MASTER EN SCIENCES INFORMATIQUES

Support efficace du Multicast dans Mobile IP

Brouckaert, Xavier

Award date:
2002

FUNDP
Institut d'Informatique

Rue Grandgagnage, 21
B - 5000 NAMUR (Belgique)

Support efficace du Multicast dans Mobile IP

Xavier BROUCKAERT

Sous la direction de O. Bonaventure

Institut d'Informatique
Facultés Universitaires Notre-Dame de la Paix
Namur

Juin 2002

Résumé

Mobile IP et les réseaux sans-fil permettent d'accéder de façon véritablement mobile à Internet. Pourtant, les communications de groupe, appelées multicast, ne sont pas gérées efficacement par Mobile IP. Ce mémoire décrit d'abord IPv6, Mobile IP et les protocoles de routage multicast. Dans la seconde partie, nous examinons, comparons et critiquons huit solutions (ie. Bidirectionnal Tunneling, Local Membership, MoM, RBMoM, MMA, MMPuA, MobiCast and MSSMv6) existant actuellement pour fournir un service Multicast efficace aux hôtes mobiles. Le résultat de la comparaison est résumé dans un grand tableau montrant les paramètres intéressants de chaque solution. Finalement, nous en proposons une autre, basée sur l'expérience acquise grâce aux solutions existantes. Notre solution s'attache à réduire le handoff tout en fournissant un routage presque optimal et se concentre sur une architecture réseau à base de Mobile IPv6 et PIM-SM. Elle se distingue des autres par le support des scopes IPv6 et par certaines optimisations spécifiques pour les routeurs PIM-SM. Une nouvelle entité, le Mobile Multicast Router, capable de créer des tunnels temporaires avec d'autres Mobile Multicast Routers ou avec des noeuds mobiles, est le concept clé pour atteindre les objectifs désirés.

Abstract

Truly seamless mobile access to the Internet has become possible using Mobile IP and wireless networks. Yet, group communications, known as multicast, are not efficiently handled by Mobile IP. This paper first describes IPv6, Mobile IP and multicast routing protocols. In the second part, we examine, compare and criticize height existing solutions (namely Bidirectionnal Tunneling, Local Membership, MoM, RBMoM, MMA, MMPuA, MobiCast and MSSMv6) for providing multicast efficiently to mobile hosts. The result of the comparison is summarized in a big table, showing relevant parameters for each solution. Finally, we propose another solution, based on the experience acquired with the existing ones. Our solution aims at reducing handoff while providing near-optimal routing and focuses on a Mobile IPv6 / PIM-SM network architecture. It distinguishes from the others by supporting IPv6 scopes and specific optimizations for PIM-SM routers. A new entity called Mobile Multicast Router, able to create temporary tunnels between other Mobile Multicast Routers or Mobile Nodes, is the key for achieving desired goals.

Remerciements

Je voudrais d'abord remercier mon promoteur O. Bonaventure et Pierre Reinbold, pour leur aide, leur patience, leurs livres, leur vivacité de réponse par courrier électronique et leurs commentaires constructifs durant la rédaction du mémoire. Sans vous, ce mémoire n'aurait jamais pu être terminé à temps. Merci aussi de m'avoir laissé traiter le sujet de mémoire que j'avais proposé.

Merci à toute l'équipe Ipinoo de France Telecom R&D. J'ai passé un très bon stage de fin d'études en leur compagnie. C'est là que j'ai découvert IPv6 et Mobile IP. Sans ce stage, je n'aurais jamais été capable de rédiger ce mémoire. Merci pour tout ce qu'ils m'ont appris et pour leur bonne humeur durant les six mois.

Merci particulièrement à Luc Beloeil de France Telecom qui a relu certains chapitres et avec qui j'ai pu discuter de la solution proposée dans ce mémoire.

Merci à ma famille, à Ariane et à mes amis qui m'ont soutenu durant ces derniers mois.

Merci d'avance à tous ceux qui liront ce document ou qui viendront assister à ma présentation !

Xavier Brouckaert

Table des matières

I Les protocoles existants **19**

1 IPv6 **21**
1.1 Introduction 21
1.2 Adressage 21
1.2.1 Adresses unicast 22
1.2.2 Adresses multicast 23
1.2.3 Adresses anycast 24
1.3 Simplification du protocole IP 24
1.4 ICMPv6 24
1.5 Configuration automatique 25
1.6 Sécurité 25
1.7 Critique 26

2 Mobilité **29**
2.1 Introduction 29
2.2 Problème 29
2.3 Mobile IPv4 31
2.3.1 Mode avec Foreign Agent 31
2.3.2 Mode sans Foreign Agent 32
2.3.3 Le tunnel inverse 33
2.3.4 Optimisation de route 33
2.3.5 Smooth handoff 33
2.4 Mobile IPv6 34
2.4.1 Exemple 36
2.5 Dynamic Mobile IPv6 - DMI 36
2.5.1 Avantages 39
2.5.2 Inconvénients 39

3 Multicast **41**
3.1 Introduction 41
3.1.1 Arbres de distribution multicast 41
3.1.2 *Source specific multicast* versus *Any source multicast* 42
3.1.3 Inscription et résiliation à un groupe 43
3.1.4 Dense mode et Sparse mode 45
3.2 Protocoles de type Dense 45
3.2.1 DVMRP 45
3.2.2 PIM-DM 46
3.3 Protocoles de type Sparse 47

3.3.1 CBT 47
3.3.2 PIM-SM 48
3.3.3 PIM-SSM 49

II Support du multicast pour les hôtes mobiles 51

4 Problème 53
4.1 Réception 53
4.2 Émission 54

5 Revue des solutions existantes 57
5.1 Introduction 57
5.2 Remote Subscription - Bidirectionnal Tunneling 57
5.2.1 Réception de trafic multicast 57
5.2.2 Emission de trafic multicast 58
5.2.3 Critique 59
5.3 Mobile Multicast Protocol - MoM 60
5.3.1 Critique 61
5.4 Local Subscription - Remote Membership 62
5.4.1 Réception de trafic multicast 62
5.4.2 Emission de trafic multicast 62
5.4.3 Critique 63
5.5 Mélange de Local et Remote Membership 63
5.6 Range Based Mobile Multicast - RBMoM 63
5.6.1 Introduction 63
5.6.2 Fonctionnement du protocole 64
5.6.3 Critique 65
5.7 Multicast by Multicast Agent - MMA 65
5.7.1 Description 65
5.7.2 Critique 66
5.8 MobiCast 67
5.8.1 Description du protocole 67
5.8.2 Emission de trafic multicast 68
5.8.3 Réception de trafic multicast 68
5.8.4 Fast Handoff 69
5.8.5 Signalement des pertes au MN 69
5.8.6 Critique 69
5.9 Mobile SSM Sources for IPv6 - MSSMv6 70
5.9.1 Critique 71
5.10 Mobile Multicast Protocol using Anycast - MMPuA 72
5.10.1 Description 72
5.10.2 Critique 74
5.11 Tableau comparatif 75

6 Nouvelle proposition 81
6.1 Introduction 81
6.2 Présentation de la solution 81
6.2.1 Définition de Mobile Multicast Router 81

6.2.2 Note préliminaire sur la sécurité 82
6.2.3 Nouveaux messages 82
6.2.4 Modifications des protocoles existants 83
6.2.5 Informations à maintenir 84
6.2.6 Fonctionnement du protocole pour la réception 85
6.2.7 Exemple 87
6.2.8 Fonctionnement du protocole pour l'émission 89
6.3 Discussion 93

Liste des tableaux

5.1 Comparatif : Bidirectionnal Tunneling et Local Subscription 76
5.2 Comparatif : MoM et RBMoM 77
5.3 Comparatif : MMA et MMPuA 78
5.4 Comparatif : Mobicast et MSSMv6 79

6.1 Comparatif : nouvelle solution 94

Table des figures

1.1 Modes de transmission en IPv6 22
1.2 Plan d'adressage agrégé pour les adresses unicast 23
1.3 Adresses multicast 23
1.4 A gauche : l'en-tête IPv4, à droite : l'en-tête IPv6 24
1.5 Structure d'un message ICMPv6 25
1.6 Les différents messages ICMPv6 28

2.1 Mobile communiquant avec un CN avant déplacement 30
2.2 Le mobile garde son adresse 30
2.3 Le mobile change d'adresse 31
2.4 Mobile IPv4 - Avec Foreign Agent 32
2.5 Mobile IPv4 - Sans Foreign Agent 33
2.6 Mobile IPv4 - Optimisation de route 34
2.7 Mobile IPv6 35
2.8 Mobile IPv6 - Binding Update 36
2.9 Mobile IPv6 - Tunnel entre le HA et MN3 36
2.10 Mobile IPv6 - Datagramme de MN3 vers MN2 contenant une option de destination Home Address 37
2.11 Mobile IPv6 - Datagramme de MN2 vers MN3 contenant un Routing Header . 37

3.1 Arbre RPT (*,G) (à gauche) et Arbre SPT (S,G) (à droite) 42
3.2 IGMP v1 43
3.3 Construction d'un arbre distribué avec un mécanisme d'inondation 46
3.4 L'arbre distribué après élagage des branches inutiles 47

4.1 Echec du RPF Check quand le mobile garde son adresse 55

5.1 Double tunnel dans Mobile IPv4 avec Foreign Agent 58
5.2 Datagramme multicast encapsulé dans un datagramme unicast à destination du Home Agent 58
5.3 Problème des tunnels parallèles 59
5.4 Problème de convergence des tunnels des HA vers le même FA 60
5.5 MoM - Le FA désigne HA1 comme DMSP 61
5.6 MoM - Boucle dans le routage des datagrammes 62
5.7 RBMoM 64
5.8 MMA 66
5.9 MobiCast - Forwarding multicast du DFA aux BS dans la même DVM 68
5.10 MSSMv6 72

6.1 Structure de la Multicast Sending List 84

6.2 Contexte 87
6.3 Dans le Home Network 88
6.4 Passage du Home au premier Foreign Network 88
6.5 Stabilisation du protocole dans le premier Foreign Network 89
6.6 Passage du premier au deuxième Foreign Network 90
6.7 Stabilisation du protocole dans le deuxième Foreign Network 90
6.8 Situation finale 91

Introduction

Depuis quelques années, les communications sans-fil connaissent un succès grandissant. Pratiquement plus personne ne peut se passer de son GSM. Le téléphone sans-fil, c'est la liberté de communiquer où et quand nous le voulons.

Parallèlement, Internet a continué à se développer, à s'introduire dans toutes les couches de la population et à offrir de plus en plus de services. La suite logique après la téléphonie est donc d'arriver à un Internet sans-fil ou *Internet mobile*. Bien entendu, le coeur du réseau fonctionnera toujours de façon filaire car les ondes radio ne permettent pas encore d'atteindre de très hauts débits. C'est du côté de l'accès à Internet que l'on trouvera des bornes radio permettant aux internautes de devenir mobiles.

La grande mode actuellement est d'accéder à Internet via un réseau sans-fil. Cela peut se faire de diverses manières : au moyen d'un GSM WAP (bientôt GPRS et UMTS), d'un PC portable ou d'un PDA équipé d'une carte réseau sans-fil et connecté grâce aux technologies radio Bluetooth ou Wifi (802.11).

Les réseaux locaux sans-fil (WLAN) possèdent certainement l'avantage d'offrir une certaine mobilité, restreinte par la portée des stations émettrices. Néanmoins, il n'est pas possible de profiter d'un roaming parfait comme c'est le cas pour les GSM. Dans une situation idyllique, un employé de bureau se connecterait à Internet via le réseau filaire rapide de son entreprise. Sa journée terminée, il changerait de réseau de façon totalement transparente en s'attachant par exemple au réseau sans-fil de sa ville ou des transports en commun. Rentré chez lui, il pourrait continuer à travailler ou se divertir en profitant de sa propre connexion Internet, sans avoir eu une seule fois conscience qu'il avait changé de réseau. Il n'aurait nul besoin d'arrêter ses applications favorites (mail, messager instantané, ...) à chaque changement de réseau. Ce rêve sera bientôt accessible grâce au protocole Mobile IP développé à l'IETF. Sans Mobile IP, l'employé perdrait inévitablement sa connexion Internet à chaque changement de réseau et il devrait reconfigurer tous les paramètres réseau propres au nouveau réseau d'attache.

Mobile IP ne se préoccupe actuellement que des communications point à point (unicast), or il y a un besoin grandissant de communications multipoint, c'est-à-dire entre plusieurs personnes. Les nouveaux services qui requièrent ce type de communication sont par exemple la vidéoconférence, les radios et télévisions en streaming, le partage des fichiers en communauté, etc. Leur plus gros défaut est sans aucun doute la qualité : mauvaise qualité et taille microscopique de l'image, coupures fréquentes, ...Le principal problème est le manque de bande passante disponible pour faire circuler des flux de meilleure qualité. En effet, supposons que dix mille personnes regardent la même vidéo en streaming provenant d'un unique serveur (par exemple, les attentas du 11 septembre 2001 sur `cnn.com`). Il y aura dix mille flux distincts transitant sur le réseau. Comme la bande passante disponible n'est pas infinie, les milliers de

flux sont ralentis et cela résulte côté client en une vidéo saccadée de très mauvaise qualité. De plus, les performances globales du réseau sont largement déteriorées.

La solution à ce problème a été inventée depuis bien longtemps par S. Deering. Il suffit de construire un arbre logique de distribution entre la source et tous les récepteurs et de propager un seul et unique flux le long de cet arbre. De cette façon, peu importe le nombre de clients, la bande passante consommée est toujours la même. Les protocoles de routage qui permettent de réaliser les communications multipoint efficacement sont appelés protocoles Multicast.

L'exemple précédent était simplifié par l'hypothèse désormais révolue que tous les clients étaient fixes (par exemple, ce sont des PC de bureau). Supposons maintenant que ces dix mille clients se déplacent et que certains changent de réseau. Chaque client mobile souhaite certainement regarder sa vidéo sans accrocs ni coupures, peu importe le type de communication (unicast ou multicast) et le réseau auquel il est attaché pour le moment. Même s'il peut paraître étrange à première vue de regarder une vidéo tout en se déplaçant, il se peut très bien que la personne qui la regarde ne se déplace pas physiquement. Elle peut par exemple être assise dans un bus, un train ou un avion et regarder la vidéo sur son portable (ou sur un GSM évolué) connecté à Internet.

Ce que le client souhaite, c'est exactement l'objectif de mon mémoire : permettre aux hôtes mobiles d'effectuer des communications multicast.

Le challenge n'est pas aussi simple qu'il y paraît. Il existe une grande diversité d'appareils, de moyens de communications, de protocoles de routage unicast et multicast. Le commun dénominateur, c'est l'Internet Protocol (IP) qui permet à deux équipements qui l'implémentent de se comprendre quels que soient la technologie sans-fil, le fournisseur d'accès ou tout autre paramètre. IP brille par sa simplicité et son universalité. C'est la raison pour laquelle il doit être le coeur de l'architecture des telecoms d'aujourd'hui et de demain. Nokia ne s'y est pas trompé. Cette firme mise actuellement sur le *Tout-IP*. IP permettra de casser les frontières traditionnelles entre les réseaux actuels fixes et mobiles, de réduire les coûts, de gagner en flexibilité, d'offrir les mêmes services partout et surtout de générer de nouveaux bénéfices [37]. Nokia prédit aussi que le nombre d'utilisateurs mobiles triplera d'ici 2005 et que le trafic généré par chacun doublera, ce qui conduira à une augmentation d'un facteur 6. Comme les profits générés par la voix (la téléphonie simple) diminueront, beaucoup d'opérateurs se tourneront vers l'offre de nouveaux services. Rapidement, il deviendra coûteux de maintenir un réseau pour la téléphonie et un autre pour les données. La seule solution sera donc de passer à un réseau unique transportant la voix et les données en utilisant IP.

Malheureusement, la version actuelle d'IP, IPv4, commence à se faire vieille. Le fameux protocole inventé il y a plus de vingt ans est victime de son succès. Ses inventeurs n'auraient pas pu prévoir le nombre astronomique d'équipements qui l'utiliseraient et les multiples services qui en découleraient. Ceci étant, le nombre d'adresses IPv4 disponibles actuellement ne suffit plus. En prévision de ces problèmes, l'IETF développe depuis 12 ans une nouvelle version d'IP, nommée IPv6. IPv6 apporte son lot de nouveautés (la plus importante étant la taille quatre fois plus grande des adresses) et quelques inconvénients principalement dûs à la transition nécessaire entre IPv4 et IPv6. Le challenge consiste donc aussi à explorer la manière dont des mobiles IPv4 ou IPv6 pourraient effectuer des communications multicast.

Comme je l'ai déjà décrit dans mon exemple, la multiplicité des technologies et des protocoles doit être complètement transparente du point de vue de l'utilisateur. Or, tous les réseaux

sont différents : certains proposent déjà le Multicast, d'autres pas encore. Il s'agit donc de gérer toutes ces contraintes le plus efficacement possible.

Les programmeurs aussi ont besoin de transparence pour la réalisation d'applications multicast. Peu importe qu'un partenaire de la communication se déplace, qu'un nouveau se joigne ou qu'un autre parte, il faut que toute cette dynamique soit gérée de façon efficace et transparente dans la couche réseau. Une bonne séparation des couches a toujours conduit à des architectures efficaces et puissantes. Il faut donc que le support de la mobilité et du multicast se fassent au niveau de la couche réseau, ce qui implique que l'interaction entre les deux doit aussi se situer dans la couche réseau.

Comme on peut le constater, un objectif tenant en une phrase mène à beaucoup de problèmes de réalisation. L'envers du décor recèle des dizaines de sous-problèmes que je vais mettre en lumière dans ce document.

Ce mémoire se compose de deux parties telles que la première établit les fondations pour la deuxième. J'expliquerai tout d'abord brièvement ce qu'est IPv6, en épinglant les différences avec son prédécesseur IPv4. Ensuite, je décrirai les protocoles de mobilité standardisés à l'IETF, c'est-à-dire Mobile IPv4 et Mobile IPv6. J'en profiterai pour parler de Dynamic Mobile IP, une optimisation de Mobile IP que j'ai implémentée durant mon stage à France Telecom R&D. Il ne restera plus qu'à expliciter les protocoles de routage multicast existant actuellement pour clore cette première partie.

La deuxième partie, le support du multicast pour les hôtes mobiles, est le coeur du mémoire. Après avoir posé le problème (décrit pour la première fois en 1996 par Acharya et Badrinath [1]), je détaillerai et critiquerai huit solutions existant actuellement. Elles ont été développées ces dernières années par des chercheurs du monde entier et fournissent des idées nouvelles ou à tout le moins des améliorations d'autres solutions. Je terminerai par un tableau comparatif des huit solutions permettant de saisir les points forts et les points faibles de chacune.

Le dernier chapitre décrit la solution à laquelle je suis arrivé en collaboration avec Luc Beloeil de France Telecom et Olivier Bonaventure, sur base du chapitre précédent. Il ne s'agit pas d'une solution miracle, mais plus d'une idée réalisable grâce aux technologies qui ont le plus de chances d'être déployées prochainement. Comme les autres auteurs, je me suis fixé certains objectifs et je suis parti de certaines hypothèses qui m'ont amené à faire certains choix que j'expliciterai en détail.

Bonne lecture...

Première partie

Les protocoles existants

Chapitre 1

IPv6

1.1 Introduction

Les réseaux IP prennent de nos jours une part de plus en plus importante dans les réseaux de télécommunication. Ces dernières années, Internet s'est imposé comme un nouveau média pour le grand public et les entreprises. La force d'IP réside dans sa simplicité, ce qui lui permet de répondre à des besoins de plus en plus variés.

Malheureusement, la technologie commence à montrer ses limites. IP est victime de son succès : le nombre d'adresses disponibles est le problème le plus préoccupant à l'heure actuelle. Le plan d'adressage inventé il y a vingt ans ne pouvait tenir compte de l'explosion du nombre d'équipements connectés au réseau dans les années qui suivirent.

IPv4, la version actuelle du protocole IP, se complexifie au fur et à mesure que de nouvelles fonctionnalités sont intégrées. La mobilité, la sécurité et la qualité de service sont de nouveaux objectifs pour l'Internet de demain. Pour pouvoir gérer ces nouveaux services avec IPv4, il faut créer des fonctions annexes qui augmentent la complexité et de ce fait, réduisent les performances des équipements. Pour en revenir à une gestion plus simple, plus intégrée et encore mieux conçue, l'IETF travaille depuis 1990 à l'élaboration d'IPv6 ([16, 17]).

La description d'IPv6 qui suit est loin d'être exhaustive. Elle permet juste de saisir les points principaux du protocole et les différences avec IPv4 [1].

1.2 Adressage

La plus grande innovation d'IPv6 est sans aucun doute l'élargissement de la taille des adresses. Au lieu des 32 bits d'IPv4 que l'on notait de façon décimale pointée, on dispose maintenant d'adresses sur 128 bits. La façon de noter les adresses a été revue aussi. On les écrit en hexadécimal, en séparant les mots de 16 bits par des " :".

IPv6 reconnaît trois types d'adresses : unicast, multicast et anycast (voir figure 1.1). L'unicast sert simplement à identifier une interface unique. Le multicast désigne un groupe d'interfaces pouvant être situées n'importe où sur Internet. Un paquet ayant pour destination une adresse multicast sera acheminé vers tous les membres du groupe. Contrairement à IPv4, il n'y a plus d'adresses de type broadcast. Le dernier type, anycast, est nouveau en IPv6. Comme

[1] Pour plus d'informations, se référer à [12]

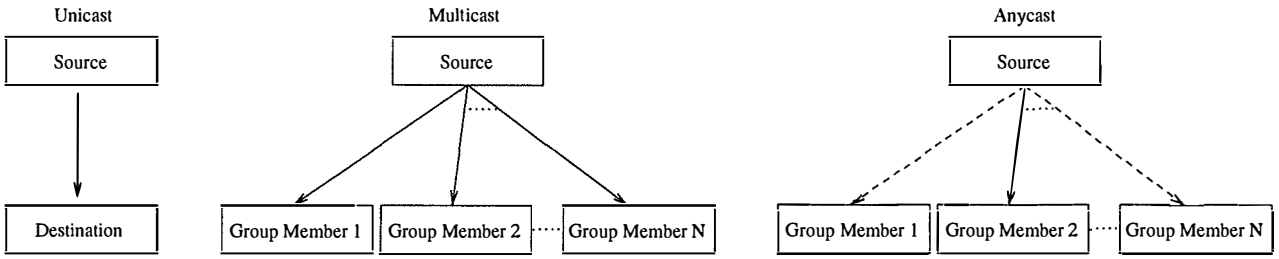

FIG. 1.1 – Modes de transmission en IPv6

pour le multicast, l'adresse désigne un groupe d'interfaces. La différence est que lorsqu'un paquet a pour destination une telle adresse, il est acheminé à un des éléments du groupe et non pas à tous.

1.2.1 Adresses unicast

Une adresse unicast IPv6 est constituée de deux parties. La première, située dans les bits de poids fort, identifie le sous-réseau, tandis que la deuxième identifie l'interface de l'équipement IPv6 dans ce sous-réseau. Le préfixe est subdivisé de façon hiérarchique en une topologie publique et une topologie de site.

Plusieurs plans d'adressage ont été proposés pour finalement aboutir au plan agrégé. Celui-ci consiste à subdiviser la topologie publique en plusieurs parties. Les trois bits de poids fort sont positionnés à 0, 0 et 1 et caractérisent le plan d'adressage. A côté, on place une unité d'aggrégation haute (TLA, Top Level Aggregator) sur 13 bits, qui symbolise l'organisation. Les unités suivantes (NLA, Next Level Aggregator), permettent de subdiviser l'organisation en différentes entités et sous-entités. Finalement, on trouve le SLA (Site Level Aggregator) qui divise chaque entité en sous-réseaux (voir Figure 1.2).

Toutes les adresses unicast ne sont pas publiques. Il existe donc des adresses dont la portée est limitée au lien (*Link Local Addresses*) et des adresses dont la portée est limitée au site (*Site Local Addresses*). Deux équipements IPv6 situés dans deux sites distincts peuvent donc posséder la même adresse unicast IPv6 site-local. Les adresses privées existaient aussi dans IPv4 mais il fallait obligatoirement utiliser les préfixes `10.0.0.0/8` et `192.168.144.0/24`.

Les 64 bits de poids faible, qui permettent d'identifier un équipement IPv6 dans un site, peuvent être positionnés manuellement ou par auto-configuration. Dans le cas des cartes réseaux de type Ethernet, l'auto-configuration se fait en insérant la chaîne hexadécimale **`0xFFFE`** après les 24 premiers bits de l'adresse MAC. On appelle ce mécanisme l'autoconfiguration sans état (voir Figure 1.2).

Contrairement à IPv4 où il n'était possible d'avoir qu'une seule adresse unicast par interface, il est permis d'avoir plusieurs adresses pour une seule interface en IPv6. Comme en IPv4, les adresses sont assignées aux interfaces et non aux noeuds. Un noeud peut être identifié par une adresse globale particulière et peut posséder plusieurs interfaces.

Les adresses et les préfixes sont valides durant une certaine période. Lors d'une renumérotation de réseau par exemple, un nouveau préfixe est annoncé, tout en continuant d'annoncer l'ancien préfixe. Les noeuds ont alors deux adresses IPv6, qu'elles utilisent selon leur

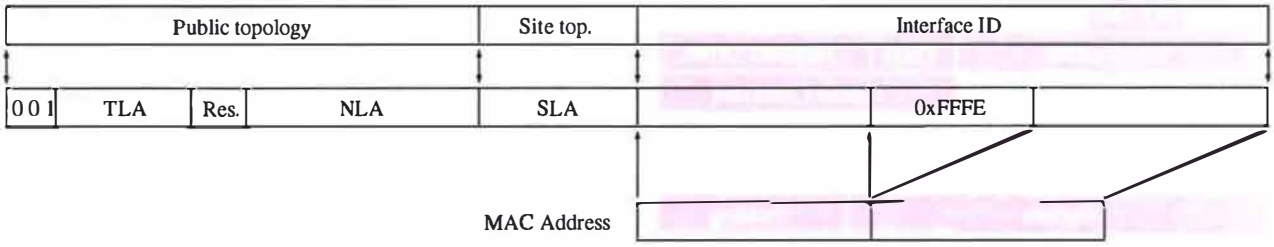

FIG. 1.2 – Plan d'adressage agrégé pour les adresses unicast

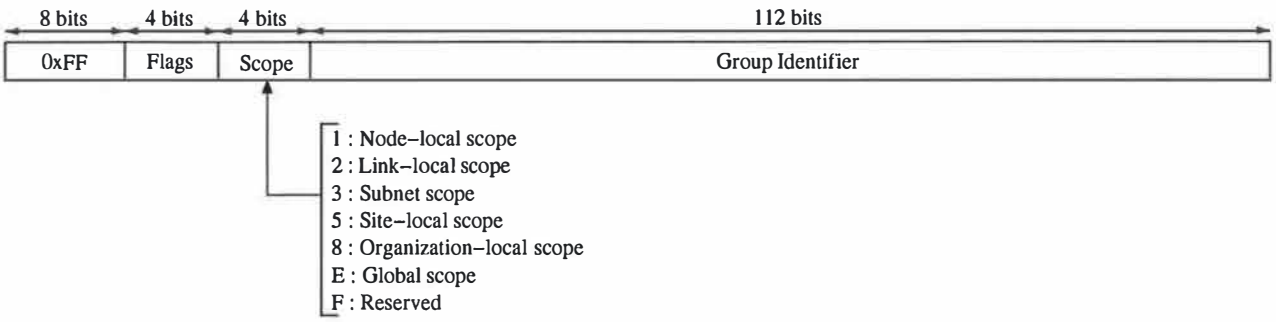

FIG. 1.3 – Adresses multicast

convenance. Après un certain temps, on invalide le premier préfixe pour pousser les machines à utiliser leur nouvelle adresse. Finalement, on arrête la diffusion du premier préfixe et les noeuds ont ainsi pu changer de préfixe en douceur.

La politique de choix de l'adresse source peut dépendre de beaucoup de paramètres. Si un hôte est connecté à plusieurs réseaux simultanément, le choix peut se faire sur base du coût d'utilisation du lien, de la bande passante disponible, etc.

1.2.2 Adresses multicast

Une adresse multicast identifie un groupe de noeuds (voir Figure 1.3). L'octet de poids fort d'une adresse multicast est toujours positionné à sa valeur maximale (`0xFF`). Les trois bits suivants sont réservés pour une utilisation future et doivent être nuls pour le moment. Le bit suivant indique si l'adresse est temporaire ou permanente.

Le champ "Scope" de quatre bits limite la portée topologique de l'adresse multicast. Plusieurs niveaux de diffusion ont été définis actuellement permettant de désigner des zones plus ou moins grandes. Par exemple, le niveau 2 (link-local) signifie qu'un datagramme multicast ne peut pas quitter le lien d'où il a été émis. Ce scope est utilisé entre autres par le protocole de découverte des voisins. Un autre exemple serait une vidéoconférence en entreprise qui utiliserait le niveau 8 (organization-local). De cette façon, on est assuré qu'un noeud situé hors de l'organisation ne peut s'abonner au groupe et donc participer à la vidéoconférence. De plus, une autre organisation pourrait utiliser la même adresse de groupe sans qu'il y ait de conflit car les datagrammes multicast resteraient confinés dans leurs organisations respectives.

Les 112 bits de poids faible identifient le groupe. Par exemple, l'identifiant `0x101` a été attribué aux serveurs NTP (*Network Time Protocol*).

