

Affaire suivie par :  
CERTA

## NOTE D'INFORMATION DU CERTA

### Objet : Migration IPv6 : enjeux de sécurité

---

Les informations publiées par le CERTA restent sous le contrôle du CERTA. Toute rediffusion, en dehors du domaine du CERTA est soumise à son autorisation écrite. Le domaine d'intervention du CERTA regroupe les administrations et les collectivités locales.

---

### Gestion du document

Référence	CERTA-2006-INF-004
Titre	Migration IPv6 : enjeux de sécurité
Date de la première version	11 septembre 2006
Date de la dernière version	–
Source(s)	
Pièce(s) jointe(s)	Aucune

TAB. 1 – *Gestion du document*

Une gestion de version détaillée se trouve à la fin de ce document.

### Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>IPv6 en pratique</b>	<b>3</b>
2.1	Avertissement . . . . .	3
2.2	Les adresses . . . . .	3
2.3	Les entêtes . . . . .	3
2.4	Des protocoles associés . . . . .	5
2.4.1	Couche sous-jacente dite de liaison . . . . .	5
2.4.2	Cas particuliers des protocoles de routage . . . . .	6
2.4.3	Cas particulier des appareils mobiles . . . . .	7
2.4.4	Des usages nouveaux . . . . .	7
<b>3</b>	<b>Processus de traduction entre IPv4 et IPv6</b>	<b>8</b>
3.1	La double-pile IP, ou Dual Stack . . . . .	8
3.2	Transport de IPv6 dans IPv4 . . . . .	8
3.3	Tunnels automatiques : IPv6 dans IPv4 . . . . .	9
3.4	Transport de IPv4 dans IPv6 . . . . .	10
3.5	La continuité du service DNS . . . . .	11
3.6	Des groupes de travail . . . . .	12

<b>4</b>	<b>La sécurité considérée dans IPv6</b>	<b>12</b>
4.1	IPsec par défaut . . . . .	12
4.2	SEND : Sécuriser la découverte du voisinage NDP . . . . .	13
4.3	Pare-feux et filtrage . . . . .	13
<b>5</b>	<b>Problèmes et risques</b>	<b>14</b>
5.1	Les piles protocolaires existantes . . . . .	14
5.1.1	Trouver IPv6 dans son système . . . . .	14
5.1.2	Existence de problèmes de mise en œuvre . . . . .	14
5.2	Les risques existants avec IPv6 . . . . .	14
5.3	Les vers, une fin annoncée? . . . . .	15
5.4	Problématiques du filtrage . . . . .	15
5.5	Les applications . . . . .	16
<b>6</b>	<b>Recommandations du CERTA</b>	<b>17</b>
6.1	Les 10 premières recommandations . . . . .	17
6.2	Désactivation possible . . . . .	17
<b>7</b>	<b>Conclusion</b>	<b>18</b>
<b>8</b>	<b>Documentation</b>	<b>18</b>

## Pourquoi lire ce document?

- 1° IPv6 est maintenant mis en œuvre dans de nombreux systèmes d'information ;
- 2° IPv6 est encore méconnu ;
- 3° IPv6 prétend résoudre certains problèmes de sécurité ;
- 4° La migration vers IPv6 est déjà engagée dans plusieurs réseaux ;
- 5° Les risques associés au déploiement IPv6 sont moins populaires.

Ce document s'adresse donc aux personnes soucieuses de comprendre les enjeux de sécurité liés à l'introduction d'IPv6. Il demande aussi un minimum de connaissances du fonctionnement actuel de certains protocoles (IPv4, ICMPv4, ARP, etc).

## 1 Introduction

Le protocole de routage principalement utilisé aujourd'hui pour les communications Internet est le protocole IP (Internet Protocol). La version la plus utilisée du protocole IP est la version 4 (on utilisera l'abréviation IPv4) et n'a fait l'objet d'aucune évolution majeure depuis la publication du document fondateur, la RFC 791, en septembre 1981. Le protocole IP dans sa version 4 s'est avéré assez robuste tout au long de l'essor de l'Internet. Cependant, un certain nombre de caractéristiques et d'évolutions n'ont pas été prises en compte, ce qui a conduit à la nécessité de l'élaboration d'un successeur au protocole IP actuel.

Parmi les évolutions actuelles mal considérées par IPv4, on peut citer :

- la croissance rapide de l'Internet qui conduit rapidement à un épuisement des adresses IPv4 disponibles ;
- la multiplication des systèmes communicants mobiles (PDAs, téléphones portables, ...) ;
- l'essor de nouveaux services de diffusion multimédia (vidéo ou radio sur l'Internet, vidéoconférence, etc).

Le successeur naturel aurait pu logiquement être IP version 5, mais cette version a été attribuée à un protocole expérimental : ST (Internet Stream Protocol), défini pour la première fois en 1979 et qui n'a jamais atteint le grand public. Le successeur fut donc choisi sous le nom de IPv6.

Les détails techniques du protocole s'appuient sur le document [RFC3513]. Nous n'aborderons que sommairement ces derniers, l'objectif n'étant pas ici de décrire les subtilités du protocole, mais plutôt d'en comprendre les enjeux relatifs à la sécurité du réseau. IPv6 est voué à se déployer dans les prochains mois et les quelques années à venir. Il est donc capital de saisir les avantages qu'il apporte en terme de sécurité, mais aussi les points plus obscurs qui ne sont pas, aujourd'hui encore, complètement éclaircis.

Une section présentant les recommandations du CERTA se trouvent en section 6.

Enfin, ce document traitant de la migration IPv4 vers IPv6, il s'adresse particulièrement aux lecteurs qui ont une connaissance des protocoles courants IPv4, ARP et ICMP.

## 2 IPv6 en pratique

### 2.1 Avertissement

Ce document ne prétend pas décrire de manière exhaustive toutes les vicissitudes du protocole. Il est néanmoins intéressant de rappeler certaines de ses propriétés pouvant aider à une meilleure compréhension des aspects de sécurité. Ce qui suit est donc une vision globale, ou un bref aperçu de ce que l'appellation IPv6 dissimule.

### 2.2 Les adresses

IPv6 est un protocole réseau, servant à véhiculer des paquets de données à travers différents éléments actifs (routeurs par exemple) du réseau. Tout comme IPv4, IPv6 doit envelopper (ou *encapsuler*) les données dans certains paquets, en ajoutant plusieurs informations. Celles-ci se trouvent dans un endroit particulier, appelé entête du paquet.

Elle précise entre autres l'adresse de l'émetteur du paquet, ainsi que celle du destinataire.

Sous IPv4, il est fréquent de voir une telle *adresse* sur 32 bits sous la forme W.X.Y.Z, avec chacune de ces lettres représentant des nombres entre 0 et 255. Par exemple :

- 213.56.176.2 : l'adresse publique du site [www.certa.ssi.gouv.fr](http://www.certa.ssi.gouv.fr) où ce document a été publié ;
- 192.168.0.1 : une adresse privée réservée à un usage interne ;
- 127.0.0.1 : l'adresse de bouclage *loopback*, quand une machine veut s'envoyer des paquets (lorsqu'elle combine les rôles de client et serveur par exemple).

Sous IPv6, le format d'une adresse change : elle fait maintenant 128 bits (donc un stock de  $2^{128}$  adresses, soit  $2^{96}$  fois plus qu'avec IPv4), et se présente en 8 mots de 16 bits (4 hexadécimaux) séparés par le caractère « : ». La représentation du préfixe, reste, elle, similaire à celle de la notation CIDR du RFC 1519 utilisée avec IPv4 [RFC1519]. Il s'agit des fameux /24, /8, etc. Deux illustrations, l'une étant l'adresse IPv6 d'une machine, l'autre celle d'un réseau, sont représentées ci-dessous :

- **Adresse IPv6 d'une machine** : FEDC:400A:210F:34ED:1111:4444:DE3E:38D9
- **Adresse IPv6 d'un réseau** : FEDC:400A:210F:34ED::/64
- **Adresse de type *loopback*** : 0:0:0:0:0:0:1 ou ::1

Dans le cas où l'adresse IPv6 est directement mise dans le lien réticulaire d'un site (ou URL), celle-ci doit apparaître entre crochets : `http://[xxx:xxxx:xxxx::xxxx:xxxx]/index.html`, parce que «:» a une autre signification dans une URL. Enfin, tout comme pour IPv4, certaines plages sont réservées. De plus amples détails se trouvent dans les RFC 3513, 4048, 4193 et 3879.

### 2.3 Les entêtes

L'entête d'un paquet est légèrement différente de celle mise en œuvre par le protocole IPv4, mais de nombreux points communs demeurent, comme l'illustrent les figures 1 et 2 : la figure 1 présente un paquet IPv6 dans sa globalité, et la figure 2 décrit les différents champs de l'entête IPv6.

Les standards se sont appuyés sur l'acquis d'IPv4 afin de simplifier les champs de l'entête, au bonheur des administrateurs réseaux et des routeurs.

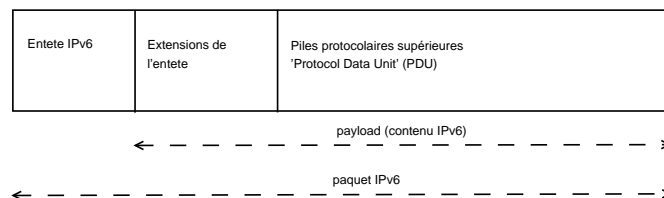


FIG. 1 – Paquet IPv6

En première remarque, il n'existe plus de champ *checksum*, qui vérifie l'intégrité du paquet : ainsi, IPv6 ne vérifie plus si une erreur est apparue au niveau de son entête au cours des différentes étapes de traitement du paquet. Il se pose alors le problème de savoir si le champ source ou destination du paquet ne contient pas d'erreur. IPv6 considère que cette tâche revient aux protocoles de niveau supérieur, qui devront mettre en place un mécanisme

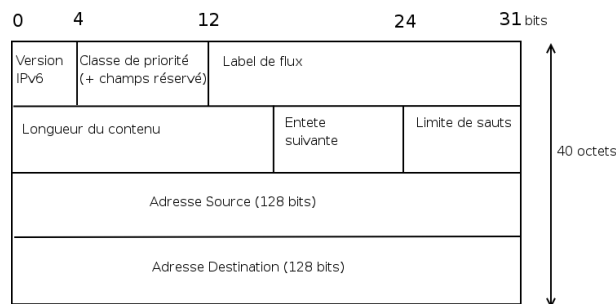


FIG. 2 – Entête IPv6 [Point6, RFC2474]

de vérification d'intégrité de bout-en-bout, incluant une pseudo-entête avec les adresses source et destination. Par exemple, UDP permet d'ajouter ces champs et vérifier les erreurs, mais ceux-ci sont facultatifs sous IPv4. Cependant, ce manque d'intégrité au niveau réseau peut poser des problèmes. D'une part, les routeurs ne faisant plus de contrôle d'intégrité, la détection d'une erreur dans le paquet ne se fera qu'à sa destination (lors de la lecture des protocoles des couches supérieures). D'autre part, il est plus difficile d'identifier les connexions réseau qui seraient à l'origine d'un taux anormal d'erreurs au cours du voyage des paquets.

