
MipShim6 : une approche combinée pour la mobilité et la multi-domiciliation

Sébastien Barré* — Amine Dhraief** — Nicolas Montavont** — Olivier Bonaventure*

* Université catholique de Louvain
Pl. Ste Barbe, 2
B-1348 Louvain-la-Neuve
{sebastien.barre,olivier.bonaventure}@uclouvain.be

** Institute Telecom / Telecom Bretagne
rue de la Châtaigneraie, 2
F-35576 Cesson-Sévigné
{amine.dhraief,nicolas.montavont}@telecom-bretagne.eu

RÉSUMÉ. Les supports de la mobilité et la multi-domiciliation dans IPv6 ont jusqu'à présent été considérés de manière indépendante et ont notamment résulté dans la publication des standards Mobile IPv6 et SHIM6 respectivement. Or, plusieurs cas d'usage que nous soulevons démontrent qu'il y a un intérêt fort pour l'utilisation simultanée de ces deux protocoles. En particulier, la solution combinée permet de remplacer le mécanisme d'optimisation de routage de Mobile IPv6 par une technique plus légère et équivalente en fonctionnalité, sans besoin d'extension des protocoles déjà définis. Nous présentons également une évaluation de notre architecture, sur base de notre implémentation pour Linux, disponible publiquement.

ABSTRACT. To date, mobility and multihoming support in IPv6 have been considered independently of each other, which lead to the publication of the standards Mobile IPv6 and Shim6, respectively. Nevertheless, several use cases that we show prove that there is a strong interest for simultaneous use of these two protocols. In particular, the combined solution allows for replacing the routing optimization mechanism of Mobile IPv6 with a lighter technique, equivalent in functionality, without needing to extend the already defined protocols. We also provide an evaluation of our architecture, based on our publicly available implementation for Linux.

MOTS-CLÉS : Multi-domiciliation, Mobilité, IPv6

KEY WORDS: Multihoming, Mobility, IPv6

1. Introduction

Les communications numériques connaissent un essor spectaculaire ces dernières années, et après l'avènement de l'Internet fixe, nous assistons aujourd'hui au développement et au déploiement de technologies sans fil de plus en plus performantes qui permettent un Internet mobile. Alors que certaines visent à couvrir l'ensemble d'un territoire, d'autres ont pour fonction d'offrir des débits importants. Cette complémentarité a poussé les constructeurs à proposer des terminaux multi-interfaces, qui intègrent des technologies de communication hétérogènes. Ces nouveaux terminaux posent un nouveau problème d'optimisation, car ils nécessitent un support de la mobilité (changement de réseaux d'accès en cours de communications, dû notamment à des déplacements), et un support de multi-domiciliation (gestion de plusieurs connexions au réseau). Les domaines de la mobilité et de la multi-domiciliation pour IPv6 ont longtemps été considérés séparément. A l'heure où le protocole Shim6 [NOR 09] (solution de multi-domiciliation pour IPv6) vient d'être approuvé comme Request For Comment (RFC), et le protocole Mobile IPv6 [PER 96, JOH 04] (MIPv6) est déjà largement stabilisé, il devient clair qu'un intérêt marqué existe pour l'utilisation conjointe des fonctions de mobilité et de multi-domiciliation.

Certains chercheurs ont proposé une extension de l'un ou l'autre des protocoles pour lui ajouter la fonctionnalité qui manquait. Alors que plusieurs voulaient étendre Shim6 pour lui permettre de gérer la mobilité [DHR 08, RAH 08, CHE 06], d'autres ont étudié l'extension de MIPv6 pour supporter la multi-domiciliation [WAK 09, ROP 08]. En 2006, une première proposition d'associer MIPv6 et Shim6 a été faite par Kianai et al. [KIA 06], mais celle-ci ne supportait pas la multi-domiciliation du réseau visité (seulement celle du réseau mère). Enfin en 2007, l'architecture la plus proche de la nôtre a été proposée par Bagnulo et al. [BAG 07]. Celle-ci est basée sur la cohabitation des protocoles Shim6 et MIPv6. Elle est aussi la première à mentionner la possibilité d'effectuer de l'optimisation de routage en faisant uniquement appel à Shim6.

Dans cet article, nous re-visitons la solution décrite dans [BAG 07] pour proposer une architecture de services modulaire et mettons en avant les contributions suivantes :

Raffinement de l'architecture : Nous décomposons l'architecture de MIPv6 et Shim6 en leurs briques de bases, de manière à obtenir une association de celles-ci qui soit à la fois robuste et aussi légère que possible.

Support avancé : Nous proposons une architecture de services modulaire, capable de supporter la panne d'un agent mère MIPv6 et le mouvement simultané de deux nœuds communicants, que Shim6 seul ne supporte pas.

Implémentation de la solution : Nous partons des implémentations existantes MIPv6 [MIY 04] et LinShim6 [BAR 07a] pour les faire collaborer, et évaluer en laboratoire notre solution MipShim6.

Cet article est organisé comme suit. Nous commençons par introduire les protocoles sur lesquels nous nous basons, MIPv6 et Shim6. Ensuite les détails de notre architecture sont présentés. Enfin, la section 5 décrit notre implémentation et évaluation.

