

IPv6, Précis et concis

Baptiste SIMON (aka BeTa)
e-glop.net

Du 23 au 25 avril 2004 aura lieu la première édition de *Libr'East of Paris*, un évènement sur trois jours, organisé par l'association IDILE, totalement dédié au logiciel Libre. Dans le cadre de ces rencontres, je vais être amené à présenter une conférence sur IPv6.

Ce document a pour but de cadrer le contenu de cette conférence en en traçant le plan et les idées directrices.

1. Introduction

1.1. D'où vient on ?

1.1.1. IPv4, les limites en terme d'adressage

- IPv4 ne permet plus d'adresser simplement et universellement tous les équipements du monde
 - NAT
 - Proxy plus nécessaires au niveau pratique qu'au niveau sécurité
- Historiquement, IPv4 vient des états-Unis d'Amerique
 - 72% des adresses IPv4 sont détenues en Amérique du Nord
 - 17% en Europe
 - 9% en Asie
 - Les miettes pour le reste du monde.

1.1.2. IPv4, les limites en terme de routage

- Impossibilité de faire de la sécurité de bout-en-bout avec des protocoles comme IPsec, le NAT rendant cela impossible/très compliqué.
- Evolution du routage dans le coeur d'Internet
 - 80.000 routes stockées par les routeurs du coeur du réseau il y a quelques années
 - 120.000 routes aujourd'hui

1.2. Comment IPv6 s'acquitte des défauts d'IPv4

1.2.1. Le routage avec IPv6

Les adresses IPv6 sont conçues pour hiérarchiser le réseau, en faire une grande arborescence.

- Gains en temps et traitement (car on ne traite plus que les préfixes des adresses IPv6)
- Gains en mémoire nécessaire (car on ne stocke plus toutes les routes anarchiquement comme en IPv4)

1.2.2. L'adressage avec IPv6

C'est l'évolution majeure (la plus "visible") du protocole.

1.2.2.1. L'adressage, vu de loin

- 128 bits, entre 1.564 et 3.911.873.538.269.506.102 (3×10^{15}) adresses par mètre carré de surface terrestre
- Adressage conçu de manière hiérarchique (découpage "géographique" clair via les préfixes).

1.2.2.2. L'adressage, vu de près

- Adresses unicast définies comme commençant par les 3 bits 001
 - 2000::/3
 - 2000:0000:0000:0000:0000:0000:0000:0000/3
- L'international : le TLA
 - Top Level Aggregator (unité d'aggregation haute)
 - 13 bits
 - ce préfixes, de 16 bits au total, sont donnés aux grands opérateurs internationaux (les RIR, Regional Internet Registries). En général, ces opérateurs ont la particularité de n'avoir aucune route par défaut (le coeur du réseau).
 - Exemple : Le RIPE-NCC (Renater, Nerim...) a, par exemple, la classe 2001:600::/23 (soit les adresses commençant par des "2001:06" ou des "2001:07")
- Le national : le sub-TLA
 - Sub-Top Level Aggregator (sous-unité d'aggregation haute)
 - 13 bits
 - Partie permettant aux unités d'aggregation haute de découper selon leurs besoins les plages d'adresses qu'ils veulent fournir aux opérateurs nationaux (par exemple).
 - Exemples : 2001:660::/35 (avec la partie réservée...) correspondant à Renater ou encore 2001:7a8::/32 pour le FAI Nerim.
- Partie réservée

- 6 bits
- Le but de cette partie est de pallier à un manque potentiel à venir de TLA ou de NLA

- La fin des opérateurs : le NLA
 - Next Level Aggregator (unité d'aggregation basse)
 - 13 bits (donc 48 bits fixes au total)
 - Partie permettant aux sous-unités d'aggregation haute de découper selon leurs besoins les plages d'adresses qu'ils veulent fournir à leurs clients.
 - Exemples : 2001:0660:7101::/48 pour l'université de Caen reliée à Renater 2001:7a8:4b09::/48 pour mon site.

- Le déploiement local : Le SLA
 - Site Level Aggregator (topologie de site)
 - 16 bits
 - Ce dernier découpage permet aux sites locaux de déployer leurs sous réseaux locaux avec 16 bits d'amplitude (soit 2^{16} , 65.536 sous réseaux).
 - Exemple : 2001:7a8:4b09:1::/64 est un des sous réseaux de mon site.

- Les machines...
 - Identifiant d'interface (découlant de l'*EUI-64*)
 - 64 bits
 - Ce suffixe permet à chaque machine d'être, en bout de chaîne, connues potentiellement de tous sur le réseau.
 - Exemple : 2001:7a8:4b09:1::80 pour le serveur web principal de mon site, 2001:7a8:4b09:1:230:1bff:feb1:defa pour ma machine personnelle.