ICMPv4 (pour *Internet Control Message Protocol*) est le protocole utilisé pour gérer des informations relatives à l'état du réseau des machines connectées. A la différence d'UDP, ICMPv4 ne vérifie que l'intégrité du message ICMP, et non la source et la destination du paquet original. Ceci a contribué à l'apparition d'un protocole équivalent mais plus adapté, nommé ICMPv6.

Comme pour IPv4, IPv6 prend en compte le nombre de sauts, ou nombre de routeurs que le paquet est autorisé à traverser. Dans IPv4, ce champ est appelé *durée de vie*, ou TTL (pour *Time-to-Live*). Il est décrémenté à chaque noeud traversé, normalement de 1. La valeur initiale de ce champ devrait être donnée par l'IANA (<http://www.iana.org>). Cette valeur, codée sur 8 bits, n'est cependant pas encore attribuée, et la plupart des mises en œuvre sont choisies au bon gré des développeurs, tout comme pour IPv4. Il reste codé sur 8 bits, donc IPv6 ne prévoit pas plus de  $2^8 = 255$  sauts pour un même paquet.

IPv4 réserve une place dans l'entête consacrée à différentes options (*traceroute*, *timestamp*, etc). Dans IPv6, ces options n'existent plus dans l'entête. Elles sont remplacées par des champs facultatifs, non intégrés à l'entête, et appelés *extensions*.

Les extensions sont prises en compte par les équipements destinataires du paquet. Le RFC 2460 donne plusieurs recommandations, mais ces champs restent de taille, de contenu et de signification variables. Il existe une recommandation concernant leur ordre d'apparition dans le paquet IPv6, mais la seule vraie restriction concerne une extension, nommée *Proche-en-Proche* (ou *Hop-by-Hop*).

Les extensions les plus significatives sont :

- l'extension *Proche-en-Proche* : cette extension, considérée par chacun des noeuds, permet au routeur traitant le paquet d'échanger de l'information avec les routeurs suivants, sous forme d'options. Plus exactement, il indique son comportement quand il rencontre un champ étrange, par exemple qu'il ne sait pas interpréter, dans le paquet IPv6. Deux options sont actuellement détaillées dans les normes [RFC2675, RFC2711]. L'une des plus inquiétante du point de vue de la sécurité est l'option «*Router Alert*», qui demande au routeur suivant d'examiner le contenu des paquets qu'il relaie, comme les protocoles RSVP (signalisation de flux) et *multicast* (*Multicast Listener Discovery*) l'exigent. Cette option permet d'inciter un routeur à interpréter les données d'un paquet, même si cela n'est pas sa fonction première, au risque d'augmenter sa charge de travail (déni de service) et de le voir effectuer une mauvaise interprétation. D'autres options devraient apparaître dans les prochains mois.
- l'extension *Destination* : cette extension contient également des options, qui seront traitées par l'équipement destinataire. Elle n'est pas encore vraiment exploitée, car elle peut paraître redondante avec l'extension précédente.
- l'extension *Routage* : cette extension permet d'imposer à un paquet une route différente de celle offerte par la politique de routage du réseau : elle met en œuvre le *routage par la source*. Le principe est identique dans IPv4.

- l’extension *Fragmentation* : comme pour IPv4, certaines applications telles que NFS sur UDP nécessitent une découpe des messages, beaucoup trop gros pour être contenu intégralement dans un paquet IP. Pour réduire le travail des routeurs intermédiaires, le processus de fragmentation se fait sur l’équipement émetteur, qui fragmente, puis sur l’équipement du récepteur, qui réassemble. Les informations concernant les routeurs intermédiaires (extensions *proche-en-proche* et *routage*) sont elles recopiées dans chaque fragment. Certaines bonnes pratiques doivent être contrôlées au cours de la fragmentation : 1) tout fragment, hormis le dernier, doit être de taille supérieure à 1280 octets ; 2) tous les fragments doivent parvenir à destination dans une fenêtre de temps raisonnable (de l’ordre de quelques dizaines de secondes au maximum).
- l’extension *Sécurité* : deux extensions de sécurité, l’une pour l’authentification AH (*Authentication Header*), l’autre pour la confidentialité ESP (*Encapsulating Security Payload*) sont définies par l’IETF. Nous détaillons celles-ci dans la section 4.
- l’extension *Mobilité* : cette option indiquant la mobilité (voir section 2.4.3) est aussi intéressante, mais elle est en cours d’évolution. Elle sert normalement à maintenir une relation entre le système mobile distant et son réseau d’origine.

Une autre option intéressante de l’extension *Proche-en-Proche* est le *jumbogramme* [RFC2675]. Cette option signale aux éléments du réseau qu’ils doivent traiter un paquet de taille extrêmement grande. Quand la longueur des données dépasse 65535 octets, le champ de l’entête précisant la longueur (codé sur 2 octets) vaut alors 0, et cette option est employée. Bien que cette option semble offrir une optimisation au niveau de la bande passante, elle peut perturber un ensemble d’éléments dans le réseau qui n’ont pas la capacité de gérer de tels paquets (sondes IDSs, pare-feux).

Par ailleurs, IPv4 n’a jamais été créé dans le but de favoriser le *multicast*, c’est-à-dire la distribution simultanée d’information vers un groupe donné de destinataires. Cependant, le *multicast* présente un attrait certain pour les différents protocoles multimédia comme RTP (pour *Real Time Protocol*), et la diffusion de données multimédia (films, musique). IPv6 se doit donc d’intégrer ce moyen de communication, et ceci est rendu possible par le biais de certaines extensions.

Dans cette brève introduction au format IPv6, il apparaît que la structure globale de l’entête IPv6 ne diffère guère de celle d’IPv4. Cependant, l’entête possède des extensions offrant de nombreuses possibilités : cela conduit à la définition de nombreux champs, souvent optionnels. Les RFCs ne détaillent pas systématiquement les valeurs par défaut. Les mises en œuvre d’IPv6 pourraient donc être identifiables selon les valeurs choisies. Par ailleurs, certaines extensions peuvent également poser directement problème, comme c’est le cas pour l’une des options de l’extension *Proche-en-Proche* : *Router Alert*, qui demande au routeur suivant d’examiner le contenu des paquets qu’il relaie, comme certains protocoles l’exigent. Cette option permet d’inciter un routeur à interpréter les données d’un paquet, même si cela n’est pas sa fonction première, au risque d’augmenter sa charge de travail (dénier de service) et de le voir effectuer une mauvaise interprétation.

En se bornant à cette présentation sommaire, il apparaît aussi que les *extensions* doivent rester intègres entre la source et la destination. Dans le cas contraire, il serait facile, pendant le trajet du paquet, de changer le message de l’extension à l’attention du destinataire.

## 2.4 Des protocoles associés

### 2.4.1 Couche sous-jacente dite de liaison

Un équivalent du protocole ARP pour IPv4 est utilisé dans IPv6 pour établir le lien entre les adresses IPs et les adresses physiques MAC : le protocole de découverte des proches voisins NDP (ou *Neighbor Discovery Protocol*). Le protocole réalise différentes fonctions, comme la résolution d’adresses, la détection d’inaccessibilité, la configuration automatique des équipements, ou la découverte des routeurs et des préfixes. La principale différence vient de l’emploi de messages standards ICMPv6 en remplacement d’ARP. ARP n’existe donc plus sous IPv6 ! Parmi les simplifications, les adresses de diffusion (aussi appelées *broadcast*) n’existent plus : le champ «nombre de sauts» de l’entête IPv6 contient la valeur maximale (255), afin de vérifier que l’information provient du même lien physique<sup>1</sup>.

L’idée globale étant de simplifier la configuration du réseau dans les systèmes d’information, IPv6 spécifie deux méthodes distinctes mais non exclusives pour configurer une machine :

**Autoconfiguration sans état** : il faut comprendre par là une configuration automatique des machines quand la gestion administrative des adresses attribuées n’est pas nécessaire [RFC2462]. Cela présente un avantage

1. Les RFCs ne font cependant pas mention de la situation d’un paquet IPv6 traversant un tunnel, et pouvant également avoir un TTL de 255

quand plusieurs systèmes d'information sont connectés sur le même lien logique, et souhaitent communiquer. Pour vérifier l'unicité des adresses utilisées, les machines doivent exécuter un mécanisme de *Détection d'Adresse Dupliquée* (ou DAD).

**Autoconfiguration avec état :** cette méthode est adoptée quand un site demande un contrôle strict de l'attribution des adresses et de la configuration réseau. Le transfert d'information se fait alors par le protocole DHCPv6. L'ensemble des éléments du protocole DHCPv6 est décrit dans [RFC3315].

Ces deux méthodes sont illustrées par la figure 3.

