

Travaux Pratiques

Initiation IPv6

Configuration, Auto-configuration, Routage

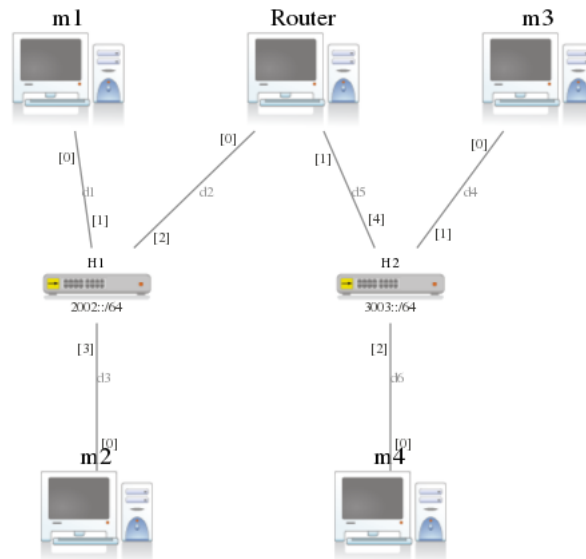
Copyright (C) 2016 Jean-Vincent Loddo
Licence Creative Commons Paternité - Partage à l'Identique 3.0 non transposé.

Séance de TP entièrement effectuée avec le logiciel Marionnet. Durée estimée : 1h30.

Prérequis. Initiation à IPv6 (format, adresses, quelques éléments de ICMPv6, commande `ip`).

1 Câblage et configuration des réseaux lien-local (locaux, privés)

On utilise 5 machines, m_1 , m_2 , *Router*, m_3 et m_4 , dont une en particulier, *Router*, équipée de 2 interfaces réseau *eth0* et *eth1*. Utilisez la distribution Debian wheezy sur toutes les machines. Les deux réseaux locaux $LAN_1 = \{m_1, m_2, Router[0]\}$ et $LAN_2 = \{Router[1], m_3, m_4\}$ seront réalisés par des hubs (de façon à pouvoir observer facilement le trafic). Déclenchez l'auto-configuration des adresses lien-local de façon à pouvoir communiquer entre les machines des deux réseaux (sans routage évidemment, qui n'a pas de sens pour les adresses lien-local). Testez les connexions avec `ping6` et `ssh` tout en observant le trafic généré avec des instances de `wireshark` tournant sur m_2 et m_4 .



Rappel syntaxe commande `ip` sous Linux (par l'exemple)

Activer/désactiver une interface

```
ip link set dev eth0 up      # équivalent à "ifconfig eth0 up"
ip link set dev eth0 down    # juste "down" à la place de "up"
```

Afficher l'état d'une interface

```
ip -6 addr show dev eth0
```

Ajouter/éliminer une adresse IPv6

```
ip -6 addr add 2001:0db8:0:f101::1/64 dev eth0
ip -6 addr del 2001:0db8:0:f101::1/64 dev eth0      # juste "del" à la place de "add"
```

Afficher le voisinage IPv6 (correspondant de ARP)

```
ip -6 neigh show
```

Afficher les routes IPv6

```
ip -6 route show
```

Ajouter/éliminer une route IPv6

```
ip -6 route add default via 2001:0db8:0:f101::1      # NETWORK/LEN (CIDR) au lieu de "default"
ip -6 route del default via 2001:0db8:0:f101::1      # juste "del" à la place de "add"
```

Activer le routage IPv6 du noyau Linux

```
echo 1 >/proc/sys/net/ipv6/conf/all/forwarding
```

2 Configuration manuelle des adresses IPv6 globales et routage

Configurer **manuellement** les interfaces de toutes les machines des deux réseaux $2002::/64$ et $3003::/64$. Par simplicité, la machine m_i aura le dernier groupe égal à i . Configurer aussi le routage statique. Testez à nouveau les connexions avec `ping6` et `ssh`, dans chacun des réseaux et entre les deux (routage), tout en observant le trafic généré avec des instances de `wireshark` tournant sur m_2 et m_4 .

3 Auto-configuration des adresses IPv6 globales et routage

Désactivez toutes les interfaces de toutes les machines pour recommencer l'exercice. Au lieu de configurer manuellement le réseau (ce qui était plutôt la norme en IPv4), nous allons profiter de l'**auto-configuration** (appelée "sans état", ce qui est plutôt la norme en IPv6). Configurez *Router* de façon qu'il se comporte en routeur IPv6, c'est-à-dire de façon qu'il envoie sur les réseaux $2002::/64$ et $3003::/64$ des messages ICMPv6 de type RA (Router Advertisement) et qu'il réponde aux messages ICMPv6 de type RS (Router Solicitation). Les routeurs sont normalement pré-équipés pour cela. Dans notre cas, s'agissant d'une machine GNU/Linux, il faudra configurer le service appelé `radvd` (Router Advertisement Demon). Voici un exemple de fichier de configuration `/etc/radvd.conf` que vous devrez adapter :

```
interface eth0 {
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    prefix 3ffe:ffff:0100:f101::/64 {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

Une fois le service en marche, réactivez tout simplement les interfaces de toutes les autres machines. La configuration devrait donc se passer de façon automatique et transparente, d'une manière assez similaire à DHCP en IPv4 (mais aucun état, ou "bail", est sauvegardé). Une fois les interfaces réactivées, testez une fois de plus les connexions avec `ping6` et `ssh`, dans chacun des réseaux et entre les deux (routage), tout en observant le trafic généré avec des instances de `wireshark` tournant sur m_2 et m_4 .

4 Annexe

```
$ nmap -6 -sT ::1      # détecte les ports TCP/IPv6 ouverts sur la machine
```