2. Mobilité dans IPv6 : MIPv6

Mobile IPv6 (MIPv6) [JOH 04] est le protocole qui gère la mobilité des hôtes dans l'Internet IPv6. Il a pour objectif d'assurer la continuité des sessions après un mouvement. Dans un envi-

ronnement mobile, la localisation des hôtes change constamment. A chaque changement de réseau, les mobiles doivent modifier le routage des paquets applicatifs, sans rompre les communications en cours. Pour cela, MIPv6 introduit un nouvel élément dans l'architecture du réseau : l'agent mère (ou Home Agent). L'agent mère attribue des adresses IPv6 aux hôtes présents dans son réseau (ou réseau mère). Une telle adresse est appelée adresse mère (ou Home Address, HoA) et représente l'identité permanente du mobile. C'est l'adresse qui doit être utilisée par les applications pour démarrer les communications, même si au niveau du routage d'autres adresses pourront être utilisées. Au cours de son déplacement, un nœud mobile obtient des adresses temporaires (ou Care-of Addresses, CoA) qui sont topologiquement correctes et pourront donc être utilisées pour le routage des paquets vers la localisation courante du nœud mobile.

La première fonctionnalité assurée par l'agent mère est de maintenir la correspondance entre l'identité du nœud et sa localisation courante, plus précisément, entre les adresses mère et care-of du mobile. Pour cela, l'agent mère maintient un cache (appelé Binding Cache) mis à jour par le mobile à chaque acquisition de nouvelle adresse ou après une certaine période de temps. La mise à jour du Binding Cache se fait par l'échange des messages Binding Update (BU) et Binding Ack (Back). Le BU, envoyé par le mobile, contient sa HoA ainsi que sa nouvelle CoA, et il est acquitté par un Back envoyé par l'agent mère. La seconde fonctionnalité fournie par l'agent mère est de capturer tout trafic destiné au mobile et de le relayer vers la nouvelle localisation de celui-ci. Pour cela il établit un tunnel IPv6-dans-IPv6 entre lui et la nouvelle localisation du mobile. Le trafic sortant du mobile est aussi relayé vers ses correspondants à travers l'agent mère via le même tunnel. Ainsi, les échanges entre le mobile et ses correspondants passent obligatoirement par l'agent mère et on parle dans ce cas de "Routage Triangulaire". La troisième fonctionnalité assurée par l'agent mère est le point de rendez-vous. En effet, deux mobiles qui changent simultanément de réseau peuvent continuer à communiquer à travers leurs agent mère respectifs. On parle alors de saut simultané.

Un des points forts de MIPv6 est qu'il n'impose pas aux correspondants d'un mobile d'implémenter MIPv6. Néanmoins, si le correspondant dispose du support de MIPv6 également, il a tout intérêt à utiliser un chemin direct vers le mobile. Cela est rendu possible par l'optimisation de routage MIPv6, ou RO (Routing Optimization), qui consiste à tenir au courant de son emplacement non seulement l'agent mère, mais également le nœud mobile. Aucun tunnel n'est nécessaire dans ce cas.

Limitations de MIPv6 : La principale limitation de MIPv6 est la non-tolérance aux pannes dans le réseau mère, notamment la panne de l'agent mère. L'agent mère constitue le point central dans l'architecture de MIPv6, et sa panne provoquerait la rupture de toutes les communications gérées par cet agent mère, y compris les communications en mode RO. Bien que [FAI 05] propose une solution basée sur une extension à MIPv6, celle-ci ne peut tirer parti d'adresses IPv6 multiples résultant d'une multi-domiciliation du réseau mère.

3. Multi-domiciliation dans IPv6 : Shim6

Le protocole Shim6 [NOR 09] a été développé à l'IETF dans le but d'offrir une solution de multi-domiciliation adaptée au protocole IPv6. Sa caractéristique principale est que chaque hôte dispose de plusieurs adresses IP, une pour chaque fournisseur d'accès auquel son réseau est connecté. Cela oblige chaque machine à choisir une adresse pour chaque communication, et éventuellement

basculer sur une autre adresse en cas de panne du chemin déterminé par la paire d'adresses utilisée à un instant donné. Alors que la détection de pannes et l'équilibrage du trafic entre les fournisseurs étaient en IPv4 contrôlés entièrement par le réseau (notamment via les protocoles de routage), une partie de ce rôle est maintenant transféré dans les hôtes.

Le protocole Shim6 permet d'échanger l'ensemble des adresses d'une machine avec un correspondant. Il fournit également un mécanisme de réécriture des adresses aux deux extrémités de la communication, de manière à présenter aux couches application et transport une adresse stable, tout en changeant l'adresse utilisée pour router le paquet au besoin. L'adresse stable utilisée comme identifiant de connexion est appelée *ULID* (Upper Layer Identifier) dans la terminologie Shim6. Les adresses se trouvant dans les en-têtes des paquets sont appelés des *localisateurs*.

La fonction d'annonce de localisateurs au correspondant nécessite une sécurisation, afin de se protéger contre un attaquant qui placerait sa propre adresse dans la liste des localisateurs possibles. [NOR 09] propose deux options pour cela, dont une seule (CGA) convient également dans un contexte de mobilité. Les CGAs (Cryptographically Generated Addresses) [AUR 05, AUR 03] sont des adresses générées de telle manière que le correspondant puisse vérifier l'origine des messages reçus par cette adresse. L'intérêt majeur des CGAs est de permettre l'usage de signatures RSA, sans avoir recours à des certificats, car l'information de clé est contenue dans l'adresse elle-même.