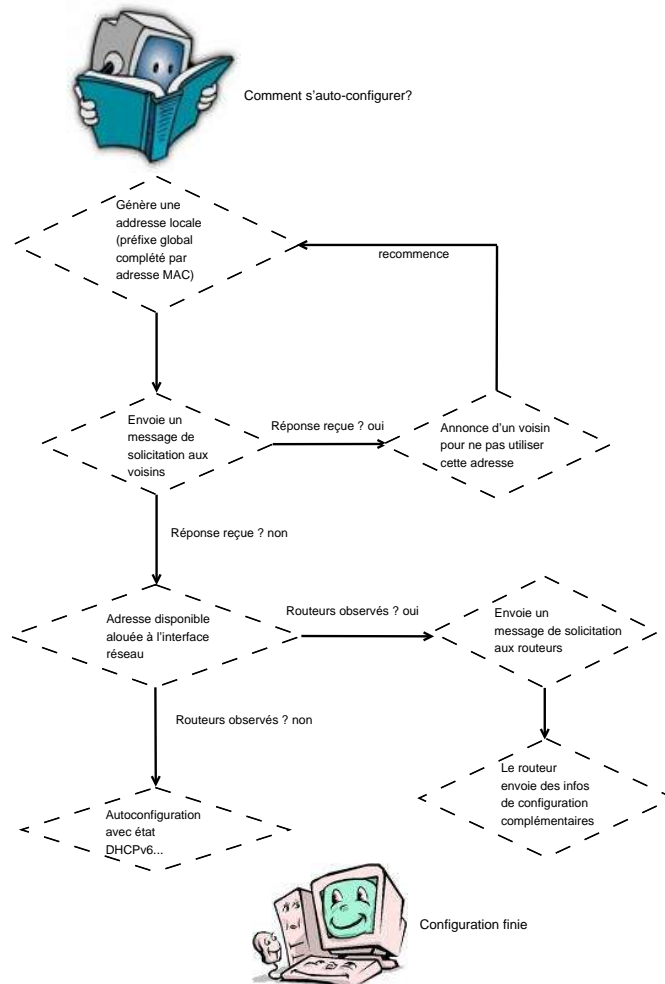


FIG. 3 – Deux méthodes de configuration automatique pour IPv6

Cette procédure d'autoconfiguration peut soulever des inquiétudes concernant la notion de «protection de la vie privée» (ou *privacy*). En effet, les adresses temporaires sont créées à partir de l'adresse MAC (qui identifie aussi le matériel réseau). Chaque machine, et donc chaque internaute, transmet un numéro de série unique sur l'Internet, et stable dans le temps. Ce numéro est transmis avec l'adresse IPv6, i.e. se retrouve dans l'envoi de courrier, les forums de discussion, l'accès aux moteurs de recherche, etc.

Des solutions existent pour remédier à ce problème, l'une proposant de remplacer l'identifiant fixe par un nombre pseudo-aléatoire [RFC3041]. Cette option est par exemple activée par défaut sous Windows XP, mais n'est pas acceptée avec certaines configurations de routage, ou avec l'utilisation de portails captifs (réseaux sans-fil notamment).

#### 2.4.2 Cas particuliers des protocoles de routage

Les protocoles de routage ne changent pas avec IPv6. Les efforts qui ont été faits consistent seulement à les adapter au nouveau format des adresses IPv6. En revanche, ils bénéficient des apports d'IPv6, comme l'authentifi-

cation et la diffusion *multicast*.

Pour le routage interne (au sein d'un sous-réseau), l'équivalent de RIPv2 est RIPng, décrit dans [RFC2080]. Il est quasiment identique, mais n'inclut plus d'authentification. Celle-ci repose sur les moyens de sécurité mis en place au niveau d'IPv6 (voir section 4.1).

Le second protocole de routage interne, OSPF passe à la version 3. Une opération a été effectuée afin de rendre le coeur du protocole indépendant du protocole réseau IPv6, et de le restreindre au transport d'informations d'adressage. Les routeurs voisins sont indiqués par un identifiant, afin de limiter l'usage des adresses IPv6.

Le protocole de routage externe (interconnexion de plusieurs sous-réseaux) MPLS est particulièrement adapté pour la transition IPv4 vers IPv6. En effet, l'interconnexion de différents réseaux IPv6 se fait de manière transparente, même au travers d'un réseau IPv4 MPLS, dans la mesure où MPLS commute des labels, et non pas des adresses IPs. Ce protocole reste cependant vulnérable à certaines classes d'attaques particulières, comme l'injection malveillante de labels au sein du réseau ou certaines attaques de niveau 2 (couche MAC).

Le second protocole de routage externe, actuellement utilisé pour le routage global de l'Internet est BGPv4. Il garde la même appellation pour IPv6. Son adaptation s'est globalement limitée à changer trois attributs dont le format dépend de l'adresse. Les règles d'agrégation de plages d'adresses restent inchangées.

Les protocoles de routage sont donc peu modifiés pour le moment. Ils gardent les avantages et faiblesses actuels. Il est aussi intéressant de noter pour RIPng la volonté de limiter l'authentification au niveau IPv6, fonctionnalité discutée au paragraphe 4.1.

### 2.4.3 Cas particulier des appareils mobiles

IPv6 est aussi un protocole destiné à interconnecter les terminaux mobiles. MIPv6 (ou Mobile IPv6) permet, en pratique, à une machine de rester joignable et de communiquer avec la même adresse, quelle que soit son medium et sa position courante. Il est par exemple peu à peu intégré dans les différents éléments de l'architecture de la téléphonie dite de *troisième génération*. Globalement, les mécanismes d'IPv6 vus dans les sections précédentes offrent une très bonne base à la gestion de la mobilité. En effet, ils résolvent plusieurs difficultés qu'avaient à résoudre les solutions de mobilité d'IPv4. Ainsi, le mécanisme de configuration sans état permet au système mobile d'acquiescer, en déplacement, une adresse IP fonctionnelle. Il peut dès lors communiquer. Le mécanisme d'annonce des routeurs facilite quant à lui la détection du mouvement, qui est essentielle à la gestion de la mobilité. De manière grossière, il est possible de distinguer, au niveau protocolaire, trois propriétés :

- L'extension *Routage* est utilisée par la source pour lister le nœud (routeur) intermédiaire par lequel le paquet doit transiter pour atteindre sa destination.
- Un routeur particulier, nommé GGSN, à cheval entre le monde GSM/UMTS et le monde filaire assure les procédures de configuration avec le terminal mobile.
- Les annonces de routeur IPv6 sont plus espacées dans le temps (le délai maximal annoncé étant de 6h, le minimum étant 4,5h - variable selon l'opérateur) pour éviter de charger le lien UMTS, souvent facturé à la quantité d'information transmise.

Le protocole *Mobile IPv6* permet au système mobile de conserver l'adresse utilisée dans son réseau d'origine, même en mouvement. Celle-ci se nomme *adresse mère* (HoA). Il acquiesce par ailleurs des adresses temporaires (CoA) locales aux réseaux qu'il visite. Une machine, *l'agent mère*, garde les traces de ces changements successifs. Le mobile utilise enfin la détection de voisinage pour voir si son routeur par défaut n'est plus accessible. Les échanges principaux sont résumés dans la figure 4 : un correspondant extérieur envoie du trafic vers *l'agent mère* qui les communique au mobile, qui lui-même communique directement avec l'hôte.

Il existe un risque lorsque le système mobile envoie des mises à jour d'association à l'agent mère. Celles-ci peuvent être fausses. Un système malveillant dans un autre réseau peut lui-même envoyer une fausse demande afin de détourner le trafic de son véritable destinataire. Il est indispensable d'utiliser une méthode d'authentification entre le mobile et son agent mère pour éviter ce scénario. Une analyse de la sécurité de Mobile IPv6 est disponible à [SSTIC]. Il faut retenir que la mobilité IPv6 est très prometteuse en terme d'exploitation, mais elle fait intervenir plusieurs acteurs au niveau protocolaire (réseau mère, réseau correspondant, système mobile), qui complexifient d'avantage les procédures de sécurisation, et qui, de fait, accroissent potentiellement les risques.

### 2.4.4 Des usages nouveaux

Plusieurs initiatives sont apparues, suite à IPv6, profitant de l'élan réformateur. La plupart sont, à la date de rédaction de document, en phase de discussion. *Shim6* est parmi les plus intéressantes. Il s'agit d'un protocole répondant au problème du *multihoming* : un site peut disposer de plusieurs liens Internet, gérés par différents fournisseurs d'accès. Chacun de ces liens fournit une adresse IP appartenant à un réseau donné. Il n'est actuellement pas

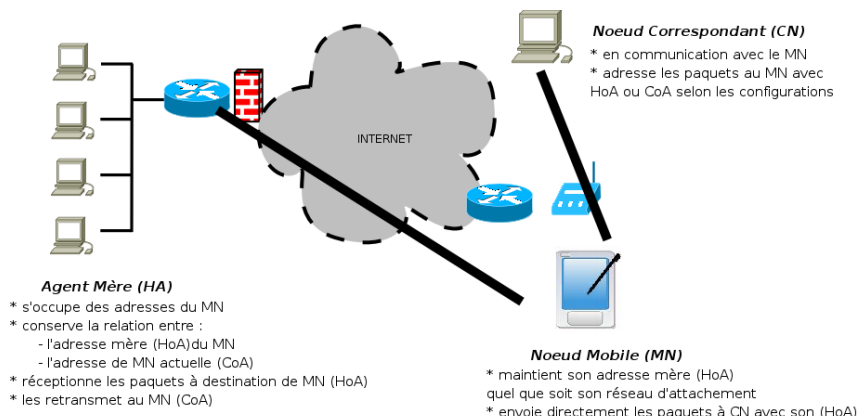


FIG. 4 – Exemple de communication sous MIPv6

possible de maintenir une communication, vers un serveur par exemple, quand le système change de lien. Shim6 propose donc de rendre plus indépendant les couches «réseau» et «transport» (modèle OSI), en distinguant deux adresses au lieu de la seule IPv6 : Les couches supérieures (transport, session, présentation, application) ne voient, selon Shim6, qu'une adresse fixe unique, appelée ULID (pour *Upper Layer Identifier*). L'ULID est une adresse IPv6. Des adresses locales maintenues servent à faire la transition quand le lien change, mais sont invisibles des autres couches protocolaires. L'ULID sert alors aux noeuds communicants pour maintenir la communication, sans se préoccuper du changement de lien du système. Ce changement de lien peut également être un changement de position d'un agent mobile, qui ne veut pas perdre sa communication au cours de l'entrée dans un nouveau réseau (roaming).

