UNIVERSITE KASDI MERBAH OUARGLA

FACULTE DES SCIENCE ET SCIENCES DE L'INGENIEUR.
****

DEPARTEMENT DE MATHEMATIQUES ET D'INFORMATIQUE

N° d'ordre :
N° de série :

Mémoire

Présenté pour l'obtention du diplôme de

MAGISTER

Spécialité : Informatique
Option : Informatique **et C**ommunication **E**lectronique

Par : MEDILEH Saci

Thème

Solutions pour la prise en charge de la mobilité dans l'Internet

Soutenu publiquement le : 24 / 05 / 2009

Devant le jury composé de :

Dr. Mohamed Benmohammed, Professeur	**(Constantine)**	**Président**
Dr. F.Z. LAALLAM, Maître de Conférences	**(Ouargla)**	**Examinateur**
Dr. Brahim BELATTAR, Maître de Conférences	**(Batna)**	**Examinateur**
Dr. Azeddine BILAMI, Maître de Conférences	**(Batna)**	**Rapporteur**

Remerciements

J'aimerais adresser de très grands remerciements à Mr. BILAMI Azeddine mon encadreur, pour ce sujet qu'il m'a proposé, pour ses encouragements à la réalisation de ce mémoire ainsi que pour ses précieux conseils et son aide.

Je suis très reconnaissent à nos enseignants de Post-graduation et le chef de département informatique, pour les efforts consacrés tout au long de notre formation.

J'exprime mes sincères remerciements aux membres du jury qui m'ont fait l'honneur d'accepter de juger ce travail.

En fin, un amical remerciement le plus vif et le plus sincère à tous mes collègues de Post-graduation et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

MEDILEH Saci

Résumé

La mobilité dans l'Internet est considérée comme le changement du point d'attachement d'un terminal mobile, lors de ses déplacements, en passant d'un réseau local à un autre. Elle est vue comme un problème de translation d'adresse qui est naturellement résolu au niveau de la couche réseau. Lorsque la mobilité est supportée au niveau de cette couche, le protocole de routage sera capable de router les paquets de et vers le terminal mobile, même si ce dernier a changé son point d'attachement au réseau.

Plusieurs travaux s'intéressent actuellement aux problèmes posés par les différentes mobilités (micro, macro…), pour une prise en compte rapide des déplacements des terminaux mobiles au sein du protocole IPv6.

Ce travail consiste en l'étude des solutions pour la prise en compte de la mobilité au niveau IPv6, en vue de faire un état de l'art complet de toutes les solutions émergentes pour résoudre les problèmes liés à la mobilité des terminaux dans l'Internet.

La solution de HMIP2L que nous proposons dans ce mémoire, est basée sur le protocole de HMIPv6 avec multiplicité des MAPs. Ces MAPs sont organisés en hiérarchie optimale à deux niveaux, des simulations et des mesures de facteurs de performances sous NS2 sont menées pour la validation de cette solution.

Mots clés :

Macromobilité, Micromobilité, Mobile IPv6 (MIPv6), mobile IPv6 hiérarchique (HMIPv6), Mobility Anchor Point (MAP).

Abstract

Mobility in the Internet is seen as the changing of the attachment point of a mobile terminal when, moving from a local area network to another. It is considered as an address translation problem, which is naturally solved at the network layer. When mobility is supported at this layer, the routing protocol will be able to route packets from and to the mobile terminal even if it has changed its point of attachment to the network.

Several works are currently interested in the mobility problems (micromobility, macromobility…). These works aim to consider the displacement of mobile node within the IPv6 protocol.

In this report, we purpose a study of the solutions that take into account the mobility problem on the IPv6 level, in order to make a state of the art of all the emergent solutions to solve the problems related to the mobility of terminals in the Internet.

Our HMIP2L proposed solution is based on an improvement of HMIPv6 protocol with multiple MAPs. These MAPs are organized into two optimum hierarchical levels. Simulations and measurements of performances factors under NS2 are considered for the validation of this solution

Key words:

Macromobility, Micromobility, Mobile IPv6 (MIPv6), Hierarchical mobile IPv6 (HMIPv6), Mobility Anchor Point (MAP).Mobility management.

ملخص:

يعرف التنقل عبر الانترنت على أنه تغيير لنقطة اتصال جهاز محمول لحظة انتقاله من شبكة محلية إلى أخرى. والتي يمكن حلها على مستوى طبقة الشبكة في نموذج الطبقات للبرتوكول IP.

يقوم برتوكول التوجيه بتوجيه الرسائل أو الحزم، من وإلى الجهاز المحمول مهما غير نقطة اتصاله بالشبكة.

قامت عدة دراسات وأبحاث لحل مشكلة التنقل عبر شبكة الانترنت في نطاق واسع أو نطاق محلي وذلك بجعل بروتوكول التوجيه IP يعمل ويتصرف بسرعة في حالة تحرك هذه الأجهزة وتغيير نقطة ارتباطها في الشبكة.

