FRÉDÉRIC ROUDAUT

# Mécanismes IPv6 avancés

Degré de difficulté

Depuis les années 80, l'Internet connaît un succès incroyable. La majeure partie des entreprises y est maintenant directement connectée, le nombre de particuliers détenteur d'un abonnement Internet auprès d'un FAI (Fournisseur d'Accès Internet) est en constante croissance.

Au moment de la définition d'IPv6, de nouveaux besoins tels que la sécurité, la mobilité sont apparus et ont pu être pris en compte lors de la phase de standardisation. Ce chapitre présente quelques-uns de ces mécanismes qui représentent une grande avancée de la couche réseau.

## IPsec

IPsec est le protocole spécifiquement conçu pour sécuriser IPv6. Il permet de réaliser des réseaux privés Virtuels ou VPNs (*Virtual Private Networks*) au niveau IP et offre les services :

- d'authentification des données,
- de chiffrement de données,
- d'intégrité des données pour garantir que les paquets n'ont pas été modifiés durant leur acheminement,
- d'anti-rejeu afin de détecter les éventuels paquets rejoués par un attaquant.

Toute implémentation IPv6 se doit de l'intégrer dans sa pile. Ce protocole est également utilisable avec IPv4 mais l'utilisation du NAT/PAT (*Network Address Translation/Protocole Address Translation*) en limite la mise en œuvre.

## Mécanismes IPsec

IPsec définit 2 protocoles de sécurisation :

- AH (*Authentication Header*),
- ESP (*Encryption Security Payload*).

Les services de sécurisation offerts par ces 2 protocoles sont distincts :

- AH permet de s'assurer que l'émetteur du message est bien celui qu'il prétend être. Il sert aussi au contrôle d'intégrité pour garantir au récepteur que personne n'a modifié le contenu d'un message lors de son acheminement et peut optionnellement être utilisé pour la détection des rejeux.
- ESP offre les mêmes services qu'AH et permet en plus de chiffrer l'ensemble des paquets ou uniquement la charge utile. ESP garantit également de façon limitée l'intégrité du flux.

## Associations de sécurité

Afin de sécuriser les échanges, les entités en présence doivent bien évidemment partager un ensemble commun d'informations telles que le protocole IPsec usité (AH ou ESP), les clés, les algorithmes … Ces différentes informations constituent des *associations de sécurité* ou SA (*Security Association*).

Chaque association de sécurité est identifiée de manière unique par un triplet comprenant un indexe de paramètres de sécurité SPI (*Security Parameters Index*), l'adresse du destinataire IP et le protocole de sécurité AH ou ESP.

Une association de sécurité est unidirectionnelle. Une communication bidirectionnelle entre 2 entités nécessite donc l'utilisation de 2 associations de sécurité.

### CET ARTICLE EXPLIQUE…

Cet article est la suite de celui publié dans le numéro précédent destiné à vous faire appréhender les techniques fondamentales d'IPv6, le nouveau mode d'adressage, les mécanismes de communication sous-jacents, la configuration automatique … bref l'ensemble des protocoles basiques qui composent l'architecture d'IPv6.

### CE QU'IL FAUT SAVOIR…

Afin d'appréhender au mieux cet article, il est préférable d'avoir des connaissances relativement solides d'IPv4 et en particulier du modèle en couche TCP/IP. Il est bien évidemment judicieux d'avoir également au préalable appréhendé les notions explicitées dans l'article précédent.

### Mode Transport et Tunnel

Les normes IPsec définissent deux modes distincts d'opération IPsec : le mode Transport et le mode Tunnel. Le mode Tunnel ne fonctionne que pour les datagrammes IP-in-IP. En mode Tunnel, le paquet IP interne détermine la stratégie IPsec qui protège son contenu tandis qu'en mode Transport, l'en-tête extérieur détermine la stratégie IPsec qui protège le paquet IP interne. Contrairement au mode Transport, le mode Tunnel ne permet pas à l'en-tête IP extérieur de dicter la stratégie de son datagramme IP interne. Enfin dans les 2 modes, l'entête extérieur est partiellement protégé mais le mode Tunnel à l'avantage de protéger intégralement son entête extérieur pouvant ainsi s'avérer fortement utile pour la création de VPN.

### AH (Authentication Header)

La mise en œuvre d'AH repose sur une extension d'entête spécifique. Celle-ci est définie dans la Figure 1.

Le rôle des différents champs de l'extension d'entête AH est précisé dans le Tableau 1.

### Protection AH et Algorithmes

AH suppose généralement une implémentation des algorithmes suivants :

- HMAC-MD5-96 (Peut être implémenté) : Cet algorithme produit une empreinte sur 128 bits tronquée à 96 bits pour le champ ICV de AH,
- HMAC-SHA1-96 (Doit être implémenté) : Cet algorithme produit une empreinte sur 160 bits tronquée à 96 bits pour le champ ICV de AH,
- AES-XCBC-MAC-96 (Devrait être implémenté) : Ce protocole utilise le chiffrement par bloc AES dans un mode d'opération de type *compteur* couplé à code d'authentification MAC (CBC-MAC). Le compteur sert à assurer un chiffrement sûr en évitant d'avoir un vecteur d'initialisation identique pour chaque message alors que le code d'authentification permet de vérifier que le message n'a pas été altéré. Cet algorithme produit également une empreinte sur 96 bits pour le champ ICV de AH.

D'autres algorithmes sont bien entendu utilisables, mais ceux précisés ci-dessus représentent l'ensemble commun minimum des implémentations d'AH.

Lors de la réception d'un paquet AH, la pile IPsec détecte l'association de sécurité concernée, en déduit les algorithmes et les clés associées, calcule la valeur du champ ICV et la compare avec la valeur fournie. Dans le cas où ces 2 valeurs coïncident l'intégrité ainsi que l'authentification des champs concernés est assurée. Ces champs diffèrent selon le mode usité transport ou tunnel.

Le rejeu des paquets est quant à lui détecté par le champ Sequence Number incrémenté à chaque paquet et également protégé par le champ ICV.

### Mode Transport

En *mode Transport* l'extension AH est insérée après l'entête IPv6 et avant les entêtes de niveau transport (TCP, UDP). De plus, AH étant vue comme une extension d'entête traitée de bout en bout de la communication, celle-ci apparait après les extensions d'entête *Hop-By-Hop Option Header*, *Routing Header* et *Fragment Header*. L'extension d'entête *Destination Options Header* est quant-à elle placée indifféremment avant ou après.

L'authentification et l'intégrité portent donc sur :

- les octets situés au dessus de l'extension d'entête AH,
- certains champs de l'entête IPv6 invariants lors de l'acheminement du paquet,
- certains champs invariants des extensions d'entête positionnées avant AH.

Les extensions d'entête positionnées après AH ne sont pas modifiées durant l'acheminement des paquets; à ce titre celles-ci sont protégées par le champ ICV. Cette protection diffère pour les extensions d'entêtes positionnés avant AH, certains champs pouvant être altérés par les routeurs présents le long du chemin.

Les sous-options présentes dans les extensions headers Hop-By-Hop et Destination Options Header disposent d'un bit positionné à 1 si l'option peut être modifiée le long du trajet. Dans le cas où ce bit n'est pas positionné la sous-option est protégée, dans le cas contraire les octets de la sous-option sont positionnés à 0 lors du calcul de l'ICV.

Cette protection ne s'applique pas non plus sur l'extension Fragment Headers

**Listing 1.** *Structure générale du fichier setkey.conf*

```
flush ;
spdflush;
#Configuration SPD
#Configuration SAD
spddump;
dump esp ;
```

**Listing 2.** *Configuration SPD sur 3ffe::1*

```
spdadd -6 3ffe::1 3ffe::2 any -P out ipsec esp/transport//require;
spdadd -6 3ffe::1 3ffe::3 any -P out ipsec esp/transport//require;
```

**Listing 3.** *Configuration SAD sur 3ffe::1*

```
add 3ffe::1 3ffe::2 esp 10
-E aes-cbc "aescbcencryption"
-A hmac-sha1 "hmacsha1authenticati";
add 3ffe::1 3ffe::3 esp 11
-E 3des-cbc "3descbcencryptiontesting"
-A hmac-sha1 "hmacsha1authenticati";
```

**Listing 4.** *Configuration SPD et SAD sur 3ffe::2*

```
spdadd -6 3ffe::1 3ffe::2 any -P in ipsec esp/transport//require;
add 3ffe::1 3ffe::2 esp 10
-E aes-cbc "aescbcencryption"
-A hmac-sha1 "hmacsha1authenticati";
```

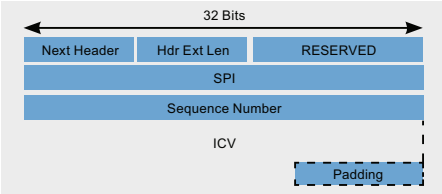


**Figure 1.** *Extension d'entête AH*

qui apparaît uniquement après la phase d'authentification.