D'autres protocoles prometteurs verront sûrement le jour avec IPv6. Ils ne sont cependant pas encore matures pour être mis en œuvre, sans de très grandes précautions, dans un réseau de production.

### 3 Processus de traduction entre IPv4 et IPv6

#### 3.1 La double-pile IP, ou Dual Stack

La *double pile IP* consiste à équiper un équipement du réseau d'une double pile protocolaire (*Dual Stack*) et d'affecter une adresse IPv4 et une adresse IPv6 à l'interface. Cela peut s'appliquer sur la plupart des systèmes d'information. Les serveurs doivent alors avoir deux *sockets*, l'une correspondant à une écoute via IPv4, et l'autre correspondant à une écoute via IPv6.

Il faut donc garder en mémoire que, dans ce cas là, les deux protocoles sont installés sur le même système : ils communiquent directement entre eux et séparément avec l'extérieur. La section 5.4 montrera quelques problèmes de sécurité que cette propriété induit.

#### 3.2 Transport de IPv6 dans IPv4

Il n'est pas toujours possible d'avoir une double pile IP ou un réseau IPv6 de bout-en-bout. Cependant, les trames IPv6 doivent pouvoir être transmises, même si un réseau intermédiaire ne supporte qu'IPv4. Plusieurs solutions sont disponibles, pour former un *tunnel* : les paquets IPv6 transitent alors encapsulés dans IPv4, ce qui s'appelle autrement *un tunnel IPv4*. Nous distinguons les tunnels statiques et les tunnels automatiques.

##### Les tunnels statiques

La solution la moins souple consiste à établir un tunnel par le protocole GRE (*Generic Routine Encapsulation*), comme cela se fait déjà sous IPv4 pour d'autres protocoles. Le tunnel est statique, et il faut alors effectuer des modifications aux deux extrémités du tunnel. Il n'y a par ailleurs, tel quel, aucune garantie de sécurité (authentification, chiffrement, etc).

Au lieu de configurer manuellement chaque extrémité des tunnels, il est aussi possible d'automatiser un peu la procédure, tout en maintenant la structure statique du tunnel. Le principe est très similaire à celui d'un VPN et est illustré par la figure 5. Des serveurs, nommés *IP Tunnel Brokers* servent pour la transition. Il faut se

connecter à l'un d'eux en IPv4 pour obtenir une adresse et accéder à la configuration du tunnel vers un réseau IPv6. Ce procédé est bien statique (ou *semi-dynamique*), dans la mesure où il nécessite de connaître et de configurer correctement IPv4 au niveau du Tunnel Broker. Ce dernier se charge du routage et des configurations des extrémités des tunnels : il reste un point vulnérable dans la mesure où tout le transport du trafic repose sur lui.

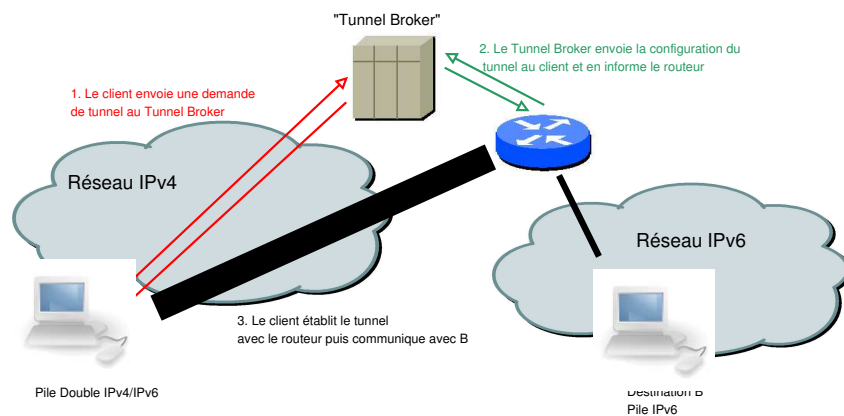


FIG. 5 – Exemple de tunnel statique sous la tutelle du Tunnel Broker

### 3.3 Tunnels automatiques : IPv6 dans IPv4

Dans le cas d'un tunnel automatique, une liaison fixe point à point est établie entre les machines impliquées (des routeurs par exemple). Ce tunnel formé fragmente les paquets selon IPv4, et met en œuvre les mécanismes de découverte des voisins.

Si une erreur survient au cours de l'acheminement IPv4, un paquet ICMPv4 est envoyé. Idéalement, le point à la source du tunnel devrait récupérer ce message, puis le traduire en un paquet ICMPv6 équivalent afin de le retourner vers la source du datagramme IPv6. Le transport de datagrammes IPv6 dans une trame IPv4 est précisé dans [RFC2893].

Il existe à l'heure actuelle quatre mises en œuvre majeures pour effectuer l'acheminement d'IPv6 sous IPv4 : 6to4, ISATAP, Teredo et 6over4.

**6to4 :** 6to4 utilise un principe d'encapsulation du trafic IPv6 dans des paquets IPv4. Chaque paquet IPv4 contient un protocole de numéro 41 (par exemple TCP a pour numéro 6, et UDP 17). Une adresse IPv6 est automatiquement attribuée dans le réseau 2002::/16 [RFC3056, RFC3068, RFC3964]. Un exemple (où l'adresse privée 192.168.10.34 remplace une adresse IPv4 publique) est présenté sur la figure 6. Une adresse IPv4 est également allouée (anycast 192.88.99.1) pour accéder aux routeurs relayant 6to4 vers des réseaux purement IPv6 et ne connaissant pas 6to4. Il s'agit d'une adresse annoncée par plusieurs réseaux disposant de serveurs de tunnels 6to4. Le système envoie donc ses paquets au serveur de tunnels le plus proche, qui le diffuse dans le réseau purement IPv6. Un exemple de tel scénario est donné par la figure 7.

**ISATAP :** ISATAP ( pour *Intra-Site Automatic Tunnel Addressing Protocol*) permet de créer automatiquement un tunnel et l'échange de flux IPv6 entre des systèmes ayant des piles IP doubles et interconnectées via un réseau IPv4. Il définit une méthode pour générer une adresse IPv6 locale et un mécanisme pour effectuer la découverte de proches voisins (*Neighbor Discovery*) par IPv4. Ainsi, lorsqu'un routeur ISATAP est installé, toute machine connaissant son adresse IPv4 peut le contacter. Il est donc impératif d'appliquer en parallèle des règles IPv4 de filtrage rigoureuses.

**Teredo :** Teredo est une extension de 6to4 avec traversée de NAT, utilisée par Microsoft. Il permet à un hôte connecté à un réseau IPv4 de communiquer en IPv6 avec l'extérieur, sans routeur particulier sur son réseau, mais aussi derrière un réseau IPv4 mettant en œuvre de la traduction d'adresses (NAT). Le principe consiste à créer un tunnel UDP en IPv4, qui a la possibilité de traverser les passerelles NAT standards. Ce n'est pas le cas de 6to4, dont le protocole 41 est rarement considéré. Une machine Windows utilise Teredo lorsqu'elle ne dispose pas de connectivité IPv6 native, ni de 6to4 ou ISATAP. Au démarrage, un client Teredo doit obtenir une adresse IPv4 de relais IPv6 auprès d'un serveur Teredo (hébergé par Microsoft par exemple). Une fois celle-ci obtenue, il peut lui envoyer les données IPv6 qu'il transmettra à la destination en IPv6. Un schéma

simplifié se trouve sur la figure 8. Le port d'écoute du serveur et du relais Teredo est le 3544 UDP. Du point de vue de la sécurité, il est important de noter que cela implique un nouvel accès ouvert au niveau du pare-feu : le client doit en effet régulièrement émettre des paquets UDP pour entretenir la connexion au niveau du routeur afin que celui-ci ne nettoie pas sa table NAT et que le serveur Teredo puisse lui envoyer des paquets si besoin.

**6over4** : éthernet virtuel sur le *multicast* IPv4 [RFC2529] : cette méthode est relativement simple mais repose sur la capacité *multicast* d'IPv4. Celle-ci n'est cependant pas supportée par toutes les infrastructures, ce qui rend cette solution assez marginale et peu supportée.

Le tableau 2 résume les points essentiels qui différencient chacune de ces méthodes d'encapsulation.

Méthodes	Caractéristiques
Tunnel Brokers	<ul style="list-style-type: none"> <li>⊕ pour les routes statiques</li> <li>- 3 composants : client, <i>Tunnel Broker</i>, <i>Tunnel Server</i></li> <li>* Le <i>Tunnel Broker</i> est souvent hébergé par un tiers</li> <li>* Le <i>Tunnel Broker</i> choisit la configuration du tunnel</li> <li>* Le <i>Tunnel Broker</i> choisit les adresses IPv6 et le serveur</li> </ul>
6to4	<ul style="list-style-type: none"> <li>⊕ pour interconnecter des îlots IPv6 d'un réseau</li> <li>* Destiné aux connexions de site-à-site</li> <li>* Le routeur de bordure doit disposer d'une adresse publique IPv4</li> <li>* TLA (<i>Top-Level Aggregation ID</i>) des adresses IPv6 : 2002::/16</li> <li>* Préfixe de l'adresse : combinaison du TLA et de l'adresse IPv4</li> <li>* Les routeurs 6to4 échangent les préfixes des sites IPv6</li> <li>* Peut utiliser des routeurs relais <i>publics</i> (192.88.99.1)</li> </ul>
ISATAP	<ul style="list-style-type: none"> <li>⊕ pour connecter des clients isolés en IPv6</li> <li>* Fonctionne avec les doubles-piles IP</li> <li>* Conversion d'adresses : <math>adresse\_IPv6 = prefixe\_IPv6::5efe:adresse\_IPv4</math></li> <li>* Compatible avec d'autres solutions (6to4, doubles-piles IP, etc)</li> </ul>
6over4	<ul style="list-style-type: none"> <li>⊕ pour connecter des machines IPv6 isolées sans tunnel explicite</li> <li>* Une des premières solutions pour construire des tunnels</li> <li>* Aussi appelé <i>Virtual Ethernet</i></li> <li>* Suppose un domaine IPv4 Multicast</li> </ul>
Teredo	<ul style="list-style-type: none"> <li>⊕ pour connecter en IPv6 des machines NATées</li> <li>* Seule solution dynamique permettant de fonctionner avec un réseau IPv4 <i>NATé</i></li> <li>* Ne nécessite pas d'adresse IPv4 publique pour la machine source</li> <li>* Fonctionne sur UDP (port 3544), et non plus sur IPv4 directement (proto 41)</li> <li>* Fonctionne avec les doubles-piles IP</li> <li>* Plusieurs composants, dont : <i>Teredo Server</i> et <i>Teredo Relay</i></li> <li>* Ces composants sont actuellement gérés par un tiers (Microsoft par ex.)</li> </ul>

TAB. 2 – Méthodes d'encapsulation usitées

### 3.4 Transport de IPv4 dans IPv6

Le transport de datagrammes IPv4 ou IPv6 dans une trame IPv6 est précisé dans [RFC2473]. L'entête indique alors le protocole qui est encapsulé, 4 pour IPv4, et 41 pour IPv6. Le RFC 2473 propose aussi des mécanismes pour éviter le bouclage dû à l'imbrication de protocoles.