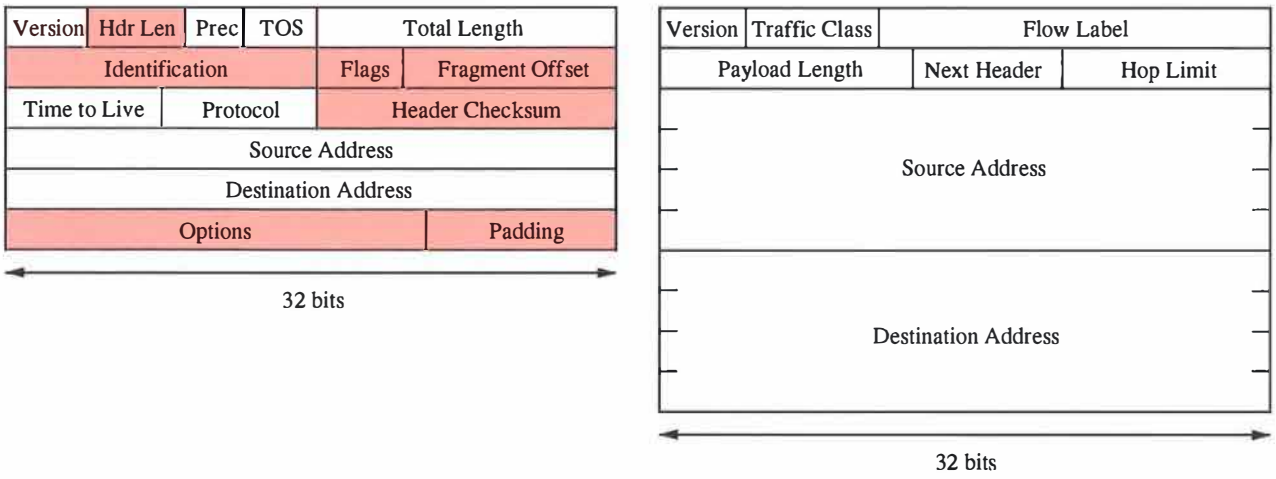

FIG. 1.4 – A gauche : l'en-tête IPv4, à droite : l'en-tête IPv6

1.2.3 Adresses anycast

Une adresse anycast n'a pas de forme particulière. Il s'agit simplement d'une adresse unicast qui est attribuée à plusieurs interfaces. L'anycast est encore un sujet de recherche et aucune expérience grandeur nature n'a été effectuée à ce sujet pour le moment.

1.3 Simplification du protocole IP

L'expérience acquise avec IPv4 a permis de distinguer ce qui était nécessaire dans l'en-tête IPv4 de ce qui était facultatif. L'en-tête occupe désormais 40 octets (à cause de la longueur des adresses). Les checksums, qui devaient être ajustés par chaque routeur intermédiaire en raison de la décrémentation du TTL, ont été supprimés (voir Figure 1.4). Pour éviter les erreurs d'en-tête, tout protocole transport se doit d'inclure un pseudo-checksum prenant en compte au minimum les adresses source et destination. La taille de l'en-tête est fixe pour faciliter le travail des routeurs. Les options sont retirées de l'en-tête et remplacées par des extensions, qui peuvent facilement être ignorées par les routeurs intermédiaires. Les champs sont alignés sur des mots de 64 bits. Le MTU minimum est de 1280 octets (64 pour IPv4), ce qui permet le tunnelage de paquets IPv6. La fonction de fragmentation a été retirée, car il y a maintenant un algorithme de découverte du Path MTU évitant de devoir recourir à la fragmentation.

La structure de l'en-tête IPv4 n'était pas extensible et ne permettait donc pas d'intégrer facilement de nouveaux services (comme la mobilité par exemple). Grâce au champ "en-tête suivant" d'IPv6, la création de nouveaux services ne pose plus problème.

Le champ "identificateur de flux" n'est pas utilisé pour le moment mais pourra être utilisé par des mécanismes de qualité de service afin de différencier les flux de façon simple et efficace.

1.4 ICMPv6

Le protocole de contrôle d'IP a été revu et amélioré. Dans IPv4, ICMP (*Internet Message Control Protocol*) servait aux tests (*Echo Request, Echo Reply*), à la détection d'erreurs et à la configuration automatique des équipements. Ces trois fonctions ont été reprises et revues dans

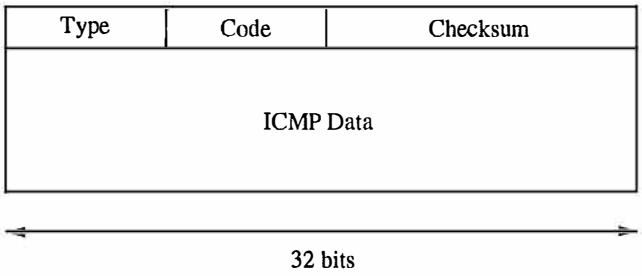

FIG. 1.5 – Structure d'un message ICMPv6

ICMPv6 [13]. La nouvelle version intègre le protocole de gestion des groupes multicast (voir Section 3.1.3) et le protocole de découverte des voisins (ARP en IPv4). On y trouve aussi des messages permettant l'annonce, la découverte et la renumérotation des routeurs (voir Figure 1.6).

Les messages ICMPv6 de compte-rendu d'erreur contiennent dans la partie "données" le paquet IPv6 ayant provoqué l'erreur. Pour éviter les problèmes de fragmentation, la taille maximale d'un message ICMPv6 est limitée à 1280 octets. Par conséquent, le contenu du paquet IPv6 peut être tronqué.

Le format des messages ICMPv6 est très simple (voir Figure 1.5). Le premier champ, *Type*, code la nature du message ICMP. Les valeurs inférieures à 127 sont réservées aux messages d'erreur. Les autres valeurs sont réservées aux messages d'information. Le deuxième champ, *Code*, précise la cause du message. Le troisième champ, *Checksum* permet de vérifier l'intégrité. Finalement, on trouve les données, qui varient bien évidemment d'après le *Type*.

1.5 Configuration automatique

La configuration automatique est l'un des principaux attraits d'IPv6 ([35], [44]). Un équipement non configuré mais connecté au réseau peut découvrir tous les paramètres nécessaires à son bon fonctionnement au niveau IP. Un mécanisme de découverte des routeurs permet de connaître les routeurs actifs sur le lien physique. Un autre mécanisme permet aux noeuds d'apprendre le ou les préfixes du réseau. En y ajoutant son identifiant d'interface, chaque machine crée de façon dynamique des adresses valides topologiquement. Toutefois, il pourrait arriver que deux noeuds utilisent les mêmes adresses (surtout si elles sont configurées manuellement, car les identifiants des cartes réseau sont globalement uniques). C'est pourquoi il existe un mécanisme de détection des adresses dupliquées. Toutes sortes de paramètres, comme la taille maximale du MTU, le nombre maximal de sauts autorisés, la disponibilité de la configuration avec état (DHCPv6 [5]) peuvent aussi être découverts dynamiquement.

1.6 Sécurité

Etant donné l'explosion des utilisateurs connectés à Internet, il est devenu indispensable d'intégrer des mécanismes de sécurité dans IPv6.

Ces mécanismes, appelés IPSec et placés au niveau IP, permettent aux couches supérieures (transport et application) d'utiliser la sécurité de façon unique. Contrairement à IPv4, tout

noeud IPv6 se doit d'implémenter IPSec. Tout noeud désirant une communication sécurisée peut donc utiliser sans problème IPSec avec ses correspondants.

Plusieurs services de sécurité sont disponibles dans IPSec. Nous ne ferons que les citer afin de savoir qu'ils existent car cela dépasse le cadre de ce document :

- La confidentialité des données en chiffrant celles-ci pour qu'elles ne soient compréhensibles que par les entités autorisées.
- L'authentification de l'origine des données garantit que les données proviennent bien de l'entité déclarée.
- L'authentification mutuelle permet aux entités de prouver leur identité.
- L'intégrité des données garantit que celles-ci n'ont pas été modifiées durant leur voyage sur le réseau.
- La prévention contre le rejeu assure que les données n'ont pas déjà été reçues.
- La non-répudiation garantit, en cas de litige, que les informations ont bien été émises ou reçues par les entités.

1.7 Critique

Après tous les avantages que nous venons de citer, il pourrait être tentant de se demander pourquoi nous ne passons pas tout de suite en IPv6. Les arguments suivants devraient ramener le lecteur enthousiasmé sur terre.

Internet fonctionne actuellement en IPv4 et il est impossible de le faire basculer en IPv6 d'un coup. Ceci implique qu'il faut laisser cohabiter les deux protocoles pendant un certain temps pour que l'évolution se fasse en douceur. Diverses solutions ont été développées pour faire communiquer les deux mondes mais elles restent relativement complexes (NAT-PT [45], DSTM [6], BIS [47], SIIT [38], SOCKS [32]).

L'expertise en IPv6 est pratiquement inexistante. Cela prend du temps et de l'argent pour former le monde des développeurs et des ingénieurs réseau à ce nouveau protocole. IPv4 a été testé dans les laboratoires sous toutes les coutures et dans toutes les situations pratiques imaginables. Changer de protocole signifie aussi qu'il faut créer une nouvelle génération d'outils de tests.

Même si le protocole a du succès dans les centres de recherche, il pourra toujours être un échec commercial. Il faut convaincre les gestionnaires de réseaux des ISP de faire passer leur réseau à IPv6 et ceci n'est pas une mince affaire. Le problème central est qu'IPv6 n'apporte rien de "sensationnel", c'est-à-dire rien qui soit possible en IPv6 et impossible en IPv4. Des fonctionnalités comme le multicast, la sécurité, la mobilité ou l'autoconfiguration existent déjà en IPv4, même si elles sont moins bien conçues ou intégrées.

Le passage à IPv6 représente un coût non négligeable pour les ISP. Ce coût risque probablement de se répercuter sur les factures des clients privés et des entreprises. Ces derniers pourraient donc très bien s'abstenir de passer à IPv6 s'ils jugent que les avantages offerts par le nouveau protocole ne compensent pas la différence de prix.

Il va falloir convertir toutes les applications d'IPv4 en IPv6. Même si les changements sont mineurs pour chaque application, cela risque quand même de prendre des années.

On ne sait pas non plus combien de temps va durer la cohabitation IPv4-IPv6. Si les routeurs doivent éternellement supporter les deux protocoles, ils seront moins efficaces que s'ils n'en supportaient qu'un. L'argument des vitesses accrues par une meilleure conception de l'en-tête des datagrammes IPv6 tombe alors à l'eau.

L'autoconfiguration sans état d'IPv6 [44] via les adresses MAC des cartes Ethernet est critiquée par les défenseurs de la vie privée. En effet, ces adresses autoconfigurées contiennent l'identifiant globalement unique de la carte réseau qui peut être utilisé par des firmes de marketing pour tracer le comportement de l'internaute à long terme. Il existe néanmoins des solutions à ce problème : l'internaute peut désactiver l'autoconfiguration et configurer ses interfaces manuellement, utiliser un schéma d'autoconfiguration qui ne soit pas basé sur l'adresse MAC (par exemple en générant un nombre aléatoire comme identifiant d'interface) ou encore attribuer plusieurs adresses globales par interface et les utiliser de façon aléatoire. Toujours est-il que l'autoconfiguration, qui est une caractéristique importante d'IPv6, perd de sa grandeur. Ces problèmes sortent du cadre de ce mémoire. Le lecteur intéressé pourra consulter le document de J.M. Dinant du CRID (FUNDP) [18] et [34].

En somme, on peut dire qu'IPv6 comporte beaucoup d'améliorations très intéressantes par rapport à IPv4 mais qu'il faudra attendre que la transition soit terminée pour pouvoir en profiter pleinement.

Error Handling	
1	Destination unreachable
2	Packet too big
3	Time exceeded
4	Parameter problem

Information	
128	Echo request
129	Echo reply

Multicast Listener Discovery (MLD)	
130	Query
131	Reply
132	Leave

Neighbor Discovery	
133	Router solicitation
134	Router advertisement (RA)
135	Neighbor solicitation
136	Neighbor advertisement
137	Redirection

Router Renumbering	
138	Router renumbering

Node Information	
139	Query
140	Reply

Inverse Neighbor Discovery	
141	Solicitation
142	Advertisement

Mobility	
150	Home Agent Address Discovery Request
151	Home Agent Address Discovery Reply
152	Mobile Prefix Solicitation
153	Mobile Prefix Advertisement

FIG. 1.6 – Les différents messages ICMPv6

Chapitre 2

Mobilité

2.1 Introduction

Les protocoles de mobilité IP permettent à un équipement de se déplacer à travers différents réseaux en gardant ses connexions et en restant toujours accessible par une même adresse primaire. Ces deux exigences posent de sérieux problèmes, car tous les protocoles existants basés sur IP font l'hypothèse que tous les noeuds du réseau sont fixes. De plus, un principe de base d'IP veut que l'adresse d'un noeud soit construite de telle façon que le routage des informations vers ce noeud se base exclusivement sur son adresse. Or, si un noeud se déplace, il faut que le routage se modifie et donc que les adresses se modifient aussi.

La section suivante décrit précisément les problèmes rencontrés lors du déplacement d'un noeud d'un réseau dans un autre réseau. Ensuite, nous verrons les deux protocoles standardisés par l'IETF (Mobile IPv4 et Mobile IPv6) permettant de résoudre la plupart des problèmes dûs à la mobilité.

2.2 Problème

On appelle noeud mobile (MN) un équipement IP qui se déplace et s'attache temporairement pour une durée indéterminée à différents réseaux. Tout MN appartient à un réseau d'origine, appelé Home Network (HN), dans lequel il possède une adresse privilégiée, la Home Address. Un mobile situé dans son Home Network se comporte comme n'importe quel autre noeud.

Supposons qu'un noeud mobile ouvre une communication Unicast avec un correspondant (CN). Les datagrammes de MN à CN ont comme adresse source MN et comme adresse destination CN. Chaque routeur du réseau qui reçoit ceux-ci sur une certaine interface consulte sa table de routage pour les retransmettre sur la bonne interface de sortie. Finalement, les datagrammes arrivent chez le correspondant (voir Figure 2.1).

Lorsque le MN quitte son Home Network et s'attache à un réseau visité (*Foreign Network*), il se trouve dans une situation telle que son adresse IP n'est plus correcte d'un point de vue topologique (son adresse n'est pas adaptée au plan de numérotation du réseau visité). Deux cas de figure sont possibles : soit le MN garde son adresse, soit il change d'adresse et utilise une adresse faisant partie du sous-réseau visité.

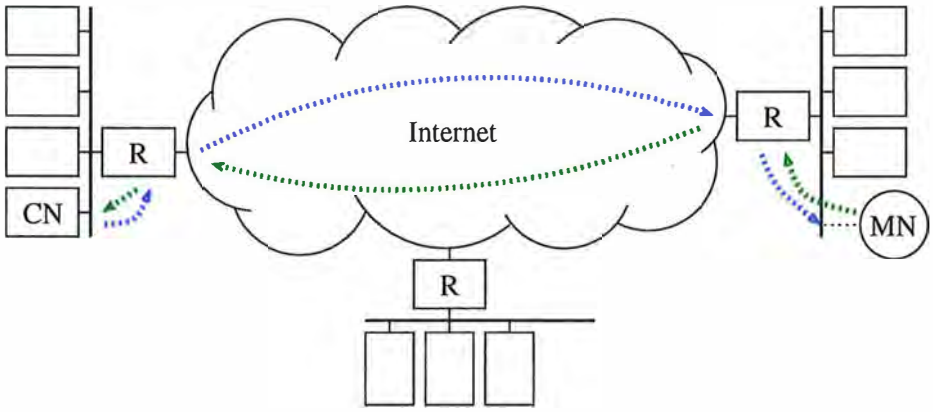

FIG. 2.1 – Mobile communiquant avec un CN avant déplacement

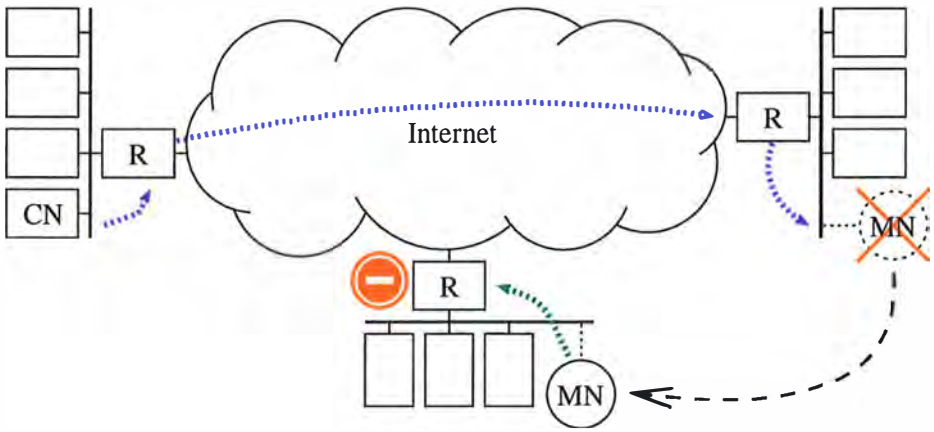

FIG. 2.2 – Le mobile garde son adresse

Si le mobile garde son adresse, son adresse n'est pas valide dans le réseau visité et la communication avec le correspondant ne peut plus continuer pour une des deux raisons suivantes. Si le routeur d'accès du réseau visité filtre en sortie les datagrammes ayant une adresse source dont le préfixe n'est pas celui du site visité, le CN ne reçoit plus les datagrammes et la connexion est coupée (si ce filtrage n'est pas mis en place, les datagrammes de MN à CN seront transmis correctement). De plus, les datagrammes de CN à MN sont toujours routés vers le réseau Home du mobile car le routage se fait sur base de la même adresse IP. Ces datagrammes n'atteignent donc pas le MN (voir Figure 2.2).

Si le mobile change d'adresse, il est considéré comme un autre noeud IP et la connexion ouverte avec le CN sous l'ancienne adresse IP ne peut plus continuer. En effet, la couche transport du CN utilise les adresses source et destination pour identifier la connexion. Comme l'adresse source des datagrammes n'est plus la même, ils ne sont pas considérés comme faisant partie de la connexion et sont jetés. De plus, les datagrammes de CN à MN sont toujours routés vers le Home Network du mobile car le CN continue à communiquer avec la même adresse IP (voir Figure 2.3).

On constate donc que pour maintenir ses connexions ouvertes lors de ses déplacements, un mobile doit :

– Garder la même adresse IP pour les couches supérieures afin de préserver les sessions.

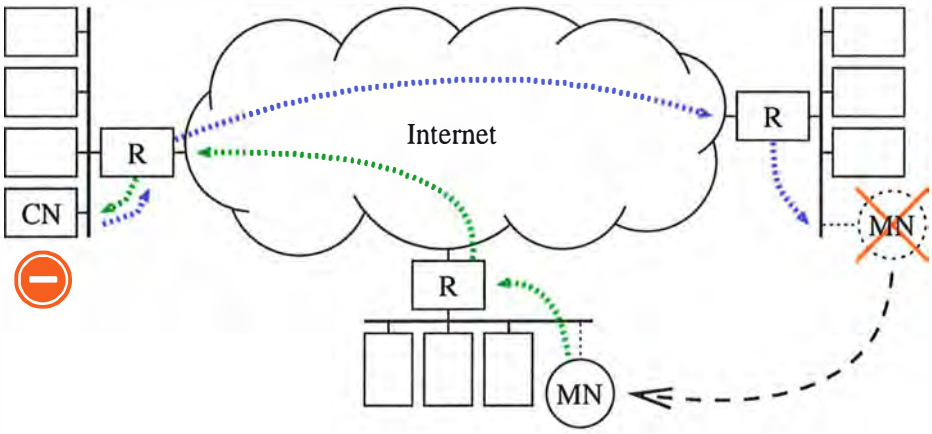

FIG. 2.3 – Le mobile change d'adresse

- Signaler sa position courante aux correspondants afin de continuer à recevoir ses datagrammes

Le protocole Mobile IP permet de répondre à ces deux exigences. Les sections qui suivent décrivent Mobile IPv4 et Mobile IPv6 en précisant les différences entre les deux versions.

2.3 Mobile IPv4

Ce protocole, consigné dans le RFC 2002 [39], puis révisé dans le RFC 3220 [40] définit les trois entités suivantes :

- Le Home Agent (HA) : il s'agit d'un routeur particulier situé sur le réseau Home du mobile. Il se charge de maintenir l'information concernant la localisation courante du mobile, et d'intercepter les datagrammes adressés à ce dernier avant de les lui retransmettre vers sa position courante.
- Le Foreign Agent (FA) : il s'agit d'un routeur particulier situé sur le sous-réseau visité par le mobile. Tout réseau visité ne doit pas forcément contenir un FA. Celui-ci se charge éventuellement d'enregistrer le mobile en tant que visiteur, de lui offrir les services de routage et de lui retransmettre ses datagrammes.
- Le Noeud Mobile (MN) : il s'agit d'un équipement IP qui se déplace de réseau en réseau. Il possède des fonctions pour détecter un changement de réseau et est capable de s'enregistrer auprès du Home Agent et éventuellement auprès du Foreign Agent.

Sur son réseau Home, le mobile fait appel aux fonctions de routage classiques pour échanger des datagrammes IP avec ses correspondants, comme s'il était un noeud fixe. Quand il se déplace et se connecte sur un réseau visité, il peut se trouver en présence ou non d'un FA, d'où les deux modes de fonctionnement.

2.3.1 Mode avec Foreign Agent

Supposons qu'un mobile MN connecté sur son réseau Home ouvre une communication avec un correspondant CN avant de se déplacer vers un réseau visité. Il commence par détecter son mouvement, c'est-à-dire le fait qu'il a changé de sous-réseau. Il détecte aussi la présence du FA. Le mobile envoie un message d'enregistrement (*Binding Update*) au FA qui le relaie au

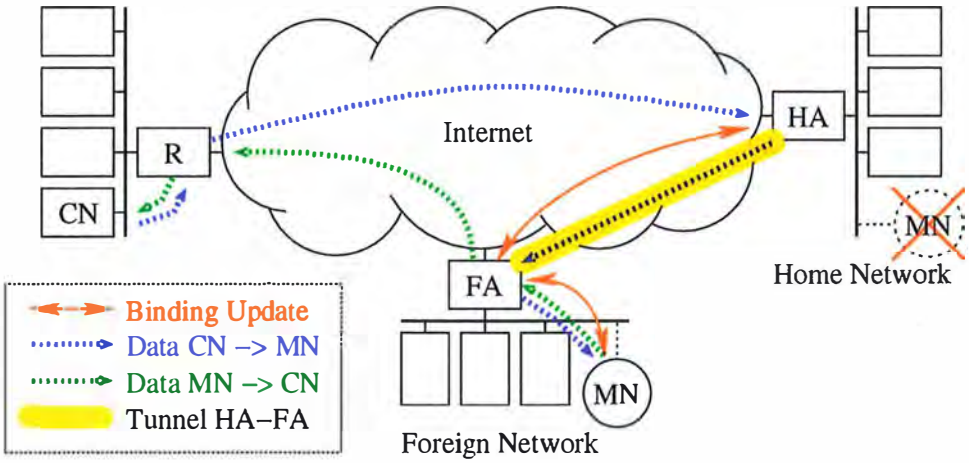

FIG. 2.4 – Mobile IPv4 - Avec Foreign Agent

HA. Le FA inscrit le mobile sur la liste de visiteurs tandis que le HA l'inscrit dans sa liste de mobiles en déplacement.

Ensuite, les datagrammes de MN vers CN sont routés directement. Le CN, qui n'est pas au courant de la nouvelle localisation du mobile, émet ses datagrammes à destination de l'adresse primaire du mobile. Ceux-ci arrivent donc dans le réseau Home. Le HA les intercepte, les encapsule et les envoie au FA qui les décapsule et les retransmet au MN (voir Figure 2.4). Quand le HA encapsule un datagramme, il le met dans un autre datagramme IP dont l'adresse source est l'adresse du HA et l'adresse destination est une adresse du FA. Cette dernière est dite CoA (Care-of Address). Elle représente le bout du tunnel côté FA et indique la position courante du MN. Ce mode a l'avantage de n'utiliser qu'une seule adresse (la CoA) pour l'ensemble des visiteurs.

2.3.2 Mode sans Foreign Agent

Lorsque le mobile se connecte sur le réseau visité, il commence par détecter son mouvement. En l'absence de FA, le mobile doit d'abord acquérir une adresse temporaire dans le réseau visité. Cette adresse est dite "Co-located Care-of Address" (CoCoA). L'acquisition de celle-ci peut se faire par configuration, par DHCP [19] ou par tout autre moyen. Le mobile s'enregistre directement avec son HA en utilisant un message *Binding Update* (BU). Comme précédemment, le HA inscrit le mobile dans sa liste de mobiles en déplacement.

Les datagrammes émis par MN vers CN sont routées directement. Le CN, lui, émet à destination de l'adresse primaire du mobile. Le HA intercepte les données, les encapsule et les envoie au MN. Quand le HA encapsule un datagramme, il le met dans un autre datagramme IP dont l'adresse source est l'adresse du HA et dont l'adresse destination est la CoCoA. Elle représente le bout du tunnel côté MN et indique sa position courante. Ce mode a l'inconvénient d'utiliser une adresse temporaire par visiteur, mais l'avantage de permettre à un mobile de fonctionner même en l'absence de FA (voir Figure 2.5).

Le routage des datagrammes est pratiquement identique dans les deux cas. La seule différence est que le bout du tunnel se situe dans le FA dans le premier cas et dans le MN dans le deuxième cas.

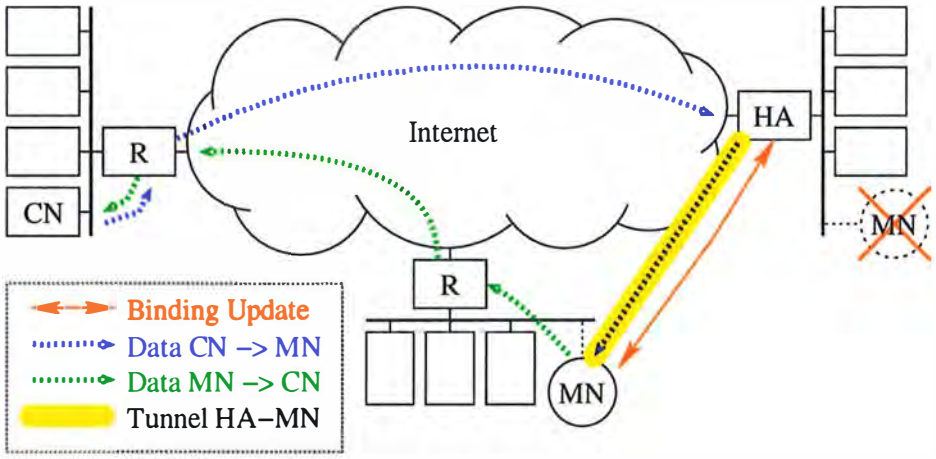

FIG. 2.5 – Mobile IPv4 - Sans Foreign Agent

2.3.3 Le tunnel inverse

Le tunnel inverse est nécessaire lorsqu'il y a filtrage sur l'adresse source des datagrammes émis, pratique de plus en plus courante pour des raisons de sécurité. Pour éviter que les paquets ne soient jetés, on met en place un tunnel inverse entre le FA et le HA. Ce tunnel est symétrique au tunnel qui existe dans le sens HA - FA avec les mêmes adresses aux extrémités et le même mode d'encapsulation. Tous les datagrammes émis par le mobile sont encapsulés par le FA et envoyés au HA qui les décapsule avant de les retransmettre à leur destinataire.

2.3.4 Optimisation de route

Jusqu'ici, il ne fallait rien modifier dans le fonctionnement du correspondant. Cependant, le fait que les datagrammes de CN à MN passent toujours par le HA produit un routage totalement inefficace. La solution pour éviter ce détour est de prévenir le correspondant de la localisation du mobile. Ceci nécessite l'implémentation de fonctions supplémentaires pour le support de la mobilité dans le correspondant. Lorsque le HA reçoit un datagramme pour un mobile qui n'est plus dans son réseau Home, il peut envoyer à l'émetteur un message *Binding Update* (BU). Ce message indique au correspondant la CoA du mobile. Si le correspondant supporte les fonctions de mobilité, il enregistre cette information et commence à encapsuler lui-même les datagrammes à destination du mobile. L'en-tête d'encapsulation a comme adresse source celle du correspondant et comme adresse destination celle du FA. Arrivés au réseau visité, les datagrammes sont décapsulés par le FA et retransmis au MN (voir Figure 2.6).

2.3.5 Smooth handoff

Lorsque le mobile arrive dans un nouveau réseau visité après en avoir quitté un précédent, il informe son HA de sa nouvelle position. Le HA peut à son tour en informer les correspondants. Cette mise à jour permet au HA et éventuellement aux correspondants d'envoyer leurs datagrammes vers le nouveau FA. Le mécanisme *Smooth handoff* a pour but d'informer le précédent FA de la nouvelle position du MN. Arrivé sur son nouveau réseau visité, le MN demande à son FA courant d'envoyer un BU à son FA précédent. A la réception du BU et après authentification, le FA précédent encapsule tous les datagrammes qu'il reçoit pour le mobile et les renvoie vers le nouveau FA. Ce dernier les décapsule et les transmet au MN.

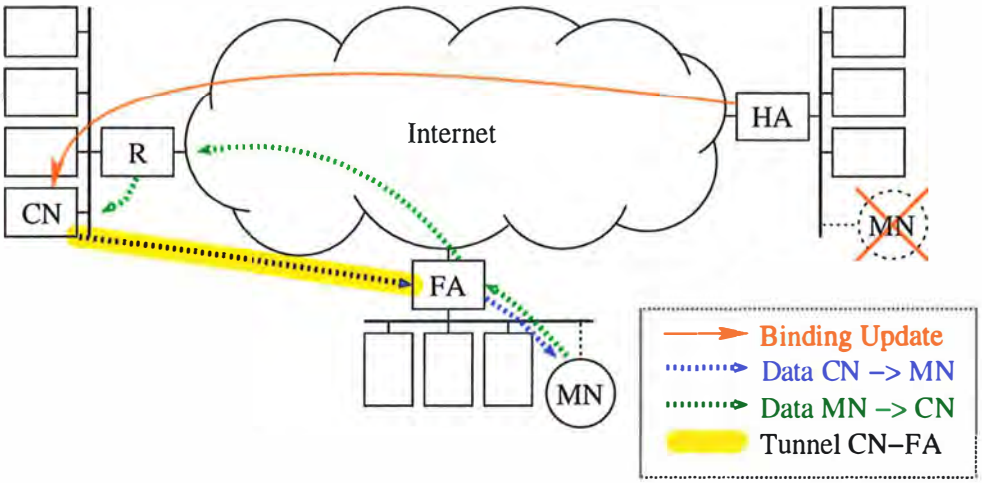

FIG. 2.6 – Mobile IPv4 - Optimisation de route

2.4 Mobile IPv6

Le support de la mobilité dans IPv6 est décrit dans un draft [30] et définit comme en Mobile IPv4 les entités Noeud Mobile et Home Agent. Il n'y a plus de Foreign Agent mais la gestion de la mobilité devient obligatoire chez tous les correspondants.

