Implémentation d'une plateforme de sécurisation de la mobilité IPv6 pour un véhicule connecté

Rapport de Stage de Fin d'études 2019

Octobre 2019

Encadré par: Mme Houda LABIOD et Mounira MSAHLI
Présenté par: Mohamed EL MANSURI

Remerciements

Je tiens tout d'abord à remercier à Mme Houda LABIOD, mon maître de stage, pour son temps précieux tout au long du déroulement de stage, parce qu'elle m'a proposé toujours des bonnes solutions dans mon travail de stage

Je remercie aussi Mme Mounira MSAHLI, maître de conférences à Telecom Paris, pour ses conseils, son soutien et remarques éclairées.

Sommaire

Depuis quelques années les possibilités de se connecter à l'Internet se multiplient. Il y a une mobilité induite par l'utilisation de plusieurs technologies d'accès comme l'Ethernet, WiFi et les technologies cellulaires 2G/3G/4G/5G. Les études actuellement conduites par les grands constructeurs et opérateurs pour fournir des infrastructures mobiles utilisant de nouvelles technologies radio, Wi-Fi et WiMax notamment, ont pour objet d'offrir la continuité des services en cours de déplacement, comme le permet la 4g dans le cas de la téléphonie mobile. Cela nécessite que les applications ne soient pas interrompues lorsque un dispositif s'attache à un autre réseau.

Par contre, des désaccords concernant la sécurisation de la mobilité IPv6 comme les risques induits par la mobilité et leur limitation et les différentes optimisations possibles, ont rendu la standardisation de la mobilité IPv6 longue et laborieuse.

Dans ce rapport, nous allons implémenter une plateforme de sécurisation de la mobilité IPv6. Cette plateforme va être composée de 4 dispositifs différents: Home Agent, Mobile Node, Foreign Router et Correspondent Node.

Nous avons utilisé des machines virtuelles pour simuler les 4 dispositifs et tester des scénarios de la mobilité IPv6: Mobilité IPv6 non sécurisée, Mobilité IPv6 sécurisée manuellement et Mobilité IPv6 sécurisée avec IKEv2.

Table des figures

1. Messages “Neighbor Solicitation” et “Neighbor Advertisement”....12
2. Messages “Router Solicitation” et “Router Advertisement”....13
3. En-tête IPv4....15
4. En-tête IPv6....15
5. Structure d’une adresse unicast globale....18
6. Structure de l’adresse “link-local”....19
7. Adresses Multicast “Well-Known”....21
8. Mappage à une adresse à noeud sollicité....22
9. Mappage à une adresse MAC Ethernet....23
10. Trame Ethernet....23
11. Utilisation d’ESP en mode transport....26
12. Utilisation d’AH en mode transport....26
13. Utilisation d’ESP en mode tunnel....26
14. Utilisation d’AH en mode tunnel....27
15. Protocole DSMIPv6....28
16. MN dans le réseau HN....30
17. MN dans un réseau FN....31
18. Le routeur FN refuse de transmettre le paquet....32
19. Tunnel entre le HA et MN....32
20. Communication entre un CN et un MN....33
21. Mécanisme “Proxy Neighbor Discovery”....34
22. Machines virtuelles à utiliser....45
23. Liste des options à activer sur l’interface graphique “menuconfig”....47
24. Modèle d’architecture générale SCOOP@F....50
25. Modèle d’architecture de mobilité IPv6....51
26. Configuration des interfaces HA....52
27. Configuration Radvd pour le HA....53
28. Configuration mip6d....53
29. Configuration des interfaces MN....54
30. Configuration mip6d....54
31. Configuration des interfaces FN....55
32. Configuration Radvd pour le FN....55
33. Configuration des interfaces CN....56
34. Configuration mip6d pour le CN....56
35. Message RA....58
36. Échange des messages BU et BA....59
37. Logs du service mip6d....59
38. Entrée BUL....60
39. Interface tunnel MN....60
40. Interface tunnel HA....60
41. Ping au MN....60
42. Message Home Test Init....61
43. Message Care-of Test Init....62
44. Messages Care-of Test, Home Test et BU....63
45. L’entrée “BUL” du MN....63
46. Ping au MN....64
47. Fichier de configuration setkey....65
48. Message BU chiffré....66
49. Messages envoyés d’un CN....67
50. Configuration IPsec côté HA....68

51. Configuration IPsec côté MN....69
52. Les messages IKE et le message BU....70
53. Logs(1)....71
54. Logs (2)....71
55. Logs (3)....72
56. Interface tunnel....72

Liste des abréviations

IPv6: Internet Protocol version 6
NDP: Neighbor Discovery Protocol
NS: Neighbor Solicitation
NA: Neighbor Advertisement
RS: Router Solicitation
RA: Router Advertisement
DNS: Service de Noms de Domaine
SLAAC: Stateless Address Autoconfiguration
IHL: Internet Header Length
ToS: Type of Service
LLA: Link Local Address
DAD: Duplicate Address Detection
ULA: Unique Local Address
MLD: Multicast Listener Discovery
MN: Mobile Node
FN: Foreign Network
CoA: Care-of Address
HoA: Home Address
NEMO BS: Network Mobility Basic Support
MNP: Mobile Network Prefix
CN: Correspondent Node
HA: Home Agent
BU: Binding Update
BA: Binding Acknowledgment
BE: Binding Error

Remerciements 0

Sommaire 1

Table des figures 2

Liste des abréviations 4

1 Contexte de Stage 8

2 État de l'art 9

2.1 Rappels IPv6 10

2.1.1 Introduction 10

2.1.2 IPv6 est là 10

2.1.3 Raison d'une transition vers l'IPv6 10

2.1.3.1 Épuisement des adresses IPv4 11

2.1.3.2 Accès limité aux client IPv6 11

2.1.3.3 Meilleure performance 11

2.1.4 Transition à IPv6 11

2.1.5 Les bases de l'IPv6 11

2.1.5.1 Système de numération hexadécimal 11

2.1.5.2 ICMPv6 Neighbor Discovery Protocol (NDP) 11

2.1.5.3 Les paquets Neighbor Solicitation (NS) et Neighbor Advertisement (NA) 12

2.1.5.4 Les paquets Router Solicitation (RS) et Router Advertisement (RA) 12

2.1.5.3 Attribution dynamique d'adresses 13

2.1.6 Comparaison d'IPv4 vs IPv6 14

2.1.6.1 Comparaison de l'en-tête IPv4 et l'en-tête IPv6 14

2.1.6.1.1 Le champ IPv4 Internet Header Length (IHL) 15

2.1.6.1.2 Le champ IPv4 Type of Service (ToS) et le champ IPv6 Traffic Class 15

2.1.6.1.3 Champ IPv6 Label 15

2.1.6.2 Fragmentation IPv6: IPv6 source only 15

2.1.6.4 Champ IPv4 Protocol et champ IPv6 Next Header 15

2.1.6.5 Champ IPv4 Time to Live (TTL) et champ IPv6 Hop Limit 15

2.1.6.6 Checksums: IPv4 16

2.1.6.7 Extension Headers 16

2.1.7 Types d'adresses IPv6 16

2.1.7.1 Adresse Unicast 16

2.1.7.1.1 Global Unicast Address 16

La structure d'une adresse unicast globale 17

2.1.7.1.2 Adresse link-local 17

Structure de l'adresse link-local 18

2.1.7.1.3 Les adresses loopback 18

2.1.7.1.4 Les adresses non spécifiées 18

2.1.7.1.5 Les adresses locales uniques 19

2.1.7.1.6 Les adresses IPv4 embarquées 19

2.1.7.2 Les adresses Multicast 19

2.1.7.2.1 Les adresses Well-Known Multicast 19
2.1.7.2.2 Les adresses de multidiffusion de noeud sollicité “Solicited-node multicast addresses” 20
2.1.7.2.2.1 Mapping d’une adresse unicast à une adresse multicast à noeud sollicité 21
2.1.7.2.2.2 Mapping à une adresse MAC Ethernet 21
2.1.7.2.2.2.1 Mapping d’une adresse à noeud sollicité à une adresse MAC ethernet 21
2.1.7.2.2.2.2 Mapping des adresses Well-Known à une adresse MAC Ethernet 22
2.1.7.2.3 Le protocole Multicast Listener Discovery 22
2.2 Rappels IPsec 22
2.2.1 Introduction 22
2.2.2 Fonctionnement IPsec 23
2.2.2.1 Services fournis par IPsec 23
2.2.2.1.1 AH : intégrité et authentification des paquets 23
2.2.2.1.2 ESP : confidentialité, intégrité et authentification des paquets 23
2.2.2.2 Modes transport et tunnel 23
2.3 MIPv6 25
2.3.1 Le bénéfice de la mobilité IP 25
2.3.2 Les technologies supplémentaires à la mobilité IPv6 25
2.3.2.1 Double pile mobilité IPv6 26
2.3.2.2 Support basique pour la mobilité des réseaux 26
2.3.3 Types de noeuds 27
2.3.4 La procédure de la mobilité IPv6 27
2.3.4.1 Home registration 27
2.3.4.2 Bidirectional Tunneling 29
2.3.4.3 Interception des paquets pour le Mobile Node 30
2.3.4.4 Returning Home 31
2.3.4.5 Route optimization 32
2.3.5 Dynamic Home Agent Address Discovery 33
2.3.6 Mobile Prefix Solicitation/Advertisement 33
2.3.7 IPsec 34
2.3.7.1 MIPv6 avec IKEv2 35
2.3.7.1.1 Les formats des paquets 35
2.3.7.1.2 Exigences 36
2.3.7.1.2.1 Exigences de la politique 36
2.3.7.2 Exigences IKEv2 36
2.3.7.3 Configuration dynamique 37
2.3.7.3.1 PAD 37
2.3.7.3.2 SPD 37
2.3.7.3.2.1 BU et BA 37
2.3.7.3.2.2 Les messages “Return Routability” 38
2.3.7.3.2.3 Les messages “Mobile Prefix Discovery” 38

2.3.7.3.2.4 Les paquets Payload 39
2.3.7.4 Négociation des SAs en utilisant IKEv2 39

3 Implémentation 40
3.1 Installation 41
3.1.1 VirtualBox 41
3.1.2 Kernel 41
3.1.3 UMIP 43
3.1.4 StrongSwan 44
3.2 Scénario 45
3.2.1 Modèle d'architecture générale SCOOP@F 45
3.2.2 Modèle d'architecture de mobilité IPv6 46
3.3 Implémentation de la Mobilité IPv6 non sécurisé 47
3.3.1 Configuration du Home Agent 47
3.3.2 Configuration du Mobile Node 49
3.3.3 Configuration du Foreign Router 50
3.3.4 Configuration du CN 51
3.3.5 Lancer l'exécution 52
3.3.6 Résultats 52
3.4 Implémentation de la Mobilité IPv6 sécurisé manuellement 59
3.4.1 Lancer le service 60
3.4.2 Résultats 60
3.5 Implémentation de la Mobilité IPv6 sécurisé avec IKEv2 61
3.5.1 Configuration du Home Agent 61
3.5.2 Configuration du MN 62
3.5.3 Lancer le service 63
3.5.4 Résultats 63

4 Conclusions 66

5 Références 67

1 Contexte de Stage

Mon stage s'est déroulé au département INFRES (département d'informatique et réseaux) de l'école Télécom Paris dans l'équipe CCN (Cyber Security for Communication and Networking).

L'objectif de ce stage est de mettre en place une solution de sécurité pour les communications V2X. Cette solution de sécurité de niveau 3 permet de prendre en compte la mobilité des véhicules dans le contexte d'un système de transport intelligent coopératif déployé.
Nos travaux s'insèrent dans le cadre de deux projets de déploiement de systèmes ITS coopératif SCOOP@F et C-Roads.

Nous avons implémenté une plateforme de simulation en utilisant des machines virtuelles qui a permis de réaliser plusieurs scénarios de test.

Dans ce rapport, nous allons commencé par un état de l'art pour mentionner des rappels sur l'IPv6 et pour expliquer le fonctionnement de la mobilité IPv6, après expliquer les différentes implémentations qui ont été faites et finalement, une conclusion.

2 État de l'art

Ce chapitre a pour objectif d'expliquer les bases de l'IPv6 et la mobilité IPv6. Il est décomposé en deux sections. La section 2.1 comporte une introduction de l'IPv6, son origine, les motifs de la transition, quelques concepts sur la transition, les bases de l'IPv6, une comparaison de l'IPv4 avec l'IPv6 et les types d'adresses IPv6. La section 2.2 comporte le bénéfice de la mobilité IPv6, les technologies supplémentaires à la mobilité IPv6, les types de noeuds, la procédure de la mobilité IPv6, "Dynamic Home Agent Discovery", "Mobile Prefix Sollicitation et l'IPsec.

2.1 Rappels IPv6

Cette section comporte une introduction de l'IPv6, son origine, les motifs de la transition, quelques concepts sur la transition, les bases de l'IPv6, une comparaison de l'IPv4 avec l'IPv6 et les types d'adresses IPv6.

2.1.1 Introduction

IPv6(Internet Protocol version 6), le successeur d'IPv4, est un protocole attendu depuis longtemps. Pour plus de 35 ans, IPv4 a été une partie intégrante de l'évolution d'Internet. IPv4 fournit 4.29 milliards d'adresses IP, un adressage qui semblait être suffisant quand IPv4 est devenu un standard en 1980. Il y avait environ 4.5 milliards de personnes, alors si chacun dans la planète avait besoin d'une adresse IP, il n'y avait aucun problème. Mais cela était conçu quand il y avait que 600 utilisateurs sur Internet, même avant World Wide Web, email, vidéo streaming, et des autres innovations. Rapidement l'espace d'adressage est devenu limité et insuffisant et il fallait remédier rapidement à cette limitation.

2.1.2 IPv6 est là

Les adresses IP utilisées aujourd'hui dépassent largement la population mondiale pour plusieurs raisons.

L'IPv6 fournit plus d'adresses qu'IPv4. Comme nous avons déjà évoqué, IPv4, avec son format 32 bits, fournit que 4.29 milliards d'adresses. En comparaison avec le format 128 bits de l'IPv6, nous avons un espace de 340 sextillions d'adresses.

IPv6 est maintenant activé par défaut sur la plupart des systèmes d'exploitation, incluant Windows, Mac Os et Linux. Tous les systèmes d'exploitation pour mobiles ont la fonction IPv6 activée par défaut, incluant Google Android, Apple iOS et Windows Mobile.

2.1.3 Raison d'une transition vers l'IPv6

Continuer à ignorer IPv6 va causer de grands répercussions sur les réseaux d'entreprises. Dans la suite de cette section, nous livrons les raisons principales de la nécessité de migration de l'IPv4 à l'IPv6.

2.1.3.1 Épuisement des adresses IPv4

Les adresses IPv4 disponibles s'épuisent. Tôt ou tard, les opérateurs de réseau auront besoin de migrer leurs réseaux à l'IPv6.

2.1.3.2 Accès limité aux client IPv6

Il y a déjà des établissements dans le monde qui utilisent seulement l'IPv6. Dans quelques endroits aujourd'hui, ce n'est pas possible d'obtenir une adresse IPv4.

2.1.3.3 Meilleure performance

Des rendements accrus peuvent être un autre bénéfice pour migrer à l'IPv6. Beaucoup de fournisseurs constatent des progrès substantiels en performance avec IPv6.

2.1.4 Transition à IPv6

Il n'y a pas une date spécifique pour basculer de l'IPv4 à l'IPv6. IPv6 probablement coexistera avec l'IPv4 pendant de nombreuses années, car beaucoup de réseaux d'entreprises continuent à utiliser IPv4. La transition à l'IPv6 est en cours et aura une tendance plus marquée.

IPv6 a une grande variété de outils pour aider à la transition de l'IPv4 à l'IPv6, incluant la tunnelisation et le NAT. La tunnelisation encapsule un paquet IPv6 dans un paquet IPv4 pour qu'il soit envoyé sur un réseau IPv4. La technologie NAT fournit un mécanisme pour transformer une adresse IPv6 dans une adresse IPv4.

2.1.5 Les bases de l'IPv6

Cette section introduit les concepts de base de l'IPv6: le système de numération hexadécimal, types d'adresses IPv6, "ICMPv6 Neighbor Discovery Protocol" (NDP) et le protocole dynamique d'attribution d'adresses.

2.1.5.1 Système de numération hexadécimal

La longueure d'une adresse IPv6 est de 128 bits, et comme nous le verrons, le système hexadécimal est le système idéal pour représenter une longue chaîne de bits.

Un seul chiffre hexadécimal peut représenter 4 bits et deux chiffres hexadécimaux peuvent représenter un octet. Pour cette raison, le système hexadécimal est couramment utilisé dans le domaine des sciences informatiques.

2.1.5.2 ICMPv6 Neighbor Discovery Protocol (NDP)

"ICMPv6 Neighbor Discovery Protocol" (NDP) ajoute une nouvelle fonctionnalité à l'ICMPv6. NDP est utilisé pour la découverte des périphériques sur le même sous-réseau. NDP inclut 5 types de

paquets: “Router Solicitation”, “Router Advertisement”, “Neighbor Solicitation”, “Neighbor Advertisement” et les messages “Redirect”.

2.1.5.3 Les paquets Neighbor Solicitation (NS) et Neighbor Advertisement (NA)

Les messages “Neighbor Solicitation” et “Neighbor Advertisement” sont utilisés pour envoyer des messages entre deux dispositifs dans un même sous-réseau. Par exemple ces messages sont utilisés pour la résolution d’adresse et ce sont les équivalents pour l’ARP de l’IPv4. Comme le montre la figure 1, les paquets NS et NA sont comparables respectivement aux messages “ARP Request” et “ARP Reply”.

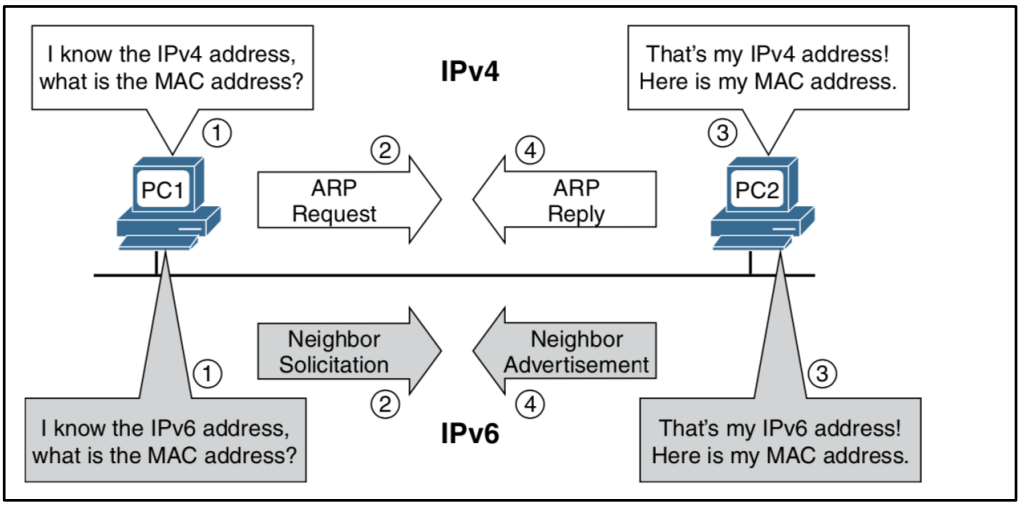


Figure 1. Messages “Neighbor Solicitation” et “Neighbor Advertisement”. [1]

2.1.5.4 Les paquets Router Solicitation (RS) et Router Advertisement (RA)

Les messages “Router Solicitation” et “Router Advertisement” sont utilisés pour envoyer des messages entre un dispositif et un routeur dans un même sous-réseau. Le message RA est envoyé par le routeur et il est considéré comme une proposition aux dispositifs sur comment obtenir des informations concernant l’adresse IPv6. Le message RS est envoyé par un dispositif pour demander un message RA du routeur, comme illustré sur la figure 2.

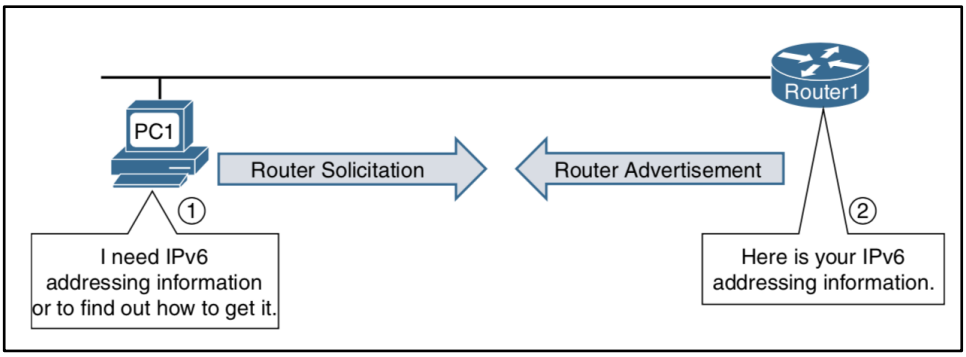


Figure 2. Les messages “Router Solicitation” et “Router Advertisement”. [1]

2.1.5.3 Attribution dynamique d'adresses

En IPv4, les dispositifs ont deux moyens pour obtenir les informations concernant l'adressage IPv4, y compris l'adresse IPv4, masque de sous-réseau, adresse de passerelle par défaut, nom de domaine et service de noms de domaine (DNS):

1. Via une configuration statique ou manuelle,
2. ou dynamiquement depuis un serveur DHCPv4.

Les adresses IPv6 peuvent être assignés statiquement comme dans IPv4. Cependant, pour attribuer les adresses dynamiquement, IPv6 a une approche différente. IPv6 utilise le message "ICMPv6 Router Advertisement" pour proposer aux dispositifs comment obtenir les informations concernant l'adressage IPv6. Le routeur IPv6 envoie périodiquement un message RA (chaque 200 secondes sur Cisco IOS) ou bien quand il reçoit une requête RS d'un dispositif. Le message RA est habituellement envoyé à tous les dispositifs IPv6.

Le message RA inclut des informations concernant l'adressage IPv6 pour les dispositifs:

- Le préfixe du sous-réseau, la longueur du préfixe de sous-réseau, et autres informations concernant le sous-réseau.
- L'adresse de passerelle par défaut. (Adresse link-local du routeur, l'adresse source du message RA)
- 3 drapeaux utilisés pour suggérer à un dispositif comment obtenir les informations concernant l'adressage IPv6. Ces drapeaux sont: le "Autonomous Address Configuration Flag" (drapeau A), le "Other Configuration Flag" (drapeau O), et le "Managed Address Configuration Flag" (drapeau M).
- Des informations optionnelles comme le nom du domaine et une liste des serveurs DNS.

Contrairement à un dispositif IPv4, un dispositif IPv6 peut déterminer son adresse IP sans les services d'un serveur DHCP.