La traduction se fait relativement bien, car IPv6 est suffisamment souple en terme d'extension. Ce scénario est cependant encore assez rare, à l'exception du cœur des réseaux de certains fournisseurs d'accès.

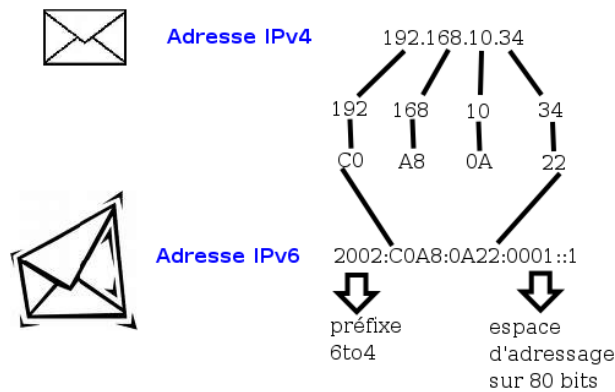


FIG. 6 – Méthode de traduction d'adresse vue par 6to4

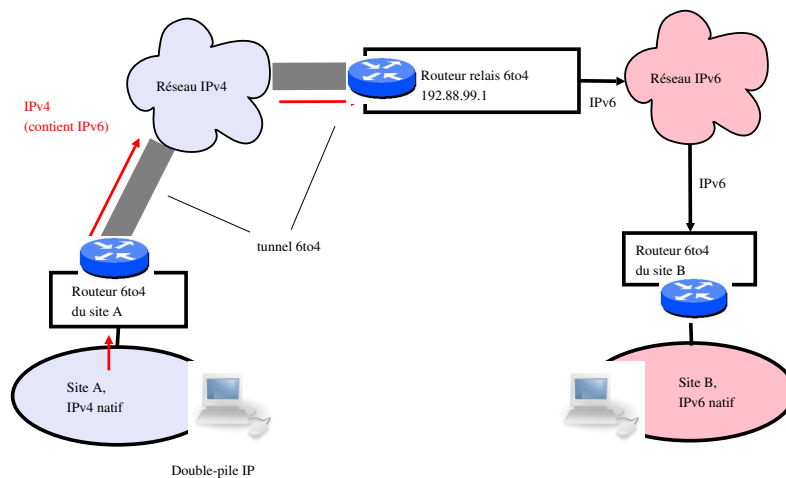


FIG. 7 – Scénario 6to4

### 3.5 La continuité du service DNS

Le protocole de résolution de noms DNS ne subit que très peu de modifications avec l'arrivée d'IPv6. Il doit supporter l'enregistrement AAAA, qui nomme une adresse IPv6, à l'instar de l'enregistrement A pour une adresse IPv4. Il doit pouvoir également supporter la résolution inverse en IPv6 (`ip6.arpa.`). Ainsi, l'arrivée d'IPv6 entraîne des problèmes d'incompatibilité. Avant IPv6, la résolution de noms DNS ne faisait intervenir qu'IPv4, et le service était donc garanti pour tous les clients DNS. Avec l'arrivée d'IPv6, l'espace de nommage devient fragmenté en des partitions accessibles via IPv4 uniquement et d'autres accessibles en IPv6 uniquement. Il est alors possible d'imaginer le scénario suivant :

**Scénario :** un client DNS se trouve dans un réseau ne supportant qu'IPv4. Il souhaite résoudre une requête DNS relative à une zone hébergée sur des serveurs ne supportant qu'IPv6. En théorie, la résolution n'aboutira pas.

La réciproque est également valable. Il existe donc quelques problèmes de continuité qui ne sont pas clairement résolus pour le moment. Certains sont cités dans [IETF]. Ce rapport fait mention des difficultés à mettre en œuvre la résolution inverse DNS, dans le cas des tunnels. Il donne aussi quelques comportements singuliers des serveurs liés à `getaddrinfo()`, la fonction chargée de lancer les requêtes DNS depuis un poste client. Celle-ci cherche d'abord un enregistrement AAAA IPv6, ou sinon, après réponse négative, essaie de trouver un enregistrement A IPv4. Mais certains serveurs ne répondent pas systématiquement à la première requête, pouvant alors provoquer des délais d'attente importants côté client.

En l'état, les solutions proposées consistent à utiliser un serveur DNS double-pile IPv4/IPv6, ou deux serveurs supportant les deux protocoles.

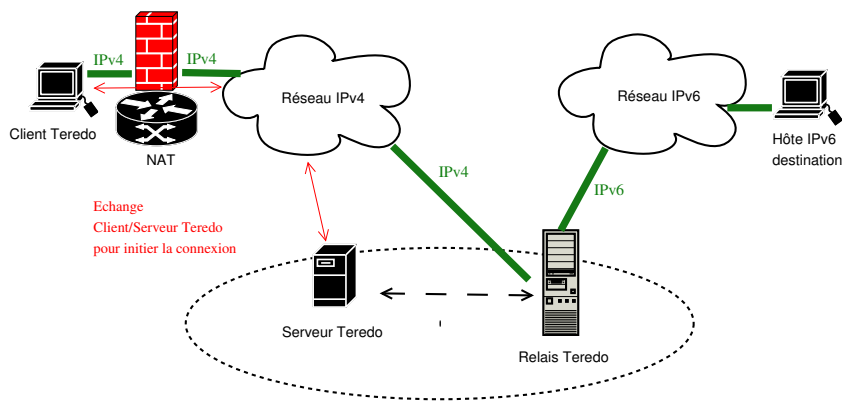


FIG. 8 – Exemple Teredo

### 3.6 Des groupes de travail

Le groupe de travail, NGtrans, maintenant dissout, a écrit un ensemble de mécanismes de transition, permettant chacun de résoudre un problème particulier. Ils répondent à un bon nombre de questions, même si les solutions proposées ne sont pas toujours aussi techniques qu'il faudrait. La plupart sont illustrées de scénarios de transition et de cohabitation. Ils abordent notamment les thèmes suivants :

- double pile IPv6
- relais applicatifs
- interconnexion de réseaux IPv6 isolés
- transport de trafic IPv6 dans IPv4
- etc.

Un groupe de travail IETF, nommé IPv6ops (pour *IPv6 operations*) vise à traiter de l'ensemble des problèmes opérationnels liés à la migration vers IPv6. Ils reprennent donc en partie les thèmes du groupe de travail précédent, mais en distinguant cette fois quatre grandes familles de déploiement :

- les réseaux de fournisseurs d'accès (FAI)
- les réseaux d'entreprises
- les réseaux personnels (SOHO)
- les réseaux mobiles (UMTS 3G, etc)

L'ensemble des documents de ces groupes de travail est disponible aux deux adresses réticulaires ci-dessous :

[http://www.6bone.net/ngtrans/ngtrans\\_project-status.html](http://www.6bone.net/ngtrans/ngtrans_project-status.html)

<http://www.ietf.org/html.charters/v6ops-charter.html>

## 4 La sécurité considérée dans IPv6

### 4.1 IPsec par défaut

IPv6 supporte par défaut les deux architectures classiques d'IPsec : AH et ESP.

L'extension d'authentification, ou AH, assure l'authentification et l'intégrité des données. L'émetteur calcule une *signature* sur un datagramme et l'émet avec le datagramme sur lequel elle porte. Le récepteur récupère cette valeur et vérifie qu'elle est correcte. Cette signature peut s'appuyer sur des clés asymétriques et éventuellement des certificats.

L'extension ESP (pour *Encryption Security Payload*) complète la précédente pour offrir la confidentialité des données. Elle permet de chiffrer l'ensemble des paquets (entête IPv6 comprise) ou seulement leur partie *transport* (tout ce qui se trouve après l'entête IP), selon les modes dits respectivement *tunnel* et *transport*.

Avant l'authentification ou le chiffrement de données IP, l'émetteur et le receveur doivent convenir des algorithmes et des clés à utiliser. Ceci se fait par le protocole IKE (pour *Internet Key Exchange*, [RFC2409]). Sans

décrire les détails de ce dernier, il est important de se souvenir que ce protocole ne convient pas, dans l'état actuel, aux échanges *multicast*. Des initiatives dans ce domaine se confirment, comme GSAKMP pour la distribution de clés, et *tesla* pour l'authentification, mais les couches IPv6 actuelles ne les utilisent pas encore pour le *multicast*. Le trafic *multicast* passe donc en clair, pour la majorité des cas, dans le réseau, à moins de considérer une procédure d'authentification et de chiffrement dans les niveaux protocolaires supérieurs.

## 4.2 SEND : Sécuriser la découverte du voisinage NDP

SEND propose de sécuriser NDP, le protocole de voisinage d'IPv6 (cf. section 2.4.1). La première proposition est l'utilisation de CGA ([RFC3972]) (pour *Cryptographically Generated Addresses*), qui crée des identifiants d'adresses à partir d'une clé publique. Ce mécanisme permet de vérifier l'identité de la machine émettrice du paquet NDP. L'adresse CGA s'obtient en appliquant une fonction de hachage à un ensemble de paramètres. L'algorithme de génération d'adresse CGA repose en grande partie sur une attribution pseudo-aléatoire à l'un d'entre eux (le paramètre *Modifieur* sur 128 bits). Il est important de vérifier que celle-ci s'effectue correctement. Il prend également en considération d'éventuelles collisions. Si trois ou plus sont détectées, l'algorithme s'interrompt. A l'adresse CGA s'ajoutent quelques options :

- une option RSA : pour signer le paquet, et ainsi vérifier l'intégrité et l'authenticité du paquet du côté destinataire.
- une option d'horodatage (*timestamp*) : pour limiter les possibilités de *rejeu*. Cela impose néanmoins une synchronisation préalable des machines.
- une option unicité (*Nonce*) : pour identifier les messages d'association de type Demande/Réponse.
- une option de découverte de certification sur le réseau.

