# Le Protocole Internet Version 6


## Résumé

Ce document présente le fonctionnement du protocole internet et en particulier IPv6. Son but est de préparer l'administrateur système et réseau à la mise en oeuvre de ce protocole. Pour atteindre ce but, une approche progressive à été utilisée. Le document commence par un ensemble de rappels de bases sur IPv4 puis une présentation des principes de base d'IPv6. Suit ensuite une présentation détaillée de ce protocole et enfin, divers cas de figure pour la mise en place d'IPv6 sont abordé à la fin du document afin d'en avoir une approche pratique.

## Abstract

This document is aim to present the IPv6 protocol in the goal of making the network and system administrator ready to use this protocol. Several example will help him to get a global approach of it.

## Table des matières

# INTRODUCTION

L'avènement du protocole IPv6 est aujourd'hui inéluctable et sa mise en oeuvre a déjà commencé à plus ou moins grande échelle dans certains pays.

Ce document a pour but de démystifier autant que possible ce protocole et de présenter les avancées proposées par IPv6. Une place importante sera accordée à la mise en oeuvre de ce protocole dans différents cas de figure allant de la configuration de base d'un système Linux ou OpenBSD à la mise en oeuvre du service DNS ou la configuration d'un serveur web Apache.

Des connaissances de base dans le protocole IPv4 sont requise pour la bonne compréhension de ce document.

# 1 - HISTORIQUE

Les travaux de l'ARPA (*Advanced Research Projects Agency*) sur l'interconnexion de réseaux ont commencé dans les années 1970 et les protocoles actuels prirent leur forme actuelle dans les années 1977-1979. Cette agence a été la première à mettre au point et à tester différentes techniques mettant en oeuvre la commutation de paquets via son célèbre réseau ARPANET.

Le réseau Internet actuel a vu le jour vers 1980 lorsque le réseau ARPANET migra vers l'utilisation des protocoles TCP/IP. Cette migration fût achevée au début de l'année 1983. A ce moment, la DCA (*Defense Communication Agency*) divisait ARPANET en 2 réseaux : un pour la recherche (nommé ARPANET) et un pour la défense (nommé MILNET).

La première pile de protocoles TCP/IP la plus largement utilisée fût implémentée dans le système BSD (*Berkeley System Distribution*) dans sa version 4.2. 90% des départements informatiques des universités américaines migrèrent alors vers TCP/IP.

Le succès de TCP/IP fût fulgurant dans la communauté scientifique et d'autres catégories d'utilisateurs commencèrent à s'y intéresser. La NSF (*National Science Foundation)* jouât un rôle important dans cette diffusion. De 1985 à 1986, elle oeuvra pour la mise en place de plusieurs réseaux régionaux et à leur interconnexion. C'est ainsi que naquit le réseau Internet.

La croissance d'Internet est actuellement estimée à 15% par mois (environ 50 millions d'ordinateurs connectés en 2001). Cette croissance exponentielle a posé un certain nombre de problèmes qui étaient inimaginables dans les années 70! La résolution de chacun de ces problèmes a, à chaque fois, apporté une révolution dans le monde Internet, mais le problème vers lequel nous nous dirigeons inexorablement au fil du temps changera la face du réseau mondial. Il causera certainement la révolution la plus importante depuis l'apparition d'Internet : la pénurie d'adresses IP…

# 2 - QU'EST-CE QUE LE PROTOCOLE INTERNET ?

Le protocole Internet est un protocole appartenant à la la couche 3 du modèles OSI (couche réseau) permettant de faire communiquer divers équipements (ordinateurs, imprimantes, téléphones mobiles?) au travers d'un réseau. Il attribue une adresse logique aux différents équipements d'un réseau en faisant abstraction de l'adresse physique qui dépend de la technologie employée pour le réseau (ethernet, token ring...). Ainsi chaque équipement actif d'un réseau (ordinateur, imprimante, routeur...) possède une adresse IP et une adresse physique.

Les adresses physiques sont utilisées dans la couche 2 du modèle OSI. Ces adresses sont théoriquement uniques et enregistrées directement dans le matériel par le constructeur et sont donc peu aisées à changer. Ces adresses servent au matériel pour effectuer la transmission des données.

Si seul ce type d'adresses étaient utilisées sur les réseaux, le remplacement du matériel deviendrait vite un cauchemar pour les administrateurs, en effet, imaginez qu'en plus de changer la carte réseau d'un serveur, il faudrait modifier la configuration du serveur de noms de domaine et des

autres composants du réseau afin que le serveur puisse communiquer avec tout le monde...

En ajoutant une couche d'abstraction supplémentaire, IP fourni une adresse logique facilement modifiable au niveau du système d'exploitation. Ensuite, afin de permettre la communication sur un réseau utilisant les adresses physiques, divers systèmes permettant de faire le lien entre l'adresse logique et l'adresse physique sont utilisés. Ces systèmes seront étudiés dans les parties consacrées à IPv4 et IPv6.

Le protocole Internet a désormais supplanté pratiquement tous les anciens protocoles de communication de la couche 3, en particulier IPX, AppleTalk...

# 3 - Pourquoi IPv6 verra-t-il le jour ?

J'ai choisi d'ajouter cette section suite à plusieurs remarques qui m'ont été faites relative à l'avènement d'IPv6. En effet, certaines personnes m'ont dit qu'elle ne croyait pas (plus) trop à son arrivée un jour, d'autres encore, m'ont carrément dit que, bien que la question d'IPv6 ai été intéressante il fut un temps, il est désormais devenu inutile grâce à l'utilisation de certaines "techniques" permettant de s'affranchir des anciennes limitations d'IPv4.

A la première catégorie de personnes, je ne peux que leur répondre d'ouvrir un peut les yeux et de faire quelques recherches sur Internet pour se rendre compte qu'IPv6 est déjà une réalité.

La plupart des logiciels, systèmes d'exploitation et matériels existant actuellement supportent ce protocole. Des exemples ? Il y a en a plein, on peut jeter pèle-mêle les systèmes Linux, *BSD, Windows 2003, les logiciels Bind, Apache, X11, Mozilla Firefox, Internet Explorer, Samba, SSH les routeurs Cisco....

Je pourrais en ajouter une couche en disant qu'IPv6 est déjà entré en phase de tests intensifs grandeur nature, par exemple, sur le réseau fédérateur Français RENATER 3 ou sur le réseau fédérateur Japonais Wide ou sur le 6Bone. Des points d'accès à Ipv6 sont déjà présents de par le monde : Paris, New York, Munich, Tokyo

A la deuxième catégorie de personnes, je dirais que les technologies permettant de s'affranchir des limitations d'IPv4 relèvent plus de bricolage que de véritables technologie. Premier exemple d'entre elles, le NAT[1]. Bien que parfaitement fonctionnelle, cette technique d'économie d'adresse n'est pas viable à long terme à cause d'un certain nombre de limitation :

- Il est impossible de faire communiquer deux équipements ayant la même adresse IP et situés chacun derrière un NAT à moins de mettre en place un système complexe d'alias
- Les techniques de NAT utilisées (association entre l'adresse IP et le port utilisé par l'émetteur et le port utilisé sur le firewall) ne permettent de partager une adresse qu'avec seulement 64000 postes théoriquement.
- Plus des deux tiers des adresses IP disponibles sont réservées par les pays d'Amérique du Nord, les pays asiatiques en particulier l'Inde et la Chine, regroupant plus du tiers de la population du globe ne disposent que de 10% des adresses IP disponibles
- L'espace d'adressage théorique d'IPv4 n'est pas utilisable en entier. Comme expliqué dans la RFC 3194, le nombre d'adresses réellement exploitable n'est que 240 millions

1 **Network Address Translation** : technique permettant de partager une adresse IPv4 entre plusieurs équipements

- Dans l'avenir, l'utilisation du protocole IP ne sera plus limité qu'au surf sur internet, il sera également utilisé pour la téléphonie mobile et d'autres domaines des télécommunications.

Comme on le voit, il sera absolument impossible pour des raisons diverses et variées de régler tous ces problèmes énoncées grâce au NAT.

Certes, IPv6 n'est pas pour demain et il faudra encore quelques années avant qu'il ne se répande à grande échelle, mais une chose est sûre, il faudra bien y passer un jour.

Un argumentaire complet sous forme de question réponses est disponible sur le site du 6Wind (www.6wind.com).

# I – RAPPEL SUR LE PROTOCOLE IPV4

# 1 - Présentation du protocole IPv4

## 1.1 - Qu'est-ce qu'IPv4

IPv4 est la première version du protocole IP a avoir été utilisée et est celle qui est utilisée actuellement. Ce protocole est défini dans la RFC 791.

## 1.2 - La trame IPv4

L'entête de la trame IPv4 est constituée de 14 champs répartis comme suit :

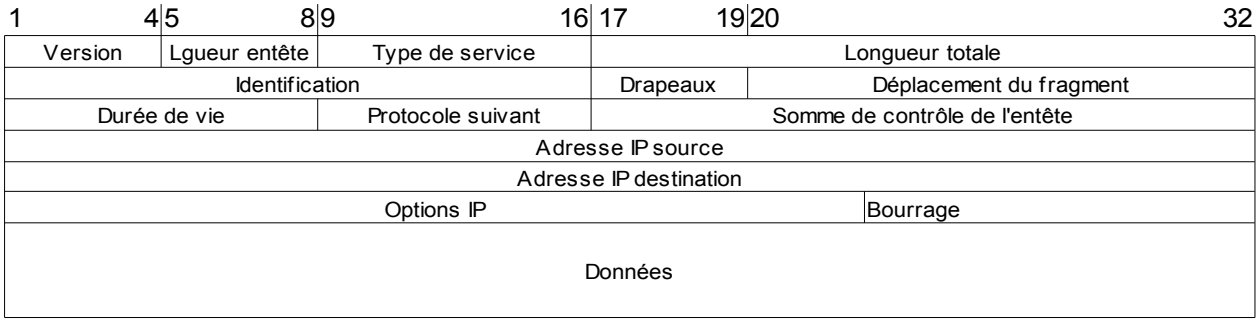



*Illustration 1: La trame IPv4*

Voici la signification des différents champs :