Le message RA va proposer d'utiliser l'une de ces trois méthodes:

- Méthode 1: "Stateless Address Autoconfiguration" (SLAAC). Le dispositif utilise les informations contenues dans le message RA pour tous ses besoins d'adressage, incluant le préfixe pour créer une adresse globale unicast. Le dispositif va utiliser l'adresse source du message RA comme l'adresse de passerelle par défaut. Cette méthode est la méthode par défaut sur les équipements Cisco IOS.

- Méthode 2: SLAAC et "Stateless DHCPv6 Server". Cette méthode est similaire à la précédente, l'équipement utilise la méthode SLAAC pour créer une adresse globale unicast et utilise l'adresse source du message RA pour configurer l'adresse de la passerelle par défaut. Cependant, cette méthode suggère en plus de contacter le serveur DHCPv6 pour recevoir des informations additionnelles qui ne sont pas contenues dans le message RA. Ces informations peuvent être une liste des adresses IP de serveurs DNS. Il est important de noter que le serveur "Stateless DHCPv6" ne fournit pas ou maintient les informations concernant l'adresse globale unicast IPv6. Ce serveur fournit seulement les informations du sous-réseau qui sont communs à tous les dispositifs du sous-réseau.

- Méthode 3: "Stateful DHCPv6 server". Cette méthode est similaire au serveur DHCP de l'IPv4. Le message RA suggère au dispositif d'utiliser un serveur DHCPv6 pour tous ses

besoins. Cependant, le dispositif peut obtenir dynamiquement l'adresse de la passerelle par défaut à partir de l'adresse source, comme dans les deux méthodes précédentes.

Les méthodes 1 et 2 suggèrent aux dispositifs d'utiliser le SLAAC pour créer leur propre adresse IPv6 globale unicast. Le dispositif utilise le préfixe affiché sur le message RA pour créer l'ID d'interface (64 bits), et ce dernier peut être créé de deux manières différentes:

- Valeur aléatoire 64 bits: Le système d'exploitation peut générer une valeur aléatoire de 64 bits pour l'ID d'interface (cette manière est activé par défaut sur les systèmes d'exploitation Windows).

- EUI-64 (Extended Unique Identifier): Cette méthode utilise les 48 bits de l'adresse MAC Ethernet et introduit 16 bits (FFFE) au milieu. Le septième bit est inversé.

2.1.6 Comparaison d'IPv4 vs IPv6

2.1.6.1 Comparaison de l'en-tête IPv4 et l'en-tête IPv6

La figure 3 illustre la structure d'une en-tête IPv4, comme défini dans le *RFC 791*[1]. Comme le montre la figure, certains champs sont les mêmes ou similaires à l'en-tête IPv6, tandis que d'autres ont été retirés.

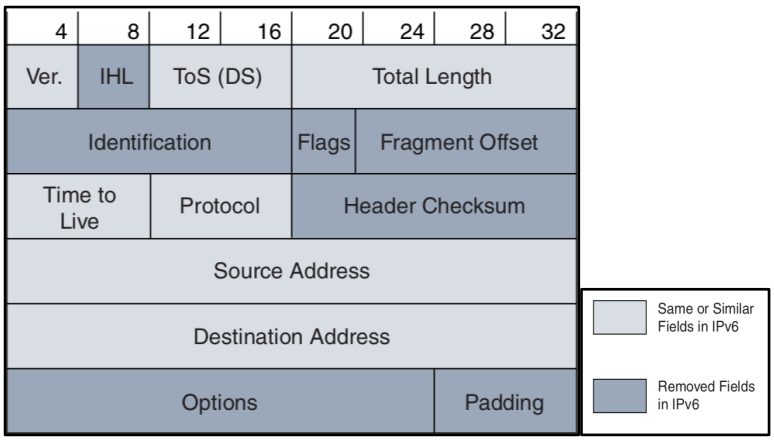
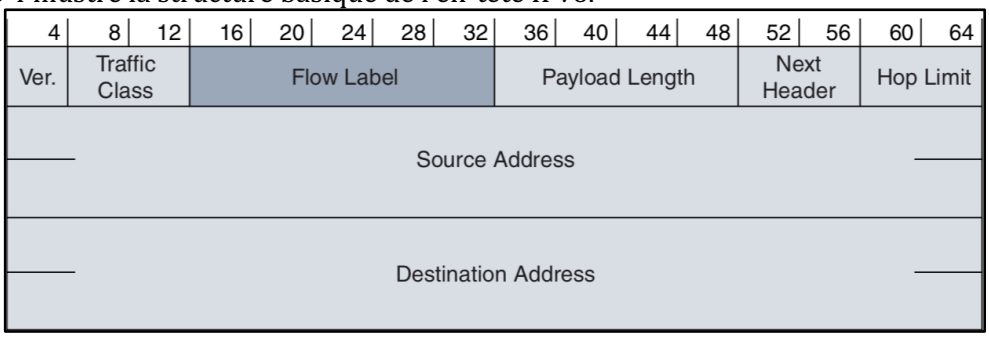


Figure 3. En-tête IPv4. [1]

La figure 4 illustre la structure basique de l'en-tête IPv6.



[1] https://tools.ietf.org/html/rfc791

Figure 4. En-tête IPv6. [1]

Une comparaison rapide de ces deux en-têtes révèle que l'en-tête IPv6 est une structure plus simple présentant moins de champs que l'en-tête IPv4. Cela rend IPv6 un protocole moins compliqué et permet de réaliser un traitement plus efficace.

2.1.6.1.1 Le champ IPv4 Internet Header Length (IHL)

Le champ IHL est la longueur de l'en-tête IPv4 en termes d'octets. Les champs "Options" et "Padding" peuvent agrandir la longueur du champ IHL qui dépasse les 20 octets, au maximum 60 octets. IPv6 n'a pas un champ IHL parce que l'en-tête principale IPv6 a une longueur fixe de 40 octets.

2.1.6.1.2 Le champ IPv4 Type of Service (ToS) et le champ IPv6 Traffic Class

Les champs ToS et "Traffic Class" sont identiques; seulement le nom a été changé en IPv6. Ces champs sont utilisés pour spécifier quel type de traitement reçoivent les paquets des routeurs.

2.1.6.1.3 Champ IPv6 Label

Ce champ est nouveau et il est utilisé pour l'étiquetage d'une séquence ou flux de paquets IPv6 envoyés d'une source à une ou plusieurs destinations.

2.1.6.2 Fragmentation IPv6: IPv6 source only

Contrairement à l'IPv4, un routeur IPv6 ne fait pas la fragmentation d'un paquet sauf s'il est la source de ce paquet. Les routeurs intermédiaires ne font jamais la fragmentation.

Quand un routeur IPv6 reçoit un paquet plus grand que la MTU de son interface de sortie, le routeur décline ce paquet et envoie un message de type "ICMPv6 Packet Too Big" à la source. Ce message inclut la taille MTU du lien en termes d'octets pour que la source puisse changer la taille du paquet et le transmettre.

2.1.6.4 Champ IPv4 Protocol et champ IPv6 Next Header

Le champ IPv4 "Protocol" précise le protocole transporté dans la partie "payload" du paquet IPv4. IPv6 possède un champ similaire, le champ Next Header, qui spécifie le type de l'en-tête attendu après la principale en-tête IPv6.

2.1.6.5 Champ IPv4 Time to Live (TTL) et champ IPv6 Hop Limit

Ces champs vont assurer que les paquets ne transitent pas entre les réseaux pour une période indéfinie.

2.1.6.6 Checksums: IPv4

Le checksum d'une en-tête IPv4 est fourni pour protéger contre la corruption d'un paquet. Ce n'est pas la même chose que CRC utilisé par la couche Ethernet, mais il est plus simple (16 bits). Chaque routeur vérifie et recalcule ce champ. Si le checksum échoue, alors le routeur va jeter le paquet.

Il n'y a pas un champ Checksum dans l'en-tête IPv6. Les concepteurs de l'IPv6 n'ont pas inclu ce champ parce que cette fonctionnalité d'assurer l'intégrité du paquet est déjà présente dans la couche 2.

2.1.6.7 Extension Headers

Les "Extension Headers" sont une importante addition dans l'IPv6. Elles sont des en-têtes optionnelles qui ajoute de la flexibilité et permettent des futures améliorations pour l'IPv6, la principale en-tête IPv6 inclut un champ Next Header et ce champ possède deux objectifs:

- Identifier le protocole porté dans la partie "payload".
- Identifier la présence d'un Extension Header.

2.1.7 Types d'adresses IPv6

Les types d'adresses IPv6 sont définis dans le *RFC 4291*[2]. Dans cette partie nous allons analyser les différents types d'adresses unicast, multicast et anycast.

2.1.7.1 Adresse Unicast

Parmi les adresses unicast les plus importantes, nous trouvons les adresses globales unicast, qui sont équivalentes aux adresses publiques IPv4, et les adresses "link-local". Retenez que l'IPv6 n'inclut pas les adresses broadcast.

Une adresse unicast identifie uniquement une interface sur un dispositif IPv6. Un paquet envoyé à une adresse unicast est reçu sur l'interface à laquelle cette adresse a été attribuée.

2.1.7.1.1 Global Unicast Address

Les adresses unicast globales sont globalement routables et joignables sur l'internet IPv6, elles sont équivalentes aux adresses publiques IPv4.

[2] https://tools.ietf.org/html/rfc4291

La structure d'une adresse unicast globale

La figure 5, illustre la structure générique d'une adresse unicast globale, incluant le Préfixe de site, l'ID de sous-réseau et l'ID d'Interface.

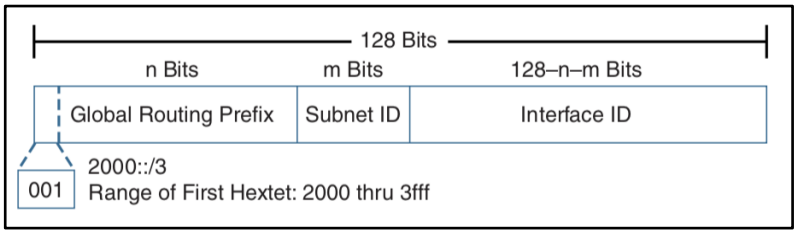


Figure 5. Structure d'une adresse unicast globale. [1]

Les adresses unicast globales courantes sont attribuées par IANA et elles commencent par la valeur binaire `001`, ou le préfixe `2000::/3`.

Une adresse unicast globale est configuré sur une interface, qui peut être configurée avec une ou plusieures adresses unicast globales.

Cela peut s'expliquer de deux façons. Tout d'abord, votre dispositif doit créer sa propre adresse unicast globale en utilisant SLAAC et peut recevoir une adresse différente en utilisant le mode "stateful DHCPv6". De deux, quand votre système d'exploitation attribue une adresse unicast globale en utilisant SLAAC, il peut créer une autre adresses unicast globale IPv6 temporaire additionnelle.

Il faut se rappeler qu'une interface IPv6 ne doit pas être configurée forcément avec une adresse unicast globale. Par contre, cette interface doit avoir une adresse de type "link-local". En d'autres termes, si une interface possède une adresse unicast globale, elle doit posséder aussi une adresse de type "link-local". Cependant, si une interface possède une adresse "link-local", elle ne doit pas posséder nécessairement une adresse unicast globale.

2.1.7.1.2 Adresse link-local

Les adresses "link-local" (LLA) sont des adresses unicast qui sont limitées à un seul lien. Le terme lien se réfère à un sous-réseau. Par conséquent, les adresses link-local sont seulement locales à un lien ou sous-réseau particulier et qui ne sont pas joignables dehors le sous-réseau. Les adresses "link-local" ont besoin d'être unique seulement dans le même sous-réseau parce que ces paquets ne peuvent pas être routables dehors le sous-réseau. En d'autres termes, les routeurs ne doivent pas transmettre aucun paquet qui a une adresse source ou destination de type link-local.

Les adresses link-local donnent à l'IPv6 des caractéristiques uniques que l'IPv4 ne peut pas offrir. Par exemple, un dispositif peut auto-configurer sa propre adresse IPv6 "link-local" au démarrage, sans un serveur DHCPv6. Cette adresse peut être utilisé pour communiquer avec les dispositifs qui sont dans le même sous-réseau, y compris le routeur local ou le serveur DHCPv6, pour obtenir une adresse unicast globale.

Les adresses "link-local" IPv6 ont plusieurs usages, y compris les suivants:

- Avant d'obtenir une adresse unicast globale, un dispositif utilise son adresse "link-local" comme une adresse source pour communiquer avec le routeur local et le serveur DHCPv6 pour obtenir une adresse unicast globale.

- Les dispositifs reçoivent dynamiquement leurs adresses de la passerelle par défaut à partir des messages “ICMPv6 Router Advertisements” émis par le routeur. L’adresse de la passerelle par défaut est l’adresse link-local du routeur présent sur le même sous-réseau.
- Les routeurs qui exécutent des protocoles comme les EIGRP pour l’IPv6 et l’OSPFv3, utilisent leurs adresses link-local pour envoyer des messages et pour établir leurs relations d’adjacence.
- Les tables de routage IPv6 avec les préfixes qui sont appris à partir des protocoles dynamiques de routage utilisent les adresses link-local comme l’adresse du saut suivant.

Structure de l’adresse link-local

La figure 6 illustre le format d’une adresse unicast “link-local”. Les adresses unicast “link-local” se trouvent dans la plage `FE80::/10`. Les 64 bits les moins significatifs sont utilisés pour l’ID d’interface.

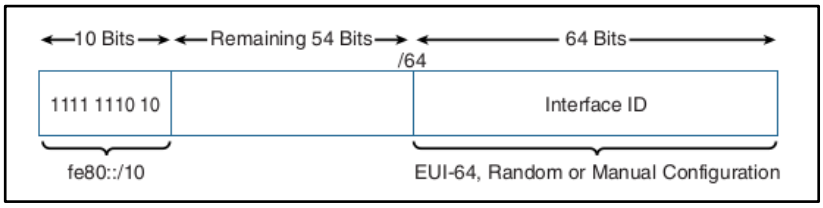


Figure 6. Structure de l’adresse “link-local”. [1]

2.1.7.1.3 Les adresses loopback

Une adresse loopback est un autre type d’adresse unicast. Une adresse loopback est `::1`, `127` zéros sauf le dernier bit qui est fixé à 1. Elle est équivalente à la plage d’adresse IPv4 `127.0.0.0/8`, généralement `127.0.0.1`.

Les adresses loopback présentent les caractéristiques suivantes:

- Une adresse loopback ne peut pas être attribuée à une interface physique.
- Un paquet qui a une adresse loopback (source ou destination) ne doit jamais être envoyé.
- Un routeur ne doit jamais envoyer un paquet qui a une adresse destination de type loopback.
- Le dispositif doit jeter un paquet reçu sur une interface si l’adresse de destination du paquet est de type loopback.

2.1.7.1.4 Les adresses non spécifiées

Une adresse non spécifiée est une adresse composée par `128` zéros. Une adresse non spécifiée est utilisée comme une adresse source pour indiquer l’absence d’une adresse. Elle ne peut pas être attribuée à une interface.

Un exemple ou l’adresse non spécifiée peut être utilisé c’est dans les messages “ICMPv6 Duplicate Address Detection” (DAD).

Les adresses non-spécifiées ont les caractéristiques suivantes:

- Une adresse source non spécifiée indique l’absence d’une adresse.

- Une adresse non spécifiée ne peut pas être attribuée à une interface physique.
- Une adresse non spécifiée ne peut pas être utilisé comme une adresse destination.
- Un routeur ne va jamais transmettre un paquet qui a une adresse source non spécifiée.

2.1.7.1.5 Les adresses locales uniques

Les adresses locales uniques (ULA) peuvent être utilisées de la même manière qu'une adresse unicast globale mais seulement pour des usages privés et ne doivent pas être routables sur Internet. Les adresses ULA sont utilisées uniquement dans un domaine limité, comme par exemple à l'intérieur un site d'entreprise ou plusieurs domaines administratifs. Les adresses ULA sont utilisés par les dispositifs qui n'ont jamais besoin de sortir à l'Internet et qui n'ont jamais besoin d'être joignables depuis l'Internet.

Les adresses ULA ont les caractéristiques suivantes:

- Elles peuvent être utilisées comme les adresses unicast globales.
- Elles peuvent être utilisées pour les dispositifs qui n'ont jamais besoin d'accéder à l'Internet.
- Elles permettent de connecter différents sites d'une manière privée sans conflit d'adresses.
- Elles sont indépendantes de n'importe quel ISP et elles peuvent être utilisées sans avoir une connexion Internet.

2.1.7.1.6 Les adresses IPv4 embarquées

Les adresses IPv4 embarquées sont des adresses IPv6 utilisées pour aider dans la transition de l'IPv4 à l'IPv6. Les adresses IPv4 embarquées transportent une adresse IPv4 dans les 32 bits de poids faible. Ces adresses sont utilisées pour représenter une adresse IPv4 au sein d'une adresse IPv6.

2.1.7.2 Les adresses Multicast

Multicast est une technique qui permet à un dispositif d'envoyer un seul paquet à plusieurs destinations en même temps, par opposition aux adresses unicast, qui envoient un seul paquet à une seule destination.

Multicast peut être une meilleure option que le "Broadcast" quand les destinataires sont seulement un sous-ensemble de tous les dispositifs qui se trouvent dans un sous-réseau. Les paquets multicast peuvent être filtrés au niveau 2 par une carte réseau Ethernet.

Une adresse multicast IPv6 possède le préfixe `FF00::/8` et définie un groupe de dispositifs connu sous le nom groupe multicast. C'est l'équivalent de l'IPv4 `224.0.0.0/4`. Un paquet envoyé à un groupe multicast toujours il présente une adresse source de type unicast. Une adresse multicast peut seulement être une adresse destination et jamais une adresse source.

2.1.7.2.1 Les adresses Well-Known Multicast

Les adresses “Well-Known Multicast” sont prédéfinies pour des groupes multicast déjà assignés. Les adresses “Well-Known Multicast” ont le préfixe `FF00::/12`. Ces adresses sont équivalentes aux adresses “Well-Known Multicast IPv4” qui se trouvent dans la plage `224.0.0.0 - 239.255.255.255`. Ce type d’adresses est généralement utilisé pour le protocole “Neighbor Discovery” et les messages de protocoles.

Sur le *RFC 2375*[3] vous pouvez observez les groupes multicast déjà attribués. La figure 7 illustre les adresses de différents groupes.

/8 Prefix	Flag	Scope	Predefined Group ID	Compressed Format	Description
Interface-Local Scope					
ff	0	1	0:0:0:0:0:0:1	ff01::1	All-nodes
ff	0	1	0:0:0:0:0:0:2	ff01::2	All-routers
Link-Local Scope					
ff	0	2	0:0:0:0:0:0:1	ff02::1	All-nodes
ff	0	2	0:0:0:0:0:0:2	ff02::2	All-routers
ff	0	2	0:0:0:0:0:0:5	ff02::5	OSPF routers
ff	0	2	0:0:0:0:0:0:6	ff02::6	OSPF designated routers
ff	0	2	0:0:0:0:0:0:9	ff02::9	RIP routers
ff	0	2	0:0:0:0:0:0:a	ff02::a	EIGRP routers
ff	0	2	0:0:0:0:0:1:2	ff02::1:2	All-DHCP agents
Site-Local Scope					
ff	0	5	0:0:0:0:0:0:2	ff05::2	All-routers
ff	0	5	0:0:0:0:0:1:3	ff05::1:3	All-DHCP servers

Figure 7. Adresses Multicast “Well-Known”. [1]

2.1.7.2.2 Les adresses de multidiffusion de noeud sollicité “Solicited-node multicast addresses”

Les adresses de multidiffusion de noeud sollicité sont automatiquement créées et attribuées à une interface pour chaque adresse unicast globale, locale unique, et “link-local” sur cette interface. Ces adresses sont automatiquement créées en utilisant un mapping spéciale de l’adresse unicast du périphérique avec le préfixe multicast par noeud sollicité `FF02:0:0:0:0:1:FF00::/104`.

L’un des avantages d’utiliser l’adresse multicast de la couche 3 est son mapping à une adresse MAC Ethernet. Cela permet que la trame soit filtrée par un switch. Cela signifie que ces paquets vont être transmis seulement aux périphériques qui sont membres d’un groupe multicast.

Ces adresses sont utilisées pour deux mécanismes essentiels de l’IPv6, les deux font partie du protocole “Neighbor Discovery Protocol” (NDP):

- La résolution d’adresse: Accomplissant beaucoup la même fonction qu’un ARP Request présent à l’IPv4, un dispositif IPv6 envoie un message Neighbor Solicitation à une adresse multicast à noeud sollicité pour apprendre l’adresse physique du dispositif qui est dans le même sous-réseau. Le dispositif connaît bien l’adresse IPv6 de la destination dans le lien mais il a besoin de connaître son adresse physique.

[3] https://tools.ietf.org/rfc/rfc2375.txt

- Détection d'adresses dupliquées (DAD): DAD permet à un dispositif de vérifier que son adresse unicast est unique dans un sous-réseau. Un message Neighbor Solicitation est envoyé aux adresses multicast à noeud sollicité pour déterminer si personne d'autre a la même adresse.

2.1.7.2.2.1 Mapping d'une adresse unicast à une adresse multicast à noeud sollicité

Comme nous l'avons mentionné, une adresse multicast à noeud sollicité est automatiquement créée pour chaque adresse unicast (adresse unicast globale, locale unique, et "link-local") dans une interface. Le préfixe de multicast par noeud sollicité (`FF02:0:0:0:0:1:FF00::/104`) est précédé par les 24 bits de faible poids de l'adresse unicast (comme illustré sur la figure 8).

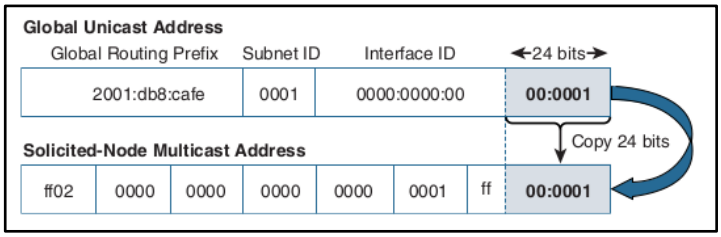


Figure 8. Mappage à une adresse à noeud sollicité. [1]

2.1.7.2.2.2 Mapping à une adresse MAC Ethernet

Les adresses multicast IPv6, y compris les adresses à noeud sollicité et les adresses "Well-Known", sont mappées à une adresse MAC Ethernet. C'est là que vous allez constater le vraie avantage d'utiliser une adresse multicast au lieu d'utiliser une adresse broadcast.