نقترح في هذه المذكرة حلا يعتمد على بنية بروتوكول التنقل الشجري HMIPv6 القياسي وذلك بمضاعفة عدد المحطات الخاصة بتسيير التنقلية داخل الشبكة MAP ووضعها في بناء شجري من مستويين اثنين، سميناه HMIP2L . لإثبات نجاعة هذا الحل قمنا بإجراء محاكاة لنموذج شبكة مصغر ببرنامج المحاكاة NS-2 محاولين قياس عدة وسائط محددة لهذا الغرض.

الكلمات المفتاحية:
التنقل عبر الانترنت، البروتوكول IP6، بروتوكول التنقل الشجري HMIPv6، نقاط تسيير التنقلية MAP.

Table des matières

Table des figures

Introduction

L'évolution des technologies a rendu possible l'apparition de nouveaux types de machines. Ces ordinateurs connus sous le nom de portables permettent à leurs utilisateurs de travailler sur différents sites. Ces ordinateurs disposent de suffisamment d'autonomie et de puissance de calcul pour répondre aux besoins actuels. Les ordinateurs portables permettent de travailler sur des données stockées localement, mais leurs utilisateurs ont de plus en plus besoin de pouvoir communiquer pendant leurs déplacements avec leurs entreprises ou d'autres collaborateurs. Jusqu'à présent ces liaisons s'effectuaient lorsque le portable se trouvait dans un lieu équipé, par exemple d'une connexion réseau ou d'une prise téléphonique. On reliait le portable à cette prise (réseau ou téléphonique), ce qui lui permettait de pouvoir entrer en communication avec son ou ses destinataires. On parle alors d'ordinateurs nomades. Mais ce type de communication n'est possible que lorsque le portable se trouve dans un local équipé. Les communications sont stoppées dès que l'ordinateur nomade est débranché de la prise réseau.

L'arrivée de nouvelles technologies de transmission de données sur des réseaux sans fil, permet de ne plus voir les ordinateurs portables comme de simples machines nomades. A l'heure actuelle, il est possible pour un ordinateur portable de communiquer même pendant ses mouvements. On ne parle plus d'ordinateurs nomades mais d'ordinateurs mobiles.

L'Internet sans fil est actuellement en pleine expansion. En effet, on voit arriver sur le marché divers produits permettant la communication entre équipements mobiles. Ces produits utilisent des technologies différentes et peuvent grossièrement se classer en deux catégories : les réseaux cellulaires pour la téléphonie mobile et les réseaux IP pour l'Internet. Les réseaux cellulaires comme GPRS se focalisent sur la meilleure gestion possible de la mobilité, c'est-à-dire permettre aux utilisateurs de pouvoir se déplacer sans rompre les communications en cours. En contre parti, le débit offert est très limité, ce qui limite la portée de ces équipements cellulaires à la téléphonie. A l'heure actuelle, de nouvelles technologies sont en train d'émerger (UMTS), offrant un débit plus élevé. Cela devrait permettre d'étendre le domaine d'application.

Donc, le but recherché est d'offrir à ces ordinateurs mobiles la possibilité de se servir de l'Internet afin de communiquer avec d'autres machines, que celles-ci soient fixes ou mobiles. Comme le protocole IP a été conçu bien avant l'apparition de ces nouvelles technologies, ses fonctionnalités ne pouvaient tenir compte de ce nouveau type d'utilisation.

Il a donc été nécessaire d'enrichir le protocole afin que celui-ci permette le support de la mobilité.

Ce mémoire est structuré de la manière suivante : le premier chapitre donne une description détaillée de la problématique de mobilité des terminaux dans les réseaux IP plus particulièrement l'Internet. Le deuxième chapitre a pour objectif de présenter un état de l'art complet de toutes les solutions émergentes pour résoudre les problèmes liés à la mobilité des terminaux dans l'Internet. Une étude comparative de ces solutions (Mobile IPv6, Hierarchical MIPv6…) dégagera les points forts ainsi que les insuffisances de chacune, et fait l'objet de troisième chapitre. Dans le quatrième chapitre nous proposons une solution, qui consiste en une modification du protocole HMIPv6. La présentation du simulateur NS-2, et l'élaboration du modèle de simulation avec une analyse des résultats font l'objet de cinquième chapitre.

Chapitre 1 :

La problématique de la mobilité des réseaux

Chapitre 1 : La Problématique de la Mobilité des Réseaux

La mobilité dans l'Internet a été introduite par l'organisme de standardisation IETF[1] qui s'est principalement penché sur la gestion des déplacements d'un ordinateur mobile sur l'Internet, c'est-à-dire du passage d'un réseau local à un autre réseau local. Ce travail a permis de définir un protocole appelé Mobile IP. Les travaux récents qui étudient l'utilisation des réseaux cellulaires (Cellular IP, Hawaii …) montrent les différents problèmes posés par la gestion des différentes mobilités des terminaux et la prise en compte rapide des déplacements. La plupart de ces travaux sont proposés par des universités américaines et ne sont qu'à leur début : ils n'étudient à l'heure actuelle que les problèmes de handoffs à l'intérieur d'une seule technologie d'accès.