Sur son réseau Home, le mobile fait appel aux fonctions de routage classiques pour échanger des datagrammes IP avec ses correspondants. Tant que le mobile est chez lui, il se comporte comme un noeud fixe, c'est-à-dire qu'il profite entre autres des caractéristiques d'autoconfiguration d'IPv6.

Supposons que le MN ouvre une communication avec un correspondant CN avant de se déplacer sur un réseau visité. Il commence par détecter son mouvement et acquiert une adresse temporaire (*Care-of Address*, CoA) grâce aux *Router Advertisements* du protocole Neighbour Discovery [35] émis par le routeur du sous-réseau visité. Le mobile signale son déplacement à son HA et à tous ses correspondants avec lesquels une communication est en cours en envoyant des messages *Binding Update*. Ceux-ci sont intégrés aux datagrammes en tant qu'options IPv6.

Par la suite, les datagrammes échangés entre le MN et le CN sont routés directement entre ces deux noeuds. Il n'y a plus besoin de tunnels car les datagrammes utilisent les options *Home Address* et *Routing Header* (voir Figure 2.7).

Les datagrammes émis par le MN contiennent comme adresse source la CoA et l'option *Home Address* précisant l'adresse par laquelle le mobile est toujours joignable. Quand le correspondant reçoit ces datagrammes, il remplace la CoA par la Home Address pour ne pas perturber la couche transport qui a besoin d'une adresse source fixe pour une session donnée.

Les datagrammes émis par le CN contiennent comme adresse destination la CoA du MN et l'option *Routing Header* contenant la Home Address du mobile. Quand le mobile reçoit ces datagrammes, il remplace lui aussi la CoA par la Home Address avant de les transmettre à la couche transport afin de rendre la mobilité transparente.

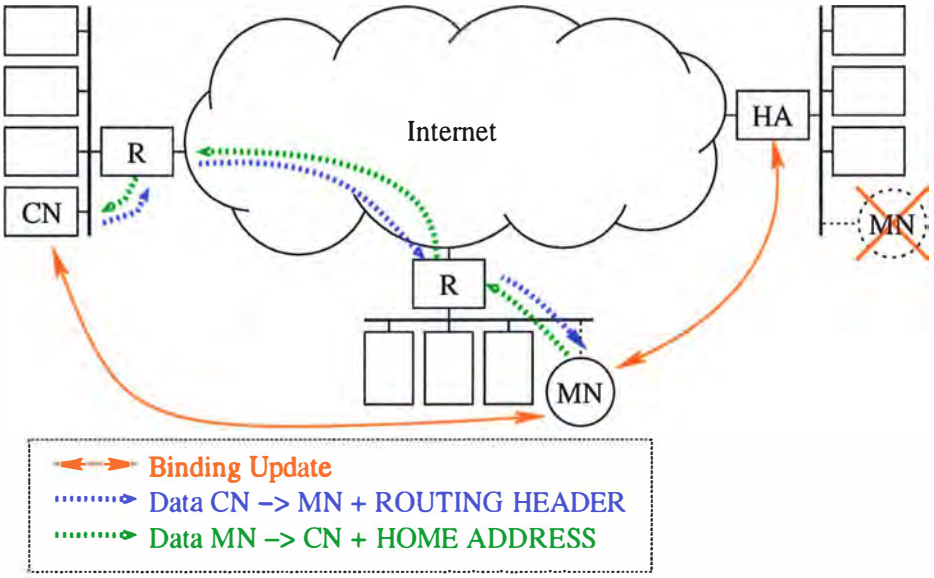

FIG. 2.7 – Mobile IPv6

Les datagrammes destinés au MN mais qui arrivent sur le réseau Home du mobile sont interceptés par le HA, encapsulés et envoyés au MN. Quand le HA encapsule un datagramme, il le met dans un autre datagramme IPv6 dont l'adresse source est celle du HA, et dont l'adresse destination est la CoA du MN qui représente le bout du tunnel côté MN et indique sa position courante.

Nous voyons toute la simplicité de Mobile IPv6 par rapport à Mobile IPv4 :

- Il n'y a pas de Foreign Agent, donc il n'est pas nécessaire de distinguer deux cas selon qu'un Foreign Agent est disponible sur le réseau visité ou pas.
- Aucun tunnel n'est nécessaire grâce à l'utilisation d'options IPv6 particulières appelées *Routing Header* et *Home Address*.
- Comme le mobile crée une adresse valide sur le réseau visité grâce aux *Router Advertisements*, il n'est pas nécessaire d'établir un tunnel inverse pour éviter les problèmes de filtrage.
- Le tunnel du HA vers le MN est utilisé seulement si le correspondant n'est pas au courant de la localisation courante du mobile. Dès que le MN reçoit un paquet tunnelé de son correspondant, il lui envoie un BU pour le prévenir d'optimiser son routage.
- Les messages *Binding Update* sont intégrés aux datagrammes de données en tant qu'options. Il n'est donc pas nécessaire de définir un nouveau format de messages comme en Mobile IPv4.
- La sécurité est intégrée au protocole IPv6 en tant qu'extension. Tous les mécanismes sont donc disponibles pour répondre aux exigences de la mobilité en termes de sécurité.
- Le mécanisme de création d'une adresse temporaire (CoA) en IPv6 est simplifié grâce aux protocoles évolués "Neighbour Discovery" et "Stateless Address Autoconfiguration" propres à IPv6.
- La pénurie d'adresses IPv4 est un problème qui peut rendre impossible la mise en oeuvre de certains services de mobilité IP. Ce problème ne se pose plus en IPv6 vu l'abondance des adresses. De plus, la possibilité en IPv6 de pouvoir configurer une interface avec plusieurs adresses facilite la mise en oeuvre des services de mobilité.

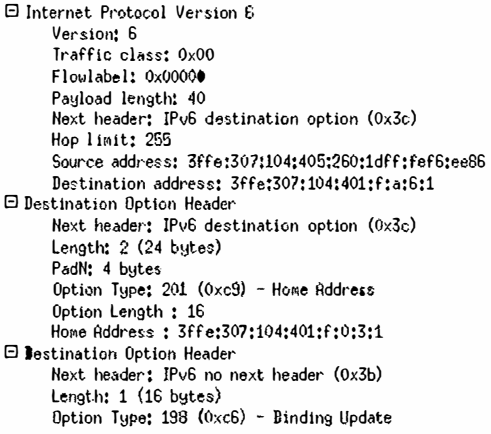

FIG. 2.8 – Mobile IPv6 - Binding Update

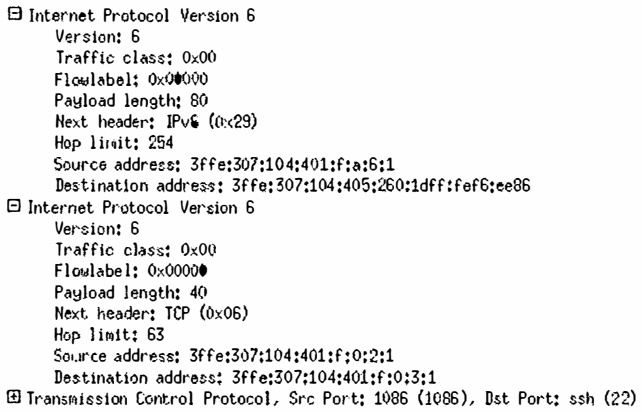

FIG. 2.9 – Mobile IPv6 - Tunnel entre le HA et MN3

2.4.1 Exemple

Afin d'illustrer le protocole Mobile IPv6, prenons un exemple concret. Soit un mobile appelé MN3 d'adresse Home `3ffe:307:104:401:f:0:3:1`. Celui-ci fait un Secure Shell (TCP, port 22) vers MN2 qui a comme adresse `3ffe:307:104:401:f:0:2:1`.

Ensuite MN3 se déplace sur le réseau `3ffe:307:104:405::/64` où il acquiert une CoA : `3ffe:307:104:405:260:1dff:fef6:ee86`. Il émet un *Binding Update* vers son Home Agent (`3ffe:307:104:401:f:a:6:1`) (voir Figure 2.8) et vers MN2 pour signaler sa nouvelle position. Tant qu'il n'a pas reçu le *Binding Update*, MN2 continue à transmettre ses datagrammes à l'ancienne adresse. Le Home Agent capture ceux-ci et les tunnelle vers MN3 (voir Figure 2.9). Les datagrammes émis par MN3 vers MN2 sont routés directement et contiennent une option de destination Home Address (voir Figure 2.10). Dans le sens inverse, les datagrammes de MN2 vers MN3 sont aussi routés directement dès que le *Binding Update* a été reçu, mais ils contiennent un *Routing Header* précisant la Home Address du destinataire (voir Figure 2.11).

2.5 Dynamic Mobile IPv6 - DMI

DMI est un draft [31] proposé par Mohamed Kassi-Lahlou et Christian Jacquenet à l'IETF. Je l'ai implémenté dans le cadre de mon stage chez France Telecom R&D à Caen [7]. Il vient

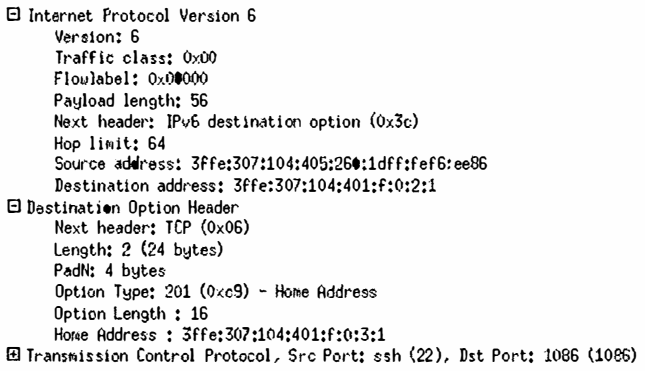

FIG. 2.10 – Mobile IPv6 - Datagramme de MN3 vers MN2 contenant une option de destination Home Address

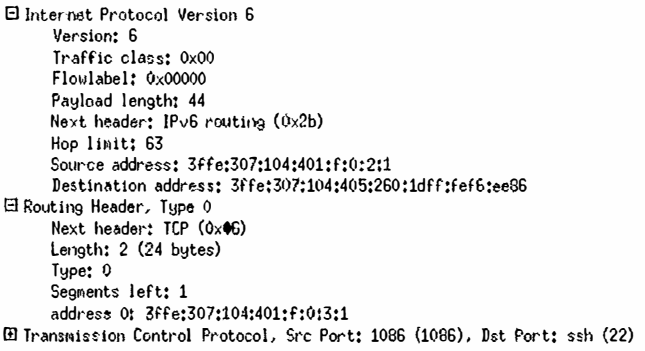

FIG. 2.11 – Mobile IPv6 - Datagramme de MN2 vers MN3 contenant un Routing Header

de la constatation suivante : dans un environnement de production, il est peu probable que les mobiles, même s'ils se déplacent, ne changent de sous-réseau, pour peu que la surface géographique couverte par un sous-réseau soit suffisante. Il est donc peu utile et efficace d'utiliser les mécanismes Mobile IPv6 dans ces situations. L'idéal serait de n'utiliser Mobile IPv6 que quand il y a changement de sous-réseau.

Regardons la situation d'un mobile s'étant déplacé dans un réseau visité et qui établit toutes sortes de connections sortantes (le mobile est surtout client) vers un ou plusieurs correspondants. Chaque paquet émis par le mobile contient comme adresse source la Care-of Address (CoA) et une option de destination de type Home Address spécifiant l'adresse Home du mobile. Arrivé chez le correspondant, celui-ci devra remplacer la source address par la home address, afin de conserver la transparence pour les couches supérieures. A son tour, le correspondant répond au mobile en lui envoyant des paquets destinés à la CoA (évitant ainsi le routage triangulaire par le Home Agent) et contenant l'option Routing Header spécifiant l'adresse Home du mobile. De la même manière, le mobile remplace l'adresse source du paquet par l'adresse contenue dans l'option avant de le transmettre à la couche supérieure. Tout ce mécanisme est nécessaire au traitement de chaque paquet et oblige en plus le correspondant à maintenir dans une table (Binding Cache) la correspondance entre la CoA et la Home Address du mobile. De même, le mobile doit conserver la trace qu'il a envoyé un *Binding Update* au correspondant pour lui dire de lui router directement les paquets à son endroit actuel.

Si le mobile n'a pas changé de sous-réseau jusqu'à la fin de la communication, tout ce travail a été fait pratiquement pour rien. En effet, si le mobile n'avait utilisé que sa CoA pour transmettre ses paquets, c'est-à-dire sans mentionner sa Home Address, cela n'aurait rien changé au résultat. En fait, pour de nombreuses applications (web, mail, ...) le correspondant n'a pas besoin de connaître l'adresse originale du mobile, s'il ne fait que répondre à ses requêtes. Cependant, n'utiliser que la CoA pour transmettre les paquets pose problème lors des changements de réseaux : comment garder la transparence pour les couches supérieures alors qu'elles n'étaient pas au courant de la Home Address ?

DMI propose une solution simple et élégante à ce problème. On considère qu'une communication sortante commence dans un réseau que l'on appelle le Temporary Home Network. On appelle la CoA acquise dans ce sous-réseau la Temporary Home Address. Grâce à ces nouvelles appellations, on peut maintenant décider de donner aux paquets la Temporary Home Address dans le champ Home Adress pour les paquets émis dans un autre sous-réseau que celui d'où la communication a débuté. En d'autres mots, la Home Address devient un concept mobile : au moment où l'on commence une communication avec un correspondant, on utilise la CoA courante comme Home Address pour cette communication. Si la communication se termine dans le même sous-réseau, on n'aura pas utilisé Mobile IP car on sera resté dans son Temporary Home Network. Si on se déplace par contre, on utilisera Mobile IP, c'est-à-dire en envoyant un Binding Update au correspondant contenant comme Home Address la Temporary Home Address, c'est-à-dire la CoA utilisée au début de la communication.

Si le mobile ouvre une nouvelle connexion avec un correspondant, il regarde d'abord son cache d'associations (Binding Update List). Ce cache contient pour chaque correspondant la Temporary Home Address qu'il faut utiliser. S'il y trouve son correspondant, il utilise la Temporary Home Address contenue dans le cache. Sinon, il ajoute le correspondant dans le cache et utilise comme Temporary Home Address pour ce correspondant sa CoA courante. Ceci implique que le mobile se trouve dans son Temporary Home Network *pour cette connexion.* Il

ne faut donc pas utiliser de *Routing Header* ni d'option *Home Address* dans les datagrammes échangés entre le correpondant et le mobile (un mobile sur son réseau Home se comporte toujours comme un noeud fixe).

Les associations sont valides plus longtemps que la durée d'une communication, ce qui permet d'utiliser la même adresse pour plusieurs communications successives. Si une association expire, la prochaine communication avec le correspondant utilisera comme Temporary Home Address la CoA valide à ce moment-là.

DMI garde la notion de Home Address définie dans Mobile IP, renommée ici Permanent Home Address. Ceci est obligatoire pour pouvoir encore établir des connexions entrantes sur le mobile. Même si celui-ci ne communique pas toujours avec la même Home Address, il faut qu'il soit toujours joignable par une adresse générique, appelée ici Permanent Home Address.

2.5.1 Avantages

Le concept de Home Address devient dynamique, ce qui signifie que le mobile peut utiliser simultanément plusieurs Temporary Home Addresses différentes pour des correspondants différents. Si les communications se terminent avant que le mobile ne change de réseau d'où elles ont commencé, DMI permet d'éviter l'utilisation de mécanismes Mobile IP (*Binding Updates*, *Routing Headers*, options *Home Address*, inscription de la correspondance entre Home Address et CoA dans la Binding Cache du correspondant).

DMI permet de n'utiliser les extensions de mobilité que quand cela est vraiment nécessaire. Il a d'ailleurs été prouvé que rares sont les mobiles qui changent vraiment de zone géographique durant une communication.

DMI consiste seulement en une modification du comportement du noeud mobile. Ainsi, un mobile utilisant DMI peut très bien communiquer avec des correspondants n'utilisant pas DMI. On garde donc une compatibilité avec Mobile IP tout en allégeant les tables des correspondants sans qu'ils s'en rendent compte, s'il n'y a pas changement de sous-réseau.

2.5.2 Inconvénients

DMI présente quand même quelques inconvénients. Si le mobile garde toujours au moins une communication avec son correspondant, l'entrée de la table ne sera jamais invalidée, Mobile IP sera toujours utilisé et DMI n'apportera aucun bénéfice.

Le correspondant ne peut plus authentifier le mobile en regardant seulement sa Home Address, vu qu'elle est variable dans DMI. De toutes façons, on se rend compte de plus en plus qu'il ne faut pas baser l'authentification d'une personne sur l'adresse qu'elle utilise mais sur d'autres paramètres plus complexes. Ceci est d'autant plus vrai qu'il est facile de falsifier l'adresse source des paquets IP pour se faire passer pour quelqu'un d'autre.

Pour des raisons de sécurité, certains serveurs (par exemple IRC) essayent de faire un reverse look-up DNS sur l'adresse du client avant de continuer la communication. Cette technique est incompatible avec DMI car le reverse lookup se fera sur la CoA et non sur la Home Address. Or il est très improbable que la CoA soit inscrite dans le DNS du réseau visité.

Chapitre 3

Multicast

3.1 Introduction

Le Multicast permet de transmettre efficacement des messages entre un ou plusieurs émetteurs et un ou plusieurs récepteurs. Efficace signifie ici qu'un datagramme ne sera pas émis plusieurs fois sur un même lien, ce qui permet d'éviter des gaspillages de bande passante. Les émetteurs et récepteurs forment ce que l'on appelle un groupe. On dit alors qu'ils en sont les membres.

Les protocoles de routage multicast fonctionnent tous sur un même principe : ils construisent un arbre de distribution entre une (ou plusieurs) source(s) et les récepteurs. La disposition en arbre a un grand avantage : il n'existe qu'un seul chemin entre n'importe quel couple de noeuds de l'arbre.

Une fois qu'un arbre est construit, tout le trafic multicast est acheminé via celui-ci. Les routeurs de l'arbre utilisent un algorithme de décision simple (*Reverse Path Forwarding Check*) un datagramme reçu sur une branche de l'arbre est retransmis sur toutes les autres branches de l'arbre, à condition que la branche de réception soit celle qui serait utilisée pour joindre la source du datagramme. Contrairement aux protocoles de routage unicast où un datagramme qui arrive sur un routeur est retransmis sur une seule interface de sortie, un datagramme multicast peut donc être retransmis sur plusieurs interfaces.

3.1.1 Arbres de distribution multicast

Il existe plusieurs sortes d'arbres de distribution multicast. L'arbre le plus simple est tel que la racine est la source de trafic et tel que les branches de l'arbre couvrent le réseau pour arriver jusqu'aux récepteurs. Comme cet arbre utilise le plus court chemin entre la source et chaque récepteur, on l'appelle "arbre de plus court chemin" (*shortest path tree, SPT*). On utilise une notation spéciale pour désigner les arbres. Pour les arbres de plus court chemin, on écrit (S,G) pour désigner un SPT de source S et d'adresse de groupe G. La figure 3.1.1 montre un arbre SPT reliant une source à deux récepteurs. Les flèches indiquent le flot des datagrammes multicast le long de l'arbre.

Il est aussi possible de créer des arbres partagés véhiculant les datagrammes de plusieurs sources. On les note (*,G), où l'étoile signifie que l'arbre n'est pas spécifique à une certaine source. Le trafic des différentes sources est d'abord acheminé vers la racine de l'arbre qui est fixée arbitrairement dans le réseau. Ensuite la racine retransmet les datagrammes le long de

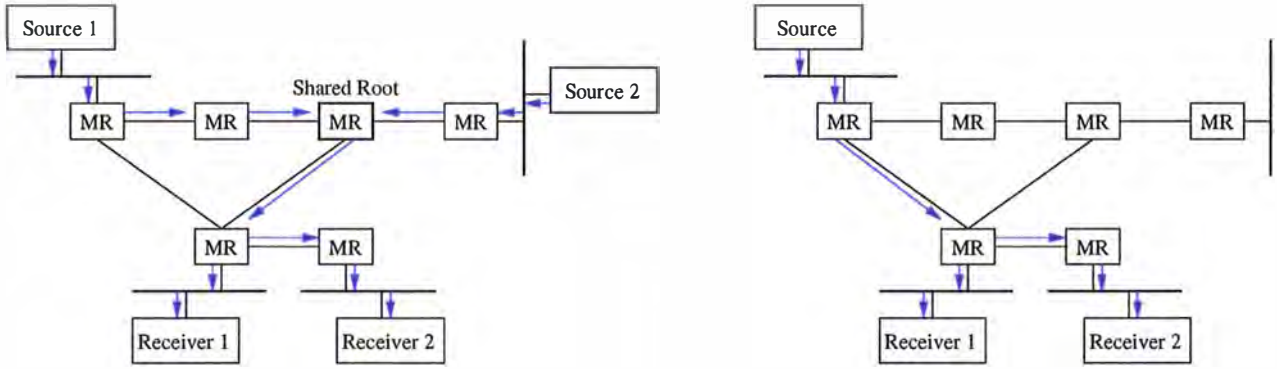

FIG. 3.1 – Arbre RPT (*,G) (à gauche) et Arbre SPT (S,G) (à droite)

l'arbre vers tous les récepteurs. Le routage est forcément moins efficace qu'avec des arbres SPT car les datagrammes font un détour par la racine mais au moins, on utilise un seul et même arbre, ce qui diminue le nombre d'états dans les routeurs. La figure 3.1.1 montre aussi un arbre partagé entre deux sources et deux récepteurs.

3.1.2 *Source specific multicast* versus *Any source multicast*

Le multicast classique est l'ASM (pour *Any source multicast*). Cela signifie que n'importe quel hôte peut émettre à destination d'un groupe donné, sans devoir s'inscrire préalablement d'une quelconque manière. Quand une source commence à émettre, elle doit aussi s'assurer que l'adresse de groupe utilisée n'est pas déjà employée par d'autres. Il n'y a donc aucune garantie sur les sources, ni sur l'allocation des adresses multicast. De plus, au fil des ans, on s'est rendu qu'il y avait beaucoup plus de communications 1-N que de communications N-N.

Ainsi est né le mode SSM (pour *Source Specific Multicast*). Dans ce mode, un service multicast est caractérisé par l'association d'une adresse de groupe G et de l'adresse unicast globale de la source S. Cette association (S,G) est appelée canal (*channel* en anglais). Cela résout de facto le problème de l'allocation des adresses, car la composante S du canal est globalement unique par définition des adresses IP publiques. Deux sources distinctes peuvent donc utiliser la même adresse de groupe sans entrer en conflit puisqu'il s'agira de deux canaux différents ((S_1,G) et (S_2,G)).

L'IANA a réservé une plage d'adresses multicast pour le déploiement exclusif de services IP multicast de type SSM. Il s'agit de la plage `232.0.0.0/8`. Le protocole PIM-SSM [4] a été développé pour fournir le service SSM efficacement et il est testé actuellement.

Ce type de protocole nécessite cependant la modification des protocoles permettant de joindre et quitter un groupe. En effet, au lieu de joindre simplement un groupe (*,G), les hôtes doivent aussi être capables de joindre un groupe et une source particulière (S,G). Ces nouvelles fonctions sont disponibles dans IGMPv3 [8] et MLDv2 [48].

Les protocoles SSM sont plus simples à déployer car leur architecture est moins complexe que les autres protocoles. Le succès commercial du multicast passera donc probablement par le déploiement d'un protocole *Source-Specific*.

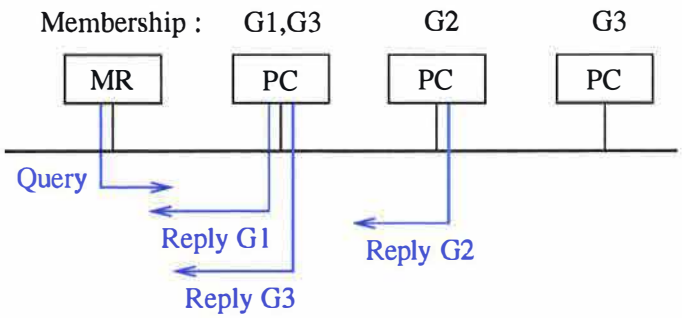

FIG. 3.2 – IGMP v1

3.1.3 Inscription et résiliation à un groupe

Dans tous les protocoles multicast développés par l'IETF, un hôte peut joindre un groupe existant sans demander d'autorisation et un membre du groupe peut aussi décider de quitter le groupe quand il le souhaite. N'importe quel hôte peut aussi décider de créer un groupe. Pour ce faire, on utilise le protocole IGMP en IPv4 [14, 24, 8], rebaptisé MLD en IPv6 [15, 48]. Les messages IGMP et MLD restent cantonnés au lien entre les hôtes et les routeurs d'accès, tandis que les protocoles de routage multicast se chargent d'acheminer efficacement les données sur le réseau. Ce sont donc les routeurs d'accès multicast qui se chargent de refléter les demandes IGMP et MLD dans leur protocole de routage multicast.

IGMPv1

Dans le protocole IGMPv1 [14], le routeur multicast demande périodiquement sur le lien les groupes qui intéressent les hôtes. Il envoie un message IGMP Query à l'adresse `224.0.0.1` (All Multicast Hosts) avec un TTL de 1. Il suffit qu'un hôte réponde pour un groupe donné. Si un hôte voit qu'un autre a déjà répondu pour un même groupe, il s'abstient d'émettre un message pour joindre le groupe. Les messages IGMP Reply sont envoyés à l'adresse du groupe correspondant. De ce fait, les réponses ne peuvent contenir qu'une seule adresse de groupe et le routeur multicast doit écouter toutes les adresses multicast pour recevoir les réponses (voir Figure 3.1.3).

Pour quitter un groupe, les hôtes n'émettent aucun message. Le routeur considère qu'il n'y a plus de membres du groupe sur le lien s'il n'a pas reçu de message IGMP Reply après un certain temps.

IGMPv2

IGMPv2 [24] est principalement une amélioration d'IGMPv1. Deux nouveaux mécanismes sont disponibles. Premièrement, un hôte peut signaler à son routeur d'accès qu'il n'est plus intéressé par un groupe particulier (IGMP Leave). Le message est envoyé à l'adresse `224.0.0.2` (All Multicast Routers).

Deuxièmement, un routeur peut envoyer un IGMP Query pour un groupe particulier, afin de savoir s'il reste des membres dans un certain groupe. Au lieu d'émettre le message à destination de tous les hôtes (`224.0.0.1`), le message est envoyé à l'adresse du groupe. Ainsi, seuls les membres du groupe traitent le message, ce qui est plus efficace.

IGMPv3

IGMPv3 [8] améliore encore le comportement de ses prédécesseurs. Le principal défaut des versions précédentes était que les routeurs multicast devaient écouter toutes les adresses multicast. Il n'était pas non plus possible pour un membre d'un groupe de filtrer les datagrammes pour une adresse source particulière. En effet, il n'y a aucun contrôle sur les émetteurs et les récepteurs dans les protocoles de routage multicast classique.

IGMPv3 est compatible avec ses prédécesseurs et supporte donc les Query génériques et les Query spécifiques à un groupe. De plus, il permet l'utilisation de filtres : une station peut limiter la réception d'un flux multicast en filtrant sur la valeur du champ adresse source des datagrammes multicast reçus. Les IGMP Query et Leave peuvent contenir deux types de filtres : soit une liste de sources désirées, soit une liste de sources non désirées. Un hôte peut donc recevoir les datagrammes de certaines sources ou de toutes les sources sauf certaines.

Dorénavant, tous les hôtes répondent aux IGMP Query par des IGMP Reply, ce qui permet au routeur Multicast de mieux comptabiliser les appartenances des hôtes aux groupes. Pour éviter des pointes de trafic après chaque IGMP Query, ceux-ci attendent un temps variable avant d'émettre leurs Replies. Chaque hôte émet un message contenant l'ensemble des groupes qui l'intéressent et non plus un groupe par message comme précédemment. Comme les hôtes émettent des IGMP Leave quand ils quittent un groupe, le routeur peut directement savoir s'il reste encore au moins une source et s'il est donc encore nécessaire d'émettre les datagrammes multicast du groupe sur le lien.

MLDv1 et MLDv2

MLDv1 [15] est l'adaptation en IPv6 d'IGMPv2 [24], tandis que MLDv2 [48] est l'adaptation d'IGMPv3 [8]. La principale différence est que MLD utilise des messages ICMPv6 au lieu de messages propres au protocole comme IGMP.

Remarque sur les *snooping switches*

Il existe sur le marché des switches Ethernet qui analysent les messages IGMP pour savoir sur quels ports se trouvent des membres de groupe multicast, afin de ne pas gaspiller la bande passante du LAN en broadcastant les datagrammes multicast sur tous les ports. Théoriquement, c'est pourtant ce qu'un switch devrait faire.

Outre le fait que ces switches empiètent sur le rôle des routeurs car ils analysent des données qui ne sont pas de leur ressort, la façon dont ils procèdent n'a pas non plus été formalisée par l'IETF. Comme l'IGMP-snooping n'est pas standardisé, les constructeurs de switches l'ont aussi implémenté à leur manière. Cela a donc entraîné des dysfonctionnements dans certains réseaux multicast, d'autant plus qu'IGMPv3 a entraîné des modifications dans la façon dont les messages sont échangés (tous les hôtes répondent aux Queries).

L'arrivée de MLD complique encore la tâche des switches car les messages MLD constituent un sous-ensemble des messages ICMPv6. Les switches ne peuvent donc pas détecter simplement un message MLD en regardant le type du protocole.

Pour résoudre ces problèmes, un draft est sorti cette année [10] pour tenter de mettre de l'ordre dans toutes ces pratiques. Dans la suite du travail, nous supposerons par simplicité que

les switches ne posent pas problème et qu'ils se contentent de faire ce pour quoi ils ont été conçus.

3.1.4 Dense mode et Sparse mode

Il existe deux grandes approches pour le routage multicast : Dense Mode et Sparse Mode. La première part de l'hypothèse que la distribution des membres sur le réseau est dense, c'est-à-dire que la plupart des sous-réseaux contiennent au moins un membre. Les protocoles de type Dense s'appuyent sur une technique d'inondation pour propager les informations à tous les routeurs du réseau. DVMRP [20] et PIM-DM [49] sont des exemples de protocoles de routage multicast de ce type.