Dans l'état actuel de la définition du protocole, le seul élément pouvant déclencher un changement d'adresse de localisation est la détection d'une panne. Plus loin dans cet article, nous proposerons un autre déclencheur de changement de localisateur, à savoir le mouvement d'un hôte. Pour la détection de pannes, un protocole spécifique associé à Shim6 a été conçu : REAP [ARK 08]. REAP remplit également une fonction de recherche de chemin opérationnel. Une panne est détectée lorsqu'un temporisateur configurable (T_{send}) expire. Une procédure d'exploration est alors initiée. Celle-ci consiste à envoyer des sondes sur les chemins connus, chaque chemin correspondant à une paire d'adresses donnée. Lorsqu'un chemin opérationnel est trouvé, la paire de localisateur devient courante, et Shim6 commence à effectuer la réécriture d'adresses vers ces localisateurs. REAP est capable de détecter des chemins unidirectionnels. Par exemple si la paire de localisateurs A1-B1 permet de joindre B, mais B1-A1 ne permet pas de joindre A, Une exploration REAP peut mener à la sélection de A1-B1 dans un sens, B2-A2 dans l'autre, comme localisateurs courants. Enfin, REAP a été étudié pour minimiser le trafic de contrôle nécessaire à son fonctionnement [ARK 08]. L'efficacité de REAP a été évaluée dans [OLI 07] et [BAR 07b].

Limitations de Shim6 : Dans un contexte de mobilité, la limitation principale de Shim6 est l'absence de support de déplacement simultané de deux nœuds en communication. En effet, dans ce cas les deux hôtes Shim6 perdent leur adresse courante pour en recevoir une nouvelle. Chacun essaye d'envoyer des sondes REAP vers l'adresse devenue invalide du correspondant. Un point de rendez-vous est nécessaire dans ce cas. C'est le rôle joué par le Home Agent de MIPv6. Une seconde limitation de Shim6 est l'absence de système de détection de mouvement. Bien sûr, suite à un mouvement, l'adresse du réseau précédent n'est plus utilisable, ce qui est interprété par Shim6 comme une panne. Celui-ci peut donc bien rétablir une communication après mouvement (non simultané), mais le rétablissement est potentiellement long. Il est possible de régler le temporisateur de détection de pannes REAP à une valeur faible, mais cela génère beaucoup de trafic de contrôle REAP **de bout en bout**, pour assurer une détection très rapide des pannes. Ici encore, MIPv6 apporte une solution,

	MIPv6	Shim6	MipShim6
Optimisation de routage	V ₋	V ⁺	V ⁺
Détection de mouvement	V ⁺	V ₋	V ⁺
Support de mouvement simultané	V	X	V
Détection de pannes	X	V	V
exploration de chemins	X	V	V

V/X : Fonction supportée/non-supportée

V⁺/V₋ : V⁺ indique un support plus efficace.

Tableau 1. Fonctionnalités supportées par Mipv6, Shim6 et MipShim6

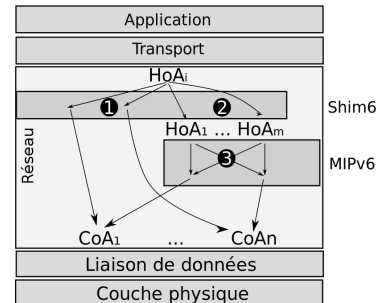


Figure 1. Architecture de la pile MipShim6

car il a été conçu pour détecter les mouvements sur base d'informations locales, donc sans générer de trafic de contrôle à travers l'Internet.

4. Une approche unifiée : MipShim6

L'objectif premier de notre architecture est d'associer les forces des protocoles MIPv6 et Shim6, au sein d'un ensemble gérant de manière souple et efficace mobilité et multi-domiciliation. Bagnulo et al. [BAG 07] ont déjà dessiné les bases d'une telle architecture. MipShim6 va plus loin encore en identifiant les éléments les plus efficaces de chaque protocole, jusqu'à supprimer certains éléments de l'un et de l'autre pour offrir un support modulaire aux applications. Le tableau 1 met en évidence les fonctions supportées par MIPv6, Shim6 et MipShim6, montrant l'intérêt de l'unification de ces technologies. L'optimisation de routage et la détection de mouvement sont supportées à la fois par MIPv6 et Shim6, mais Shim6 (qui n'utilise que des chemins directs) négocie plus efficacement l'usage d'un nouveau chemin. À l'inverse MIPv6 a bénéficié d'un grand travail de recherche dans le domaine de la détection de mouvement [KRI 07, MON 06, DAL 03], et le réalise donc plus efficacement que Shim6, qui considère les mouvements comme des pannes. Les mouvements simultanés de deux hôtes ne sont supportés que par MIPv6 actuellement, à cause de la nécessité de disposer d'un point de rendez-vous (l'agent mère), inexistant dans Shim6. Enfin, Shim6 est capable de surveiller un chemin "de bout en bout", et tenter de trouver une autre paire d'adresses fonctionnelle en cas de panne du chemin courant. Cela permet notamment de supporter les pannes d'agent mère, par simple basculement vers un autre agent. Dans cette section, nous décrivons notre architecture, ainsi que la manière dont chaque fonctionnalité est supportée par MipShim6.

