

Travaux Pratiques

Initiation IPv6

Auto-configuration, Routage et Filtrage

Copyright (C) 2019 Jean-Vincent Loddo

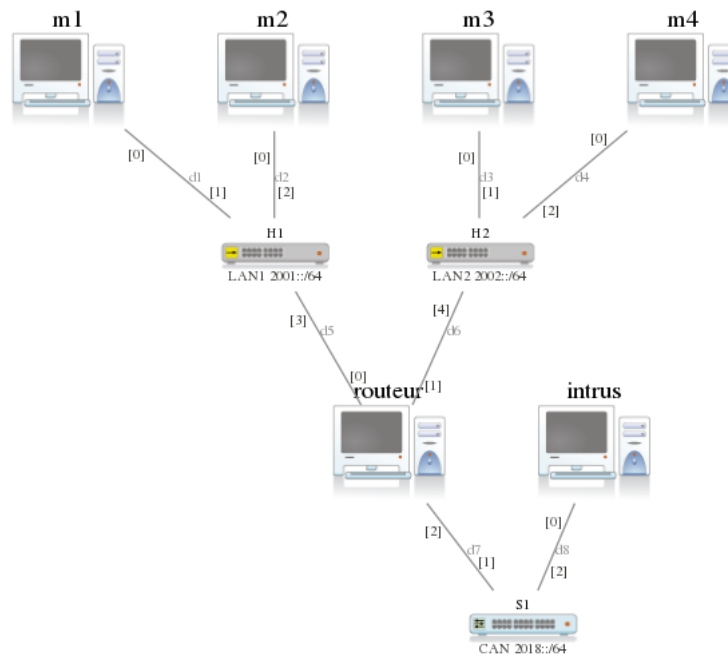
Licence Creative Commons Paternité - Partage à l'Identique 3.0 non transposé.

Séance de TP entièrement effectuée avec le logiciel Marionnet. Durée estimée : 1h30.- 2h

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

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

On utilise 6 machines, m_1 , m_2 , *Router*, m_3 , m_4 et *intrus*, 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
ip link set dev eth0 down
```

```
# équivalent à "ifconfig eth0 up"
# 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 dev eth0
ip -6 addr del 2001:0db8:0:f101::1 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 Auto-configuration des adresses IPv6 globales et routage

Configurez *Router* de façon qu'il se comporte en routeur IPv6, c'est-à-dire de façon qu'il envoie sur les réseaux 2001::/64 et 2002::/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 .

3 Filtrage IPv6

Activez le service de connexion à distance sécurisées (`ssh`) sur toutes les machines ordinaires m_1 , m_2 , m_3 , m_4 . Toute machine faisant partie de LAN_1 ou LAN_2 pourra se connecter à distance sur toute autre machine de ces deux réseaux locaux. Cependant, *intrus* pourra se connecter seulement sur m_1 , en spécifiant son adresse globale, pas sur les autres. En utilisant `ip6tables`, prévoir donc sur *routeur* des règles de filtrage interdisant tout autre accès aux machines des réseaux locaux de la part de l'extérieur (c'est-à-dire, dans notre cas, *intrus*).