Ce mécanisme de sécurisation ne NDP n'est pas encore déployé, et reste anecdotique. Cependant, une mise en œuvre du protocole SEND en source libre est offerte par Docomo à l'adresse réticulaire suivante :

- [http://www.docomolabs-usa.com/lab\\_opensource.html](http://www.docomolabs-usa.com/lab_opensource.html)

## 4.3 Pare-feux et filtrage

Plusieurs outils de filtrage existent, et sont mis à disposition dans les systèmes d'exploitation avec la couche IPv6. Parmi ceux-ci :

- Sous Linux, il existe *ip6tables*, dont l'usage est très similaire à celui de son prédécesseur *iptables* pour IPv4 ;
- Dans les versions BSD, *pf* permet de filtrer IPv6 de manière assez simple (voir les exemples fournis dans le fichier *pf.conf*) ;
- Mac OS s'appuie sur l'utilitaire *ipfw* (*ip6fw*) pour offrir le filtrage IPv6.
- Microsoft fournit également un pare-feu à partir de la version XP SP1, nommé *Internet Connection Firewall* puis *Windows Firewall*. L'utilisateur définit une seule configuration, qui s'applique pour IPv4 et IPv6.

Les règles de filtrage se basent souvent sur les champs suivants :

- adresses source et destination ;
- les protocoles supérieurs ;
- les ports source et destination (niveau transport) ;
- champs tels que les labels de flux (*flow labels*) ou les classes de trafic (*traffic class*).

Cependant, les différentes options de tunnel offertes par IPv6 sont difficilement contrôlables au niveau d'un pare-feu, et les règles simples de connectivité (ports source/destination) sont facilement contournables. Les pare-feux ne gardent souvent pas les détails de l'état des connexions.

Enfin, comme il a été rapidement mentionné ci-dessus, les extensions du protocole IPv6 sont évolutives et peuvent provoquer des problèmes décisionnels. Les documents RFCs ne donnent que très peu de directives sur les actions à entreprendre pour le filtrage IPv6.

Les «systèmes de détection d'intrusions» (IDSs) se mettent progressivement à IPv6. Pour ceux qui s'appuient sur des ensembles de signatures, le changement de code n'est pas majeur. Ils doivent pouvoir capturer et analyser des paquets aux différents formats IPv6. Cependant, il y a encore peu d'attaques connues ciblant les nouveaux protocoles associés à IPv6. L'ensemble de règles reste encore à écrire. Sans cela, l'intérêt des IDSs reste minime. L'usage d'IPsec va aussi compliquer la tâche des sondes surveillant le trafic du réseau. Si les paquets sont chiffrés, il sera plus difficile d'y voir des indications de code malveillant. A l'heure actuelle, la plupart des outils IDSs se bornent à détecter la présence de paquets IPv6 dans le réseau, sans regarder la sémantique ni le contenu : leur action est encore loin d'être satisfaisante.

Les pare-feux et IDSs compatibles IPv6 sont encore en phase de développement. Il est néanmoins raisonnable de prévoir que le besoin de sécurisation des postes terminaux va s'accroître, compte tenu de la difficulté de cette tâche au niveau des équipements du réseau.

## 5 Problèmes et risques

### 5.1 Les piles protocolaires existantes

#### 5.1.1 Trouver IPv6 dans son système

Le code IPv6 est disponible dans les équipements terminaux récents (PCs, imprimantes, etc) et dans les routeurs. Cette phase de développement est achevée pour la plupart des équipementiers. Les fabricants de systèmes d'interconnexion ont intégré IPv6 dans leurs systèmes d'exploitation. Une liste de systèmes est disponible à l'adresse réticulaire suivante:

<http://www.ipv6.org/impl/>

#### 5.1.2 Existence de problèmes de mise en œuvre

Cependant, plusieurs avis du CERTA tendent à indiquer que la mise en œuvre des différentes piles protocolaires (ou des applications les utilisant) présentent encore des faiblesses. Le tableau 1 liste quelques avis du CERTA à ce sujet, visibles sur son site Internet :

Référence CERTA	Titre de l'avis	Date de parution
CERTA-2003-AVI-119	« Vulnérabilité IPv6 dans Solaris 8 »	24/07/2003
CERTA-2004-AVI-028	« Vulnérabilité IPv6 dans les noyaux BSD »	20/02/2004
CERTA-2004-AVI-222	« Vulnérabilité dans le JunOS de Juniper »	06/07/2004
CERTA-2006-AVI-313	« Vulnérabilités du serveur http Apache 2.0.x »	08/10/2004
CERTA-2005-AVI-014	« Multiples vulnérabilités dans Exim »	17/02/2005
CERTA-2005-AVI-032	« Vulnérabilité IPv6 dans Cisco IOS »	27/01/2005
CERTA-2005-AVI-309	« Vulnérabilité de la pile IPv6 des équipements Cisco »	11/08/2005
CERTA-2006-AVI-289	« Vulnérabilité IPv6 dans JunOS de Juniper »	12/07/2006
CERTA-2006-AVI-414	« Multiples vulnérabilités dans Sun Solaris »	29/09/2006

TAB. 3: Quelques avis du CERTA concernant IPv6

Il est important de noter que toutes les mises en œuvre d'IPv6 conformes doivent intégrer IPsec. Cependant, la plupart de celles actuelles ne l'utilisent pas convenablement. L'une des raisons étant que les échanges en *multicast* ne sont pas adaptés aux protocoles de gestion des associations liés à IPsec. Les extensions ESP et AH ne sont également pas compatibles avec la traduction d'adresses/ports. Le fait que ces deux extensions ne soient pas systématiquement applicables et appliquées provoque une certaine méfiance quant à la garantie des principes de confidentialité, d'intégrité et d'authentification dans un usage courant des protocoles. Il faut aussi noter que certains produits (routeurs, passerelles) offrent une version d'IPsec non fonctionnelle avec IPv6. Ils affichent donc bien commercialement les deux protocoles, mais il apparaît par la suite qu'IPsec ne fonctionne que sous IPv4.

Une autre question est celle de la priorité, pour les systèmes mettant en œuvre à la fois IPv4 et IPv6. Dans le cas de la découverte réseau, ceci peut se faire, comme nous l'avons vu, sous IPv6 par le *Neighbor Discovery*, ou sous IPv4 par le biais de Netbios. Dans ces conditions, il n'existe pas nécessairement de procédures pour forcer la communication par l'un ou l'autre des protocoles.

### 5.2 Les risques existants avec IPv6

Ce chapitre recense brièvement les premiers risques associés au déploiement IPv6 qui ont été rapportés, soit sous forme d'outils, soit sous forme de concept.

Une des toutes premières utilisations malveillantes concernant IPv6 concerne les portes dérobées. Une fois qu'une machine est compromise, le code malveillant active la couche IPv6 présente sur la machine, mais non utilisée avant la compromission. Le code ouvre ensuite une porte dérobée via IPv6 [USCERT]. De cette manière, la porte dérobée a peu de chances d'être détectée par des tests d'audit traditionnels. Par ailleurs, le trafic IPv6 se joue assez aisément des pare-feux et systèmes de détection d'intrusions encore largement incompatibles avec IPv6.

Il existe déjà dans le domaine public des outils exploitant certaines vulnérabilités du protocole IPv6, que ce soit des faiblesses protocolaires, des erreurs de mise en œuvre ou des problèmes de configuration.

Cela inclut :

- des scanners IPv6 permettant d'identifier toutes les machines qui ont activé IPv6, ainsi que les ports ouverts.
- des outils pour créer différents types de tunnels, permettant aux personnes malveillantes, mais pas seulement, de véhiculer des données par des canaux cachés (nouveaux protocoles, ports, chiffrement, etc).
- des outils pour lancer des dénis de service, en inondant de paquets certains ports de la machine, afin de perturber son fonctionnement ; les transitions IPv4 vers IPv6, coûteuses en ressources, sont pour le moment les plus vulnérables à ce genre d'attaques.
- des outils d'usurpation de paquets de découverte de voisinage ICMP pour les attaques de type «homme-aux-milieu».
- des techniques de dénis de service, qui empêchent l'accès d'un système au réseau (medium) ; une personne malveillante peut confectionner une réponse bloquante, quand le système envoie son adresse provisoire (ou DAD, pour *Duplicate Address Detection*) afin de vérifier qu'aucun autre système l'utilise.
- des routeurs fictifs injectant de mauvaises routes et devenant *de facto* des routeurs de référence (connus par les autres systèmes).
- des outils réactifs : quand un nouveau système rejoint le réseau IPv6, un mécanisme lance rapidement un script.
- des outils gênants, qui réduisent la taille des paquets dans les échanges (*Maximum Transmission Unit* ou MTU) avec autre système afin de limiter ses services.
- des outils d'amplification de trafic (*smurf*) qui consistent à envoyer en *multicast* différents paquets, suscitant chacun une réponse systématique des machines voisines. Le RFC 2463 précise qu'aucune réponse ICMP ne doit être émise si le destinataire du paquet est une adresse *multicast*. Mais une vulnérabilité, désormais corrigée, dans des produits Cisco, rappelle que les mises en œuvre des couches IPv6 ne sont pas nécessairement en accord avec les RFCs.
- etc