2.1.7.2.2.2.1 Mapping d'une adresse à noeud sollicité à une adresse MAC ethernet

Les figures 8 et 9 illustrent la continuation du mapping d'une adresse unicast à une adresse à noeud sollicité à une adresse multicast MAC ethernet

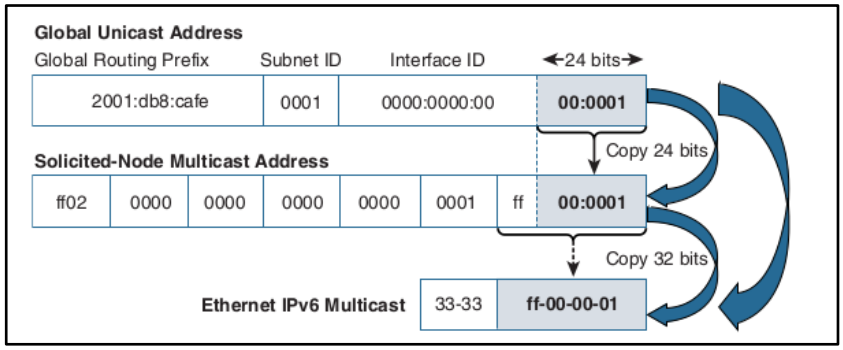


Figure 9. Mappage à une adresse MAC Ethernet. [1]

Les adresses multicast à noeud sollicité ont été créées en utilisant les 24 bits de poids faible de l'adresse unicast. Ensuite, les adresses MAC Ethernet ont été créées en utilisant les 32 bits de poids faible des adresses multicast à noeud sollicité.

Plusieurs dispositifs vont partager la même adresse multicast à noeud sollicité parce que cette adresse a été créée seulement à partir de 24 bits de faible poids au lieu d'utiliser les 64 bits de l'ID d'interface. Cela ne cause aucun problème. Le protocole de la couche supérieur, comme le message ICMPv6 Neighbor Solicitation, contient l'adresse IPv6 de la destination. La destination concernée, le dispositif que son adresse IPv6 correspond à l'adresse de destination, continue à traiter le paquet et les autres dispositifs arrêtent simplement de traiter le paquet.

Cela signifie également que la carte réseau NIC va accepter toutes les trames Ethernet qui ont une adresse multicast mappée à une adresse multicast à noeud sollicité.

La trame Ethernet qui porte le paquet IPv6 (Figure 10) et le message Neighbor Solicitation possède une adresse multicast mappée sur l'adresse physique de destination.

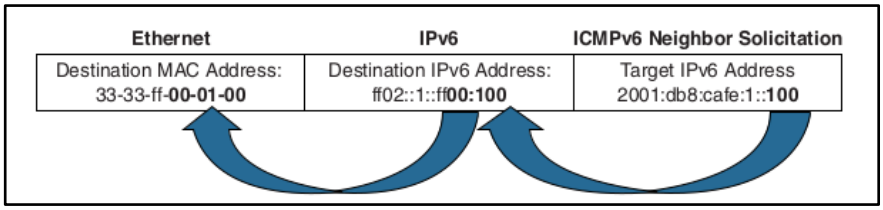


Figure 10. Trame Ethernet. [1]

2.1.7.2.2.2.2 Mapping des adresses Well-Known à une adresse MAC Ethernet

Les adresses multicast "Well-known" sont aussi mappées à des adresses MAC Ethernet.

2.1.7.2.3 Le protocole Multicast Listener Discovery

Les routeurs IPv6 utilisent le protocole "Multicast Listener Discovery" (MLD) pour découvrir les clients multicast dans un sous-réseau. Quand un routeur IPv6 reçoit un paquet multicast dans une interface, il va falloir transmettre ce paquet depuis une ou plusieurs interfaces. Quand une interface correspond à un réseau LAN, le routeur va avoir besoin de déterminer si certains des dispositifs qui se trouvent dans le réseau LAN sont membres du paquet multicast. Le routeur IPv6 va utiliser le protocole MLDv2 pour cette fin.

2.2 Rappels IPsec

Cette section comporte l'introduction d'IPsec et son fonctionnement. Cette section introduit des bases sur IPsec pour comprendre bien comment les communication de la mobilité IPv6 sont sécurisées.

2.2.1 Introduction

IPsec permet, par encapsulation, de protéger en confidentialité, intégrité et anti-rejeu un flux au niveau de la couche réseau.

2.2.2 Fonctionnement IPsec

IPsec, de par ses subtilités, est souvent partiellement compris et peu maîtrisé. Les choix de configuration, y compris ceux par défaut, ne sont pas toujours judicieux et l'emploi d'IPsec peut alors offrir un niveau de sécurité plus faible que celui attendu.

2.2.2.1 Services fournis par IPsec

Les services de sécurité fournis par IPsec reposent sur deux protocoles différents qui constituent le coeur de la technologie IPsec :

AH : « Authentication Header » (protocole n°51).
ESP : « Encapsulation Security Payload » (protocole n°50).

Ces deux protocoles peuvent être utilisés indépendamment ou, plus rarement, de manière combinée.

2.2.2.1.1 AH : intégrité et authentification des paquets

Le protocole AH, qui est utilisé de manière moins fréquente qu'ESP, permet d'assurer l'intégrité et, employé avec IKE, l'authentification des paquets IP. C'est à dire qu'AH permet d'une part de s'assurer que les paquets échangés n'ont pas été altérés et d'autre part de garantir l'identité de l'expéditeur d'un paquet. Il garantit aussi une protection contre le rejeu.

2.2.2.1.2 ESP : confidentialité, intégrité et authentification des paquets

Le protocole ESP permet quant à lui d'assurer la confidentialité, l'intégrité et, employé avec IKE (voir infra), l'authentification des données échangées. Il garantit aussi une protection contre le rejeu. Il est possible d'utiliser uniquement les fonctions d'intégrité et d'authentification sans chiffrement (ce qui peut satisfaire la plupart des cas d'usage d'AH et justifie donc l'abandon d'AH). Certaines implémentations permettent à l'inverse la protection en confidentialité sans mécanisme de contrôle d'intégrité : cet usage, lui-aussi obsolescent, doit être évité. La suppression du service d'intégrité ne présente aucun avantage (le coût en performance des opérations de contrôle d'intégrité est en général négligeable devant celui du chiffrement) et expose l'utilisateur à un certain nombre d'attaques connues et réalistes.

2.2.2.2 Modes transport et tunnel

Indépendamment du choix entre AH et ESP, il est possible d'utiliser IPsec dans deux modes distincts: le mode tunnel et le mode transport.

Dans le mode transport, les données associées à AH ou à ESP viennent se greffer sur le paquet IP initial (c'est à dire celui qu'on aurait envoyé en l'absence d'IPsec). Le paquet IP résultant contient un paquet AH ou ESP qui contient lui-même le contenu du paquet initial (un segment TCP par exemple). On peut remarquer que l'en tête IP initiale doit être modifiée: son champ protocole doit indiquer 50 ou 51 pour ESP ou AH en lieux et place par exemple de 6 (TCP) ou 17 (UDP). C'est l'en tête (AH ou ESP) qui indiquera le protocole encapsulé qui était auparavant indiqué dans l'en-tête IP.

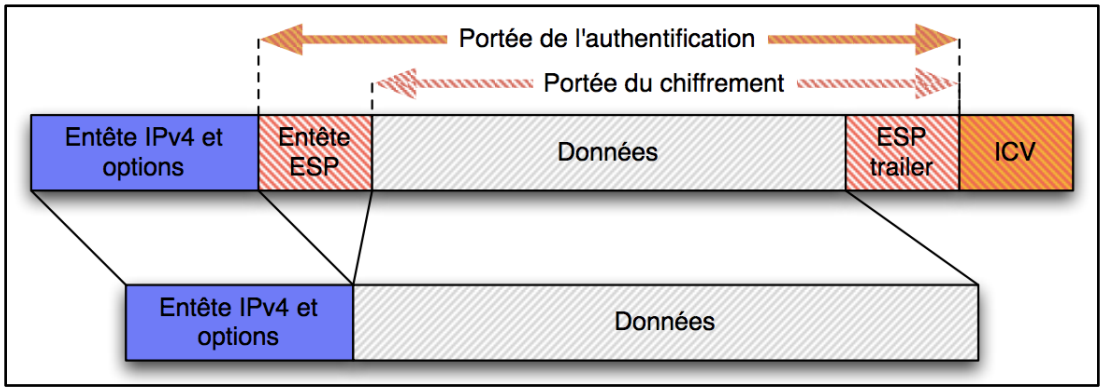


Figure 11. Utilisation d'ESP en mode transport.

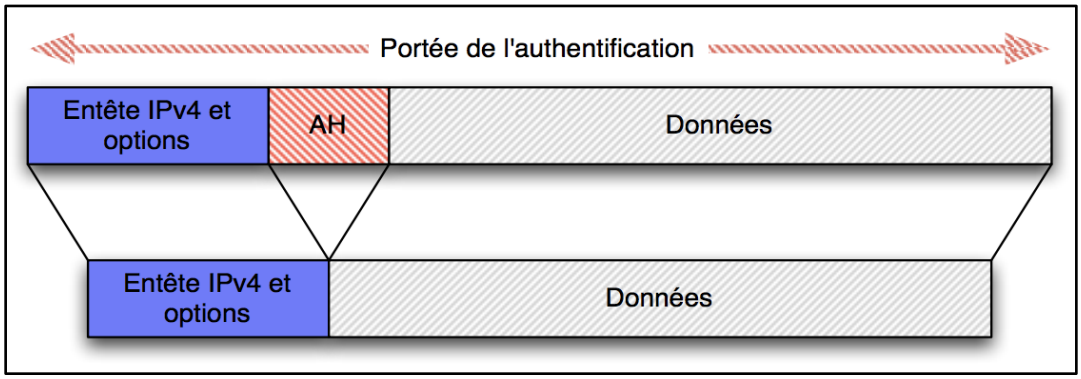


Figure 12. Utilisation d'AH en mode transport. [2]

Dans le mode tunnel en revanche, un nouveau paquet IP est généré pour contenir un paquet AH ou ESP qui contient lui-même le paquet IP initial sans modification. Dans ce mode, il y a donc en définitive deux en-têtes IP. L'en-tête externe sera effectivement utilisé pour le routage dès l'émission du paquet. L'en-tête interne, qui peut être chiffrée dans le cas où l'on utilise ESP avec le service de confidentialité, ne sera traitée que par le destinataire (du paquet externe).

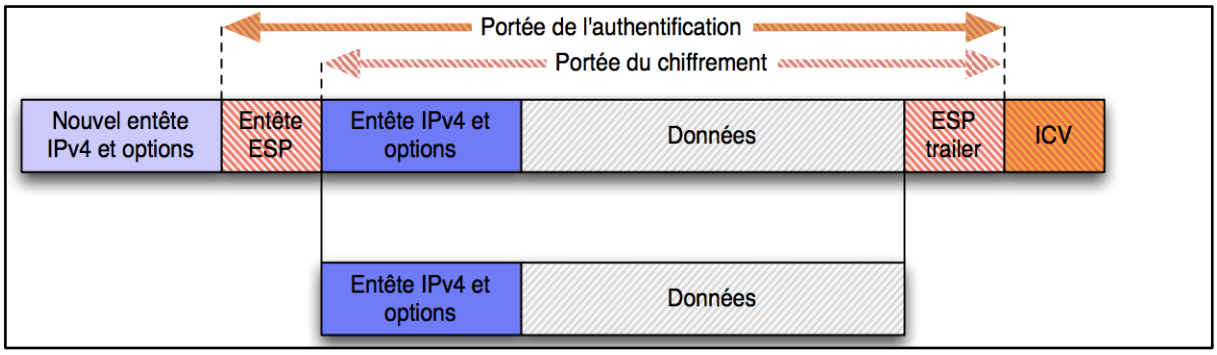


Figure 13. Utilisation d'ESP en mode tunnel. [2]

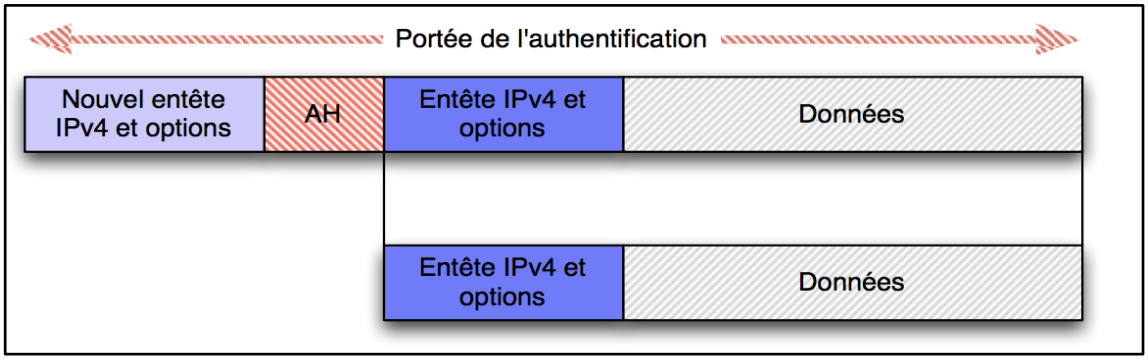


Figure 14. Utilisation d'AH en mode tunnel. [2]

2.3 MIPv6

Cette section comporte le bénéfice de la mobilité IPv6, les technologies supplémentaires à la mobilité IPv6, les types de noeuds, la procédure de la mobilité IPv6, "Dynamic Home Agent Discovery", "Mobile Prefix Sollicitation et l'IPsec.

2.3.1 Le bénéfice de la mobilité IP

La mobilité IP fournit la fonction de mobilité à tous les dispositifs IP. Cette fonction est une bonne solution parce que le protocole est situé dans la couche réseau, pourtant, il peut utiliser différentes technologies de la couche liaison de données. Elle peut être présente sur les réseaux cellulaires, les réseaux WiFi, les réseaux Ethernet, et autres réseaux futurs qui vont supporter IP. Quand une connexion rapide est disponible comme Ethernet, le noeud peut être attaché à un réseau câblé. Si le noeud a besoin de se déplacer aux autres réseaux, il peut être attaché à n'importe quel moyen qui support la couche IP.

Les applications de la couche supérieure n'ont pas besoin de prêter attention à la pile Mobilité IP. La mobilité IP en présentant aux application une couche IP transparente, les applications peuvent être utilisées sans aucun changement.

2.3.2 Les technologies supplémentaires à la mobilité IPv6

Il a été proposé que les concepteurs du protocole doivent faire marcher leurs protocoles indépendamment de la versione IP, ce qui veut dire que le nouveau protocole doit considérer IPv4 et aussi IPv6.
Cependant, ils faut pas ignorer l'environnement existant, IPv4. Il y a quelques années, les gens discutaient comment nous pouvons transiter de l'IPv4 vers l'IPv6 pour une longue durée. Nous pourrions même utiliser les technologies NAT en plus de l'IPv6 comme une solution intermédiaire pour le futur Internet. IPv4 ne peut pas être remplacé par d'autres technologies parce qu'il a été largement répandu dans le monde entier, pas seulement pour les chercheures mais aussi pour les réseaux industriels, infrastructures économiques, infrastructures de communications et réseaux de divertissement.

2.3.2.1 Double pile mobilité IPv6

DSMIPv6 est un protocole actuellement envisagé à l'IETF qui fournit une capabilité IPv4 au protocole mobilité IPv6.

La procédure DSMIPv6 élargit le mécanisme basique de la mobilité IPv6. Il est utilisé pour deux cas différents que le protocole de mobilité IPv6 ne supporte pas: le cas d'un réseau IPv4 étranger et le cas de l'adresse mère de l'IPv4.

Quand un terminal mobile (MN: Mobile Node) se déplace à un réseau étranger (FN:Foreign Network) et le réseau étranger supporte seulement l'IPv4, le MN ne peut pas conserver la connectivité tant qu'il utilise la mobilité IPv6 parce que la mobilité IPv6 suppose que tous les réseaux étrangers supportent l'IPv6. En DSMIPv6, le MN obtient l'adresse IPv4 du réseau étranger et l'utilise comme une adresse temporaire (CoA: Care-of Address). Rappelez le fait que la mobilité IPv6 est une sorte de protocole de mise en tunnel. Dans le cas de l'adresse CoA IPv4, le MN envoie un message de registration à son agent mère (HA: Home Agent) pour associer cette adresse à l'adresse mère (HoA: Home Address) du MN. Une fois que ce message est envoyé et accepté, un tunnel IPv6-over-IPv4 est créée entre le MN et le HA. Tous les paquets IPv6 envoyés depuis le MN sont encapsulés et transférés au HA en utilisant le tunnel IPv6-over-IPv4. Le HA désencapsule les paquets et les envoie à la destination finale. Le fait que les paquets désencapsulés sont des paquets IPv6, les noeuds destinataires ne savent pas que le MN est dans un réseau IPv4. La figure 11 illustre ce mécanisme.

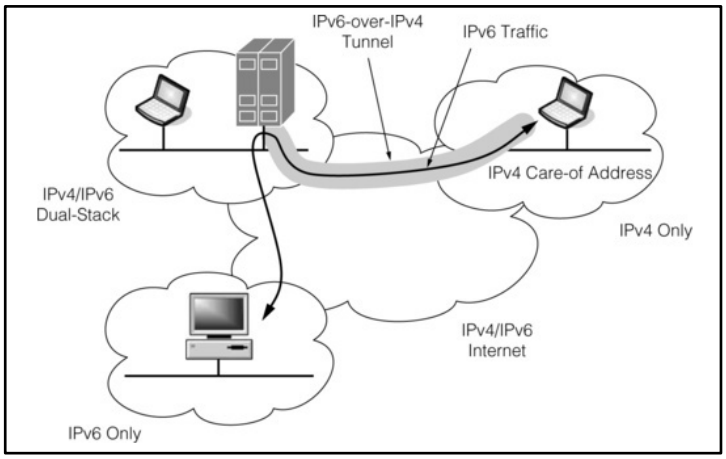


Figure 15. Protocole DSMIPv6. [3]

2.3.2.2 Support basique pour la mobilité des réseaux

L'une des raisons pour lesquelles c'est difficile de déployer la mobilité IPv6 c'est que le MN doit supporter la nouvelle pile du protocole pour utiliser la fonction de mobilité. Le support basique pour la mobilité des réseaux (NEMO BS: Network Mobility Basic Support) fournit l'une des solutions au problème en fournissant un routeur mobile.

Deux types d'entités en mouvement. La première est une simple entité en mouvement comme un téléphone mobile. La deuxième est un groupe des entités en mouvement qui se déplacent ensemble. Un exemple du dernier type est un système de transport comme les trains et les bus. Dans ce cas, beaucoup de personnes se déplacent ensemble comme une entité unique en mouvement. Si tous les personnes veulent utiliser le service de la mobilité IP avec la mobilité IPv6, tous les équipements qui utilisent doivent être mis à jour. Il est évident qu'il y a de l'optimisation

dans ce cas. Si tous les personnes se déplacent ensemble, pourquoi ont-ils besoin de gérer individuellement leurs équipements?

NEMO BS introduit une notion d'un réseau qui se déplace et ajoute une fonction de routage à la mobilité IPv6. Avec NEMO BS, un routeur mobile possède un préfixe de réseau fixe appelé Mobile Network Prefix (MNP).

2.3.3 Types de noeuds

La spécification de la mobilité IPv6 a défini 3 types de noeuds:

1. Le premier type est le MN, qui a la capacité de se déplacer autour des réseaux IPv6 sans rupture de connexion pendant le déplacement.
2. Le deuxième type est le HA, qui va servir de support dans le HN pour les MN. Un HA est un routeur qui a la fonction de proxy pour les MN quand ils sont en dehors du HN.
3. Le troisième type c'est le noeud correspondant (CN: Correspondent Node). Le CN est un noeud IPv6 qui communique avec le MN. Un CN ne doit pas forcément supporter la mobilité IPv6; n'importe quel noeud IPv6 peut être un CN.

2.3.4 La procédure de la mobilité IPv6

Dans ce chapitre nous allons discuter la procédure détaillée de la mobilité IPv6.

2.3.4.1 Home registration

Quand un MN est situé dans le HN, le noeud joue le rôle d'un noeud IPv6 fixe. La figure 12 illustre la situation.

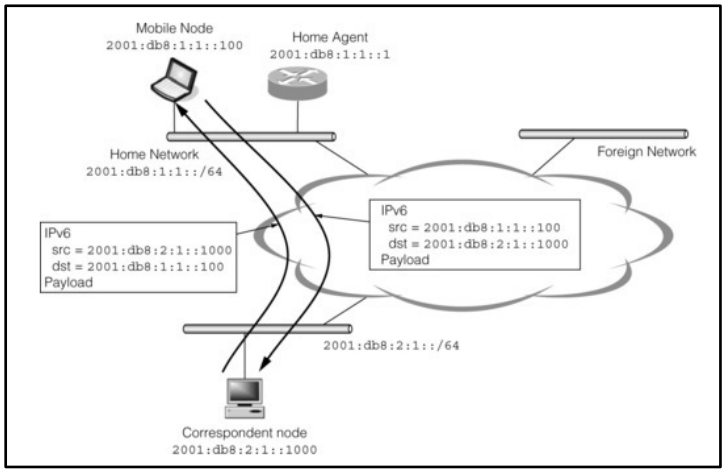


Figure 16. MN dans le réseau HN. [3]

Un noeud mobile obtient son adresse IPv6 de son HN. L'adresse attribuée dans le HN est appelé HoA. Quand un MN envoie un paquet, l'adresse source du paquet est attribué à une des adresses HoA du MN. L'adresse de destination du paquet est attribuée à l'adresse de l'autre dispositif avec lequel le MN est entrain de se communiquer. Quand ce dispositif envoie un paquet à MN, l'adresse source et destination sont attribuées à l'adresse du dispositif et l'adresse HoA du MN respectivement.

Quand le MN se déplace à un FN, le MN va obtenir l'adresse du FN. L'adresse ou les adresses que le MN va obtenir sont appelées CoAs. Si le MN détecte qu'il est dans un FN, le noeud va créer une entrée pour garder son statut et la maintenir. Cette entrée est appelé "Binding Update List entry". Elle va contenir les informations concernant le HoA et les CoAs du MN, la durée de vie de l'entrée, etc.

Le MN va envoyer un message Binding Update (BU) à son HA pour lui notifier de sa position actuelle. L'adresse source du message est attribuée au CoA qui est prise de la liste des adresses CoA. L'adresse de destination est attribuée à l'adresse du HA. Le message inclut aussi un champ appelé "Home Address option" qui contient l'adresse HoA du MN. Ce message doit être protégé par le mécanisme IPsec ESP.

Quand un HA reçoit un message BU, il ajoute l'information reçue à sa base de donées interne. Les informations gardées dans un HA sont appelées "Binding Cache". Le HA répond avec un message Binding Acknowledgment (BA) au message BU. Si le MN ne reçoit pas le message BA, il va renvoyer les message BU jusqu'à obtenir le message BA. Cette procédure est appelée "Home Registration". La figure 13 illustre cette procédure.