1.2.2.3. Construction du suffixe EUI-64

Avec des interfaces possédant des adresses MAC IEEE 802.3 :

- 24 bits d'identifiant constructeur (ex: 0x02301B)
- 16 bits de bourrage (0xFFFFE (sans raison))
- 24 bits d'identifiant d'interface (ex: 0xB1DEFA)
- Cela donne donc au final un EUI-64 de type 0230:1bff:feb1:defa

Avec des interfaces non universelles (PPP par exemple) :

- Si l'identifiant d'interface est unique sur le réseau physique, on *bourre* l'*EUI-64* de 0 à gauche des bits de cet identifiant, jusqu'à avoir 64 bits au total.

- Si l'interface ne possède pas d'adresse physique (comme les interfaces PPP), il est conseillé de prendre l'adresse MAC d'une autre interface, si possible. Sinon, une génération aléatoire reste possible (les conflits IPv6 se détectant tous seuls)

1.2.3. En plus...

Avec IPv6, on voit arriver l'intégration de diverses fonctionnalités, telles que la sécurité ou la mobilité, dont on reparlera un peu plus tard.

On voit aussi intégrés des principes de *QoS* empruntés entre autre à ATM, via le champ d'en-tête IPv6 *Traffic Class*. Cela permet donc de faire de la qualité de service (théoriquement) de "bout en bout", atout très intéressant pour la visio-conférence et la VoIP par exemple...

2. Aspects Techniques d'IPv6

2.1. IPv6, au premiers abords

2.1.1. La trame IPv6

Le format d'en-tête d'une trame IPv6 est défini dans la RFC 2460 (<http://www.faqs.org/rfcs/rfc2460.html>) tel que suit :

- *Version* : identique à IPv4, spécifie sur 4 bits la version du protocole courant. Ici, ce champ est égal à 6 (0110).
- *Traffic Class* : sur 8 bits, remplit le même rôle que le champ TOS en IPv4. Il identifie le type de contenu encapsulé dans la trame IPv6, afin de permettre des traitements particuliers, entre autre durant le routage de celle-ci.
- *Flow Label* : il pousse plus loin les fonctionnalités du champ précédent. Il s'agit ici de faire optionnellement de la *QoS*, même si, pour le moment, il n'existe que des *drafts* concernant cette fonctionnalité. Ce champ fait 20 bits, et il est, pour le moment, fortement inspiré des options de QoS que l'on trouve avec ATM.
- *Payload Length* : sur 16 bits, ce champ permet de définir la taille de la trame IPv6, en octet, afin de pouvoir en définir la fin. Pour plus de détails, notons que la taille indiquée prend en compte tous les headers à partir de *Payload Length* (exclu), ainsi que tout le corps de la trame. Le reste n'est pas compté (puisque toujours identique).
- *Next Header* : spécifie sur 8 bits le premier entête suivant contenu dans les données véhiculées par la trame IPv6. On trouve ici l'une des particularités les plus intéressantes d'IPv6 : a possibilité de définir des extensions. En effet, de manière "classique", le *Next Header* sera un en-tête TCP par exemple, mais il peut aussi s'agir d'en-têtes IPv6 supplémentaires/optionnels.
- *Hop Limit* : donne le nombre de sauts maximum que peut faire une trame IPv6, depuis sa source jusqu'à sa destination. Ce champ a une valeur maximale de 255, de par sa taille de 8 bits. Équivalent du champ *TTL* en IPv4.

- *Source Address* : indique l'adresse ayant émis la trame. Il s'agit bien entendu ici d'une adresse IPv6 classique, codée sur 128 bits.
- *Destination Address* : cf. *Source Address*.

2.1.2. La notion de portée

Avec IPv4, il existe des adresses *réservées*, définies par la RFC 1918, et des adresses dites *publiques*.

Avec IPv6 arrive notion de *scope*

4 portées différentes :

2.1.2.1. global

- Adressage commun à toutes les machines
- Pour l'unicast, ce sont les adresses préfixées par `2000::/3`
- Routage sans restriction de portée de tous les datagrammes dont les adresses ont une portée globale.
- Exemple : `2001:7a8:4b09:1:230:1bff:feb1:defa`

2.1.2.2. site-local (optionnel)

- Adressage commun aux machines d'un même site
- Adresses préfixées par `fec0::/48`
- Routage des datagrammes dont les adresses ont une portée de lien-local uniquement sur les interfaces liées au réseau physique du site local
- Exemple : `fec0::1:230:1bff:feb1:defa`

2.1.2.3. lien-local

- Adressage commun aux machines d'un même lien physique
- Adresses préfixées par `fe80::/64`
- Aucun routage de ces paquets au niveau réseau (couche 3).
 - Seuls les équipements directement liés entre eux par la couche de liaison de données (couche 2) peuvent utiliser ces adresses pour communiquer entre eux.
 - Si jamais un équipement terminal reçoit un datagramme sur son adresse de lien-local et que le champ *nombre de sauts* a été décrémenté (255 au départ ici), le datagramme est rejeté : il est "passé par" un routeur, et ne provient donc pas du lien-local.
- Exemple : `fe80::230:1bff:feb1:defa`

2.1.2.4. Host

- Toute machine doit avoir une interface dite de *loopback*
- Équivalent IPv4 : 127.0.0.1/32
- Une adresse unique en IPv6 : ::1/128

