APPRENONS IPV6 SANS PEINE PAR LA PRATIQUE : INTRODUCTION

par Alix Mascret
[membre fondateur de l'Éof et de formation-libre.fr]
Jean-Philippe Gaulier
[directeur de l'Éof et membre fondateur de formation-libre.fr]

IPv6, tout le monde parle d'IPv6. « C'est le futur », dit-on. « Oui mais nous sommes dans le présent », répondront certains. Certes, IPv6 met bien du temps à s'étendre et trop peu de fournisseurs de services et d'accès à Internet le proposent pour l'instant. Il n'en reste pas moins que la migration d'IPv4 à IPv6 est quelque chose que nous vivrons de gré ou de force dans les années à venir. Nous vous proposons donc de vous former à la fois à son architecture, mais aussi et surtout à son utilisation. Une formation progressive reposant sur la meilleure méthode qui puisse exister : la pratique !

1 Présentation générale

1.1 Petit rappel des faits entre amis

Actuellement, le protocole de traitement de la couche réseau est IPv4, pour *Internet Protocol* version 4. Ce protocole a été développé par Vinton Cerf suite à une demande du DOD, pour ARPAnet. Plus de quarante ans ont passé et un certain nombre de limitations sont apparues. Tout d'abord, le nombre d'adresses IP disponibles est limité à 4 294 967 296, soit 2^32. Cette restriction, lors de la création du protocole, semblait ne pouvoir jamais être atteinte. Cependant, l'évolution de l'informatique, l'explosion de l'usage Internet et la naissance du tout embarqué créent une sérieuse remise en question de ce *pool* qui pouvait être considéré comme acquis. Ainsi, en 1992, après l'ouverture commerciale d'Internet, il fallut, une année plus tard, déclencher un plan d'urgence, puisqu'il ne restait plus aucune classe B disponible. Ces mesures se concrétisèrent en deux points :

- La création de la notation CIDR et sa mise en place. Le résultat fut une diminution du gaspillage de l'espace d'adressage, ainsi que la possibilité d'agrégation, ce qui entraîne une réduction de la taille des tables de routage.
- La création et la mise en place d'un plan d'adressage privé **[rfc 1918]** et du NAT (traduction d'adresse réseau).

Ces mesures palliatives ne sont que temporaires. De fait, les mesures techniques induisent des contraintes et de nouveaux problèmes. Les protocoles dynamiques tels que le FTP doivent être traités indépendamment. Une couche de sécurité supplémentaire est obligatoire pour assurer l'intégrité et la confidentialité. L'humanité continue de croître, les applications technologiques aussi. Il est donc essentiel, dès ce moment-là, de travailler sur le successeur d'IPv4, en tâchant de résoudre les manques inhérents à ce dernier. C'est en 1995 que l'IETF fournira une première mouture d'IPv6, avant de finaliser une seconde version fin 1998.

1.2 Soyons pragmatiques

Seize années se sont écoulées depuis la création d'IPv6. Personne ne croit en son arrivée, pas plus que son intérêt. Seulement, un petit village d'irréductibles combat encore et toujours l'incrédulité. Composés d'une communauté internationale, ils se reconnaissent par quelques signes rituels : ils parlent d'une tortue qui danse **[kame]**, d'une terre qui tourne

[ipv6_forum] ou encore d'un drapeau qui flotte **[ipv6_uk]**, mais surtout, de *Star Wars* en ASCII en couleurs ! Le site du RIPE annonce une fin de distribution d'IPv4 au mois de février pour les ressources directes de l'IANA, et fin novembre pour ses représentants régionaux. Nous avons l'opportunité de vivre ce que peu d'entre nous ont pu faire lors de l'arrivée d'IPv4 : vivre le passage d'un monde à l'autre et de faire partie de l'aventure dès le début (ou simplement partager une curiosité technique) ! Nous vous proposons de nous accompagner à travers une série d'articles qui vous permettront de maîtriser les bases de ce protocole et son arrivée sur le réseau.

2 Présentation de la maquette

Pour vous aider à intégrer les principes d'IPv6, nous vous proposons de concevoir une maquette qui, à la fin, devrait ressembler à ce que vous avez sur la figure 1.

Et nous allons nous appuyer sur l'environnement de conception de réseau Netkit **[netkit]**. Rassurez-vous, elle n'est pas aussi compliquée qu'il y paraît. Voici quelques clés pour lire le schéma. L'hôte correspond à votre machine physique, c'est elle qui hébergera les différentes machines virtuelles. Elle sera reliée à Internet par votre connexion normale et nous verrons comment le relier à un réseau IPv6, si votre fournisseur d'accès ne vous offre pas encore cette option (nativement).

Les nœuds nommés CXXX sont des clients, ceux qui sont nommés RX sont des routeurs. Le nœud S fera office de serveur de nom et d'applications. DH6 héberge un serveur DHCPv6. Le routeur R, avec six interfaces, est un peu particulier. Il est relié à l'hôte via une interface TUN/TAP afin de pouvoir accéder à Internet dans le cas où on souhaite, par exemple, télécharger et installer un paquet.

Ensuite, on distingue le réseau IPv6 (constitué de trois segments ip60.0, ip61.0 et ip61.1, segments IPv6 uniquement) et un réseau IPv4 (constitué de trois segments ip40.0, ip41.0 et ip41.1, segments IPv4 exclusivement). On voit que les segments ip41.0 et ip41.1 sont reliés par le segment ip60.0 et que les segments ip61.0 et ip61.1 sont reliés par le segment ip40.0. Cette partie de la maquette un peu particulière nous permettra d'aborder tout ce qui touche aux tunnels et à l'interconnexion des réseaux.