4.1. Architecture de MipShim6

L'architecture que nous proposons est illustrée dans la figure 1. Afin de garantir la stabilité des identifiants de couche transport et application, une seule adresse leur est présentée. Comme le recommande également [BAG 07], l'adresse utilisée à ce niveau est une adresse mère, en raison de son

caractère plus durable que des CoAs. Dans cet article, nous appelons cette adresse un ULID (Upper Layer Identifier), pour utiliser la terminologie Shim6. Si un mobile dispose de plusieurs adresses mères, n'importe laquelle peut servir d'ULID, notre architecture garantissant la continuité des communications indépendamment de l'ULID choisi.

Tout paquet quittant la couche transport passe en premier lieu par la sous-couche Shim6, qui fournit un service de traduction d'adresses de bout en bout. Le service Shim6 est illustré à la figure 1, et présente deux composantes, numérotées **1** et **2** dans la figure. La première est une réécriture du ULID (qui est une adresse mère) en l'une des adresses care-of découvertes dans le réseau visité. Dans cet article, nous appelons ce type de traduction **Shim6-RO**, car l'usage immédiat d'une adresse care-of sans passer par MIPv6 permet d'utiliser un chemin direct vers le correspondant, comme le ferait la RO de MIPv6, si elle était utilisée.

La seconde composante de la figure 1 est une réécriture du ULID en l'une des adresses mères disponibles. Cela n'est utile que lorsque le nœud mobile s'est déplacé en même temps que son correspondant, et doit passer par le point de rendez-vous (l'agent mère) pour parvenir à le joindre (voir figure 2). Dans ce cas deux transformations successives s'opèrent. D'abord, Shim6 remplace l'adresse mère utilisée comme ULID par celle qui est utilisée comme localisateur (si nécessaire). C'est le cas de MN dans la figure 2, qui remplace HoA_1 par HoA_2 , l'agent mère HA_1 étant en panne. Ensuite le paquet est transmis à la sous-couche MIPv6, qui encapsule le paquet pour qu'il soit envoyé vers l'agent mère. La traduction par Shim6 d'une adresse mère en une autre permet à la fois de supporter la multi-domiciliation du réseau mère et la panne d'un agent mère.

Enfin la troisième composante est la couche MIPv6. Elle est principalement responsable de la gestion des sauts simultanés, mais aussi de la détection du mouvement et de l'initialisation d'une communication, lorsqu'un contexte Shim6 n'a pas encore été établi. MIPv6 assure l'encapsulation des paquets pour l'envoi vers le nœud correspondant en passant par l'agent mère. Comme le précise le standard [JOH 04], il est possible pour MIPv6 de maintenir plusieurs CoAs, bien qu'une seule d'entre elles puisse être enregistrée auprès de l'agent mère. Cette possibilité permet à MIPv6 de tenter d'enregistrer une autre CoA en cas d'échec d'enregistrement de l'une d'entre elles.

Shim6 peut déclencher un changement de localisateur suite à une panne détectée par REAP, ou suite à une notification de mouvement, en provenance de MIPv6. S'ensuit alors une exploration REAP, qui consiste à envoyer des sondes sur chacun des chemins, les CoAs étant préférées puisqu'elles permettent un routage plus direct. Un exemple montrant les messages échangés lors d'un basculement en Shim6-RO après mouvement est donné à la figure 3 : Shim6 attend l'établissement du tunnel avant d'envoyer un message *UR* au correspondant. Le nœud distant devra cependant encore faire un test de joignabilité par sondes REAP, avant de faire basculer le flux de données sur le nouveau chemin.

Support d'adresses mères multiples : Des adresses mères multiples peuvent être attribuées à un nœud mobile pour deux raisons principales. La première est que le réseau mère peut être multi-domicilié, et recevoir plusieurs préfixes IPv6. Dans ce cas chaque préfixe sert à la génération d'une nouvelle adresse mère. La seconde réside dans l'utilisation d'agents mères redondants. Un préfixe est attribué par agent mère de telle manière que l'hôte puisse choisir (grâce à la couche Shim6) par quel agent passer. Cela est utile pour permettre une tolérance aux pannes d'agents mères, ainsi qu'une répartition de la charge entre agents.

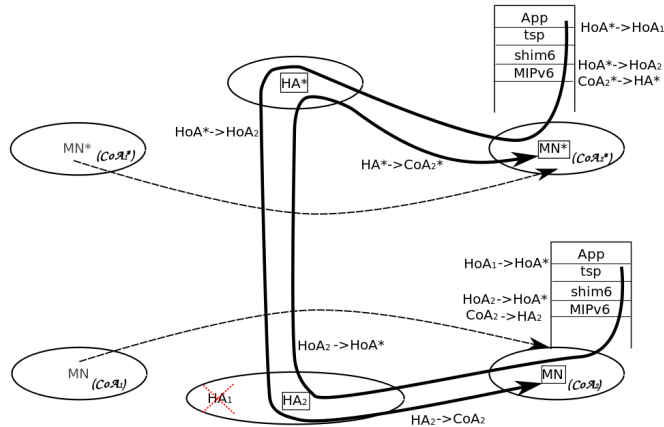


Figure 2. Scénario de saut simultané avec MipShim6

Support de CoAs multiples : L'usage de CoAs multiples permet le support de multi-domiciliation du réseau visité. [WAK 09] propose une extension pour supporter les CoAs multiples, mais celle-ci est rendue inutile par le support intégré à Shim6.