L'Internet du futur aura plusieurs objectifs :

- Utiliser les nouveaux protocoles afin de supporter de plus en plus de réseaux et d'utilisateurs : cela passera par l'adoption d'IPv6 et des protocoles associés.
- Expérimenter de nouvelles solutions et des nouvelles fonctionnalités. La mobilité des utilisateurs est l'un des points forts de cet aspect. L'Internet mobile nouvelle génération n'existe pas encore et il convient de définir rapidement les protocoles qui permettront sa démocratisation au sein des communautés d'utilisateurs mais également son adoption par les différents opérateurs.

Il existe à ce jour peu de projets qui proposent, étudient et développent des solutions pour l'Internet Mobile nouvelle génération. Il est donc essentiel de pouvoir proposer une vision globale de la prise en compte sur les terminaux mobiles de la mobilité aussi bien au niveau IPv6 qu'au niveau de la gestion des interfaces multiples sans fil.

La problématique de la mobilité des réseaux est liée au modèle d'adressage de TCP/IP.

1.1 Modèle d'Adressage

L'ensemble des noeuds se trouvant sur le même lien logique constitue un sous-réseau. Les noeuds sont de deux types. Ceux qui relient un sous-réseau `a un autre sont des routeurs, les autres de simples stations. A chaque sous-réseau, correspond un préfixe qui permet d'identifier la position du sous-réseau dans la hiérarchie de l'Internet. Un réseau est donc un ensemble de sous-réseaux partageant le même préfixe IP et connectés `a l'Internet par le biais d'un ou plusieurs routeurs externes. Tous les noeuds ayant une interface sur un sous-réseau donné ont donc une adresse IP correspondant au préfixe de ce sous-réseau.

Cette adresse identifie `a la fois la position topologique du noeud, et le noeud lui-même. L'adresse IP est donc intrinsèquement liée `a la position du noeud dans la topologie Internet.

1.2 Problématique de la Mobilité

Un noeud mobile (MN) [Thierry 2001] est un noeud qui change son point d'attachement dans la topologie Internet, c'est-à-dire qui se déplace d'un sous-réseau à un autre. Dans le cas des stations, nous parlons de station mobile, dans le cas de routeur, de routeur mobile (MRs). Les routeurs d'accès (ARs) sont les routeurs qui desservissent les liens o`u il est permis aux noeuds mobiles de prendre ancrage. Le point d'attachement initial dans le réseau mère est appelé le sous-réseau mère tandis que chaque point d'attachement subséquent est appelé sous-réseau visité.

1 Internet Engineering Task Force, http://www.ietf.org, Organisme de standardisation des protocoles Internet,

Le problème posé par la mobilité vient essentiellement du modèle d'adressage de TCP/IP qui confond le rôle d'identifiant d'interface de l'adresse IP, et son rôle d'identifiant de la localisation dans la topologie Internet qui est hiérarchisée. Si un noeud change d'emplacement dans la topologie Internet l'adresse IP qui identifie l'interface qui change d'emplacement doit changer.

Le changement d'adresse a pour conséquence de couper les sessions ouvertes qui se servent de l'adresse IP comme identificateur tandis que le changement d'emplacement nécessite un re-routage. Le support de la mobilité a donc pour but, d'une part de définir un mécanisme permettant de maintenir les sessions ouvertes lors des déplacements, et d'autre part de déterminer la nouvelle position du noeud dans la topologie (localisation et routage). Ceci se fait généralement au prix de messages de signalisation.

1.3 Problématique de la Mobilité des Réseaux Mobiles

Un réseau mobile [Thierry 2001] est défini comme un ensemble de sous-réseaux connectés `a l'Internet par l'intermédiaire d'un ou plusieurs routeurs (MRs) qui changent leurs points d'ancrage (AR) à l'Internet. Ils sont connectés à l'Internet par le biais d'un ou plusieurs routeurs mobiles3. Les interfaces d'un MR connectées sur un sous-réseau mère ou un sous-réseau visité sont nommées interfaces externes tandis que toutes les autres interfaces sont nommées interfaces internes. Les Noeuds du réseau mobile (MNN, terme générique) localisés à l'intérieur du réseau mobile sont de plusieurs types.

Il convient de différencier une station fixe résidant de manière permanente dans le réseau mobile (station fixe locale ou LFN), d'une station mobile appartenant au réseau mobile (station mobile locale ou LMN), et d'une station mobile n'appartenant pas au réseau mobile mais s'y attachant (station mobile étrangère ou VMN). Ceci est illustré sur la Figure 1 qui montre un réseau mobile se déplaçant de son sous-réseau mère vers un autre. Que ce soit un noeud qui se déplace ou tout un réseau, le problème est relativement similaire. Cependant, à la problématique habituelle de la mobilité s'en ajoute d'autres propres aux réseaux mobiles.