La Figure 2 montre ainsi le positionnement de l'extension d'entête AH en mode transport ainsi que la portée de l'authentification/intégrité.

### Mode Tunnel

En mode Tunnel l'extension AH est insérée avant l'entête IPv6. Un nouvel entête IPv6 est alors inséré en tête. La Figure 3 et le Tableau 2 présentent le mode de construction de ce ne nouvel entête ainsi que le positionnement des champs de l'entête Intérieur.

## ESP (Encryption Security Payload)

La mise en œuvre d'ESP repose sur une extension d'entête spécifique. Celle-ci se décompose en 2

Ces 2 parties sont définies dans la Figure 4.

Le rôle des différents champs de l'extension d'entête ESP est précisé dans le Tableau 3.

### Protection ESP et Algorithmes

La protection d'ESP repose sur le choix des algorithmes d'authentification et de chiffrements. Ceux-ci peuvent être distincts ou combinés, c'est-à-dire qu'authentification et chiffrement sont réalisés par le même algorithme.

Dans le cas d'une protection combinée, ESP suggère l'utilisation d'AES-CCM ou AES-GCM déjà utilisé pour respectivement le 802.11i et le 802.1ae.

Dans le cas d'une protection séparée, ESP suppose généralement une implémentation des algorithmes d'authentification suivants :

- *NULL* Authentication (Peut être implémenté),
- *HMAC-MD5-96* (Peut être implémenté) : Cet algorithme produit une empreinte sur 128 bits tronquée à 96 bits pour le champ ICV de AH,
- *HMAC-SHA1-96* (Doit être implémenté) : Cet algorithme produit une empreinte sur 160 bits tronquée à 96 bits pour le champ ICV de AH,
- *AES-XCBC-MAC-96* (Devrait être implémenté) : Ce protocole utilise le chiffrement par bloc AES dans un mode d'opération de type *compteur* couplé à code d'authentification MAC (CBC-MAC). Le compteur sert à assurer un chiffrement sûr en évitant d'avoir un vecteur d'initialisation identique pour chaque message alors que le code d'authentification permet de vérifier que le message n'a pas été altéré. Cet algorithme produit également une empreinte sur 96 bits pour le champ ICV de AH.

Les algorithmes de chiffrements définis sont alors les suivants :

- *NULL* Encryption (Doit être implémenté),
- *AES-CBC* (Doit être implémenté) : AES supporte 3 tailles de clé : 128, 192 et 256 bits. La taille de clé par défaut est de 128 bits. AES-CBC nécessite un IV de 16 octets,
- *3DES-CBC* (Doit être implémenté) : Cet algorithme utilise une clé effective de 192 bits. Il est réalisé par application de 3 DES-CBC, chacun utilisant une clé de 64 bits (dont 8 bits de parité). *3DES-CBC* nécessite un IV de 8 octets,
- *AES-CTR* (Devrait être implémenté) : AES en mode compteur supporte 3 tailles de clé : 128, 192 et 256 bits. La taille de clé par défaut est de 128 bits. AES-CTR nécessite un IV de 16 octets,
- *DES-CBC* (Ne devrait pas être implémenté).

D'autres algorithmes sont bien entendu utilisables, mais ceux précisés ci-dessus représentent l'ensemble commun minimum des implémentations d'ESP. Il est à noter qu'une association de sécurité ESP ne doit à aucun moment utiliser conjointement un algorithme d'authentification et de chiffrement nul.

En mode tunnel, ESP depuis sa dernière version, propose un mode de confidentialité de flux par l'utilisation du champ TFC. Ce champ permet d'adjoindre des octets de bourrage de taille aléatoire. La taille ce

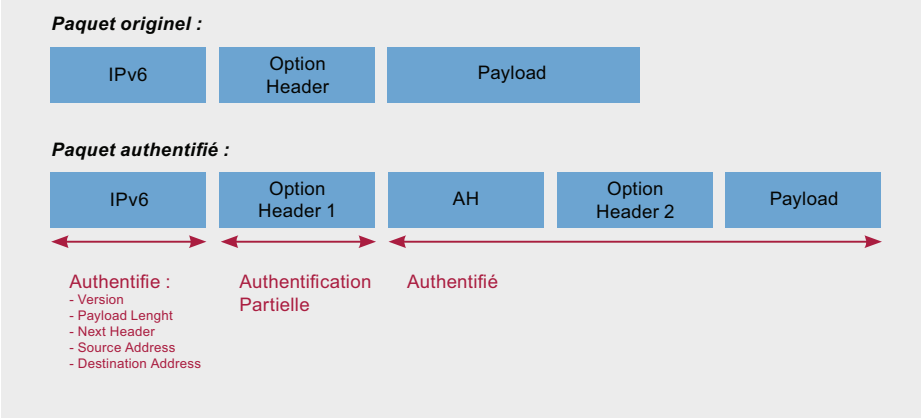


**Figure 2.** *AH en mode transport*

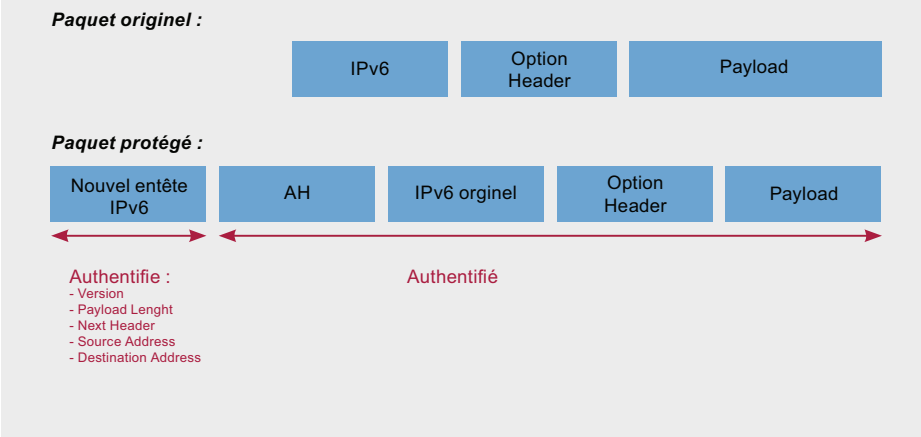


**Figure 3.** *AH en mode tunnel*


FRHACK
WHO will test your security
if YOU DON T ? !!
the 1st international technical IT Security conference organized in France
September 2009
http://www.frhack.org
SECURITY SYSTEM BREACH
INTRUSION DETECTED
FRHACK is organized by JA-PSI
French IT Security Company
http://www.ja-psi.com

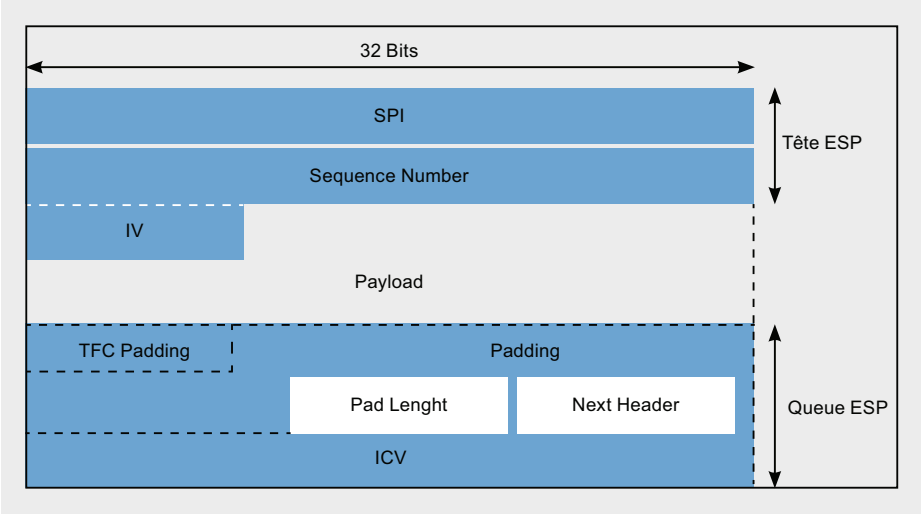


**Figure 4.** *Extension d'entête ESP*

champ n'étant précisée par aucun autre champ, celle-ci peut être déduite du champ *Payload Length* de l'entête IP intérieure au tunnel. Ce champ TFC pourrait également être utilisé en mode transport à la condition bien entendu que le protocole de niveau transport comporte une indication sur la taille de sa charge utile (cas de TCP, UDP, ICMP).

Lors de la réception d'un paquet ESP, la pile IPsec détecte l'association de sécurité concernée et en déduit les algorithmes et les clés associées.