La deuxième approche, Sparse Mode, suppose au contraire que les membres sont éparpillés sur le réseau et qu'il ne faut pas gaspiller la bande passante disponible avec une technique d'inondation comme en mode Dense. Les protocoles de type Sparse utilisent donc d'autres techniques pour propager les informations. CBT [2] et PIM-SM [22] sont deux exemples de protocoles de ce type.

3.2 Protocoles de type Dense

3.2.1 DVMRP

Le premier protocole de routage multicast inventé s'appelle DVMRP pour *Distance Vector Multicast Routing Protocol* [20]. Il a été beaucoup utilisé sur le réseau multicast IPv4 expérimental, appelé MBONE.

DVMRP construit un arbre différent pour chaque source et le groupe des destinataires. Chaque arbre distribué est un arbre de recouvrement minimum ayant comme racine la source et comme feuilles les membres du groupe. L'arbre permet d'avoir un plus court chemin entre la source et chaque receveur, en prenant comme métrique le nombre de sauts. Il est construit à la demande, en utilisant une technique *Flood and Prune*, quand une source commence à émettre des datagrammes à destination d'un groupe multicast.

L'approche utilisée par DVMRP est de supposer qu'initialement, tous les hôtes du réseau sont membres du groupe multicast. Le routeur d'accès qui est connecté à la source retransmet tous les datagrammes multicast de celle-ci aux routeurs adjacents. Ceux-ci font de même, en prenant garde d'éviter toute redondance pour finalement atteindre les membres du groupe (voir Figure 3.3).

Chaque routeur multicast utilise un algorithme pour décider sur quelles interfaces de sortie il doit retransmettre les datagrammes multicast qu'il a reçu, comme expliqué précédemment. Quand un routeur reçoit un datagramme multicast, il consulte sa table de routage unicast pour voir si le datagramme est arrivé par le plus court chemin jusqu'à la source. S'il en est ainsi, il retransmet le datagramme sur toutes ses interfaces, sauf celle d'où le datagramme provenait. Sinon, le routeur jette simplement le datagramme. Ce mécanisme, appelé *Reverse Path Forwarding*, assure qu'il n'y aura pas de boucle durant la formation de l'arbre et que les plus courts chemins entre la source et les membres du groupe seront utilisés pour propager les datagrammes multicast.

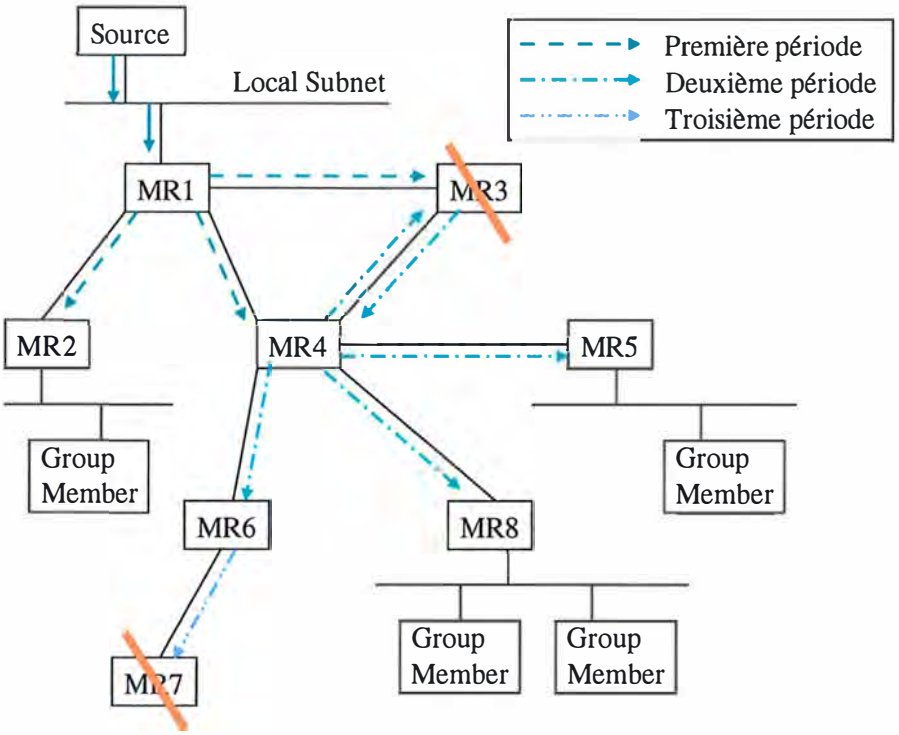

FIG. 3.3 – Construction d'un arbre distribué avec un mécanisme d'inondation

La partie Prune du protocole élimine les branches de l'arbre qui ne conduisent à aucun membre du groupe. Les hôtes utilisent le protocole IGMP pour signaler à leur routeur d'accès les groupes qui les intéressent. A partir de ces informations, le routeur qui constate qu'il n'y a aucun hôte membre d'un groupe donné, envoie un message Prune au routeur en amont, afin qu'il arrête de lui faire parvenir les datagrammes du groupe. Chaque routeur doit donc maintenir une table de routage pour savoir si certaines branches de l'arbre ont été coupées ou non. Le processus d'élagage continue jusqu'à ce qu'on obtienne un arbre distribué minimum, connectant la source aux membres du groupe (voir Figure 3.4).

Les hôtes du réseau peuvent joindre et quitter le groupe à tout moment. Ces changements doivent se refléter dans l'arbre. C'est pourquoi DVMRP reconstruit périodiquement l'arbre afin de tenir compte de la nouvelle distribution des membres.

DVMRP fonctionne bien pour un petit réseau où la distribution des membres est dense. Par contre, le processus d'inondation-élagage consomme beaucoup trop de bande passante pour un grand réseau comme Internet où les membres sont plus éparpillés. Les routeurs DVMRP doivent aussi maintenir des informations dans leur table de routage pour chaque couple (source,groupe). Pour toutes ces raisons, on peut dire que DVMRP ne convient pas pour un grand réseau.

3.2.2 PIM-DM

Protocol Independant Multicast - Dense Mode (PIM-DM) ressemble fortement à DVMRP. Les deux protocoles utilisent les mécanismes de *RPF Check* et de *Flood and Prune* pour construire un arbre de distribution dont la racine est la source.

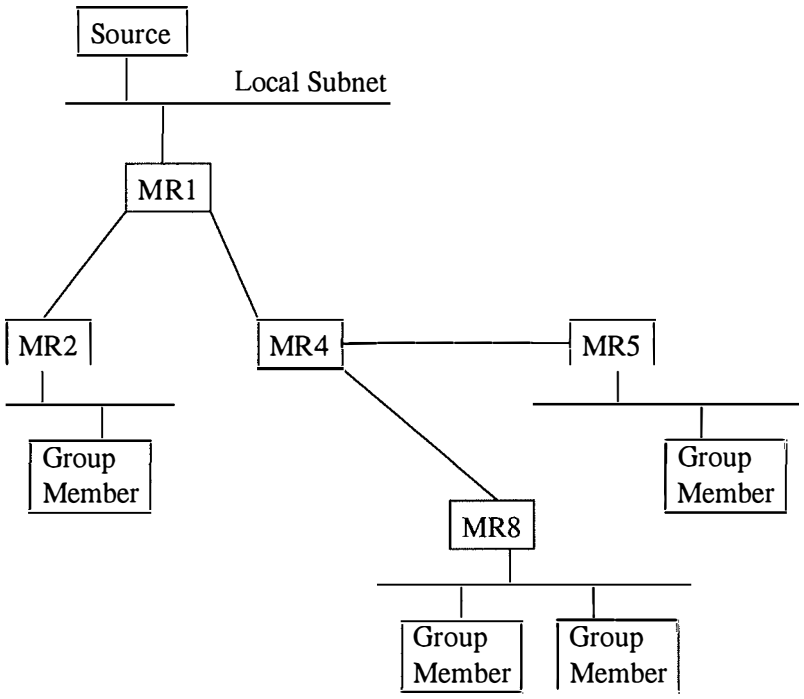

FIG. 3.4 – L'arbre distribué après élagage des branches inutiles

La grande différence entre DVMRP et PIM-DM est que PIM est totalement indépendant du protocole de routage unicast utilisé sur le réseau, alors que DVMRP utilise des mécanismes spécifiques du protocole de routage unicast. PIM-DM est aussi moins complexe que DVMRP. Les auteurs de PIM-DM ont privilégié la compatibilité et la simplicité, tandis que les créateurs de DVMRP ont privilégié l'efficacité. DVMRP est en fait capable d'utiliser des informations du protocole de routage unicast pour éviter d'émettre des datagrammes multicast sur une branche de l'arbre qui sera de toutes façons rapidement élaguée. Autrement dit, un routeur MR1 peut savoir qu'un routeur adjacent MR2 est sur le plus court chemin à la source. Ceci permet à DVMRP de réduire considérablement le nombre de messages de l'inondation.

3.3 Protocoles de type Sparse

Les protocoles précédents étaient conçus pour des réseaux de taille moyenne. Beaucoup de problèmes surviendraient si on les utilisait sur un grand réseau comme Internet avec par exemple, une centaine de groupes de vidéoconférence, où chaque groupe ne contiendrait que quelques membres. Les messages de contrôle induits par les mécanismes Flood et Prune périodiques auraient vite fait de dégrader les performances du réseau. Pour gérer les groupes multicast épars, il faut s'y prendre autrement afin que le trafic de contrôle ne se propage que sur les liens situés entre les membres du groupe.

3.3.1 CBT

Contrairement à DVMRP qui construit un arbre distribué pour chaque couple (Source, Groupe), *Core Based Trees* (CBT [2]) construit un seul arbre qui est partagé par tous les membres du groupe, peu importe le nombre de sources. Le fait d'utiliser un seul arbre permet déjà d'économiser beaucoup de ressources sur chaque routeur car le nombre d'états est

fortement diminué.

Les arbres partagés de CBT utilisent un routeur particulier, appelé le Core. Un routeur qui souhaite joindre l'arbre envoie un message Join au Core. A la réception du Join, le Core acquitte le message. L'acquittement, en cours de chemin, crée des états dans les routeurs intermédiaires afin d'étendre l'arbre distribué au nouveau routeur de façon optimale. En fait, les messages Join ne doivent pas obligatoirement arriver jusqu'au Core. Si le message arrive à un routeur qui est déjà membre de l'arbre, ce dernier peut directement acquitter le message et attacher le nouveau venu à l'arbre.

Un datagramme multicast émis par une des sources du groupe est progressivement acheminé jusqu'au Core, qui redistribue ensuite le datagramme sur le reste de l'arbre.

Le défaut de CBT est bien sûr que tout le trafic se concentre autour du Core. Certaines versions de CBT permettent d'utiliser plusieurs Core afin de faire du load-balancing.

3.3.2 PIM-SM

PIM-SM [22] ressemble globalement à CBT. Le Core s'appelle ici Rendezvous Point. Les arbres partagés qui étaient bidirectionnels dans CBT, sont unidirectionnels dans PIM. Cela signifie que dans CBT, les datagrammes peuvent se propager dans les deux sens le long de l'arbre, tandis qu'avec PIM, les datagrammes peuvent seulement partir de la racine et descendre jusqu'aux feuilles.

Le gros avantage de PIM-SM est sa flexibilité. En effet, dans CBT, les arbres sont toujours partagés entre toutes les sources, tandis qu'avec PIM-SM, un routeur peut choisir de construire un arbre de plus court chemin pour une source donnée. Ceci permet premièrement de soulager le Rendezvous Point et deuxièmement, d'avoir un routage plus optimal, car les arbres partagés n'offrent pas forcément un routage optimal.

PIM-SM construit d'abord un arbre partagé pour tous les membres du groupe comme CBT. Ensuite, un routeur peut décider de changer son mode de connexion avec une certaine source et opter pour un arbre de plus court chemin. Le routeur envoie alors un message Join (S,G) et un nouvel arbre est construit. Ce processus achevé, les branches de l'arbre partagé qui ne sont plus nécessaires sont élaguées.

Quand une source S commence à émettre à destination de G, le routeur d'accès de S encapsule les datagrammes à destination du Rendezvous Point dans des messages PIM Register. Ensuite, le RP joint directement l'arbre à la source en émettant un message PIM Join (S,G) en direction du routeur d'accès. Une fois que l'arbre à la source est construit et que les datagrammes arrivent nativement, le RP demande au routeur d'accès d'arrêter le tunnel en lui envoyant un message PIM Register-Stop.

Comme pour les autres protocoles, PIM-SM rafraîchit régulièrement ses informations pour suivre l'évolution des membres du groupe. Comme son nom l'indique, PIM est indépendant du protocole de routage unicast utilisé, même s'il a bien besoin des tables de routage unicast pour fonctionner. Il existe aussi des versions améliorées de PIM-SM où il est possible d'avoir plusieurs Rendezvous Point, afin d'éviter les points centraux d'échec.

PIM-SM tend progressivement à devenir le leader des protocoles de routage multicast. Il est déjà déployé dans de grands réseaux backbone internationaux comme par exemple OpenTransit

(France Telecom). Plusieurs Rendezvous Points ont été configurés selon le mode anycast dans ce réseau afin d'optimiser le routage et d'augmenter la redondance. Des accords de peering multicast ont aussi été conclus avec d'autres grands opérateurs (Sprint, vBNS). PIM-SM a aussi été déployé dans le réseau Intranet de France Telecom R&D. On peut donc s'attendre à voir apparaître des offres multicast à base de PIM pour les entreprises dans les prochains mois.

3.3.3 PIM-SSM

PIM-SSM est une adaptation de PIM-SM destinée à le faire fonctionner en mode *Source-Specific* au lieu du mode *Any Source*. Les modifications apportées sont mineures. Elles permettent d'éviter la construction d'arbres RPT (*Rendezvous Point Tree*) en construisant directement un arbre SPT (*Shortest Path Tree*) qui est la deuxième étape du protocole PIM-SM. Les routeurs ne créent pas d'état (*,G) et n'émettent pas de messages Join (*,G) pour la construction d'arbre RPT. Ils ne doivent pas non plus émettre de message Register à destination du Rendezvous Point. PIM-SSM réduit donc considérablement le trafic de contrôle et le nombre d'états à maintenir sur chaque routeur.

Deuxième partie

Support du multicast pour les hôtes mobiles

Chapitre 4

Problème

Les protocoles de routage multicast ont été conçus en partant de l'hypothèse que tous les hôtes sont fixes. Regardons en détail ce qui pose problème en l'état actuel quand un hôte mobile souhaite émettre et recevoir du trafic multicast. Ceci nous permettra de mieux comprendre les solutions qui seront décrites dans le chapitre suivant.

4.1 Réception

Supposons qu'un mobile s'abonne à un groupe multicast (G) dans son réseau Home. Comme nous l'avons vu précédemment, le protocole de routage multicast construit un arbre, dont le mobile est une feuille. Cet arbre permet de router de façon optimale les datagrammes destinés au groupe de la source jusqu'aux membres.

Supposons maintenant que le mobile se déplace et s'attache à un autre routeur d'accès, situé dans un autre sous-réseau. Cinq problèmes peuvent survenir.

Premièrement, les datagrammes multicast destinés à G continuent d'affluer vers l'ancien sous-réseau et n'atteignent donc plus le mobile. Si le mobile était le seul membre de G dans l'ancien sous-réseau, tous ces datagrammes sont transmis inutilement. Ce trafic inutile dure jusqu'à ce que le routeur multicast du réseau Home se rende compte qu'il n'y a plus aucun hôte qui est intéressé par G. Cela prend donc au maximum la période entre deux Query IGMP/MLD, plus le timeout après lequel le routeur multicast n'attend plus de réponses de la part des hôtes.

Deuxièmement, il se peut aussi qu'aucun hôte du réseau visité ne soit membre de G. Le mobile ne reçoit donc plus les datagrammes de G et doit attendre le prochain Query du routeur d'accès du réseau visité pour refaire partie du groupe. Ensuite, le routeur d'accès joint l'arbre distribué selon les règles définies dans son protocole de routage multicast, ce qui peut prendre un temps non négligeable. Finalement, les datagrammes de G arrivent de façon native par le nouveau routeur d'accès qui fait maintenant partie de l'arbre distribué. La rupture de service est donc beaucoup trop longue. La plupart des applications ne pourront pas supporter une coupure aussi grande et la communication multipoint s'en trouvera avortée.

Troisièmement, certains routeurs multicast IPv4 ne retransmettent les datagrammes multicast que si le *Time To Live* (TTL) est supérieur à une valeur donnée. Il se peut donc qu'un mobile entre dans un nouveau réseau qui se situe à un trop grand nombre de sauts de la source, et perde ainsi tout contact avec le groupe.

Quatrièmement, il se peut très bien que le réseau visité par le mobile n'offre pas de service multicast. Le mobile perd alors tout contact avec son groupe et la communication est perdue.

Cinquièmement, l'introduction des scopes en IPv6 pose un nouveau problème. Si un mobile s'abonne à un groupe organization-local et qu'il quitte son organisation, la communication sera coupée. Ce n'est peut-être pas ce qui est désiré par les utilisateurs. Par exemple, une personne participant à une vidéoconférence en multicast disponible dans son organisation pourrait, durant la conférence, quitter son entreprise et s'attacher à un autre réseau (par exemple le réseau sans-fil d'un train). Il souhaiterait certainement que la communication perdure, même si les datagrammes de la vidéoconférence ne sont pas sensés sortir de l'entreprise. La solution la plus simple serait d'utiliser une adresse de groupe de scope global mais c'est risqué d'un point de vue sécurité (une entreprise concurrente pourrait capturer les datagrammes de la conférence). De plus, les scopes perdraient de leur utilité car au moment de créer un groupe, on choisirait toujours une adresse de scope global au cas où un des membres quitterait le scope.

4.2 Émission

Supposons maintenant qu'un mobile émette du trafic multicast depuis son réseau Home, puis qu'il se déplace dans un autre réseau, tout en conservant son adresse.

On retrouve le même problème qu'en mode unicast. L'adresse source avec laquelle sont émis les datagrammes dans le nouveau réseau est topologiquement incorrecte. Si le routeur du lien local vérifie l'adresse source des datagrammes, ceux-ci seront jetés.

De plus, les routeurs multicast utilisent l'algorithme de *Reverse Path Forwarding Check* décrit précédemment. Si un datagramme multicast ne provient pas de l'interface qui mène à la source par le plus court chemin, il est jeté. Or, cela risque forcément d'arriver très souvent si le mobile émet ses datagrammes avec la même adresse source depuis un autre réseau (voir Figure 4.1).

Nous en déduisons que le mobile ne peut pas utiliser continuellement la même adresse source pour émettre ses datagrammes multicast. Il doit acquérir une nouvelle adresse appelée Care-of Address (CoA).

Si, le mobile utilise simplement sa CoA dans l'adresse source de ses datagrammes multicast, il sera tout d'abord identifié comme une autre source. Ceci pourrait entraîner des problèmes au niveau applicatif. En effet, une application multicast simple pourrait identifier les différents flux d'un groupe par l'adresse IP source. Par exemple, pour une vidéoconférence à cinq personnes, l'application doit afficher les cinq flux vidéos dans cinq fenêtres différentes. Pour savoir de quelle personne provient une image, elle pourrait se baser sur l'adresse IP de l'émetteur. Donc, si un émetteur est un mobile qui change d'adresse, il sera reconnu par l'application comme étant un nouveau participant à la vidéoconférence.

Le fait de changer d'adresse pose aussi un problème de taille. Les protocoles de routage multicast construisent souvent des arbres distribués dont la racine est la source. Si la source se déplace et change d'adresse, le protocole sera fortement mis à contribution car il faudra reconstruire un tout nouvel arbre et il faudra aussi invalider le précédent. Cela va donc entraîner un trafic de contrôle énorme, consommer des ressources supplémentaires sur les routeurs

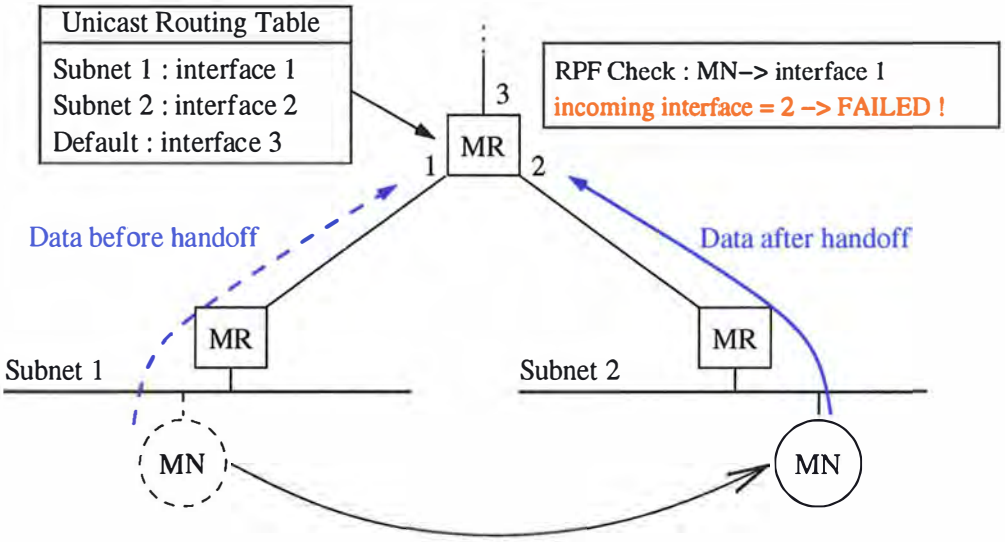

FIG. 4.1 – Echec du RPF Check quand le mobile garde son adresse

multicast et prendre un temps non négligeable pour se stabiliser. Si les déplacements du mobile sont fréquents, les performances du réseau risquent de chuter rapidement.

Chapitre 5

Revue des solutions existantes

5.1 Introduction

Maintenant que le problème a été posé, il est temps de rentrer dans le vif du sujet. Ce chapitre contient huit solutions développées par différents chercheurs. Chacune est décrite puis critiquée et comparée objectivement aux autres. Ce chapitre se termine par un tableau comparatif des huit solutions reprenant les différents paramètres significatifs de celles-ci.

Les premiers à avoir tenté de résoudre le problème étaient Acharya et Badrinath en 1996 ([1]). A cette époque, Mobile IP n'était encore qu'une ébauche de protocole. Leur description ne correspond plus vraiment à la réalité, c'est pourquoi elle ne fait pas partie des solutions sélectionnées.

D'autres chercheurs ont aussi voulu intégrer dans leur solution la fiabilité ([50, 9]). Elles ne sont pas non plus reprises ici car IP est fondamentalement non fiable. Les datagrammes peuvent être perdus, réordonnés ou dupliqués durant leur voyage. En mode unicast, c'est TCP qui se charge de combler ces lacunes. Il nous semble donc préférable de laisser un autre protocole situé dans une couche supérieure s'occuper de la fiabilité en multicast.

Deux autres solutions n'ont pas été reprises dans ce document car elles étaient trop basiques et ne proposaient rien de vraiment intéressant ([33, 41]).

5.2 Remote Subscription - Bidirectionnal Tunneling

La première solution pour permettre aux hôtes mobiles d'émettre et de recevoir du trafic multicast est d'établir un tunnel bidirectionnel avec le Home Agent, afin de rendre la mobilité transparente pour les autres membres du groupe. Cette solution est décrite brièvement par l'IETF dans Mobile IPv4 [40, section 4.4] et Mobile IPv6 [30, section 10.19].

Le tunnel est créé entre le Home Agent et la Care-of Address. En IPv4, la CoA désigne le Foreign Agent (FACoA) ou le mobile lui-même (CoCoA). Tandis qu'en IPv6, il s'agit toujours du mobile.

5.2.1 Réception de trafic multicast

Quand le mobile est sur son réseau Home, il se comporte comme tout autre noeud. Supposons qu'il se trouve dans un réseau visité. Pour signaler à son Home Agent les groupes qui

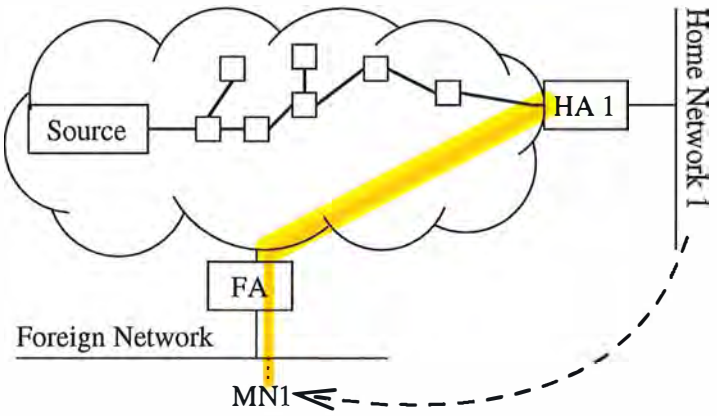

FIG. 5.1 – Double tunnel dans Mobile IPv4 avec Foreign Agent

FIG. 5.2 – Datagramme multicast encapsulé dans un datagramme unicast à destination du Home Agent

l'intéressent, il va envoyer dans le tunnel ses messages IGMP (MLD en IPv6). Le Home Agent s'abonne à la place du mobile et se charge de retransmettre le trafic multicast voulu dans le tunnel vers le mobile.

Concrètement, dans le cas IPv4 avec CoCoA ou IPv6, le Home Agent encapsule le datagramme multicast dans un autre datagramme unicast, à destination de la CoA. Le mobile qui reçoit le datagramme le décapsule et traite le datagramme multicast.

Dans le cas FACoA, le Home Agent doit procéder à une double encapsulation : premièrement encapsuler le datagramme multicast dans un datagramme unicast à destination de la Home Address du mobile, et deuxièmement, réencapsuler le tout dans un datagramme unicast à destination de la FACoA. Le FA qui reçoit ce datagramme procède à la première décapsulation puis le retransmet au mobile. Ce dernier le décapsule à son tour, pour découvrir le datagramme multicast (voir Figure 5.1).

5.2.2 Emission de trafic multicast

En IPv4, le mobile émet ses datagrammes multicast en utilisant le tunnel inverse, comme il peut aussi le faire en mode unicast. Dans le cas avec Foreign Agent, deux modes s'offrent à lui : soit le mobile encapsule lui-même ses datagrammes, soit il demande au Foreign Agent de s'en charger. Le choix du mode se fait par l'utilisation de l'extension "Delivery Style" lors de l'enregistrement du mobile auprès du FA.

En IPv6, le mobile encapsule son datagramme multicast dans un autre datagramme ayant comme adresse source sa CoA, et comme adresse destination l'adresse du Home Agent (voir Figure 5.2).

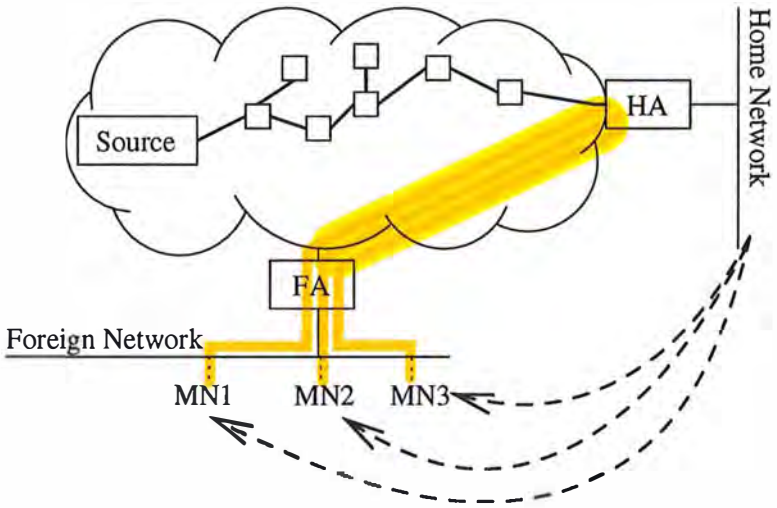

FIG. 5.3 – Problème des tunnels parallèles

5.2.3 Critique

Le plus gros avantage du *Bidirectionnal Tunneling* est sans aucune doute la transparence induite par les tunnels. Les autres membres du groupe et les routeurs multicast n'ont jamais connaissance du changement d'adresse du mobile, ce qui évite beaucoup de problèmes. Cette technique fonctionne aussi dans les réseaux visités qui n'offrent pas le multicast car tout le trafic multicast est tunnelé sous la forme de trafic unicast via le Home Agent. Ceci a justement comme inconvénient que le Home Agent devient très sollicité vu qu'il doit assurer le suivi des communications unicast et multicast même si les mobiles en déplacement ne changent pas de sous-réseau durant la communication. De plus, le routage en triangle est très inefficace et les traitements des paquets sont lourds vu qu'il faut continuellement les encapsuler et les décapsuler.

Problème des tunnels parallèles

Ce problème survient en Remote Subscription quand plusieurs mobiles du même réseau Home visitent le même réseau et sont abonnés au même groupe. Si les tunnels vont du Home Agent jusqu'à chaque mobile (IPv4 CoCoA et IPv6), le Home Agent qui reçoit un datagramme multicast doit tunneler successivement pour chaque CoA, ce qui résulte en une perte flagrante d'efficacité (voir Figure 5.3). Le problème existe aussi quand le tunnel se situe entre le Home et le Foreign Agent car les datagrammes sont doublement encapsulés !

Problème de la convergence des tunnels

Supposons que deux mobiles de réseaux Home différents arrivent dans un même réseau visité et soient abonnés au même groupe. Les deux Home Agents vont tunneler les datagrammes vers le même réseau, ce qui est inefficace (voir Figure 5.4). Ce le sera d'autant plus, s'il y a déjà des hôtes sur le réseau visité qui sont abonnés de façon native au même groupe. De fait, les mêmes datagrammes arriveront par les deux tunnels et aussi de facon native. Toutefois, les datagrammes provenant des tunnels et diffusés sur le lien sont encore encapsulés une fois, à destination unicast du mobile.