- **Version** (4 bits) : indique quelle est la version du protocole (ici 4)
- **Longueur de l'entête** : indique la longueur de l'entête du datagramme
- **Type de services** (8 bits) : indique aux routeurs comment doit être géré le datagramme
- **Longueur totale** (16 bits) : indique quelle est en octets la longueur totale du datagramme (entête et données)
- **Identification** (16 bits) : identifiant permettant de réassembler le datagramme
- **Drapeaux** : divers drapeaux de contrôle
- **Déplacement du fragment** : indique quelle est la position du paquet si celui-ci est un fragment de datagramme
- **Durée de vie** (8 bits) : indique le nombre de routeurs que peut traverser le datagramme
- **Protocole** (8 bits) : identifie le protocole de niveau supérieur (TCP, ICMP...) utilisé pour transmettre le message
- **Total de contrôle entête** (16 bits) : permet de détecter les erreurs de transmission dans l'entête.
- **Adresse IP source** (32 bits) : renseigne l'adresse IP de l'expéditeur
- **Adresse IP destination** (32 bits) : renseigne l'adresse IP du destinataire
- **Options IP éventuelles** (taille inférieure ou égale à 32 bits) : options concernant des fonctionnalités de mise au point
- **Bourrage** : le champ option n'a pas de taille fixe. Le bourrage permet de faire atteindre à ce champ une taille multiple de 32 bits (4 octets)

## 1.3 - Le système d'adressage

Les adresses IPv4 sont codées sur 32 bits ce qui permet d'attribuer 4 294 967 296 adresses. Elles sont notées sous la forme de 4 chiffres compris entre 0 et 255 sous la forme :

192.168.0.23

A cette adresse est également ajouté un masque indiquant à quel réseau appartient l'équipement auquel cette adresse a été attribuée. Ce masque indique quelle partie de l'adresse

renseigne sur l'adresse du réseau et est noté de la manière suivante :

192.168.0.23/24 (notation moderne) ou 192.168.0.23/255.255.255.0 (notation ancienne)

Dans cet exemple, le "/24" indique que les 24 premiers bits composant l'adresse forme l'adresse du réseau, ici, le réseau a donc pour numéro : 102.168.0.0.

Par la suite, c'est la notation "moderne" (également, appelée notation "C.I.D.R.") qui sera employée dans ce document.

### *1.3.1 - Classes d'adresses*

Le protocole IPv4 défini 5 classes d'adresses nommées simplement adresse de classe A, B, C, D ou E. Ces classes définissent combien d'ordinateurs et de réseaux il est possible de constituer sur un site en vue de leur raccordement à Internet avec des adresses publiques qui sont attribuées par le fournisseur d'accès. Pour une utilisation dans un réseau local non connecté à Internet, il n'est pas obligatoire de suivre cette norme si ce n'est par question d'habitude.

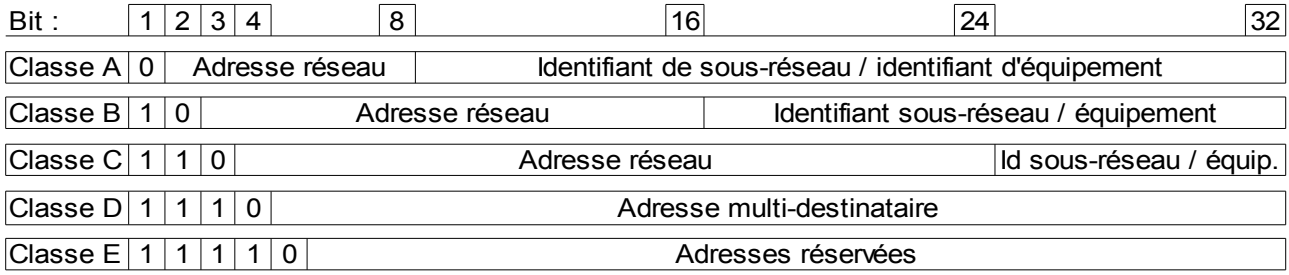



*Illustration 2: Classes d'adresses de réseau*

Les adresses de classe A sont reconnaissables sous leur forme binaire car leur premier bit est à zéro. Ces adresses sont comprises entre 1.0.0.0 et 126.0.0.0 et ont par défaut un masque de 255.0.0.0. Ces adresses sont utilisées pour les réseaux comportant plus de 65 536 ordinateurs (216). Elles attribuent 7 bits pour l'identification du réseau et 24 pour l'identification de chaque machine.

Les adresses de classe B ont leurs deux premiers bits commençant par 10. Elles permettent d'allouer des adresses comprises entre 128.1.0.0 et 191.255.0.0 avec le masque par défaut qui est de 255.255.0.0. Elles sont utilisées sur les réseaux intermédiaires comportant entre 256 (28) et 65 535 ordinateurs. 14 bits sont alloués pour l'identification du réseau et 16 pour identifier chaque ordinateur.

Les adresses de classe C ont leurs 3 premiers bits commençant par 110. Les adresses qu'elles permettent d'allouer sont comprises entre 192.0.1.0 et 233.255.255.0 et le masque par défaut est 255.255255.0. Elles sont utilisées pour les petits réseaux comportant moins de 256 machines. 21 bits sont attribués à l'identification du réseau et 8 à l'identification de chaque ordinateur.

Les adresses de classe D sont identifiables grâce à leurs 4 premiers bits qui sont 1110 et sont comprises entre 224.0.0.0 et 239.255.255.255. Ce sont des adresses multidestinataires.

Enfin, les adresses de classe E, reconnaissables à leurs 5 premiers bits égaux à 11110 sont réservées pour une utilisation ultérieure. Elles sont comprises entre 240.0.0.0 et 255.255.255.255

Chacune de ces classes réserve des adresses privées pour les réseaux privés devant être raccordés à un réseau public. Pour la classe A, ce sont des adresses commençant par 10.x.x.x, en classe B : 172.16.x.x à 172.31.x.x et en classe C : 192.168.x.x. A priori, aucun équipement n'a le droit de communiquer sur un réseau public avec une adresse privée. Ces plages sont définies dans la RFC 1918.

### *1.3.2 - Adresses particulières*

Un certain nombre d'adresses particulières sont définies par IPv4. Ces adresses sont réservées à des usages particuliers.

*a - La boucle locale*

La boucle locale correspond à une interface réseau virtuelle présente sur la quasi totalité des équipements. Elle est utilisée pour les communications entre les processus. Ces processus peuvent aussi bien être des jeux ou les systèmes d'impression sur Unix (Cups).

Dans la pratique, l'adresse de cette interface est toujours 127.0.0.1/8. Normalement, un paquet émis sur cette interface ne devrait jamais apparaître sur un réseau.

*b - Les adresses de broadcast*

Les adresses de broadcast (également appelées adresses de diffusion) correspondent à l'adresse la plus haute que l'on puisse trouver sur un réseau. Par exemple, pour le réseau 192.168.0.0/24, l'adresse de broadcast sera 192.168.0.255.

L'adresse de broadcast 255.255.255.255 est principalement utilisée par les équipements ne disposant pas d'adresse IP sur le réseau (par exemple, lors d'une auto-configuration via DHCP). Les équipements utilisant cette adresse de broadcast annoncent généralement l'adresse IP 0.0.0.0 (pas d'adresse IP).

# 2 - PROTOCOLES ASSOCIÉS

## 2.1 - La résolution d'adresse physique

La résolution d'adresse physique permet de faire le lien entre l'adresse IPv4 logique d'une machine et son adresse physique. Afin de comprendre pourquoi la résolution d'adresse physique est difficile avec certaines technologies de réseaux, nous allons considérer le cas de la technologie ethernet qui est la plus utilisée dans les réseaux locaux.

Sur les réseaux ethernet, les adresses physiques sont codées sur 48 bits ce qui rend impossible de créer un lien direct entre ce type d'adresses et une adresse IPv4 codée sur 32 bits. Aussi, il a été créé un protocole spécifique permettant de réaliser un lien indirect entre ces deux adresses. Il s'agit du protocole ARP (Address Resolution Protocol) qui fournit un système efficace et simple de faire cette résolution.

Quand un ordinateur A veut connaître l'adresse physique d'un ordinateur B, il diffuse une trame spéciale sur le réseau demandant à l'ordinateur B de répondre en indiquant son adresse physique. Tout les ordinateurs du réseau reçoivent cette trame mais seul l'ordinateur B reconnaît son adresse IP. Il renvoie alors la réponse. A enregistre alors cette adresse dans une table de correspondances et peut alors directement transmettre à B le datagramme IP en se servant de l'adresse physique.

Cette technologie fait donc appel à un second protocole de couche 3 en plus du protocole IP. Ce protocole multiplie également le nombre de trames de diffusion circulant sur le réseau et est

donc susceptible de provoquer des saturations.

## 2.2 - Le protocole ICMP

Le protocole ICMP (Internet Control Message Protocol) est défini dans la RFC 792. Il a pour rôle l'échange de messages d'information de base entre des équipements communiquants.

Les messages diffusés par ICMP servent à la gestion des erreurs aussi bien qu'à la transmission d'informations. Ces paquets peuvent être émise dans plusieurs cas tel que :

- L'utilisation de la commande ping permettant de vérifier si un poste est bien configuré sur le réseau
- Dans le cas d'un paquet dont la durée de vie a expirée
- ....

Le format des paquets ICMP est simple : Un champ "type" indique de quel type est le message ICMP, un champ code permet d'avoir des informations plus détaillées, un checksum, une suite de champ dont le contenu varie suivant le type de message et enfin, une partie du paquet IP ayant provoqué l'émission du paquet ICMP.

## 2.3 - Le protocole IGMP

Le protocole IGMP (Internet Group Message Protocol) est défini dans la RFC 1112. Son rôle est de gérer les groupes multicast dans IPv4.

Un groupe multicast est un ensemble d'équipement écoutant sur un même adresse IP. Cette technologie peut être entre autre utilisée dans le cas de la diffusion de flux en continu tel que des flux de radio sur Internet ou pour la télévision sur ADSL.

# II – LE PROTOCOLE IPV6

# 1 - Présentation du protocole IPv6

Références des RFC : 2460.

## 1.1 - Qu'est-ce qu'IPv6 ?

IPv6 est la prochaine génération du protocole Internet.

Pourquoi IPv6 et non IPv5 ? Tout simplement à cause du numéro de version contenu dans les 4 premiers bits indiquant le numéro du protocole. En effet, le numéro 5 correspond au protocole STP (Stream Protocol, RFC 1819).