4.2. Mouvement avec MipShim6

Un mouvement se caractérise par une perte de la CoA courante, et l'acquisition d'une nouvelle CoA. MIPv6 emprunte des techniques génériques d'IPv6 pour détecter les mouvements, et ensuite effectuer un échange BU/BA avec l'agent mère lors de l'acquisition d'une nouvelle CoA. Dans notre architecture, nous proposons de réutiliser la détection de mouvement de MIPv6. Ainsi, il nous est possible de réutiliser des optimisations de la détection de mouvements déjà développées pour MIPv6 [MON 02, DAL 03].

Suivant que le nœud mobile se trouve ou non en mode Shim6-RO au moment du mouvement, nous distinguons deux cas de figure. Si il ne se trouve pas en mode Shim6-RO, le tunnel est simplement mis à jour, et Shim6 peut recevoir une notification de déplacement, lui permettant de déclencher un basculement immédiat en Shim6-RO. Dans le deuxième cas de figure, le nœud mobile s'est déplacé alors qu'il se trouvait en mode Shim6-RO. MIPv6 met aussi à jour son tunnel, bien qu'il soit inutilisé. Il envoie comme dans le cas précédent une notification à Shim6, qui peut changer de localisateur pour passer en Shim6-RO.

Nous remarquons par ailleurs qu'il est possible que Shim6 détecte le mouvement avant MIPv6 ou en même temps. Cela est possible lorsque le temporisateur de détection de pannes est fixé à une valeur inférieure au temps nécessaire à MIPv6 pour terminer son opération de détection de mouvement et basculement. Dans ce cas, le mouvement est en fait perçu comme une panne par Shim6, qui commence immédiatement à envoyer des sondes. Celles-ci seront infructueuses jusqu'à ce que MIPv6 ait terminé de mettre à jour son tunnel avec l'agent mère. Bien que ce cas mène au même résultat que celui du second scénario, il est préférable de l'éviter car il génère plus de trafic de contrôle. Cela est possible par un réglage du temporisateur de détection de pannes Shim6 à une valeur supérieure à la durée moyenne de gestion de déplacement par MIPv6.

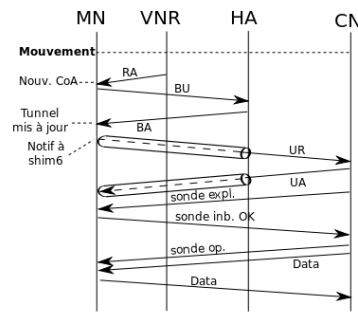


Figure 3. Séquence de messages lors d'un mouvement

4.3. Atouts de l'architecture

Saut simultané : Alors que l'agent mère n'est plus indispensable en cas de mouvement de l'un des nœuds individuellement (grâce à la capacité de Shim6 de basculer en échangeant des messages par la nouvelle CoA directement), il reste nécessaire en cas de mouvement simultané. Dans ce cas, les deux nœuds vont mettre à jour leurs tunnels respectifs une fois la nouvelle CoA acquise, et ensuite envoyer une notification à Shim6 pour qu'il tente de basculer en Shim6-RO. Les deux correspondants s'échangent alors des mises à jour de localisateurs Shim6 à travers les tunnels MIPv6, et basculent finalement en Shim6-RO.

Optimisation de routage : L'une des principales contributions de notre architecture est que nous proposons de supprimer complètement la RO (Routing Optimization) MIPv6. Shim6 dispose en effet de mécanismes plus légers qui permettent de remplacer complètement la RO standard, et même la RO optimisée [ARK 07].

5. Implémentation et évaluation

L'une de nos contributions majeures est la création d'une implémentation de MipShim6. Celle-ci est issue de l'association des implémentations LinShim6 [BAR 07a] et UMIP [MIY 04]. LinShim6 a été conçu dans l'objectif d'être à terme compatible avec MIPv6, et la même architecture (XFRM) a donc été utilisée.