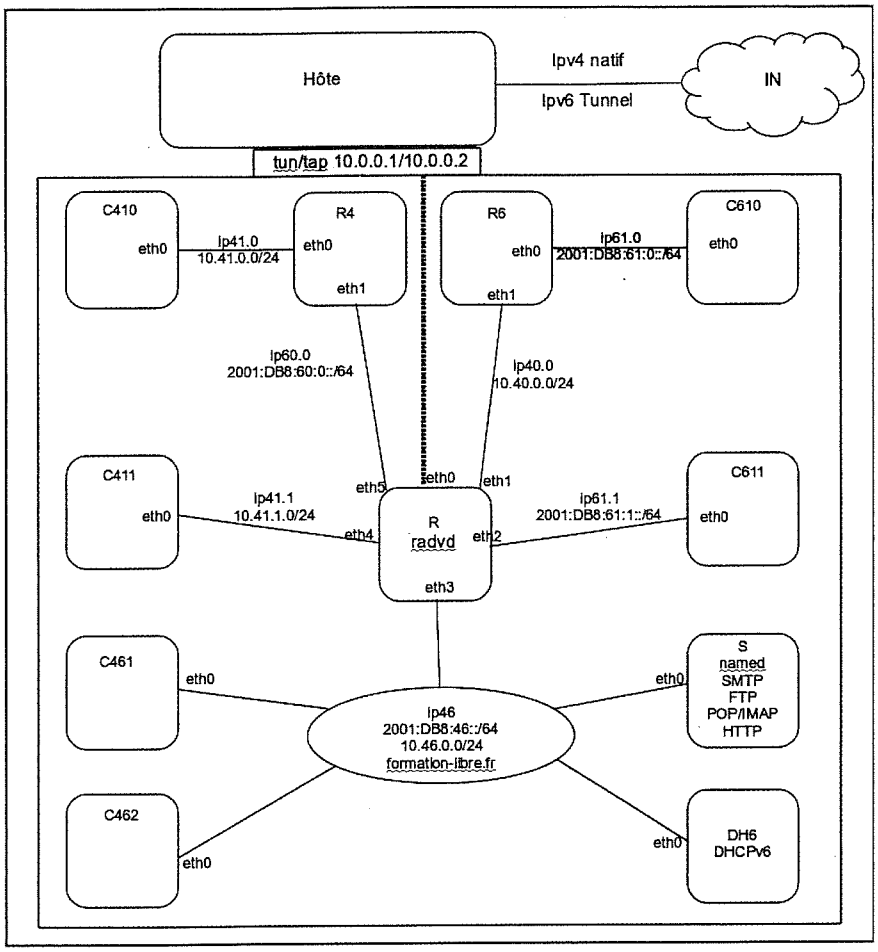

Figure 1

On distingue également un segment ip46 qui, lui, sera double pile (*dual stack*), en ce sens qu'il supportera les deux piles de protocoles IPv4 et IPv6. Les adresses IP des interfaces ne sont pas indiquées pour ne pas surcharger le schéma, mais on partira du principe général que les adresses les plus hautes sont données aux routeurs. Les noms des segments seront utilisés pour définir les domaines de collision. Un domaine de collision, au sens netkit du terme, est un concentrateur (*hub*) virtuel.

IPv6 : première partie

Cette première partie va nous donner l'occasion :

- d'aborder rapidement l'adressage IPv6 pour cerner ce qui, techniquement, va changer au premier abord ;
- de relier votre machine, chez vous ou au bureau, aux réseaux IPv6 via un opérateur qu'on appelle tunnel *broker* ;
- de mettre en place les premières briques de la maquette au travers d'un lab Netkit, lab qui nous permettra de manipuler les premières commandes système et réseau applicables à IPv6.

3 Présentation d'IPv6

Nous n'allons pas tout de suite entrer dans les détails, mais il nous faut, pour démarrer, assembler les premiers éléments sur lesquels nous allons ensuite nous appuyer. Nous approfondirons cette connaissance au fur et à mesure des articles. Nous allons nous concentrer sur l'adressage IPv6. Les adresses IPv6 sont codées sur 128 bits et non plus 32 comme pour IPv4. Les 64 bits de poids fort sont consacrés à la représentation du réseau et les 64 bits de poids faible à l'identification des hôtes. Les adresses sont représentées sous la forme de 8 digits hexadécimaux, chaque digit représentant 16 bits sous forme hexadécimale.

Voyons notre première adresse IPv6 :

```
2001:470:1f14:10b9:0000:0000:0000:2
```

est une adresse valide. Elle peut également être réécrite sous la forme :

```
2001:470:1f14:10b9::2
```

puisqu'on considère que l'écriture d'un ensemble de zéros contigus peut être simplifiée, mais pas plus d'une fois sur une adresse. Réécrire :

`2:0000:0000:0000:2:0000:0000:2` en `2::2::2` ne serait pas valide. « Pourquoi ? » se demandera l'élève attentif au fond de la classe, à côté du radiateur. Nous allons te répondre, mon petit Denis, c'est parce qu'en ce cas, il est impossible de définir quelle partie de l'adresse représente le 2 central. Nous pourrons donc écrire `2::2:0000:0000:2` ou `2:0000:0000:0000:2::2`. Pour être précis et pragmatique, cette restriction d'écriture est spécifiée par le paragraphe 2.2.2 de la **[RFC 2373]**.