## 1.2 - La trame IPv6

Une trame IPv6 est composé d'un entête de base auquel peuvent s'ajouter des entêtes d'extension suivant les besoins. Ceci est l'entête de base d'IPv6 :

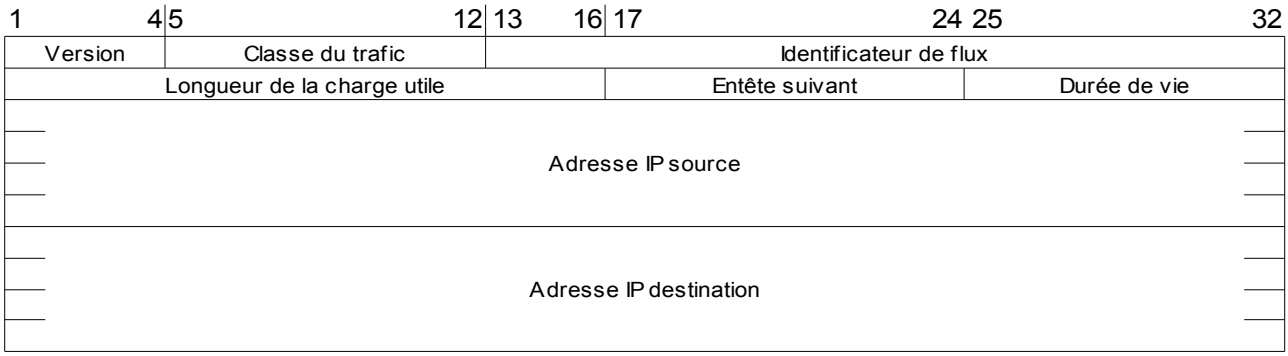



*Illustration 3: L'entête de base IPv6*

La taille de cet entête de base est fixé à 40 octets. Les champs sont les suivants :

- **Version** (4 bits) : indique la version du protocole utilisé (ici 6)
- **Classe de trafic** (8 bits) : permet de définir une classe de trafic.
- **Identificateur de flux** (20 bits) : indique aux routeurs de garantir une qualité de services spécifiques
- **Longueur de la charge utile** (16 bits) : indique le nombre d'octets transportés par le datagramme hors entête de base
- **Entête suivant** (8 bits) : identifie l'entête optionnel suivant l'entête de base
- **Durée de vie** (8 bits) : durée de vie de la trame en nombre de routeurs traversables
- **Adresse IP Source** (128 bits) : adresse de l'expéditeur
- **Adresse IP Destination** (128 bits) : adresse du destinataire

## 1.3 - Système d'adressage

Les adresses IPv6, du fait de leur longueur importantes sont notées de manière différents que les adresse IPv4. Ce système garanti que l'on aura suffisamment d'adresses IP si un jour l'Homme veut étendre le réseau Internet jusqu'à la Lune ou Mars, en effet, on peut désormais attribuer $3{,}4.10^{38}$ adresses...

Ceci est un exemple d'adresse IPv6 :

fe80:0000:0000:0000:0240:caff:febf:dd81/64

On remarquera tout d'abord l'utilisation du système hexadécimal pour la notation et ensuite la notation du masque. Cette dernière utilise désormais exclusivement la notation CIDR.

Lorsque l'on examine cet exemple d'adresse, on s'aperçoit qu'elle contient un nombre important de zéros successifs. On peut parfaitement condenser cette écriture de cette manière en remplaçant la suite de zéros par « :: » (deux fois deux points) et en supprimant les zéros inutiles :

fe80::240:caff:febf:dd81/64

Cependant, il faut respecter certaines règles, cette compression ne peut être effectuée qu'une seule fois dans une adresse, ainsi :

- si l'adresse de départ est : fe80:0000:0000:caff:240:0000:febf:dd81, on ne peut pas la condenser en fe80::caff:240::feb:dd81. En effet, il devient alors impossible de déterminer combien de zéros sont compris entre chaque groupe de « :: ».
- il n'est pas possible également de condenser en supprimant le zéro de fe80, en effet, fe8 n'est pas égal à fe80 mais à 0fe8...

Une adresse IPv6 est composée de deux parties : la partie identifiant du réseau, appelée préfixe, qui est repérée grâce au masque de réseau et la partie adresse de l'équipement.

Enfin, en IPv6, il devient standard d'avoir plusieurs adresses sur une même interface réseau.

### *1.3.1 - Types d'adresses*

IPv6 défini 3 type d'adresses :

- Unicast (point à point) : cette adresse spécifie une machine particulière à laquelle le datagramme doit être envoyé par le plus court chemin.
- Anycast : la destination est un groupe d'ordinateurs partageant une même adresse. Ce type d'adresse indique que le datagramme doit être transmis à un seul membre de ce groupe (par exemple, le plus proche). Ces adresses ne peuvent servir qu'en tant qu'adresses de destination.
- Multicast (point à multipoint) : la destination est un groupe d'ordinateur partageant une même adresse. Ce type d'adresse indique que le datagramme doit être transmis à tout les membres du groupe éventuellement en utilisant les possibilités de diffusion au niveau du matériel. Ces adresses ne peuvent servir qu'en tant qu'adresses de destination. Ces adresses sont toutes identifiables par leur préfixe commençant par FF.

### *1.3.2 - Adresse particulières*

Comme IPv4, le protocole IPv6 défini un certain nombre d'adresses ou de plages d'adresses dédiées à un usage particulier :

- ::1/128 : la boucle locale
- FE80 : l'adresse lien-local servant en local à désigner une interface réseau.
- FF01:: : Noeud local. Un paquet émis sur cette adresse ne quitte jamais l'équipement
- FF02::1 : Lien local. Sert à désigner tous les noeuds IPv6 situés sur le même lien local que l'interface de connexion. Ils ne doivent pas être routés sur un autre réseau.
- FF02::2 : Sert à désigner tous les routeurs connectés sur le même lien local que l'interface de connexion.
- FF02::3 : Sert à désigner tous les hôtes situés sur le lien local.
- :: désigne l'adresse indéterminée.
- FF05:: : Désigne tous les hôtes du réseau local.
- FEC0::/10 : Adresse de réseau privé

*a - L'adresse de lien local*

Cette adresse est composée du préfix FE80::/10 à laquelle est accolée l'adresse physique de l'équipement en notation EUI-64, par exemple :

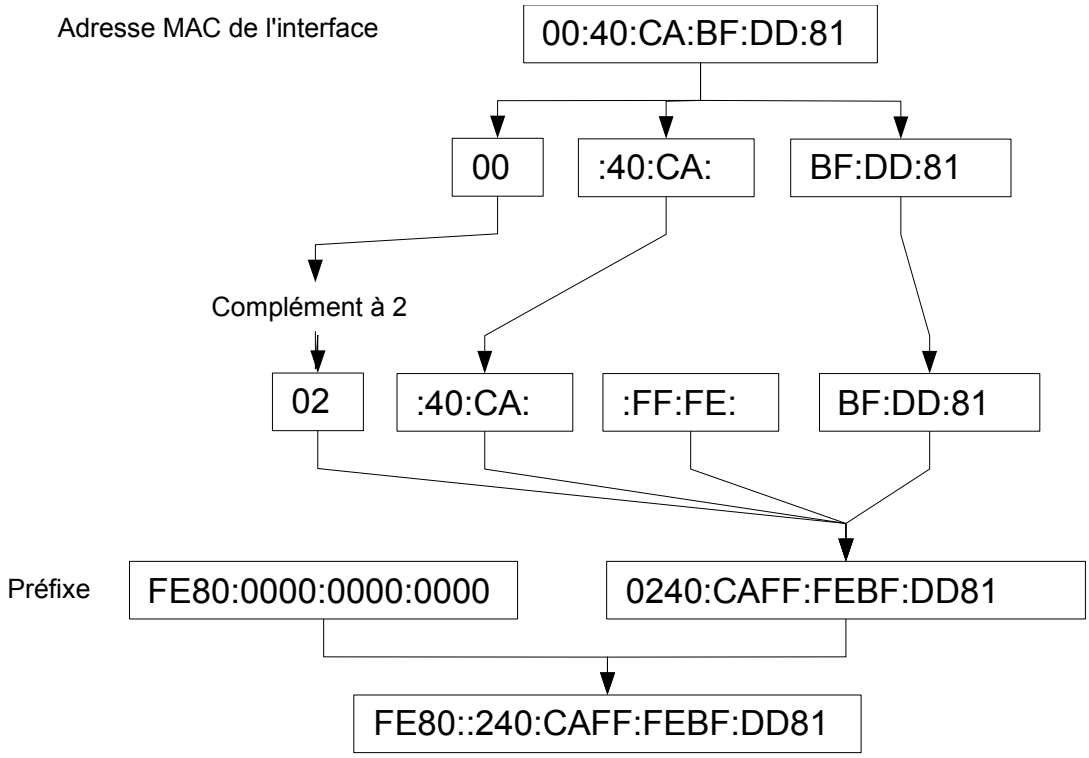



*Illustration 4: Constitution d'une adresse de lien local*

*b - L'adresse de site local*

Cette adresse est l'adresse qui sera valide à l'intérieur de tout le réseau d'un site, par exemple, sur tout le réseau d'une entreprise. La taille des adresses IPv6 permettent également de créer des sous-réseaux hiérarchisés.

Si le site n'est pas relié à Internet, il pourra utiliser des adresses de réseau privé dont le préfixe sera FEC0::/10.

## 1.4 - Le protocole ICMPv6

Le protocole ICMPv6 est la nouvelle version du protocole ICMP adapté à IPv6. Cette nouvelle version apporte une réorganisation au niveau des types de message. Il garde cependant les même fonctionnalités de base que la version pour IPv4 auxquelles s'ajoutent les fonctions autrefois déléguées aux protocoles IGMP et ARP qui deviennent donc obsolètes.

Un certain nombre d'autres fonctionnalités nouvelles sont ajoutées. La plus importante d'entre elle est le système d'autoconfiguration du réseau apportée par IPv6 (voir la section 1.6 - L'autoconfiguration du réseau, page 17).

## 1.5 - La résolution des adresse physiques

La résolution des adresses physiques passe par l'utilisation du protocole ICMPv6 décrit plus

haut. Ce système est décrit dans la RFC 2461.

L'équipement désirant connaître l'adresse physique d'un autre équipement enverra un paquet ayant comme adresse de destination une adresse de multicast sollicité. Cette adresse a pour entête le préfixe d'une adresse multicast de lien local désignant soit tous les noeuds, soit tous les routeurs. A cet entête est accolé le motif "FF00" et l'entête occupe les 104 premiers bits de l'adresse. Les 24 derniers bits correspondent aux 24 derniers bits de l'adresse IPv6 de l'équipement à joindre.

```
FF02::1:FF00:[24 derniers bits de l'adresse IP de l'équipement]
```

L'autre équipement renverra un paquet ICMPv6 contenant les renseignements demandés.

## 1.6 - L'autoconfiguration du réseau

Le protocole IPv6 met à disposition un système d'auto-configuration du réseau qui permet à chaque machine de se construire elle même une adresse qui sera valide sur le réseau auquel elle est reliée.