2.1.3. Autoconfiguration sans état

L'autoconfiguration IPv6 est l'une de ses pièces maîtresses. Elle permet :

- La découverte des préfixes d'adressage et paramétrage du suffixe *EUI-64*
 - Vérification de l'inexistence de conflits d'adresses sur le réseau
 - Configuration des adresses de lien-local, site-local (si existant) et globales
- L'autoconfiguration du routage (*ICMP Router Discovery*)
- La découverte des paramètres du lien physique
 - Taille du MTU (*Maximum Transfert Unit*)
 - Nombre maximal de sauts autorisé
- La possibilité de déployer une infrastructure de mobilité IPv6 efficace

Cependant l'autoconfiguration a aussi ses limites. Par exemple : l'absence d'autoconfiguration de la sécurité ou encore la difficulté à mettre en place des mécanismes de DNS dynamiques.

2.2. IPv6, plus en profondeur

2.2.1. Les autres fonctionnalités d'IPv6

2.2.1.1. Mobilité

La mobilité dans IPv6 est encore un point sensible. En effet, il manque encore de spécifications définitives permettant une interopérabilité optimale entre les divers systèmes déployés.

2.2.1.1.1. Théorie

- Explication de ce que sont les principes de *macro-mobilité* et de *roaming*
 - Le *roaming*, c'est ce qui existe pour les téléphones cellulaires : la possibilité de passer d'un relai à l'autre de manière souple et progressive. Autrement dit, avertir ses correspondants du changement en cours de *point d'accès*.
 - La *macro-mobilité*, c'est la possibilité de changer de réseau IP de manière complètement transparente à l'utilisation. Cela met donc en oeuvre les concepts, plus généraux, de *roaming*.

- Mise en oeuvre théorique dans IPv6
 - Notions
 - Réseau mère
 - Adresse mère et adresses temporaires
 - Mise à jour des associations et durée de vie des adresses
- Fonctionnement global
 - Tunnels IPv6-IPv6 depuis le réseau mère
 - Utilisation des adresses temporaires

2.2.1.1.2. Pratique

- Délicat : implémentations n'existant que sous *Windows*, *BSD* (KAME (<http://www.kame.net/>)) et *Linux* (MIPL (<http://www.mobile-ipv6.org/>))
- Implémentations trop souvent in-interopérables, à cause d'un manque de spécifications complètes et officielles.

2.2.1.2. Sécurité

La sécurité dans IPv6 s'apparente, en terme d'implémentation, à l'IPsec d'IPv4. Comme pour IPv4, IPsec souffre de plusieurs points faibles :

- Pas de protection des échanges multicast
- Pas d'infrastructure PKI mondiale
- Pas de réelle autoconfiguration (il existe le principe de DNSSEC...)

Cependant, il existe certaines particularités qui rendent la sécurisation d'IPv6 très intéressante :

- Pas de NAT, d'où la possibilité de sécuriser les communications IPv6 de bout-en-bout.
- Intégration d'IPsec à la manière d'une extension d'IPv6 (en-tête *Next Header*) au lieu d'une surcouche (comme pour IPv4).
- L'obligation (de par les spécifications du protocole) d'implémenter IPsec dans IPv6, permettant ainsi de garantir que chaque équipement connecté peut échanger des données de manière sécurisée par dessus IPv6.

2.2.2. Autres usages d'IPv6

2.2.2.1. Multicast

- Historiquement, avec IPv4
 - Existe depuis longtemps
 - Peu répandu pour les utilisateurs Lambda
 - "Classe D"

- Aujourd'hui, avec IPv6
 - Préfixe `ff00::/8` (`ff01::`, `ff02::`, ...)
 - Les 4 bits suivants sont tous à 0 pour les adresses multicast temporaires, et le dernier à 1 pour les adresses permanentes (délivrées par les autorités du réseau). En général, nous obtenons donc `ff00::/12` (temporaire).
 - Les 4 bits suivants définissent la notion de *scope* vue pour l'unicast. (ex: **2** pour le lien-local, **5** pour le site-local, **e** pour les adresses globales.)

- intérêts majeurs (comme pour IPv4) sont :
 - Exemple théorique : mise à disposition de serveurs **de type** NTP stratum 1 pour tous (car NTP ne pourrait pas fonctionner comme ça de ce que je connais)
 - Diffusion d'informations en continu (streams, ex: webradios, télévisions...)
 - Visio-conférences, VoIP...
 - Jeux en réseau, travail collaboratif...

- Atouts majeurs par rapport au multicast IPv4 :
 - Existence d'IPv6-multicast depuis les origines d'IPv6
 - Possibilité de prévoir l'IPv6-multicast dans tous les éléments du réseau
 - Utilisation d'IPv6-multicast de bout-en-bout, jusque dans chaque foyer

- La petite nouveauté : les clients DHCPv6 (*autoconfiguration statefull*, perdant beaucoup de son sens avec IPv6, de part ses objectifs et sa conception) utilisent IPv6-multicast pour trouver les serveurs DHCPv6, le broadcast n'existant plus.