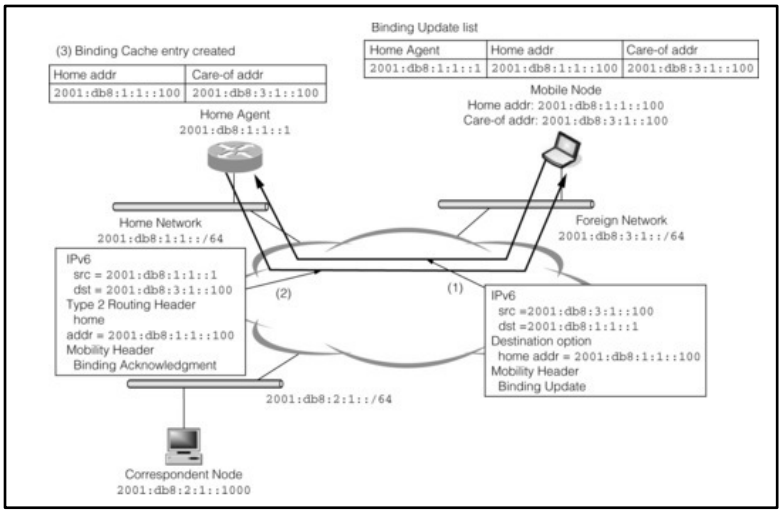


Figure 17. MN dans un réseau FN. [3]

Un message BU inclut aussi numéro de séquence. Si un HA a déjà une "Binding Cache" correspondante et le numéro de séquence du message BU reçu est plus petit que celui enregistré sur l'entrée de la cache, le HA va retourner un message BA avec un code d'erreur 135 et le dernier numéro de séquence. Le MN va alors envoyer le message BU avec le numéro de séquence correct pour compléter la procédure "Home Registration".

Quand le MN attribue une valeur au drapeau A, le noeud va renvoyer le message BU jusqu'à recevoir le message BA. La première fois que le MN va enregistrer sa localisation chez le HA, le HA doit s'assurer que l'adresse HoA n'est pas utilisée dans le HN par un autre noeud en effectuant la procédure DAD. Généralement, la procédure DAD prend qu'un seconde. Si le MN ne reçoit pas le message BA après avoir envoyé la dernière retransmission, le MN va effectuer le protocole "Dynamic Home Agent Address Discovery" pour trouver un autre HA localisé dans le HN.

Un message BU inclut un champ appelé "Alternate Care-of Address option" pour protéger les informations concernant l'adresse CoA du MN. Un MN a besoin de mettre l'adresse CoA dans le champ "Alternate Care-of Address option" pour que le paquet soit protégé par l'en-tête ESP.

Le champ “Lifetime” du message BU est attribué à la durée de vie la plus petite des adresses CoA et HoA du MN. Si le HA accepte la durée de vie demandée, le message BA doit inclure la même valeur.

Un MN maintient sa liste d’entrée “binding update” pour le “Home Registration” en envoyant un message BU périodiquement.

2.3.4.2 Bidirectional Tunneling

Quand un MN et un HA s’échangent les messages de type “Binding”, ces noeuds vont créer un tunnel entre eux. Les adresses du tunnel vont être attribuées à l’adresse du HA et l’adresse CoA du MN. Ce tunnel est utilisé pour cacher la localisation du MN aux CNs. Le CN ne va pas savoir si le MN est dans le HN ou dans un FN. Notez que les paquets envoyés à l’adresse “link-local” du MN ne vont pas être transmis au MN même si le drapeau F est activé dans le message BU du MN. Le drapeau est utilisé pour protéger l’adresse link-local pour qu’elle soit utilisée seulement pour les noeuds qui se trouvent dans le sous-réseaux et pas ailleurs.

Un MN généralement utilise son adresse HoA comme un adresse logique pour envoyer les paquets. Cela va assurer que la communication entre le MN et les autres noeuds survie quand le MN se déplace à un autre réseau. Cependant, un MN ne peut pas envoyer un paquet où son adresse source est attribuée à l’adresse CoA du MN. Ce paquet sera alors incorrect topologiquement et le routeur qui est dans le FN peut le jeter. La figure 14 illustre cette procédure.

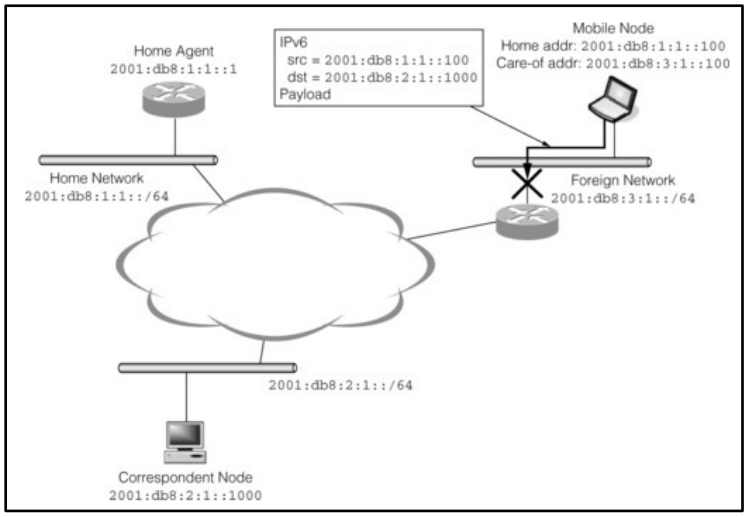


Figure 18. Le routeur FN refuse de transmettre le paquet. [3]

Pour éviter ce problème, un MN envoie des paquets où leur adresse source est attribuée à l’adresse HoA du MN en utilisant un tunnel créé entre le MN et son HA. La figure 15 illustre cette procédure.

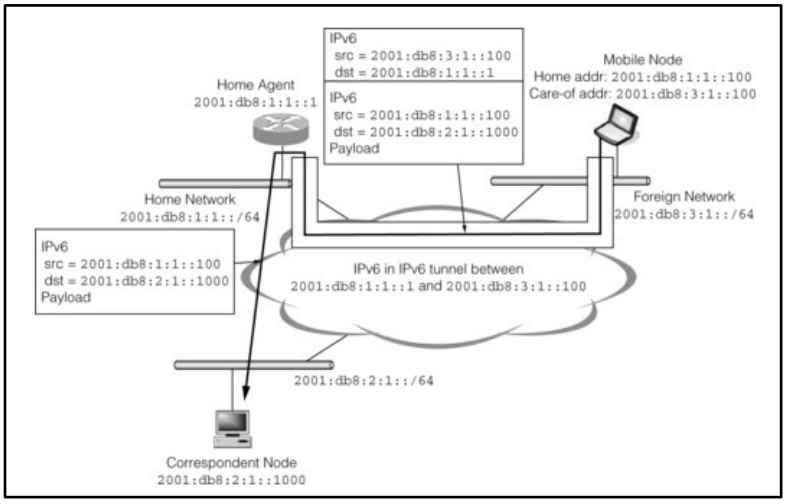


Figure 19. Tunnel entre le HA et MN. [3]

Un paquet est encapsulé dans une autre en-tête IPv6 dont l'adresse source et destination sont les adresses CoA du MN et son HA respectivement. Le paquet est désencapsulé dans le HA, et celui-la va transmettre le paquet à la destination finale. Le paquet a l'impression d'être envoyé par un noeud qui est rattaché au HN.

Quand un CN envoie des paquets au MN, le tunnel est utilisé aussi. Tous les paquets dont l'adresse destination est le HoA du MN sont envoyés au HN du MN. Ces paquets sont interceptés par le HA du MN ,si le HA possède une "Binding Cache Entry" valide pour le MN, et envoyés au MN en utilisant la tunnelisation IPv6-in-IPv6. L'adresse source et destination de l'en-tête IPv6 sont les adresses du HA et CoA du MN respectivement. La figure 16 illustre la procédure.

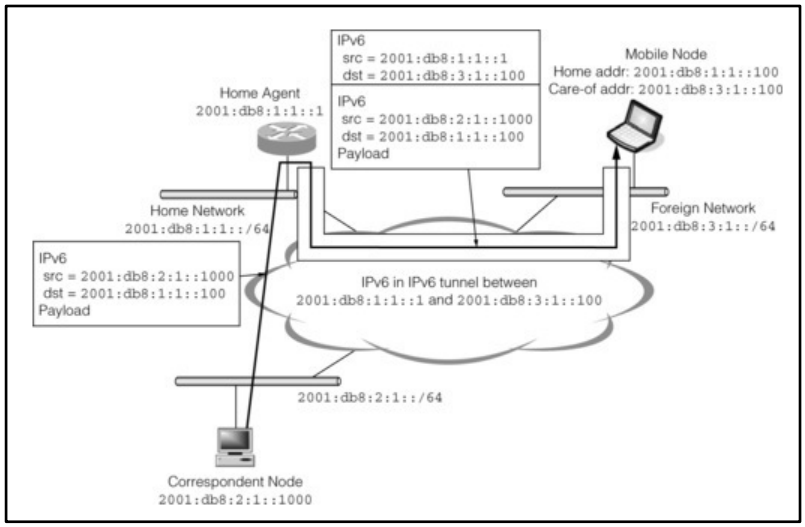


Figure 20. Communication entre un CN et un MN. [3]

2.3.4.3 Interception des paquets pour le Mobile Node

Un HA doit intercepter les paquets envoyés à un MN et transmettre ces paquets en utilisant le tunnel entre le HA et le MN.

Pour recevoir ces paquets qui sont envoyés au MN, un HA utilise le mécanisme "Proxy Neighbor Discovery". Quand un HA crée une entrée "Binding Cache" après avoir reçu le message BU d'un MN, le HA commence à répondre aux messages NS envoyés à l'adresse HoA ou à l'adresse

multicast à noeud sollicité de l'adresse HoA. Le HA répond avec un message NA en réponse aux messages de type NS. Dans les messages NA, le HA inclut son adresse link-local. Par conséquent, tous les paquets envoyés à l'adresse HoA du MN sont envoyé à l'adresse link-local du HA. Le HA transmet les paquets reçus au tunnel établi entre le HA et le MN comme nous avons déjà expliqué précédemment. La figure 17 illustre le comportement de ce mécanisme.

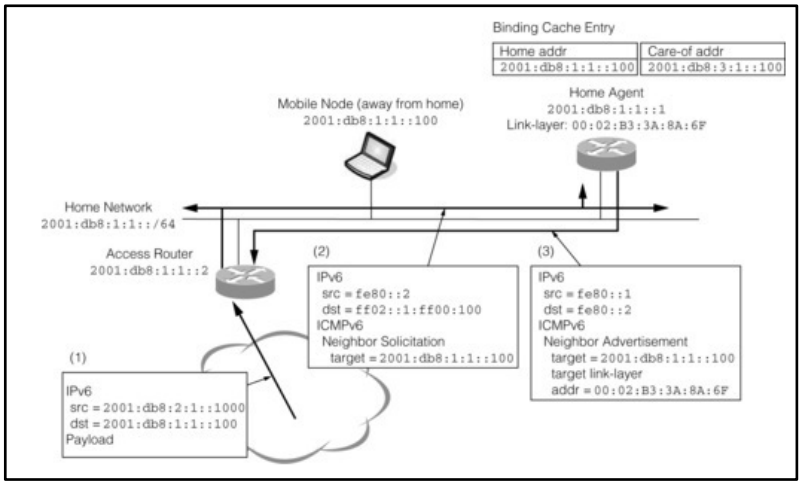


Figure 21. Mécanisme "Proxy Neighbor Discovery". [3]

2.3.4.4 Returning Home

Quand un MN retourne à son HN, il doit effacer les informations concernant les messages "Binding" déjà enregistrés avec le HA et le CN. La procédure pour désenregistrer ces informations est presque pareille à la procédure utilisée pour faire l'enregistrement. Le message utilisé pour cela est un message BU.

Tout d'abord, un MN doit envoyer un message BU à son HA. L'adresse source de ce message doit être l'adresse CoA du MN. Toutefois, dans ce cas, l'adresse source est attribuée à l'adresse HoA du MN puisque les adresses CoA et HoA sont identiques quand le MN est au HN. Ce message contient aussi le champ "Home Address option" qui contient l'adresse HoA. Le champ "Lifetime" est attribué à la valeur 0 pour indiquer le désenregistrement. Également, le message contient un champ "Alternate Care-of Address option" pour garder l'adresse CoA (qui est la même que son HoA). Ce message doit être protégé par le mécanisme IPsec ESP.

Dans certains cas, le MN ne peut pas savoir l'adresse link-layer de son HA, qui est nécessaire pour envoyer un paquet au HA. Dans ce cas, le MN doit effectuer la procédure "Neighbor Discovery", mais nous devons prendre en compte une chose: Si le HA possède une entrée valide "Binding Cache" pour l'adresse link-local du MN, le MN ne peut pas utiliser son adresse "link-local" pendant la procédure "Neighbor Discovery" parce que le HA agit à titre d'un serveur proxy de l'adresse. Cet usage peut être considéré une duplication d'adresse. Quand un MN a besoin de résoudre l'adresse "link-local" de son HA quand il retourne à son HN, il envoie un message "Neighbor Solicitation" d'une adresse non spécifiée. Quand le HA reçoit ce message, il répond avec un message "Neighbor Advertisement" à une adresse multicast (tous les noeud) comme est déjà expliqué sur le *RFC2461*[4]. Un mobile node peut apprendre l'adresse "link-layer" du HA en analysant le message "Neighbor Advertisement".

[4] https://www.ietf.org/rfc/rfc2461.txt

Si un HA accepte un message BU, il répond avec un message BA. Un HA arrête sa fonction proxy pour le MN et éteint le tunnel entre le HA et le MN. Finalement, il enlève l'entrée "Binding Cache" pour ce MN.

Un MN aussi éteint le tunnel entre lui et le HA après avoir reçu le message BA de son HA. Cette procédure est appelée "Home Deregistration".

Il y a une possibilité que les messages de signalisation peuvent être jetés à cause des erreurs de communication. Si un message BU envoyé par le MN pour la procédure "Home Deregistration" est perdu, le MN va renvoyer un autre message BU jusqu'à recevoir le message BA. Si le message BA est perdu aussi, la situation va être un peu compliquée parce que l'entrée "Binding Cache" du MN qui envoie le message pour se désenregistrer est supprimée du HA quand le message BA a été envoyé. Le MN va renvoyer le message BU parce qu'il n'a pas encore reçu le message BA correspondant. Quand le HA reçoit un message BU pour la procédure "Home Deregistration" mais il ne possède pas l'entrée "Binding Cache" correspondante, il va répondre au MN avec un message BA avec le code 133. Quand le MN qui est retourné à son HN reçoit le message BA avec le code 133, le MN doit considérer que le message BA a été perdu et il complète la procédure "Home Deregistration".

Un MN peut désenregistrer son adresse de son HA même s'il ne retourne pas à son HN (par exemple, quand le MN arrête sa fonction de mobilité dans un FN). Dans ce cas, une procédure similaire est utilisée pour désenregistrer l'adresse. Le message BU envoyé du MN va avoir différentes adresses CoA et HoA mais le champ "Lifetime" est attribué à 0. Le HA va supprimer son entrée "Binding Cache" et arrête d'intercepter les paquets pour ce MN.

2.3.4.5 Route optimization

Quand le MN se communique avec les autres noeuds, tous les paquets sont transmis par le HA si le MN est loin de son HN. Cela cause des délais de communication, spécialement si le MN et son CN sont localisés dans des réseaux très proches et le HA est très loin de ces deux réseaux. Le pire cas c'est quand le MN et le CN sont localisés dans le même réseau.

La spécification de la mobilité IPv6 fournit une solution à ce problème. Si le CN support la mobilité IPv6, le chemin entre le MN et le CN peut être optimisé. Pour optimiser ce chemin, le MN doit envoyer un message BU au CN. Le message ne doit pas avoir les drapeaux H et L activés parce que ce message ne demande pas une procédure "Home Registration". Par contre, le drapeau 1 doit être activé; mais ce n'est pas obligatoire. Si le drapeau A est activé, le CN répond avec un message BA. Notez que même si le drapeau A n'est pas activé, le CN doit répondre avec un message BA quand une erreure arrive pendant la procédure, à l'exception d'une erreur d'authentification.

Un message BU doit être protégé par la procédure "Return Routability". Le message doit contenir un champ "Binding Authorisation Data option". Ce champ contient la valeur hachage du message BU, cette valeur est calculée avec une clef partagée générée suite à la procédure "Return Routability". Si la valeur du hash est incorrecte, le message est jeté. De la même manière, un message BA envoyé par un CN doit contenir le champ "Binding Authorisation Data option" pour protéger le contenu.

Une fois l'échange du message BU (et BA, si le drapeau A est activé) est complété, le MN commence à échanger des messages du trajet optimisé (Route-Optimized Packets) avec le CN. L'adresse source des paquets est attribué à l'adresse CoA du MN. Le MN ne peut pas attribuer son adresse HoA directement à l'adresse source de son paquet, parce que les routeurs intermédiaires vont

jeter ces paquets pour éviter les attaques spoofing. L'information de l'adresse HoA est gardé dans l'option "Home Address option" dans le champ "Destination Options" de l'en-tête du paquet.

Quand le CN reçoit un paquet qui a l'option "Home Address option", il va vérifier s'il possède une entrée "Binding Cache" reliée au HoA du MN. S'il n'y a aucune entrée, le CN répond avec un message "Binding Error (BE)" avec le code 0. Le MN doit renvoyer le message BU pour créer l'entrée "Binding Cache" dans le CN s'il reçoit le message BE. Cette procédure de validation évite les noeuds malicieux d'utiliser une adresse CoA qui n'est pas la sienne.

Quand le CN envoie un paquet au MN, il utilise l'en-tête "Type 2 Routing". Le HoA du MN est attribué à l'en-tête Routing et l'adresse destination du paquet est attribué à l'adresse CoA du MN. Le paquet ne va pas arriver au HN du MN. Au lieu de cela, le paquet est routé au FN où se trouve le MN.

2.3.5 Dynamic Home Agent Address Discovery

Un MN peut ne pas savoir l'adresse de son HA quand il essaie d'envoyer un message BU pour la procédure "Home Registration". Par exemple, si le MN redémarre dans un réseau FN, il n'y a pas des informations concernant le HA sauf si ces information sont préconfigurées.

Le mécanisme "Dynamic Home Agent Address Discovery" est utilisé pour avoir les informations concernant l'adresse du HA quand le MN est localisé dans un FN. Un MN envoie une requête "Dynamic Home Agent Address Discovery" quand il a besoin de savoir l'adresse de son HA. L'adresse source du message est attribuée à l'adresse CoA du MN et l'adresse destination du message est attribuée à une adresse anycast du HA, cette adresse peut être calculé à partir du "Home Prefix". Ce message ne doit pas contenir le champ "Home Address option", puisque ce message doit être envoyé avant que la première procédure "Home Registration" est complété.

Dans le HN, les HAs maintiennent une liste des adresses globales de tous les HAs qui se trouvent dans le HN en écoutant à tous les messages "Router Advertisement".

Chaque HA possède une adresse anycast spéciale, appelée "Home Agent Anycast Address", qu'elle est calculée comme nous avons déjà expliqué. Une requête "Dynamic Home Agent Address Discovery" est distribuée à un des HAs dans un HN grâce au mécanisme anycast address. Le HA qui reçoit ce message va répondre au MN avec une réponse "Dynamic Home Agent Address Discovery" contenant toutes les adresses des HAs que le HA connaît. Cette liste est ordonnée par la valeur de préférence de chaque HA. S'il y a plusieurs HAs avec la même valeure de préférence, les adresses doivent être ordonnées aléatoirement avec équilibrage de charge. Pour éviter la fragmentation du paquet, la longueur totale du message doit être plus petite que le MTU du chemin vers le MN. Si la liste est très longue à inclure dans un même paquet, les HAs qui possèdent une valeur de préférence basse sont exclues de ce message.

Si un MN ne reçoit pas un message de réponse, le noeud va renvoyer le message de requête.

En principe, le mécanisme "Home Agent Address Discovery" peut être utilisé comme un mécanisme pour notifier les MNs des HAs disponibles dans le HN. Cependant, en ajoutant/supprimant les HAs peuvent causer des problèmes de configuration IPsec.

2.3.6 Mobile Prefix Solicitation/Advertisement

Une adresse IPv6 possède une durée de vie. Cette durée de vie est extraite de la durée de vie contenu dans le préfixe. Si l'adresse HoA du MN va prochainement expirer, le MN envoie un message "Mobile Prefix Solicitation" pour acquérir l'information récente à propos de "Home Prefixes". L'adresse source du message est attribuée à l'adresse CoA du MN. L'adresse destination du message est attribuée à l'adresse du HA avec lequel le MN est déjà enregistré. Le message doit inclure un champ "Home Address option", qui contient son adresse HoA). Puisque la procédure "Home Registration" exige les information concernant le HN, le mécanisme "Prefix Discovery" ne peut pas être utilisé pour trouver les "Home Prefixes" quand le MN est démarré dans un réseau FN, mais il peut être utilisé seulement pour connaître des nouveaux "Home Prefixes" ou des "Home Prefixes" obsolètes.

Quand un HA reçoit un message "Mobile Prefix Solicitation" d'un MN, le HA répond au MN avec un message "Mobile Prefix Advertisement". L'adresse source du message doit être l'adresse destination du message "Mobile Prefix Solicitation" correspondant. L'adresse destination du message est l'adresse CoA du MN. Une en-tête "Type 2 Routing" qui contient l'adresse HoA du MN doit exister. La liste de l'information modifiée concernant le préfixe suit au message "Advertisement".

Contrairement au message RA, la liste de "Prefix Information options" envoyée contient des HAs qui se trouvent dans le réseau HN doit être cohérente. Pour vérifier cette cohérence, chaque HA doit être configuré pour avoir la même information concernant le préfixe de son HN, ou il doit écouter les messages RA des autres HAs et construire une autre liste. Un MN envoie un message de sollicitation au HA avec lequel est enregistré. Si l'information concernant le préfixe retournée diffère des autres préfixes retournés des autres HAs, le MN peut considérer d'une manière incorrecte que quelques information concernant les préfixes sont disparus.

Le HA peut envoyer un message "Mobile Prefix Advertisement" même si le MN ne demande pas l'information concernant le préfixe dans les cas suivants:

- L'état des drapeaux du "Home Prefix" que le MN utilise est changé.
- La durée de vie d'un "Home Prefix" est reconfigurée.
- Un nouveau "Home Prefix" est ajouté.
- L'état des drapeaux ou la valeur de la durée de vie d'un "Home Prefix" qui n'est pas utilisé par un MN change.

2.3.7 IPsec

La mobilité IPv6 utilise le mécanisme IPsec pour protéger les messages de signalisation MIPv6. Les spécifications concernant comment protéger ces messages sont disponibles dans le *RFC3776*[5].

Les messages échangés directement entre le MN et le HA sont protégés par le mécanisme IPsec transport mode. Les messages BU et BA doivent être protégés par le protocole IPsec ESP ou AH. Les messages "Mobile Prefix Solicitation" et "Mobile Prefix Advertisement" doivent être protégés aussi par le mécanisme IPsec.

Les messages échangés entre le MN et le CN, et les messages reliés au HA, sont protégés par le mécanisme IPsec tunnel mode. Les messages "Home Test Init et Home Test" doivent être protégés par l'en-tête IPsec ESP avec le mode IPsec tunnel. Comme nous le verrons dans cette partie, les politiques de sécurité du mode tunnel doivent supporter l'en-tête de mobilité, c'est-à-dire, elle doit pouvoir envoyer et recevoir les messages "Home Test Init" et "Home Test" seulement à

[5] https://tools.ietf.org/html/rfc3776

travers le tunnel IPsec. Cela est nécessaire quand deux MNs communiquent entre eux avec la procédure “Route Optimisation”. Si un MN ne peut pas spécifier les messages “Home Test Init/Home Test” comme une spécification de politique, un message BU n’est pas bien “tunnelé” vers le HA du MN qui envoie le message BU.