Notre architecture utilise CGA comme technique de sécurisation de l'échange d'adresses. UMIP quant à lui utilise le protocole Neighbour Discovery dans la détection de mouvement, ainsi que dans la gestion du retour au réseau mère, après déplacement dans un réseau visité (qui suppose des dispositions particulières, décrites dans [JOH 04]). Pour cette raison nous donnons le contrôle complet de la configuration d'adresses au démon `mip6d`, et désactivons l'autoconfiguration du kernel. Nous modifions UMIP en lui annexant les bibliothèques CGA de LinShim6, de manière à ce qu'il autoconfigure directement des adresses CGA. LinShim6 est quant à lui modifié pour survivre dans un environnement mobile. En particulier, un contexte Shim6 peut dans MipShim6 être maintenu par le démon `shim6d`, alors qu'aucun localisateur n'y est stocké. Cela est nécessaire pour supporter des périodes transitoires de mouvement où aucune adresse n'est attribuée. La partie noyau constitue le point de rencontre de MIPv6 et Shim6. Le framework `xfrm`, sur lequel les patches sont basés, a pour principe qu'un paquet sortant traverse une couche de décision (*xfrm policy*), dans laquelle le chemin d'un paquet à travers la couche réseau est déterminé. Ce chemin consiste en une suite de 0, 1 ou plusieurs *transformateurs* (d'où le nom `xfrm`) par lesquels le paquet devra passer. Dans notre cas, les transformateurs sont MIPv6, shim6, les deux ou aucun des deux en fonction du contrôle imposé par les démons `mip6d` et `shim6d`. Si les deux démons imposent au noyau une décision concernant le même flux, la couche de décision d'`xfrm` enregistre que le paquet devra passer par les deux transformateurs, en commençant par shim6. Le traitement des paquets entrants est plus simple, car il est gouverné par une lecture séquentielle des extensions d'en-tête du paquet. Ainsi, un paquet encapsulé se verra décapsulé, et un paquet muni de l'extension Shim6 sera dirigé vers les fonctions de réécriture Shim6 dans le noyau. L'implémentation de MipShim6 est disponible sur demande aux auteurs.

Evaluation : Notre objectif est de mettre en exergue les apports de notre solution aussi bien par rapport à une solution de multi-domiciliation (Shim6) que par rapport à une solution de mobilité

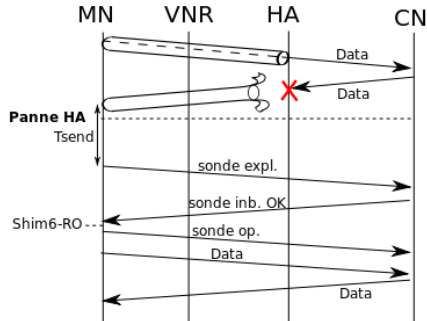


Figure 4. Scénario 1 : panne de l'agent mère

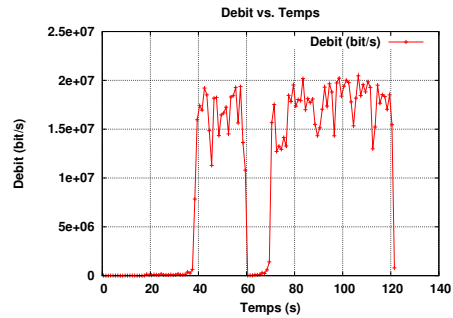


Figure 5. Évolution du débit de TCP avant et après une panne de l'agent mère en fonction du temps

(MIPv6). Généralement, ces apports ne s'expriment pas en temps de latence plus courts, mais en support possible par rapport aux protocoles traditionnels qui ne gèrent pas ces fonctionnalités. Notre évaluation se base sur deux scénarios distincts. Alors que le premier scénario met en avant les bénéfices majeurs de notre solution par rapport à une solution de mobilité simple (tolérance aux pannes), le second présente son bénéfice par rapport à une solution de multi-domiciliation telle que Shim6 (support du mouvement simultané).

L'objectif du premier scénario est de démontrer que notre architecture de gestion unifiée de mobilité et de multi-domiciliation est tolérante aux pannes éventuelles du réseau mère. Ce premier scénario est basé sur la plate-forme représentant un modèle typique du déploiement de MIPv6 :

- un mobile enregistré auprès de l'agent mère et communiquant avec un nœud correspondant. Le réseau mère est muni d'un accès sans-fil 802.11g à 54Mbit/s.
- un réseau visité muni d'un accès sans-fil 802.11g à 54Mbit/s.

Initialement le mobile est dans le réseau visité et établit une connexion TCP avec le nœud correspondant. Le trafic TCP est généré par le téléchargement d'un fichier depuis le nœud correspondant avec l'application *secure copy* (scp). Le mobile n'activant pas l'optimisation de routage (RO), le trafic échangé entre le mobile et le nœud correspondant passe obligatoirement par l'agent mère via un tunnel. Le téléchargement du fichier dure 120s. Au bout de 60s nous provoquons la panne de l'agent mère. Lorsque le temporisateur Tsend expire, REAP détecte la panne de l'agent mère et déclenche une exploration sur l'ensemble des adresses du nœud. L'exploration de l'adresse care-of étant concluante, Shim6 redirige le trafic sur cette adresse et passe en mode Shim6-RO (voir Fig. 4). Afin d'évaluer les performances de notre solution dans ce scénario, nous nous intéressons à deux supports d'observation. Le premier concerne l'évolution du débit de la session TCP avant et après la panne de l'agent mère. Le second est le temps de récupération des applications (ou Application Recovery Time (ART)) après la panne de l'agent mère. Nous reprenons la définition de l'ART par [OLI 07] comme le temps écoulé entre le dernier segment TCP reçu avant la panne et le premier segment TCP reçu après la panne.