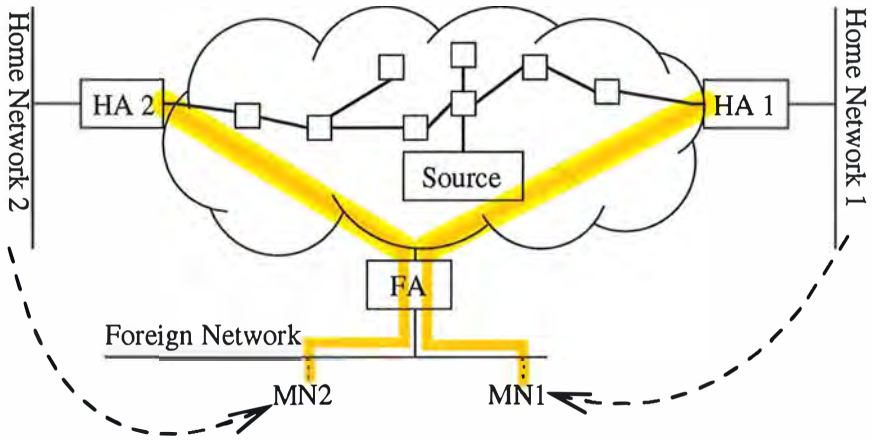

FIG. 5.4 – Problème de convergence des tunnels des HA vers le même FA

5.3 Mobile Multicast Protocol - MoM

Il existe une différence importante entre ce qui est expliqué dans le standard Mobile IPv4 [40] et l'explication de ce même standard donnée dans le document décrivant MoM [26]. Concernant la réception de datagrammes multicast, les auteurs de MoM pensent que le Home Agent ne tunnelle qu'une seule fois à destination du Foreign Agent, et que celui-ci, après décapsulation, retransmet les datagrammes en multicast sur le lien local. Cette différence élimine directement le problème des tunnels parallèles car la retransmission multicast sur le lien permet à plusieurs mobiles abonnés au même groupe de profiter du datagramme. On voit bien que cette nouvelle description est plus intéressante que ce qui est défini dans Mobile IP mais elle induit un problème particulier qui sera expliqué plus loin.

Néanmoins, le problème de la convergence des tunnels subsiste et c'est ce que MoM permet de résoudre. Pour éviter que plusieurs HA ne retransmettent les mêmes datagrammes multicast vers un même Foreign Agent, un système de sélection de Home Agent est mis en place. Le Foreign Agent désigne un des Home Agents comme étant le *Designated Multicast Service Provider* (DMSP) pour un groupe multicast donné. Le HA qui est DMSP retransmet les datagrammes multicast une seule fois dans le tunnel, même s'il a sous sa responsabilité plusieurs mobiles dans le réseau visité à l'autre bout du tunnel. Les autres HA qui ne sont pas DMSP sont désactivés. Ils ne doivent plus retransmettre de datagramme multicast dans leur tunnel (voir Figure 5.5).

L'utilisation de DMSP pose un nouveau problème. Quand un mobile quitte le réseau visité alors que son HA était le DMSP, le FA doit procéder à un changement de DMSP (*DMSP Handoff*). En effet, le DMSP n'a plus aucune raison de tunneler des datagrammes vers le Foreign Network puisqu'il n'a plus aucun mobile sur ce réseau visité. Comme il reste des mobiles appartenant à d'autres HA sur le FN, il faut qu'un des HA reprenne du service pour servir les mobiles restants. Etant donné que le mobile qui a changé de réseau ne signale pas à son ancien Foreign Agent qu'il s'est déplacé, le Foreign Agent doit attendre le timeout pour se rendre compte qu'il faut changer de DMSP. Plusieurs algorithmes de choix de DMSP sont possibles et ont été évalués dans MoM [26].

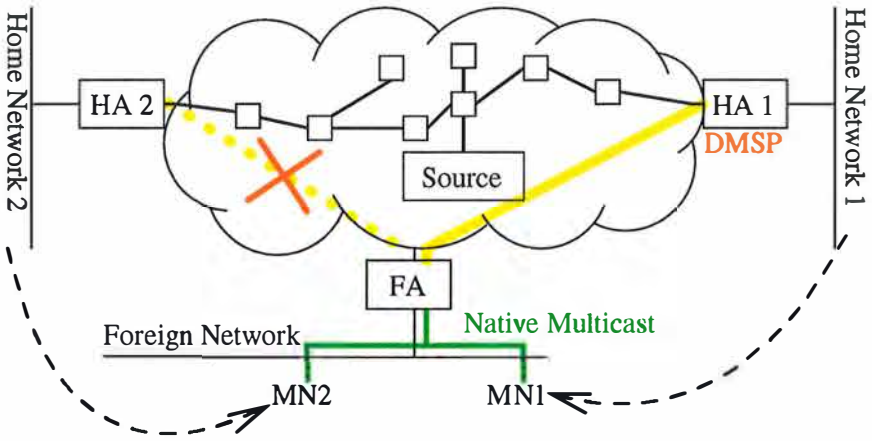

FIG. 5.5 – MoM - Le FA désigne HA1 comme DMSP

5.3.1 Critique

MoM est une optimisation du Bidirectionnal Tunneling de Mobile IP. Il est donc forcément plus efficace grâce à l'utilisation des DMSP. Il possède les mêmes avantages comme la transparence et la simplicité.

Ce n'est pourtant pas la solution idéale. Il est toujours possible d'avoir des duplications de trafic : il suffit qu'un hôte sur le FN soit abonné de façon native au même groupe que les mobiles en visite. Il est possible de résoudre cela simplement en permettant au FA de désactiver tous les HA pour un groupe G si le FA reçoit déjà les datagrammes multicast de G de façon native. Les autres inconvénients de MoM sont que le protocole nécessite l'utilisation de Foreign Agents (inexistants en IPv6) et que les tunnels sont permanents, ce qui produit un routage triangulaire inefficace.

Comme annoncé plus haut, un problème particulier découle du fait que le FA retransmet nativement les datagrammes multicast décapsulés sur le lien où se situe le mobile (contrairement à ce qui est décrit dans Mobile IPv4). Prenons un exemple pour illustrer le problème (voir Figure 5.6). Si le HA se trouve en aval du FA sur l'arbre multicast et que le mobile se trouve sur le lien reliant le FA au routeur multicast en aval, il va y avoir un bouclage des datagrammes. En effet, les datagrammes sont retransmis d'abord nativement sur le lien. Le mobile les traite, tout comme le routeur multicast qui se trouve sur le même lien. Ce dernier les retransmet en aval sur l'arbre. Ils finissent par arriver au Home Agent, qui les tunnelle vers le FA. Celui-ci les décapsule, les retransmet de nouveau nativement sur le lien où se trouve le mobile. Le mobile reçoit de nouveau les datagrammes, tout comme le routeur multicast qui les retransmet aveuglément en aval puisqu'ils proviennent de la bonne branche (RPF Check). Ce processus continue jusqu'à ce que le TTL des datagrammes atteigne 0. Ils sont alors jetés.

Une solution à ce problème est d'obliger le FA à émettre les datagrammes décapsulés avec un TTL de 1. De cette façon, le routeur multicast situé sur le même lien que le mobile est obligé de jeter les datagrammes vu qu'ils ont un TTL nul après décrémentation.

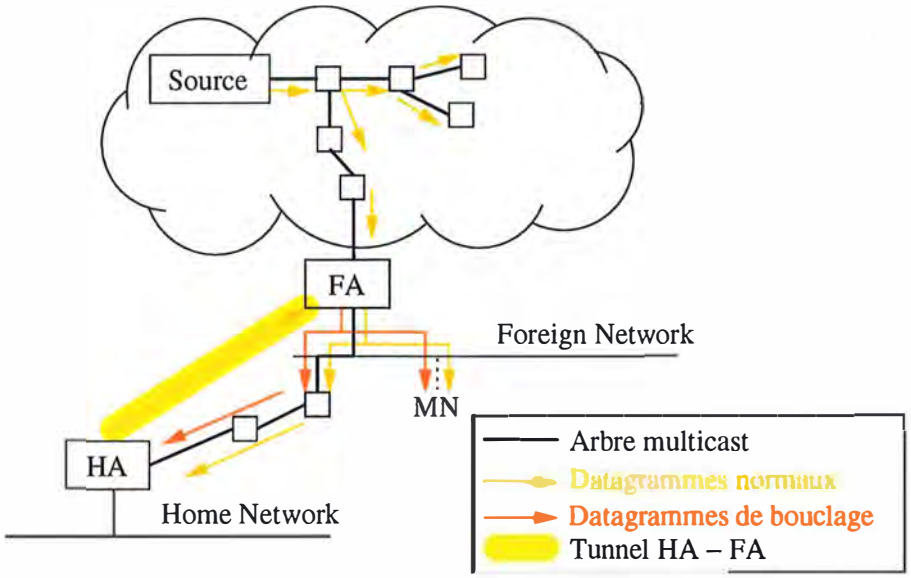

FIG. 5.6 – MoM - Boucle dans le routage des datagrammes

5.4 Local Subscription - Remote Membership

Le *Local Subscription* est la deuxième solution simple de l'IETF permettant aux mobiles de recevoir et d'émettre des datagrammes multicast.

5.4.1 Réception de trafic multicast

Le mobile qui arrive sur un réseau visité s'enregistre de façon locale aux groupes pour lesquels il s'était déjà abonné dans son réseau précédent. Cela suppose bien entendu qu'un routeur multicast soit disponible sur chaque réseau visité. Pour l'émission de messages IGMP (IPv4), le mobile devrait utiliser sa CoCoA. Sinon, il peut toujours utiliser sa Home Address. Le fait de s'abonner localement permet d'obtenir un routage optimal. Malheureusement, si personne n'était abonné au même groupe que le mobile, le temps nécessaire au protocole de routage multicast pour étendre l'arbre distribué jusqu'au routeur multicast du réseau visité pourrait être très élevé.

Le mobile risque aussi d'attendre très longtemps les IGMP/MLD Queries car celles-ci n'arrivent par défaut que toutes les 125 secondes. Pour contourner ce problème, le mobile peut néanmoins (mais c'est vivement conseillé dans ce cas) émettre un IGMP/MLD Join non sollicité. Cela signifie que le mobile ne va pas attendre le Query pour envoyer son Join.

5.4.2 Emission de trafic multicast

Le fait d'émettre avec une nouvelle adresse source (la Colocated Care-of Address ou la Care-of Address) du trafic multicast va déclencher la création d'un nouvel arbre multicast, ce qui est une opération plus ou moins coûteuse suivant le protocole multicast utilisé.

De plus, le mobile va apparaître aux yeux des récepteurs multicast comme un nouvel émetteur puisque l'adresse source est différente. Ceci pourrait mener, suivant l'application, à une rupture de service, si l'adresse source a de l'importance.

Le mobile ne peut pas émettre les datagrammes multicast avec sa Home Address car ses datagrammes seraient jetés par les routeurs multicast puisqu'ils arrivent d'une branche de l'arbre qui ne mène pas à la source (RPF Check).

Néanmoins, en IPv6, il est toujours possible au mobile de rajouter une option Home Address au datagramme pour permettre aux récepteurs de substituer l'adresse source du paquet avec la Home Address (comme en Unicast). Remarquons que les protocoles de routage multicast ne sont pas conçus pour tenir compte des options de destination Home Address et que par conséquent, tout changement du champ adresse source du datagramme multicast entraîne la formation d'un nouvel arbre multicast, même si une Home Address est spécifiée. De toutes façons, il ne serait d'aucune utilité que les protocoles de routage multicast tiennent compte de cette option car alors, l'algorithme de RPF Check serait inutilisable. Si une source peut se situer à divers endroits de l'arbre, il est impossible pour un routeur de décider sur quelles branches il doit retransmettre un datagramme donné.

5.4.3 Critique

Comme on le voit, le *Remote Membership* est diamétralement opposé au *Bidirectionnal Tunneling*. Le principal avantage de cette méthode est bien sûr qu'elle offre un routage optimal mais cette exigence a un coût. Le délai nécessaire à la reconstruction ou à l'extension de l'arbre distribué pour inclure le réseau visité par le mobile peut être prohibitif selon le protocole de routage multicast utilisé et la configuration de l'arbre. De plus, il faut qu'un routeur multicast soit présent sur chaque réseau visité. Cette méthode a l'avantage d'être équivalente en IPv4 et IPv6 car elle ne repose pas sur l'utilisation de Foreign Agents comme le *Bidirectionnal Tunneling* et MoM.

5.5 Mélange de Local et Remote Membership

Dans [3], les auteurs font remarquer que l'on peut mélanger *Local* et *Remote Membership* pour l'émission et la réception. Par exemple, dans une topologie à base de PIM Dense Mode, les auteurs jugent plus intéressant le *Remote Membership* pour la réception, mais ils préfèrent le tunneling vers le Home Agent pour l'émission, afin de ne pas perturber l'adresse source. En effet, sans tunneling, l'adresse source est modifiée dans les datagrammes puisqu'elle est remplacée par la CoA. Le mobile est alors considéré comme une nouvelle source par les routeurs multicast, ce qui déclenche le processus très lourd d'inondation de PIM-DM.

5.6 Range Based Mobile Multicast - RBMoM

5.6.1 Introduction

RBMoM [11] est une amélioration de MoM et une généralisation des théories précédentes pour IPv4. Nous verrons que le *Remote Subscription* et le *Local Subscription* sont les extrêmes de RBMoM. Par la suite, on utilisera l'abbréviation MHA pour parler d'un Multicast Home Agent, c'est-à-dire un Home Agent qui est aussi un routeur multicast.

L'idée à la base de RBMoM est d'utiliser une "limite de service" (*service range*) afin de limiter la longueur des chemins empruntés par les datagrammes multicast et de contrôler la fréquence de reconstruction des arbres multicast. RBMoM s'emploie à trouver le juste milieu

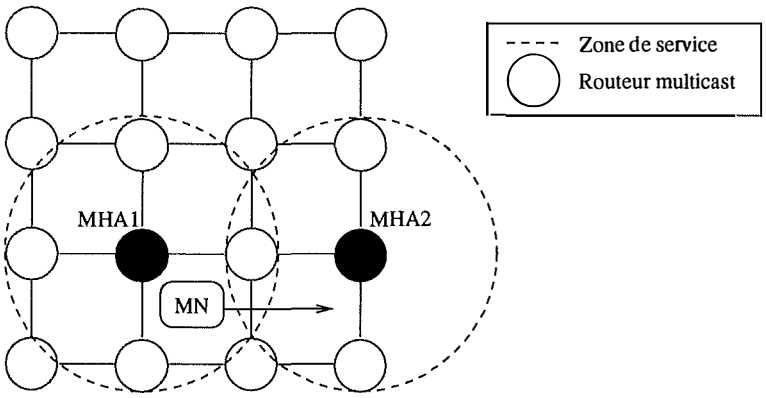

FIG. 5.7 – RBMoM

entre *Local* et *Remote Subscription*, c'est-à-dire en essayant de ne pas reconstruire l'arbre multicast à chaque changement de réseau et en même temps, en essayant d'obtenir un routage presque optimal.

5.6.2 Fonctionnement du protocole

Chaque mobile ne peut avoir qu'un seul Home Agent (HA), mais il peut avoir plusieurs MHA successivement. Le HA s'occupe du routage unicast vers le mobile, tandis que le MHA s'occupe du routage multicast. Le MHA initial du mobile est le HA. Chaque MHA sert ses mobiles tant qu'ils restent dans sa "zone de service". Si un mobile sort de la zone de service de son MHA, un autre MHA doit prendre le relais (on parle alors de *MHA Handoff*). La zone de service est un concept qui peut prendre diverses formes. Ici, par souci de simplicité, on définit la zone de service comme étant le nombre de sauts maximum entre le mobile et le MHA.

Si la zone de service est infinie, on se trouve dans le cas du *Remote Subscription* car le HA reste constamment le MHA ; à l'opposé, si la zone de service est nulle, on se trouve dans le cas du *Local Subscription* car il faudra changer de MHA à chaque handoff du mobile. La limite de service correspond donc à la longueur maximale du tunnel entre le MHA et le mobile.

Le HA sait pour chaque mobile sous sa responsabilité quel est son MHA courant. Quand un mobile arrive sur un nouveau réseau, il s'enregistre auprès du Foreign Agent (FA) avec Mobile IP. Le FA contacte alors le HA pour signaler que le mobile se trouve dans son réseau et pour connaitre le MHA du mobile. Le FA calcule ensuite la distance (le nombre de sauts) entre le mobile et le MHA.

Si cette distance est plus grande que la limite choisie, un nouveau MHA doit être sélectionné. Par simplicité, on choisit le FA comme nouvel MHA. Celui-ci, pour assumer son nouveau rôle, doit s'inscrire aux groupes multicast dont le mobile est membre. Le FA (également MHA) doit ensuite signaler au HA qu'il prend en charge le mobile.

Si, au contraire, la distance calculée par le FA est inférieure ou égale à la limite choisie, le FA ne fait rien de particulier. Il laisse le MHA courant du mobile continuer son travail.

Par optimisation, on peut introduire une règle d'exception : si le FA est déjà abonné aux groupes multicast, il peut initier un changement de MHA même si la distance calculée est inférieure à la limite.

Chaque MHA maintient une liste des mobiles dont il s'occupe. Quand un MHA s'occupe d'un nouveau mobile, il faut qu'il signale à l'ancien MHA du mobile d'arrêter le service pour ce mobile.

Notons que le FA peut décider de redonner le flambeau au HA s'il constate que la distance entre le mobile et le HA est redescendue en dessous de la limite.

Le problème de la convergence des tunnels survient aussi avec RBMoM mais il est toujours possible d'appliquer la méthode des DMSP de MoM. Concrètement, pour éviter que plusieurs datagrammes multicast identiques destinés au même groupe et provenant de différents MHA n'arrivent vers le FA, le FA désigne un des MHA comme étant le DMSP (pour un certain groupe) et interdit aux autres MHA de transmettre leurs datagrammes. RBMoM propose une petite amélioration à ce mécanisme : comme le FA peut lui aussi être MHA, il peut s'autodésigner comme DMSP et désactiver tous les autres MHA.

5.6.3 Critique

RBMoM est très intéressant du fait qu'il est l'intermédiaire entre les solutions proposées précédemment. Cette méthode offre un routage quasi optimal et évite de surcharger le HA du mobile. La complexité est légèrement accrue mais toujours acceptable. Elle est aussi évolutive car on peut redéfinir la notion de *zone de service* et améliorer l'algorithme de changement de MHA. Il reste cependant difficile de choisir le nombre de sauts maximum à autoriser avec la définition actuelle. Cela dépend de beaucoup de paramètres, comme le nombre de handoffs par seconde, la topologie du réseau, etc.

Comme MoM, RBMoM repose sur l'utilisation de Foreign Agents, ce qui pose problème pour l'adaptation en IPv6. Contrairement aux protocoles précédents, RBMoM ne s'occupe que de fournir une réception multicast efficace aux mobiles. Autrement dit, il ne propose rien pour l'émission.

Il faut aussi remarquer qu'un MHA est responsable de tous les groupes auxquels est abonné un mobile. Un mobile ne peut donc pas avoir un certain MHA pour un groupe et un autre MHA pour un autre groupe. Supposons qu'un groupe soit retransmis nativement sur un réseau visité et que le FA ne soit pas MHA du mobile en visite. Le MHA retransmettra de façon redondante dans un tunnel les datagrammes multicast qui arrivent déjà nativement.

5.7 Multicast by Multicast Agent - MMA

5.7.1 Description

Le protocole MMA [27] introduit deux entités : le Multicast Agent (MA) et le Multicast Forwarder (MF). Les MA fournissent le service multicast aux noeuds mobiles (comme les Foreign Agents dans les solutions précédentes). Chaque MA possède un MF par groupe multicast, c'est-à-dire un MA qui lui transmet les datagrammes du groupe dans un tunnel. Le MF d'un MA pour un certain groupe peut être le MA lui-même, si le MA reçoit le groupe nativement via le protocole de routage multicast. Sinon, il demande un tunnel à un autre MA qui devient son MF (voir Figure 5.8).

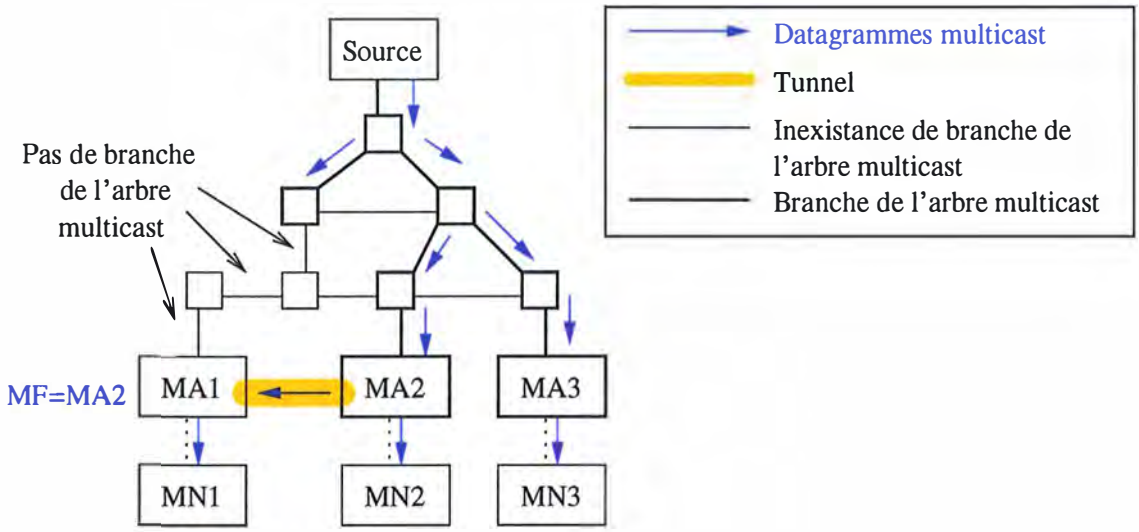

FIG. 5.8 – MMA

Chaque mobile sait pour chaque groupe auquel il est abonné quel est le MF courant. Quand le mobile arrive dans un autre réseau, il signale au MA les groupes qu'il recevait et les MF associés. Sur base de ces nouvelles informations et de celles qu'il possédait déjà, le MA choisit un MF pour chaque groupe, mais son choix peut retomber sur le MF qu'il utilisait précédemment. Ensuite, le MA signale aux mobiles en visite sur son réseau les changements de MF s'il y en a eu.

S'il y a changement de MF pour un certain groupe pour une quelconque raison, le MA émet un message *Forwarding Request* au nouveau MF et un message *Forwarding Stop* à l'ancien MF. Evidemment, si le MF pour ce groupe était le MA lui-même, il n'est pas nécessaire d'émettre de messages.

Pour un groupe donné, chaque MA maintient la liste des mobiles intéressés par celui-ci, la liste des MA pour lesquels il faut tunneler, le nombre d'hôtes statiques membres du groupe et l'adresse du MF s'il y en a un. Quand un datagramme du groupe arrive via la bonne branche de l'arbre de distribution multicast, il est retransmis nativement à tous les membres du groupe (hôtes statiques, noeuds mobiles), tout comme il est tunnelé vers tous les MA qui l'ont demandé (par un *Forwarding Request*).

Un mobile qui souhaiterait un délai plus faible peut demander explicitement au MA visité de joindre l'arbre multicast.

5.7.2 Critique

MMA ressemble à RBMoM vu que l'on recherche aussi le juste milieu entre la reconstruction d'arbres multicast et l'optimalité du routage. Mais ici, il n'y a pas de limite de zone en dehors de laquelle on change de MA. Le routage multicast est complètement désolidarisé du routage unicast car le Home Agent ne sait pas quel est le routeur qui s'occupe du trafic multicast de son mobile. Ici, c'est le mobile lui-même qui signale au MA les informations utiles. Le protocole MMA part aussi du principe qu'il vaut mieux tunneler du routeur précédent quand le routeur présent n'est pas encore abonné au groupe, afin de diminuer le handoff, ce qui n'est pas du tout pris en compte dans RBMoM.

Tout comme RBMoM, MMA ne propose rien quant à l'émission. Il n'a pas été conçu pour IPv6 et fait l'hypothèse que tous les réseaux visités contiennent un MA qui implémente le protocole MMA. Néanmoins, le protocole reste relativement simple et est très efficace du point de vue routage (presque optimal). Les modifications protocolaires sont minimales. En effet, il suffit de créer deux nouveaux messages (*Forwarding Request* et *Forwarding Stop*) et de modifier les messages d'enregistrement pour y inclure la liste de couples (groupe,MF précédent).

5.8 MobiCast

La différence majeure entre MobiCast [43] et les autres solutions décrites dans ce document est que MobiCast tient compte de la micro-mobilité. Les autres protocoles supposent que les changements de réseau arrivent peu fréquemment car ils ne s'occupent que de macro-mobilité. Au contraire, la micro-mobilité signifie que l'on tient compte des changements de cellules des réseaux sans-fils qui arrivent très fréquemment pour le mobile. Les antennes (*base stations*, BS) couvrent des surfaces géographiquement très réduites.

Dans MobiCast, il est important de noter que les BS sont des équipements de niveau trois (des routeurs), et non pas de niveau deux (switches ou hubs). Les BS sont donc capables de participer au routage unicast et multicast, d'analyser les en-têtes IP pour la qualité de service, etc. Les BS d'une zone ne forment donc pas un Wireless LAN mais bien une topologie complète. Le but de MobiCast est d'offrir un service multicast efficace, tout en maintenant une bonne qualité durant les handoffs, aux hôtes mobiles se déplaçant dans un environnement constitué de petites cellules sans fils, chacune sous la responsabilité d'une BS.

Comme les changements de cellules sont très fréquents, les méthodes *Local Membership* et *Remote Membership* ne sont pas conseillées. En effet, la première méthode oblige l'arbre de distribution multicast à se reformer continuellement, tandis que la deuxième demande de modifier trop souvent les tunnels. La solution proposée dans MobiCast permet d'éviter ces désagréments tout en maintenant un bon routage et de faibles pertes durant les handoffs.

5.8.1 Description du protocole

Pour fixer les idées, supposons que nous ayons comme structure un campus, composé de différents sous-réseaux, de routeurs, et d'une partie sans-fil permettant aux noeuds mobiles d'accéder au réseau du campus (et de là, à Internet) via les Base Stations.

Une nouvelle entité est créée : le *Domain Foreign Agent* (DFA), qui permet de rendre la mobilité transparente dans le Foreign Network pour l'arbre de distribution multicast. Pour le campus tout entier, supposons qu'il y ait un seul DFA, qui soit responsable de tous les mobiles étrangers en visite sur le campus. L'adresse du DFA est diffusée périodiquement en broadcast sur le réseau jusqu'aux mobiles, via les BS (voir Figure 5.9).

Quand un mobile décide de s'attacher au réseau, il s'enregistre avec le DFA et signale à son HA qu'il utilise l'adresse DFA comme Care-of Address. Il s'agit ici d'un mécanisme en tous points semblable à Mobile IP (macro-mobilité). Tout le trafic multicast ultérieur du mobile, en émission ou en réception, se fera via le DFA.

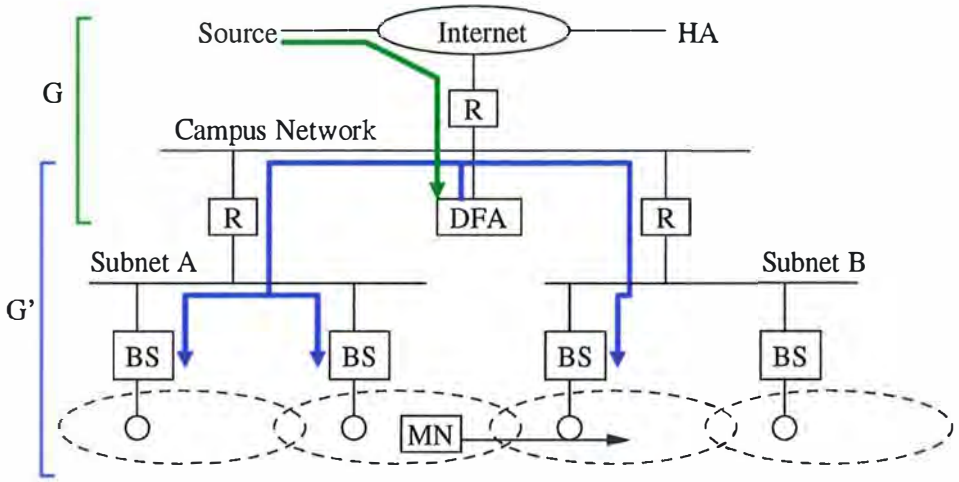

FIG. 5.9 – MobiCast - Forwarding multicast du DFA aux BS dans la même DVM

Quand le mobile change de cellule, c'est-à-dire qu'il change de BS mais ne sort pas du campus, son déplacement est invisible pour le reste du groupe multicast. Aucun recalcul de l'arbre n'est nécessaire tant que le mobile se déplace à l'intérieur du domaine.

5.8.2 Emission de trafic multicast

Si un MN est une source de trafic multicast, celui-ci ne peut utiliser l'adresse du DFA comme adresse source car les paquets pourraient être jetés par l'algorithme de RPF Check. Le mobile ne peut pas non plus utiliser d'adresse Co-located, car MobiCast requiert justement l'utilisation de la CoA. MobiCast propose de contourner ce problème en obligeant le MN à encapsuler ses datagrammes multicast dans des datagrammes unicast à destination du DFA. Le DFA décapsule les datagrammes et les envoie sur l'arbre multicast à la place du MN, en positionnant l'adresse source des datagrammes multicast à l'adresse du DFA.

Pour que cela fonctionne, il faut que les applications ne tiennent pas compte de l'adresse source pour identifier l'émetteur. Elles doivent donc inclure une autre identification dans le contenu même du datagramme, afin, par exemple, de distinguer un datagramme multicast émis par DFA pour le compte de MN1 d'un autre datagramme émis lui aussi par DFA mais pour le compte de MN2.

Les auteurs de MobiCast admettent que leur méthode est assez contraignante. Aussi, s'il n'est pas possible d'imposer cela aux applications, il est toujours possible d'en revenir à un tunnel inverse via le HA pour l'émission des datagrammes multicast.

5.8.3 Réception de trafic multicast

Pour recevoir du trafic multicast, le MN demande d'abord à sa Base Station (BS) de l'inscrire au groupe G grâce à un message IGMP. Cette demande est relayée par la BS jusqu'au DFA, tout en tenant compte de l'identité du MN. Ensuite, le DFA transmet une nouvelle adresse multicast G', l'adresse multicast translatée de G, à la BS. Le DFA s'enregistre au groupe G et retransmet les datagrammes de G aux mobiles de son domaine sur G'. Les Base Stations qui hébergent des MN membres de G reçoivent les datagrammes de G en s'inscrivant

à G'. Finalement, les BS retransmettent les datagrammes aux mobiles de leur cellule qui sont membres de G.