Le routeur joue un rôle important dans ce système d'auto-configuration qui s'effectue en quatre phases :

1. Découverte des routeurs;
2. Découverte des préfixes;
3. Détection des adresses dupliquées;
4. Découverte des paramètres.

Cela s'effectue en utilisant le protocole ICMPv6.

Tout d'abord, la découverte des routeurs permet aux équipements de trouver tout les routeurs connectés sur le même lien physique. Au démarrage, l'équipement envoie sur le réseau un message de sollicitation de routeur sur l'adresse multicast réservée ff02::2. Le routeur (si il y en a un) renvoie une trame dans laquelle il informe l'équipement de sa présence et du préfixe du réseau. L'équipement se compose alors une adresse valide sur le réseau de la même manière que pour l'adresse de lien local en remplaçant le "FE80" par le préfixe annoncé par le routeur.

Ce système, bien que fonctionnel est actuellement toujours en développement. En effet, il est envisagé d'étendre ce principe à la découverte du service DNS entre autres.

## 1.7 - Les entêtes d'extension

Le protocole a été fait pour être facilement extensible. Derrière l'entête de base peut se trouver directement l'entête du protocole de couche supérieure (TCP par exemple) mais aussi une extension de couche 3.

# 2 - Comparaison entre IPv4 et IPv6

Du fait que le protocole IPv6 apporte un nombre conséquent de nouveauté et qu'il est toujours en développement, une comparaison exhaustive entre les deux protocoles ne peut être envisagée. Cette section a pour but de mettre en avant les changements qui seront les plus évidents pour les administrateurs réseaux.

## 2.1 - Différences au niveau des trames

La première chose que nous remarquons dans l'étude de la trame IPv6 concerne la taille des adresses IP. En effet, celles-ci sont désormais codées sur 128 bits contre 32 en IPv4. Cela permet donc d'allouer en théorie 340 282 366 920 938 463 374 607 431 768 211 456 (soit 2128, environ 3,4×1038) adresses contre 4 294 967 296 (232, environ 4×106) pour IPv4.

Cette très large plage d'adresse permet de résoudre le problème énoncé dans l'historique, à savoir une pénurie d'adresses IP même si on peut se demander pourquoi avoir pris un chiffre aussi grand (on peut ainsi attribuer 1024 adresses IP par mètre carré sur toute la surface du globe).

La deuxième chose que l'on constate est que le nombre de champs obligatoires dans l'entête de la trame a diminué : 6 dans IPv6 contre 13 dans IPv4. Certains de ces champs ont été remplacés par d'autres, ont été rendu optionnels voire supprimés. Cela permet un allègement de charge pour les routeurs.

Enfin, la disparition du champ « total de contrôle d'en tête » est également un fait notoire. En effet, ce système de contrôle d'erreur a fini par devenir inutile avec l'évolution du matériel, qui devient de plus en plus fiable mais également parce qu'il est redondant, en effet, les protocoles des couches supérieures (comme TCP) et inférieures (comme ethernet) incluent eux aussi un système permettant de contrôles l'ensemble du datagramme.

## 2.2 - La fragmentation

La fragmentation consiste à découper un datagramme en datagrammes de plus petite taille. Ceci est dû aux différentes technologies employées par les réseaux, en effet, un réseau ethernet et un réseau token ring par exemple ne transmettent pas des paquets de même taille. Cette taille est appelée MTU (Maximum Transfert Unit). Aussi, si un paquet doit passer sur un réseau dont le MTU est inférieur au réseau qu'il quitte, il doit être découpé de manière à pouvoir transiter sur ce réseau.

En IPv4, ce découpage est effectué par les éléments d'interconnexion tel que les routeurs ou les ponts. Cela impose une charge de travail considérable à cet équipement aussi, en IPv6, cette fragmentation est réalisée directement par l'émetteur qui doit découvrir par lui même quel sera le plus petit MTU jusqu'au destinataire. IPv6 nécessite pour fonctionner correctement que la technologie employée pour la transmission garantisse une taille minimale de MTU de 1500 octets.

# 3 - Compatibilité entre IPv4 et IPv6

Afin de garantir une transition le plus en douceur possible, IPv6 permet une certaine compatibilité avec IPv4. Un certain nombre de mécanismes sont chargés d'effectuer des opérations de "conversion" entre les adresses sur 128bits d'IPv6 et celles d'IPV4.

## 3.1 - Double pile

Ce principe consiste à avoir un système capable de gérer en même temps les deux protocoles simultanément. Aujourd'hui, les systèmes d'exploitation modernes et un certain nombre d'équipements sont capable de faire cela.

Les deux piles cohabitent en même temps et indépendamment sur le système et en fonction du service demandé, le système va utiliser la pile IPv4 ou la pile IPv6 indifféremment. Ce système implique d'avoir les deux piles configurées correctement et est entièrement transparent pour l'utilisateur.

Cette solution est actuellement la meilleure et doit être utilisée en priorité, les autres ne sont que des pis-aller temporaires dont l'utilisation ne doit être faite qu'en cas d'impossibilité absolue d'utiliser de l'IPv6 natif.

### *3.1.1 - Adresses IPv4 mappées*

Une machine IPv6 est cappable de communiquer avec une machine IPv6 aussi bien qu'avec une machine IPv6 en utilisant des adresses IPv4 mappées. Il s'agit seulement d'une représentation des adresses IPv4 dans IPv6. En réalité, ces adresses n'apparaissent jamais sur le réseau.

Ces adresses sont de la forme ::FFFF:a.b.c.d, par exemple, ::FFFF:192.168.216.1.

### *3.1.2 - Adresses IPv4 compatibles*

Une machine IPv6 communiquant avec une autre machine IPv6 via un tunnel automatique IPv6 sur IPv4 peut utiliser des adresses IPv4 compatibles IPv6.

Ces adresses sont notées la manière suivante : ::a.b.c.d, par exemple : ::192.168.1.4

## 3.2 - Les tunnels IPv6 dans IPv4

Cette technologie consiste à créer des "tunnels". Cela signifie que lors d'un transfert de trames IPv6 entre deux équipements, celles-ci seront encapsulées à l'intérieur d'une trame IPv4. Il existe deux types de tunnels : les tunnels statiques et les tunnels automatiques.

### *3.2.1 - Les tunnels statiques*

Les tunnels statiques fonctionnent suivant le même principe que les réseaux privés virtuels (VPN). Cette technique impose de connaître les adresses IPv4 de chaque extrémité du tunnel.

L'utilisation de tunnels ne permet pas de profiter de toutes les innovations du protocole IPv6. L'intérêt de cette technologie est de pouvoir faire communiquer entre eux deux sites IPv6 distants à travers un réseau ne supportant que le protocole IPv4.

### *3.2.2 - Les tunnels dynamiques*

Les tunnels dynamiques permettent la création automatique de tunnels IPv6 sur IPv4 en cas d'absence de connectivité IPv6 native. Il existe divers protocoles pour réaliser cette opération

*a - Le 6to4*

Ce système est décrit dans la RFC n° 3056 et 3068 et est dédié à la communication entre

deux sites IPv6 distants via un réseau IPv4.

La présence d'un routeur est nécessaire. Ce routeur attribuera aux équipements du réseau des adresse IPv6 constituée à partir de son adresse IPv4 publique. Par exemple, si l'adresse publique du routeur est 23.54.65.123 et le préfixe du réseau IPv6 est FEC0, les opérations suivantes seront réalisées :

Transformation de l'adresse IPv4 en notation hexadécimale (17.36.41.7B) puis ajout de cette adresse au préfixe IPv6 (FEC0:1736:417B). A cette adresse de réseau sera ajoutée les identifiants de chaque équipement.

La trame sera ensuite encapsulée dans une trame IPv4 puis transmise au routeur 6to4 distant.

*b - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*

Ce système est conçu pour permettre à un équipement IPv6 isolé dans un réseau IPv4 de communiquer avec un réseau IPv6 distant via une passerelle supportant IPv6.

Une adresse ISATAP sera composé ainsi :

Préfix du réseau distant : 0000:5EFE : adresse IPv4 en décimal de l'équipement.

Après encapsulation dans une trame IPv4, les données seront envoyées vers la passerelle qui les retransmettra à leur destinataire.

*c - Toredo*

Cette technologie a été développée par Microsoft pour palier aux défaut de 6to4. Cette technologie n'étant disponible à ma connaissance que sous Windows et de plus propriétaire et non normalisée, il n'en sera pas question ici.

# III - Présentation avancée d'IPv6

# 1 - Domain Name System et IPv6

Références des RFC : 1886.

Le protocole IPv6 apporte un certain nombre de changement quant au fonctionnement du système de nom de domaine. En effet, il faut désormais prendre en compte la taille nouvelle des adresses. Cependant, la configuration n'est pas tellement différente que pour l'IPv4.

Les extensions au système DNS apportées par IPv6 sont décrites dans les RFC 1886, 2874 et 3363.

Un exemple de mise en oeuvre d'un serveur DNS avec Bind est fourni en page 37.

## 1.1 - Enregistrements DNS directs

IPv6 utilise des enregistrements de type "AAAA". Aussi, un enregistrement IPv6 complet de base pourra ressembler à celà :

```
www6.ma-compagnie.fr.          IN          AAAA        fec0:1::1
```

Il est intéressant de noter que l'on peut sans problème mettre dans le même fichier des enregistrements IPv4 et IPv6.

Il n'est pas recommandé d'utiliser des enregistrements de type "AAAA" pour les adresses IPv4 mappées (::FFFF:v.x.y.z) mais d'utiliser des adresses de type A.

Certaines documentations anciennes peuvent encore évoquer l'utilisation d'enregistrements "A6". Ces enregistrements, définis dans la RFC 1886 ont été dépréciés par la RFC 3363.

## 1.2 - Enregistrements DNS inverses

Pour les enregistrements inverses, IPv6 utilise le domaine ip6.arpa. Aussi, un enregistrement de base pour une machine d'adresse fec0:1::240:caff:febf:dd81/64 ressemblera à ceci :

```
1.8.d.d.f.b.e.f.f.f.a.c.0.4.2.0.0.0.0.0.0.0.0.1.0.0.0.c.e.f.ip6.arpa
     IN    PTR   www6.ma-compagnie.fr
```

Effectivement, on est en droit de se dire "glops !". Il existe un certain nombre d'utilitaires permettant de générer ces lignes. De plus, une notation quelque peut simplifiée existe. Elle consiste à définir une origine correspondant à la partie commune de toutes les adresses (généralement l'adresse du réseau) et à n'enregistrer que l'identifiant unique de l'équipement, par exemple :

```
     $ORIGIN 0.0.0.0.0.0.0.0.1.0.0.0.0.c.e.f.ip6.arpa.
     1.8.d.d.f.b.e.f.f.f.a.c.0.4.2.0     IN    PTR   www6.ma-cie.fr
     2.5.d.3.f.4.e.f.f.f.b.5.1.8.2.a     IN    PTR   dns.ma-cie.fr
```

Certaines documentations anciennes peuvent encore évoquer l'utilisation du domaine inverse "ip6.int".