Notez que les messages de requête et réponse de type “Dynamic Home Agent Address Discovery” ne peuvent pas être protégés parce que le MN ne connaît pas l’adresse de son HA avant d’échanger ces messages. L’adresse du HA est exigée pour configurer une base de données de politiques de sécurité pour protéger les messages.

Des associations de sécurité (SA) pour chaque politique peuvent être configurées par opération manuelle. Le mécanisme IKE peut être utilisé pour créer ces SAs dynamiquement, mais il exige une modification au programme IKE. Généralement, les adresses de SAs IKE sont dérivées des adresses qui sont utilisées pour exécuter la négociation IKE. Dans le cas de la mobilité IPv6, quand un MN se déplace de son HN à un FN, l’adresse HoA ne peut pas être utilisée jusqu’à compléter la procédure “Home Registration”. Cependant, nous avons besoin d’une SA entre l’adresse HoA et l’adresse du HA pour compléter la procédure “Home Registration”. Le programme IKE doit utiliser une adresse CoA pour la négociation IKE et créer des SAs pour les adresses qui ne sont pas utilisées dans la négociation IKE. Souvent, peu d’implémentations supportent cette fonction.

Il y a d’autres problèmes qui sont causés par le mécanisme de la configuration des politiques IPsec. La configuration des politiques IPsec est généralement statique, dans l’opération MIPv6 cependant, nous avons besoin de changer les politiques dans les situations suivantes:

- Quand un nouveau HA est installé, un MN doit installer des nouveaux politiques du mode tunnel et transport pour le HA.

2.3.7.1 MIPv6 avec IKEv2

Le *RFC3776* décrit comment l’IPsec est utilisé avec MIPv6 pour protéger les messages de signalisation.

2.3.7.1.1 Les formats des paquets

Le format du paquet pour le BU quand il est envoyé en mode tunnel se présente comme suit:

- En-tête IPv6 (source = CoA, destination = adresse HA)
- En-tête ESP en mode tunnel
- En-tête IPv6 (source = HoA, destination = adresse HA)
- En-tête Mobilité
  Binding Update
  Alternate Care-of Address option (CoA)

Le format du paquet pour le BA envoyé au MN quand il se trouve dehors le HN:

- En-tête IPv6 (source = adresse HA, destination = CoA)
- En-tête ESP en mode tunnel
- En-tête IPv6 (source = adresse HA, destination = HoA)
- En-tête Mobilité
  Binding Acknowledgment

2.3.7.1.2 Exigences

Cette partie décrit les règles obligatoires et les exigences pour tous les HAs et MNs MIPv6 pour que IPsec avec IKEv2 (comme le protocole de gestion de clés) peuvent être utilisés pour protéger le trafic entre le MN et le HA

2.3.7.1.2.1 Exigences de la politique

Le HA doit être capable d'éviter que le MN utilise ses SAs pour envoyer un BU au nom d'un autre MN. Avec la configuration IPsec manuelle, le HA doit vérifier que la SA a été créée pour une adresse en particulier. Avec IKEv2, le HA doit être capable de vérifier que l'identité présenté dans l'échange IKE_AUTH est permise pour créer des SAs pour une adresse particulière.

Le HA utilise la "Peer Authorization Database (PAD) pour stocker le statut de chaque noeud. Plus précisément, ces information de statut présentent des informations utilisées pour authentifier le MN et l'information d'authentification qui relie l'identité du MN à l'adresse HoA du MN. Cela va permettre au HA d'éviter qu'un MN crée des SAs pour un autre MN. Dans le cas de IKEv2, le HA va créer une entrée PAD temporaire en reliant l'identité du nouveau noeud authentifié et la nouvelle adresse HoA allouée.

Quand un paquet destiné à un noeud est comparé à des politiques de sécurité IPsec ou sélecteurs d'une SA, une adresse montrée dans le champ "Home Address destination option" est considérée comme l'adresse source du paquet.

Une adresse HoA dedans l'en-tête "Type 2 Routing" destinée à un noeud est considérée comme l'adresse destination du paquet.

Quand le MN retourne à son HN et se désenregistre de son HA, le tunnel entre le HA et le MN est supprimé. Les politiques de sécurité, qui ont été utilisées pour protéger le trafic "tunnélé" entre le MN et le HA doivent être rendues désactivées. Les SAs correspondantes peuvent être conservées ou supprimées, cela dépend de comment ont été créées. Si les SAs ont été créées dynamiquement avec IKEv2, elles seront alors supprimées automatiquement quand elles expirent. Si les SAs ont été créées manuellement, elles doivent être conservées pour être utilisées ultérieurement quand le MN se déplace encore une fois. Les SAs qui protègent les messages BU, BA et Mobile Prefix Discovery ne doivent pas être supprimées parce qu'elles dépendent pas de l'adresse CoA et pourtant elles peuvent être utilisées encore une fois.

Le MN doit utiliser le champ "Home Address Destination option" contenu dans les messages BU et Mobile Prefix Solicitations quand le mode transport de la protection IPsec est utilisé, pour que le HoA soit visible au moment de vérifier les politiques IPsec.

Le HA doit utiliser le champ "Type 2 Routing header" dans les messages BA et Mobile Prefix Advertisements envoyés au MN quand le mode transport de la protection IPsec est utilisé, pour que le HoA soit visible pendant les vérifications de la politique IPsec.

2.3.7.2 Exigences IKEv2

Le MN doit utiliser son adresse CoA comme l'adresse source dans les protocoles d'échanges, quand il utilise le protocole ISAKMP.

Le MN et le HA doivent créer les SAs basées sur l'adresse HoA, pour que ses SAs survivent aux changement dans les adresses CoAs. Quand le protocole IKEv2 est utilisé, l'adresse HoA doit être portée comme l'adresse IP initiatrice dans le payload TSi pendant l'échange CREATE_CHILD_SA.

2.3.7.3 Configuration dynamique

Cette partie décrit l'utilisation de l'IKEv2 pour installer les SAs exigées.

2.3.7.3.1 PAD

Les entrées PAD (Peer Authorization Database) présentes sur le MN et le HA sont les suivantes:

Mobile Node:

- if (remote_identity = home_agent_identity_1)

    Then authenticate (shared secret/certificate/)
    and authorize CHILD_SA for remote address home_agent_1

Home Agent:

- if (remote_identity = user_1)

    Then authenticate (shared secret/certificate/EAP)
    and authorize CHILD_SAs for remote address home_address_1

2.3.7.3.2 SPD

Les entrées SPD présentes sur le MN et le HA sont les suivantes:

2.3.7.3.2.1 BU et BA

Les entrées SPD présentes dans le MN et le HA pour protéger les messages BU et BA

Mobile Node:

- if (local_address = home_address_1 &
remote_address = home_agent_1 &
proto = MH & local_mh_type = BU & remote_mh_type = BAck)

    Then use SA ESP transport mode
    Initiate using IDi = user_1 to address home_agent_1

Home Agent:

- if (local_address = home_agent_1 &
remote_address = home_address_1 &
proto = MH & local_mh_type = BAck & remote_mh_type = BU)

    Then use SA ESP transport mode

2.3.7.3.2.2 Les messages "Return Routability"

Les entrées SPD présentes dans le MN et le HA pour protéger les messages "Return Routability"

Mobile Node:

- if (local_address = home_address_1 & remote_address = any &
proto = MH & local_mh_type = HoTi & remote_mh_type = HoT)

    Then use SA ESP tunnel mode
    Initiate using IDi = user_1 to address home_agent_1

Home Agent:

- if (local_address = any & remote_address = home_address_1 &
proto = MH & local_mh_type = HoT & remote_mh_type = HoTi)

    Then use SA ESP tunnel mode

Dans ces cas, au moment que l'adresse CoA change, les entrées de SPD sur les deux noeuds doivent être mises à jour.

*2.3.7.3.2.3 Les messages "*Mobile Prefix Discovery"

Les entrées SPD présentes dans le MN et le HA pour protéger les messages "Mobile Prefix Discovery"

Mobile Node:

- if (local_address = home_address_1 &
remote_address = home_agent_1 &
proto = ICMPv6 & local_icmp6_type = MPS &
remote_icmp6_type = MPA)

    Then use SA ESP transport mode
    Initiate using IDi = user_1 to address home_agent_1

Home Agent:

- if (local_address = home_agent_1 &
remote_address = home_address_1 &
proto = ICMPv6 & local_icmp6_type = MPA &
remote_icmp6_type = MPS)

    Then use SA ESP transport mode

2.3.7.3.2.4 Les paquets Payload

Les entrées SPD présentes dans le MN et le HA pour protéger le trafic entre le CN et le MN

Mobile Node:

- if (interface = IPv6 tunnel to home_agent_1 &
source = home_address_1 & destination = any & proto = X)

Then use SA ESP tunnel mode
Initiate using IDi = user_1 to address home_agent_1

Home Agent:

- if (interface = IPv6 tunnel to home_address_1 &
source = any & destination = home_address_1 & proto = X)

Then use SA ESP tunnel mode

2.3.7.4 Négociation des SAs en utilisant IKEv2

Les messages de signalisation MIPv6 sont généralement déclenchés par le MN. Le MN envoie un message BU au HA quand il se déplace dehors le HN et obtient une nouvelle CoA.

Le MN initie le protocole d'échange IKEv2 si les SAs ne sont pas présentes. Un mécanisme possible pour l'authentification mutuelle est la clée partagée entre le MN et le HA. Le HA utilise l'identité du MN pour identifier la clée partagée correspondante.

Si une clée partagé est utilisée, le MN utilise cette clée pour générer le "AUTH Payload" dans l'échange "IKE_AUTH". Si le MN utilise les certificats, il utilise sa clée privée pour générer le "AUTH Payload" dans l'échange "IKE_AUTH".

Le MN inclut toujours son identité dans le "Payload IDi" présent dans l'échange "IKE_AUTH". Le MN peut utiliser ces différents types d'identités pour s'identifier auprès du HA:

- Le HoA: Le MN peut utiliser son adresse HoA pour son identité. Dans ce cas, le champ ID Type est attribué au ID_IPV6_ADDR.
- FQDN: Le nom de domaine qui donne la position exacte de son noeud dans l'arborescence DNS en indiquant tous les domaines de niveau supérieur.

Dans l'échange "IKE_AUTH", le MN inclut le HoA et les sélecteurs appropriés dans le payload TSi (Traffic Selector-initiator) pour négocier les SAs à fin de protéger les messages BU et BA.

Après l'exécution de l'échange "IKE_AUTH", le MN initie l'échange "CREATE_CHILD_SA" pour négocier des SAs additionnelles pour protéger la procédure "Return Routability", les messages "Mobile Prefix Discovery" et le "payload". Les échanges "CREATE_CHILD_SA" sont protégés par les SAs IKEv2 créées pendant les échanges "IKE_SA_INIT".

C'est important que les SAs soient créées en se basant sur l'adresse HoA du MN pour que les SAs survivent aux changements d'adresses.

Quand une authentification basée sur une PKI est utilisée entre le MN et le HA, l'identité présentée par le MN dans le "Payload IDi" doit correspondre à l'identité que le HA obtient du certificat. Le

HA utilise l'identité présentée dans le payload IDi pour vérifier qu'elle correspond à celle du certificat. Si le MN présent son adresse HoA dans le "Payload IDi", le HA doit vérifier que ce HoA correspond à celle du champ IPAdress dans l'extension SubjectAltName.

Dans ce chapitre nous avons expliqué les bases d'IPv6 et MIPv6 avant d'expliquer comment nous avons fait l'implémentation. Dans le chapitre 3, nous allons expliquer comment nous avons fait l'implémentation des différents scénarios de MIPv6.

3 Implémentation

Dans ce chapitre, nous allons implémenter trois scénarios de MIPv6 différents: La mobilité IPv6 non sécurisé, la mobilité IPv6 sécurisée manuellement et la mobilité IPv6 sécurisée avec IKEv2.

Ce chapitre est composé de 5 sections: L'installation des logicielles nécessaires pour l'implémentation de la mobilité IPv6, scénario de la mobilité IPv6 et les trois sections détaillées de l'implémentation des trois scénarios de la mobilité IPv6.

3.1 Installation

Dans cette partie nous allons expliquer comment nous avons fait l'installation des logiciels nécessaires pour la mise en oeuvre d'un scénario de mobilité MIPv6.

3.1.1 VirtualBox

La première chose est de choisir sur quelle machine installer tous les logiciels nécessaires. Nous avons choisi d'utiliser 4 machines virtuelles pour simuler le "Home Agent", le "Mobile Node", le "Correspondent Node" et le "Foreign Router". L'image à installer sur les machines virtuelles doit être forcément une image "debian" pour supporter tous les logiciels que nous allons mentionner dans ce chapitre.

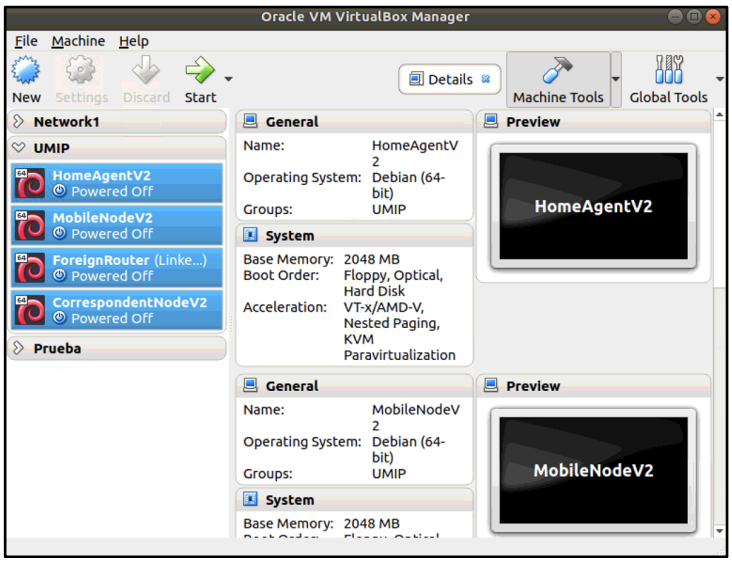


Figure 22. Machines virtuelles à utiliser

3.1.2 Kernel

Le Kernel est le coeur de votre distribution. Il assure la communication entre les logiciels et le matériel, la gestion des tâches d'une machine et la gestion du matériel. Notez que les versions les plus anciennes du Kernel ne supportent pas la fonction de mobilité, par contre il y a eu beaucoup de patchs qui permettaient la simulation de cette fonctionnalité malgré l'ancienneté du Kernel.

Pourtant, nous allons installer une version récente qui supporte la mobilité IPv6 et qui nous permet de simuler le scénario de notre projet.

1. Tout d'abord nous allons sur la page *www.kernel.org* pour télécharger les sources officielles de linux qui nous intéresse. Nous avons opté pour le choix de la version "linux-4.10.1" dans notre travail. L'installation est recommandée d'être faite dans */usr/src* parce que c'est ici où se trouvent les sources par défaut:

   ```
   user1:~/ cd /usr/src
   ```

2. Télécharger les sources:

   ```
   user1:~/usr/src/ wget http://www.kernel.org/pub/linux/kernel/v4.x/linux-4.10.1.tar.gz
   ```

3. Extraire l'archive:

   ```
   user1:~/usr/src/ tar xf linux-4.10.1.tar.gz
   ```

4. Faire un lien symbolique intitulé *linux* vers le noyau:

   ```
   user1:~/usr/src/ ln -s /usr/src/linux-4.10.1/ /usr/src/linux
   ```

5. Accéder à l'archive extrait précédemment:

   ```
   user1:~/usr/src/ cd linux-4.10.1/
   ```

6. Installer quelques paquets avant la compilation:

   ```
   user1:~/usr/src/linux-4.10.1/ sudo apt-get install automake autoconf make bison flex libssl1.0-dev indent quilt build-essential ipsec-tools radvd
   ```

7. Si vous n'avez pas de configuration à la racine des sources (*./configure*), il faut en récupérer une de préférence compatible avec le kernel courant. Le rôle de la commande "oldconfig" va chercher cette configuration dans */boot/config.$(uname -r*:

   ```
   user1:~/usr/src/linux-4.10.1/ sudo make oldconfig
   ```

8. Il faudra aussi installer ces paquets:

   ```
   user1:~/usr/src/linux-4.10.1/ sudo apt-get install libncurses5-dev libncursesw5-dev
   ```

9. Maintenant nous allons sélectionner toutes les options qui vont supporter la mobilité IPv6 et la sécurité IPsec (Figure 19):

   ```
   user1:~/usr/src/linux-4.10.1/ sudo make menuconfig
   ```

```
 General setup
 --> Prompt for development and/or incomplete code/drivers
[CONFIG_EXPERIMENTAL]
 --> System V IPC [CONFIG_SYSVIPC]

 Networking support [CONFIG_NET]
 --> Networking options
     --> Transformation user configuration interface
[CONFIG_XFRM_USER]
     --> Transformation sub policy support [CONFIG_XFRM_SUB_POLICY]
     --> Transformation migrate database [CONFIG_XFRM_MIGRATE]
     --> PF_KEY sockets [CONFIG_NET_KEY]
     --> PF_KEY MIGRATE [CONFIG_NET_KEY_MIGRATE]
     --> TCP/IP networking [CONFIG_INET]
     --> The IPv6 protocol [CONFIG_IPV6]
         --> IPv6: AH transformation [CONFIG_INET6_AH]
         --> IPv6: ESP transformation [CONFIG_INET6_ESP]
         --> IPv6: IPComp transformation [CONFIG_INET6_IPCOMP]
         --> IPv6: Mobility [CONFIG_IPV6_MIP6]
         --> IPv6: IPsec transport mode
[CONFIG_INET6_XFRM_MODE_TRANSPORT]
         --> IPv6: IPsec tunnel mode [CONFIG_INET6_XFRM_MODE_TUNNEL]
         --> IPv6: MIPv6 route optimization mode
[CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION]
         --> IPv6: IP-in-IPv6 tunnel (RFC2473) [CONFIG_IPV6_TUNNEL]
         --> IPv6: Multiple Routing Tables
[CONFIG_IPV6_MULTIPLE_TABLES]
         --> IPv6: source address based routing [CONFIG_IPV6_SUBTREES]

 File systems
 --> Pseudo filesystems
     --> /proc file system support [CONFIG_PROC_FS]
```

Figure 23. Liste des options à activer sur l'interface graphique "menuconfig".

10. Compiler le kernel:

```
user1:~/usr/src/linux-4.10.1/ sudo make
```

11. Installer les paquets suivants:

```
user1:~/usr/src/linux-4.10.1/ sudo apt-get install libelf-dev
```

12. Installer les modules sélectionnés à l'étape 9:

```
user1:~/usr/src/linux-4.10.1/ sudo make modules_install
```

13. Installer le kernel:

```
user1:~/usr/src/linux-4.10.1/ sudo make install
```

14. Installer les headers:

```
user1:~/usr/src/linux-4.10.1/ sudo make headers_install
```

15. Une fois que nous avons fini, nous devons redémarrer l'ordinateur

```
user1:~/usr/src/linux-4.10.1/ sudo reboot
```

3.1.3 UMIP

Maintenant que nous avons un kernel qui supporte la mobilité IPv6 (MIPv6) nous pouvons installer le software UMIP (Universal Mobile IP for Linux). UMIP est un software open source qui implémente les protocoles Mobile IPv6 et NEMO Basic Support pour le système d'exploitation Linux. Pour l'installation de ce logiciel, il faut suivre les indications suivantes:

1. Avant nous devons nous placer dans le répertoire */usr/src*:

   ```
   user1:~/ cd /usr/src
   ```

2. Télécharger les sources officielles du logiciel de github:

   ```
   user1:~/usr/src/ git clone git://git.umip.org/umip/umip.git
   ```

3. Aller au dossier "*umip*":

   ```
   user1:~/usr/src/ cd umip/
   ```

4. Installer et exécuter si besoin les différents outils GNU pour la construction d'applications (autoconf, autoheader, aclocal, automake, libtoolize et autopoint) à chaque niveau de l'arborescence des sources:

   ```
   user1:~/usr/src/umip/ autoreconf -i
   ```

5. Indiquer les en-têtes du noyau pour paramétrer la compilation du software et activer la possibilité d'avoir un terminal virtuel pour accéder aux informations du "Binding Cache" et "Binding Update List":

   ```
   user1:~/usr/src/umip/ CPPFLAGS='-isystem /usr/src/linux/usr/include/'
   ./configure --enable-vt
   ```

6. Compiler et installer le software:

   ```
   user1:~/usr/src/umip/ sudo make
   user1:~/usr/src/umip/ sudo make install
   ```

3.1.4 StrongSwan

La dernière étape de l'installation c'est installer le software StrongSwan pour implémenter IKEv2 avec la mobilité IPv6 et pouvoir ,par exemple, protéger les messages BU et BA dynamiquement.

Pour cela, il faudra suivre les étapes suivantes:

1. Télécharger StrongSwan de la page officielle:

   ```
   user1:~/ sudo wget http://download.strongswan.org/strongswan-
   5.7.2.tar.gz
   ```

2. Décompresser le paquet:

   ```
   user1:~/ sudo tar xjvf strongswan-5.7.2.tar.gz
   ```

3. Configurer le logiciel:

```
user1:~/ sudo ./configure --prefix=/usr/local/ --
libexecdir=/usr/local/libexec --libdir=/usr/local/lib --
sysconfdir=/usr/local/etc
```

4. Installer le paquet suivant:

```
user1:~/ sudo apt-get install libgmp-dev
```

5. Compiler et installer le logiciel:

```
user1:~/ sudo make
user1:~/ sudo make install
```

3.2 Scénario

Dans cette section nous allons le modèle d'architecture générale SCOOP@F et le modèle d'architecture de mobilité IPv6.

3.2.1 Modèle d'architecture générale SCOOP@F

La figure 20 illustre le modèle d'architecture générale SCOOP@F.