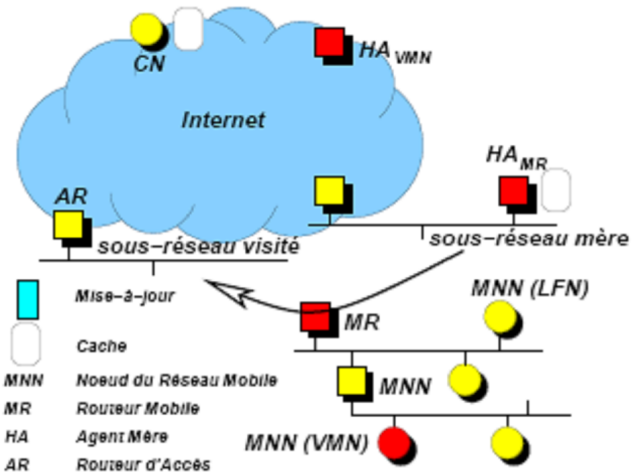


Figure 1. : Terminologie pour les Réseaux Mobiles

Tout d'abord, comme tous les paquets transmis entre MNNs et leurs correspondants (CNs) transitent nécessairement par un MR, le changement de point d'ancrage du seul MR a un impact sur le routage vers l'ensemble des MNNs. Ceux-ci peuvent donc sembler mobiles du point de vue des CNs. En revanche, la structure interne d'un réseau mobile est préservée lors des déplacements du MR. Son déplacement ne causant pas de changement du point

d'ancrage physique des MNNs, seul le (ou les) MR est tenu de changer l'adresse de son interface externe.

Les MNNs peuvent conserver leur adresse. Le changement de point d'ancrage doit donc si possible être géré de manière transparente pour les MNNs afin de ne pas nécessiter de fonctionnalités nouvelles dans l'ensemble des implémentation IPv6.

La mobilité enchaînée (ou imbriquée) est un type de configuration rendue possible par la mobilité des réseaux. Dans le cas d'une VMN, nous faisons face à une double mobilité, celle du réseau mobile, et celle du VMN qui prend ancrage dans le réseau mobile. Dans le cas d'un routeur mobile faisant lui-même office de passerelle à un réseau mobile qui `a son tour permet l'ancrage d'un VMN (station ou routeur), nous avons trois niveaux de mobilité. La mobilité des réseaux rend en principe possible un nombre illimité de niveaux de mobilité. Deux niveaux de mobilité sont illustrés sur la Figure 1. Le réseau mobile y accueille une station mobile (VMN) issue d'un autre sous-réseau (identifié par HAVMN). La mobilité imbriquée permet donc d'imaginer tout un ensemble de configurations. Pour chacune se pose la question d'un routage dit triangulaire qu'il faut optimiser.

Enfin, nous avons parlé de mobilité globale, qui est exacerbée par les applications et configurations possibles des réseaux mobiles. Dans un tel cas, la sécurisation des données de contrôle, le contrôle d'accès aux ressources du réseau visité, et la possibilité de se connecter par l'intermédiaire de plusieurs routeurs mobiles disposant au total de plusieurs interfaces externes (réseau multi-domicilié) posent des questions cruciales [Thierry 2001].

1.4 Le Support de la Mobilité des Réseaux

Le problème d'adressage peut être résolu de plusieurs manières. Le moyen le plus répandu, qui est aussi celui préconisé par l'IETF, fait usage de deux adresses. L'une est permanente et utilisée en tant qu'identifiant d'interface ; l'autre est temporaire et est utilisée en tant qu'identifiant de la position dans la topologie (routage). C'est l'approche la plus évidente et la plus simple car elle ne remet pas en cause la pérennité des différents protocoles de la couche réseau et ne nécessite l'ajout de fonctionnalités aux entités mobiles (au sens IP du terme) ainsi que dans un routeur situé sur le sous-réseau mère (home agent ou HA).

Le support de la mobilité des réseaux [Thierry 2001] est pris en compte par le groupe NEMO[1] de l'IETF. Le groupe est sur le point de finaliser une solution simple (NEMO Basic Support) afin de pouvoir déployer rapidement des réseaux mobiles. Cette solution est établie sur le modèle Mobile IPv6 (protocole de gestion de la mobilité des stations) selon des règles préalablement édictées par le groupe de travail. L'une des règles fondamentale est de ne pas imposer de modifications sur les noeuds localisés derrière le routeur mobile (LFNs) et de maintenir les sessions, sans optimisation de routage. NEMO Basic Support permet d'établir un tunnel bidirectionnel entre le HA et le MR. Le principe de base est que tous les noeuds du réseau mobile partagent le (ou les) même préfixe.

Le protocole établit ainsi une relation entre le préfixe commun `a toutes les stations résidant dans le réseau mobile et l'adresse temporaire (careof address) MRcoa obtenue par le routeur mobile à chacun de ses points d'ancrage. Cette relation est enregistrée auprès du HA au moyen de nouvelles options contenues dans le message de mise à jour (BU). Ce message instruit le HA de re-diriger les paquets destinés aux stations résidants dans le réseau mobile en les encapsulant du HA jusqu'au MR.

Le groupe traite également des problèmes de sécurité et de multi-domiciliation. La question de l'optimisation de routage n'est pour l'instant pas abordée. En revanche, le groupe ne traite pas pour l'instant des problèmes d'optimisation de routage pour lequel il existe déjà.