### 5.3 Les vers, une fin annoncée?

L'arrivée d'IPv6 tend à faire penser que les vers virulents qui se propagent sur l'Internet vont disparaître, ou du moins se raréfier. Cet argument s'appuie sur le fait que les adresses IP passent de 32 à 128 bits, augmentant la taille de l'espace d'adressage d'un facteur  $2^{96}$ . Il est donc beaucoup moins évident de balayer l'ensemble des adresses afin de trouver celles potentiellement vulnérables. Les principales techniques de propagation actuelles des vers sont : le balayage exhaustif, ou le balayage pseudo-aléatoire. Dans de telles conditions, la probabilité de voir sa propre machine touchée serait donc bien plus faible, et les vers utilisant ces techniques auront davantage de difficultés à se propager dans l'espace IPv6. Malheureusement, des chercheurs de l'Université de Columbia ont récemment montré que cette idée n'était pas complètement valable. En effet, ils démontrent que les vers auront aussi de nombreuses sources d'information à leur disposition, leur permettant de déterminer les réseaux existants, puis de se propager localement au sein de ceux-ci. Ils citent entre autres :

- la table des voisins de réseau, fournissant leurs adresses physiques ainsi que leurs adresses IPv6 ;
- les tables de routage créées par les nouveaux protocoles de routage (OSPFv6, ou RIPng) ;
- la prévision des adresses IPv6 au cours de la configuration sans état. Celle-ci s'appuie sur l'identifiant de l'interface, spécifique au fabricant de la carte. L'espace des possibilités est donc considérablement réduit ;
- l'observation des réponses aux requêtes *multicast* ainsi que celles utilisées pour la découverte de services (anycast, SLP, DNS, DHCP) permet d'avoir une vue locale du réseau ;
- les journaux de serveurs qui donnent un bon aperçu des adresses actives ;
- les protocoles de pair-à-pair (P2P), très bavards ;
- etc

La propagation des vers sera sûrement moins rapide que dans le monde IPv4, mais il est illusoire de croire en leur disparition. Par ailleurs, la phase de transition entre IPv4 et IPv6 pourrait voir apparaître de nouvelles formes virales, profitant des couches double-pile pour pénétrer les systèmes.

### 5.4 Problématiques du filtrage

Comme il a été montré dans la section 4.3, des outils de filtrage existent et sont fournis avec les distributions des systèmes d'exploitation les plus courants. Le filtrage en lui-même est cependant plus difficile qu'avec IPv4, et

nécessite patience et expérience. En effet, les machines peuvent maintenant présenter deux portes d'entrée, comme l'illustre la figure 9. Ainsi, quand un service se lie à un port IPv6, il sera par défaut lié au port IPv4 correspondant.

Le filtrage doit alors en tenir compte, pour éviter des transgressions de la politique de sécurité. Il est très facile d'imaginer un pare-feu ne bloquant pas ICMPv6. Sous cette condition, une personne malveillante peut cartographier le réseau depuis l'extérieur en adressant quelques paquets ICMPv6 aux machines. Plusieurs questions viennent naturellement à l'esprit et les réponses ne sont pas toutes évidentes. Voyons plus en détails les scénarios suivants :

- Une machine Linux intègre une double-pile (*dual stack*) : `iptables` est utilisé pour configurer les paquets IPv4, et `ip6tables` est utilisé pour filtrer les paquets IPv6. Ces deux outils sont indépendants. Il n'y a alors aucun moyen de vérifier automatiquement la cohérence de règles entre les deux.
- Une machine Windows Vista contient une pile unique, intégrant à la fois IPv4 et IPv6. Si le pare-feu reste semblable à celui de XP, le filtrage ne peut se faire qu'au niveau transport (TCP/UDP), en choisissant les ports à bloquer. La règle s'applique donc aux deux portes d'entrée de la figure 9. Imaginons maintenant un réseau en IPv6 ou en IPv4. Cela revient à dire que tout serveur (ou toute machine ayant un port ouvert), sera accessible par IPv6 ou IPv4. La politique de sécurité n'est alors pas respectée, à moins d'ajouter un filtre sur le réseau et devant le serveur pour compléter celui de Microsoft Windows et préciser la version IP autorisée.

Ces scénarios sont donnés à valeur illustrative. Il existe pour chacun des solutions techniques, mais elles ne sont pas intuitives, et doivent être adaptées. Les codes existants sont rares.

Les outils de filtrage actuels ne sont pas adaptés aux tunnels «IPv4 dans IPv6» et «IPv6 dans IPv4». Imaginons le scénario où une machine IPv6 veut créer un tunnel IPv6 traversant le réseau IPv4 dans lequel elle se trouve. Elle peut utiliser la solution du Tunnel Broker présentée en figure 5. L'administrateur veut s'assurer au niveau réseau que la machine effectuant le tunnel communique en IPv6 avec le bon destinataire. Il n'y a pas de technique simple pour filtrer les paquets d'encapsulation IPv4 contenant les paquets IPv6 avec la bonne source et la bonne destination en IPv6. Le Tunnel Broker peut s'en charger au cours du transfert, mais il n'est pas forcément intégré dans le réseau et peut appartenir à un tiers. La politique de sécurité s'applique pourtant autant au trafic encapsulé que celui IPv4. Il est envisageable d'avoir un mécanisme simple, qui décide, à l'aide d'`iptables` par exemple, de donner un paquet IPv6 encapsulé à `ip6tables` pour appliquer les règles IPv6 qui conviennent, mais de telles mises en œuvre ne sont pas encore déployées.

Le filtrage doit être considéré avec la plus grande attention pour respecter les politiques d'accès rigoureuses déjà mises en œuvre. L'intégration d'IPv6 ne doit pas modifier, ou rendre plus laxistes, celles-ci.

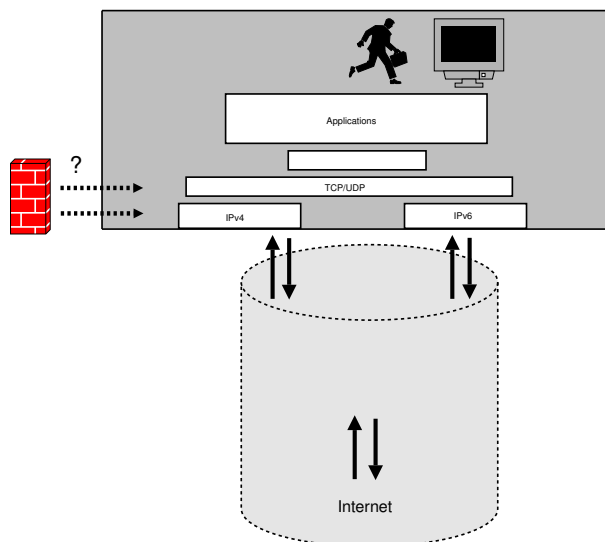


FIG. 9: Les problématiques du filtrage

## 5.5 Les applications

Les applications devraient en théorie être indépendantes des couches protocolaires plus basses. Ce n'est malheureusement pas toujours le cas : certaines catégories d'applications comme le serveur de messagerie, le web

doivent aussi subir des modifications. Les plus courantes (sendmail, exim, ssh, apache, mozilla, openLDAP, squid, etc) supportent maintenant nativement IPv6. Certaines applications peuvent encore poser problème. La stratégie adoptée lorsque le système utilise les deux piles est mis en œuvre localement dans chacune des applications. Un schéma standard est le suivant : l'application cherche à résoudre en IPv6 un nom (DNS AAAA). Si elle y parvient, elle se connecte alors en IPv6. Dans le cas où l'une des deux actions précédentes échoue, l'application réessaie en IPv4. [JRES05]. Elle est donc vulnérable à certains dysfonctionnements DNS, aussi appelés trous noirs :

- le serveur ne répond pas à la requête DNS AAAA, forçant l'application à abandonner au bout de quelques dizaines de secondes d'attente.
- le serveur répond par un message d'erreur indiquant que le domaine auquel appartient le nom n'existe pas, au lieu de répondre qu'il n'y a pas d'adresse IPv6 associée au nom demandé.

Très souvent, il existe des relais applicatifs (ou *serveurs mandataires*) permettant de faciliter la conversion, pour permettre d'accéder à un service au réseau IPv4 inaccessible en IPv6 (et inversement). L'équipement IPv6 émet sa requête vers le relai applicatif qui interprète le contenu de celle-ci et la retransmet en IPv4 vers le serveur d'application légitime. Un ou plusieurs relais peuvent être installés en fonction des applications et des services disponibles. Cette solution a l'avantage de ne pas modifier les serveurs existants. En revanche, il n'y a plus de connexion point-à-point entre le client et le serveur, étant donné que le relai applicatif interprète les données échangées. Certains services ne peuvent fonctionner sans cette condition (le plus connu étant `telnet`).

## 6 Recommandations du CERTA

### 6.1 Les 10 premières recommandations

La première recommandation, avant toute autre, est de prendre pleinement conscience qu'IPv6 est maintenant présent dans les systèmes d'exploitation. La politique de sécurité doit par voie de conséquence l'intégrer.

De manière plus générale, il est, dans la mesure du possible, conseillé de suivre les points décrits dans la table ci-dessous :

1. Filtrer toutes les adresses IPv6 du réseau au niveau du pare-feu de sortie vers l'Internet.  
ex : ne pas oublier qu'une machine utilise plusieurs adresses IPv6.
2. Employer des adresses IPv6 conformes, mais pas prévisibles pour les systèmes critiques.
3. Bloquer tous les services inutiles au niveau du pare-feu.  
ex. : pour éviter les tunnels sortants, bloquer le protocole 41 des paquets IPv4.  
ex. : pour interdire Teredo, bloquer les ports UDP 3544 (src/dst).
4. Filtrer de manière sélective les paquets ICMP (ICMPv4 et ICMPv6).  
ex : distinguer les différents type de messages (erreurs et information, adressage, etc)
5. Interdire la fragmentation des paquets au niveau IPv6, si cela est possible.  
Sinon, celle-ci doit être correctement filtrée (tailles min et max).
6. Sélectionner les extensions des entêtes IPv6 qui ont légitimité dans le réseau.  
ex : faut-il autoriser les jumbogrammes, ou l'extension *Router Alert* ?
7. Utiliser IPsec pour les communications avec des systèmes critiques, et pour les protocoles de routage.  
L'authentification doit être déployée dès que possible, via IPsec et IEEE 802.1X (Radius, etc).
8. Renforcer la sécurité au niveau des stations (antivirus, pare-feu, services nécessaires strictement).
9. Garder les bonnes pratiques IPv4 pour appliquer les correctifs de sécurité et les mises à jour.
10. Développer des surveillances adaptées à IPv6 (blocage de périmètres, surveillance du volume, etc).  
Vérifier que les outils de sécurité utilisés comprennent correctement IPv6.