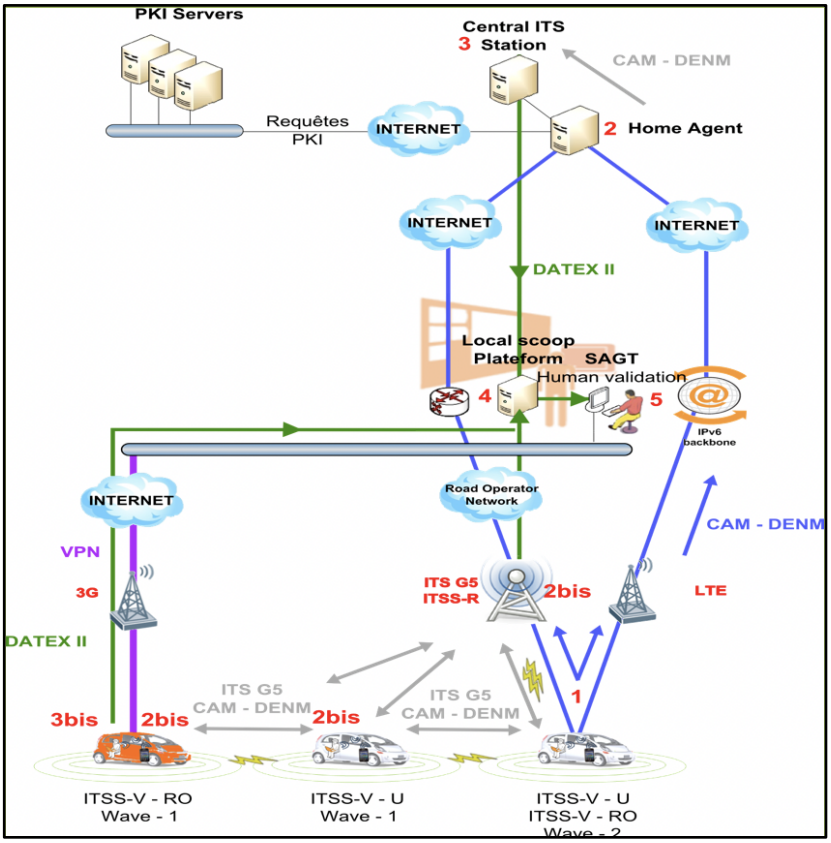


Figure 24. Modèle d'architecture générale SCOOP@F. [4]

3.2.2 Modèle d'architecture de mobilité IPv6

La figure 21 illustre le modèle d'architecture simplifié pour pouvoir valider la mobilité du véhicule au sein de cette architecture.

Pour illustrer le fonctionnement de la mobilité IPv6, nous avons suivi 3 étapes:

1. Étape 1: Mettre en place un environnement de la mobilité IPv6 non sécurisée pour maîtriser les communications échangées entre les différents dispositifs.
2. Étape 2: Mettre en place un environnement de la mobilité IPv6 sécurisée manuellement (Les SAs sont installées manuellement).
3. Étape 3. Mettre en place un environnement de la mobilité IPv6 sécurisée avec IKEv2 (dynamiquement).

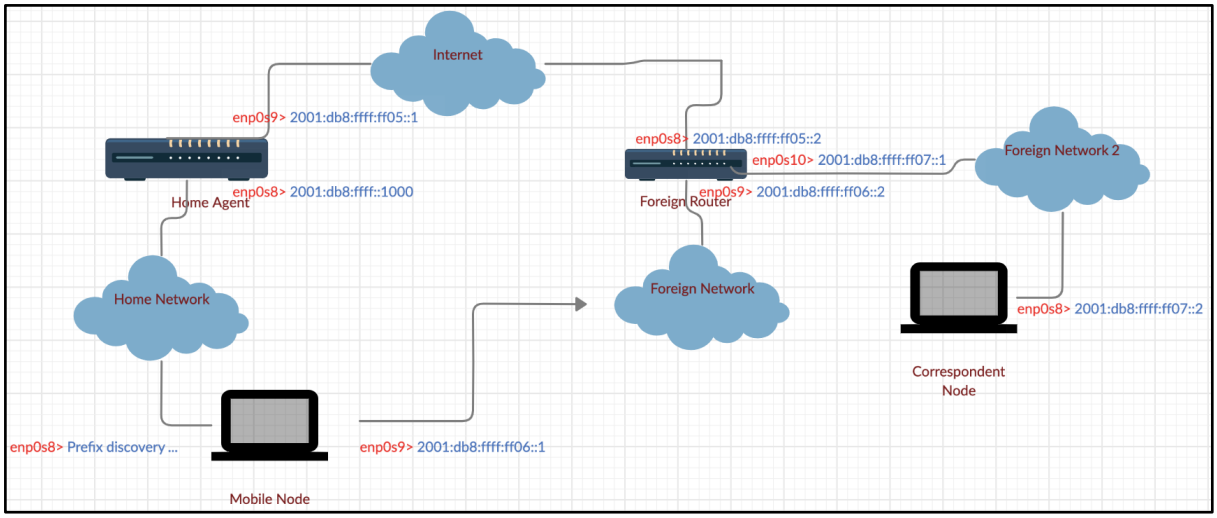


Figure 25. Modèle d'architecture de mobilité IPv6.

3.3 Implémentation de la Mobilité IPv6 non sécurisé

Dans cette partie nous décrivons les étapes pour implémenter un environnement comprenant un HA, un MN et un CN.

Le MN représente un véhicule (comme montre la figure 20). Le CN peut représenter dans l'architecture SCOOP@F les serveurs de la PKI ou le Noeud National.

Nous commençons par mettre le HA et le MN dans le même réseau et après faire que le Mobile Node se déplace à un réseau étranger pour analyser les différents messages échangés entre le Mobile Node et le Home Agent (BU et BA). Pour simuler un changement de réseau de la part du MN, il suffit d'éteindre l'interface "enp0s8" et allumer l'interface "enp0s9" (interface éteinte dès le début). Ensuite, nous allons mettre un CN dans un autre réseau étranger et vérifier que la communication entre le MN et ce CN est établie quand celui-là utilise l'adresse HoA du MN pour lui envoyer des messages. Finalement, nous allons activé l'option "Route Optimization" pour que le CN et le MN optimisent le chemin pour se communiquer plus rapidement et éviter les délais de communication.

Cet environnement ne va pas être sécurisé, pour tant, n'importe quel utilisateur qui se met entre le Home Agent et le Mobile Node peut intercepter et analyser en détails les paquets échangés.

Dans cette partie nous allons lister les fichiers de configurations utilisés sur chaque dispositif pour implémenter notre scénario de la Mobilité IPv6 non sécurisé. Nous allons configurer séparément le HA, le MN, le routeur du réseau étranger et le CN. Chacun de ces dispositifs est simulé avec une machine virtuelle en utilisant une image Debian.

3.3.1 Configuration du Home Agent

Pour bien configurer le HA de notre scénario, il faut suivre plusieurs étapes de configurations:

1. La première étape est de configurer toutes les interfaces de notre HA. Pour cela, il faut éditer le fichier */etc/interfaces/network*:

   ```
   HomeAgent:~/ sudo gedit /etc/interfaces/network
   ```

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug enp0s8
iface enp0s8 inet6 static
    address 2001:db8:ffff:0::1000
    netmask 64

allow-hotplug enp0s9
iface enp0s9 inet6 static
    address 2001:db8:ffff:05::1
    netmask 64
```

Figure 26. Configuration des interfaces HA.

2. Activer le routage des paquets sur le HA. Allez au fichier */etc/sysctl.conf* et vérifier que l'option IP Forwarding est activé:

   ```
   net.ipv6.conf.all.forwarding=1
   ```

3. Créer un fichier *radvd.conf* dans le dossier */etc/* pour que le HA puisse envoyer des messages RA:

```
interface enp0s8
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 3;
    MaxRtrAdvInterval 1;
    AdvIntervalOpt on;
    AdvHomeAgentFlag on;
    AdvHomeAgentInfo on;
    HomeAgentLifetime 1800;
    HomeAgentPreference 10;

       prefix 2001:db8:ffff:0::1000/64
       {
           AdvRouterAddr on;
           AdvOnLink on;
           AdvAutonomous on;
       };
};
```

Figure 27. Configuration Radvd pour le HA.

4. La configuration UMIP pour le HA va être créée dans */usr/local/etc* sous le nom *mip6d.conf*:

```
NodeConfig HA;
DebugLevel 10;

Interface "enp0s8";
DefaultBindingAclPolicy deny;

UseMnHaIPsec disabled;
KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 2001:db8:ffff:0::1000;
    HomeAddress 2001:db8:ffff:0::1/64;

    IPsecPolicy Mh UseESP 11 12;
    IPsecPolicy TunnelPayload UseESP 13 14;
    IPsecPolicy ICMP UseESP 15 16;
}
```

Figure 28. Configuration mip6d pour le HA.

5. Pour assurer la connectivité de notre topologie présentée précédemment, il faudra configurer statiquement le HA pour apprendre les routes vers tous les sous-réseaux de la topologie:

```
HomeAgent:~/ sudo ip -6 route add 2001:db8:ffff:ff06::/64 via
2001:db8:ffff:ff05::2
HomeAgent:~/ sudo ip -6 route add 2001:db8:ffff:ff07::/64 via
2001:db8:ffff:ff05::2
```

3.3.2 Configuration du Mobile Node

Pour configurer le MN, nous allons procéder presque de la même manière que pour le HA:

1. La première étape est de configurer toutes les interfaces de notre MN. Pour cela, il faut éditer le fichier */etc/interfaces/network*:

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug enp0s9
iface enp0s9 inet6 static
    address 2001:db8:ffff:06::1
    netmask 64
```

Figure 29. Configuration des interfaces MN.

2. La configuration UMIP pour le MN va être créée dans */usr/local/etc* sous le nom *mip6d.conf*:

```
NodeConfig MN;
DebugLevel 10;
OptimisticHandoff enabled;
DoRouteOptimisationMN disabled;
MnHaxHaBindingLife 60;

Interface "enp0s8"{
    MnIfPreferece 1;
}
Interface "enp0s9"{
    MnIfPreferece 2;
}

MnHomeLink "enp0s8" {
    HomeAgentAddress 2001:db8:ffff:0::1000;
    HomeAddress 2001:db8:ffff:0::1/64;
}

UseMnHaIPsec disabled;
KeyMngMobCapability disabled;

IPsecPolicySet {
    HomeAgentAddress 2001:db8:ffff:0::1000;
    HomeAddress 2001:db8:ffff:0::1/64;

    IPsecPolicy Mh UseESP 11 12;
    IPsecPolicy TunnelPayload UseESP 13 14;
    IPsecPolicy ICMP UseESP 15 16;
}
```

Figure 30. Configuration mip6d pour le MN.

3.3.3 Configuration du Foreign Router

La configuration du FR est plus simple:

1. La première étape est de configurer toutes les interfaces de notre FN. Pour cela, il faut éditer le fichier */etc/interfaces/network*:

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug enp0s8
iface enp0s8 inet6 static
    address 2001:db8:ffff:05::2
    netmask 64

allow-hotplug enp0s9
iface enp0s9 inet6 static
    address 2001:db8:ffff:06::2
    netmask 64

allow-hotplug enp0s10
iface enp0s10 inet6 static
    address 2001:db8:ffff:07::1
    netmask 64
```

Figure 31. Configuration des interfaces FN.

2. Créer un fichier *radvd.conf* dans le dossier */etc/* pour que le FN puisse envoyer des messages RA.

```
interface enp0s9
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 3;
    MaxRtrAdvInterval 1;
    AdvIntervalOpt on;
    AdvHomeAgentFlag on;
    AdvHomeAgentInfo on;
    HomeAgentLifetime 1800;
    HomeAgentPreference 10;

       prefix 2001:db8:ffff:ff06::/64
       {
           AdvRouterAddr on;
           AdvOnLink on;
           AdvAutonomous on;
       };
};
```

Figure 32. Configuration Radvd pour le FR.

3. Activer le routage des paquets sur le MN. Allez au fichier */etc/sysctl.conf* et vérifier que l'option IP Forwarding est activée:

   ```
   net.ipv6.conf.all.forwarding=1
   ```

4. Pour assurer la connectivité de notre topologie présentée précédemment, il faudra configurer statiquement le HA pour apprendre les routes vers tous les sous-réseaux de la topologie:

   ```
   user1:~/ sudo ip -6 route add 2001:db8:ffff::/64 via 2001:db8:ffff:ff05::1
   ```

3.3.4 Configuration du CN

La configuration du CN est similaire à celle du MN:

1. La première étape est de configurer toutes les interfaces de notre MN. Pour cela, il faut éditer le fichier */etc/interfaces/network*:

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug enp0s8
iface enp0s8 inet6 static
    address 2001:db8:ffff:07::2
    netmask 64
    gateway 2001:db8:ffff:ff07::1
```

Figure 33. Configuration des interfaces CN.

2. La configuration UMIP pour le CN va être créée dans */usr/local/etc* sous le nom *mip6d.conf.* Cette configuration sera utilisée seulement au cas où le CN et le MN utilisent le chemin optimisé pour se communiquer:

```
NodeConfig CN;
DoRouteOptimizationCN enabled;
```

Figure 34. Configuration mip6d pour le CN.

3.3.5 Lancer l'exécution

Pour faire marcher notre scénario MIPv6, il faut suivre les étapes suivantes:

1. Lancer le service *radvd* côté HA et FR pour qu'ils puissent envoyer le préfixe de leur réseau et d'autres informations pour que le MN puisse savoir sur quel réseau il se trouve:

```
HomeAgent:~/ sudo radvd -C /etc/radvd.conf
ForeignRouter:~/ sudo radvd -C /etc/radvd.conf
```

2. Vérifier que le MN a reçu le message RA pour auto-configurer son interface HN et lancer le service mip6d sur les deux côtés (ce service va être lancé le côté CN **SEULEMENT** quand nous activions le chemin optimisé) et essayer de lancer ce service au même temps sur les différents dispositifs:

```
HomeAgent:~/ sudo mip6d -c /usr/local/etc/mip6d.conf
MobileNode:~/ sudo mip6d -c /usr/local/etc/mip6d.conf
CorrespondentNode:~/ sudo mip6d -c /usr/local/etc/mip6d.conf
```

3. Lancer Wireshark pour analyser les paquets.
4. Activer l'interface "enp0s9" et désactiver l'interface "enp0s8" pour simuler un changement de réseau du côté MN.

```
MobileNode:~/ sudo ip link set enp0s8 down
MobileNode:~/ sudo ifup enp0s9
```

3.3.6 Résultats

Pour analyser les résultats en détail, nous avons choisi d'utiliser WireShark.

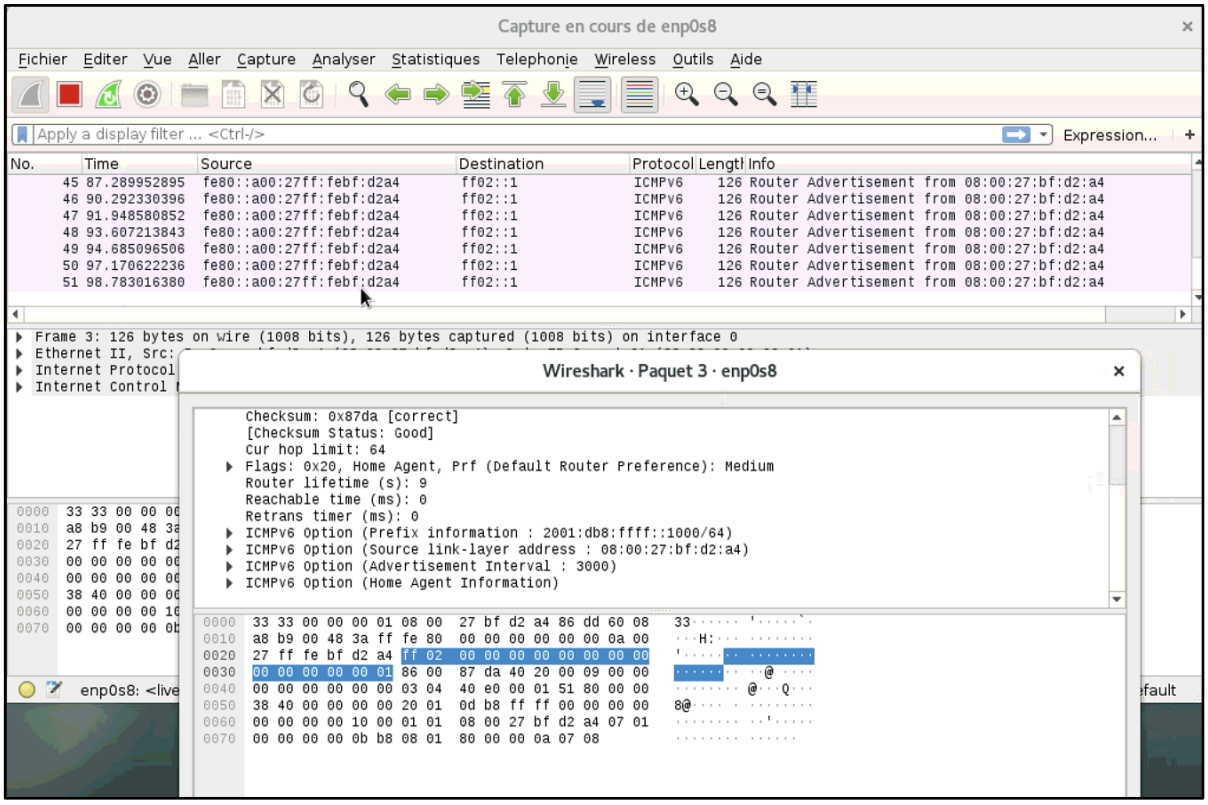


Figure 35. Message RA.

La figure 30 illustre les messages RAs envoyés par le HA sur le réseau HN à tous les noeuds du sous-réseau pour que le MN puisse auto-configurer son interface connectée au réseau HN. Ce message contient l'adresse source "link-local" du HA, le préfixe du sous-réseau et autres informations.

Le MN envoie un message RS pour recevoir le message montré sur l'image 30. Une fois que le MN reçoit le message RA, il va auto-configurer l'interface "enp0s8" connectée au HN en attribuant une adresse globale à cette interface.

L'adresse "link-local" du HA est utilisée par le MN comme l'adresse IP de passerelle par défaut.

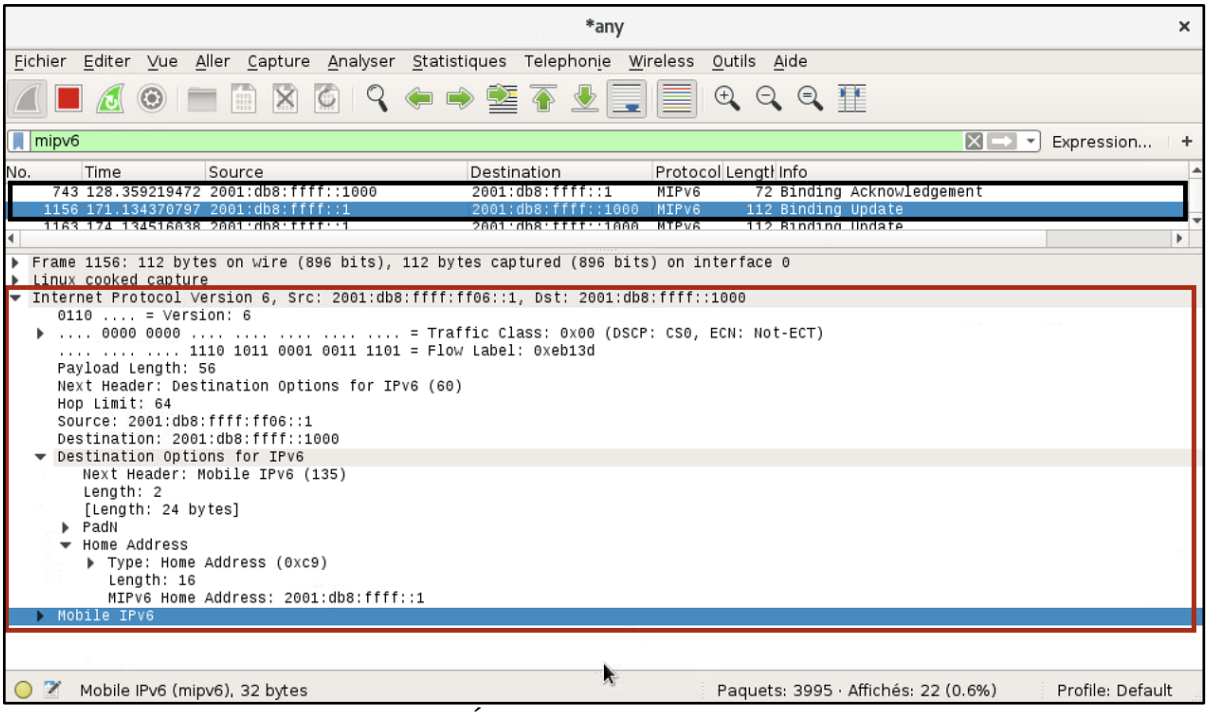


Figure 36. Échange des messages BU et BA.

La figure 31 illustre les message “Binding Update” et “Binding Acknowledgement”. Les messages BU et BA sont échangés même avant que le MN change de réseau (Normalement cette action ne doit pas être effectuée mais le logiciel UMIP le fait de cette manière pour que le MN sache son adresse HoA. Au moment que le MN change de réseau, il reçoit des messages RA et ces messages lui permettent de détecter ce changement et il envoie tout de suite un message BU au HA. Si vous analysez le paquet IPv6, vous allez constater l'existence des champs que nous avons déjà expliqués dans ce rapport: l’adresse source, l’adresse destination, le champ “Destination Options for IPv6 et le champ “Mobile IPv6”. En analysant les paquets “Binding Update”, nous n’avons pas trouvé des informations concernant l’adresse “Care-Of Address”, ce qui explique qu’il n’y a eu aucun changement.

```
Mon Sep 30 16:12:34 mh_bu_parse: Binding Update Received
Mon Sep 30 16:12:34 tunnel_mod: modifying tunnel 7 end points with from 2001:db8
:ffff:0:0:0:0:1000 to 2001:db8:ffff:ff06:0:0:0:1
Mon Sep 30 16:12:34 mh_send_ba: status Binding Update accepted (0)
Mon Sep 30 16:12:34 mh_send: sending MH type 6
from 2001:db8:ffff:0:0:0:0:1000
to 2001:db8:ffff:0:0:0:0:1
Mon Sep 30 16:12:34 mh_send: remote CoA 2001:db8:ffff:ff06:0:0:0:1
```

Figure 37. Logs du service mip6d côté HA.

La figure 32 illustre quelques logs qui montrent le moment de la réception du message BU envoyé par le MN au HA, l’établissement d’un tunnel entre les deux et l’envoi d’un message BA.

```
== BUL_ENTRY ==
Home address    2001:db8:ffff:0:0:0:0:1
Care-of address 2001:db8:ffff:ff06:0:0:0:1
CN address      2001:db8:ffff:0:0:0:0:1000
 lifetime = 60,  delay = 32000
 flags: IP6_MH_BU_HOME IP6_MH_BU_ACK
 ack wait
 dev enp0s9 last_coa 2001:db8:ffff:ff06:0:0:0:1
 lifetime 47 / 60 seq 33742 resend 72 delay 32(after 20s) expires 47
```

Figure 38. Entrée BUL au niveau du HA.

La figure 32 illustre l’entrée BUL d’un MN au moment de recevoir le message BA.

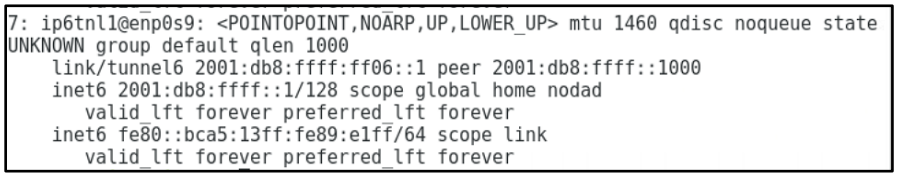


Figure 39. Interface tunnel MN.