Si la protection en authentification est activée, le récepteur calcule l'ICV sur le paquet ESP sans ce champ ICV. Si le champ calculé coïncide avec le champ transmis, l'intégrité est assurée sur les champs concernés. Ces champs diffèrent selon le mode usité, transport ou tunnel. Vient ensuite le déchiffrement du paquet avec l'algorithme et la clé fournie par l'association de sécurité.

Le rejeu des paquets est quant à lui détecté à la manière d'AH par le champ Sequence Number incrémenté à chaque paquet et également protégé par le champ ICV.

## Mode Transport

En mode *Transport* l'extension ESP est insérée de la même manière que l'extension AH, après l'entête IPv6 et avant les entêtes de niveau transport (TCP, UDP). ESP étant également vue comme une extension d'entête traitée de bout en bout de la communication, celle-ci apparaît après les extensions d'entête *Hop-By-Hop Option Header*, *Routing Header* et *Fragment Header*. L'extension d'entête *Destination Options Header* est quant à elle placée indifféremment avant ou après.

Le chiffrement porte donc sur les octets situés au dessus de l'extension d'entête ESP à l'exception des champs SPI, *Sequence Number*, ICV.

L'authentification éventuelle réalisée par le champ ICV porte sur l'ensemble des octets situés au dessus de l'extension d'entête ESP.

La Figure 5 montre ainsi le positionnement de l'extension d'entête ESP en mode transport ainsi que la portée de l'authentification/intégrité et du chiffrement.

## Mode Tunnel

En mode Tunnel l'extension ESP est insérée avant l'entête IPv6. Un nouvel entête IPv6 est alors inséré en tête. La Figure 6 et le tableau 4 présentent le mode de construction de ce ne nouvel entête ainsi que le positionnement des champs de l'entête Intérieur.Dans le cas d'une utilisation en mode tunnel la totalité du paquet initial est donc chiffrée.

## Topologies de mises en œuvre

IPsec a un intérêt majeur principalement par son mode ESP dans le cas où l'on souhaite :

- chiffrer et/ou authentifier du trafic de bout en bout ou jusqu'à une passerelle. Dans ce cas les entités en présence doivent préférentiellement disposer d'un adressage public, le NAT étant assez difficilement compatible avec IPsec. Une telle topologie a un intérêt majeur pour assurer la confidentialité entre 2 entités ou pour un utilisateur nomade par exemple,
- créer un VPN entre sites distants. Ce besoin intervient dans le cas où l'on veut par exemple interconnecter des réseaux privés distants au travers d'un réseau public.

Ces 2 modes opérationnels sont résumés dans la Figure 7. On précise que dans cette figure la protection est symétrique, ce qui n'est pas forcément le cas, les associations de sécurité étant unidirectionnelles.

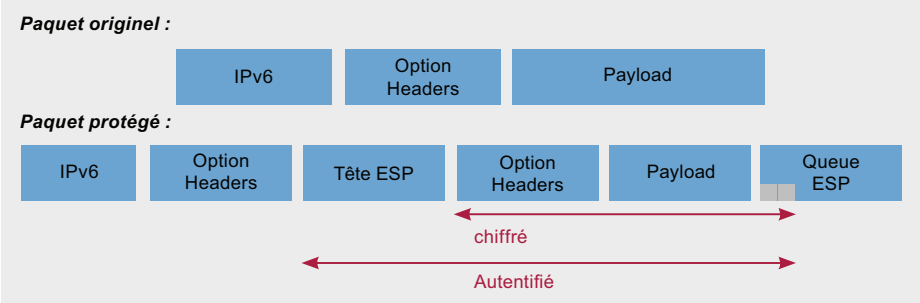


**Figure 5.** *ESP en mode transport*

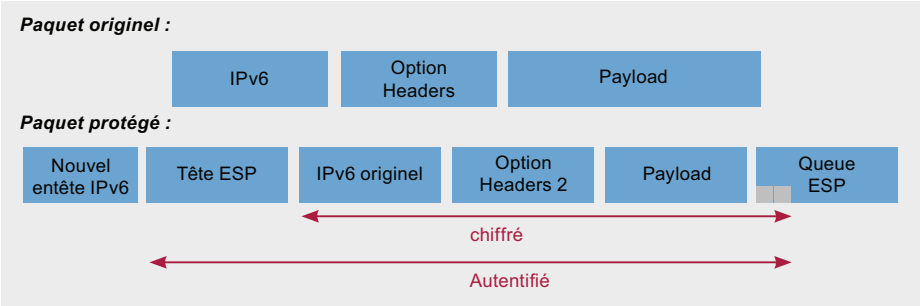


**Figure 6.** *ESP en mode tunnel*

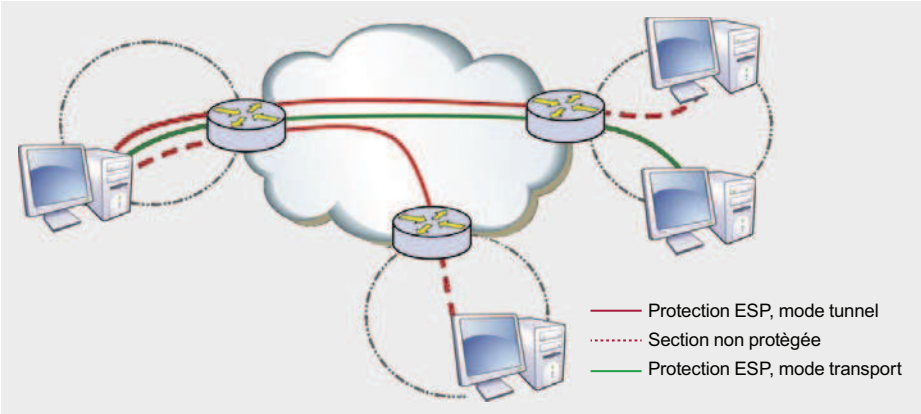


**Figure 7.** *Topologies ESP*

### IKE (Internet Key Exchange)

Il a été précédemment indiqué qu'AH et ESP nécessitaient des clés de chiffrements par le biais des associations de sécurité. Cette gestion des clés peut donc être manuelle; mais dans un environnement conséquent, une telle gestion devient irréalisable. De plus, cette méthode implique une définition totalement statique des associations de sécurité et un non-renouvellement des clés.

Le protocole IKE a donc été développé pour une gestion automatique des associations de sécurité, en particuliers des clés ainsi que des algorithmes à usiter.

IKE fait appel aux éléments suivants :

- un protocole de gestion des associations de sécurité, ISAKMP (*Internet Security Association and Key Management Protocol*), définissant des formats de paquets pour créer, modifier et détruire des associations de sécurité. Ce protocole sert également de support pour l'échange de clés préconisé par les protocoles de gestion de clés. Il assure aussi l'authentification des partenaires d'une communication,
- un protocole d'échange de clés de session basé sur SKEME et Oakley qui repose sur la génération de secrets partagés Diffie-Hellman,
- un domaine d'interprétation ou DOI (*Domain of Interpretation*) qui définit tous les paramètres propres à l'environnement IPsec, à savoir les protocoles d'échanges de clés, les paramètres d'associations de sécurité à négocier …,
- les clés utiles lors de l'authentification mutuelle des équipements IPsec qui intervient en préalable à toute négociation d'association de sécurité.

Ces clés peuvent être des clés partagées *Public Key Infrastructure*).

A l'heure actuelle 2 versions cohabitent, IKEv1 très complexe et IKEv2 qui en est une version simplifiée pour sa mise en œuvre ainsi que par son mécanisme.

## Mobilité de Machines : MIPv6

En termes de mobilité on distingue 2 mécanismes principaux : la micro-mobilité et la macro-mobilité. La micro-mobilité est celle utilisée entre autre par le wifi. Les entités en cours de déplacement se réassocient à des stations de base et conservent leur possibilité de connectivité au sein d'un domaine. Cette gestion des handovers est relativement fine et limite la signalisation au sein du réseau. Ces mécanismes sont cependant peu efficaces au sein de plusieurs domaines. En effet les adresses IP sont dans ce dernier cas renégociées pour mapper au domaine et pouvoir ainsi être routable.

La macro-mobilité résout ce problème en conservant une connectivité IP même lors d'un changement de domaine. Les adresses IP originelles continuent d'être utilisées lors des communications. De même les sessions TCP peuvent ainsi rester fonctionnelles lors d'un déplacement entre domaines. IPv6 intègre ce concept dans le protocole MIPv6 (*Mobile IPv6*).

### Concepts

Avant de poursuivre il s'agit de définir les mots clés principaux usités par MIPv6.