3.1 Structure de la partie réseau d'une adresse IPv6

Les adresses sont structurées selon un modèle dit « agrégé » et la notation utilisée pour caractériser les agrégats est la notation CIDR **[RFC 1519]**. Voyons les 32 bits dits de poids fort à partir d'un exemple :

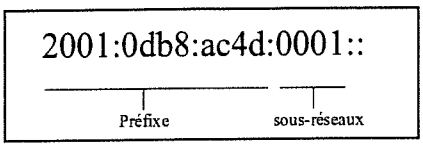

Figure 2

Le préfixe que l'on pourrait noter `2001:0db8:ac4d::/48` correspond à l'adresse de réseau du site. C'est ce qui est attribué par un opérateur. Ensuite, les 8 bits suivants peuvent être utilisés pour la création de sous-réseaux. `2001:0db8:ac4d:1010::/52` ou `2001:0db8:ac4d:1010::/54` sont valides.

3.2 Structure de la partie hôte d'une adresse IPv6

Cette partie est codée sur 32 bits. Elle permet d'identifier un hôte dans un réseau. Il y a plusieurs façons de configurer cette partie de l'adresse, nous y reviendrons.

Avant d'aller plus loin, commençons à étudier les composants d'un réseau IPv6 :

- le lien (*link*), qui est un segment Ethernet bordé par des routeurs ;
- les voisins (*neighbor*), qui sont les nœuds situés sur un même lien (link) que le nœud émetteur.

3.3 Les types d'adresses :

IPv6 définit trois types d'adresses :

- Unicast, communication entre deux nœuds.
- Multicast, identifie des nœuds appartenant à un même groupe de diffusion. Un paquet envoyé à une adresse multicast est envoyé à tous les nœuds appartenant à ce groupe. **[RFC 3306][RFC 3307]**
- Anycast. Ce type d'adresse est nouveau par rapport à ce que nous connaissons. Son rôle, schématiquement expliqué ici, est de pouvoir obtenir d'une façon optimum (au sens protocole) l'adresse d'un serveur offrant un service parmi un ensemble de serveurs offrant ce même service sans se préoccuper de savoir où il est. Nous approcherons ce type d'adresses lorsque nous aborderons les tunnels 6to4 dans le chapitre dédié à l'interconnexion de systèmes.

On s'aperçoit que la notion de diffusion ou broadcast n'existe plus. En effet,

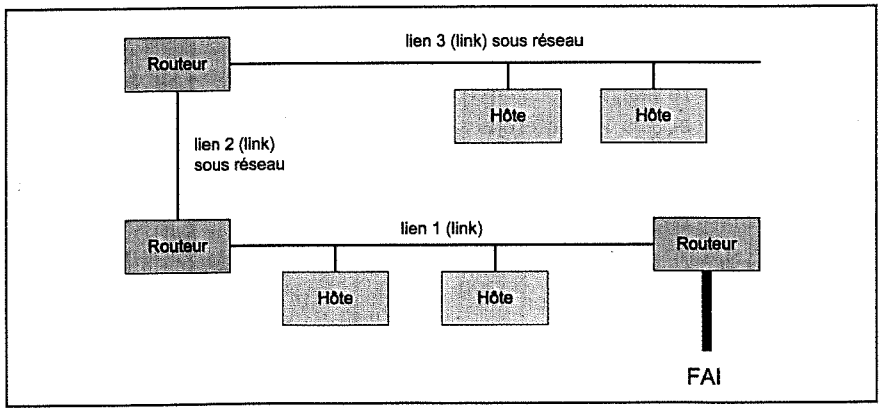

Figure 3

Nous retrouvons les mêmes dispositifs que sur un réseau IPv4, toutefois certains termes n'ont pas la même signification et il est important de bien en connaître le sens. Dans la terminologie, les plus importants sont :

il n'y a plus de protocole ARP tel qu'on le connaissait. La technique utilisée sur IPv6 est celle de « découverte des voisins » par multidiffusion.

3.4 Les préfixes réservés

Comme pour IPv4, il y a des adresses réseau utilisées à des fins bien précises.

- `2001:0db8::/32` est réservée à la documentation **[RFC3849]**.
- `2002::/16`, qui indique un préfixe de routage pour un tunnel 6to4.
- `fe80::/10`, indique une adresse de type lien local.
- `ff00::/8`, indique une adresse dite multicast.
- `fc00::/7`, préfixe des Adresses Locales Uniques utilisé pour les réseaux privés. Ce préfixe donne deux ensembles `fc00::/8` (non utilisé) et `fd00::/8`.
- `fd00::/8`, utilisé pour les sites. Ces dernières, dites « Locale Unique » ou ULA pour *Unique Local Address* **[RFC 4193]** ne sont, a priori, pas routables.

Chacune de ces classes d'adresses a une finalité bien définie, nous aurons l'occasion de le voir.

À ce stade, nous en avons assez pour commencer à expérimenter la découverte de ce nouvel environnement réseau. Deux expériences sont proposées :

- La première consiste à installer chez soi un accès IPv6 afin de pouvoir consulter des sites qui ne seraient pas accessibles autrement que par IPv4.
- La seconde est un laboratoire réalisé sous Netkit mettant en œuvre 4 machines et qui doit servir à manipuler les commandes d'administration de réseau appliqué à IPv6.