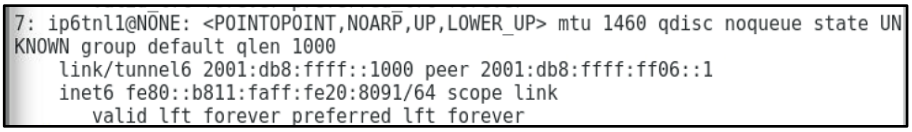


Figure 40. Interface tunnel HA.

Les figures 33 et 34 illustrent la création des interfaces tunnel entre le HA et le MN.

Figure 41. Ping au MN.

La figure 33 illustre la valeur TTL d'un paquet envoyé du CN au MN qui se trouve dans un réseau étranger qui est proche de lui. Ce paquet ne va pas être transmi directement, il va communiquer d'abord avec le HA et le HA va envoyer ce paquet à traver le tunnel déjà établi. C'est la raison pour laquelle cette valeur est énorme.
Pour activer le chemin optimisé, il faut changer une ligne du fichier de configuration *mip6d* du MN:

```
DoRouteOptimisationMN enabled;
```

Notez que cette fois, le service mip6d doit être lancé presque en même temps depuis les trois dispositifs (HA,MN et CN).

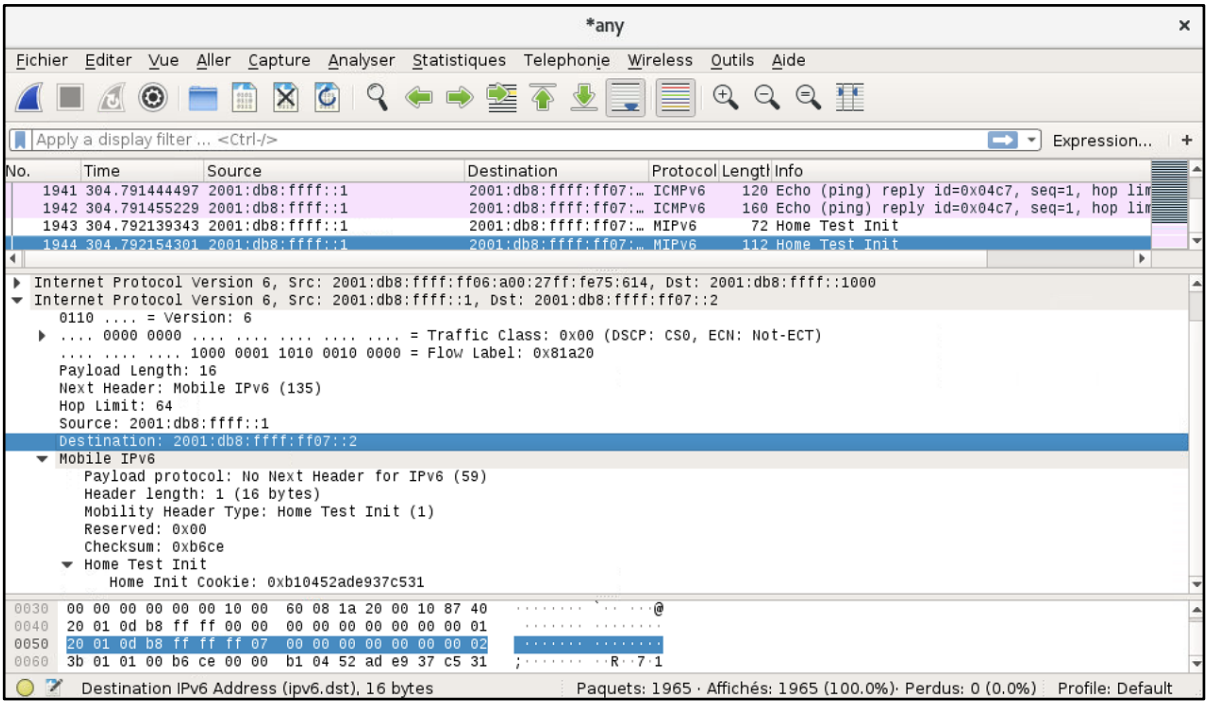


Figure 42. Message Home Test Init envoyé par le MN.

La figure 34 illustre le moment où le CN essaye d'utiliser le mode chemin optimisé pour envoyer ses paquets au MN qui se trouve juste à côté. Vous observez qu'il y a un message de type "Home Test Init" envoyé par le MN au CN et ce paquet contient un champ appelé "Home Init Cookie".

Même si ce paquet contient l'adresse source attribuée à l'adresse HoA du MN et l'adresse destination est attribuée à celle du CN, ce message va d'abord passer par le HA et ensuite va être envoyé au CN parce que l'adresse source est une adresse du réseau HN pour tant ce paquet doit obligatoirement passer par le réseau HN.

Le CN va recevoir un autre message de type "Care-of Test Init" qui va contenir un autre champ appelé "Care-of Init Cookie". Ce dernier message à différence de l'autre il va être envoyé directement du MN au CN sans passer par le HA. Pourtant, l'adresse source de ce paquet est attribuée à l'adresse CoA du MN. Ce message est illustré sur la figure 35.

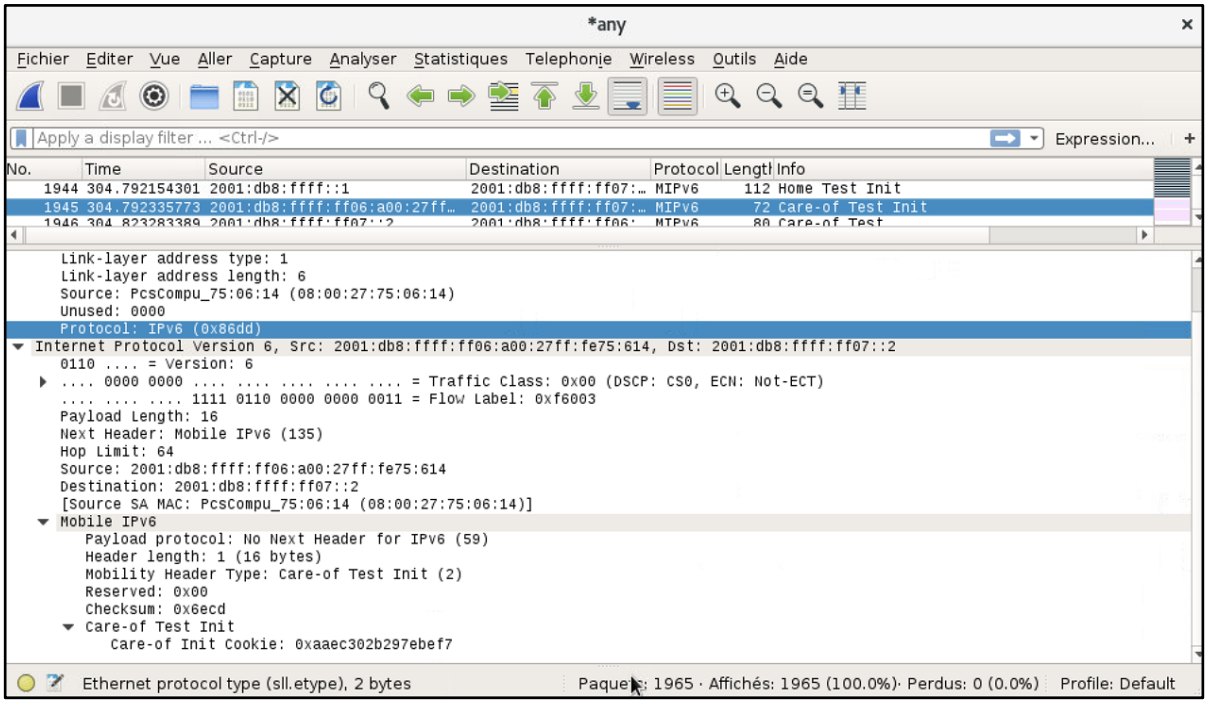


Figure 43. Message Care-of Test Init envoyé par le CN.

Comme vous pouvez le constater le constater, le CN a reçu jusqu'à maintenant deux cookies différentes: le premier est présent dans le message "Home Test Init" qui passe par le HA avant d'arriver au CN et le deuxième est présent dans le message "Care-of Test Init" qui est directement envoyé au CN sans passer par le HA.

En envoyant ces deux messages, les MN garantit au CN qu'il peut communiquer avec les deux adresses (HoA et CoA).

Le CN au moment de recevoir ces deux messages, va envoyer deux messages différents "Home Test" et "Care-of Test". Le message "Home Test" va contenir la même valeur de cookie envoyée par le MN dans le message "Home Test Init" et en plus, il va envoyer une valeur dans le champ appelé "Home Keygen Token". Ces deux cookies vont être le moyen d'authentification dans les messages de signalisation comme le BU émis au CN. Le message "Care-of Test" va contenir la même valeur de cookie envoyée par le MN dans le message "Care-of Test Init" et en plus, il va contenir une valeur dans le champ appelé aussi "Home Keygen Token". Ces messages sont illustrés sur la figure 36.

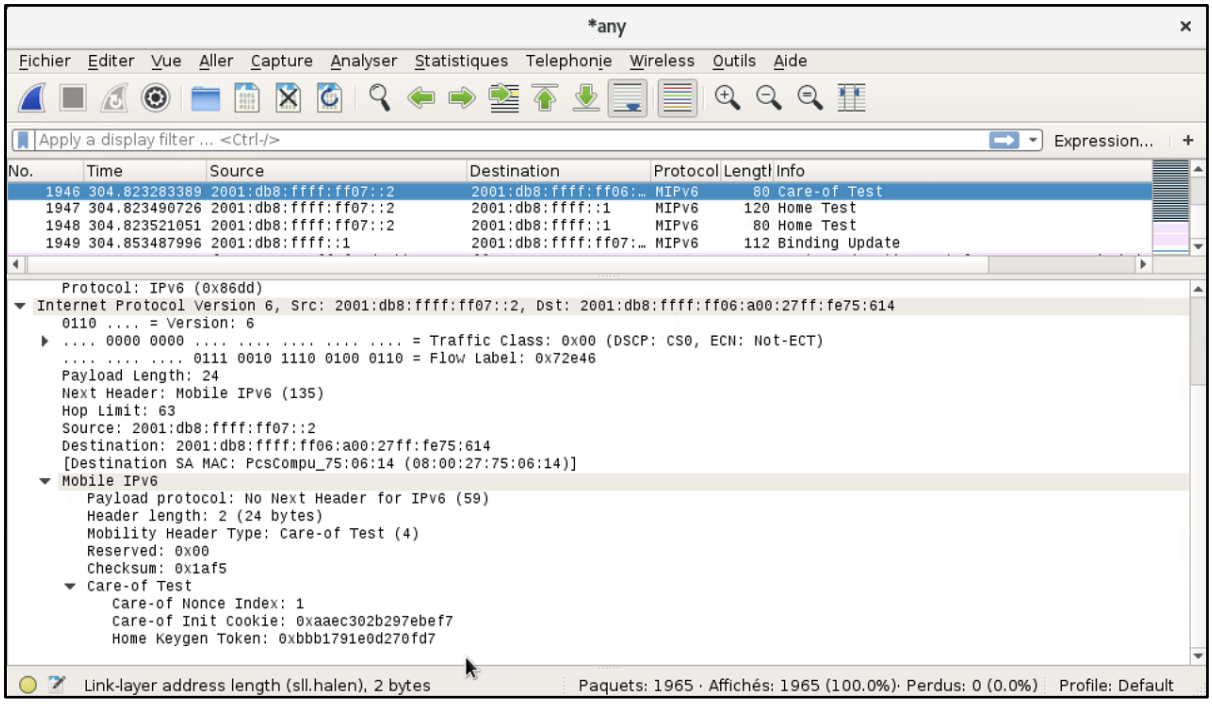


Figure 44. Messages Care-of Test, Home Test et BU échangés entre le MN et le CN.

Sur la figure 36 vous pouvez constater que le message BU est envoyé à l'adresse du CN cette fois-ci.

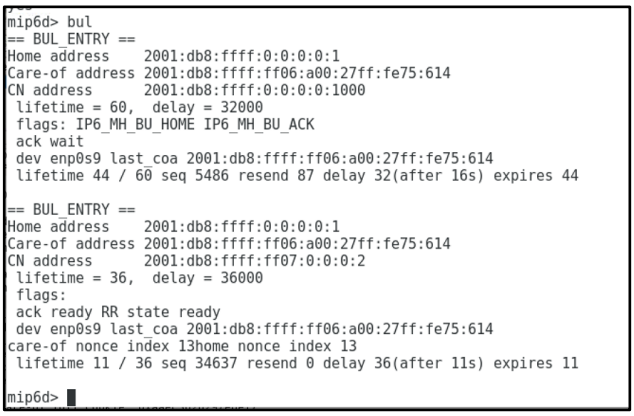


Figure 45. L'entrée "BUL" du MN

La figure 37 illustre la nouvelle entrée du MN après avoir envoyé un BU au CN.

Les étapes à suivre pour obtenir cette entrée sont les suivantes:

```
MobileNode:~/ sudo telnet localhost 7777
mip6d> verbose yes
mip6d> bul
```

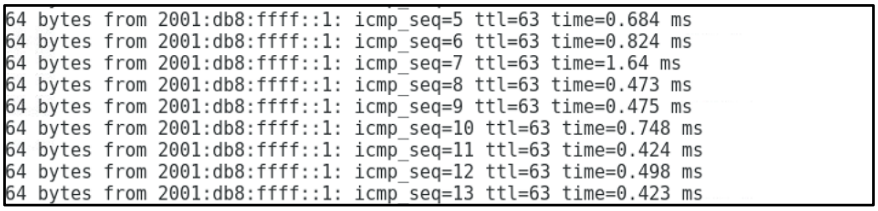


Figure 46. Ping au MN

La figure 38 illustre le moment d'envoyer des messages icmpv6 pour vérifier si ces paquets arrivent plus rapide qu'avant.

3.4 Implémentation de la Mobilité IPv6 sécurisé manuellement

Dans cette partie nous allons sécuriser les échanges entre le Home Agent et le Mobile Node. Plus précisément, nous allons configurer la sécurité manuellement, c'est à dire, installer les SAs manuellement. Pour cela, il faut créer un fichier de configuration additionnel sur les dispositifs (HA et MN) et changer une ligne dans le fichier mip6d sur le HA et le MN.

La ligne à changer est la suivante:

```
UseMnHaIPsec disabled;
```

Cette fois-ci, au lieu de "disabled" nous allons attribuer la valeur "**enabled**".

Le fichier de configuration sera ajouté dans le chemin */usr/local/etc/* et il s'appelle *setkey.conf*:

```
flush;
spdflush;
# MN1 -> HA transport SA for BU
add 2001:db8:ffff::1 2001:db8:ffff::1000 esp 0x11
    -u 11
    -m transport
    -E 3des-cbc "MIP6-011--12345678901234"
    -A hmac-sha1 "MIP6-011--1234567890" ;
# HA -> MN1 transport SA for BA
add 2001:db8:ffff::1000 2001:db8:ffff::1 esp 0x12
    -u 12
    -m transport
    -E 3des-cbc "MIP6-012--12345678901234"
    -A hmac-sha1 "MIP6-012--1234567890" ;
# MN1 -> HA tunnel SA for any traffic
add 2001:db8:ffff::1 2001:db8:ffff::1000 esp 0x13
    -u 13
    -m tunnel
    -E 3des-cbc "MIP6-013--12345678901234"
    -A hmac-sha1 "MIP6-013--1234567890" ;
# HA -> MN1 tunnel SA for any traffic
add 2001:db8:ffff::1000 2001:db8:ffff::1 esp 0x14
    -u 14
    -m tunnel
    -E 3des-cbc "MIP6-014--12345678901234"
    -A hmac-sha1 "MIP6-014--1234567890" ;
# MN1 -> HA transport SA for ICMP (including MPS/MPA)
add 2001:db8:ffff::1 2001:db8:ffff::1000 esp 0x15
    -u 15
    -m transport
    -E 3des-cbc "MIP6-015--12345678901234"
    -A hmac-sha1 "MIP6-015--1234567890" ;
# HA -> MN1 transport SA for ICMP (including MPS/MPA)
add 2001:db8:ffff::1000 2001:db8:ffff::1 esp 0x16
    -u 16
    -m transport
    -E 3des-cbc "MIP6-016--12345678901234"
    -A hmac-sha1 "MIP6-016--1234567890" ;
```

Figure 47. Fichier de configuration setkey.

La figure 39 illustre comment vont être sécurisés les différents messages: BU, BA, Payload et ICMP. Ce fichier contient le mode IPsec utilisé pour chaque échange, les algorithmes de chiffrement dans le mode ESP et aussi dans le mode AH.

Tous les autres fichiers de configuration mentionnés dans la section 3.3 seront réutilisés (radvd, mip6d, interfaces).

3.4.1 Lancer le service

Dans cet exemple nous allons procéder presque de la même manière que la précédente pour lancer le service. La différence est le lancement du service *setkey* sur les deux machines (HA et MN) pour installer les SAs, et le service est lancé juste avant de lancer le service *mip6d*.

```
HomeAgent:~/ sudo setkey -f /usr/local/etc/setkey.conf
MobileNode:~/ sudo setkey -f /usr/local/etc/setkey.conf
```

3.4.2 Résultats

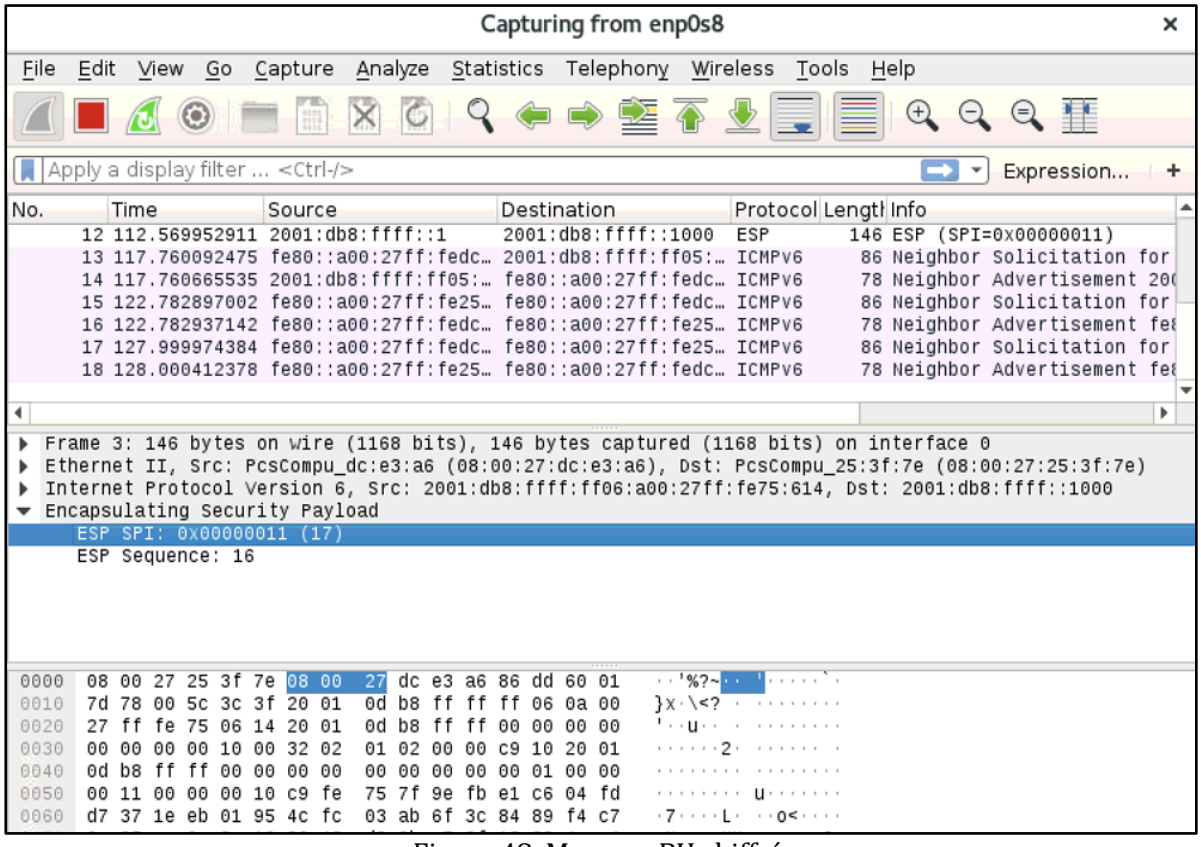


Figure 48. Message BU chiffré.

La figure 40 illustre le moment où le MN envoie le message BU. Cette fois-ci le message est chiffré grâce aux SAs installées sur les deux machines. En analysant ce paquet sur WireShark nous ne pouvons pas savoir de quel message s'agit-il, mais nous reconnaissons que c'est un message BU parce que dans le fichier *setkey.conf* et *mip6d.conf* nous avons spécifié que le message BU aura un champ ESP SPI avec la valeur 0x11.

Par contre, nous n'avons pas réussi à voir le paquet BA par WireShark. Nous sommes arrivés à la conclusion qu'il a eu une faille dans le software, mais nous avons constaté que le tunnel a été bien établi entre le HA et le MN (vérifié en consultant les interfaces créées sur chaque dispositif) .

En plus, pour garantir que le tunnel IPsec a été bien établi, nous avons envoyé des messages de la part du CN au MN et ces messages ont été interceptés par WireShark est ils sont chiffrés aussi en utilisant le SPI que nous avions déjà attribué. La figure 41 illustre les messages chiffrés interceptés.