En d'autres mots, nous avons affaire à deux groupes G et G', situés à deux niveaux différents. G est le groupe multicast global, tandis que G' est dérivé de G, se limite à un domaine et permet au DFA du domaine de cacher la mobilité des mobiles de son domaine en réinjectant les datagrammes de G dans G'.

5.8.4 Fast Handoff

Pour éviter toute coupure de réception multicast lors des handoffs quand un mobile passe dans une autre cellule, les cellules adjacentes sont organisées en groupe appelé DVM (*Dynamic Virtual Macrocells*). Quand une BS transmet la demande d'inscription à un groupe d'un MN au DFA, une adresse G' est transmise à la BS. En plus de s'inscrire au groupe G', celle-ci demande à ses collègues BS de la même DVM de s'inscrire également à G'. Ainsi, les datagrammes multicast de G arrivent aussi aux BS des cellules adjacentes où ils sont bufferisés. Si le MN arrive dans la cellule adjacente, la BS peut directement fournir au MN des datagrammes de G, stockés dans sa cache, réduisant ainsi le temps de handoff, puisqu'elle avait déjà joint G'.

5.8.5 Signalement des pertes au MN

Quand un mobile émetteur de trafic multicast change de cellule, certains datagrammes sont perdus. MobiCast permet de signaler ces pertes au MN. Les applications pourront ainsi décider de ce qu'il faut faire en connaissance de cause. Pour ce faire, chaque BS note l'ID du dernier datagramme envoyé sur G par le MN. Quand le mobile arrive sur une nouvelle BS, celle-ci signale à l'ancienne BS le déplacement du mobile. Dans l'acquittement fourni par l'ancienne BS se trouve l'ID du dernier datagramme envoyé sur G par le MN. La nouvelle BS retransmet ensuite cette information au mobile.

5.8.6 Critique

MobiCast est une solution très intéressante de par sa vision hiérarchique du problème. Tant qu'un mobile reste sous le contrôle du même DFA, il n'y a pas de reconstruction d'arbre multicast, même si le mobile se déplace de BS en BS. Le DFA devient néanmoins crucial. S'il vient à faillir, toutes les communications multicast présentes et futures sont impossibles à assurer.

Le calcul de l'adresse translatée en IPv4 n'est pas explicité par les auteurs et doit être assez difficile. Premièrement, il y a très peu d'adresses multicast disponibles en IPv4. Deuxièmement, il faut garantir que l'adresse translatée ne soit pas déjà utilisée.

Du point de vue de l'émission, les contraintes pour être efficace sont assez gênantes. De plus, il n'est jamais conseillé de changer l'adresse source d'un datagramme (comme le DFA doit le faire) car cela interdit l'utilisation de protocoles de sécurité.

L'adaptation en IPv6 de MobiCast serait encore plus intéressante. En effet, le calcul de l'adresse translatée pourrait être fait en changeant simplement le scope de l'adresse multicast. Au lieu de joindre un groupe de scope global (préfixe `ff0e::/16`), les mobiles joindraient le même groupe de scope site-local (préfixe `ff05::/16`). Le DFA n'aurait qu'à changer le scope

des datagrammes reçus sur l'arbre global avant de les retransmettre sur l'arbre site-local (en considérant le campus comme un site). Concernant l'émission, il suffirait d'ajouter un *Routing Header* aux datagrammes comme cela est permis dans Mobile IPv6 pour éviter les problèmes décrits plus haut.

Finalement, il serait tout à fait possible d'éviter les adresses translatées. Si le DFA sépare bien l'extérieur et l'intérieur du campus, il peut tout à fait gérer la transparence de la mobilité en utilisant toujours la même adresse, simplement en prenant garde de ne pas répercuter aveuglément les messages du protocole de routage d'une zone dans l'autre zone.

5.9 Mobile SSM Sources for IPv6 - MSSMv6

Deux français sont à l'origine de MSSMv6 [28], draft sorti en Janvier 2002, qui examine le problème de mobiles émetteurs de trafic multicast, dans une topologie à base de PIM-SSM IPv6.

Tout d'abord, rappelons-nous que le service offert sur la couche réseau en mode *Source-Specific* repose sur des *canaux*, c'est-à-dire sur l'association de l'adresse IP source (S) et de l'adresse IP destination du groupe (G). Les datagrammes du groupe transitant sur le réseau ne diffèrent pas du multicast classique. Seule l'inscription à un canal diffère et nécessite MLDv2 (l'équivalent d'IGMPv3 en IPv4). Comme le canal est identifié en partie par l'adresse source, il est impossible d'utiliser telle quelle la méthode *Local Subscription.* L'adresse source des datagrammes multicast émis par le mobile serait la Care-Of Address courante du mobile, ce qui signifierait un changement de canal à chaque changement de réseau.

La seule méthode simple qui peut fonctionner en SSM sans modification est donc le *Remote Subscription (ou Bidirectionnal Tunneling).* Le canal sera identifié par (G,H), c'est-à-dire l'adresse de groupe et la Home Address du mobile. Les récepteurs s'abonneront aussi à (G,H), sans savoir à aucun moment que la source est mobile. Le mobile dans son réseau Home émettra ses datagrammes comme un hôte normal. Une fois en déplacement, il tunnellera ses datagrammes vers son Home Agent, qui les décapsulera, et les retransmettra sur l'arbre multicast.

MSSMv6 est une adaptation de la méthode *Local Membership* destinée à la faire fonctionner avec les canaux. Dans ce protocole, le mobile qui commence une communication multicast (pour un groupe G) émet ses datagrammes de façon native (sans tunnel) avec une adresse source IPv6 égale à sa Care-of Address, tout en ajoutant une option de destination Home Address contenant sa Home Address (H).

Cette façon de procéder a plusieurs implications directes. Le protocole multicast identifie le canal par (G,CoA), car il ne tient pas compte de l'option Home Address contenue dans les datagrammes. Ce qui implique que les hôtes qui veulent se joindre au groupe doivent émettre des messages MLD Join (G,CoA) (et non MLD Join(G,H)). Il faut donc que le canal soit annoncé par (G,CoA). Néanmoins, la façon de l'annoncer n'a pas encore été formalisée.

La racine de l'arbre se trouve dans le FN d'où la communication multipoint a débuté et non plus dans le HN. Le routage est donc plus efficace.

Au niveau applicatif, la communication est identifiée par (G,H) car la Home Address contenue dans une option de destination des datagrammes et la Source Address ont été substituées pour assurer la transparence.

Supposons maintenant que le mobile se déplace dans un autre sous-réseau. Il faut que le mobile prévienne les membres du groupe qu'il a une nouvelle CoA (nCoA). Les membres pourront ainsi joindre le nouvel arbre (G,nCoA).

S'il y a un mécanisme de fast-handover (ce qui permet au mobile d'encore émettre sur l'ancien réseau tout en connaissant déjà nCoA), le MN envoie un BU spécial sur (G,CoA) contenant une nouvelle sous-option *SSM-Source Handover Notification.*

Sinon, le BU est encapsulé vers l'ancien routeur d'accès (oAR) qui va le décapsuler et le forwarder sur (G,CoA).

Dans les deux cas, l'adresse IPv6 source du datagramme contenant le BU doit être égale à l'ancienne Care-of Address (CoA).

Une fois qu'un récepteur reçoit un BU avec *SSM-Source Handover Notification*, il devrait joindre le nouveau canal (G,nCoA). Le récepteur ne doit pas quitter (G,CoA) tant qu'il n'a pas encore reçu de datagrammes sur (G,nCoA). Dès qu'il en reçoit, il devrait initier l'élagage de (G,CoA). Si le récepteur n'arrive pas à joindre le nouveau canal, il devrait maintenir son inscription pour l'ancien.

Même si le handover est terminé, le mobile devrait continuer à émettre aussi sur (G,CoA) tant qu'oAR ne lui a pas notifié qu'il n'y a plus de récepteurs pour le groupe (G,CoA) (voir Figure 5.10). Cette notification pourrait se faire au moyen du protocole MSNIP [23] mais n'a pas été formalisée.

Comme il peut y avoir des récepteurs sur l'ancien lien d'oAR, oAR devrait aussi retransmettre les datagrammes décapsulés du MN sur ses interfaces locales (plus précisément, les interfaces sur lesquelles il envoie des MLD Query).

Le MN doit encore mettre à jour sa façon d'annoncer la session, avec nCoA. Comme il n'y a pas encore de façon standardisée pour annoncer une session, la mise à jour de l'annonce n'est pas détaillée.

5.9.1 Critique

Le prix à payer pour obtenir un routage optimal en mode SSM est élevé. En effet, il y a un routage redondant tant que tous les récepteurs ne sont pas passés sur le nouvel arbre. Ceci est potentiellement très dangereux. En effet, à supposer qu'un seul des récepteurs ne veuille pas passer sur le nouvel arbre, il faut conserver tous les arbres (réduits progressivement par le processus d'élagage) créés par le mobile à chaque changement de réseau.

Si les changements de réseau du mobile sont fréquents, ce protocole est forcément inadapté car beaucoup de mémoire est utilisée sur les routeurs et beaucoup de bande de passante est utilisée en trafic de contrôle (les BU périodiques et le forwarding sur l'ancien arbre si nécessaire).

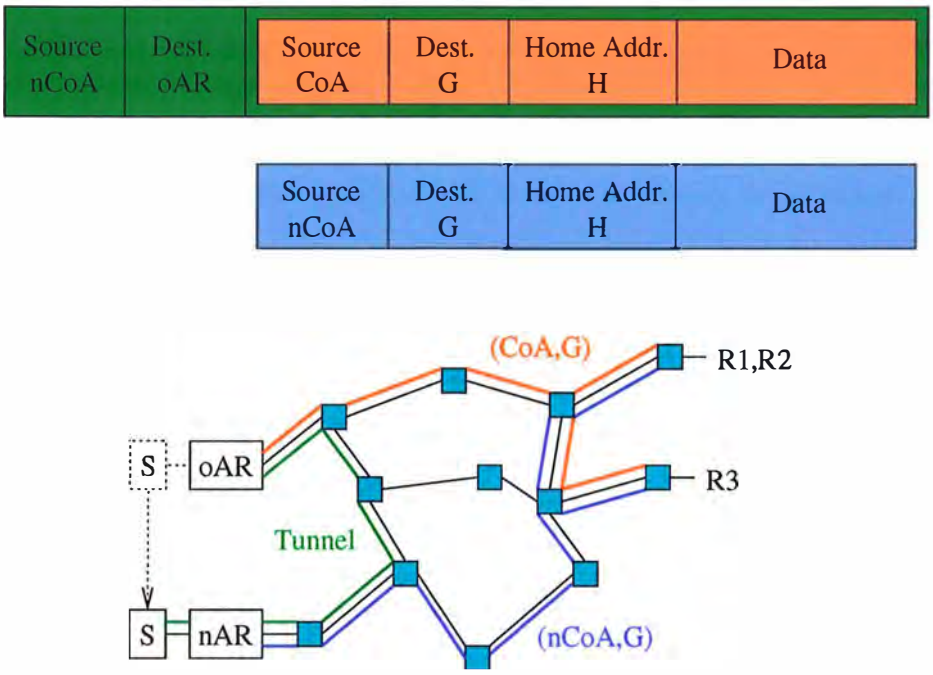

FIG. 5.10 – MSSMv6

Le protocole impose aussi de modifier l'implémentation des CN pour qu'ils sachent traiter la nouvelle option *SSM-Source Handover Notification*. Ceci n'est pas encore trop grave finalement, vu le faible déploiement d'IPv6, de PIM-SSM et de Mobile IPv6.

Le protocole demande aussi à l'ancien routeur d'accès de pouvoir gérer des tunnels. Ce rôle qui n'est pas prévu à la base pour un routeur multicast nécessiterait plutôt la définition d'une nouvelle entité. Rien n'est prévu si l'ancien routeur d'accès ne possède pas ces fonctions évoluées.

MSSMv6 manque aussi de formalisme. Il s'agit plus d'une idée que d'une solution en l'état actuel de la proposition. C'est assez compréhensible car le document en est à sa première version.

5.10 Mobile Multicast Protocol using Anycast - MMPuA

5.10.1 Description

MMPuA [29] définit le concept de Multicast Agent (MA). On suppose qu'un MA est disponible dans chaque sous-réseau. Chaque MA maintient quatre listes pour l'appartenance dynamique d'un noeud mobile (MN) au groupe G :

- La *Membership List* (ML(G)) contient la liste des membres statiques du groupe G.
- La *Visitor List* (VL(G)) contient la liste des noeuds mobiles étrangers qui sont en visite pour le moment chez ce MA.
- La *Away List* (AL(G)) contient la liste des noeuds mobiles qui ont quitté ce MA.
- La *Tunneling List* (TL(G)) contient la liste des FA qui sont intéressés par la réception des datagrammes multicast du groupe G.

Le protocole MMP est séparé en trois grandes phases :

- La phase d'initialisation qui configure les groupes multicast et anycast sur les agents de mobilité, les routeurs et les noeuds mobiles.
- La phase d'enregistrement qui permet au noeud mobile de s'enregistrer dynamiquement.
- La phase de transmission multicast qui permet à tous les membres du groupe de recevoir les datagrammes multicast.

Phase 1 : Initialisation

Chaque MA vide ses quatre listes : ML(G) = VL(G) = AL(G) = TL(G) = {}.

Formation de l'arbre multicast L'arbre multicast se forme sur les routeurs à l'aide du protocole CBT (c'est le protocole de routage multicast choisi par les auteurs dans leur description). Un des routeurs est sélectionné comme étant le coeur (ou la racine) de l'arbre.

Configuration du groupe anycast des MA Les MA qui permettent aux mobiles de joindre un groupe donné G forment un groupe anycast d'adresse G_A. Le protocole spécifie que les agents du même groupe G_A doivent partager la même identification pour les enregistrements de noeuds mobiles. Cela signifie que si deux MA appartiennent à G_A, ils peuvent se déléguer l'un l'autre l'identification et la retransmission multicast des noeuds mobiles pour le groupe G.

Configuration du groupe anycast des routeurs de l'arbre Une deuxième adresse anycast T_A est attribuée et configurée par tous les routeurs multicast situés sur l'arbre du groupe G. Quand l'arbre CBT du groupe G est construit, tous les routeurs de l'arbre (y compris le coeur) joignent le groupe anycast T_A qui est diffusé à tous les routeurs multicast par le coeur (broadcast limité aux routeurs multicast). Les routeurs multicast qui ne font pas partie de l'arbre reçoivent aussi l'adresse T_A. Parmi ces routeurs se trouvent des Foreign Agents qui pourraient être intéressés plus tard par G. Ils vont donc noter dans une table la correspondance entre G et T_A. La table de routage anycast permettra au routeur de choisir dynamiquement une "meilleure" route pour joindre l'arbre CBT quand ce sera utile.

Phase 2 : Enregistrement dynamique

Enregistrement sur le Home Network Le mobile MN utilise l'adresse anycast G_A pour joindre le MA le plus "proche". Le MA identifie le MN et lui attribue une CoA. De plus, il ajoute la CoA à sa *Membership List* ML(G). Deux cas peuvent se présenter. Si le MA est déjà membre de l'arbre de G, tout est déjà bien configuré. Sinon, deux sous-cas peuvent se présenter. Si le MA est un routeur multicast qui n'est pas membre de l'arbre de G, il lance une requête "Join G_A" à destination de l'adresse anycast T_A afin de joindre l'arbre de G via le plus "proche routeur" membre de l'arbre. Par contre, si le MA n'est pas un routeur multicast, il construit un tunnel anycast vers le plus proche routeur de l'arbre.

Enregistrement sur un Foreign Network Supposons que le mobile, membre de G_A et attaché à MA_1, se déplace dans un autre sous-réseau où il s'attache à MA_2. Si MA_2 est aussi membre de G_A, MN peut aussi utiliser l'adresse G_A pour s'enregistrer auprès de MA_2. MA_2 insère la nouvelle CoA (nCoA) dans sa liste de visiteurs du groupe G (VL(G)). MA_1 déplace le MN de sa liste de membres ML(G) à sa liste de mobiles en voyage AL(G).

Si MA_2 n'est pas membre de G_A, il crée un tunnel bidirectionnel avec MA_1. Il ajoute aussi la nouvelle CoA du MN dans sa liste VL(G). Après réception du message de demande de tunnel, MA_2 ajoute l'adresse de MA_1 dans sa liste TL(G).

Le noeud mobile s'en va Quand le noeud mobile quitte un réseau, il doit le signaler à son MA avec un message de désenregistrement. Le MA place alors le MN dans sa liste AL(G). Si ML(G), VL(G) et TL(G) sont vides, le MA peut se désinscrire du groupe G en envoyant le message approprié du protocole CBT. Le MA peut configurer un certain timeout après lequel les entrées de la liste VL(G) et TL(G) sont effacées.

Phase 3 : Transmission

Emission Un noeud mobile peut générer un message à destination du groupe G. Le message est d'abord transmis encapsulé au MA, qui le décapsule et le réencapsule à destination de T_A. Le paquet est ensuite routé à destination du plus proche routeur membre de l'arbre G. Quand un routeur de T_A reçoit le paquet, il le décapsule et le retransmet sur G. Evidemment, si le MA est déjà membre de G_A, la première étape de tunnelage est inutile.

Réception Quand un MA reçoit un paquet encaspulé d'un routeur de l'arbre CBT, il le retransmet aux noeuds présents dans les listes ML(G) et VL(G). De plus, il tunnelle le paquet vers les noeuds de la liste TL(G), s'il y en a.

5.10.2 Critique

La description originale de MMPuA manque singulièrement de clarté. Il n'est par exemple jamais précisé si le protocole est prévu pour IPv4 ou IPv6. Il est dit au début que l'anycast est une nouveauté d'IPv6, ce qui laisse à penser que le protocole est conçu pour IPv6 mais plus tard, la description fait référence plusieurs fois aux Foreign Agents qui n'existent pas en IPv6 !

Il n'est pas non plus expliqué comment l'adresse G_A est calculée à partir de l'adresse de G. Le broadcast limité aux routeurs multicast semble être un gros gaspillage de bande passante. De plus, tous les routeurs multicast doivent noter la correspondance entre G et T_A qui ne sera potentiellement jamais utilisée.

L'anycast est encore très peu connu en IPv6 et il y a d'ailleurs beaucoup de discussions autour de ce sujet à l'IETF. Il est donc sans doute un peu prématuré d'utiliser cette technique à outrance pour le support du multicast sur les hôtes mobiles.

L'argument d'utiliser une adresse anycast pour les agents de mobilité afin d'éviter aux mobiles de devoir apprendre l'adresse unicast de l'agent ne correspond pas vraiment à la réalité. Un mobile apprend qu'il a changé de réseau avec les Router Advertisements et ceux-ci sont émis par les agents de mobilité. Le mobile connaît donc directement l'adresse du MA.

Ce protocole a quand même le mérite de proposer une solution concernant l'émission. Il s'agit de tunneler les datagrammes vers le plus proche routeur de l'arbre en utilisant une adresse anycast. Cette idée fonctionne bien pour CBT car c'est un protocole de routage multicast qui utilise des arbres partagés bidirectionnels, mais ce n'est pas applicable à d'autres protocoles comme PIM où les arbres ne sont pas bidirectionnels. Néanmoins, une variante de PIM, appelée

BIDIR-PIM, est développée actuellement pour permettre l'utilisation d'arbres bidirectionnels dans PIM [25].

5.11 Tableau comparatif

Le tableau comparatif qui suit résume les différentes caractéristiques des huit solutions. Pour des raisons de mise en page, il est divisé sur quatre pages. La première colonne contient les différents paramètres évalués. Une barre oblique (/) signifie que le paramètre n'est pas approprié à la solution. Par exemple, les critères concernant le Foreign Agent n'ont aucune signification pour la solution MSSMv6 puisqu'il n'y a pas de Foreign Agent en IPv6.

Les différents paramètres évalués répondent aux questions suivantes :
- La solution propose-t-elle quelque chose concernant l'émission, la réception ou les deux ?
- Est-elle conçue pour IPv4, IPv6 ou les deux ?
- Quels sont les protocoles de routage multicast et les protocoles de gestion de groupes qui peuvent fonctionner avec la solution ?
- Quelles entités faut-il modifier pour implémenter le protocole ?
- Quelle est la complexité de la solution par rapport aux autres ?
- La solution a-t-elle été conçue pour être compatible avec Mobile IP ou est-elle plutôt indépendante ?
- Y a-t-il une nouvelle entité définie dans le protocole et si oui, comment s'appelle-t-elle ?
- Quels sont les messages de contrôle sur le réseau visité ou dans le protocole de routage multicast en distinguant l'émission de la réception.
- Est-ce que la solution utilise un peu, beaucoup ou tout le temps des tunnels ?
- Est-ce que la solution produit un routage efficace des datagrammes ?
- Le délai pour recevoir les datagrammes en situation de mobilité est-il plus élevé qu'en situation fixe (at Home) ?
- Quelles sont, grosso-modo, les informations à maintenir en plus sur les différentes entités ?
- La solution est-elle transparente ? (c'est-à-dire, est-ce que les correspondants ou les routeurs multicast se rendent compte de la mobilité du noeud mobile ?)
- La solution supporte-t-elle les scopes IPv6 ?
- Est-ce qu'un routeur multicast est nécessaire dans chaque Foreign Network ? (même question pour le Foreign Agent et la nouvelle entité)
- Y a-t-il un point central d'échec ? (c'est-à-dire, une entité qui est fortement sollicitée ou qui concentre beaucoup de trafic)
- Finalement, est-ce que la solution est adaptable en IPv6 ?

Au vu du tableau, on se rend compte que pour fournir un service multicast efficace aux hôtes mobiles, il faut créer une nouvelle entité. Celle-ci peut évidemment être incorporée dans le Home Agent, le routeur multicast ou le Foreign Agent. Les tunnels sont souvent utilisés pour assurer la continuité du service. Les solutions évoluées permettent d'atteindre un routage presque optimal en limitant les tunnels. Pour agir de façon plus intelligente selon le contexte, il faut maintenir plus d'informations dans les différentes entités. Les solutions proposées supposent généralement par simplicité que le protocole proposé est déployé sur tout le réseau, ce qui est discutable. Elles sont toutes conçues à la base pour IPv4 (à part MSSMv6).

TAB. 5.1 – Comparatif : Bidirectionnal Tunneling et Local Subscription

	Bidirectionnal Tunneling	Local Subscription
Support de la réception	oui	oui
Support de l'émission	oui	oui en v6, problématique en v4
Conçu pour IPv4	oui	oui
Conçu pour IPv6	oui	oui
Protocoles de routage multicast compatibles	tous	tous sauf SSM
Protocoles de gestion de groupes compatibles	IGMP/MLD	IGMP/MLD
Entités à modifier :		
Routeur multicast	non	non
Noeud mobile	non	oui, légèrement
Home Agent	non	oui, légèrement
Foreign Agent (v4 seulement)	non	non
Correspondant	non	non
Complexité relative	aucune	aucune
Compatible avec Mobile IP	oui	oui
Nouvelle entité	non	non
Messages de contrôle (réception) :		
Sur le réseau visité	IGMP/MLD tunnelé vers HA	IGMP/MLD local
Dans le protocole de routage multicast	aucun	potentiellement beaucoup
Messages de contrôle (émission) :		
Dans le protocole de routage multicast	aucun	beaucoup
Tunnels	permanents	aucun
Routage	catastrophique	optimal
Problème des tunnels convergents	oui	non
Problème des tunnels parallèles	oui	non
Délai relatif	grand si les tunnels sont longs	faible
Informations à maintenir (en plus) sur :		
Routeur multicast du lien	rien	MN membre de G
Noeud mobile	rien	rien
Home Agent	Liste (MN x Liste de groupes à tunneler)	rien
Foreign Agent	rien	rien
Correspondant	rien	rien
Nouvelle entité	/	/
Transparence	totale	aucune
Support des scopes IPv6	ceux disponibles dans le HN	échec à la sortie du scope + ceux du FN
Nouvelle entité nécessaire dans chaque FN	/	/
MR nécessaire dans chaque FN	non	oui
FA nécessaire dans chaque FN	non	non
Point central d'échec	le HA	aucun
Adaptable en IPv6	déjà fait	déjà fait

TAB. 5.2 – Comparatif : MoM et RBMoM

	MoM	RBMoM
Support de la réception	oui	oui
Support de l'émission	non	non
Conçu pour IPv4	oui	oui
Conçu pour IPv6	non	non
Protocoles de routage multicast compatibles	tous sauf SSM	tous sauf SSM
Protocoles de gestion de groupes compatibles	IGMP	IGMP
Entités à modifier :		
Routeur multicast	non	non
Noeud mobile	non	oui
Home Agent	oui	oui
Foreign Agent (v4 seulement)	oui	oui
Correspondant	non	non
Complexité relative	presque aucune	moyenne
Compatible avec Mobile IP	oui	oui mais ajout d'une entité (MHA)
Nouvelle entité	DMSP	DMSP et MHA
Messages de contrôle (réception) :		
Sur le réseau visité	IGMP local vers FA tunnelé vers HA	IGMP local vers FA / MHA
Dans le protocole de routage multicast	aucun	si changement de MHA
Messages de contrôle (émission) :		
Dans le protocole de routage multicast	/	/
Tunnels	permanents	épisodiques
Routage	non optimal	presque optimal
Problème des tunnels convergents	non (DMSP)	non (DMSP)
Problème des tunnels parallèles	non (FA)	non (FA)
Délai relatif	grand si les tunnels sont longs	raisonnable
Informations à maintenir (en plus) sur :		
Routeur multicast du lien	rien	rien
Noeud mobile	rien	son MHA
Home Agent	Liste (groupe à tunneler x actif x FA)	idem MoM + Liste (MN x MHA courant)
Foreign Agent	Liste (G x DMSP)	Liste (G x DMSP)
Correspondant	rien	rien
Nouvelle entité	voir HA	voir HA
Transparence	oui	tant qu'on reste dans la zone de service
Support des scopes IPv6	/	/
Nouvelle entité nécessaire dans chaque FN	non	voir FA
MR nécessaire dans chaque FN	non	voir FA
FA nécessaire dans chaque FN	oui	oui
Point central d'échec	Le HA mais moins qu'en Bidir. Tun.	aucun
Adaptable en IPv6	oui	oui

TAB. 5.3 – Comparatif : MMA et MMPuA

	MMA	MMPuA
Support de la réception	oui	oui
Support de l'émission	non	oui
Conçu pour IPv4	oui	oui
Conçu pour IPv6	non	non
Protocoles de routage multicast compatibles	tous sauf SSM	CBT
Protocoles de gestion de groupes compatibles	IGMP	IGMP
Entités à modifier :		
Routeur multicast	non	oui
Noeud mobile	oui	oui
Home Agent	non	non
Foreign Agent (v4 seulement)	non	oui
Correspondant	non	non
Complexité relative	moyenne	grande
Compatible avec Mobile IP	plutôt indépendant de MIP	plutôt indépendant de MIP
Nouvelle entité	MA et MF	MA
Messages de contrôle (réception) :		
Sur le réseau visité	IGMP + Forwarding request/stop	IGMP + Demande de tunnels
Dans le protocole de routage multicast	si demande explicite du mobile	Broadcast de T_A
Messages de contrôle (émission) :		
Dans le protocole de routage multicast	/	
Tunnels	entre MF et MA	entre les MA et entre MN et MA pour l'émission
Routage	presque optimal	presque optimal
Problème des tunnels convergents	non	non
Problème des tunnels parallèles	non	non
Délai relatif	faible	?
Informations à maintenir (en plus) sur :		
Routeur multicast du lien	voir 'Nouvelle entité'	voir 'Nouvelle entité'
Noeud mobile	Liste (groupe,MF)	Liste (groupe,MA) + G_A
Home Agent	/	/
Foreign Agent	/	/
Correspondant	/	/
Nouvelle entité	Liste (groupes,MF) à tunneler	ML,VL,AL,TL
Transparence	aucune	aucune
Support des scopes IPv6	/	/
Nouvelle entité nécessaire dans chaque FN	oui	oui
MR nécessaire dans chaque FN	oui	oui
FA nécessaire dans chaque FN	/	/
Point central d'échec	/	/
Adaptable en v6	oui	oui

TAB. 5.4 – Comparatif : Mobicast et MSSMv6

	MobiCast	MSSMv6
Support de la réception	oui	non
Support de l'émission	oui	oui
Conçu pour IPv4	oui	non
Conçu pour IPv6	non	oui
Protocoles de routage multicast compatibles	tous sauf SSM	PIM-SSM seulement
Protocoles de gestion de groupes compatibles	IGMP	MLDv2
Entités à modifier :		
Routeur multicast	oui s'il est DFA	oui
Noeud mobile	oui	oui
Home Agent	si HA est DFA	non
Foreign Agent (v4 seulement)	remplacé par le DFA	/
Correspondant	non	oui
Complexité relative	grande	moyenne
Compatible avec Mobile IP	oui	oui
Nouvelle entité	DFA	MR amélioré
Messages de contrôle (réception) :		
Sur le réseau visité	Entre BS et DFA, entre BS adjacentes	/
Dans le protocole de routage multicast	suivant le type de handoff	/
Messages de contrôle (émission) :		/
Dans le protocole de routage multicast		Notif.+ création d'un nouvel arbre
Tunnels	en émission, jusqu'au DFA	temporaire vers l'ancien oAR
Routage	presque optimal	optimal
Problème des tunnels convergents	non	/
Problème des tunnels parallèles	non	/
Délai relatif	faible, grâce aux DVM	?
Informations à maintenir (en plus) sur :		
Routeur multicast du lien	Ici BS : la correspondance entre G et G'	MN qui ont demandé un tunnel
Noeud mobile	aucune	Anciens canaux actifs et MR associés
Home Agent	aucune	aucune
Foreign Agent	Ici DFA : Liste de (G,G')	/
Correspondant	rien	Liste de (canal, canal courant)
Nouvelle entité	/	/
Transparence	oui, à l'intérieur d'un campus	aucune
Support des scopes IPv6	non	non
Nouvelle entité nécessaire dans chaque FN	/	oui
MR nécessaire dans chaque FN	/	oui
FA nécessaire dans chaque FN	/	/
Point central d'échec	le DFA	aucun
Adaptable en v6	oui	/

Chapitre 6

Nouvelle proposition

6.1 Introduction

Pour se fixer un cadre de travail, nous devons d'abord situer le contexte dans lequel nous voulons développer notre solution. Nous faisons l'hypothèse que le réseau supporte nativement IPv6 et Mobile IPv6. Pour ce qui est du routage multicast, nous supposons que le protocole actif est PIM Sparse Mode version 2 pour IPv6, car c'est celui qui a le plus de chance de s'imposer prochainement. Les noeuds mobiles implémentent tous MLDv2. Ils ne changent pas de sous-réseau fréquemment car les zones géographiques couvertes par un sous-réseau IP sont suffisament grandes. Cela correspond par exemple à des réseaux d'entreprises dans lesquels on a déployé des stations de base 802.11.