Nous irons plus loin la prochaine fois dans l'étude des adresses IP, afin de voir comment se passe l'autoconfiguration des clients, la découverte des voisins, comment des préfixes de réseaux peuvent être annoncés aux clients.

4 Mise en place d'une connexion IPv6

Il existe une possibilité d'avoir IPv6 chez soi, sur son poste, même si notre FAI ne permet pas encore cela (bouh !). La technique a été mise au point en prévision du déploiement d'IPv6. Il s'agit des tunnels brokers. Nous n'allons pas entrer dans des considérations techniques des tunnels IPv4/IPv6, nous allons juste l'utiliser, car c'est assez simple à mettre en place et ça permet surtout d'avoir une connexion IPv6 immédiatement.

Un tunnel broker ou serveur de tunnel est un dispositif mis en place par un opérateur afin de favoriser le déploiement de réseau IPv6. Il en existe plusieurs, comme SixXs **[SixXs]**, Hurricane Electric **[HE]**, gogo6/freenet6 **[Gogo]**, ... Le principe est de pouvoir créer presque automatiquement un tunnel IPv4/IPv6 pour un poste ou un groupe de postes (Fig. 4).

Ce type de solution présente plusieurs avantages :

- Expérimenter sur un réseau, chez soi, ... la mise en œuvre d'IPv6.
- Pouvoir disposer d'un accès IPv6 pour quelques heures ou quelques jours simplement et facilement quel que soit l'endroit où on se trouve pour peu qu'on dispose au moins d'un accès IPv4 (sur un salon ou dans une salle de conférence, par exemple).
- Répondre à une problématique de connectivité IPv6 dans un contexte de mobilité.

Il existe plusieurs techniques ou protocoles et toutes les solutions ne permettent pas de répondre à tous les scénarios possibles. La situation d'un poste derrière un pare-feu pose une contrainte supplémentaire. Chez freenet6, le mode de fonctionnement est de type client/serveur soit sur TCP, soit sur UDP. Le client doit être en double pile IPv4 (pour la négociation) et IPv6 (une fois la négociation réalisée). Un tunnel IPv6 dans IPv4 est mis en place. L'objectif étant de pouvoir automatiser le plus possible la connexion. Des protocoles comme TSP (*Tunnel Setup Protocol*), que nous reverrons, permettent cela. Son rôle est de pouvoir simplifier la procédure d'établissement de la liaison.

Un client TSP IPv4 établit un dialogue TCP ou UDP selon la configuration avec un serveur (*listener*), figure5.

Une fois la connexion établie, un script sur le client réalise alors la configuration locale et la connexion IPv6. Chez Hurricane Electric, la technique consiste à créer un tunnel dit 6in4 **[RFC 4213]**, qui permet d'encapsuler des paquets IPv6 dans de l'IPv4.

Cette technique a également été mise au point afin d'assurer le déploiement d'IPv6. Elle permet, par exemple, de relier une machine nativement sous IPv4 à un réseau IPv6 (fournisseur de tunnel) ou de relier deux réseaux IPv6 séparés par un réseau IPv4.

Nous allons expérimenter cela avec l'opérateur Hurricane Electric (HE). L'objectif est d'obtenir un accès IPv6 au domicile ou sur le lieu de travail. La solution, telle que nous la décrivons ici, convient pour une machine, pas pour un groupe de machines, mais en vous penchant sur ce que propose HE, vous pourrez aller beaucoup plus loin.

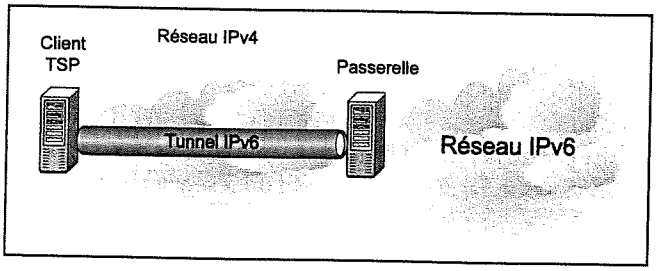

Figure 4

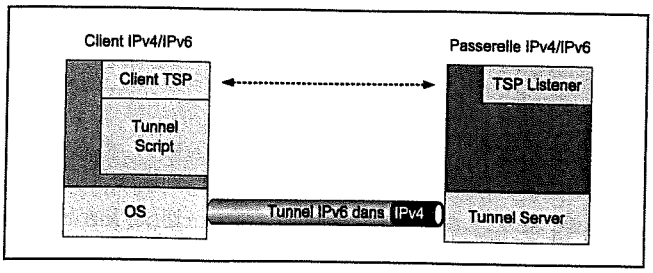

Figure 5

La première chose à faire est de vous créer un compte chez l'opérateur **[tb]**. La procédure est rapide : une fois votre adresse mail vérifiée, vous allez recevoir un compte et un mot de passe que vous devrez conserver. Vous pouvez alors vous connecter sur le site et créer votre premier tunnel **[Ist Tunnel]**. Dans la zone « IPv4 endpoint: (your side of the tunnel) », vous renseignez votre adresse publique. Vous pouvez l'obtenir par exemple via **[IPV6-FL]**, si vous êtes encore en IPv4. Il n'y a rien d'autre à faire, vous soumettez le formulaire. Une fois le tunnel créé côté opérateur, une page vous donne les paramètres. Il ne reste plus qu'à créer le tunnel côté client sur votre machine. La procédure est donnée sur la même page pour pas mal de systèmes.