1 Référence a la transcription anglaise de "la mobilité des réseaux" : NEtwork MObility

Chapitre 2 :

La mobilité : état de l'Art

Chapitre 2 : La mobilité : Etat de l'Art

2.1 Introduction

Le terme de mobilité défini la situation intermédiaire entre le nomadisme et les réseaux ad hoc. Le nomadisme représente le déplacement d'un équipement IP entre des communications. Le nomadisme doit permettre à l'utilisateur de ne pas avoir à reconfigurer son équipement lui-même après chacun de ses déplacements ; Cette fonctionnalité passe par une gestion centrale d'adressage. Ce mécanisme est utilisé principalement par les fournisseurs de service Internet.

À l'opposer, les réseaux ad hoc représentent des réseaux composés uniquement d'hôtes mobiles. Chaque hôte mobile a une fonctionnalité de routage et les hôtes mobiles maintiennent des routes entre eux en fonction de leur joignabilité.

Enfin, la mobilité d'un équipement IP dans l'Internet est le cas intermédiaire ; C'est la possibilité pour un hôte mobile de poursuivre ses communications pendant un changement de point d'attache à l'Internet. Les communications deviennent donc indépendantes de la localisation et l'hôte mobile doit toujours pouvoir continuer à utiliser son adresse IP principale. Le protocole qui résout les problèmes associés à la mobilité est Mobile IP. Bien entendu, ce protocole doit permettre des communications avec des hôtes correspondants qui ne l'implémentent pas.

L'objet de cette partie est de donner un aperçu de la mobilité IP telle que décrite dans le protocole Mobile IP, son fonctionnement et ses limites. On verra donc dans la partie suivante, quelques nouveaux termes associés à Mobile IP. Dans les deux parties successives, on détaillera tout d'abord l'échange de messages qui a lieu lors du changement de point d'attache d'un mobile puis on explicitera le déroulement des communications des mobiles sur l'Internet, aussi bien pour la version 4 que pour la version 6 du protocole Internet IP.

2.2 Les technologies sans fil

Il existe actuellement deux technologies en cours de normalisation [Thomas 2006], pour le support des transmissions sans fil dans un réseau local : la technologie WaveLAN et le protocole 802.11. Ce dernier est très récent, il vient d'être ratifié par l'IEEE[1]. Nous commençons par faire un état des fonctionnalités des différents protocoles. Ces deux procédés permettent la mise en place d'un réseau local sans fil, appelé plus communément WLAN (Wireless LAN) ou RLAN (Radio LAN). Le débit d'un réseau WLAN varie de quelques kbits/s à quelques Mbit/s. Un réseau local sans fil est composé de deux types d'équipements :

- Les points d'accès qui font le lien entre le réseau filaire et les équipements terminaux sans fil.
- Les cartes d'accès sans fil qui équipent les stations et permettent à ces dernières de communiquer avec le reste des équipements du réseau en s'affranchissant du câblage.

La vitesse de déplacement des équipements munis des cartes d'accès sans fil se situe dans des limites fixées entre 5 et 10 km/h. Actuellement la normalisation française autorise ces équipements à fonctionner dans une bande de fréquence située entre 2,446 et 2,4835 Ghz.

1 Institut of Electrical and Electronics Engineers, www.ieee.org, Organisme de standardisation des technologies de transmission,

2.2.1 La technologie WaveLAN

La technologie WaveLAN est une technologie propriétaire [Thomas 2006]. Plusieurs constructeurs ont développé des solutions de réseaux locaux sans fil. Chacune de ces solutions est généralement incompatible avec les autres. Toute fois, plusieurs d'entre eux utilisent les mêmes composants de base. C'est notamment le cas de la société DIGITAL (produits commercialisés actuellement par la société CABLETRON) qui utilise la technologie issue des produits de la société LUCENT TECHNOLOGIES. Il existe également d'autres produits comme ceux fournis par la société BREEZECOM. L'ensemble de ces produits, même s'ils sont incompatibles entre eux, offre généralement les mêmes fonctionnalités. Parmi celles-ci je citerai, l'attachement de station à un point d'accès. Cette fonctionnalité de base permet à une station d'éviter le problème du câblage. En effet, il suffit d'un point d'accès relié au réseau filaire et d'une carte sans fil équipée d'une antenne. Celle-ci est branchée dans la station et lui permet de se comporter comme n'importe quel autre équipement de type ethernet. Cette fonctionnalité est généralement enrichie par le support du nomadisme (roaming). Ce procédé permet à une station équipée toujours de la même carte de se déplacer et de changer de base d'accès de manière transparente. Cette fonctionnalité n'est valable que dans le cas d'un réseau ponté (ou réseau plat). Si un routeur sépare les équipements d'interconnexion sans fil, les fonctionnalités de nomadisme ne permettent plus de transmettre du trafic IP. Ces deux fonctionnalités sont généralement celles que l'on retrouve sur la station. A cela s'ajoute un certain nombre de logiciels de visualisation de l'état de la connexion sans fil, de la qualité du signal, du bruit… Les équipements d'interconnexion sans fil possèdent également des mécanismes d'administration (via SNMP), avec des possibilités de filtrage de protocoles.

2.2.2 Le protocole 802.11

Le protocole 802.11 offre les mêmes fonctionnalités que la technologie WaveLAN . Il a toutefois l'avantage d'être ratifié au sein de l'IEEE. Ce qui lui permet théoriquement d'offrir une compatibilité entre les équipements des différents constructeurs.