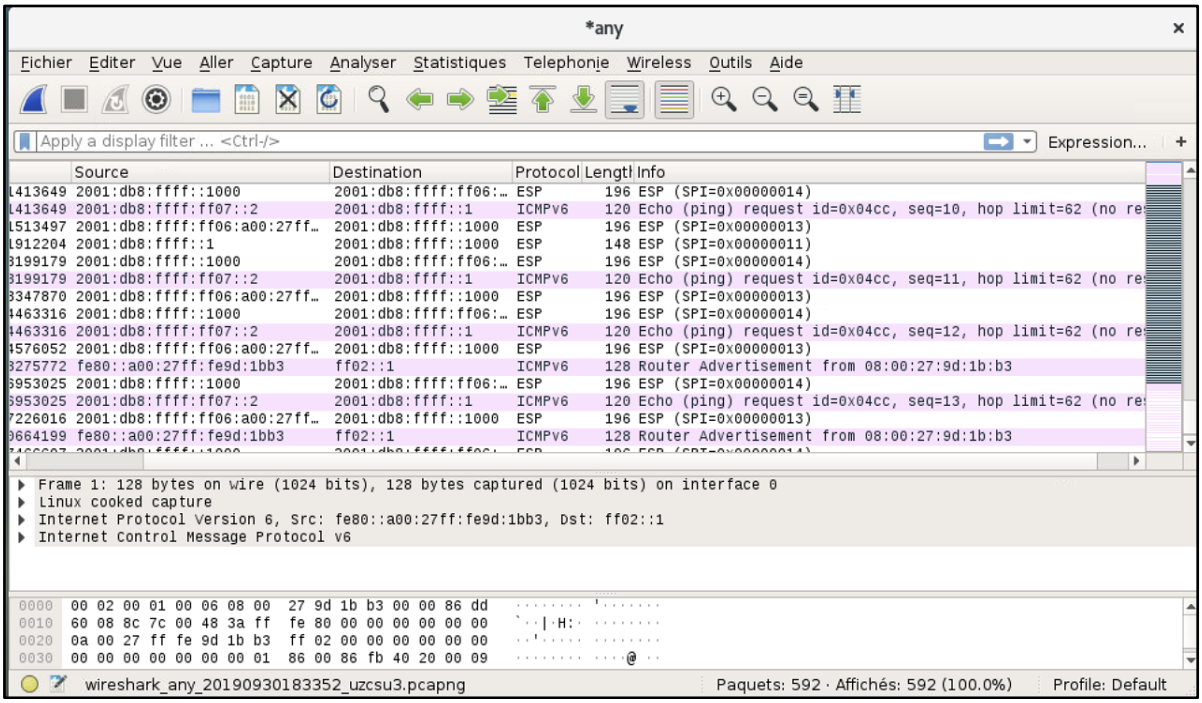


Figure 49. Messages envoyés d'un CN.

Le mode chemin optimisé n'est pas supporté quand nous utilisons l'IPsec. C'est la raison pour laquelle les messages envoyés par le CN doivent passer par le HA du MN.

3.5 Implémentation de la Mobilité IPv6 sécurisé avec IKEv2

Maintenant, nous allons utiliser le software StrongSwan pour garantir la sécurité entre le Home Agent et le Mobile Node dynamiquement. Avant de lancer le service du StrongSwan, il nous faut configurer le HA et le MN pour supporter IKEv2.

3.5.1 Configuration du Home Agent

Nous allons réutiliser encore une fois les mêmes fichiers de configuration mentionnés précédemment, sauf le fichier *setkey.conf*. En plus, nous allons créer un fichier de configuration *ipsec.conf* dans le chemin */usr/local/etc/ipsec.conf*:

```
config setup
        charondebug="knl 2"
conn %default
        keyexchange=ikev2
        reauth=no
        mobike=no
        installpolicy=no
        ike=aes128-sha256-modp2048!
        esp=aes128-sha256-modp2048!
conn mh
        also=ha
        leftsubnet=2001:db8:ffff::1000/128
        leftprotoport=135/0
        rightprotoport=135/0
        type=transport_proxy
conn tunnel
        also=ha
        leftsubnet=::/0
        type=tunnel
conn ha
        left=2001:db8:ffff::1000
        leftcert=homeagentCert.pem
        leftid="CN=homeagent"
        right=%any
conn mobilenode
        rightsubnet=2001:db8:ffff::1/128
        rightid="CN=mobilenode"
conn mobilenode-mh
        also=mobilenode
        aslo=mh
        auto=add
conn mobilenode-tunnel
        also=mobilenode
        also=tunnel
        auto=add
```

Figure 50. Configuration IPsec côté HA.

La figure 42 illustre toutes les SAs que doivent être installés sur le HA pour envoyer des messages sécurisés au moment de communiquer avec le MN.

Généralement, il y a deux mécanismes d'authentification entre le HA et le MN: clée partagée et les certificats délivrés par une PKI. Le mécanisme qui va être utilisé est spécifiée dans le fichier *ipsec.secrets* dans le chemin */usr/local/etc/*. Dans notre cas, nous allons utiliser une PKI pour la distribution des certificats et authentification. Ce fichier va contenir la ligne suivante:

```
>: RSA homeagentKey.pem
```

Le certificat du HA et la clée privée vont être dans les fichiers */usr/local/etc/ipsec.d/certs* et */usr/local/etc/ipsec.d/private* respectivement.

3.5.2 Configuration du MN

Nous allons réutiliser encore une fois les mêmes fichiers de configuration mentionnés précédemment, sauf le fichier *setkey.conf*. En plus, nous allons créer un fichier de configuration *ipsec.conf* dans le chemin */usr/local/etc/ipsec.conf*:

```
config setup
        charondebug="knl 2"
conn %default
        keyexchange=ikev2
        reauth=no
        mobike=no
        installpolicy=no
        ike=aes128-sha256-modp2048!
        esp=aes128-sha256-modp2048!
conn mh
        also=ha
        rightsubnet=2001:db8:ffff::1000/128
        leftprotoport=135/0
        rightprotoport=135/0
        type=transport_proxy
        auto=route
conn tunnel
        also=home
        rightsubnet=::/0
        auto=route
conn home
        leftcert=mobilenodeCert.pem
        leftid="CN=mobilenode"
        leftsubnet=2001:db8:ffff::1/128
        right=2001:db8:ffff::1000
        rightid="CN=homeagent"
```

Figure 51. Configuration IPsec côté MN.

La figure 43 illustre toutes les SAs que doivent être installés sur le MN pour envoyer des messages sécurisés au moment de communiquer avec le HA.

Comme nous l'avons déjà mentionné le mécanisme qui va être utilisé est spécifiée dans le fichier *ipsec.secrets* dans le chemin */usr/local/etc/*. Dans notre cas, nous allons utiliser la distribution des certificats et authentification au moyen d'une PKI. Ce fichier va contenir la ligne suivante:

```
>: RSA mobilenodeKey.pem
```

Le certificat du MN et la clée privée vont être dans les fichiers */usr/local/etc/ipsec.d/certs* et */usr/local/etc/ipsec.d/private* respectivement.

3.5.3 Lancer le service

Dans ce cas, nous allons procéder presque de la même manière que dans le cas précédent. La différence c'est que au lieu d'utiliser le fichier *setket.conf* nous allons utiliser le fichier *ipsec.conf* et nous allons le lancer de la manière suivante:

```
HomeAgent:~/ sudo ipsec start
MobileNode:~/ sudo ipsec start
```

3.5.4 Résultats

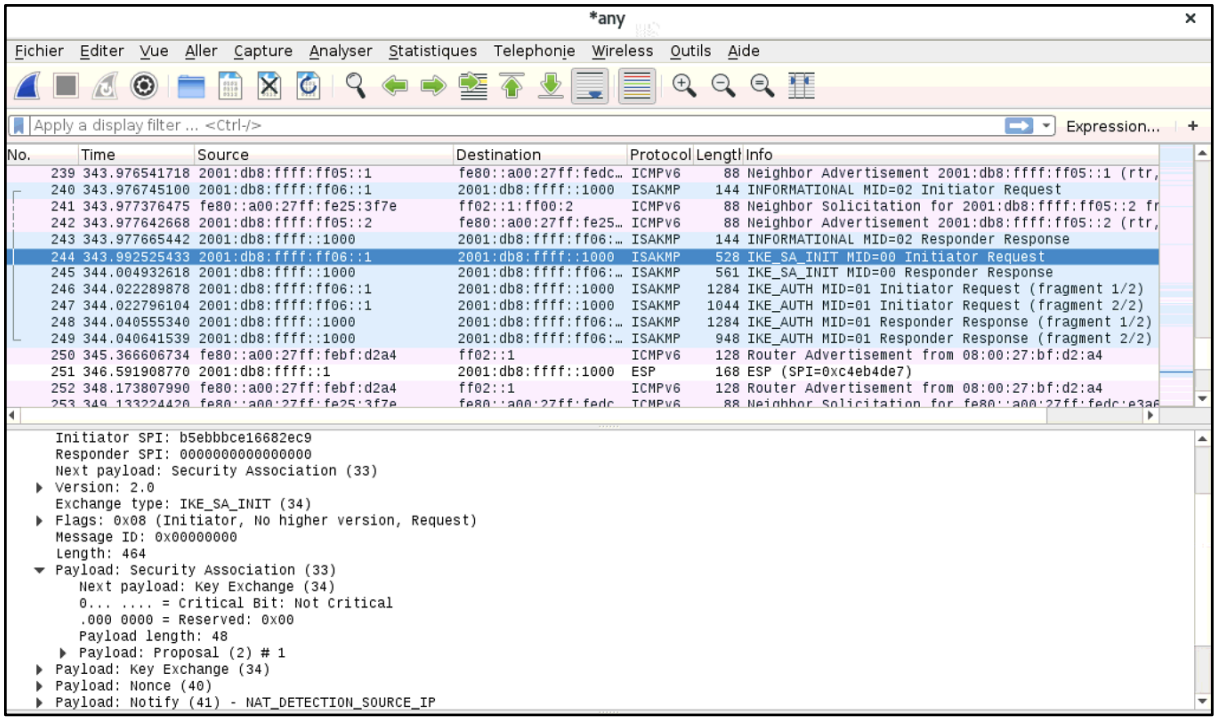


Figure 52. Les messages IKE et le message BU échangés entre le HA et le MN.

La figure 44 illustre comment le Home Agent et le Mobile Node vont échanger la clé de chiffrement pour envoyer les messages d'une manière sécurisée. Par contre, comme dans le cas précédent, nous observons qu'il n'y a que le message Binding Update.

Pour analyser les échanges entre le Home Agent et le Mobile Node avec plus de détails, nous allons analyser le fichier *syslog* qui se trouve dans le chemin */var/log/* du Home Agent.
Les logs qui se trouvent dans le Mobile Node sont similaires.

```
Sep 18 18:07:06 debian org.gnome.Shell.desktop[972]: Window manager warning: Invalid WM_TRANSIENT_FOR window
0x1000007 specified for 0x100002d (Unsaved pa).
Sep 18 18:10:15 debian charon: 00[DMN] Starting IKE charon daemon (strongSwan 5.8.0, Linux 4.10.1, x86_64)
Sep 18 18:10:15 debian charon: 00[KNL] known interfaces and IP addresses:
Sep 18 18:10:15 debian charon: 00[KNL]   lo
Sep 18 18:10:15 debian charon: 00[KNL]     127.0.0.1
Sep 18 18:10:15 debian charon: 00[KNL]     ::1
Sep 18 18:10:15 debian charon: 00[KNL]   enp0s3
Sep 18 18:10:15 debian charon: 00[KNL]     10.0.2.15
Sep 18 18:10:15 debian charon: 00[KNL]     fe80::a00:27ff:fe17:44c3
Sep 18 18:10:15 debian charon: 00[KNL]   enp0s8
Sep 18 18:10:15 debian charon: 00[KNL]     2001:db8:ffff::1000
Sep 18 18:10:15 debian charon: 00[KNL]     fe80::a00:27ff:febf:d2a4
Sep 18 18:10:15 debian charon: 00[KNL]   enp0s9
Sep 18 18:10:15 debian charon: 00[KNL]     2001:db8:ffff:ff05::1
Sep 18 18:10:15 debian charon: 00[KNL]     fe80::a00:27ff:fe25:3f7e
Sep 18 18:10:15 debian charon: 00[CFG] loading ca certificates from '/usr/local/etc/ipsec.d/cacerts'
Sep 18 18:10:15 debian charon: 00[CFG]   loaded ca certificate "CN=authoritycertification" from '/usr/local/etc/
ipsec.d/cacerts/caCert.pem'
Sep 18 18:10:15 debian charon: 00[CFG] loading aa certificates from '/usr/local/etc/ipsec.d/aacerts'
Sep 18 18:10:15 debian charon: 00[CFG] loading ocsp signer certificates from '/usr/local/etc/ipsec.d/ocspcerts'
Sep 18 18:10:15 debian charon: 00[CFG] loading attribute certificates from '/usr/local/etc/ipsec.d/acerts'
Sep 18 18:10:15 debian charon: 00[CFG] loading crls from '/usr/local/etc/ipsec.d/crls'
Sep 18 18:10:15 debian charon: 00[CFG] loading secrets from '/usr/local/etc/ipsec.secrets'
Sep 18 18:10:15 debian charon: 00[CFG]   loaded RSA private key from '/usr/local/etc/ipsec.d/private/
homeagentKey.pem'
Sep 18 18:10:15 debian charon: 00[LIB] loaded plugins: charon aes des rc2 sha2 sha1 md5 mgf1 random nonce x509
revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem fips-prf gmp curve25519 xcbc cmac
hmac attr kernel-netlink resolve socket-default stroke vici updown xauth-generic counters
Sep 18 18:10:15 debian charon: 00[JOB] spawning 16 worker threads
Sep 18 18:10:15 debian charon: 05[CFG] received stroke: add connection 'mobilenode-mh'
Sep 18 18:10:15 debian charon: 05[CFG]   loaded certificate "CN=homeagent" from 'homeagentCert.pem'
Sep 18 18:10:15 debian charon: 05[CFG] added configuration 'mobilenode-mh'
Sep 18 18:10:15 debian charon: 07[CFG] received stroke: add connection 'mobilenode-tunnel'
Sep 18 18:10:15 debian charon: 07[CFG]   loaded certificate "CN=homeagent" from 'homeagentCert.pem'
Sep 18 18:10:15 debian charon: 07[CFG] added child to existing configuration 'mobilenode-mh'
```

Figure 53. Logs côté HA (1)

La figure 45 illustre comment le Home Agent installe le certificat CA, sa clée privée et les protocoles qu'il supporte.

```
Sep 18 18:11:44 debian mip6d[1406]: UMIP Mobile IPv6 for Linux v1.0 started (Home Agent)
Sep 18 18:11:47 debian charon: 10[NET] received packet: from 2001:db8:ffff:0:a00:27ff:fefe:21d7[500] to
2001:db8:ffff::1000[500] (464 bytes)
Sep 18 18:11:47 debian charon: 10[ENC] parsed IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N
(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
Sep 18 18:11:47 debian charon: 10[IKE] 2001:db8:ffff:0:a00:27ff:fefe:21d7 is initiating an IKE_SA
Sep 18 18:11:47 debian charon: 10[CFG] selected proposal: IKE:AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/
MODP_2048
Sep 18 18:11:47 debian charon: 10[IKE] sending cert request for "CN=authoritycertification"
Sep 18 18:11:47 debian charon: 10[ENC] generating IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP)
CERTREQ N(FRAG_SUP) N(HASH_ALG) N(CHDLESS_SUP) N(MULT_AUTH) ]
Sep 18 18:11:47 debian charon: 10[NET] sending packet: from 2001:db8:ffff::1000[500] to
2001:db8:ffff:0:a00:27ff:fefe:21d7[500] (497 bytes)
Sep 18 18:11:47 debian charon: 11[NET] received packet: from 2001:db8:ffff:0:a00:27ff:fefe:21d7[500] to
2001:db8:ffff::1000[500] (1220 bytes)
Sep 18 18:11:47 debian charon: 11[ENC] parsed IKE_AUTH request 1 [ EF(1/2) ]
Sep 18 18:11:47 debian charon: 11[ENC] received fragment #1 of 2, waiting for complete IKE message
Sep 18 18:11:47 debian charon: 12[NET] received packet: from 2001:db8:ffff:0:a00:27ff:fefe:21d7[500] to
2001:db8:ffff::1000[500] (1060 bytes)
Sep 18 18:11:47 debian charon: 12[ENC] parsed IKE_AUTH request 1 [ EF(2/2) ]
Sep 18 18:11:47 debian charon: 12[ENC] received fragment #2 of 2, reassembled fragmented IKE message (2208 bytes)
Sep 18 18:11:47 debian charon: 12[ENC] parsed IKE_AUTH request 1 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH N
(USE_TRANSP) SA TSi TSr N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
```

Figure 54. Logs côté HA (2)

La figure 46 illustre qu'est-ce qui se passe quand le service mip6d est lancé. D'abord nous observons que le MN envoie le paquet "IKE_SA_INIT" pour négocier les algorithmes supportés. Ensuite, le MN inclut son HoA et les "TSi" pour négocier les SAs dans le payload "IKE_AUTH" en utilisant sa clée privée pour le chiffrement.

```
Sep 18 18:11:47 debian charon: 12[IKE] received cert request for "CN=authoritycertification"
Sep 18 18:11:47 debian charon: 12[IKE] received end entity cert "CN=mobilenode"
Sep 18 18:11:47 debian charon: 12[CFG] looking for peer configs matching 2001:db8:ffff::1000
[CN=homeagent]...2001:db8:ffff:0:a00:27ff:fefe:21d7[CN=mobilenode]
Sep 18 18:11:47 debian charon: 12[CFG] selected peer config 'mobilenode-mh'
Sep 18 18:11:47 debian charon: 12[CFG]   using certificate "CN=mobilenode"
Sep 18 18:11:47 debian charon: 12[CFG]   using trusted ca certificate "CN=authoritycertification"
Sep 18 18:11:47 debian charon: 12[CFG] checking certificate status of "CN=mobilenode"
Sep 18 18:11:47 debian charon: 12[CFG] certificate status is not available
Sep 18 18:11:47 debian charon: 12[CFG]   reached self-signed root ca with a path length of 0
Sep 18 18:11:47 debian charon: 12[IKE] authentication of 'CN=mobilenode' with RSA_EMSA_PKCS1_SHA2_384 successful
Sep 18 18:11:47 debian charon: 12[IKE] authentication of 'CN=homeagent' (myself) with RSA_EMSA_PKCS1_SHA2_384
successful
Sep 18 18:11:47 debian charon: 12[IKE] IKE_SA mobilenode-mh[1] established between 2001:db8:ffff::1000
[CN=homeagent]...2001:db8:ffff:0:a00:27ff:fefe:21d7[CN=mobilenode]
Sep 18 18:11:47 debian charon: 12[IKE] scheduling rekeying in 9784s
Sep 18 18:11:47 debian charon: 12[IKE] maximum IKE_SA lifetime 10324s
Sep 18 18:11:47 debian charon: 12[IKE] sending end entity cert "CN=homeagent"
Sep 18 18:11:47 debian charon: 12[CHD] my address: 2001:db8:ffff::1000 is a transport mode proxy for
2001:db8:ffff::1000
Sep 18 18:11:47 debian charon: 12[CHD] other address: 2001:db8:ffff:0:a00:27ff:fefe:21d7 is a transport mode
proxy for 2001:db8:ffff::1
Sep 18 18:11:47 debian charon: 12[CFG] selected proposal: ESP:AES_CBC_128/HMAC_SHA2_256_128/NO_EXT_SEQ
Sep 18 18:11:47 debian charon: 12[KNL] got SPI c15e9e7a
Sep 18 18:11:47 debian charon: 12[KNL] adding SAD entry with SPI c15e9e7a and reqid {1}
Sep 18 18:11:47 debian charon: 12[KNL]   using encryption algorithm AES_CBC with key size 128
Sep 18 18:11:47 debian charon: 12[KNL]   using integrity algorithm HMAC_SHA2_256_128 with key size 256
Sep 18 18:11:47 debian charon: 12[KNL]   using replay window of 32 packets
Sep 18 18:11:47 debian charon: 12[KNL]   HW offload: no
Sep 18 18:11:47 debian charon: 12[KNL] adding SAD entry with SPI c839c773 and reqid {1}
Sep 18 18:11:47 debian charon: 12[KNL]   using encryption algorithm AES_CBC with key size 128
Sep 18 18:11:47 debian charon: 12[KNL]   using integrity algorithm HMAC_SHA2_256_128 with key size 256
Sep 18 18:11:47 debian charon: 12[KNL]   using replay window of 0 packets
Sep 18 18:11:47 debian charon: 12[KNL]   HW offload: no
Sep 18 18:11:47 debian charon: 12[IKE] CHILD_SA mobilenode-mh{1} established with SPIs c15e9e7a_i c839c773_o and
TS 2001:db8:ffff::1000/128[mobility-header/0] === 2001:db8:ffff::1/128[mobility-header/0]
```

Figure 55. Logs côté HA (3)

La figure 47 illustre comment le HA reçoit le certificat du MN pour vérifier que le MN n'est pas entrain d'installer de SAs pour un autre MN. Ensuite, il installe les SAs pour les sélecteurs qu'il a reçu.

```
7: ip6tnl1@enp0s9: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1460 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/tunnel6 2001:db8:ffff:ff06:a00:27ff:fe75:614 peer 2001:db8:ffff::1000
    inet6 2001:db8:ffff::1/128 scope global home nodad
       valid_lft forever preferred_lft forever
    inet6 fe80::acae:30ff:feb4:4426/64 scope link
       valid_lft forever preferred_lft forever
```

Figure 56. Interface tunnel côté HA.

La figure 49 illustre comment le HA a créé le tunnel entre le HA et le MN une fois que le HA a reçu le BU.

4 Conclusions

Ce projet de stage fin d'études m'a aidé à comprendre de manière détaillée le fonctionnement complexe du protocole Mobile IPv6 après avoir configuré chaque dispositif qui participe dans le scénario de la Mobilité IPv6 considéré dans le stage. Nous avons vu que l'avantage principal de ce protocole est la communication sans interruption d'un terminal mobile quand celui-ci change son point d'attachement.

Nous avons testé, analysé et validé plusieurs scénarios afin de comprendre la différence entre eux, comme la mobilité IPv6 non sécurisée, sécurisée manuellement et sécurisée avec IKEv2. Ces différents tests effectués nous ont permis de comprendre plusieurs protocoles en détail.

Tout au long du projet nous avons rencontré des problèmes de configuration, comme par exemple au moment de sécuriser les messages de type "Home Test Init", "Care-of Test Init", "Home Test" et "Care-of Test". Ces messages sont échangés entre un CN et un MN au moment d'utiliser un chemin optimisé. Étant donné que ces messages n'étaient pas sécurisés, n'importe quel utilisateur qui se trouve à l'écoute du trafic dans le réseau CN peut intercepter et analyser ces messages. Pourtant, c'était plus sécurisé de passer par le Home Agent au lieu de contacter directement le MN. Cette procédure peut occasionner des délais supplémentaires.

Comme perspectives, il serait intéressant de:

1. Travailler sur la partie de la sécurité des messages échangés entre le CN et le MN pour assurer la confidentialité et l'intégrité pour avoir des meilleurs performances.
2. Évaluer les performances du à l'utilisation de mobilité IPv6.
3. Porter l'implémentation sur des équipements réels dans la plateforme ITS-Sec déployée par l'équipe CCN.

5 Références

[1] IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6, De Rick Graziani
[2] Recommandations de sécurité relatives à IPsec 1 pour la protection des flux réseau - ANSSI - https://www.ssi.gouv.fr/uploads/2012/09/NT_IPsec.pdf
[3] Mobile IPv6: Protocols and Implementation, De Qing Li, Tatuya Jinmei, Keiichi Shima.
[4] Scoop@F
RFC 791 - INTERNET PROTOCOL
RFC 1812 - Requirements for IP Version 4 Routers
RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification
RFC 4301 - Security Architecture for the Internet Protocol
RFC 4303 - IP Encapsulating Security Payload (ESP)
RFC 6275 - Mobility Support in IPv6
The OpenSource IPsec-based VPN Solution - https://www.strongswan.org/