Les objectifs de notre solution sont sensiblement les mêmes que ceux des propositions du chapitre précédent. Nous nous attacherons d'abord à réduire le temps de handoff. En effet, la majorité des applications multicast seront en temps réel, il faut donc que le handoff soit le plus rapide possible pour éviter des coupures désagréables dans l'application. Nous voulons aussi obtenir un routage quasi-optimal, en évitant les tunnels si possible. Il faut que le trafic de contrôle dans les différents protocoles utilisés ne soit pas trop élevé car les liens sans-fils ne disposent généralement pas d'une grande bande passante. Il faut aussi garder une solution suffisament simple pour qu'elle soit déployable, tout en modifiant le moins possible Mobile IPv6 qui est en bonne voie de standardisation. Comme le multicast n'est pas encore totalement déployé, nous devons tenir compte des Foreign Networks qui ne supportent pas forcément le multicast. Finalement, il ne faut pas non plus oublier les scopes, une des nouveautés d'IPv6. Ceux-ci ne sont pas encore bien définis et ne simplifient pas la définition du protocole. Nous supposons qu'un mobile doit pouvoir s'abonner à tous les groupes disponibles dans les zones qu'il traverse. S'il passe d'un site dans un autre site, il doit pouvoir s'abonner au groupe d'adresse G quand il se trouve dans le premier site et s'abonner à un autre groupe ayant la même adresse G mais se situant dans le second site. Il faut aussi que la communication continue même quand le mobile quitte le scope du groupe initial.

6.2 Présentation de la solution

6.2.1 Définition de Mobile Multicast Router

Nous définissons une nouvelle entité par rapport à Mobile IPv6 : le Mobile Multicast Router (MMR). Son rôle est de transmettre les datagrammes de certains groupes multicast aux noeuds

mobiles ou à d'autres MMR. De plus, il peut aussi servir de tunnel inverse pour des mobiles qui souhaiteraient continuer à émettre en multicast depuis le réseau du MMR. Un MMR est donc un routeur multicast (PIM-SM dans notre cas) avec des fonctions supplémentaires permettant un support efficace de la mobilité. Nous supposerons que tout mobile possède un MMR privilégié sur son réseau Home, qui peut être aussi son Home Agent. Si le mobile ne connaît pas son Home-MMR par configuration statique, il peut le découvrir dynamiquement grâce à un processus expliqué plus loin.

6.2.2 Note préliminaire sur la sécurité

Un MMR peut utiliser une liste d'accès pour ne pas offrir ses services à tous les mobiles, de la même façon qu'un Home Agent en Mobile IPv6. Nous n'allons pas spécialement approfondir le côté sécurité du protocole. Néanmoins, tout comme en Mobile IPv6, il serait bon que les nouveaux messages soient sécurisés via IPSec et que les différentes entités aient entre elles des associations de sécurité. Pour bien faire, il devrait même être obligatoire de chiffrer les données quand un MMR émet des datagrammes multicast hors de leur scope initial via un tunnel, car les datagrammes sont sensés rester dans la zone délimitée par le scope de l'adresse multicast.

6.2.3 Nouveaux messages

Notre protocole nécessite la définition de deux nouveaux messages ICMPv6 : Tunnel Request (TR) et Tunnel Acknowledgement (TA). Ces messages ont une sémantique similaire aux *Binding Update* et *Binding Acknowledgement* de Mobile IPv6. Les uns servent à assurer le suivi des communications multicast tandis que les autres assurent le suivi des communications unicast.

Tunnel Request

Un message Tunnel Request (TR) est émis par un MMR ou un mobile pour demander au récepteur du message d'établir un tunnel. Deux flags (Send et Receive) contenus dans le message précisent le sens de propagation des messages dans le tunnel. Seul un mobile a le droit de positionner le flag Send à 1 : cela lui permet d'envoyer des datagrammes multicast depuis le réseau du MMR visé, en encapsulant ceux-ci préalablement à destination du MMR. Si le flag Receive est à 1, l'émetteur demande au MMR de recevoir les datagrammes de certains groupes multicast. Pour ce faire, il ajoute dans le message la liste des groupes désirés. Les messages Tunnel Request contiennent un Lifetime qui indique la durée de vie préférée du tunnel. Pour maintenir un tunnel en vie, il faut que l'entité qui demande le tunnel émette périodiquement de nouveaux TR afin de rafraîchir le Lifetime de l'état du MMR visé.

Tunnel Acknowledgement

Suite à un message Tunnel Request est émis un message d'acquittement Tunnel Acknowledgement signalant si le tunnel a bien été établi et le motif de l'échec le cas échéant. Par exemple, un MMR peut refuser de tunneler un groupe s'il ne reçoit pas pour le moment ce groupe via son protocole de routage multicast. Il peut aussi refuser pour des raisons de sécurité, des raisons de surcharge, ou autres. Les codes d'erreurs restent à définir.

Dynamic Mobile Multicast Router Address Detection

Tout comme il existe un mécanisme de découverte dynamique du Home Agent dans Mobile IPv6, nous introduisons le même mécanisme pour la découverte dynamique des Mobile Multicast Routers. Ce mécanisme est utile quand un mobile ne connaît pas son Home-MMR par configuration ou s'il souhaite découvrir un MMR sur le lien qui ne s'est pas encore annoncé par les messages RA. Tous les MMR qui participent à ce processus utilisent la même adresse anycast bien connue. Le mobile émet une requête vers cette adresse anycast et un seul des MMR répond en lui communiquant son adresse. Seuls les MMR disponibles sur le lien sont intéressants pour le mobile, c'est pourquoi il faut que le message de découverte dynamique ne sorte pas du lien en mettant un TTL de 1.

6.2.4 Modifications des protocoles existants

Router Advertisements

Un noeud mobile détecte qu'il a changé de réseau en recevant un RA annonçant un préfixe différent de celui qu'il utilisait. Tous les réseaux visités ne sont pourtant pas identiques. Il se peut qu'il n'y ait pas de MMR, ou carrément pas de routeur Multicast du tout. Afin que le mobile puisse agir directement de façon optimale, il faut qu'il puisse détecter grâce aux RA dans quelle situation il se trouve. Pour ce faire, nous ajoutons deux flags dans les RA : le premier indique que le routeur qui les émet participe au routage multicast; le deuxième indique que le routeur qui les émet est un MMR.

Différents routeurs multicast (MR) peuvent se trouver sur un même lien. Un processus d'élection via des messages PIM Hello périodiques permet de choisir quel sera l'unique MR actif parmi les MR du lien. Le MR qui est élu est appelé Designated Router (DR). Les autres MR continuent d'écouter les messages d'abonnement IGMP/MLD mais n'agissent pas au niveau de leur protocole de routage multicast. Si le DR est mis hors-service pour une quelconque raison, il y a une nouvelle élection et un autre MR devient DR.

Cela implique qu'un MMR ne doit positionner le flag Multicast à 1 dans les RA que s'il est le DR. Ceci est utile pour le mobile car s'il voit que le flag Multicast du RA est à 0 et que le flag MMR du RA est à 1, il comprendra qu'il a découvert un MMR qui n'est pas un DR. Cela ne servirait donc à rien d'établir des tunnels avec ce MMR.

MLD Join

Nous avons vu précédemment que quand les tunnels étaient construits entre les mobiles et les MMR, cela entraînait le problème des tunnels parallèles. Comme les datagrammes étaient décapsulés au niveau du mobile, les autres mobiles du même réseau visité ne pouvaient pas en profiter. Pour éviter cela, il faut que le tunnel soit établi, quand c'est possible, entre deux MMR. Ainsi, le MMR qui décapsule les datagrammes peut les retransmettre en multicast natif sur le lien et en faire profiter tous les membres du groupe. Pour chaque groupe inscrit dans un message MLD Join, nous ajoutons une option spécifiant l'adresse du précédent MMR qui servait ce groupe au mobile. Quand le MMR du réseau visité reçoit ce MLD Join, il peut ainsi établir un tunnel pour ce groupe avec le MMR dont l'adresse est contenue dans le MLD Join. Evidemment, si le routeur fait déjà partie de l'arbre de distribution multicast de ce groupe ou s'il a déjà établi un tunnel pour obtenir les datagrammes du groupe, cette information n'est d'aucune utilité.

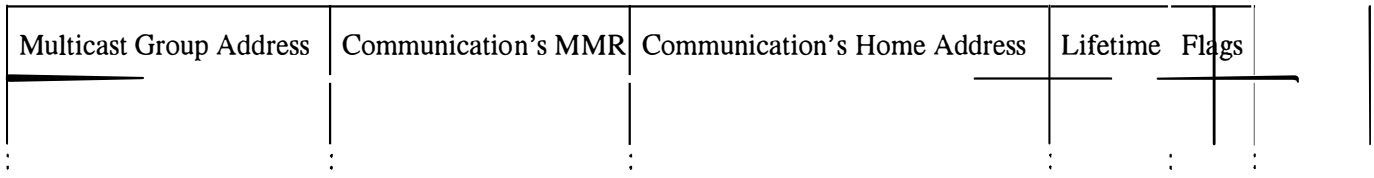

FIG. 6.1 – Structure de la Multicast Sending List

6.2.5 Informations à maintenir

Sur le noeud mobile

Chaque noeud mobile maintient de façon conceptuelle une Multicast Sending List (MSL) et une Multicast Receiver List. La première permet de savoir à destination de quels groupes le noeud émet, tandis que la deuxième donne la liste des groupes dont le noeud est membre.

Multicast Sending List Chaque élément de la MSL (voir Figure 6.1) est identifié par le couple composé de l'adresse du groupe et du MMR propre à la communication. Le MMR de la communication est celui qui servira de bout de tunnel quand le mobile sortira du scope du groupe. Si le mobile reste dans le scope, le mobile émet nativement (si un routeur multicast est disponible) pour obtenir un routage plus efficace.

On y trouve la "Communication's Home Address" qui représente la Home Address symbolique avec laquelle la communication a commencé. Pour rappel, toute communication doit avoir un identifiant. Dans le cas d'IP, il s'agit du quadruplet (adresse source, port source, adresse destination, port destination). Le concept de Home Address permet de rendre transparente la mobilité d'une source en ajoutant une option de destination Home Address. L'adresse source du quadruplet est ainsi comparée avec la Home Address du datagramme reçu qui est invariable tout au long de la communication. Généralement, on utilise comme Home Address la Home Address primaire du mobile, c'est-à-dire celle qu'il a dans son réseau Home. Ici, ce ne sera pas toujours le cas : supposons qu'un mobile veuille émettre à destination d'un groupe site-local sur un site visité différent de son site Home : il vaut mieux qu'il utilise sa CoA courante comme Home Address vu qu'il n'est théoriquement pas possible que le mobile puisse émettre de son réseau Home à destination d'un groupe site-local d'un site non Home. De plus, il se pourrait que la Home Address du mobile soit site-local et puisse alors interférer avec un mobile du site visité émettant avec la même Home Address site-local elle-aussi.

On trouve aussi dans une ligne de la MSL le Lifetime de celle-ci. Celui-ci diminue continuellement. Une fois qu'il atteint 0, l'entrée est effacée de la liste. Quand un datagramme est émis à destination du groupe, le Lifetime est remis à sa valeur maximum.

Finalement, on trouve différents flags bien utiles :
- "Available in Home Network" (H) permet de savoir si le Home-MMR peut servir de bout de tunnel pour la communication. Si oui, le champ Communication's MMR est initialisé avec l'adresse du Home-MMR.
- "Currently Tunnelling via MMR" (T) permet de savoir si on utilise actuellement le tunnel via le Communication's MMR ou si on émet nativement
- "Moved" (M) permet de savoir si l'on s'est déplacé du réseau dans lequel la communication a démarré, et donc s'il faut rajouter une option de destination Home Address contenant la Communication's Home Address.

Sur le MR

Pour optimiser la gestion des sources mobiles avec PIM-SM, il faut rajouter quelques informations dans les états (S,G) des routeurs multicast :

- Home Address : si ce champ existe, c'est que la source ayant comme Home Address la valeur contenue dans ce champ est en train d'émettre à destination de G avec la Care-of Address S. Ce champ permet de savoir que la source est mobile et aussi de déduire, quand une nouvelle source apparaît, qu'il ne s'agit pas en fait d'une ancienne source qui s'est déplacée.
- Handoffs Counter : ce champ calcule le nombre de handoffs déjà opérés par la source.
- Uptime cumulé : c'est la somme des uptimes des différents états (S,G) ayant le même champ Home Address.

Sur le MMR

Chaque MMR maintient une liste de tunnels actifs pour les entités qui lui ont demandé ce service. Cette liste est mise à jour suite à la réception d'un message Tunnel Update. Elle reflète simplement les informations qui sont comprises dans les messages Tunnel Update.

6.2.6 Fonctionnement du protocole pour la réception

Abonnement à un groupe

Un MN qui souhaite s'abonner à un nouveau groupe s'abonne toujours de façon native pour optimiser le routage, à condition qu'il y ait un routeur multicast disponible sur le réseau courant. Pour ce faire, il utilise le message MLD Join. Une fois abonné, il note dans sa Multicast Receiver List l'adresse du groupe et le fait que les datagrammes arrivent de façon native.

Si le MN souhaite s'abonner à un nouveau groupe multicast sans pour autant être en contact avec un routeur multicast, il faut d'abord qu'il essaye d'en trouver un. Pour ce faire, il émet un message ICMPv6 Dynamic Mobile Multicast Router Discovery, similaire au message de découverte dynamique du HA en MIPv6. Si un MMR répond, le mobile émet son MLD Join normalement. Par contre, si aucune réponse n'est reçue, la dernière chance du mobile consiste à s'adresser à un MMR que le mobile connaît déjà (soit par configuration, soit grâce à l'information disponible dans une cache de MMR).

Handoff

Un mobile se comporte par défaut comme spécifié dans Mobile IPv6. Il y a cependant toute une série d'actions complémentaires à effectuer pour assurer le suivi des communications multicast quand le mobile change de point d'attachement.

S'il existe un MMR sur le réseau visité (le MN le sait grâce aux flags contenus dans les RA), le mobile demande au nouveau MMR de lui faire parvenir les datagrammes multicast des groupes auxquels il était précédemment abonné et qui sont dans la même zone. Pour ce faire, il émet un MLD Join contenant la liste des groupes et les MMR correspondants s'il y en a. Le MMR doit maintenant se débrouiller pour satisfaire la demande du MN. Il se peut que certains groupes demandés soient déjà disponibles, car d'autres noeuds du réseau pourraient être membres des mêmes groupes. Dans ce cas, le MMR n'a rien de particulier à faire. Pour

les groupes qu'il ne reçoit pas encore, le MMR utilise son protocole de routage multicast (PIM-SM) et émet des messages Join(*,L) en amont.

Afin de diminuer le temps de handoff, le MMR demande aussi des tunnels temporaires pour ces mêmes groupes aux MMR qui sont notés dans le message MLD Join du MN. Pour ce faire, le MMR envoie des Tunnel Updates vers les MMR contenant une liste de groupes à tunneler. Comme le MN provient d'un réseau adjacent, les tunnels devraient théoriquement être courts. Etant donné que les tunnels ne servent que comme roues de secours en attendant que les datagrammes voulus arrivent nativement, il n'est pas nécessaire d'utiliser un Lifetime très grand dans les TU.

Le MMR qui reçoit une demande de tunnel acquitte directement celle-ci. Le message Tunnel Acknowledgement contient la liste des groupes demandés et un numéro indiquant le succès ou l'échec de la requête pour chaque groupe.

Dès que les datagrammes d'un groupe multicast arrivent nativement vers le nouveau MMR, celui-ci arrête le tunnel devenu redondant. Pour ce faire, il envoie un TU contenant la liste des groupes qu'il faut encore tunneler au MMR responsable de l'autre bout du tunnel.

Que le MMR reçoive les datagrammes par tunnel ou nativement, il retransmet ceux-ci en multicast (en les décapsulant si nécessaire) sur le lien où se trouve le MN. Ceci permet d'éviter le problème des tunnels parallèles car d'autres hôtes peuvent en profiter.

Il n'y a pas de problème de convergence de tunnels pour les groupes disponibles dans la zone quand il y a un MMR sur le nouveau réseau visité car les tunnels se situent entre MMRs et non pas entre MMRs et MNs.

Jusqu'ici, nous ne nous sommes occupés que des groupes que le mobile pouvait sous-traiter au nouveau MMR. Si ce nouveau MMR se trouve en dehors du site primaire du MN, il ne peut pas avoir accès aux groupes de scope site-local du site primaire du MN. Le MN doit donc créer les tunnels lui-même pour tous les groupes auxquels il était abonné mais qui ne sont plus disponibles d'un point de vue topologique. Le même principe s'applique quel que soit le scope (organization-local, etc).

Un MN demande un tunnel de la même façon que le MMR précédemment. Il doit seulement rajouter une option de destination Home Address contenant sa Home Address pour que le MMR qui reçoit la demande veuille bien accepter de forwarder un groupe multicast hors de son scope normal.

Nous avions fait l'hypothèse auparavant qu'il y avait un MMR sur le nouveau réseau visité. S'il n'y en a pas, le mobile peut toujours initier un processus de découverte dynamique de MMR. Si cela ne donne toujours rien, il faut bien que le mobile se résigne à construire tous les tunnels lui-même.

Néanmoins, il se peut qu'un routeur multicast (non MMR) soit quand même disponible sur le réseau visité. Ceci est visible dans les Router Advertisements car le flag Multicast est à 1 et le flag MMR est à 0. Le mobile peut donc quand même joindre les groupes in-scope de façon native.

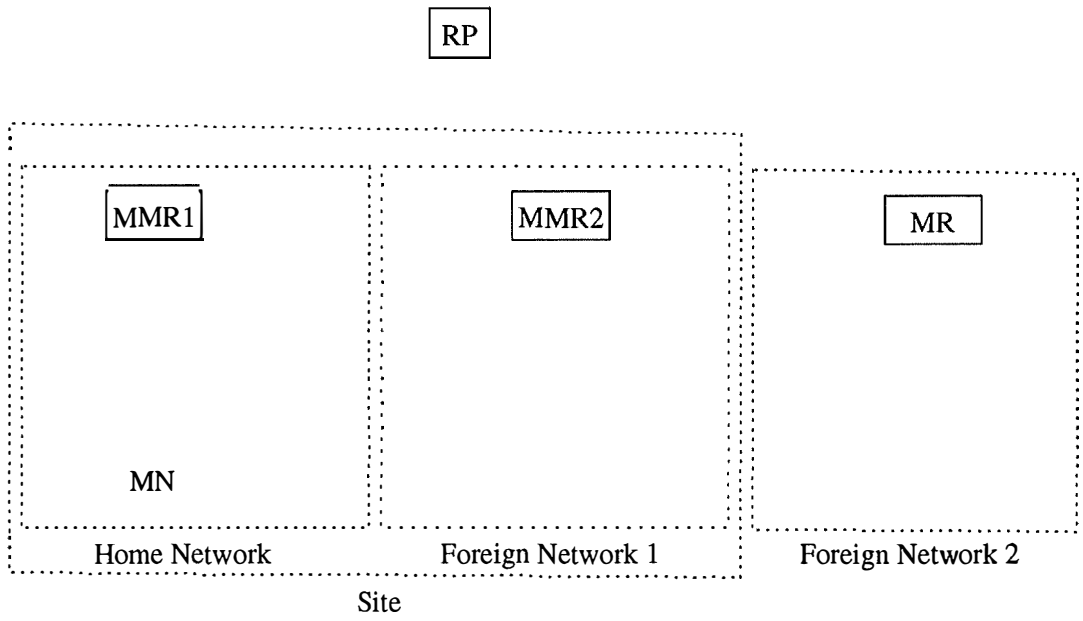

FIG. 6.2 – Contexte

6.2.7 Exemple

Contexte

Soit un noeud mobile situé dans son Home Network. Il s'abonne d'abord à deux groupes : le premier, L, est de scope site-local, tandis que le deuxième, G, est de scope global. Le mobile se déplace d'abord dans un premier FN appartenant au même site dans lequel existe un MMR. Ensuite, il se déplace de nouveau et arrive dans un autre réseau, cette fois hors de son site. Voyons comment le protocole se comporte pour permettre au mobile de continuer ses communications multicast. Remarquons que nous ne détaillerons pas les messages Mobile IPv6. Seules les différences propres au multicast seront explicitées.

Dans le Home Network

Intialement, le MN est dans son HN, où se trouve un MMR (MMR1). Comme MMR1 est forcément un routeur multicast, il s'abonne à L (site-local) et G (global) avec MLD. Après un certain temps, les datagrammes de L et G arrivent vers le MN (voir Figure 6.3).

Passage du Home au premier Foreign Network

Puis le MN se déplace dans un FN du même site où il y a un MMR numéro 2 (voir Figure 6.4). Il émet un MLD Join((L,MMR1),(G,MMR1)). MMR2 qui ne recevait pas L,G de façon native demande à MMR1 un tunnel temporaire pour ces deux groupes. Il émet un TU(L,G) vers MMR1 qui l'acquitte par un TA(L,G). MMR2 joint aussi l'arbre multicast de L et H par son protocole de routage multicast en envoyant un PIM-Join(L) et un PIM-Join(G) vers le RendezVous Point.

Stabilisation du protocole dans le premier Foreign Network

Après un certain temps, les datagrammes de L et G arrivent nativement (voir Figure 6.5). MMR2 demande donc de couper les tunnels à MMR1. Il envoie un TU à MMR1 contenant

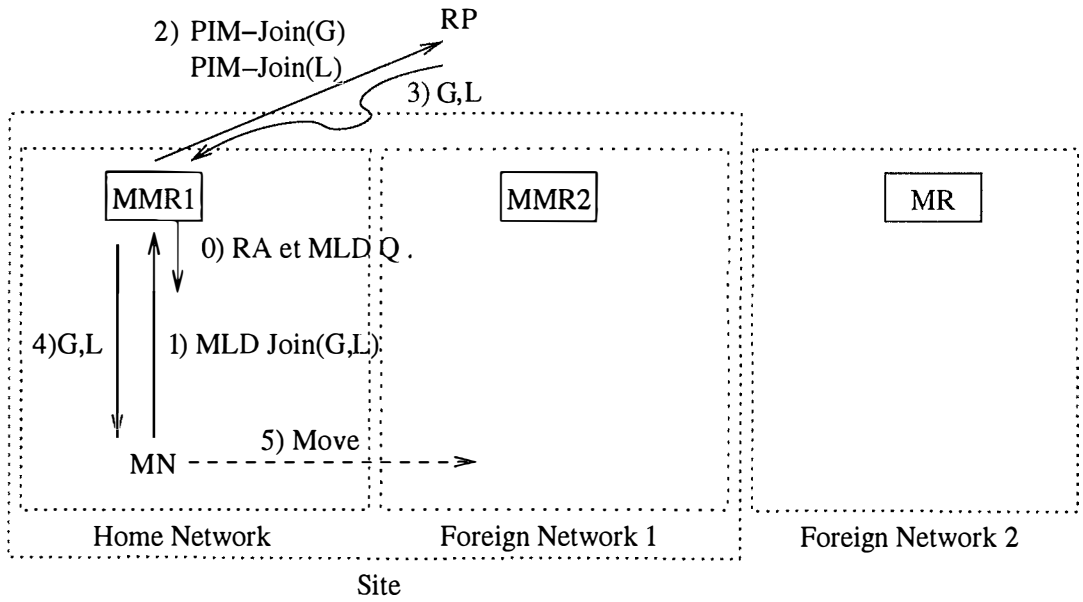

FIG. 6.3 – Dans le Home Network

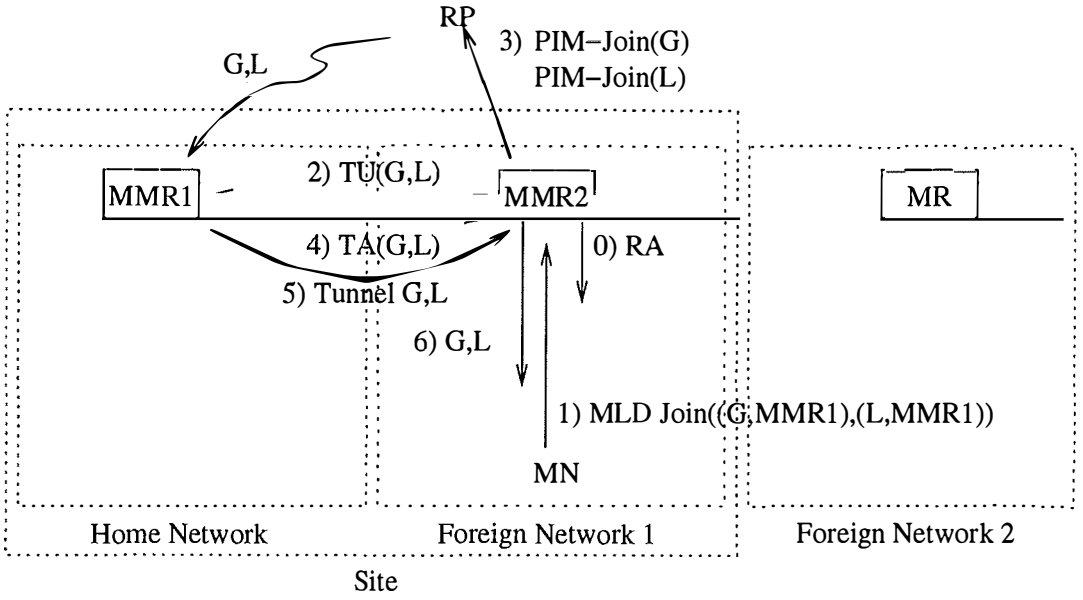

FIG. 6.4 – Passage du Home au premier Foreign Network

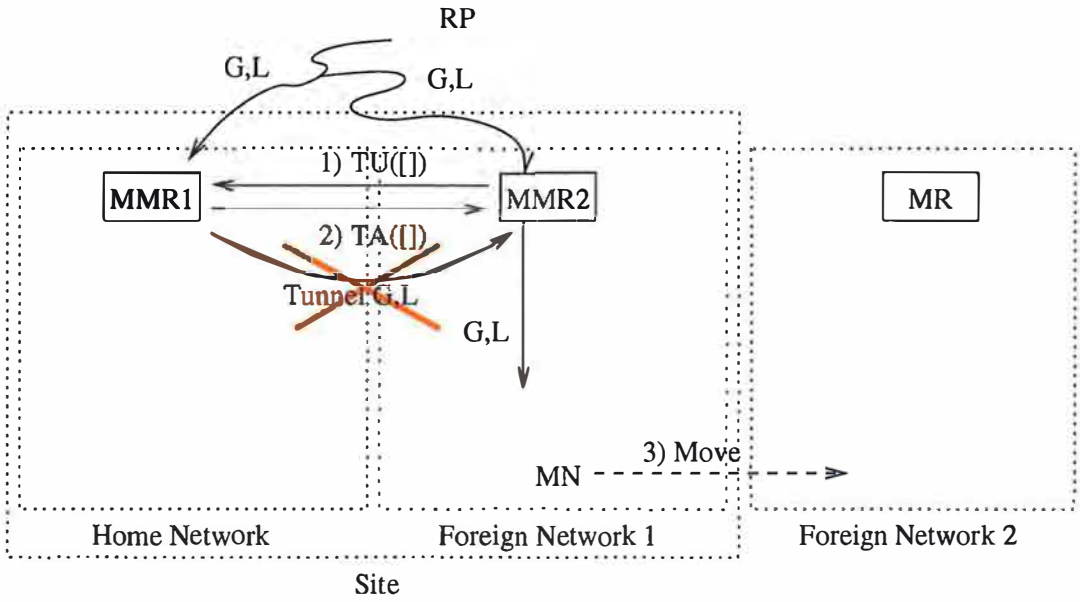

FIG. 6.5 – Stabilisation du protocole dans le premier Foreign Network

une liste vide. MMR1 acquitte de nouveau ce TU et stoppe le tunnel.

Passage du premier au deuxième Foreign Network

Le MN continue à se déplacer et arrive dans FN2 où il n'y a pas de MMR mais juste un routeur multicast (voir Figure 6.6). Le MN s'en rend compte grâce aux flags des RA. Il faut que le MN se débrouille seul pour diminuer le handoff. Il émet un TU [L,H] vers MMR2 (que MMR2 acquitte). Le préfixe annoncé dans le RA lui fait comprendre qu'il est sorti de son site. Le tunnel établi avec MMR2 pour le groupe L sera donc permanent. Tandis que pour le groupe global G, il peut le joindre nativement et c'est ce qu'il fait en émettant un MLD Join(G).

Stabilisation du protocole dans le deuxième Foreign Network

Dès que les datagrammes de G arrivent nativement, le mobile arrête son tunnel pour G avec MMR2 (voir Figure 6.7). Il envoie un TU vers MMR2 avec la liste des groupes pour lesquels il souhaite encore un tunnel, c'est-à-dire la liste singleton [L].

Situation finale

L'exemple se termine. La figure 6.8 illustre la situation finale. On suppose que le mobile ne se déplace plus.