L'utilisation de ces technologies dans un réseau local équipé de routeurs, pose un certain nombre de problèmes. Le plus important et le plus incontournable est l'acheminement du trafic IP. En effet, le réseau est équipé de point d'accès sans fil, eux-mêmes séparés par des routeurs. Les technologies de type WaveLAN ou 802.11 (qui se situent au niveau 2 de la couche OSI) ne permettent pas de gérer la mobilité des équipements au niveau IP. Il est alors nécessaire d'utiliser conjointement à ces techniques de transmission sans fil, un protocole additionnel de niveau IP [Thomas 2006].

2.3 Terminologie et architecture

Dans cette partie ont décrits les principaux termes utilisés dans Mobile IP. Et l'architecture utilisée pour le bon fonctionnement [Nicolas 2001].

2.3.1 Modèle OSI

L'Internet est le plus grand réseau existant à l'heure actuelle. Il rassemble plusieurs millions d'ordinateurs et d'utilisateurs. Pour que ces machines puissent coopérer, il a été nécessaire de définir un certain nombre de standards. L'architecture des machines de l'Internet repose sur une approche en couches, similaire au modèle OSI (Interconnexion des Systèmes Ouverts) en 7 couches. Ces sept couches sont décrites dans la Figure 2.

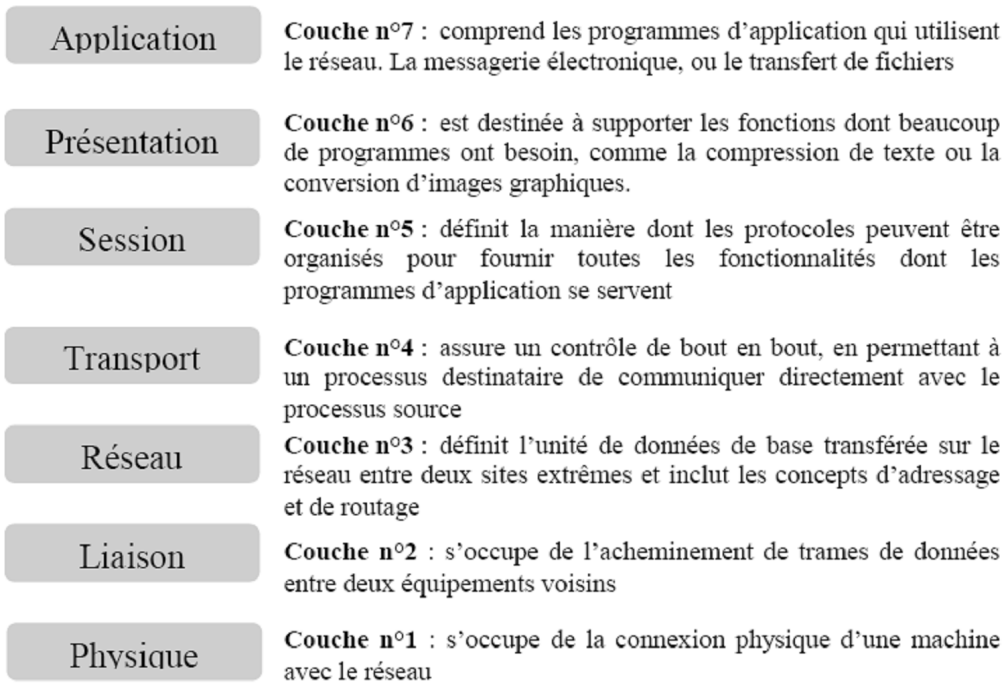


Figure 2. : Modèle des couches OSI

La couche liaison gère l'accès au médium physique et certaines informations peuvent être très utiles dans la gestion de la mobilité.

La couche réseau est actuellement implémentée par le protocole IP dans l'Internet. Donc il est nécessaire de proposer des solutions aux impacts causés par la mobilité sur le fonctionnement de ce protocole.

2.3.2 Nouveaux équipements

Tout d'abord, on distingue deux types de réseaux selon la position du nœud mobile.

On appellera réseau mère ou réseau principal le réseau auquel est rattaché le nœud mobile administrativement. C'est le réseau dans lequel il est déclaré dans le DNS et sur lequel il obtient une adresse IP principale. D'un autre côté, on appelle réseau visité ou réseau étranger un réseau où le nœud mobile se trouve à un moment donné lors de ses déplacements.

Dans Mobile IP, quatre nouveaux acteurs sont définis [Nicolas 2001] :

- **Nœud mobile :** équipement IP qui est capable de se déplacer sur Internet et implémentant le protocole Mobile IP
- **Agent mère :** routeur d'accès avec une interface sur le même lien que le nœud mobile (dans le réseau mère du mobile)
- **Agent visité :** routeur d'accès avec une interface sur le lien courant du nœud mobile (dans un réseau visité). Cet agent n'existe que dans Mobile IPv4.
- **Point d'accès :** équipement intermédiaire entre le réseau filaire et le nœud mobile qui offre la connexion aux nœud mobiles qui lui sont rattachés L'architecture de base des ces équipements est présentée dans la Figure 3