- *Réseau Mère* : Réseau auquel la machine appartient initialement,
- *Nœud correspondant* : machine dialoguant avec le mobile,
- *Home Address* : Adresse IPv6 dans le réseau mère,
- *Care-of Address* : Adresse IPv6 dans le réseau visité,
- *Agent Mère* : Machine du réseau mère servant *d'interface* entre le mobile et le nœud correspondant.

MIPv6 utilise intensivement la notion de tunnels. Schématiquement, les paquets transmis par le mobile dans un réseau étranger passent par *l'Agent Mère* présent dans le réseau mère, avant d'être retransmis au Nœud correspondant. Le chemin de retour est identique. Le routage sous-jacent apporte le paquet jusqu'à *l'Agent Mère* qui le retransmet au mobile dans son réseau visité.

L'agent mère doit également à tout moment être capable de localiser ses mobiles en déplacement. Il utilise pour cela un cache baptisé *Binding Cache* associant *Home Address* et *Care-of Address* de ses différents mobiles. Un mécanisme de signalisation protégé par IPsec en mode ESP

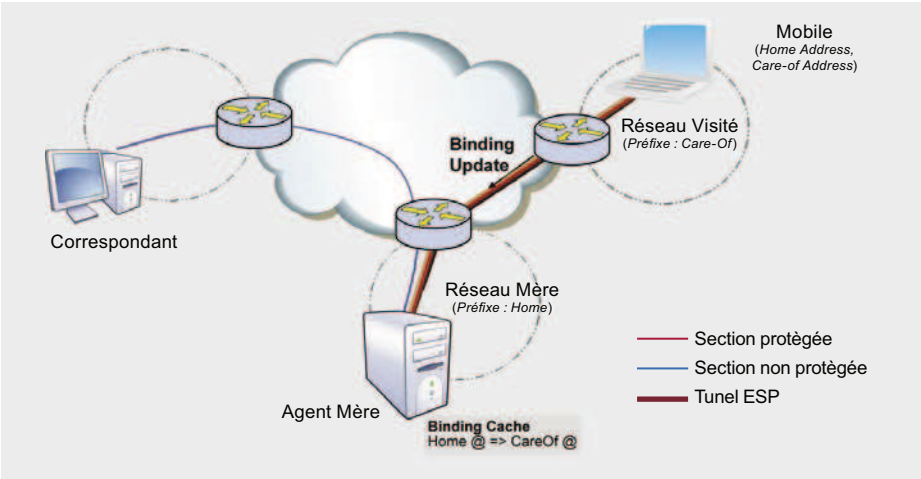


**Figure 8.** *Communication MIPv6 Basique sans optimisation*

est par conséquent usité pour mettre à jour ce cache. Il ne sera pas fait état des paquets MIPv6 ici, il s'agit simplement de savoir que cette mise à jour s'effectue à l'aide de paquets particuliers nommés Binding Update. Ceux-ci sont généralement acquittés par *l'Agent Mère* par des *Binding Acknowledgment*.

L'ensemble des mécanismes basiques de MIPv6 se situe au niveau de la couche IP dans le modèle TCP/IP. Ils ont été modelés pour permettre une communication avec des entités n'ayant pas conscience des protocoles de mobilité. Ils n'ont aucun impact sur les couches de niveau transport et applicative. Pour le correspondant cette communication est totalement transparente.

### Mobilité de Machines : MIP6

Le mobile situé dans son réseau mère utilise sa Home Address pour dialoguer avec des Nœuds correspondants de manière classique. Lorsqu'il se déplace dans un réseau visité la procédure est la suivante :

- Le mobile obtient une nouvelle adresse IP par combinaison de son adresse MAC et du nouveau préfixe réseau, la *Care-of Address*. Il dispose toujours de sa *Home Address*,
- Le mobile transmet un *Binding Update* à l'agent mère afin de mettre à jour son cache d'association. Ce paquet étant protégé par IPsec en mode ESP, l'authentification, l'anti-rejeu, la
- L'agent mère aura alors à charge de capturer les paquets auparativement transmis au mobile. Il utilise dans cette optique les possibilités offertes par le protocole de découverte des voisins (*Neighbor Discovery*) en annonçant son adresse MAC comme destinataire de l'ensemble des paquets unicast à destination du mobile. Les caches NDP des machines présentent sur le lien mère seront ainsi remis à jour,
- Lorsque le mobile souhaite dialoguer avec un nœud correspondant il peut choir d'utiliser son nouvel adressage, ou de masquer sa mobilité par l'utilisation de sa *Home Address*. Dans ce dernier cas il construit un tunnel ESP avec son *Agent Mère* et encapsule les paquets à destination de son correspondant. L'adresse source de la partie interne est ainsi la *Home Address*, l'adresse destination est celle du correspondant.
- Le paquet parvient à *l'Agent Mère*, qui vérifie son authentification, le déchiffre, le désencapsule et le retransmet sur le réseau.
- Le correspondant pourra y répondre de manière symétrique. Cette réponse sera capturée par *l'Agent Mère*, chiffrée et authentifiée avant d'être retransmise au mobile dans le tunnel ESP. En cas d'un déplacement en cours de communication le binding cache aura été mis à jour permettant à l'Agent mère de retrouver son mobile.

La Figure 8 positionne ces différentes entités dans un contexte MIPv6.

### Optimisations de routes

Les échanges entre mobiles et correspondants n'étant pas toujours les plus optimums en matière de routage, MIPv6 intègre un mode d'optimisation pour les correspondants intégrant des

**Table 1.** *Rôle des différents champs de l'extension d'entête AH*

| Champs | Taille | Rôle |
|---|---|---|
| Next Header | 8 bits | Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête. Similaire au champ Protocol en IPv4. |
| Payload Len | 8 bits | Spécifie la longueur -2 en mots de 32 bits de l'extension d'entête AH. |
| RESERVED | 16 bits | Positionné à 0. |
| SPI | 32 bits | Security Parameters Index utilisé par le récepteur pour trouver l'association de sécurité à utiliser. |
| Sequence Number | 32 bits | Compteur incrémenté à chaque paquet. Permet en particulier de détecter le rejeu. |
| ICV | Variable Selon | Integrity Check Value. Destiné à la validation de l'intégrité du paquet. Doit être un multiple de 32 bits. |
| Padding | Variable | Utilisé pour des besoins d'alignement d'entête. Sa taille est telle que l'extension d'entête AH est un multiple de 64 bits (32 bits pour IPv4). |

**Table 2.** *Construction de l'entête IPv6 extérieure pour AH en mode tunnel*

| Champs de l'entête IPv6 | Entête Extérieur | Entête Intérieur |
|---|---|---|
| Version | Positionné à la valeur 6. | Aucune modification. |
| DS | Copié depuis l'entête intérieur. | Aucune modification. |
| ECN | Copié depuis l'entête intérieur. | Positionné à 0. |
| Flow Label | Copié depuis l'entête intérieur ou configuré. | Aucune modification. |
| Payload Length | Construit. | Aucune modification. |
| Next Header | Positionné à la valeur de AH (51) | Aucune modification. |
| Hop Limit | Construit. | Décrémenté d'une unité |
| Source Address | Construit. | Aucune modification. |
| Destination Address | Construit. | Aucune modification. |
| Extensions Headers | Jamais copié mais peut apparaître en postambule. | Aucune modification. |

# Salon Informatique Maubeuge

Vendredi 5 & Samedi 6 juin 2009

**Entrée gratuite**

## Logiciels libres

## Ateliers
## Exposants
## Demonstrations

## Challenge des IUT de France

## Challenge de securité informatique

www.salon-informatique-maubeuge.com

Partenaires :

ACISSI

Plus de 40 ans – Depuis 1967 – iut Valenciennes – L'EXPERIENCE

FEDERATION SYNTEC

VILLE DE MAUBEUGE

RÉGION Nord-Pas de Calais

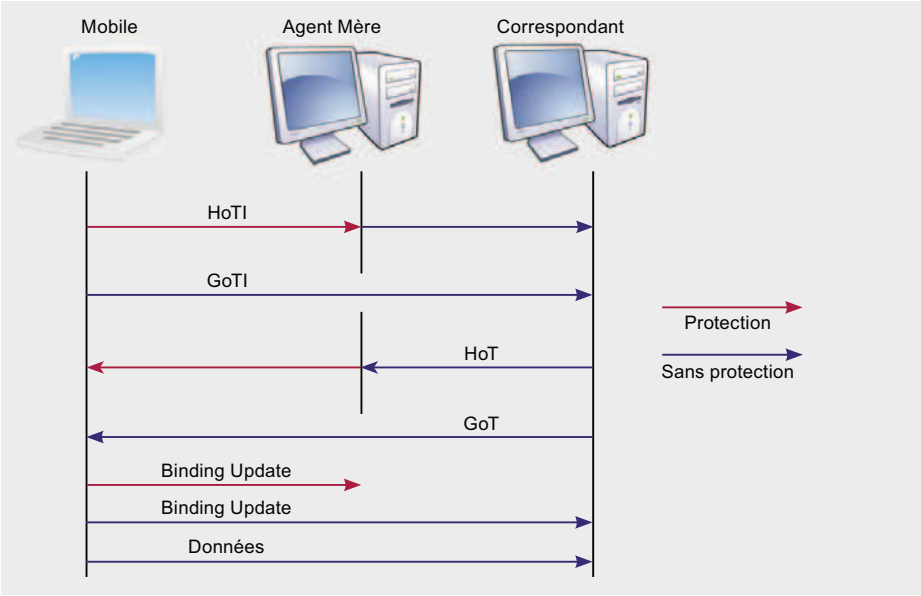


**Figure 9.** *Return Routability Procédure*

fonctions spécifiques. Il s'agit de supprimer simplement la passerelle occasionnée par *l'Agent Mère*.