Attention, notez bien que si votre ordinateur est derrière un pare-feu, pis une *box, l'opération ne marchera pas. Il faut en effet que vous soyez directement joignable sur Internet ou que vos règles de pare-feu laissent passer le ping (icmp, echo request/echo reply).

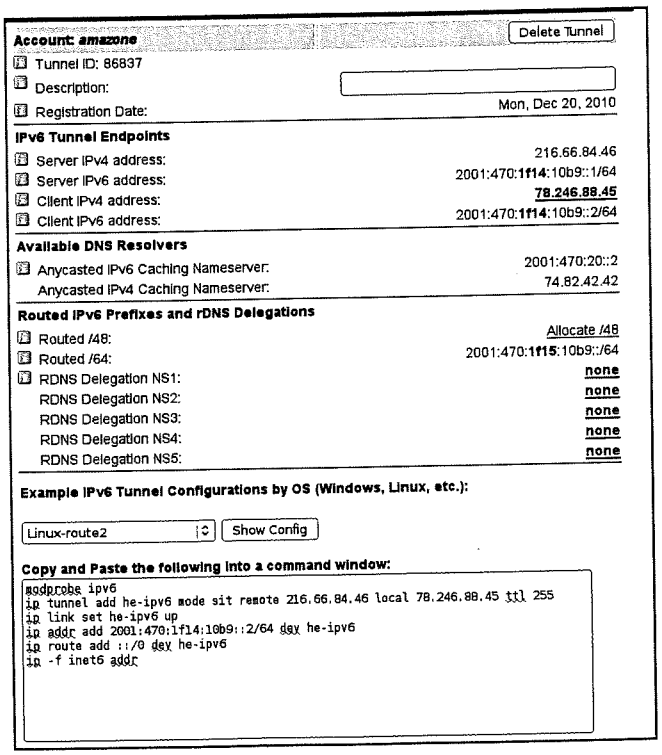

Figure 5

Sur GNU/Linux, ça donne :

```
modprobe ipv6
ip tunnel add he-ipv6 mode sit remote 216.66.84.46 local x.y.z.t ttl 255
ip link set he-ipv6 up
ip addr add 2001:470:1f14:11bc::2/64 dev he-ipv6
ip route add ::/0 dev he-ipv6
ip -f inet6 addr
```

L'adresse **x.y.z.t** est celle de votre machine, fort probablement une adresse en **192.168.Z.T**, quelque chose qui est distribué par votre point d'accès à Internet.

```
ifconfig he-ipv6 | grep inet6
        adr inet6: 2001:470:1f14:11bc::2/64 Scope:Global
        adr inet6: fe80::c0a8:102/128 Scope:Lien
```

Normalement, vous êtes sur IPv6. Sur le site du projet kame **[kame]**, vous devriez voir s'animer la petite tortue ou accéder à Google en IPv6 **[Google IPv6]**. De la même façon, vous pouvez faire un traceroute sur la pile IPv6 :

```
$ traceroute6 -n www.afnic.fr
traceroute to www.afnic.fr (2001:660:3003:2::4:20), 30 hops max, 80 byte packets
 1  2001:470:1f14:11bc::1  80.987 ms  86.033 ms  91.281 ms
 2  2001:470:0:7d::1  97.670 ms  97.656 ms  98.687 ms
(...)
 8  2001:660:3003:8001::7:227  94.409 ms  95.571 ms  97.133 ms
 9  2001:660:3003:2::4:20  98.316 ms  92.403 ms  91.427 ms
```

Cela ne fait pas tout, mais en quelques minutes, vous avez pu avoir un accès à Internet en IPv6. Vous pouvez commencer par le site de l'association G6 **[G6]**, qui donne quelques liens sur des sites grand public et techniques, accessibles en IPv6, ou encore celui de SixXS **[Déploiement]**, qui donne des indications sur le déploiement d'IPv6 par pays.

5 IPv6 et le jeu de commandes

Nous utilisons tous, si ce n'est chaque jour, disons régulièrement, un jeu de commandes comme **ifconfig**, **route**, **traceroute**, **ip**, **netstat**... Ces commandes sont-elles compatibles IPv6 ? La réponse est « oui » à quelques petites différences près. On peut classer les outils en trois grandes catégories :

- Ceux qui supportent les deux piles de protocoles mais qui, par défaut, prennent en charge IPv4. Elles nécessitent donc une option sur la ligne de commandes (comme **ifconfig**, **ip**, **netstat**, par exemple).
- Ceux qui sont développés spécifiquement pour IPv6 (**ping6**, **netcat6**, **mrd6**, **ndisc6** qui finissent subtilement par **6**...).
- Les outils additionnels qui peuvent rendre service parfois, comme **ipv6calc**, **subnetcalc** ou d'autres, que nous aurons l'occasion de citer ou d'utiliser dans la suite de cette découverte d'IPv6.

On ne cite plus les analyseurs de trame **tcpdump** et **wireshark**, qui eux, supportent complètement cette pile de protocole depuis un moment.

Nous allons nous concentrer essentiellement sur les outils courants : **ifconfig**, **ip**, **netstat**, **route** et voir comment les manipuler. Le mieux étant souvent d'expérimenter, nous allons passer par un lab Netkit mettant en œuvre trois des machines de la maquette, R4, R et C461 reliant deux segments physiques en respectant les interfaces indiquées sur le schéma global. On reliera le routeur R à l'hôte via une interface TUN/TAP. Voici la description du lab Netkit :

```
machines="R R4 C461"

R[0]=tap,10.0.0.1,10.0.0.2
R[3]="ip46"
R[5]="ip60"
R[mem]=64
R4[1]="ip60"
C461[0]="ip46"
```

Trois machines et deux segments représentés par les domaines de collision ip46 et ip60. Vous pouvez lancer le lab.