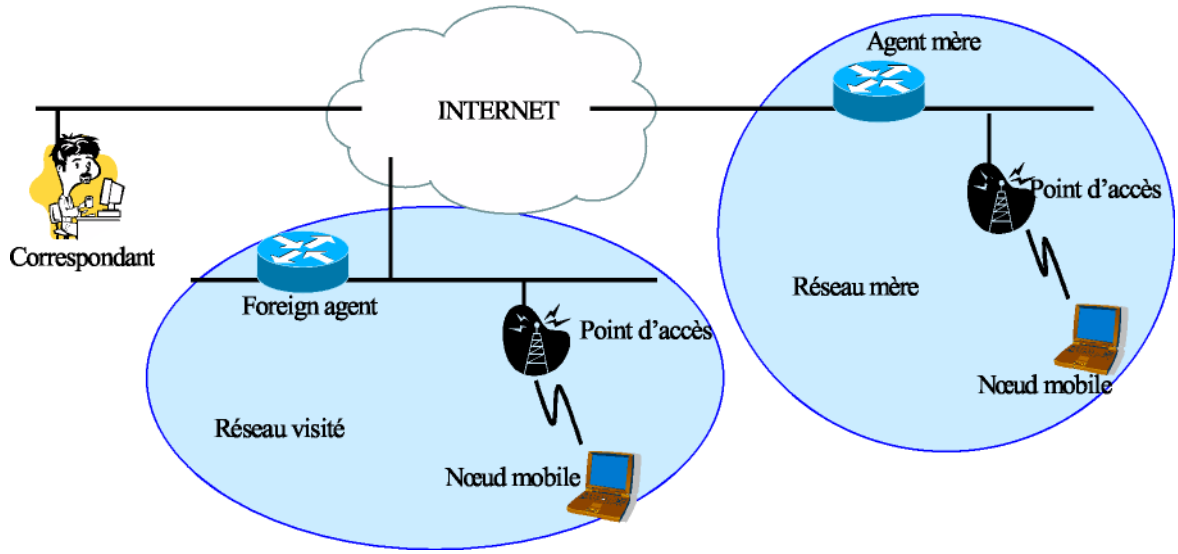


Figure 3. : Architecture de base

Les agents de mobilité (agent mère et agent visité) maintiennent une liste des nœuds mobiles qu'ils gèrent. Cette liste est appelée cache d'association ; elle associe l'adresse principale du mobile à son adresse temporaire. Le rôle principal de ces agents de mobilité est d'encapsuler (resp. décapsuler) les paquets en transit entre les correspondants et les nœud mobiles en ajoutant (resp. en enlevant) un en-tête d'adressage (voir le détail de l'encapsulation et de la décapsulation dans les parties MIPv4 et MIPv6). Dans MIPv6, les correspondants d'un nœud mobile détiennent aussi un cache d'association ce qui leur permet de connaître l'adresse temporaire du mobile associée à son adresse principale. De plus amples détails seront donnés dans la section suivante.

2.3.3 Adressage

Un nœud mobile [Nicolas 2001] peut avoir plusieurs adresses simultanément. L'adresse qu'un nœud mobile acquiert lors de sa première connexion dans son réseau mère est dite adresse principale. C'est l'adresse qui sera toujours utilisée par le mobile et ses correspondants pour identifier les communications au niveau applicatif. Effectivement, le protocole TCP utilise les adresses source et destination pour identifier une communication. Si l'on précisait à tous les correspondants la nouvelle adresse temporaire du nœud mobile à chaque déplacement, la communication nécessiterait d'être réinitialisée à chaque fois. La communication serait donc rompue à chaque déplacement du nœud mobile ce qui est

complètement inefficace. C'est pourquoi c'est cette adresse principale que les correspondants utiliseront comme adresse de destination, quelque soit la position du nœud mobile.

En plus, le nœud mobile peut détenir de(s) adresse(s) temporaire(s), dite(s) adresse temporaire. Cette adresse est obtenue par le nœud mobile à chaque entrée dans un réseau visité. Le nœud mobile devra indiquer cette adresse à son agent mère (et éventuellement à ses correspondants dans MIPv6) périodiquement pour qu'il puisse maintenir une correspondance entre adresse principale et adresse temporaire.

2.4 La mobilité de l'utilisateur

La mobilité de l'utilisateur [Chaouchi 2006] concerne la mobilité du terminal et la mobilité personnelle. La mobilité du terminal permet à un terminal mobile de changer son point d'attachement au réseau sans perdre la connexion en cours. Les réseaux IP supportent la mobilité du terminal en utilisant le protocole Mobile IP.

La mobilité personnelle permet à un utilisateur d'utiliser n'importe quel terminal mobile ou fixe disponible et de n'importe quel réseau pour accéder à ses services personnels souscrits dans son réseau d'origine.

La mobilité personnelle est liée à la gestion de la localisation de l'utilisateur et à la gestion de la portabilité des services. Un identificateur personnel universel est nécessaire pour réaliser la mobilité personnelle.

La troisième génération des réseaux mobiles (3G) travaille sur la définition d'un identificateur d'utilisateur qui peut être utilisé sur les terminaux mobiles ou fixes pour ainsi permettre la convergence fixe et mobile des services. Le concept VHE (Virtual Home Environnement) est aussi introduit par la génération 3G pour supporter la gestion de la portabilité des services définis par la mobilité personnelle [Chaouchi 2006].