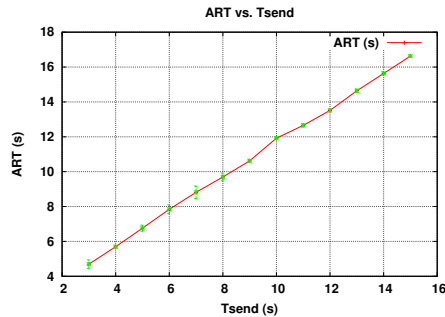


Figure 6. Temps de récupération de l'application après une panne de l'agent mère en fonction du timer T_{send}

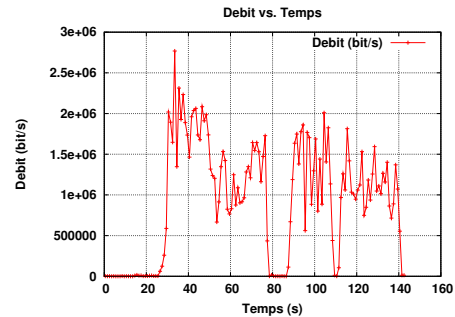


Figure 7. Évolution du débit de TCP avant et après un saut simultané en fonction du temps

La courbe présentée dans la Figure 5 illustre l'évolution du débit de TCP en fonction du temps. On observe que le débit remonte bien à sa valeur initiale après le temps nécessaire à l'exploration REAP. Il devient même légèrement plus élevé, parce que le flux TCP utilise le chemin direct vers le correspondant après la panne, alors qu'il passait par le tunnel avant. Nous pouvons également noter que si MIPv6 avait été utilisé seul, nous n'aurions pas observé la reprise du trafic entre les deux correspondants.

La seconde mesure réalisée dans ce scénario est l'ART. L'ART dépend de la valeur de T_{send} choisie car REAP découvre la panne de l'agent mère après l'expiration de ce temporisateur. Par ailleurs, dans notre solution, le temporisateur de retransmission de TCP (RTO) est remis à zéro lorsque REAP termine son exploration. Cette optimisation (proposée par [OLI 07] dans le cas de la multi-domiciliation) réduit le temps de récupération de l'application. La courbe présentée à la Figure 6 illustre l'évolution de l'ART en fonction de T_{send} . Nous remarquons que l'ART suit une fonction linéaire ($ART \simeq T_{send} + 1.69s$). Ce résultat confirme que le temps de récupération dépend principalement de T_{send} , à une constante près, qui représente principalement la durée d'exploration.

L'objectif du deuxième scénario est de mettre en évidence que notre architecture de gestion unifiée de mobilité et de multi-domiciliation gère le saut simultané. Nous nous basons sur une plateforme de test comprenant les éléments suivants :

- deux nœuds mobiles A et B disposant du même agent mère
- quatre réseaux visités ayant un accès sans-fil 802.11g à 54Mbit/s, notés VN1, VN2, VN3, VN4

Initialement, A se trouve dans VN1 et B dans VN2. Nous n'activons aucune optimisation de routage, tous les messages échangés entre mobiles passent obligatoirement par l'agent mère à travers des tunnels. Nous générons le même trafic que dans le premier scénario. Nous provoquons un déplacement simultané des mobiles : A passe dans VN3, et B dans VN4. Après ce déplacement simultané, chaque mobile obtient une nouvelle adresse care-of et met à jour son Binding Cache (MIPv6) avec l'agent mère (échange de BU/Back). Ensuite, chaque mobile doit mettre à jour le contexte Shim6 de son correspondant. Étant donné que nous ne mettons pas de RO en place, le localisateur préféré de chaque mobile est sa HoA. Ensuite, chaque mobile envoie un UR (*Update Request*) à

son correspondant via l'agent mère et ainsi les *Update Requests* sont envoyés via des tunnels aux nouvelles localisations des mobiles (VN3 et VN4). La courbe présentée dans la Figure 7 illustre l'évolution du débit TCP en fonction du temps dans le cas d'un saut simultané. Cette courbe montre que notre architecture gère le saut simultané car l'échange TCP reprend bien après le mouvement des deux mobiles. Si Shim6 avait été utilisé seul, la connexion aurait été rompue.

6. Conclusion

MIPv6 a fait l'objet de plusieurs propositions d'amélioration par le passé. Par exemple [WAK 09] ajoute le support d'adresses care-of multiples dans MIPv6, [ARK 07] propose une série de mécanismes destinés à rendre l'optimisation de routage MIPv6 plus efficace, et [FAI 05] propose une solution de redondance des Home Agent.

MipShim6, dont le concept de base est commun avec [BAG 07], supporte toutes ces fonctionnalités en une seule architecture sans aucune modification protocolaire à MIPv6 ou Shim6, comme le montre le tableau 1. De plus, MipShim6 remplace l'optimisation de routage de MIPv6 par une version plus légère et souple basée sur Shim6. Nous avons étendu les implémentations les plus avancées de MIPv6 et Shim6, respectivement UMIP et LinShim6, pour les rendre inter-opérables en accord avec notre architecture. Enfin, nous avons montré concrètement les atouts de l'architecture par des mesures en laboratoire.

Le travail futur dans ce domaine consistera à optimiser le fonctionnement de MipShim6 en lui associant des techniques efficaces de détection de mouvement, ainsi que des méthodes rapides de passage en Shim6-RO, pouvant concurrencer celles proposées dans [ARK 07]. Cela permettra d'associer en une architecture modulaire et complète les solutions avancées disponibles actuellement dans la littérature, mais encore isolées et non déployées.