5.1 La commande ifconfig

Cette commande que vous connaissez bien donne rapidement un état des lieux de la configuration des interfaces physiques sur chaque machine. Sans paramètre, vous pouvez la tester, pour activer les interfaces qui ne le sont pas par défaut : **`ifconfig ethx up`** et consulter la configuration. Cela donne, respectivement pour R, R4 et C461 :

```
R:~# /sbin/ifconfig eth3 up
R:~# /sbin/ifconfig eth5 up
R4:~# /sbin/ifconfig eth1 up
C461:~# /sbin/ifconfig eth0 up
```

Une fois activées, on peut déjà noter que chaque interface, mis à part l'interface eth0 de R :

- n'a pas d'adresse IPv4 ;
- dispose par contre d'une adresse IPv6 qui a comme caractéristique d'être en **`fe80::/10`**. Ces adresses, qui sont de type lien local (**`Scope:Link`**), ne sont utilisables que sur un segment Ethernet, car elles ne traversent pas les routeurs et ne sont donc pas visibles sur un réseau étendu et par extension sur Internet **[RFC 5156] [RFC 4291][RFC 3587]**. Elles ont un rôle important, car elles peuvent servir, comme on le verra plus tard, à participer à l'autoconfiguration des adresses IPv6 des interfaces pour donner une visibilité plus étendue aux hôtes (**`Scope:Global`**).

Notez également le masque **`/64`**, car il a son importance et nous y reviendrons quand nous aborderons la commande **`ping6`**.

Maintenant, essayons de changer, modifier ou ajouter une adresse IPv6, avec **`ifconfig`** sur l'interface **`eth0`** de C461 (en appliquant la méthode shadockienne, chère à nos petites têtes blondes, ou la théorie d'exécuter la commande avant de lire le manuel) :

```
C461:~# ifconfig eth0 2001:db8:46::1/64 (erreur)
```

Qu'indique l'aide : **`ifconfig --help`**. Tiens ? On peut préciser le type d'adresse (ici IPv6) que l'on veut affecter. Essayons cela.

```
C461:~# ifconfig eth0 inet6 add 2001:db8:46::1/64 (succès).
```

On vérifie et que constate-t-on ?

L'interface **`eth0`** a bien deux adresses, une de type lien local et une autre de type lien global. Sur R4 et R, nous pouvons configurer les interfaces :

```
R4:~#  ifconfig eth1 inet6 add 2001:db8:60:0::fe/64
R:~# ifconfig eth5 inet6 add 2001:db8:60:0::ff/64
R:~# ifconfig eth3 inet6 add 2001:db8:46::ff/6
```

Nous allons en rester là pour l'instant, mais on retiendra que :

- une interface peut avoir plusieurs adresses IPv6 ;
- **`ifconfig`** nous permet d'ajouter ou supprimer une adresse IPv6 avec l'option **`inet6`**.

5.2 La commande route

Là, vous n'êtes déjà plus surpris, vous vous doutez déjà que le scénario va se reproduire sous la même forme ou sous une forme assez proche et vous avez raison. La commande **`route -n`** sur C461, par exemple, n'indique rien car pour l'instant, la maquette est complètement centrée sur IPv6. Par contre, la commande **`route -n -A inet6`** nous retourne pas mal d'informations. Il faut donc bien utiliser ou préciser à la commande **`route`** la pile de protocoles sur laquelle on souhaite qu'elle soit appliquée. La commande **`route`** est utilisée aussi pour ajouter ou supprimer une route.

```
# Ici une route explicite pour le réseau 2001:db8:60:0::/64
C461:~# route -A inet6 add 2001:db8:60:0::/64 gw 2001:db8:46::ff
# Ici une route par défaut
C461:~# route -A inet6 add default gw 2001:db8:46::ff
```

sont des commandes valides.

Maintenant que nous avons pu voir tout cela, remarquons également que la commande **`route -A inet6 | grep "fe80::/64"`** retourne, sur R, et cela serait vrai pour toute machine disposant de plusieurs interfaces sur des segments IPv6, une route, de même préfixe, mais sur des interfaces différentes. Et là encore, quand nous utiliserons la commande **`ping6`**, cela aura toute son importance.

5.3 La commande netstat

Cette commande a un comportement un peu différent. Sans paramètre, elle indiquera l'état des connexions (serveur et en activité) sur les deux piles de protocoles. Avec le paramètre **`-A inet`**, elle omettra de mettre ce qui concerne IPv4. Activez par exemple ssh sur R :

et regardez le résultat des commandes :

```
R:~# /etc/init.d/ssh start
R:~# netstat -natup
R:~# netstat -natup -A inet6
```

La seconde commande ne produit le résultat que pour les adresses IPv6. Si vous n'avez rien, ne vous inquiétez pas, c'est que vous (ou votre mainteneur) avez désactivé l'écoute du service ssh sur IPv6 (enfin, probablement ;)).

5.4 La commande ip

Vous le savez probablement déjà, **ip** est la seule commande en usage dans les huttes des panoram*nix du net. Nous allons passer un peu de temps dessus, car il est vrai qu'elle permet de renseigner sur des valeurs ou paramètres que les autres commandes ne font pas.