### 6.2 Désactivation possible

Si la migration vers IPv6 est reportée, dans l'attente d'une meilleure visibilité, il est fortement recommandé de désactiver les piles IPv6 et les protocoles associés. Il est aussi conseillé de limiter les installations des applications aux piles protocolaires existantes (fonctionnement sous IPv4 uniquement).

Les paragraphes suivants donnent quelques indices pour désactiver les piles protocolaires non utiles sur les systèmes d'exploitation les plus fréquents.

Comme le support d'IPv6 est expérimental sous Windows XP, la configuration se fait en ligne de commandes avec `netsh`. Il est désactivé par défaut. Pour s'en assurer et le désactiver si ce n'est pas le cas, il suffit de :

1. Lancer une console (Démarrer -> Programmes -> Accessoires -> Invite de commandes)

2. Taper `netsh int ipv6 uninstall`, puis la touche 'Entrée'
3. fini!

La plupart des systèmes BSD, que ce soit BSD/OS, FreeBSD ou OpenBSD disposent d'IPv6 depuis plusieurs années (FreeBSD depuis sa version 4.0). Actuellement, ces systèmes proposent IPv6 en standard. Si IPv6 n'est pas activé à l'installation, il est possible de rappeler le programme de configuration `/stand/sysinstall` pour reconfigurer les interfaces. En revanche, si IPv6 est installé, il est plus difficile de le faire disparaître *a posteriori*. Cela doit passer par la recompilation du noyau.

Les versions MAC OS intègrent IPv6, à partir de la version Jaguar 10.2. Le système d'exploitation attribue une adresse IPv6 par défaut à l'interface réseau, qui est joignable de l'extérieur. Si IPv6 n'est effectivement pas utilisé, il est recommandé de le désactiver :

- Ouvrir *Préférences Système* puis cliquer sur *Réseau*
- Si la ligne « Adresse IPv6 : » est complétée, cliquer sur *Configurer IPv6*
- Sélectionner *Non*
- Fini!

Pour les versions récentes de Linux (noyau 2.6.X), IPv6 est installé par défaut. A moins de recompiler le noyau, il n'est pas évident de le désactiver complètement. En revanche, la plupart des distributions Linux (Debian, Ubuntu, Fedora) intègrent nativement un outil d'administration pour le filtrage de paquets IPv6, nommé `ip6tables` (équivalent de l'utilitaire `iptables` pour IPv4). Il est recommandé de filtrer IPv6 si celui-ci n'est pas utilisé. Cela peut se faire par la série de commandes suivantes :

- `ip6tables -F`
- `ip6tables -P INPUT DROP`
- `ip6tables -P OUTPUT DROP`
- `ip6tables -P FORWARD DROP`

Les outils de surveillance traditionnels (`tcpdump` et `ethereal`) supportent également IPv6, et interprètent l'ensemble des options (extensions) déployées. Ils permettent de comprendre plus précisément les activités IPv6 du réseau. Le document [SANSv6] liste sous forme de mémento les détails de filtrage possibles.

## 7 Conclusion

IPv4 ne disparaîtra pas brutalement du monde Internet. A la date de rédaction de ce document, sa disparition totale n'est d'ailleurs pas envisagée. Cependant, nous sommes actuellement dans une période transitoire, où IPv6 émerge. Ce protocole, prometteur, introduit de nombreux changements. Ce document se focalise sur les implications en terme de sécurité de ce dernier, sans toutefois remettre en cause les avantages certains qu'il présente et qui justifie son développement.

Comme il a été montré dans ce document, l'étape de standardisation des protocoles de base d'IPv6 peut être considérée comme achevée. IPv6 résoud plusieurs problèmes de sécurité, mais, malgré cela, plusieurs demeurent, et certains sont directement liés à cette phase transitoire.

De manière schématique, la sécurité IPv6 en l'état n'est pas nécessairement meilleure que celle IPv4. Les mécanismes d'attaques sont très semblables, et les nouveaux protocoles autour d'IPv6 ne changent pas ces vulnérabilités. D'autres peuvent également apparaître, du fait de la jeunesse et de la qualité de mise en œuvre des protocoles associés.

Il est important, dès maintenant, d'envisager ce tournant avec le plus grand soin, et de ne pas oublier qu'IPv6 fait déjà partie de nos systèmes d'information. La SSI doit donc le considérer comme tel, et prendre les mesures nécessaires afin d'obtenir à terme une transition maîtrisée.

## 8 Documentation

### Références

- [IETF] A. Durand, J. Ihren, P. Savola. "Operational Considerations and Issues with IPv6 DNS", IETF Draft : <http://www3.ietf.org/proceedings/06mar/IDs/draft-ietf-dnsop-ipv6-dns-issues-12.txt>
- [IPv6TF] Site de l'IPv6 Task Force France : <http://www.fr.ipv6tf.org/>

- [JRES05] M. Herrb. "Quelques points durs d'IPv6". Présentation faite à JRES 2005, Marseille, France :  
<http://www.jres.org/slides/72.pdf>
- [NetBSD] Documentation NetBSD : Réseau IPv6 de NetBSD :  
<http://www.netbsd.org/fr/Documentation/network/ipv6/>
- [NISCC] "Security considerations fro IPv6", *NISCC Technical Note*, 24 avril 2006 :  
<http://www.niscc.gov.uk>
- [OPS6] Groupe de travail IETF IPv6ops, IPv6 Operations :  
<http://www.ietf.org/html.charters/v6ops-charter.html>
- [Point6] Version électronique du livre "IPv6, théorie et pratique" (ISBN 284177337X), *Editions O'Reilly* :  
<http://livre.point6.net/>
- [Renater] IPv6 utilisé dans le réseau Renater :  
<http://sem2.renater.fr/ipv6/>
- [RFC791] "RFC 791 : Internet Protocol" :  
<http://www.ietf.org/rfc/rfc0791.txt>
- [RFC1519] "RFC 1519 : Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy" :  
<http://www.ietf.org/rfc/rfc1519.txt>
- [RFC2080] "RFC 2080 : RIPng for IPv6" :  
<http://www.ietf.org/rfc/rfc2080.txt>
- [RFC2409] "RFC 2409 : The Internet Key Exchange (IKE)" :  
<http://www.ietf.org/rfc/rfc2409.txt>
- [RFC2460] "RFC 2460 : Internet Protocol, Version 6 (IPv6) Specification" :  
<http://www.ietf.org/rfc/rfc2460.txt>
- [RFC2462] "RFC 2462 : IPv6 Stateless Address Autoconfiguration" :  
<http://www.ietf.org/rfc/rfc2462.txt>
- [RFC2473] "RFC 2473 : Generic Packet Tunneling in IPv6 Specification" :  
<http://www.ietf.org/rfc/rfc2473.txt>
- [RFC2474] "RFC 2474 : Definition of the Differentiated Services Field (DS Field) in the IPv4 and the IPv6 Headers" :  
<http://www.ietf.org/rfc/rfc2474.txt>
- [RFC2529] "RFC 2529 : "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels" :  
<http://www.ietf.org/rfc/rfc2529.txt>
- [RFC2675] "RFC 2675 : IPv6 Jumbograms" :  
<http://www.ietf.org/rfc/rfc2675.txt>
- [RFC2711] "RFC 2711 : IPv6 Router Alert Option" :  
<http://www.ietf.org/rfc/rfc2711.txt>
- [RFC2893] "RFC 2893 : Transition Mechanisms for IPv6 Hosts and Routers" :  
<http://www.ietf.org/rfc/rfc2893.txt>
- [RFC3041] "RFC 3041 : Privacy Extensions for Stateless Address Autoconfiguration in IPv6" :  
<http://www.ietf.org/rfc/rfc3041.txt>
- [RFC3056] "RFC 3056 : Connection of IPv6 Domains via IPv4 Clouds" :  
<http://www.ietf.org/rfc/rfc3056.txt>
- [RFC3068] "RFC 3068 : "An Anycast Prefix for 6to4 Relay Routers" :  
<http://www.ietf.org/rfc/rfc3068.txt>
- [RFC3315] "RFC 3315 : Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" :  
<http://www.ietf.org/rfc/rfc3315.txt>
- [RFC3513] "RFC 3513 : Internet Protocol Version 6 (IPv6) Addressing Architecture" :  
<http://www.ietf.org/rfc/rfc3513.txt>
- [RFC3879] "RFC 3879 : Deprecating Site Local Addresses" :  
<http://www.ietf.org/rfc/rfc3879.txt>
- [RFC3964] "RFC 3964 : Security Considerations for 6to4" :  
<http://www.ietf.org/rfc/rfc3964.txt>
- [RFC3972] "RFC 3972 : Cryptographically Generated Addresses (CGA)" :  
<http://www.ietf.org/rfc/rfc3972.txt>

- [RFC4048] "RFC 4048 :RFC 1888 Is Obsolete" :  
<http://www.ietf.org/rfc/rfc4048.txt>
- [RFC4193] "RFC 4193 : Unique Local IPv6 Unicast Addresses" :  
<http://www.ietf.org/rfc/rfc4193.txt>
- [SANSv6] SANS Institute. "IPv6 TCP/IP and TCPDUMP : pocket reference guide" :  
[http://www.sans.org/resources/ipv6\\_tcpip\\_pocketguide.pdf](http://www.sans.org/resources/ipv6_tcpip_pocketguide.pdf)
- [SSTIC] A. Ebalard, G. Valadon. "La sécurité dans Mobile IPv6". *Publié à SSTIC 2006*, Rennes, France.
- [USCERT] US-CERT. "Malware Tunneling in IPv6".  
[http://www.us-cert.gov/reading\\_room/IPv6Malware-Tunneling.pdf](http://www.us-cert.gov/reading_room/IPv6Malware-Tunneling.pdf)
- [6BONE] Groupe de travail IETF ngtrans :  
[http://www.6bone.net/ngtrans\\_project-status.html](http://www.6bone.net/ngtrans_project-status.html)

## **Gestion détaillée du document**

**11 septembre 2006** version initiale.