this is FRRouting (version 9.0.1).  
Copyright 1996–2005 Kunihiro Ishiguro , et al.
```

```
r1#
```

2. Plus d'info sur FRRouting ici : <https://frrouting.org/>

3. Documentation OSPF : <https://docs.frrouting.org/en/latest/ospfd.html>

Notez que **root@r1 :/#** est devenu **r1#**. Vous êtes maintenant dans le terminal *vttysh* du routeur r1, vous permettant donc de configurer FRRouting. Ce terminal fonctionne un peu différemment des autres terminaux que vous avez utilisé jusqu'à présent. Pour afficher les commandes disponibles, vous pouvez taper **?**. De même, après avoir tapé une commande, vous pouvez entrer **?** pour afficher les options disponibles.

La commande **show** permet d'afficher de nombreuses informations sur la configuration du routeur. Par exemple,

```
r1# show interface eth0
```

permet d'afficher les informations de l'interface eth0 du routeur r1, dont l'adresse lui étant associée. Ou encore,

```
r1# show ip route
```

permet d'afficher la table de routage du routeur r1.

Configuration d'OSPF

Sur chaque routeur, démarrez le service OSPF avec les commandes suivantes :

```
r1# configure terminal
r1(config)# router ospf
r1(config-router)# exit
```

Notez qu'après la première commande, **r1#** est devenu **r1(config)#**. Cela signifie que vous êtes dans le mode de configuration. Dans *vttysh*, le préfixe du terminal indique dans quel mode vous vous trouvez. Les commandes disponibles sont différentes en fonction du mode. Pour la suite du laboratoire soyez vigilant sur le préfixe du terminal dans les instructions données. Si une commande que vous entrez n'est pas reconnue, vérifiez que vous êtes bien dans le bon mode de configuration. Pour quitter le mode de configuration, vous pouvez taper **exit**.

Sur r1 activez OSPF sur son interface eth0 avec la commande suivante :

```
r1(config)# interface eth0
r1(config-if)# ip ospf area 0
r1(config-if)# exit
```

L'argument **area 0** est lié à un concept d'OSPF qui n'est pas au programme du cours. Néanmoins, il est obligatoire de le spécifier pour que la commande fonctionne. Pour toute vos configurations OSPF de ce laboratoire et du projet, vous pouvez utiliser **area 0**.

Dans le cas où vous avez fait une erreur de configuration, vous pouvez annuler une commande en la réentrant avec le préfixe **no** devant. Si par exemple, vous avez activé OSPF sur l'interface eth0 de r1 par erreur, vous pouvez le désactiver avec la commande suivante :

```
r1(config)# interface eth0
r1(config-if)# ip ospf area 0
r1(config-if)# no ip ospf area 0
```

Faites de même avec l'interface eth2 de r1 et l'interface eth0 de r2. Puis attendez un peu que OSPF converge et affichez la table de routage OSPF de r1 et r2 avec la commande suivante :

```
r1# show ip ospf route
```

Normalement les tables devraient contenir les deux mêmes entrées. Une pour le préfixe X.0.12.0/24 et une pour le préfixe X.0.11.0/24. Depuis h1 vous devriez être capables de ping l'interface eth0

de r2 mais pas son interface eth1.

Maintenant activez OSPF sur l'interface eth1 de r2 et r3, ainsi que sur l'interface eth2 de r3. Normalement vous devriez pouvoir ping h2 depuis h1 et inversement. Si ça ne marche pas directement, attendez un peu que OSPF converge. Faites un traceroute pour voir le chemin emprunté par les paquets.

Avec la configuration actuelle, le protocole OSPF fait régulièrement envoyer à R1 des paquets *hello* à destination de h1. Or, ces messages sont inutiles pour h1 car il n'utilise pas OSPF. Il faut donc configurer l'interface eth2 de r1 pour qu'elle soit *passive* pour OSPF, et donc que r1 n'envoie plus de paquets liés à OSPF sur cette interface.

```
r1(config)# interface eth2
r1(config-if)# ip ospf passive
r1(config-if)# exit
```

Faites de même pour l'interface eth2 de r3.

Maintenant, activez OSPF sur l'interface eth1 de r1 et r4 et sur l'interface eth0 de r4 et r3. Par défaut, un poids égal à 10 est attribué à chaque interface. Pour les interfaces de r4, attribuez un poids de 9 à l'interface eth0 :

```
r4(config)# interface eth0
r4(config-if)# ip ospf cost 9
r4(config-if)# exit
```

Attribuez aussi un poids de 11 à l'interface eth1 de r4.

Faites un traceroute de h1 vers h2 et de h2 vers h1. Qu'observez-vous ?

Maintenant jouez avec les poids de manière à ce que les paquets passent par r2 et r3 pour aller de r1 à r4.

Une fois cela terminé, afficher les tables de routage de chaque routeur et faites valider par le tuteur. Vous pouvez ensuite éteindre le laboratoire avec la commande suivante :

```
$ kathara lclean
```