2.2.2.2. Anycast

- Encore au stade de la recherche
- Principes :
 - Plusieurs machine possédant la même IPv6
 - On accède plus à un service qu'à une machine
 - On accède à la première machine que l'on est à même de "rencontrer" sur le réseau (gain en trafic et en répartition géographique).
 - Petit exemple avec un réseau local (préfixe virtuel...)

- Extrêmement intéressant pour des services comme :
 - les DNS root
 - les sites web les plus fréquentés de par le monde
 - tous les services dont la réplication est possible et dont la répartition géographique peut être très importante.

3. Concrètement

3.1. Aujourd'hui, le net : état de l'art

3.1.1. 6Bone / M6Bone

- 6Bone
 - C'est le penchant IPv6 expérimental du BackBone IPv4
 - Il existe depuis l'été 1996 (son penchant Européen, le G6Bone, est arrivé un peu avant, au printemps 1996)
 - Sa fusion avec le BackBone est prévue pour le 6/6/6 (6 juin 2006)
- M6Bone
 - C'est le penchant IPv6 du MBone IPv4
 - Le M6Bone arrive en octobre 2001
 - Il est encore trop peu répandu à mon goût (tout comme IPv6 dans sa globalité) : ça peut devenir frein au développement d'IPv6, en enlevant de l'intérêt.

3.1.2. Providers internationaux et FAI "de bout de chaîne"

- Le sujet des providers internationaux est plus du ressort de Tayeb Ben Meriem.
- Les FAI français :
 - Nerim
 - Gitoyen
 - :c(

3.2. Mettre en oeuvre IPv6

3.2.1. Windows, BSD

3.2.1.1. Windows

Non testé.

Description :

- Pile TCP/IPv6 existant pour NT4, 2000 et XP (en théorie, pour 2003 Server également)
- Utilisation sommaire mais simple
- Très peu de fonctionnalités de routage

Déploiement :

- Sous 2000, il "suffit" d'exécuter un fichier téléchargé à l'adresse suivante (<http://www.microsoft.com/WINDOWS2000/technologies/communications/ipv6/>). Sous XP, la commande "ipv6 install" doit suffire pour installer sa pile IPv6.

Applications :

- ipv6.exe (affiche des informations sur la pile IPv6)
- ping6.exe
- tracert6.exe (traceroute)
- ftp
- ttcp (Test TCP)
- telnet
- Internet Explorer (de mémoire, seule la version 5.5 est IPv6 compliant)
- xchat (version Win32)
- Il doit en exister d'autres...

3.2.1.2. FreeBSD

Non approfondi.

Description :

- Support natif d'IPv6 (projet *KAME*).

Déploiement :

- Soit exécuter `/stand/sysinstall`, et répondre "oui" au support d'IPv6, soit modifier le fichier `/etc/rc.conf` avec `ipv6_enable=YES`. Puis rebooter le système.

- Pour fixer des IPv6, il faut rajouter des lignes comme celle qui suit :

```
ipv6_ifconfig_fxp0_alias0="2001:7a8:2001:7a8:4b09:3::21"
```
- ... (rtadvd_enable=YES, ipv6_gateway_enable=YES, ...)

Support :

- La plupart des applications incluses dans *FreeBSD* sont IPv6 compliant

3.2.1.3. NetBSD/OpenBSD

Non approfondi.

Description :

- Support natif d'IPv6 depuis *NetBSD* version 1.5 et *OpenBSD* version 2.5 (projet *KAME*).

Déploiement :

- Créer un fichier du type `/etc/ifconfig.interface` où *interface* est l'interface à configurer... Y ajouter, par exemple, la ligne suivante :

```
inet6 2001:7a8:2001:7a8:4b09:3::53 prefixlen 64
```

- Renseigner l'option `ip6mode` dans `/etc/rc.conf` avec les valeurs suivantes :
 - `router` pour faire de cette machine un routeur IPv6
 - `autohost` pour en faire un hôte autoconfiguré
 - `host` pour en faire un hôte non autoconfiguré
- ... (rtadvd=YES, ...)

Support :

- La plupart des applications incluses dans *NetBSD* et *OpenBSD* supportent IPv6
- Firewalling statefull
- A noter la présence de *NFS* et *RPC* IPv6 compliant
- Pour *NetBSD*, noter aussi l'absence de *Bind9* (donc pas de *Bind* supportant IPv6), et l'absence du support IPv6 dans *Postfix*.