# 2 – Le système d'adressage en détail

Références des RFC : 1884, 2373.

La longueur des adresses utilisées en IPv6 étend grandement les possibilités de routage offertes par ce protocole et favorise donc l'émergence de nouvelles applications dont la mise en oeuvre en IPv4 n'était pas des plus aisée.

L'architecture d'adressage d'IPv6 est décrite dans la RFC 1884.

## 2.1 - Plan d'adressage public

Dans un plan d'adressage public, les adresses sont découpées en plusieurs parties :

| 3 bits | 13 bits | 13 bits | 19 bits | 16 bits | 64 bits |
|---|---|---|---|---|---|
| FP (Format Prefix) | TLA (Top Level Aggregation) | SubTLA ( sub-Top Level Aggregation) | NLA (Next Level Aggregation) | SLA (Site Level Aggregation) | Interface ID |
| Indique le type d'adresse | Identifiant attribué par l'IANA | Identifiant de délégataire régional | Identifiant du fournisseur d'accès | Identifiant de sous-réseau utilisé par le client | Identifiant du noeud sur le réseau du client |
| Zone Internet | | | | LAN | |

Ce système permet aux clients d'obtenir une adresse de réseau sur 32 bits. En respectant la répartition des bits pour le SLA et l'identifiant d'interface, il leur est donc possible de créer 65536 sous-réseaux soit l'équivalent d'un plan d'adressage utilisant des adresses IPv4 de classe A.

## 2.2 - Plan d'adressage privé

IPv6 défini deux types d'adresses utilisables dans un réseau privé. Les adresses pour une utilisation sur un lien local et les adresses pour une utilisation sur tout un site. Les adresses dédiées à une utilisation sur un réseau privé commencent par FE.

*a - Les adresses de lien local*

Ces adresses ne peuvent être utilisées que dans un même domaine de diffusion, c'est à dire qu'un routeur ne doit pas les transmettre sur un autre réseau ou sous-réseau. Elles permettent à des noeuds situés sur le même lien de communiquer directement entre eux et sont utilisées entre autre lors de l'auto-configuration avec un routeur.

Elles sont généralement générées automatiquement par les noeuds et sont composée de cette manière :

FE80::identifiant d'interface/64

*b - Les adresses de site local*

Ces adresses sont utilisables sur l'intégralité d'un réseau privé. Un routeur ne doit pas transmettre de adresses de site local sur un réseau public.

Elles peuvent être attribuées manuellement ou à l'aide d'un système d'autoconfiguration et sont composées de cette manière :

FEC0:identifiant de sous-réseau:identifiant d'interface

## 2.3 Les adresses multicast

Une adresse multicast sert à désigner un groupe de noeud sur un réseau. Un noeud peut appartenir à plusieurs groupes.

### *2.3.1 - Format des adresses multicast*

Les adresses multicast suivent le schéma indiqué sur la figure suivante :

| 8 bits | 4 bits | 4 bits | 112 bits |
|---|---|---|---|
| Toujours à : 1111 1111 | Drapeau | Portée | Identifiant du groupe |

*Illustration 5: Format d'une adresse multicast*

Les 8 premiers bits sont tous à 1 et servent à indiquer qu'il s'agit d'une adresse multicast.

Drapeau est un ensemble de 4 bits dont les trois premiers sont réservés et doivent donc être initialisés à zéro. Le dernier indique, si il est égal à 0 que l'adresse est une adresse permanente assignée par IANA. S'il est à 1, il s'agit d'une adresse temporaire.

Portée sert à indiquée la limite de zone dans laquelle l'adresse est valable. Ses valeurs actuellement attribuées sont :

- 0 : Réservé
- 1 : Limité au noeud local
- 2 : Limité au domaine de diffusion délimité par des routeurs
- 4 : Limité au site
- 8 : Limité à l'organisation propriétaire du réseau
- E : La portée est globale
- F : Réservé

### *2.3.2 - Adresses multicast prédéfinies*

La RFC 1884 réserve un ensemble d'adresses multicast pour des usages particuliers :

- Adresses servant à désigner tous les noeuds d'un réseau : FF01::1 et FF02::1
- Adresses servant à désigner tous les routeurs d'un réseau : FF01::2 et FF02::2
- Adresse servant à désigner tous les serveurs DHCP et les relais DHCP : FF02::C
- Adresse sollicitation de noeud : FF02::1:XXXX:XXXX

## 2.4 - Adresses d'un même noeud

Un noeud IPv6 est tenu de reconnaître automatiquement un certain nombre d'adresses pointant sur lui même. Ces adresses diffèrent suivant le type de noeud.

### *2.4.1 - Adresses requises pour un hôte*

Les hôtes (postes clients, serveurs, imprimantes...) sont tenu de considérer les adresses suivantes les désignant :

- Son adresse de lien local assignée à chaque interface.
- L'adresse unicast qui lui a été assignée.
- L'adresse de boucle locale.
- L'adresse de multicast désignant tous les noeuds du réseau.
- L'adresse multicast de sollicitation de noeud correspondant à chaque adresse unicast et/ou anycast lui étant assignée.
- L'adresse multicast de tous les groupes auxquels l'hôte est susceptible d'appartenir.

### *2.4.2 - Adresses requises pour un routeur*

- Son adresse de lien local assignée à chaque interface.
- L'adresse unicast qui lui a été assignée.
- L'adresse de boucle locale.
- L'adresse anycast de routeur sur chaque sous-réseau auquel il est connecté.
- Toutes les adresses anycast pour lesquelles il a été configuré.
- L'adresse de multicast désignant tous les noeuds du réseau.
- L'adresse de multicast désignant tous les routeurs du réseau.
- L'adresse multicast de sollicitation de noeud correspondant à chaque adresse unicast et/ou anycast lui étant assignée.
- L'adresse multicast de tous les groupes auxquels le routeur est susceptible d'appartenir.

# 3 - IPSec : La sécurité et IPv6

Références des RFC : 2401.

## *3.1 - Qu'est-ce qu'IPSec ?*

IPSec est un système mis au point pour le protocole IPv6 (puis adapté par la suite à IPv4) dans le but d'apporter un système de sécurité au niveau de la couche réseau du protocole IP.

Ces systèmes de sécurité ont pour rôle de garantir soit l'authenticité des données, soit leur confidentialité, soit les deux.

## *3.2 - Le système AH*

AH signifie "*Authentification Header*". Il s'agit d'un entête supplémentaire ajouté après l'entête de base IPv6 et dont le rôle est d'assurer l'authenticité des datagrammes IP.

### *3.2.1 - L'entête AH*

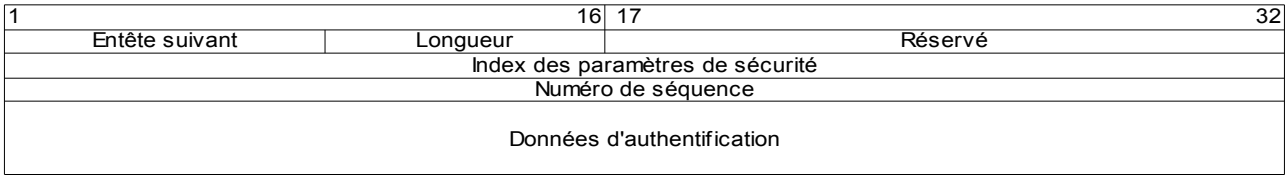



*Illustration 6: L'entête d'authentification*

- Entête suivant :
- Longueur :
- Réservé :
- Index de paramètres de sécurité :
- Numéro de séquence :
- Données d'authentification :

### *3.2.2 - Le système d'authentification*

Pour assurer l'intégrité et l'authentification des paquets, IPSec utilise un système de hachage des messages (HMAC). Pour calculer ce hachage, IPSec utilise les algorithme de hachage MD5 ou SHA, une clef secrète et le contenu du datagramme à authentifier.

## *3.3 - Le système ESP*

ESP signifie "*Encpasulating Security Payload*". C'est le système qui assurera la confidentialité des données en les cryptant.

Son principe est de prendre le datagramme de départ, de le crypter soit en entier, soit uniquement les données puis de l'encapsuler dans une trame IP classique.

### *3.3.1 - Le cryptage*

ESP peut utiliser un certain nombre d'algorithme de cryptage : NULL, DES, 3DES, AES, et Blowfish en particulier.

## *3.4 - Les modes de fonctionnement*

IPSec supporte deux modes de fonctionnement. Le mode « transport » et le mode « tunnel ».

Le mode transport a généralement pour but de sécuriser les communications entre deux hôtes. Le mode tunnel est généralement dédié à la mise en place de réseaux privés virtuels (VPN) entre deux réseaux distants. La principale différence entre les deux modes est l'adresse IP de l'équipement qui recevra le datagramme IP.

### *3.4.1 - Le mode transport*

En mode transport, les données encapsulées sont reçues par le même destinataire que le paquet d'origine.


*Illustration 7: Encapsulation en mode transport*

### *3.4.2 - Le mode tunnel*

En mode tunnel, deux équipements supplémentaires apparaissent : des passerelles. Leur rôle est d'encapsuler et de crypter l'intégralité des paquets à destination du réseau distant et de le transmettre à la passerelle d'entrée de cette autre réseau.

La passerelle d'arrivé vérifiera que le paquet provient du bon expéditeur puis elle le décryptera et le transmettra au destinataire final.

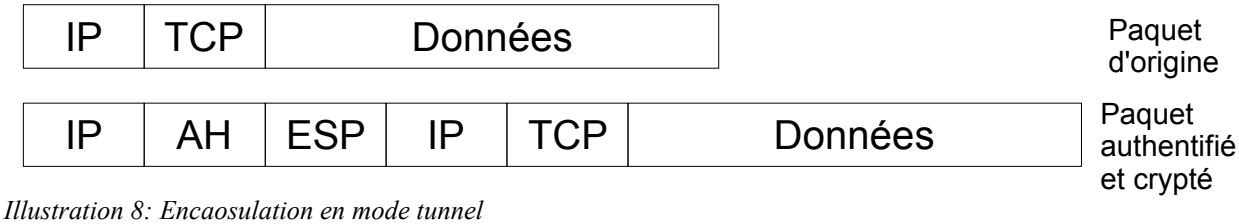



*Illustration 8: Encaosulation en mode tunnel*

# III - Mise en oeuvre

Ce document présentera un mise en oeuvre d'IPv6 dans les cas suivants :

# 1 - Mise en oeuvre sur un système Linux 2.6

Le support d'IPv6 dans le noyau Linux a été implémenté pour la première fois dans un noyau de la série 2.2. Sans que je puisse le garantir, les manipulations présentées ci-après devraient fonctionner sans problème sur un noyau 2.4 avec peut-être quelques adaptations.