Pour cela MIPv6 définit 2 nouvelles options :

- *Routing Header de type 2* : qui est simplement une extension d'entête *Routing Header contenant* la *Home Address* du mobile
- *Home Address Option* : qui est une sous-option de l'extension d'entête *Destination Option Header* traité uniquement par le récepteur du paquet.

Lorsqu'un correspondant supporte l'optimisation de routage, il maintient tout comme *l'Agent Mère* une table des associations pour tous les mobiles avec lesquels il est en communication. Une vérification axée autour d'ICMPv6 est préalable avant toute optimisation.

Le principe est alors assez proche de celui usité avec *l'Agent Mère* :

- Le mobile en déplacement transmet un *Binding Update* au correspondant pour lui faire état de sa nouvelle localisation après en avoir fait de même à son *Agent Mère*. Ce correspondant mettra alors à jour son *Binding Cache*.
- Lorsque le mobile veut transmettre un message au correspondant, il utilise en adresse source sa *Care-of Address* mais ajoute l'option *Home Address Option*.
- Le paquet subira le routage classique entre le mobile et le correspondant, remontera dans la pile MIPv6 de ce correspondant qui échangera *Care-of Address* du champ adresse source et *Home Address* présentes dans l'option *Home Address Option*. Pour la pile IPv6, le paquet sera transparent comme provenant directement du mobile depuis son réseau *Mère*. Dans le cas où ce paquet est protégé par IPsec, les vérifications seront donc basées sur l'adresse mère.
- Avant de répondre, le correspondant cherchera dans sa table d'association la *Care-Of Address* du mobile. Il transmettra alors le paquet en utilisant cette *Care-Of Address* en destination et y ajoutera l'option *Routing Header* de type 2 remplie avec la *Home Address*.
- Le paquet parviendra donc au mobile qui échangera préalablement l'adresse de destination avec la *Home Address*. Le paquet remontera donc également dans les couches de manière totalement transparente.

Ce mécanisme donne donc des trajectoires optimums en matière de routage et permet de limiter les contraintes en matière d'ingress et d'outgress *filtering*. Ce mécanisme de mise à jour d'association pose cependant d'importants problèmes en matière de sécurité. En effet, il est aisé de protégé les échanges de signalisation entre le mobile et l'agent mère du fait de la relation administrative qui

### Return Routability procédure

Cette procédure est destinée à la protection partielle des associations de sécurité entre mobile et correspondant dans le cas de l'optimisation de route. Elle repose sur une utilisation de 4 messages principaux :

- HoTI : *Home Test Init*,
- CoTI : *Care-of Test Init*,

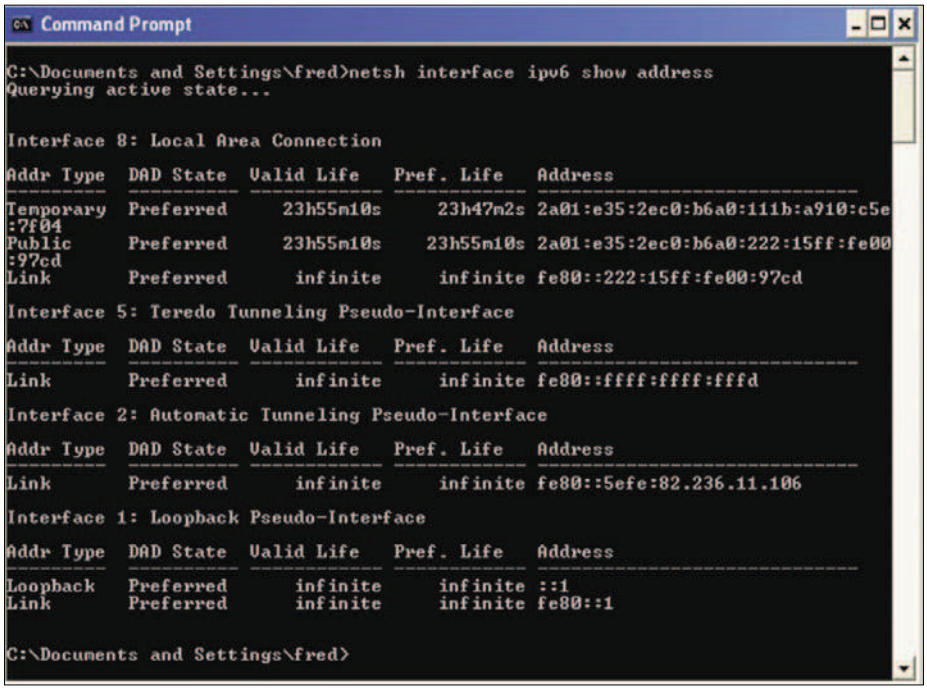


**Figure 10.** *Commande netsh show sous Windows XP*

- HoT : *Home Test*,
- CoT : *Care-of Test*.

Les correspondants intégrant l'optimisation de route doivent préalablement disposer de nonces ainsi que d'une clé secrète notée *Kcn*.

La procédure usitée est la suivante :

- un message HoTI est émis depuis la *Home Address* du mobile vers le correspondant via l'agent mère. Il contient une valeur aléatoire sur 64 bits, le *Home Init cookie*,
- parallèlement un message CoTI est émis depuis la *care-of address* du mobile, directement vers le nœud correspondant. Celui-ci contient une seconde valeur aléatoire sur 64 bits, le *Care-of Init cookie*,
- en réponse au message HoTI, un message HoT, est émis par le correspondant à destination de la *Home Address* du mobile via *l'Agent Mère*. Ce paquet contient entre autre l'index d'un nonce choisi par le correspondant ainsi qu'un Home Keygen token calculé par : `premier (64, HMAC_SHA1 (Kcn, (home address | nonce | 0 )))`
- de même, en réponse au message CoTI, un message CoT est émis par le correspondant vers la *Care-of Address* du mobile. Ce paquet contient entre autre l'index d'un autre nonce choisi par le correspondant ainsi qu'un *Home Keygen* token calculé par : `premier (64, HMAC_SHA1 (Kcn, (care-of address | nonce | 0 )))`

A l'issue de ces différentes étapes, le mobile calcule une clé notée Kbm :

```
Kbm = SHA1 ( "home keygen token" |
    "care-of keygen token")
```

La Figure 9 présente le cheminement de ces différents messages au travers de l'Internet.

Cette clé sera utilisée lors de la mise à jour des associations pour authentifier le mobile par le calcul d'un HMAC.

Cette procédure repose sur l'hypothèse forte qu'aucun espion n'écoute à la fois les messages CoT et HoT qui normalement empruntent des chemins distincts. Dans le cas contraire il lui serait aisé de calculer Kbm et de générer des faux messages d'association. Cette écoute n'est pas faisable dans le réseau visité puisque les échanges entre *Agent Mère* et mobile sont chiffrés. Pratiquement cette attaque est aisée dans le réseau du correspondant mais celle-ci n'est pas évaluée comme étant plus risquée que celles que l'on peut retrouver dans un contexte sans mobilité par simple *IP-spoofing*, *NDP spoofing*…

Afin de réduire les risques, les nonces ainsi que la clé Kcn sont régulièrement mis à jour.