6.2.8 Fonctionnement du protocole pour l'émission

Contrairement à la réception, il n'est pas nécessaire d'établir un tunnel temporaire avec le dernier MMR rencontré. En effet, si le mobile émet de façon native sur son nouveau réseau, tout en rajoutant une option de destination Home Address, le routeur multicast tunnelle les datagrammes dans des messages PIM Register à destination du Rendezvous Point en considérant qu'il a affaire à une nouvelle source. Ensuite, le RP décapsule les datagrammes et

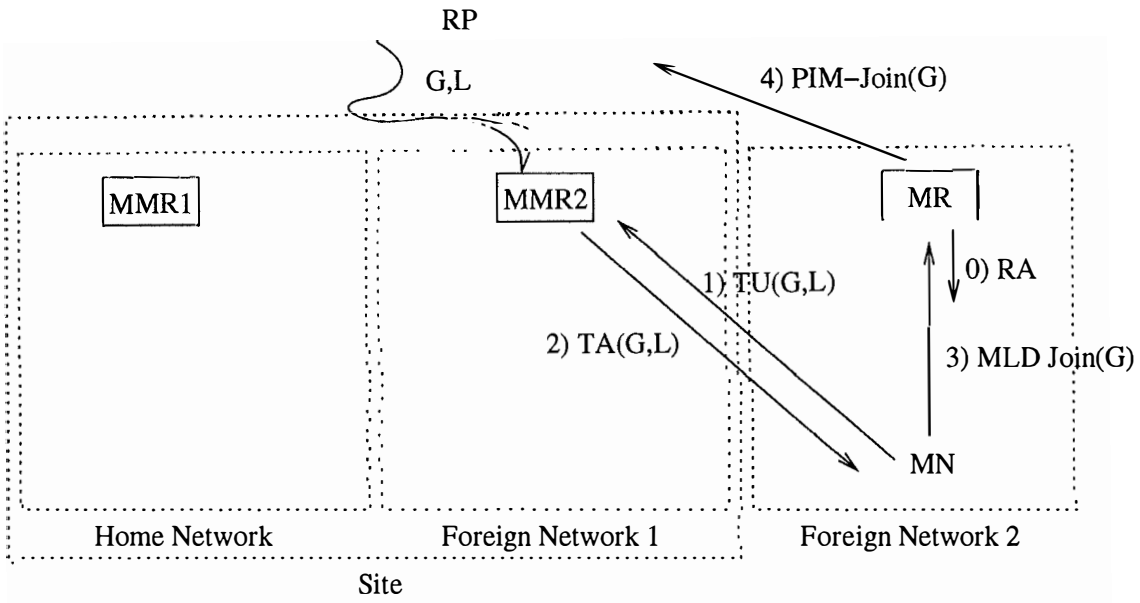

FIG. 6.6 – Passage du premier au deuxième Foreign Network

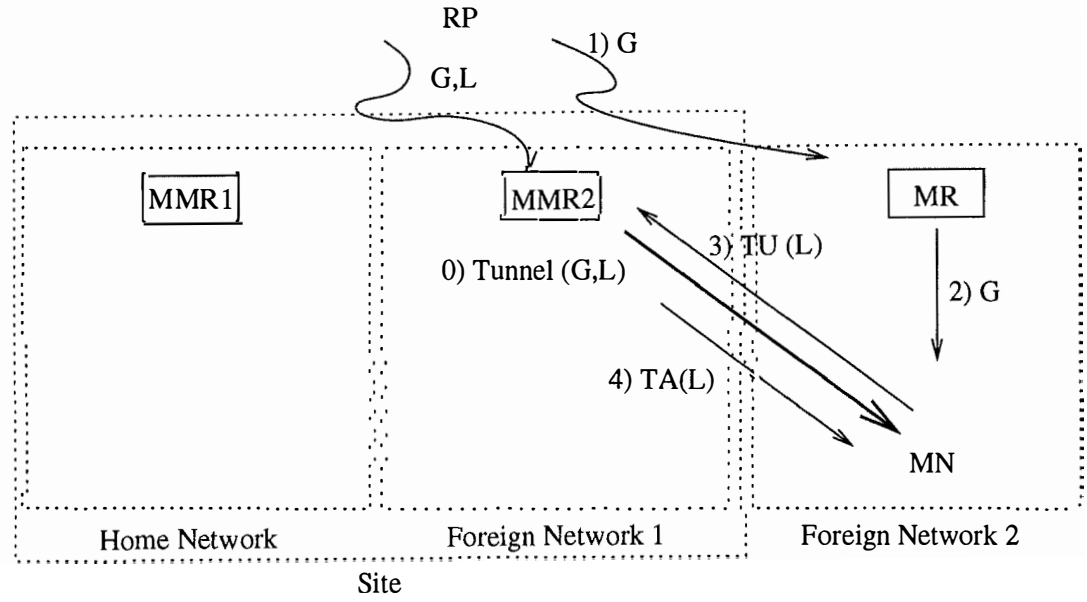

FIG. 6.7 – Stabilisation du protocole dans le deuxième Foreign Network

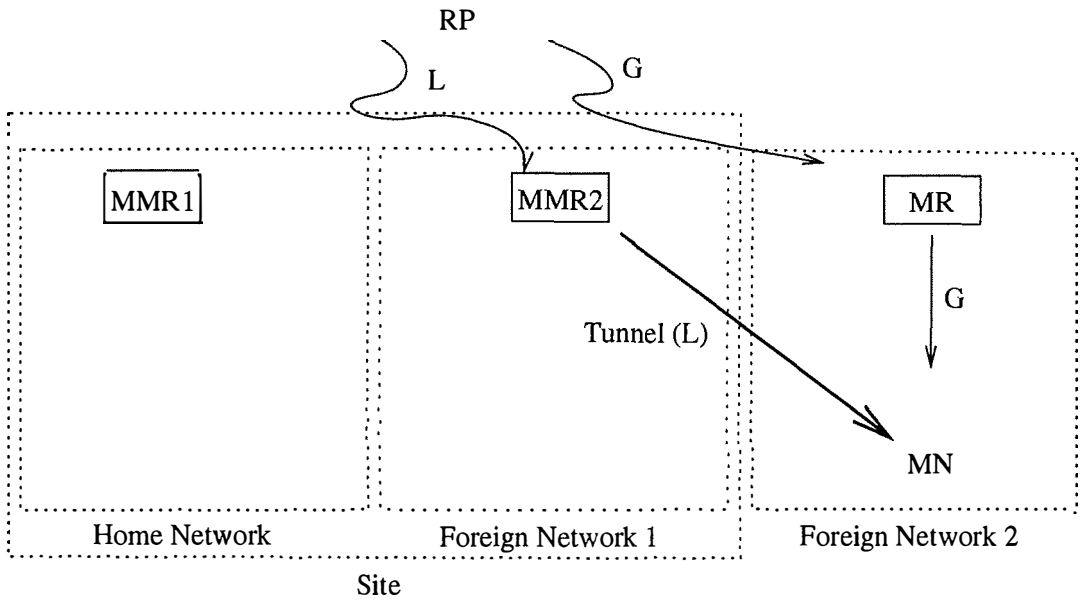

FIG. 6.8 – Situation finale

retransmet les informations sur l'arbre (*,G). Le RP joint aussi le nouvel arbre (S,G) afin de couper le tunnel.

Les routeurs PIM-SM actuels ont un paramètre de configuration qui permet de définir à partir de quel moment le routeur doit joindre l'arbre à la source. La configuration par défaut est telle que le routeur joint (S,G) dès le premier paquet reçu de la source S. Cette façon de procéder est très intéressante quand les sources ne sont pas mobiles, car cela permet de réduire la charge du RP, puisque les routeurs n'ont plus besoin des informations de l'arbre (*,G).

Malheureusement, si les sources sont mobiles, ce n'est pas une bonne idée de joindre directement l'arbre (S,G). En effet, si une source est fortement mobile, le trafic de contrôle généré par les routeurs qui voudront joindre à chaque fois le nouvel arbre sera prohibitif.

Si nous arrivons à mesurer la stabilité d'une source, nous pourrons améliorer l'algorithme de passage du RPT au SPT et trouver un juste milieu entre l'utilisation des arbres SPT et RPT quand les sources se déplacent.

Comportement du MN

Initialisation de l'émission Supposons qu'un mobile veuille commencer à émettre à destination d'un groupe G. Il ajoute une ligne dans sa MSL en suivant les règles suivantes.

S'il n'y a pas de routeur multicast sur le réseau courant, le mobile est obligé de transmettre à destination de G dans un tunnel via son Home-MMR. Pour que cela soit possible, il faut aussi que le Home-MMR ait accès à ce groupe, sinon, la communication ne peut démarrer.

Si le mobile se trouve sur un réseau où il y a un routeur multicast mais pas de MMR, le mobile peut commencer à émettre nativement. Malheureusement, si le Home-MMR n'est pas dans le scope du groupe, la communication sera coupée quand le mobile quittera le scope.

Enfin, si le mobile se trouve sur un réseau pourvu d'un MMR, le mobile peut émettre nativement et établir un tunnel avec le MMR s'il lui arrive de sortir du scope.

Handoff Supposons que le mobile se déplace tout en ayant une communication multicast en cours pour lequel il est émetteur.

Si le mobile reste dans le scope du groupe, il peut continuer à émettre nativement avec sa nouvelle Care-of Address pour peu qu'il y ait un routeur multicast sur le réseau visité. Pour ne pas perturber les récepteurs, le mobile ajoute une option de destination Home Address dans laquelle il place l'adresse qu'il utilisait quand cette communication a débuté. Celle-ci se trouve dans la MSL.

Si le mobile quitte le scope du groupe, il est obligé d'établir un tunnel avec un MMR qui se trouve dans le scope. Le MN émet donc un Tunnel Request avec le flag Send à 1 à destination du MMR contenu dans la ligne de MSL correspondant à la communication.

Comportement du MMR

A la réception d'un datagramme multicast encapsulé, le MMR vérifie que la source S du datagramme encapsulé figure dans sa Tunnel List, que le Send Flag est à 1 et qu'il existe un état (S,G) dans la MRIB (table de routage multicast, *Multicast Routing Information Base*) où G est l'adresse de destination du datagramme encapsulé. Si toutes ces conditions sont remplies, il ajoute, si ce n'est déjà fait, l'interface virtuelle du tunnel dans l'"incoming interface list" de l'état (S,G), il rafraîchit la valeur du timer et enfin, il forwarde le datagramme.

Si le MMR qui reçoit le datagramme du mobile est le Home-MMR de celui-ci, la troisième condition ne doit pas être remplie. En effet, un mobile a le droit de commencer à émettre à destination d'un groupe directement dans un tunnel vers son Home-MMR. Si c'est le cas, le Home-MMR n'a pas encore d'état (S,G). Ce relâchement permet à un mobile d'émettre à destination de groupes disponibles sur son Home Network sans devoir obligatoirement s'y trouver au début de la communication.

Un MMR peut donc refuser de forwarder un datagramme pour ces raisons :
- Le mobile ne s'est pas enregistré auprès du MMR (Le mobile ne figure pas dans la Tunnel List).
- Le flag Send n'est pas positionné à 1 dans la Tunnel List pour cette source (Le mobile s'est enregistré mais n'a pas demandé la fonctionnalité de Tunnel en mode émission).
- Il n'y a pas d'état (S,G) dans la MRIB (MMR non Home-MMR) : la source, quand elle était attachée au MMR, n'a pas émis de datagrammes à destination de G. Il n'y a donc aucune raison pour qu'elle ne fasse pas cela nativement.

Comportement d'un routeur multicast PIM-SM

Notre protocole prévoit aussi une légère modification du fonctionnement des routeurs PIM-SM afin d'éviter une explosion du nombre d'états pour les sources trop mobiles. Cette optimisation est facultative et indépendante du reste de la solution. De plus, seuls les routeurs proches de réseaux sans-fils pourraient en profiter. Il ne faut donc pas forcément implémenter cela sur tous les routeurs PIM-SM du réseau.

Traitement d'un datagramme provenant d'une nouvelle source Quand un datagramme multicast d'une source S' est reçu par un routeur PIM-SM, il y a création d'un état (S',G) si la source est inconnue (le datagramme provient donc forcément de l'arbre RPT).

Toutefois, il peut s'agir d'une source mobile qui s'est déplacée et qui continue d'émettre nativement. Si c'est le cas, le datagramme contient une option de destination Home Address, ce qui permet au routeur de distinguer les sources mobiles des autres sources.

Comme nous l'avons expliqué plus haut, nous maintenons dans les états (S,G) du routeur la Home Address utilisée par S. Il suffit au routeur de rechercher si un état (S,G) contient dans le champ Home Address la même adresse que celle contenue dans le datagramme reçu. Plusieurs états pourraient contenir la même Home Address, c'est pourquoi le routeur sélectionne l'état le plus récent, c'est-à-dire celui contenant la variable "nombre de handoffs" maximale. A partir de cet état, il crée un nouvel état en reprenant comme uptime cumulé celui de l'état choisi et en incrémentant le nombre de handoffs.

Il n'est pas judicieux d'effacer directement l'état correspondant à l'ancienne adresse source car un mobile qui implémenterait un mécanisme de fast-handoff pourrait émettre simultanément par son ancienne et sa nouvelle adresse source. Cependant, il serait quand même intéressant d'éviter une explosion d'états quand une source est vraiment trop mobile. Comme le timeout d'un état est de trois minutes, il serait bon d'effacer l'ancien état si son uptime est inférieur à trois minutes.

Traitement d'un datagramme provenant d'une source active Grâce aux informations supplémentaires qui sont stockées dans les états (S,G), il devient possible de mesurer la stabilité d'une source au moyen du rapport entre l'uptime cumulé et le nombre de handoffs. Au lieu de regarder simplement si le trafic du groupe dépasse un seuil donné comme c'est le cas actuellement quand un routeur souhaite passer en SPT, nous proposons d'ajouter une condition qui consiste à vérifier que la stabilité est supérieure à une valeur donnée.

6.3 Discussion

La solution que nous proposons est déployable car elle supporte des milieux hétérogènes (avec MMR, sans MMR et avec MR, sans MR). Elle privilégie l'optimalité du routage et la diminution du temps de handoff au détriment de la simplicité. Les modifications protocolaires nécessaires ne sont pas excessives et sont réalisables facilement, compte tenu du faible déploiement d'IPv6 et du Multicast. De plus, elle s'intègre et est compatible avec Mobile IPv6. Malheureusement, le support des scopes a fortement complexifié le protocole. La solution peut aussi être vue comme un agrégat d'optimisations de Mobile IPv6. En effet, certaines fonctionnalités sont indépendantes les unes des autres : la modification du fonctionnement des routeurs PIM-SM n'est pas obligatoire, le mécanisme de découverte dynamique du MMR n'est là que pour diminuer l'effort de configuration et la partie émission peut être totalement dissociée de la partie réception. Les différents paramètres qui avaient été évalués sur les solutions existantes ont de nouveau été évalués sur la nouvelle solution, afin de pouvoir mesurer les progrès réalisés (voir Tableau 6.1).

TAB. 6.1 – Comparatif : nouvelle solution

	Nouvelle solution
Support de la réception	oui
Support de l'émission	oui
Conçu pour IPv4	non
Conçu pour IPv6	oui
Protocoles de routage multicast compatibles	tous sauf SSM
Protocoles de gestion de groupes compatibles	MLDv2 + modifications
Entités à modifier :	
Routeur multicast	oui (facultatif)
Noeud mobile	oui
Home Agent	non
Foreign Agent (v4 seulement)	/
Correspondant	non
Complexité relative	assez grande
Compatible avec Mobile IP	oui
Nouvelle entité	MMR
Messages de contrôle (réception) :	
Sur le réseau visité	MLD Join local + TU/TA vers MMR
Dans le protocole de routage multicast	PIM Join en amont sur nouveau FN
Messages de contrôle (émission) :	
Dans le protocole de routage multicast	Extension RPT et plus si passage SPT
Tunnels	temporaires, seulement quand nécessaire
Routage	presque optimal
Problème des tunnels convergents	non
Problème des tunnels parallèles	non
Délai relatif	faible
Informations à maintenir (en plus) sur :	
Routeur multicast du lien	Dans chaque état (S,G) :Home Address, Uptime cumulé, Handoffs counter
Noeud mobile	MSL, MRL
Home Agent	rien
Foreign Agent	/
Correspondant	rien
Nouvelle entité	Tunnel List
Transparence	non
Support des scopes IPv6	oui
Nouvelle entité nécessaire dans chaque FN	non
MR nécessaire dans chaque FN	non
FA nécessaire dans chaque FN	/
Point central d'échec	aucun
Adaptable en v6	/

Conclusion

Nous voici arrivés au terme de ce mémoire. L'enjeu était, rappelons-le, de fournir efficacement le service Multicast aux hôtes mobiles, s'attachant à divers réseaux tout au long de leurs communications.

Nous avons vu que les solutions préliminaires proposées par l'IETF dans Mobile IP sont trop simples pour être efficaces dans tous les cas et qu'elles représentent des extrêmes. Le Bi-directionnal Tunneling offre la transparence mais conduit à un routage catastrophique, tandis que le Local Membership met fortement à contribution le protocole de routage pour obtenir un routage optimal. De plus, la coupure de service est potentiellement longue car la reconstruction des arbres n'est pas un processus instantané.

Les chercheurs se sont d'abord rendu compte que le problème de l'émission était parfaitement dissociable de la réception. Les premières solutions s'attachèrent d'abord à optimiser la réception. Certains problèmes comme la convergence des tunnels ou les tunnels parallèles ont d'abord été résolus dans MoM. RBMoM améliore MoM en introduisant le concept de "zone de service" qui permet d'obtenir un routage quasi-optimal.

MMA évite les zones de service au moyen de deux nouvelles entités : le Multicast Agent et le Multicast Forwarder. En désolidarisant les routages unicast et multicast et en donnant au mobile le rôle d'indiquer au Multicast Agent où celui-ci peut trouver les datagrammes multicast qu'il désire, MMA obtient aussi un routage presque optimal.

Mobicast préfère une vue hiérarchique, en séparant le routage multicast en deux niveaux (à l'intérieur d'un campus et en dehors). La nouvelle entité de Mobicast, le Domain Foreign Agent, se charge de rendre la mobilité intra-campus transparente pour l'extérieur du campus.

MSSMv6 se distingue largement des autres solutions. Premièrement, elle est conçue pour IPv6 (les précédentes ont toutes été conçues pour IPv4, sauf les solutions simplistes de l'IETF qui sont hybrides). Deuxièmement, c'est la seule qui résout le problème du Source-Specific Multicast. La façon de procéder est certainement lourde dans certaines conditions, mais c'est le prix à payer pour obtenir un meilleur routage.

Enfin, MMPuA tente d'utiliser l'anycast pour résoudre tous les problèmes. L'anycast étant très récent, il est encore un peu tôt pour savoir si cette solution est réellement déployable.

Au vu des différentes solutions, on se rend compte que tout l'art du Multicast Mobile consiste à utiliser le moins de tunnels possibles pour optimiser le routage. Mais ceux-ci sont nécessaires pour optimiser le handoff en attendant que le protocole de routage multicast étende l'arbre jusqu'au nouveau réseau visité par le mobile. Les tunnels doivent aussi être courts. Il

vaut donc mieux que ceux-ci soient établis entre le routeur multicast de l'ancien réseau visité et celui du nouveau réseau visité. Pour établir les tunnels, il faut définir de nouveaux messages, appelés *Tunnel Request* et *Tunnel Stop* dans ma solution. Toutes les solutions non simplistes introduisaient une nouvelle entité. Dans notre cas, il s'agit du *Mobile Multicast Router* qui est en quelque sorte la concaténation des deux entités proposées dans MMA. Ce sont là tous les ingrédients de base dans la solution proposée dans ce mémoire.

Néanmoins, les ingrédients ne suffisent pas. Il faut encore développer une solution qui soit déployable, c'est-à-dire qui colle à la réalité. A cet effet, nous sommes partis de l'hypothèse que le réseau reposait sur les protocoles IPv6, Mobile IPv6 et PIM-SM. La justification est très simple : ce sont les protocoles qui ont le plus de chance de s'imposer prochainement. IPv4 commence lentement sa retraite bien méritée, Mobile IPv6 est bien mieux conçu que Mobile IPv4 et PIM-SM gagne progressivement du terrain par rapport aux autres protocoles de routage multicast. Le faible déploiement du multicast a aussi été pris en compte en autorisant les mobiles a établir les tunnels eux-mêmes quand aucun routeur multicast n'est disponible sur le réseau visité. Dès que c'est possible, les tunnels sont créés entre Mobile Multicast Routers, pour éviter les problèmes de tunnels convergents ou parallèles.

Il sera nécessaire de faire encore beaucoup de recherche avant d'arriver à une solution unifiée qui puisse devenir un standard. Tous les problèmes de sécurité qui n'ont été que survolés dans ce mémoire doivent encore être résolus. Certaines notions propres à IPv6 comme les scopes ou l'anycast devront sans doute encore être débattues durant des mois à l'IETF avant d'obtenir une définition stable et claire.

Concernant la solution proposée dans ce mémoire, des efforts sont encore nécessaires avant de pouvoir soumettre un draft à l'IETF. Il s'agit de définir de façon non ambigüe les messages du protocole et le comportement précis des différentes entités suivant la situation rencontrée.

Des simulations permettraient aussi de désigner de façon objective le meilleur protocole. Malheureusement, il y a tellement de paramètres qui entrent en jeu qu'il serait difficile d'en tirer des conclusions globales.

Il serait aussi intéressant de voir si les solutions actuelles sont compatibles avec *Homeless Mobile IPv6* [36], *Hierarchical Mobile IPv6* [42] ou *Fast Handovers for Mobile IPv6* [46], des améliorations de Mobile IPv6 qui sont développées actuellement.

Des chercheurs français examinent actuellement le problème des *réseaux mobiles* [21]. Au lieu d'avoir un mobile qui se déplace de réseaux en réseaux, c'est tout un réseau complet qui change de point d'attachement. Ce ne sont plus les hôtes qui sont mobiles mais les routeurs. Ainsi, supposons que les voyageurs d'un avion soient connectés au réseau local de celui-ci et que le routeur sans-fil de l'avion soit connecté à l'Internet terrestre. Quand l'avion se déplace, les voyageurs restent fixes à l'intérieur de l'avion, mais le routeur de l'avion change de point d'attachement pour continuer à fournir l'accès Internet. Il serait bon de nouveau d'examiner comment fournir le service multicast dans ce contexte.

J'espère que le lecteur aura saisi l'intérêt de ma recherche et que bientôt, n'importe qui pourra participer à des communications multicast depuis son mobile via des réseaux sans-fil.

Bibliographie

[1] Arup Acharya, Ajay Baker, and B.R. Badrinath. *IP Multicast Extensions for Mobile Internetworking.* IEEE 0743-166X/96, March 1996.

[2] A. Ballardie. *Core Based Trees (CBT version 2) Multicast Routing.* Internet RFC, RFC 2189, September 1997.

[3] Christian Bettstetter, Anton Riedl, and Gerhard Geβler. *Interoperation of Mobile IPv6 and Protocol Independent Multicast Dense Mode.* International Conference on Parallel Processing, 2000.

[4] S. Bhattacharyya, C. Diot, L. Guiliano, R. Rockell, J. Meylor, D. Meyer, G. Shepherd, and B. Haberman. *An Overview of Source-Specific Multicast(SSM) Deployment.* Internet Draft, draft-ietf-ssm-overview-01.txt, August 2001.

[5] J. Bound, M. Carney, C. Perkins, and R. Droms. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6).* Internet Draft, draft-ietf-dhc-dhcpv6-21.txt, November 2001.

[6] J. Bound, L. Toutain, O. Medina, F. Dupont, A. Durand, and H. Afifi. *Dual Stack Transition Mechanism (DSTM).* Internet Draft, draft-ietf-ngtrans-dstm-05.txt, November 2001.

[7] Xavier Brouckaert. *Dynamic Mobile IP : Rapport de stage.* Rapport interne France Telecom, January 2002.

[8] Brad Cain, Steve Deering, Bill Fenner, Isidor Kouvelas, and Ajit Thyagarajan. *Internet Group Management Protocol, Version 3.* Internet Draft, draft-ietf-idmr-igmp-v3-09.txt, January 2002.

[9] Ke Chien-An and Liao Wanjiun. *Reliable Mobile Multicast Protocol (RMMP) : A Reliable Multicast Protocol for Mobile IP Networks.* IEEE 0-7803-6596-8/00, 2000.

[10] M. Christensen and F. Solensky. *IGMP and MLD snooping switches.* Internet Draft, draft-ietf-magma-snoop-01.txt, January 2002.

[11] Richard Lin Chunhung and Kai-Min Wang. *Mobile Multicast Support in IP Networks.* IEEE 0-7803-5880-5/00, 2000.

[12] Gisèle Cizault. *IPv6 : Théorie et pratique (troisième édition).* O'Reilly, March 2002.

[13] A. Conta and S. Deering. *ICMP for the Internet Protocol, Version 6.* Internet RFC, RFC 2463, December 1998.

[14] S. Deering. *Host Extensions for IP Multicasting.* Internet RFC, RFC 1112, August 1989.

[15] S. Deering and W. Fenner. *Multicast Listener Discovery (MLD) for IPv6.* Internet RFC, RFC 2710, October 1999.

[16] S. Deering and R. Hinden. *Internet Protocol Version 6 (IPv6).* Internet RFC, RFC 1883, December 1995.

[17] S. Deering and R. Hinden. *Internet Protocol Version 6 (IPv6).* Internet RFC, RFC 2460, December 1998.

[18] J.M. Dinant. *Les risques majeurs de IPv6 pour la protection des données à caractère personnel.* Internet `http://www.droit-technologie.org`, November 2001.

[19] R. Droms. *Dynamic Host Configuration Protocol.* Internet RFC, RFC 2131, March 1997.

[20] D.Waitzman, C. Partridge, and S. Deering. *Distance Vector Multicast Routing Protocol.* Internet RFC, RFC 1075, November 1988.

[21] Thierry Ernst, Ludovic Bellier, Alexis Olivereau, Claude Castelluccia, and Lach Hong-Yon. *Mobile Networks Support in Mobile IPv6 (Prefix Scope Binding Updates).* Internet draft, draft-ernst-mobileip-v6-network-02.txt, 2001.

[22] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. *Protocol Independent Multicast-Sparse Mode (PIM-SM).* Internet RFC, RFC 2362, June 1998.

[23] B. Fenner, H. Holbrook, and I. Kouvelas. *Multicast Source Notification of Interest Protocol (MSNIP).* Internet Draft, draft-ietf-magma-msnip-00.txt, February 2002.

[24] W. Fenner. *Internet Group Management Protocol (IGMP) Version 2.* Internet RFC, RFC 2236, November 1997.

[25] Mark Handley, Isidor Kouvelas, Tony Speakman, and Lorenzo Vicisano. *Bi-directional Protocol Independent Multicast (BIDIR-PIM)*, June 2001.

[26] Tim G. Harrison, Carey L. Williamson, Wayne L. Mackrell, and Richard B. Bunt. *Mobile Multicast (MoM Protocol) : Multicast Support for Mobile Hosts.* MobiCom 97, September 1997.

[27] Shin Hee-Sook and Suh Young-Joo. *Multicast Routing Protocol in Mobile Networks.* IEEE 0-7803-6283-7/00, 2000.

[28] C. Jelger and T. Noel. *Supporting Mobile SSM Sources for IPv6.* Internet Draft, draft-jelger-mssmsv6-00.txt, July 2002.

[29] Weijia Jia, Wanlei Zhou, and Joerg Kaiser. *Efficient algorithm for mobile multicast using anycast group.* IEE Proc. Communications, February 2001.

[30] David B. Johnson and Charles Perkins. *Mobility Support in IPv6.* Internet Draft, draft-ietf-mobileip-ipv6-15.txt, July 2001.

[31] Mohammed Kassi-Lahlou and Christian Jacquenet. *Dynamic Mobile IP (DMI).* Internet Draft, draft-kassi-mobileip-dmi-00.txt, January 2001.

[32] H. Kitamura. *A SOCKS-based IPv6/IPv4 Gateway Mechanism.* Internet RFC, RFC 3089, April 2001.

[33] A.J. McAuley, E. Bommaiah, A. Misra, R. Talpade, S. Thomson, and K.C. Young, Jr. *Mobile Multicast Proxy.* IEEE 0-7803-5538-5/99, 1999.

[34] T. Narten and R. Draves. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* Internet RFC, RFC 3041, January 2001.

[35] T. Narten, E. Nordmark, and W. Simpson. *Neighbor Discovery IP Version 6 (IPv6).* Internet RFC, RFC 2461, December 1998.

[36] P. Nikander, J. Lundberg, C. Candolin, and T. Aura. *Homeless Mobile IPv6.* Internet Draft, draft-nikander-mobileip-homelessv6-01.txt, February 2001.

[37] Nokia. *MITA : Mobile Internet Technical Architecture.* IT Press, ISBN 951-826-499-6.

[38] E. Nordmark. *Stateless IP/ICMP Translation Algorithm (SIIT).* Internet RFC, RFC 2765, February 2000.

[39] C. Perkins. *IP Mobility Support.* Internet RFC, RFC 2002, October 1996.

[40] C. Perkins. *IP Mobility Support for IPv4.* Internet RFC, RFC 3220, January 2002.

[41] Kaur Satwant, Madan Bharat, and Ganesan Subra. *Multicast support for Mobile IP using a modified IGMP.* IEEE 0-7803-5668-3/99, 1999.

[42] Hesham Soliman, Claude Castelluccia, Karim El-Malki, and Ludovic Bellier. *Hierarchical MIPv6 mobility management.* Internet Draft, draft-ietf-mobileip-hmipv6-03.txt, February 2001.

[43] Cheng Lin Tan and Stephen Pink. *MobiCast : A Multicast Scheme for Wireless Networks,* 1999.

[44] S. Thomson and T. Narten. *IPv6 Stateless Address Autoconfiguration.* Internet RFC, RFC 2462, December 1998.

[45] G. Tsirtsis and P. Srisuresh. *Network Address Translation - Protocol Translation (NAT-PT).* Internet RFC, RFC 2766, February 2000.

[46] G. Tsirtsis, A. Yegin, C. Perkins, G. Dommety, K. El-Malki, and M. Khalil. *Fast Handovers for Mobile IPv6.* Internet Draft, draft-ietf-mobileip-fast-mipv6-01.txt, April 2001.

[47] K. Tsuchiya, H. Higuchi, and Y. Atarashi. *Dual Stack Hosts using the Bump-In-The-Stack Technique (BIS).* Internet RFC, RFC 2767, February 2000.

[48] R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, and B. Haberman. *Multicast Listener Discovery Version 2 (MLDv2) for IPv6.* Internet Draft, draft-vida-mld-v2-02.txt, January 2002.

[49] L. Wei, D. Estrin, D. Farinacci, A. Helmy, D. Meyer, S. Deering, and V. Jacobson. *Protocol Independant Multicast Version 2 Dense Mode : Protocol Specification,* June 1999.

[50] Yoon Wonyong, Lee Dongman, Yu Chansu, and Kim Myungchul. *Tree-Based Reliable Multicast in Combined Fixed/Mobile IP Networks.* IEEE 0-7695-0912-6/00, 2000.