Là encore, **ip --help 2>&1 | grep inet6** indique bien que nous avons la possibilité de préciser la famille de protocoles sur laquelle nous agissons. Pour nous, ce sera **-f inet6**, ou plus court, **-f 6**, mais l'usage est plus destiné à filtrer la sortie que pour la configuration proprement dite. **ip addr show** affichera tout. **ip -6 addr show** n'indiquera que ce qui concerne IPv6. Maintenant, pour l'essentiel et très rapidement résumé :

Voir les adresses :

```
ip [-6] addr show [interface]
```

Ajouter ou supprimer une adresse :

```
ip addr [add|del] address/masque dev interface
```

Afficher les routes :

```
ip [-6] route show [ interface]
```

Ajouter ou supprimer une route : **ip route [add|del] destination/masque via passerelle**. Par exemple, en prenant toujours C461 comme exemple, les commandes suivantes sont valides :

```
C461:~# ip  route add  2001:db8:60:0::/64 via  2001:db8:46::ff
C461:~# ip  route del  2001:db8:60:0::/64 via  2001:db8:46::ff
C461:~# ip  route add default via 2001:db8:46::ff
C461:~# ip  route del default via 2001:db8:46::ff
```

Sur la commande **ip**, le guide d'administration de la couche IP **[Linux-IP]** est une excellente source.

5.5 La commande ping6

ping6 remplace, sur IPv6, la commande **ping** pour IPv4. Voyons concrètement si elle fonctionne comme on pourrait le souhaiter. Relevez sur C461 l'adresse de type lien local de **eth0** et sur R celle de **eth3**. Mettons que vous ayez :

```
C461 : eth0 inet6 addr: fe80::10de:89ff:fe19:21dd/64 Scope:Link
R    : eth0 inet6 addr: fe80::3047:cbff:fe52:16e2/64 Scope:Link
```

À partir de R, tentez :

```
ping  -c1 fe80::10de:89ff:fe19:21dd
ping6 -c1 fe80::10de:89ff:fe19:21dd
ping6 -c1 -I eth3  fe80::10de:89ff:fe19:21dd
```

Avez-vous noté les messages d'erreur sur les deux premières commandes ? Seule la dernière commande fonctionne. Les commandes **ping** et **ping6** sont donc bien différentes. Quand il s'agit d'adresser un hôte sur un lien local, il est indispensable de passer en paramètre l'interface concernée.

Voyons maintenant comment cela se passe avec les adresses de type global (**Scope:Global**). Nous avons déjà configuré les interfaces des machines du lab pour utiliser la commande **ifconfig**, mais revenons un peu dessus. Voilà le plan d'adressage. Il n'est pas arbitraire, pensez à suivre les recommandations des RFC.

- Domaine ip60 : **2001:db8:60:0::/64**
- Domaine ip46 : **2001:db8:46::/64**

On utilise des sous-réseaux 60.0 et 46 afin de les rapprocher de l'espace de nommage des segments pour faciliter la lecture. On prend, par pure convention, les adresses les plus hautes pour les routeurs, cela donne donc :

```
R:~#    ifconfig eth5 inet6 add 2001:db8:60:0::ff/64
R:~#    ifconfig eth3 inet6 add 2001:db8:46::ff/64
R4:~#   ifconfig eth1 inet6 add 2001:db8:60:0::fe/64
C461:~# ifconfig eth0 inet6 add 2001:db8:46::1/64
```

Maintenant, revenons sur R pour voir ce que donne la commande **ping6 -c 3 2001:db8:46::1**. Nous voyons que ça fonctionne. Nous sommes sur un segment (vu au niveau IPv6) global, plus besoin de préciser l'interface.

5.6 Conclusion sur le jeu de commandes

Ce lab, bien que très simple, peut permettre de commencer à se familiariser avec d'autres aspects du fonctionnement d'IPv6. Prenons deux exemples. Le premier est une transposition de ce que nous connaissons sur IPv4 : le routage statique ; le second un aspect plus spécifique à IPv6 concernant la résolution d'adresse MAC.

5.6.1 Entrée dans le routage statique

Il serait possible de mettre des routes entre les segments ip60 et ip46, sur R4 et c461 en se servant de R comme routeur :

```
# Activation de l'ipforwarding sur R qui relie les segments ip60 et ip46 :
R:~# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
# On met des routes explicites (on aurait pu mettre des routes par défaut) sur R4 et C461.
C461:~# ip  route add  2001:db8:60:0::/64 via  2001:db8:46::ff
# ou alors avec la commande route
C461:~# route -A inet6 add 2001:db8:60:0::/64 gw 2001:db8:46::ff
R4:~#  ip route add 2001:db8:46::/64 via 2001:db8:60:0::ff
```

Désormais, sur R4, **ping6 -c1 2001:DB8:46::1** doit répondre.