3.2.2. GNU/Linux (plus approfondi)

3.2.2.1. From Scratch

Description :

- Support partiel d'IPv6 depuis les noyaux 2.2.x
- Support natif d'IPsec (ESP & AH) pour IPv6 depuis les noyaux 2.6.x
- Support d'IPv6-mobility via le projet MIPL (<http://www.mobile-ipv6.org/>) (patches à appliquer à certains noyaux uniquement)
- Existence d'extensions aux noyaux et aux outils associés à travers le projet USAGI (<http://www.linux-ipv6.org/>)

Déploiement :

- Installer le paquet `iproute`, pour avoir, entre autre, la commande `ip`.
- Configurer son noyau
 - Ajouter (au moins) les options `CONFIG_IPV6=y`, `CONFIG_INET6_AH=y`, `CONFIG_INET6_ESP=y` (uniquement avec les noyaux 2.6.x pour les deux dernières options). Mettez `=y` pour les avoir en static dans le noyau, `=m` pour les avoir en module.
 - Compilez votre noyau avec vos nouvelles options et déployez, au besoin, vos nouveaux modules.
 - Insérez vos modules (`modprobe ipv6 ...`) ou rebootez sur votre nouveau noyau
- L'adressage
 - `ifconfig` vous montrera déjà que votre noyau IPv6 fonctionne

```
$ /sbin/ifconfig
eth0    Link encap:Ethernet  HWaddr 00:30:1B:B1:DE:FA
        inet addr:172.16.0.2  Bcast:172.16.255.255  Mask:255.255.0.0
        inet6 addr: fe80::230:1bff:feb1:defa/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
        RX packets:9329 errors:0 dropped:0 overruns:0 frame:0
        TX packets:10612 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:2517509 (2.4 MiB)  TX bytes:1465573 (1.3 MiB)
        Interrupt:18 Base address:0xe000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:78 errors:0 dropped:0 overruns:0 frame:0
        TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:5980 (5.8 KiB)  TX bytes:5980 (5.8 KiB)
```

- Vous pouvez maintenant paramétrer une adresse IPv6 statique choisie arbitrairement avec la commande :
`ip -6 addr add 2001:7a8:4b09:1::abba/64 dev eth0`

- Le routage

- `ip -6 route` vous donnera déjà les routes initialisées par défaut sur votre système

```
fe80::/64 dev eth0 metric 256 mtu 1280 advmss 1220 metric10 64
ff00::/8 dev eth0 metric 256 mtu 1280 advmss 1220 metric10 1
unreachable default dev lo proto none metric -1 error -101 advmss 1220 metric10 255
```

- Pour ajouter manuellement une route, faire : `ip -6 route add 2000::/3 via fe80::250:fcff:fe6d:b602 dev eth0` où `fe80::250:fcff:fe6d:b602` est l'adresse lien-local de votre routeur unicast par défaut et `eth0` votre interface réseau.

- Paramétrage de *radvd* pour l'autoconfiguration *stateless*

- Installer *radvd* (par les sources ou via votre distribution) sur votre routeur
 - Renseignez le fichier `/etc/radvd.conf` de la manière suivante (exemple)

```
interface eth1
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;

    AdvLinkMTU 1280;

    # Disable Mobile IPv6 support
    AdvHomeAgentFlag off;

    prefix 2001:7a8:4b09:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

- Lancez *radvd* par la commande `/sbin/radvd` ou par les rc-scripts traditionnels.

Support :

- Tous les logiciels supportant *IPv6* sont portables sous *GNU/Linux* de la même manière (exception faite de NFS/RPC qui posent d'autres problématiques).

3.2.2.2. RedHat Linux

Non approfondi.

Description :

- Support officiel d'IPv6 depuis la *RedHat* 7.1 (8.0 pour la *Mandrake*)
- Pas de support *a priori* des options de sécurité IPv6, ni de la mobilité

Déploiement :

- Vérifiez la présence du fichier représentatif du support d'IPv6 :
`/etc/sysconfig/network-scripts/network-functions-ipv6`
- Ajoutez le support d'IPv6 dans votre noyau avec la commande `modprobe ipv6`.
- L'ajout de `NETWORKING_IPV6=yes` dans le fichier `/etc/sysconfig/network` permettra d'automatiser les opérations relatives à IPv6 à chaque démarrage. (Il est nécessaire de redémarrer le service *network* ou de rebooter la machine pour rendre cela efficace.)

3.2.2.3. Debian GNU/Linux

Description :

- Support d'IPv6 à partir de la *woody*

Déploiement :

- Ajout dynamique du support d'IPv6 dans votre noyau avec `modprobe ipv6`
- Configuration statique d'IPv6 : Ajouter les options suivantes dans votre `/etc/network/interfaces`