## 1.1 - Ajout du support du protocole IPv6 dans le noyau

Le noyau fourni par la plupart des distributions modernes offre le support de l'IPv6. Pour vous en assurer, il suffit, en tant que super-utilisateur (root) de saisir les commandes suivantes :

```
modprobe ipv6
ifconfig | grep inet6
```

Si la deuxième commande affiche quelque chose, c'est que le support est activé, sinon, il faudra recompiler le noyau.

### *1.1.1 - Options du noyau*

Sur un noyau 2.6, les options relatives à IPv6 se trouvent dans :

```
Device Drivers --->
      Networking Support --->
            Networking Options --->
```

Les options à activer sont :

| | |
|---|---|
| The IPv6 Protocol (EXPERIMENTAL) | Activation de la pile IPv6 |
| IPv6: Privacy Extensions (RFC 3041) support | Activation du système d'autoconfiguration |
| IPv6: AH transformation | Support pour l'authentification IPSec |
| IPv6: ESP transformation | Support pour le cryptage IPSec |
| IPv6: IPComp transformation | Support pour la compression des données pour IPSec |
| IPv6: tunnel transformation | Support pour la création de tunnels IPv6 dans IPv6 |
| IPv6: IPv6-in-IPv6 tunnel | Support pour la création de tunnels IPv6 dans IPv6 |

Pour l'activation de la version IPv6 de NetFilter (le firewall de Linux), il faut activer les options qui se trouvent dans :

```
Device Drivers --->
      Networking Support --->
            Networking Options --->
                  Network packet filtering --->
                        IPv6: Netfilter Configuration  --->
```

Une fois ces options activées (en module ou en dur), il suffit de recompiler le noyau à l'aide des commandes classiques.

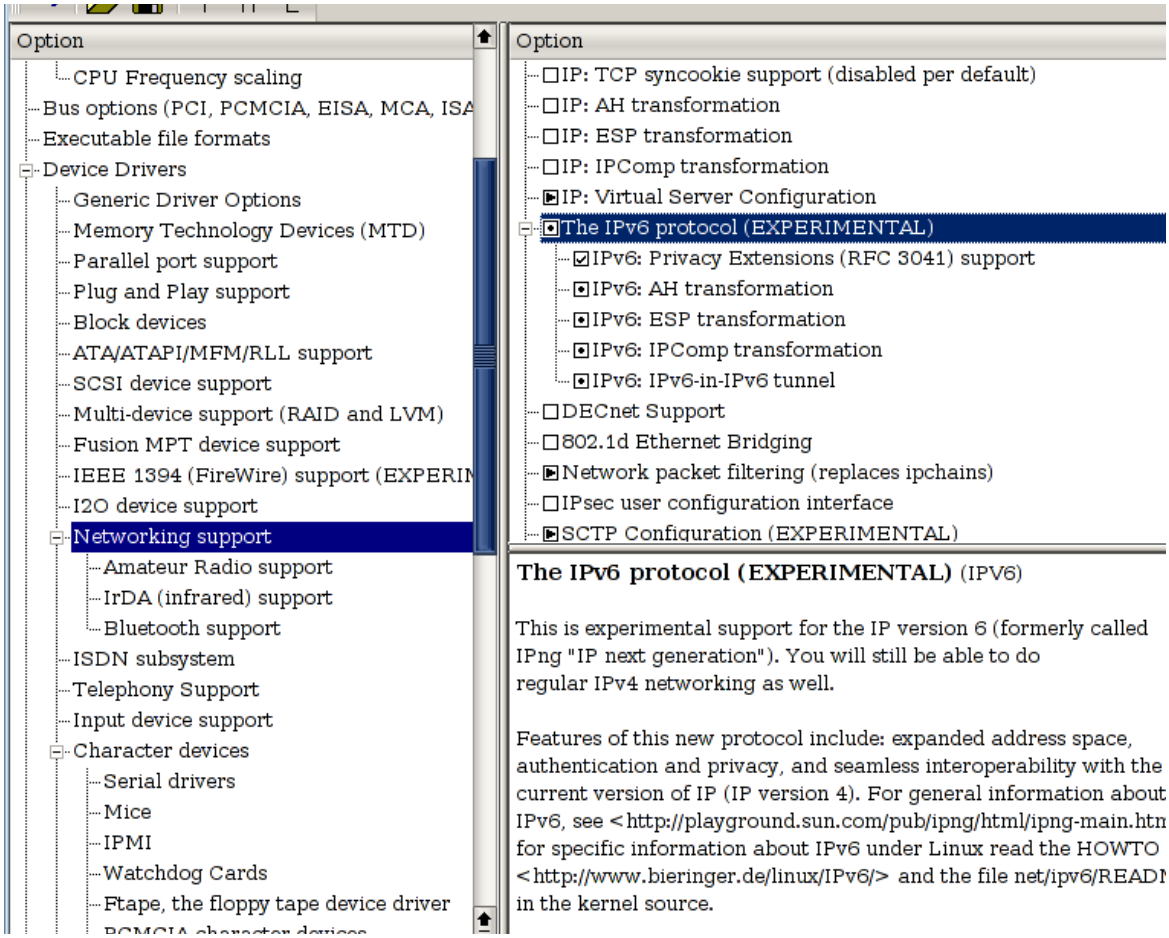


*Illustration 9: Configuration du noyau Linux*

Enfin, après redémarrage et chargement (éventuel) du module, la commande "ifconfig" permet de vérifier que le support d'IPv6 est bien chargé : Il suffit d'avoir au minimum l'adresse de portée "lien local" (celle commençant par FE80).

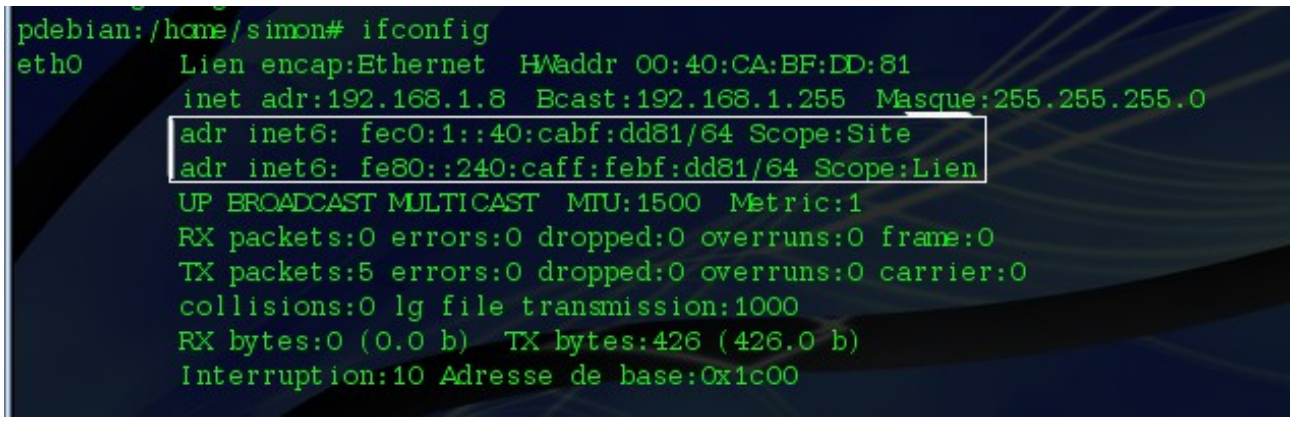


*Illustration 10: Vérification du bon fonctionnement d'IPv6 sous Linux*

## 1.2 - Configuration des paramètres IPv6

### *1.2.1 - Configuration des adresses*

La plupart des distributions proposent déjà la possibilité de configurer les adresses IPv6 dans les fichiers de configuration, pour savoir comment faire, il faudra se reporter à leurs documentations respectives.

Cependant, la procédure manuelle et universelle est présentée ci-dessous.

La configuration d'une adresse IPv6 se fait à l'aide de la commande "ifconfig" (en supposant que eth0 est l'interface à configurer) :

```
ifconfig eth0 add fec0:1::ab5:1/64
```

De même, la suppression d'une adresse IPv6 se fait avec :

```
ifconfig eth0 del fec0:1::ab5:1/64
```

### *1.2.2 - Renseignement du serveur DNS*

La configuration du serveur DNS auquel le poste s'adressera se fait de la même manière qu'en IPv4. La syntaxe du fichier /etc/resolv.conf est la même, on indique seulement une adresse IPv6 au lieu de l'adress IPv4

### *1.2.3 - Configuration du routage*

La manipulation des tables de routage se fait à l'aide de la commande "route" à laquelle il faut passer le paramètre "--inet6" en plus des autres paramètres :

*a - Affichage des routes*

L'affichage des routes disponibles se fait à l'aide de la commande suivante :

```
route -A inet6
```

*b - Manipulation des routes*

L'ajout et la suppression d'une route vers un réseau distant se fait avec les commandes suivantes :

```
route --inet6 add <réseau>/<longueurpréfixe> gw <adresseipv6>
route --inet6 del <réseau>/<longueurpréfixe> gw <adresseipv6>
```

On peut également remplacer "<réseau>/<longueurpréfixe>" par "default" si on veut ajouter une route par défaut.

Soit par exemple : (en supposant que FEC0:2::/64 est le réseau à atteindre et FEC0:1::1 est l'adresse du routeur)

```
route --inet6 add  fec0:2::/64 gw fec0:1::1
route --inet6 del  fec0:2::/64 gw fec0:1::1
```

*c - Activation du routage IPv6 sous Linux*

L'activation du routage pour IPv6 peut être fait de manière contrôlé interface réseau par interface réseau ou bien toutes les interfaces en même temps. Cela se fait avec les commandes :

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

ou

```
sysctl net.ipv6.conf.***.forwarding=1
```

Où *** correspond à l'interface à configurer (all, eth0, ...).