### Mobilité de Réseaux : NEMO

MIPv6 gère la mobilité d'un hôte tandis que NEMO assure la mobilité d'un réseau IPv6 entier, appelé réseau mobile. Dans le cas de NEMO, la complexité est centralisée sur un équipement dédié : le routeur mobile. Ainsi, chaque mouvement (lorsque le réseau mobile se déplace d'un réseau d'accès vers un

**Table 3.** *Rôle des différents champs de l'extension d'entête ESP*

| Champs | Taille | Rôle |
|---|---|---|
| SPI | 32 bits | Security Parameters Index utilisé par le récepteur pour trouver l'association de sécurité à utiliser. |
| Sequence Number | 32 bits | Compteur incrémenté à chaque paquet. Permet en particulier de détecter le rejeu. |
| IV | Variable Selon l'algorithme usité | Vecteur d'initialisation éventuel pour les algorithmes de chiffrement. |
| TFC Padding | Variable | Traffic Flow Confidentiality. Utilisé pour une protection contre les attaques statistiques. |
| Padding | Variable | Utilisé pour des besoins d'alignement d'entête. Sa taille est telle que l'extension d'entête ESP est un multiple de 64 bits (32 bits pour IPv4). |
| Pad Length | 8 bits | Indique la taille du champ Padding en octets. |
| Next Header | 8 bits | Décrit l'entête de la couche immédiatement supérieure ou la prochaine extension d'entête. Similaire au champ Protocol en IPv4. |
| ICV | Variable Selon l'algorithme usité | Integrity Check Value. Destiné à la validation de l'intégrité du paquet. Doit être un multiple de 32 bits. |

**Table 4.** *Construction de l'entête IPv6 extérieure pour ESP en mode tunnel*

| Champs de l'entête IPv6 | Entête Extérieur | Entête Intérieur |
|---|---|---|
| Version | Positionné à la valeur 6. | Aucune modification. |
| DS | Copié depuis l'entête intérieur. | Aucune modification. |
| ECN | Copié depuis l'entête intérieur. | Positionné à 0. |
| Flow Label | Copié depuis l'entête intérieur ou configuré. | Aucune modification. |
| Payload Length | Construit. | Aucune modification. |
| Next Header | Positionné à la valeur de ESP (50) | Aucune modification. |
| Hop Limit | Construit. | Décrémenté d'une unité |
| Source Address | Construit. | Aucune modification. |
| Destination Address | Construit. | Aucune modification. |
| Extensions Headers | Jamais copié mais peut apparaître en postambule. | Aucune modification. |

autre) est transparent pour l'ensemble des hôtes IPv6 du réseau mobile. Un hôte IPv6 standard peut ainsi bénéficier d'une connectivité permanente au sein d'un réseau mobile sans avoir toutefois besoin de protocoles additionnels.

NEMO, couplé avec certaines extensions, gère notamment la mobilité des réseaux IPv6, la

## Pratique & Mise En œuvre

La majeure partie des Systèmes d'exploitation, des softwares des équipements réseaux actuels disposent d'un support IPv6. Vous pourrez le vérifier sur le site de *l'IPv6 Ready Logo Committee*, programme mondial de certification IPv6. Vous obtiendrez sur ce site le détail des implémentations actuellement certifiées et vous pourrez aisément y constater l'important retard de l'Europe.

Les infrastructures réseaux européennes ont également accumulées un retard considérable dans cette migration … Et pourtant les réseaux de l'enseignement et de la recherche proposent depuis déjà plusieurs années un support Natif d'IPv6 voir du multicast IPv6. Heureusement quelques ISP (*Internet Service Provider*), tels que Free, offrent depuis quelques mois un adressage IPv6.

Ce paragraphe a pour objectif de vous faire appréhender la mise en œuvre basique d'IPv6 sur les principaux systèmes usités, à savoir Windows et Linux. On suppose que vous disposez d'ores et déjà d'un adressage IPv6 parce que vous êtes par exemple dans une des situations précédemment évoquées. On rappelle que les Internet IPv4 et IPv6 sont bien distincts même si une utilisation des machines en double pile permet la superposition de certaines portions. Dans le cas contraire, si vous désirez plus qu'un réseau local, il vous faudra utiliser un des mécanismes de transition décrit dans l'article précédent. Nous vous conseillons préférentiellement un tunnel broker et en second choix un tunnel 6to4.

### Avec Windows

La majeure partie des versions courantes de Windows disposent d'un support IPv6 : Vista, XP SP1, XP SP2, Server 2003, 2008. Sous Vista et Server 2008 ce support est activé par défaut. Sous XP ou Server 2003, il vous faudra l'activer au préalable.

Selon les versions de Windows les mécanismes disponibles sont plus ou moins complets.

La mobilité IPv6 ne prend en compte que la partie correspondant ; ni Home Agent ni Nœud Mobile ne sont disponibles ;

Sous Windows XP et Server 2003, IPsec pour IPv6 offre les mécanismes AH et ESP mais le chiffrement ainsi que la gestion automatique des clés n'est pas disponible. Seul Vista et Server 2008 offrent ces fonctionnalités.

Vista et Server 2008 permettent une utilisation de DHCPv6.

### Activation de la pile IPv6 & Configuration des Adresses

Ce besoin ne se retrouve que sous XP et Windows Server 2003 dont la pile IPv6 est par défaut désactivée. Cette activation se fait par le biais de l'outil ipv6.exe sous Windows XP où de la commande netsh disponible sur toutes les versions.

Sous Windows XP, il s'agit d'exécuter :

```
ipv6 install
```

Bien entendu les interfaces concernées doivent accepter la connectivité TCP/IPv6 dans le menu *Propriétés* adéquat.

Une adresse Lien-locale associée à chacune de vos carte réseau sera

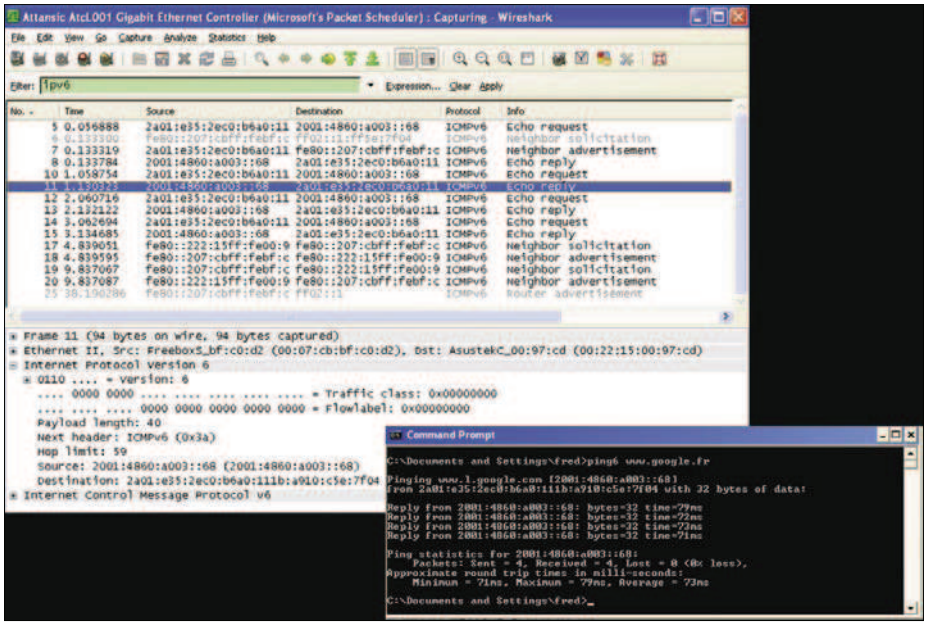


**Figure 11.** *Ping6 www.google.fr*

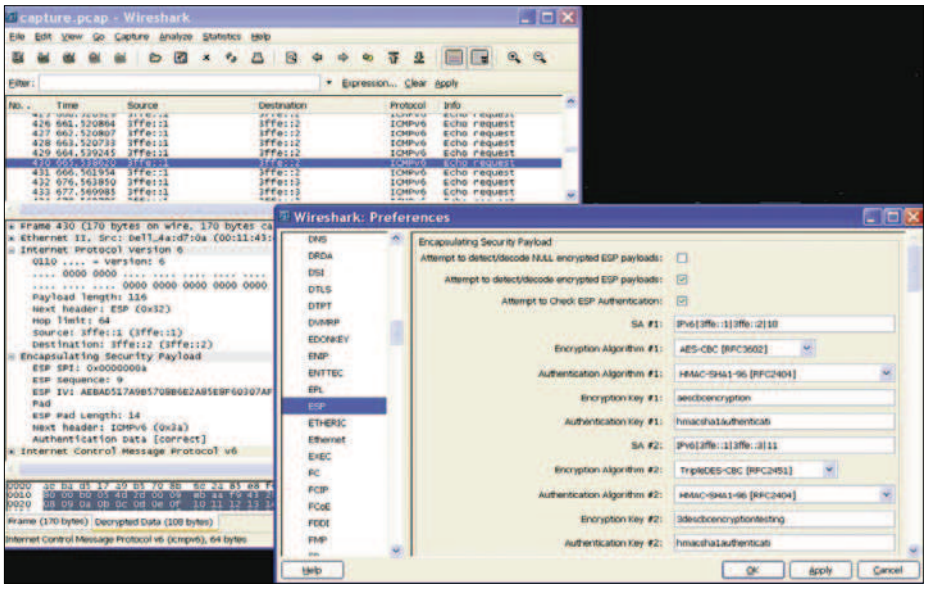


**Figure 12.** *Déchiffrement IPsec avec Wireshark*


Cisco Partner Learning Solutions · Microsoft Gold Certified Partner · Nortel Education Provider · VMware authorized Training Center · HP ProCurve Networking by HP · Check Point Software Technologies Ltd. We Secure the Internet.