```
iface eth0 inet6 static
    pre-up modprobe ipv6
    address 2001:7a8:4b09:1::abba
    netmask 64
    #mtu 1280
    #up echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
    #gateway fe80::250:fcff:fe6d:b602
```

- Installation de *radvd* par la commande `apt-get install radvd`

Support :

- Pour avoir un ensemble applicatif le plus large possible supportant IPv6, il est conseillé d'ajouter un des miroirs Debian-IPv6 (<http://debian.fabbione.net/>) dans votre `/etc/apt/sources.list` (puis `apt-get update`)
- Il existe aussi une liste de diffusion publique pour le support IPv6 de la Debian :
`debian-ipv6@lists.debian.org`

3.2.2.4. Autres

- Gentoo Linux : <http://doc.gentoo.org/Members/BeTa/ipv6-gentoo-howto>
- Suse Linux : <http://www.feyrer.de/IPv6/SuSE73-IPv6+6to4-setup.html>
- Slackware : <http://www.slackware-fr.com/>

3.2.3. Les logiciels tiers

3.2.3.1. Vue générale

Il existe en général au moins un logiciel supportant l'IPv6 (nativement ou via des patches) pour chaque domaine. Il existe quelques exceptions comme *Squid*, pour lesquelles il est très difficile d'obtenir des versions compatibles IPv6. Cependant ces exceptions sont rares.

3.2.3.2. Postfix

Postfix supporte IPv6 de manière stable depuis ses versions 2.x.x à travers des patches. Vous pouvez les retrouver à l'adresse <http://www.ipnet6.org/postfix/>

Exemple d'intégration d'IPv6 dans la configuration de *Postfix*

```
mynetworks = 192.168.5.0/24, 192.168.1.0/24, 192.168.2.0/24,
             127.0.0.0/8, [::1]/128,
             [2001:7a8:4b09:3::]/64, [2001:7a8:4b09:1::]/64,
             [2001:7a8:4b09::]/64, [2001:7a8:4b09:16::]/64
```

3.2.3.3. Apache

- *Apache-2.0.x* supporte nativement IPv6.
- *Apache-1.3.x* supporte IPv6 via des patches

Si vous utilisez des *wildcards* pour spécifier vos *binds* IP, vous ne verrez aucune différence pour le support d'IPv6, du moment que votre système le supporte lui-même. Si vous souhaitez préciser manuellement les IP à *binder*, il sera nécessaire d'ajouter les [], par exemple

```
<VirtualHost [2001:7a8:4b09:3::80]:80>
  [...]
</VirtualHost>
```

3.2.3.4. Bind9

Bind supporte nativement IPv6 dans ses enregistrements DNS à partir de ses versions 8.x.y (avec $x > 0$), et au niveau connectivité réseau, à partir de ses version 9.x.

Pour les enregistrements DNS, il s'agit là d'enregistrements tout à fait classiques mais de type AAAA ou PTR tout ce qu'il y a de plus classiques.

Pour les opérations réseau sur IPv6, *Bind* a besoin d'une part qu'on lui précise la nécessité d'écouter sur IPv6 via la directive `listen-on-v6 { any; }` dans son fichier `/etc/bind/named.conf`. Pour la notation des adresses IPv6 dans ce même fichier, il n'y a pas de différence avec les adresses IPv4, comme vous pouvez le constater :