## 1.3 - Fonctionnalités avancées

### *1.3.1 - Mise en place de l'autoconfiguration*

Le système d'autoconfiguration d'IPv6 nécessite la présence d'un routeur. Nous allons donc configurer notre système Linux pour qu'il se comporte comme un routeur sans toutefois entrer dans une configuration avancée du routage, inutile ici.

*a - Configuration du routeur*

L'autoconfiguration des postes distants est réalisée par le programme Radvd[2] qu'il faudra donc installer. Il est fourni avec la plupart des distributions, sinon, sa compilation ne devrait pas poser de problème pour le Linuxien avancé que vous êtes.

Il faut d'abord commencer par activer le routage IPv6 au niveau du noyau comme expliqué dans la section sur le routage. Ensuite, la configuration de radvd s'effectue en éditant le fichier /etc/radvd.conf :

```
interface eth0          <--- Interface à configurer
      {
         AdvSendAdvert on;
         prefix fec0:1::/64 <-- préfixe de à annoncer réseau et masque
         {
            AdvOnLink on;
            AdvAutonomous on;
         };
      };
```

Un certain nombre d'autres options sont possible, la page de manuel de radvd contient toutes les informations nécessaires pour effectuer une configuration avancée.

Le démon pourra être lancé avec la commande :

```
/etc/init.d/radvd start
```

*b - Configuration sur les postes clients*

Aucune configuration n'est nécessaire sur les postes clients. Il suffit seulement que le

2 Router Advertisement Daemon, (http://v6web.litech.org/radvd/)

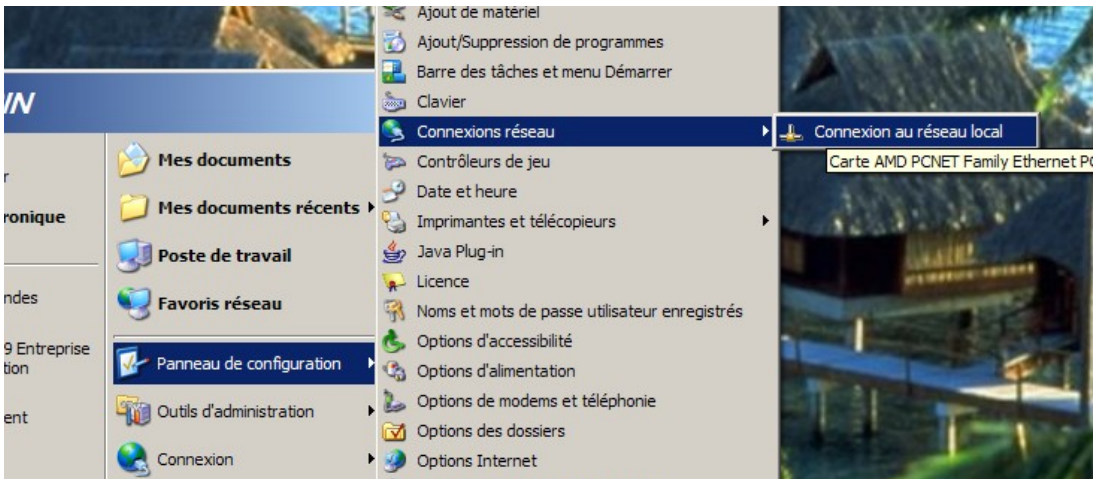
support IPv6 soit activé et le système effectuera tout seul la recherche du routeur distant..

# 2 - Mise en oeuvre sous Windows 2003 Server

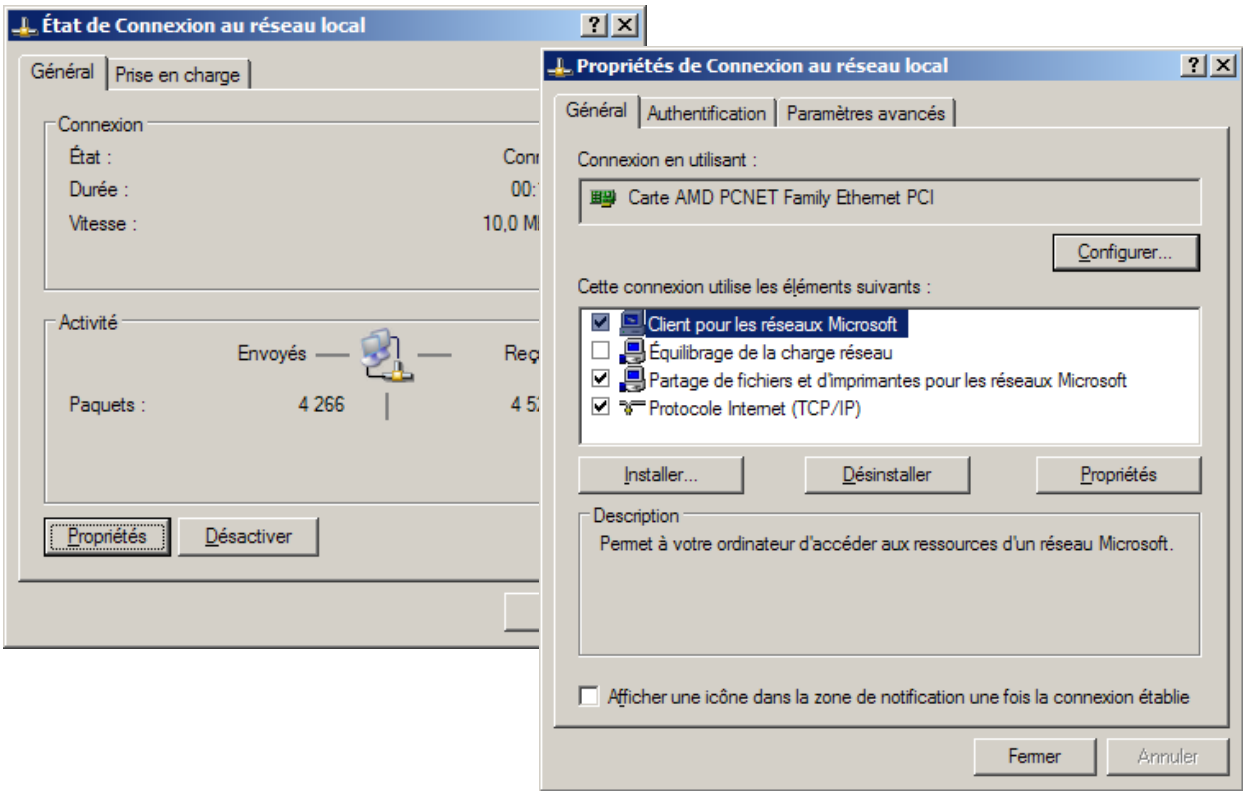
Le dernier système d'exploitation de Microsoft supporte le protocole IPv6 de manière native depuis ses versions XP-SP1 et 2003 Server.

## *2.1 - Installation du protocole*

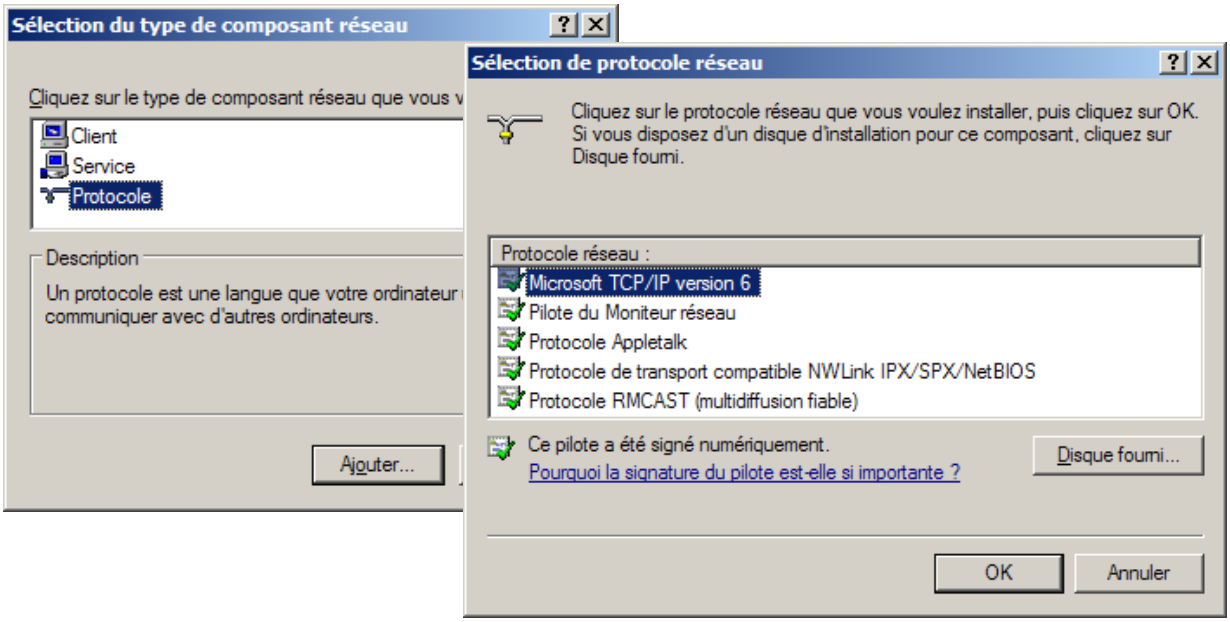
Pour activer la prise en charge de ce protocole avec Windows 2003, il faut aller dans le panneau de configuration, choisir connexions réseau puis cliquer sur la connexion réseau sur laquelle on désire installer la prise en charge du protocole IPv6 :




Une fois la connexion sélectionnée, dans la fenêtre qui apparaît, cliquer sur propriétés :



Puis dans Installer, choisir Protocole, Ajouter et enfin « Microsoft TCP/IP version 6 »




Et enfin, la commande « ipconfig » nous permet de vérifier que l'installation s'est effectuée correctement :