Pour accompagner les ingénieurs, DSI, chefs de projets, administrateurs réseau et tous les responsables de la sécurité des systèmes d'information dans leurs missions, une seule formation n'est pas suffisante….

L'offre de formations de **Global Knowledge** s'étend au-delà des considérations techniques des produits installés, pour vous apporter le recul nécessaire au métier de la «sécurité» et vous permettre d'appréhender les enjeux et la nature des risques auxquels les systèmes d'information se trouvent confrontés. Retrouvez nos solutions sur **www.globalknowledge.fr**

Atelier de préparation à la certification CISSP

**15-19 Juin 2009, Paris**

Global Knowledge™

alors automatiquement configurée par concaténation du préfixe fe80 et de votre identifiant d'interface défini depuis l'adresse MAC associée. Les interfaces reliées à un réseau IPv6 constitué d'un routeur annonçant des Router Advertisement, obtiendront de même automatiquement une adresse globale unicast, unique, routable et contenant l'adresse MAC de l'interface concernée.

La commande `ipconfig /all` (ou `netsh show`) vous prouvera votre connectivité. La figure 10 vous montre une telle configuration sous Windows XP.

Vous constaterez également une adresse supplémentaire, qualifiée de Temporaire. Il s'agit en fait d'une adresse globale, de durée de vie relativement courte destinée au masquage de l'adresse MAC (disponible depuis le SP2). Au besoin vous pourrez la désactiver par :

```
ipv6 -p gpu useTemporaryAddresse no
```

### Connectivité, chemin

Lorsque vous disposerez d'une adresse routable ou simplement pour tester la connectivité entre deux machines, vous pourrez utilisez la commande ping6 qui est le pendant de ping pour IPv4. Cette commande génère un ensemble de paquets *ICMPv6 Echo Request* et affiche les réponses associées *ICMPv6 Echo Reply*.

La Figure 11 montre un tel ping6 sur www.google.fr désormais adressable en IPv6. Les paquets résultants de cette commande sont également indiqués par capture du trafic avec Wireshark.

En IPv4, pour connaître le trajet suivi par les paquets, on utilise généralement la commande tracert. En IPv6, il s'agit désormais de tracert6

### Cache des voisins (NDP Cache)

La résolution MAC/Adresse en IPv4 donne naissance au cache ARP obtenu par arp –an par exemple. En IPv6, il s'agit désormais du cache NDP qui peut être obtenu par :

`ipv6 nc`, OU `netsh interface ipv6 show neighbors`.

### Diverses Commandes

L'ensemble des configurations essentielles IPv6 sous Windows s'effectuent à l'aide de Netsh (et/ou *ipv6.exe* sous Windows XP). Hormis celles précédemment définies, les commandes essentielles sont indiquées dans le Tableau 5.

### Accès Web en IPv6

Classiquement en IPv4, les URLs (*Uniform Resource Locator*) utilisées dans les accès HTTP (*HypertexT Transfert Protocol*) utilisent le nommage DNS (*Domain Name System*). Avant toute requête l'adresse du serveur HTTP est donc généralement préalablement traduite par le biais des serveurs DNS. Avec IPv6, il en est de même, le browser dans un premier temps recherche l'ensemble des adresses IP associées au serveur HTTP. Si celui-ci dispose d'une adresse IPv6, il tentera dans un premier temps de le joindre par IPv6. En cas d'échec, c'est le protocole IPv4 qui sera utilisé. Nous rappelons qu'il n'est pas indispensable que le serveur DNS soit adressé en IPv4 pour retourner des adresses IPv6.

**Table 5.** *Les commandes essentielles IPv6 sous Windows*

| Commande Netsh | Rôle |
|---|---|
| `netsh interface ipv6 show interface` | Affiche les interfaces IPv6 |
| `netsh interface ipv6 set interface [[interface=]String] [[forwarding=]{enabled \| disabled}] [[advertise=]{enabled \| disabled}] [[mtu=]Integer] [[siteid=]Integer] [[metric=]Integer] [[store=]{active \| persistent}]` | Permet d'activer le forwarding des interfaces, les annonces de Router Advertisement |
| `netsh interface ipv6 add address [[interface=]String] [address=]IPv6Address [[type=]{unicast \| anycast}] [[validlifetime=]{Integer \| infinite}] [[preferredlifetime=]{Integer \| infinite}] [[store=]{active \| persistent}]` | Permet d'ajouter des adresses IPv6 aux interfaces |
| `netsh interface ipv6 show bindingcacheentries` | Affiche le Binding cache utilisé par MIPv6 |
| `netsh interface ipv6 show routes [[level=]{normal \| verbose}] [[store=]{active \| persistent}]` | Affiche les routes IPv6 |
| `netsh interface ipv6 add route [prefix=]IPv6Address/Integer [[interface=]String] [[nexthop=]IPv6Address] [[siteprefixlength=]Integer] [[metric=]Integer] [[publish=]{no \| yes \| immortal}] [[validlifetime=]{Integer \| infinite}] [[preferredlifetime=]{Integer \| infinite}] [[store=]{active \| persistent}]` | Ajoute une route IPv6 dans la table de routage |
| `netsh interface ipv6 renew [[interface=]String]` | Permet la réinitialisation des adresses IPv6 |

**Table 6.** *Commandes principales pour la configuration d'IPv6 sous linux*

| Commande ip | Rôle |
|---|---|
| `ip -6 address show [dev <périphérique>]` | Affiche les adresses IPv6 |
| `ip -6 addr add <adresseipv6>/ <longueurdupréfixe> dev <interface>` | Ajoute une adresse IPv6 |
| `ip -6 route show [dev <périphérique>]` | Affiche les routes IPv6 |
| `ip -6 route add <réseauipv6>/ <longueurdupréfixe> via <adresseipv6> [dev <périphérique>]` | Ajoute une route IPv6 |
| `ip -6 neigh show [dev <périphérique>]` | Affiche les voisins NDP |
| `ip -6 neigh add <adresseIPv6> lladdr <adressedelacouche-lien> dev <périphérique>` | Ajoute un voisin NDP |

A l'heure actuelle la majeure partie des navigateurs supporte IPv6 par défaut, Firefox, Internet Explorer … A titre d'exemple vous pourrez vous connecter sur *www.kame.net*. Si vous disposez d'un accès extérieur IPv6 ainsi que d'un browser compatible vous devriez voire en première page une tortue animée. Le cas échéant celle-ci sera fixe.

Avec IPv6, les adresses étant 4 fois plus longues, les URLs contenant des IPs devraient encore moins se pratiquer. Cependant ceci reste possible et pour différencier les :: de l'adresse avec la section port de l'URL, il faut entourer l'IP de [ ]. (Exemple : *http://[2001:4860:a003::68]* pour accéder à google en IPv6).

### Avec Linux

Quelle que soit la distribution contemporaine utilisée, celle-ci contient IPv6. Vous pourrez néanmoins tester la présence de son support dans le noyau par vérification de la présence du chemin : */proc/net/if_inet6*. Le module IPv6 doit également être chargé avant toute utilisation. Un appel à lsmod vous le confirmera.

### Activation de la pile IPv6 & Configuration des Adresses

Sous Linux l'ensemble des configurations IPv6 peut être réalisé à l'aide des anciennes commandes ifconfig, netstat …

Depuis les noyaux au moins supérieur au 2.4, le sous-système réseau a été complètement réécrit. L'iproute2 étend ainsi grandement les possibilités et centralise les configurations réseaux. La commande principale est ip.

Le Tableau 6 présente donc quelques une des options principales pour configurer Ipv6.

Le système Linux étant l'un des mieux documentés, si l'une des options vous manque vous pouvez toujours utiliser la commande `man` (exemple : `man 8 ip`).

### Mise en œuvre mode routeur

Afin de mettre en place un routeur et/ou une passerelle vous devez activer le forwarding entre les différentes interfaces réseaux. Ceci peut être réalisé par le biais de fichiers de configuration spécifiques à chaque distribution (généralement sous l'arborescence */etc/sysconfig/network*) ou directement par dialogue avec le Kernel. Ce dialogue est temporaire et à chaque reboot, il sera réinitialisé (sauf utilisation de script de démarrage, généralement */etc/rc.local*). Il se réalise par des appels à la commande sysctl ou par écriture dans les fichiers propres au kernel.