```
allow-transfer
{
    213.42.145.91;
    2001:7a8:4b09:3::/64;
    ::1/128;
};
```

Par contre, il est intéressant de noter les directives `transfer-source-v6` et `notify-source-v6` permettant de "forcer" l'adresse source IPv6 que Bind utilisera pour communiquer avec ses correspondants. Cela peut s'avérer grandement utile, IPv6 permettant l'utilisation d'adresses multiples sur chaque interface.

3.2.3.5. Jabberd

Jabberd supporte nativement IPv6 depuis la version 1.4.3. Si vous ne spécifiez pas d'IP spécifique à *binder*, votre serveur écoutera automatiquement sur toutes vos adresses IPv6, sans modification supplémentaire.

3.2.3.6. Mozilla, KDE, pure-ftpd, etc...

Il existe de nombreuses applications supportant IPv6 dont je n'ai pas parlé ici. Au nombre d'entre elles, on peut trouver :

- de nombreux serveurs FTP, *OpenSSH*, *Cups* (à partir de sa version 1.2 à venir), ...
- le projet *KDE* (les sockets *KDE* supportent IPv6 par défaut)
- les divers outils *Mozilla* ainsi que leurs dérivés (*Thunderbird*, etc...)
- *irssi*, *xchat*, *gaim*, ...
-

3.3. Quelques solutions de migration

3.3.1. Dual-Stack

En français : "double pile".

Cela consiste à relier à la fois en IPv4 et en IPv6 tout équipement relié au réseau. Ainsi, en privilégiant la pile IPv6, les équipements l'utilisent quand il est disponible, et passent sur leur pile IPv4 en cas d'indisponibilité.

Inconvénients :

- Ne résoud pas les problèmes de pénurie d'adresses IPv4
- Maintenance plus lourde sur l'ensemble des équipements du réseau

Avantage :

- Mécanisme de transition le plus simple et le plus souple

3.3.2. Mono-stack, Relais applicatifs

En français : "simple pile, et relais applicatifs".

Cette stratégie consiste ne relier tous les équipements terminaux que sur IPv6, et de prévoir des relais applicatifs ("proxies" ou "serveurs mandataires") double pile pour accéder au réseau v4 inaccessible en v6.

Inconvénients :

- Oblige à rendre par avance toutes les applications des "équipements terminaux" compatibles IPv6,
- Oblige à mettre en place un relai applicatif par service (un proxy DNS, un proxy web, un proxy pour chaque service de messagerie, ...), et qu'il supporte à la fois IPv4 et IPv6.

Avantages :

- Permet d'avoir des clients uniquement v6
- Résoud partiellement le problème de pénurie des adresses IPv4

3.3.3. Mono-stack, NAT-PT

En français : "Simple pile, translation de protocole"

Ici, tous les équipements terminaux possèdent une pile IPv6 simple et sont configurés comme s'ils évoluaient dans un monde totalement v6. Pour ce faire, il est indispensable de disposer d'un relai DNS spécial ainsi que d'un routeur capable de faire de la translation d'adresses d'IPv6 vers IPv4.

Au niveau du relai DNS, ce dernier doit transformer toutes les requêtes sur des enregistrements A en enregistrements AAAA particuliers. En effet, une adresse de type 216.239.59.99 sera alors traduite en 2001:7a8:4b09:ffff:216:239:59:99 (simplifiée pour l'exemple) par le relai.

Par la suite, les équipements voulant accéder à 216.239.59.99 depuis le réseau v6 uniquement, y accéderont par l'adresse IPv6 2001:7a8:4b09:ffff:216:239:59:99 et un routeur double pile capable de comprendre cela fera une translation d'adresse comme en IPv4 avec le NAT, en convertissant l'IPv6 en IPv4.

Inconvénients :

- Oblige à avoir rendu toutes les applications des "équipements terminaux" compatibles IPv6 par avance
- Oblige à mettre en place un relai DNS et un routeur spécialisés dans le NAT-PT

Avantages :

- Seuls un relai DNS accompagné d'un routeur spécialisés sont nécessaires en plus des services habituels

- Permet d'avoir des clients uniquement v6
- Bonne alternative à la pénurie d'adresses IPv4

3.3.4. Un exemple de scénario de transition

- IPv6 dans les routeurs
- Connexion aux réseaux IPv6 mondiaux
- IPv6 dans les serveurs internes
- IPv6 dans les terminaux
- Présence majoritaire d'IPv6 dans le réseau interne
- Compatibilité IPv4 (délégation au fournisseur d'accès, mise en place des mécanismes vus plus haut, etc...)

4. Conclusion

4.1. Prospectives

4.1.1. Techniques

IPv4 commence à poser de très gros soucis en :

- Asie (particulièrement)
- Afrique
- Amérique du Sud

IPv6 arrive à très grands pas en Asie :

- Au niveau des infrastructures réseau (la Chine dispose actuellement du plus grand réseau IPv6 au monde)
- Au niveau du développement applicatif (particulièrement visible dans le monde *OpenSource* avec des projets comme *KAME*, *USAGI*, *WIDE*, ...)
- Au niveau des besoins des utilisateurs (besoin d'un accès direct à l'Internet mondial)

IPv6 est relativement applicable dès aujourd'hui :

- À petite et moyenne échelle, car il existe un grand nombre de solutions à la fois applicatives et matérielles, même si un grand nombre de portages de logiciels propriétaires reste à faire (produits Microsoft en particulier)
- À plus grande échelle, *Cisco*, *Juniper*, *6Wind* etc... disposent déjà de solutions intégrant nativement IPv6.

Arrivée de nouveaux besoins :

- Téléphonie mobile de Nème génération

- Nouveaux services (télésurveillance, VoIP/téléphonie sur IP, ...)
- Véhicules communicants, réseaux de capteurs, électronique embarquée, etc...

Si on prend en compte un renouvellement du matériel réseau régulier, on peut considérer qu'il faut mettre à jour tout matériel ayant plus de 5 ans d'âge :

- Depuis 2003, il est possible d'intégrer IPv6 dans ses équipements, en général sans surcout dissuasif.