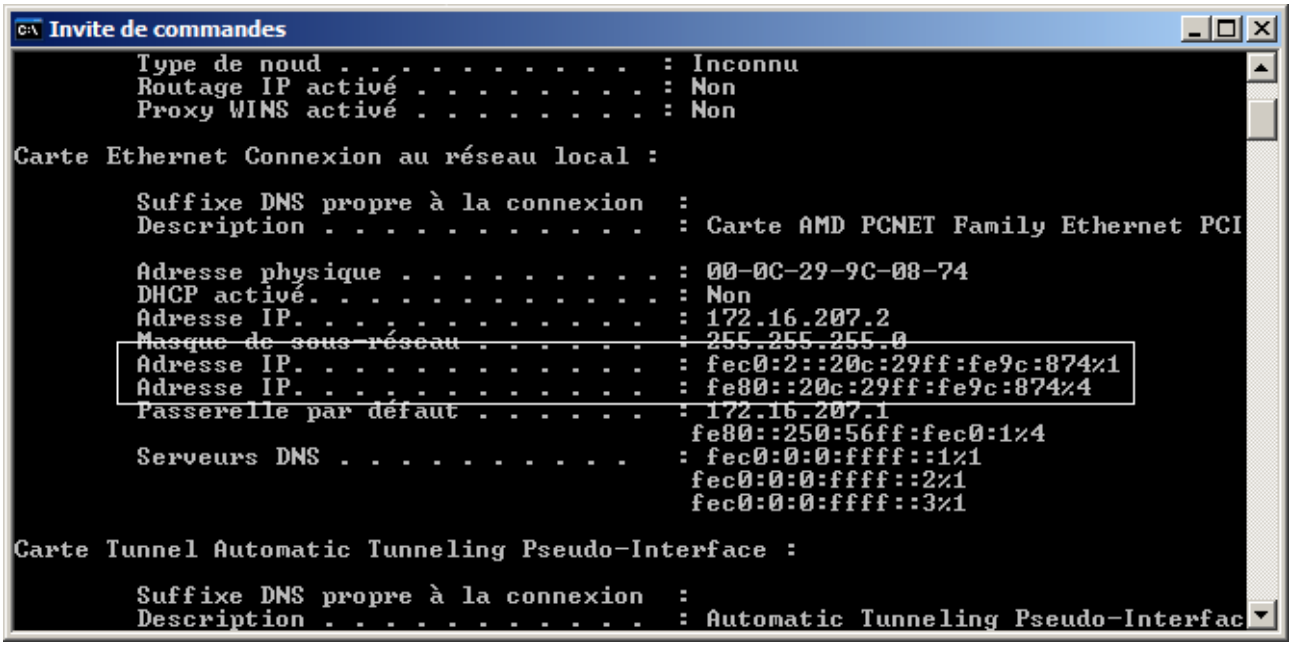



*Illustration 11: Vérification de la configuration de Windows*

## *2.2 - Configuration*

Actuellement, la configuration d'IPv6 sous Windows ne peut se faire qu'en ligne de commande à l'aide de l'utilitaire netsh. Il s'agit d'une sorte d'interface permettant d'accéder à tout la configuration réseau de Windows qui ne devrait pas trop dépayser les personnes ayant déjà manipulé du matériel Cisco.

### *2.2.1 - Configuration des adresses IP*

L'ajout / suppression d'une adresse IP se fait à l'aide de la commande :

```
netsh
interfaces
ipv6
add address <nom de la connexion> <adresse IP>
del address <nom de la connexion> <adresse IP>
```

### *2.2.2 - Configuration des DNS*

L'ajout / suppression d'un serveur DNS se fait avec les commandes :

```
netsh
interfaces
ipv6
add dns <nom de la connexion> <adresse IP du serveur DNS>
add del <nom de la connexion> <adresse IP du serveur DNS>
```

### *2.2.3 - Configuration du routage*

L'ajout et la suppression des routes se font à l'aide des commandes suivantes :

```
netsh
interfaces
ipv6
add route <préfixe du réseau distant> <nom de la connexion>
      <adresse de la passerelle>
del route <préfixe du réseau distant> <nom de la connexion>
      <adresse de la passerelle>
```

# 3 - Mise en place d'un serveur DNS avec Bind

Le serveur DNS de l'Université de Berkeley supporte pleinement le protocole IPv6 depuis déjà un certain temps. Il s'agit de l'implémentation de référence du système de nom de domaine et fonctionne sur pratiquement tous les systèmes du marché. C'est la version 9 qui servira ici pour les manipulations.

## 3.1 - Activation du support d'IPv6

Par défaut, Bind ne répond pas aux requêtes envoyées à l'aide du protocole IPv6. Notez cependant qu'il peut renvoyer des enregistrements IPv6 à un hôte ayant fait une requête en IPv4.

Il faudra donc définir les options globales suivantes en plus des options déjà existantes :

```
options {
      listen-on-v6{
            any;
      };
};
```

L'option "listen-on-v6" permet de définir sur quelle(s) interface(s) Bind acceptera les requêtes envoyées en utilisant IPv6.

## 3.2 - Création des zones IPv6

Une fois IPv6 activé, on peut créer des zones pour les domaines à servir. Nous allons donc créer une zone directe et deux zone indirectes (nous allons mélanger IPv6 et IPv4...). Le domaine utilisé comme exemple sera "ma-compagnie.fr" et nous supposerons que le réseau IPv4 utilisera des adresses de classe C : 192.168.1.0/24 et le réseau IPv6 des adresses de préfixe FEC0:1::/64.

```
// Création de la zone directe
zone "ma-compagnie.fr" {
      type master;
      file "ma-cie.fr.hosts"
}

// Création de la zone inverse IPv4
zone "0.1.168.192.in-addr.arpa"{
      type master;
      file "ma-compagnie.fr.ipv4.rev
}

// Création de la zone inverse IPv6
zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0. \
                        0.0.0.0.0.0.0.1.0.0.0.0.c.e.f.ip6.arpa"{
      type master;
      file "ma-compagnie.fr.ipv6.rev
};
```

Les entêtes des fichiers de zone auront la même allure que pour de l'IPv4. Voici le contenu de chacun d'eux :

Définition de la zone directe :

```
$ttl 38400
ma-compagnie.fr.  IN      SOA     ns.ma-compagnie.fr. root.localhost.  (
            1076953709
            10800
            3600
            604800
            38400
      )

; Indication du serveur DNS
ma-compagnie.fr.        IN    NS          ns.ma-compagnie.fr.
ns.ma-compagnie.fr.     IN    AAAA        fec0:1::1
ns.ma-compagnie.fr.     IN    A           192.168.1.1

; Indication du serveur de courriel
@                       IN    MX          mail.ma-compagnie.fr.
mail.ma-compagnie.fr.   IN    CNAME       srv-prin.ma-compagnie.fr.

; Indication du serveur Web
www.ma-compagnie.fr.    IN    CNAME       srv-prin.ma-compagnie.fr.

; Indications des postes
srv-prin.ma-compagnie.fr.     IN    AAAA  fec0:1::2
srv-prin.ma-compagnie.fr.     IN    A     192.168.1.2

poste12                 IN    AAAA  fec0:1::12
poste12                 IN    A     192.168.1.12
poste24                 IN    AAAA  fec0:1::24
poste24                 IN    A     192.168.1.24
poste122                IN    AAAA  fec0:1::122
poste122                IN    A     192.168.1.122
```

Définition de la zone inverse IPv4 :

```
$ttl 38400
0.1.168.192.in-addr.arpa.     IN    SOA   ns.ma-compagnie.fr..
root.localhost. (
      1076953773
      10800
      3600
      604800
      38400
)

$ORIGIN 1.168.192.in-addr.arpa.
0           IN          NS          ns.ma-compagnie.fr.
1           IN          PTR         ns.ma-compagnie.fr.
2           IN          PTR         srv-prin.ma-compagnie.fr.
12          IN          PTR         poste12.ma-compagnie.fr.
24          IN          PTR         poste24.ma-compagnie.fr.
122         IN          PTR         poste122.ma-compagnie.fr.
```

Définition de la zone inverse IPv6 :

```
$ttl 38400
@     IN    SOA   ns.ma-compagnie.fr. root.localhost. (
```

```
      1076953773
      10800
      3600
      604800
      38400
)
IN    NS    ns.ma-compagnie.fr.

$ORIGIN 0.0.0.0.0.0.0.0.1.0.0.0.0.c.e.f.ip6.int.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR   srv-prin.ma-compagnie.fr.

12.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR   poste12.ma-compagnie.fr.
24.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR   poste24.ma-compagnie.fr.
122.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN     PTR   poste122.ma-compagnie.fr.
```

Après démarrage, le serveur DNS est prêt à fonctionner.

# IV - LEXIQUE

**AES** (Advanced Encryption Standard) : Algorithme de cryptage.

**Bind** : Berkeley Internet Name Domain. C'est l'implémentation de référence du système de nom de domaines (voir DNS).

**Blowfish** :

**DES** (Data Encryption Standard) : Algorithme de cryptage à clef symétrique. L'utilisation de cet algorithme est déconseillée. Il est avantageusement remplaçable par AES ou 3DES.

**DNS** (Domain name system) : Système de nom de domaine en Français. C'est un système permettant d'associer un nom facile à mémoriser à une adresse IP.

**Hachage** : Un hachage est une empreinte alpha numérique obtenue à partir d'une suite quelconque de bits. Cette empreinte est censée être unique : deux suites de bits différentes ne peuvent avoir la même empreinte (ou signature).

**HMAC** :

**IANA** (Internet Assigned Numbers Authority) : Organisme chargé d'attribuer les adresses IP aux différents fournisseurs d'accès internet.

**MD5** : Algorithme de hachage.

**SHA** : Algorithme de hachage.

# V - RÉFÉRENCES ET PROGRAMMES

# 1 - Request for comments

RFC 1884 : *IP Version 6 Addressing Architecture*

RFC 1886 : *DNS Extensions to support IP version 6*

RFC 2094 : *Group Key Management Protocol (GKMP) Architecture*

RFC 2373 : *IP version 6 addressing architecture*

RFC 2401 : *Security Architecture for the Internet Protocol*

RFC 2402 : *IP Authentication Header*

RFC 2406 : *IP Encapsulating Security Payload (ESP)*

RFC 2409 : *The Internet Key Exchange*

RFC 2460 : *Internet Protocol, Version 6 (IPv6) Specification*

RFC 2461 : *Neighbor Discovery for IP Version 6*

RFC 2473 : *Generic Packet Tunneling in IPv6 Specification*

RFC 2874 : *DNS Extensions to Support IPv6 Address Aggregation and Renumbering,*

RFC 2893 : *Transition mechanism for IPv6 hosts and routers*

RFC 2928 : *Initial IPv6 Sub-TLA ID Assignment*

RFC 3041 : *Privacy extensions for stateless address*

RFC 3056 : *Connection of IPv6 Domains via IPv4 Clouds*

RFC 3068 : *An Anycast Prefix for 6to4 Relay Routers*

RFC 3315 : *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3363 : *Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)*

RFC 3513: *Internet Protocol Version 6 (IPv6) Addressing Architecture*

# 2 - Documentation

Linux + IPv6 HOWTO

BIND 9 Administrator Reference Manual

# 3 - Sites internet

# 4 - Programmes utiles

Bind

Radvd

GNU / Linux Gentoo