Il s'agit sous IPv6 de l'arborescence */proc/sys/net/ipv6*. Le fait d'écrire *1* dans le fichier */proc/sys/net/ipv6/conf/all/forwarding* activera le forwarding entre toutes les interfaces. Au besoin, le contrôle du forwarding par interface doit être réalisé en utilisant les jeux de règles de *netfilter-IPv6* (à l'aide *d'ip6tables*) en spécifiant les périphériques d'entrée et de sortie.

Il vous faudra certainement en plus activer les *Router Advertisements* afin de permettre aux machines présentes sur le lien de s'autoconfigurer. Ces paquets sont générés suite au démarrage du démon *radvd*. Ce démon utilise un fichier de configuration présent généralement dans */etc/radvd.conf*. Ce fichier stipule les principaux paramètres des *Router Advertisements*, à savoir :

- le préfixe,
- la durée de vie du préfixe,
- la fréquence des envois d'annonce,

En dernier point il vous faudra peut-être activer un protocole de routage intra-domaine (*Ripng, OSPfv3*) voir inter-domaine (Is-Is, BGP-4+).

### Commandes et outils principaux

Les commandes principales disponibles sous Linux sont équivalentes à celle précédemment évoquées pour Windows. Les principales sont les suivantes :

- `ping6` (*Packet INternet Grouper*) : pour diagnostiquer la connectivité réseau. (Exemple : `ping6 [-I <périphérique>] FF02::1` vous donnera l'ensemble des interfaces présentes sur le lien-local,
- `traceroute6` : pour détecter le chemin emprunté par les paquets,
- `tracepath6` : similaire au `traceroute6`, trace le chemin vers une destination donnée tout en découvrant la MTU le long de ce chemin,
- `nslookup`, `host` : utiles pour la résolution DNS en v4 ou v6.

L'ensemble des outils classiques réseaux disponibles sur Linux a été adapté à IPv6 : ssh, telnet, ftp, netcat, nmap …

Le firewall iptable classique dispose également d'une variante baptisée *ip6table* pour IPv6.

### Mise en œuvre d'IPsec

La pile IPsec est maintenant intégrée en natif sur les noyaux 2.5.47 et supérieurs ; les versions inférieures nécessitaient l'installation de piles spécifiques style FreeS/WAN ou celle du projet japonais USAGI. L'implémentation actuelle repose d'ailleurs sur celle du projet USAGI. Elle peut cependant ne pas être activée par défaut pour IPv6 ; il vous faudra donc potentiellement relancer préalablement une compilation du noyau et y activer AH, ESP voir IPComp (*Compression de charge IP*).

La configuration des politiques IPsec ainsi que des clés et algorithmes en mode partagé s'effectue à l'aide de l'outil setkey, dérivant du projet KAME et fournie avec le package ipsec-tools. Si vous choisissez un mode de configuration automatique des associations de sécurité, il vous faudra user d'un outil supplémentaire, racoon ou racoon2 selon la version d'IKE choisie.

**Table 7.** *Références*

| Lien | Titre |
|---|---|
| *http://livre.point6.net/index.php* | IPv6 Théorie et Pratique - Gisèle Cizault |
| *http://ipv6ready.org* | Site de l'IPv6 Ready Logo Committee |
| *http://www.deepspace6.net/docs/ipv6_status_page_apps.html* | Statut des applications réseaux supportant IPv6 |
| *http://mirrors.deepspace6.net/Linux+IPv6-HOWTO-fr/* | HOWTO IPv6 pour Linux |
| *http://www.linux-france.org/prj/inetdoc/guides/Advanced-routing-Howto/* | HOWTO du routage avancé et du contrôle de trafic sous Linux |
| *http://wiki.wireshark.org/ESP_Preferences* | Le module de déchiffrement et d'authentification ESP pour Wireshark |

Par simplification nous choisirons un mode manuel de gestion des associations de sécurité.

- `3ffe::2` : par ESP (Chiffrement : aes-cbc, clé : aescbcencryption ; Authentification : hmac-sha1, clé : hmacsha1authenticati ; SPI : 10);
- `3ffe::3` : par ESP (Chiffrement : 3des-cbc, clé : 3descbcencryptiontesting ; Authentification : hmac-sha1, clé : hmacsha1authenticati ; SPI : 11)

Chacune de ces différentes machines devra donc être configurée pour prendre en compte ce paramétrage IPsec. Ceci peut se réaliser par définition d'un fichier de configuration nommé par exemple *setkey.conf* utilisant le format suivant (Listing 1).

Si l'on considère `3ffe::1`, Il faut donc dans un premier temps définir les SPD (Security Policy Database) afin que tout trafic sortant en direction de `3ffe::2` et `3ffe::3` soit protégé par Ipsec (Listing 2).

Dans un second temps il faut indiquer les SPI, les clés ainsi que les algorithmes à utiliser au niveau de la SAD (Listing 3).

Bien entendu, `3ffe::2` et `3ffe::3` doivent comporter les SPDs et SADs correspondantes afin que toute trafic reçu puisse être authentifié et déchiffré. La configuration de ces différents éléments sur `3ffe::2` sera donc proche de Listing 4.

Ainsi tout trafic provenant de 3ffe::1 sera protégé par ESP en mode transport avec les clés et algorithmes définies.

Afin d'activer ces paramètres il vous faudra utiliser `setkey` : `setkey -f setkey.conf`

Vous remarquerez que seuls les échanges depuis `3ffe::1` vers `3ffe::2` ainsi que ceux depuis `3ffe::1` vers `3ffe::3` sont protégés. La réciproque n'est pas vraie ; par exemple les paquets provenant de `3ffe::2` vers `3ffe::1` ne sont en aucun cas protégés. Vous pouvez dès à présent vérifier ces assertions par un ping depuis `3ffe::1` vers `3ffe::3`. Les *Echo Request* doivent être protégés par IPsec tandis que les *Echo Reply* circuleront en clair. Afin de faciliter cette analyse vous pourrez utiliser Wireshark ainsi que le module ESP intégré permettant le déchiffrement des paquets.

**Table 8.** *Liste des RFCs relatives à IPv6*

| Norme | Titre |
|---|---|
| RFC 2403 | The Use of HMAC-MD5-96 within ESP and AH |
| RFC 2404 | The Use of HMAC-SHA-1-96 within ESP and AH |
| RFC 2405 | The ESP DES-CBC Cipher Algorithm With Explicit IV |
| RFC 2409 | The Internet Key Exchange (IKE) |
| RFC 2451 | The ESP CBC-Mode Cipher Algorithms |
| RFC 3566 | The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec |
| RFC 3602 | The AES-CBC Cipher Algorithm and Its Use with IPsec |
| RFC 3686 | Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP) |
| RFC 3775 | Mobility Support in IPv6 |
| RFC 3776 | Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents |
| RFC 3963 | Network Mobility (NEMO) Basic Support Protocol |
| RFC 4109 | Algorithms for Internet Key Exchange version 1 (IKEv1) |
| RFC 4301 | Security Architecture for the Internet Protocol |
| RFC 4302 | IP Authentication Header |
| RFC 4303 | IP Encapsulating Security Payload (ESP) |
| RFC 4306 | Internet Key Exchange (IKEv2) Protocol |
| RFC 4307 | Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2) |
| RFC 4385 | Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) |
| RFC 4835 | Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) |
| RFC 4877 | Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture |
| RFC 4885 | Network Mobility Support Terminology |
| RFC 4886 | Network Mobility Support Goals and Requirements |
| RFC 4887 | Network Mobility Home Network Models |
| RFC 4888 | Network Mobility Route Optimization Problem Statement |
| RFC 4889 | Network Mobility Route Optimization Solution Space Analysis |

## Conclusion

Au travers de ces différents articles vous avez pu vous initier aux divers mécanismes principaux composant IPv6. Ces mécanismes sont relativement nombreux, la modification de la couche réseau nécessite en effet beaucoup d'adaptation. IPv6 est un protocole mature, ses premières bases ont été normalisées en 1998, et n'ont cessée d'être raffinée depuis par l'IETF. La majeure partie des systèmes d'exploitation permettant actuellement de mettre en œuvre ce protocole; le nombre d'adresses IPv4 allouable étant presque épuisé, la transition est inéluctable … c'est donc dès maintenant qu'il s'agit de se familiariser avec ses concepts, sa mise en œuvre et les nouvelles opportunités offertes par IPv6.

### Références

Vous trouverez dans les Tableaux 7 et 8, les références, normes ainsi que des liens Web où vous obtiendrez des renseignements complémentaires sur les divers mécanismes évoqués à travers cet article.

**Frederic Roudaut**
Il travaille actuellement chez *Orange Labs* (anciennement France Telecom R&D) à *Sophia Antipolis* pour le compte *d'Orange Business Services IT&Labs* depuis 1 an et demi.