- D'ici 2008, on peut donc décemment croire que tous auront été renouvelés et qu'ils seront donc tous compatibles avec IPv6.
- Le matériel ne sera plus un facteur limitant le déploiement d'IPv6.
- Le logiciel ne devrait plus non plus en être un frein (l'*OpenSource* est déjà largement en mouvement, le logiciel propriétaire suit aussi, mais plus doucement)

4.1.2. Humaines

Il existe et existera encore de gros freins humains au développement d'IPv6, en particulier en Amérique du Nord et en Europe :

- Réticence claire de la grande majorité de "petits" FAI par rapport à IPv6
 - Disposition d'un nombre encore suffisant d'IPv4
 - Vue très pragmatique des couts d'une migration qui n'aura, selon eux, pas forcément de retombées commerciales du côté de l'utilisateur *final*.
- Problème de formation : même si IPv6 n'est pas si complexe que ça à appréhender, il existe un vrai manque dans les formations spécialisées sur ce domaine.

Cependant, malgré ces freins, la présence d'IPv6 avance dans les esprits :

- On peut voir de plus en plus de grands opérateurs nationaux ou internationaux se relier, malgré leurs apparentes réticences, aux réseaux IPv6 (6Bone ou BackBone) qui prennent de plus en plus d'ampleur.
- Tout le monde se prépare plus ou moins secrètement au passage à IPv6, le jour où le besoin sera clairement identifié.

IPv6 explosera réellement dans les utilisations de tous les jours lorsque ses avantages techniques entraineront son adoption par le grand public.

4.1.3. Financières

Il arrivera un jour où IPv4 coutera plus cher à maintenir que ses équivalent IPv6 :

- Problématiques de compétences avec IPv4 (on peut rêver...)
- Problématique de routage au niveau du BackBone entraînant de gros surcouts au niveau matériel
- Problématique de fonctionnalités plus coûteuses à mettre en place avec IPv4

- Il est plus couteux, par exemple, de mettre en place des fonctionnalités IPsec sur IPv4 que sur IPv6 quand les technologies sont maîtrisées.
 - La mobilité dans IPv4 est plus couteuse en ressources réseau et humaines que son équivalent avec IPv6.
- Problématique de rareté des adresses IPv4 ("Tout ce qui est rare est cher...")

4.2. Enjeux économique-politiques de demain

Comme nous avons pu le voir, IPv6 arrive... même s'il lui faut le temps. Cependant, on peut reconnaître trois des grandes clés de l'avenir d'IPv6 :

- L'anticipation à l'échelle "micro"
- Les enjeux industriels mondiaux
 - Ouverture potentielle du marché à de nouveaux acteurs industriels
 - Le besoin réel de voir arriver IPv6 en Asie (par exemple) donne de vraies raisons aux industriels de cette région du monde de reprendre l'avance qu'ils ont toujours eu du mal rattrapper avec IPv4 face aux grands constructeurs occidentaux
- Les enjeux politiques mondiaux
 - Notre société actuelle s'articule autour de l'information. Ainsi, Internet prend une place de plus en plus grandissante.
 - L'arrivée d'IPv6 permet la révision complète de l'échiquier mondial du réseau des réseaux.
 - Alors que l'Amérique du Nord et l'Europe étaient habituées à la répartition actuelle des adresses IPv4, leur assurant un status privilégié, "figé", IPv6 repositionne les différents acteurs d'Internet sur un pied d'égalité.

Ainsi, selon le *Gartner Group* (analyste industriel) et Adam Judd (vice-président de *Juniper Networks* Asie), l'adoption massive d'IPv6 n'aura probablement pas lieu, au minimum, avant 2008. En attendant cette échéance, il nous appartient à tous de rester vigilant et d'anticiper cette nouvelle "étape" technologique.

5. Annexes

5.1. L'auteur

Baptiste SIMON (<http://www.e-glop.net/>) <baptiste.simon@e-glop.net (mailto:baptiste.simon@e-glop.net)>

Administrateur systèmes GNU/Linux & UNIX

À la recherche d'emploi (<http://www.e-glop.net/cv/>)

5.2. Support IPv6 francophone

Il existe une liste de diffusion francophone ouverte à tous traitant d'IPv6. Vous pourrez la trouver sur <http://lists.e-glop.net/mailman/listinfo/ipv6-fr>. N'hésitez pas à nous y rejoindre.

5.3. Aperçu des divers formats de ce document

Ce document a été rédigé au format RST (<http://docutils.sourceforge.net/>) avec KWrite puis converti aux formats DN-XML et Docbook avec `dn2dbk.xsl` (<http://membres.lycos.fr/ebellot/dn2dbk/>).

Les différentes versions ci-dessous ont été réalisées avec les feuilles XSLT officielles de docbook¹ et les outils du paquet `xmlto` (<http://cyberelk.net/tim/xmlto/>) de la Debian GNU/Linux.

Retrouvez toutes ces version ici :

- XHTML (<http://www.e-glop.net/howtos/ipv6-complet.xhtml>)
- HTML (<http://www.e-glop.net/howtos/ipv6-complet.html>)
- PDF (<http://www.e-glop.net/howtos/ipv6-complet.pdf>)
- postscript (<http://www.e-glop.net/howtos/ipv6-complet.ps>)
- Texte brut (<http://www.e-glop.net/howtos/ipv6-complet.txt>)
- RTF (<http://www.e-glop.net/howtos/usbstorage-sdcard.rtf>)
- reStructuredText (<http://www.e-glop.net/howtos/ipv6-complet.rst>)
- DocBook - XML (<http://www.e-glop.net/howtos/ipv6-complet.db-xml>)
- DN-XML (<http://www.e-glop.net/howtos/ipv6-complet.dn-xml>)
- XSL-FO (<http://www.e-glop.net/howtos/ipv6-complet.fo>)

5.4. Licence de publication

Ce document issu de www.e-glop.net (<http://www.e-glop.net/>) est soumis à la licence Creative Commons by-sa (<http://creativecommons.org/licenses/by-sa/1.0/>). Permission vous est donnée de distribuer, modifier des copies de ce document (traduction, modifications, adaptation, etc...) tant que vous respectez la licence sus-citée.

Notes

1. paquet `docbook-xsl` (<http://www.docbook.org/wiki/moin.cgi/DocBookXslStylesheets>) sur Debian GNU/Linux