5.6.2 Analyse de trame - on recherche ARP

Il s'agit ici, sans rentrer dans les détails, car nous ne nous concentrons que sur les outils, de toucher du doigt ce qui se passe sur un segment. Que se passe-t-il, sous IPv6, quand nous envoyons un ping sur une autre adresse IPv6 ? Comment la résolution d'adresse MAC est-telle réalisée ? Voyons-nous des trames de diffusion (*broadcast*) ? Pour voir cela, nous allons utiliser un analyseur de trames sur R avant de lancer le ping6. Tentons l'expérience :

```
R:~# tcpdump -n -i eth0 -w /hostlab/out.snap.ping6&
R:~# ping6 -c1 2001:DB8:46::1
```

Vous pouvez arrêter la capture sur R avec **fg** puis CTRL+C et ouvrir avec Wireshark le fichier **out.snap.ping6**, qui est dans le répertoire où vous avez lancé votre lab. Comme vous pouvez le constater, pas de trame ARP ni de trame de diffusion, mais avant de pouvoir envoyer une trame ICMP echo request, le protocole a quand même besoin de connaître l'adresse MAC du destinataire (Fig. 6).

C'est ce que montrent les deux premières trames : *Neighbor Solicitation* et *Neighbor Advertisement*. Le protocole de découverte des voisins (Neighbor) sous IPv6 n'utilise pas de la diffusion, potentiellement trop coûteuse en bande passante, mais du multicast pour la première requête : Neighbor Solicitation. Nous aurons l'occasion de revenir sur ce protocole de découverte des voisins : NDP (*Neighbor Discovery Protocol*).

Vous pouvez consulter le cache des voisins sur une machine avec la commande **ip neigh show**, tout comme on le fait sous IPv4 avec la commande **arp**.

Impossible toutefois de partir dans cette séquence sans jouer une dernière fois avec les commandes et les adresses IPv6. Tentons une expérience.

Que dit **/etc/hosts** sur PC0 : **ff02::1 ip6-allnodes** ? C'est celle-ci qui nous intéresse. Que se passe t-il si on faisait un ping6 sur cette adresse ? Aurait-on les adresses de tous les voisins ? Bonne question !

Voyons notre cache local sur R :

```
R:~# ip neigh show
```

Visiblement pas grand chose.

Tentons les commandes :

```
R:~# ping6 -Ieth5 -c1 ff02::1
R:~# ping6 -Ieth3 -c1 ff02::1
R:~# ip neigh show
```

Alors là, oui, la communauté de voisins (à ne pas confondre avec la fête des voisins), ce qui correspondait au cache ARP en IPv4, se peuple. Nous n'allons pas aller plus loin pour le moment, mais nous reviendrons sur ces adresses multicast. L'objet ici était de manipuler les commandes et de montrer comment elles nous permettent de commencer à rentrer dans l'environnement des protocoles.

6 Conclusion

Vous venez de faire vos premiers pas dans IPv6, félicitations. Évidemment, si vous avez toujours refusé de boire un verre de CIDR, de jouer avec l'hexa quand vous étiez petit ou encore banni toute notion de routage de votre mode de vie, tout cela a pu vous paraître quelque peu abstrait. Nous touchons du doigt l'entrée dans une nouvelle ère et vous en êtes les acteurs majeurs. Vous avez l'opportunité de permettre à Internet de revenir à sa conception initiale : tout poste connecté peut à la fois être client ET serveur. Deux cafés, l'addition. Plus de pleutrerie derrière un NAT, plus de raison de se voir refuser l'attribution d'une adresse publique, vous pouvez dorénavant connecter votre grand-mère en IPv6. Tout du moins sa cafetière...

N'hésitez pas à nous écrire pour nous permettre d'orienter les articles qui vont suivre, pour vous permettre de mieux répondre aux questions que vous pouvez vous poser. En attendant, on vous donne rendez-vous à la prochaine publication de cette série, qui reviendra sur les notions de plan d'adressage, de routage et de DHCP.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	2001:db8:46::ff	ff02::1:ff00:1	ICMPv6	Neighbor solicitation
2	0.000321	2001:db8:46::1	2001:db8:46::ff	ICMPv6	Neighbor advertisement
3	0.000350	2001:db8:46::ff	2001:db8:46::1	ICMPv6	Echo request
4	0.000558	2001:db8:46::1	2001:db8:46::ff	ICMPv6	Echo reply

Figure 6

Pour finir, nous rappelons aux plus candides que même avec des lunettes 3D et une connexion IPv6 fonctionnelle, la tortue de kame imprimée sur papier ne devrait effectuer aucune danse... Sinon, n'hésitez pas à nous consulter !! ;)

Liens

Retrouvez la maquette et les labs sur notre site : http://formation-libre.fr/articles/

IPv6 fanboy

- [kame] http://kame.net
- [ipv6_forum] http://www.ipv6forum.com/
- [ipv6_uk] http://www.ipv6.org.uk/
- [Starwars] telnet://towel.blinkenlights.nl
- [Google IPv6] http://ipv6.google.com/
- [Mon IPv6] http://whatismyipv6.com/
- [Gogo6] http://v6.gogo6.com/
- [IPV6-FL] http://ipv6.formation-libre.fr

Lab

- [netkit] http://netkit.org

Tunnel Brokers

- [SixXs] http://www.sixxs.net/
- [HE] http://www.he.net/
- [Gogo] http://gogonet.gogo6.com/
- [tb] http://tunnelbroker.net/register.php
- [1st Tunnel] http://tunnelbroker.net/ipv6_normal.php
- [WIMI] http://www.whatsmyip.org/

Sites

- [G6] http://www.g6.asso.fr/
- [Déploiement] http://www.sixxs.net/tools/grh/dfp/
- [Linux-IP] http://linux-ip.net/html/index.html

RFC

Pour lire les RFC citées dans l'article, rendez-vous sur http://www.rfc-editor.org/rfcsearch.html et insérez le numéro de la RFC dans le champ de recherche. ■