7. Bibliographie

- [ARK 07] ARKKO J., VOGT C., HADDAD W., « Enhanced Route Optimization for Mobile IPv6 », RFC 4866 (Proposed Standard), mai 2007.
- [ARK 08] ARKKO J., VAN BEIJNUM I., « Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming », June 2008, Internet Draft, draft-ietf-shim6-failure-detection-13.txt, expired.
- [AUR 03] AURA T., « Cryptographically Generated Addresses (CGA) », *Proc. 6th Information Security Conference (ISC'03)*, vol. LNCS 2851/2003, Bristol, UK, October 2003, Springer, p. 29-43.
- [AUR 05] AURA T., « Cryptographically Generated Addresses (CGA) », RFC 3972 (Proposed Standard), mars 2005, Updated by RFCs 4581, 4982.
- [BAG 07] BAGNULO M., GARCIA-MARTINEZ A., AZCORRA A., « IPv6 Multihoming Support in the Mobile Internet », *IEEE Wireless Communications*, vol. 14, n° 5, 2007.
- [BAR 07a] BARRÉ S., BONAVENTURE O., « Implementing SHIM6 using the Linux XFRM framework », *Routing In Next Generation workshop*, Madrid, Spain, Dec 2007.
- [BAR 07b] BARRÉ S., BONAVENTURE O., « Improved Path Exploration in shim6-based Multihoming », *SIG-COMM 2007 Workshop "IPv6 and the Future of the Internet"*, 2007.

- [CHE 06] CHEN T., WENYU L., « A Novel IPv6 Communication Framework : Mobile SHIM6 (M-SHIM6) », *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*, November 2006, p. 1-3.
- [DAL 03] DALEY G., PENTLAND B., NELSON R., « Movement detection optimizations in mobile IPv6 », *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, septembre 28–octobre 1, 2003, p. 687–692.
- [DHR 08] DHRAIEF A., MONTAVONT N., « Toward Mobility and Multihoming Unification- The SHIM6 Protocol : A Case Study », *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, Las Vegas, April 2008, p. 2840-2845.
- [FAI 05] FAIZAN J., EL-REWINI H., KHALIL M., « Introducing reliability and load balancing in mobile IPv6-based networks », *Wirel. Commun. Mob. Comput.*, vol. 8, n° 4, 2005, p. 483–500, John Wiley and Sons Ltd.
- [JOH 04] JOHNSON D., PERKINS C., ARKKO J., « Mobility Support in IPv6 », RFC 3775 (Proposed Standard), juin 2004.
- [KIA 06] KIANI A., KHAN S., WENBING Y., « A Novel Mechanism to Support Session Survivability in Heterogeneous MIPv6 », *Emerging Technologies, 2006. ICET '06. International Conference on*, November 2006, p. 38-43.
- [KRI 07] KRISHNAN S., MONTAVONT N., NJEDJOU E., VEEREPALLI S., YEGIN A., « Link-Layer Event Notifications for Detecting Network Attachments », RFC 4957 (Informational), août 2007.
- [MIY 04] MIYAZAWA K., NAKAMURA M., « IPv6 IPsec and Mobile IPv6 Implementation of Linux », *Proceedings of the Linux Symposium*, vol. 2, July 2004, p. 371-380.
- [MON 02] MONTAVONT N., NOEL T., « Handover management for mobile nodes in IPv6 networks », *IEEE Communications Magazine*, vol. 40, n° 8, 2002, p. 38–43.
- [MON 06] MONTAVONT N., NOEL T., « Fast movement detection in IEEE 802.11 networks : Research Articles », *Wirel. Commun. Mob. Comput.*, vol. 6, n° 5, 2006, p. 651–671, John Wiley and Sons Ltd.
- [NOR 09] NORDMARK E., BAGNULO M., « Shim6 : Level 3 Multihoming Shim Protocol for IPv6 », Internet draft, draft-ietf-shim6-proto-12.txt, work in progress, February 2009.
- [OLI 07] DE LA OLIVA A., BAGNULO M., GARCIA-MARTINEZ A., SOTO I., « Performance Analysis of the REAchability Protocol for IPv6 Multihoming », *Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2007)*, Sept 2007.
- [PER 96] PERKINS C. E., JOHNSON D. B., « Mobility support in IPv6 », *MobiCom '96 : Proceedings of the 2nd annual international conference on Mobile computing and networking*, New York, NY, USA, 1996, ACM, p. 27–37.
- [RAH 08] RAHMAN M., ATIQUZZAMAN M., « SEMO6 - a multihoming-based seamless mobility management framework », *Military Communications Conference, 2008. MILCOM 2008. IEEE*, San Diego, November 2008, p. 1-7.
- [ROP 08] ROPITAU T., MONTAVONT N., « Implementation of a Flow Binding Mechanism », *the 4th IEEE PercomWorkshop on Pervasive Wireless Networking*, March 2008.
- [WAK 09] WAKIKAWA R., DEVARAPALLI V., TSIRTSIS G., ERNST T., NAGAMI K., « Multiple Care-of Addresses Registration », Internet draft, draft-ietf-monami6-multiplecoa-13.txt, work in progress, April 2009.