2.4.1 Macro-mobilité

On peut voir également *la macromobilité* comme concernant un déplacement entre deux sites différents. Le protocole Mobile IP c'est l'amélioration de protocole IP pour la prise en charge de la mobilité des terminaux entre les différents domaines. Mobile IPv4 [Khouaja 2006] définit trois entités fonctionnelles pour la gestion de la mobilité : le nœud mobile et deux agents de mobilité, l'agent mère et l'agent étranger. Le nœud mobile est configuré avec une adresse IP permanente, appelée adresse mère et appartenant à son réseau mère. L'agent mère est un routeur spécifique du réseau mère, il enregistre la localisation du mobile lorsque celui-ci est en visite dans un réseau externe. L'agent étranger est un routeur du réseau externe (réseau "visité") dans lequel le nœud mobile est localisé à un moment donné. Il est généralement utilisé par le nœud mobile pour obtenir l'adresse IP temporaire correspondant à sa nouvelle localisation, adresse qu'il enregistre auprès de son agent mère.

Le fonctionnement du protocole Mobile IP est le suivant : le nœud mobile écoute en continu les annonces d'agents diffusées par les agents de mobilité ; lorsqu'il s'aperçoit que le préfixe réseau diffusé dans ces annonces change, il en déduit qu'il a changé de réseau ; il cherche alors à acquérir une nouvelle adresse temporaire. Cette nouvelle adresse peut être celle de l'agent étranger ou être obtenue par un mécanisme d'auto configuration tel que DHCP.

Dans le premier cas, le nœud mobile s'enregistre auprès de son agent étranger qui l'enregistre ensuite auprès de son agent mère.

Dans le deuxième cas, le nœud mobile annonce sa nouvelle adresse temporaire directement à son agent mère en échangeant des requêtes et réponses d'enregistrement. Une fois l'enregistrement terminé, l'agent mère intercepte les paquets envoyés à l'adresse mère du

nœud mobile et les encapsule pour les rediriger via un "tunnel IP" vers la nouvelle adresse temporaire. Lorsque cette adresse temporaire est celle de l'agent étranger, celui-ci doit décapsuler les paquets du mobile avant de les lui livrer. Lorsque l'adresse temporaire est celle obtenue par auto configuration du nœud mobile, il décapsule lui-même les paquets.

Afin de maintenir son enregistrement valide auprès de ses agents de mobilité, le nœud mobile doit périodiquement le renouveler. Lorsqu'il retourne dans son réseau mère, il annule l'enregistrement en cours et indique ainsi à son agent mère de ne plus intercepter les paquets qui lui sont destinés. En outre, dans Mobile IP, le nœud mobile doit utiliser son adresse mère comme adresse source des paquets émis afin d'assurer la continuité de ses connexions. Cependant, dans des réseaux utilisant des routeurs filtrants et pour des raisons de sécurité, on impose que l'adresse source d'un paquet émis appartienne au réseau. L'extension Tunnel inverse propose de résoudre ce problème en livrant les paquets IP émis via un tunnel entre l'adresse temporaire du nœud mobile et l'agent mère ; les paquets sont ensuite décapsulés par l'agent mère et livrés aux nœuds correspondants en utilisant l'adresse mère du nœud mobile comme adresse source.

Enfin, la version de base de Mobile IPv4 impose que les paquets adressés aux nœuds mobiles soient routés à travers l'agent mère. Ils suivent donc des chemins qui peuvent être assez longs et non optimaux. L'extension nommée Optimisation de routage [Khouaja 2006] remédie à ce problème en permettant à un nœud correspondant de maintenir un cache d'associations qui pointe continuellement sur l'adresse temporaire courante du nœud mobile. Il peut alors router directement les paquets sans passer par l'agent mère. Ces caches d'associations sont créés et mis à jour par des nouveaux types de messages.

2.4.2 Micro-mobilité :

Pour bien définir le problème, Mobile IP permet de gérer la mobilité mais en passant par l'agent home et que cette technique ne s'applique qu'à des déplacements lents ou à des changements de lieu de l'utilisateur, et non du terminal. Un utilisateur mobile n'a pas de terminal à lui ou du moins ne se déplace pas avec son terminal mais se reconnecte sur un terminal qu'il trouve dans le lieu visité. On parle en ce cas de *micromobilité*. Dans les applications avec terminal mobile, le terminal suit l'utilisateur, et l'application se déroule toujours sur le même terminal. On parle alors de *macromobilité.*

La micromobilité, elle, concerne le déplacement entre deux points d'attachements appartenant à un même domaine. *La macromobilité* comme concernant un déplacement entre deux domaines différents.

Dans l'environnement de l'Internet mobile, où les cellules peuvent devenir minuscules, il n'est plus concevable qu'à chaque changement de cellule l'agent visité avertisse l'agent home au risque de surcharger démesurément le réseau. Il faut donc cacher à l'agent home que le terminal se déplace. Les études de performance concernant IP Mobile dans le cas des déplacements entre deux stations de base d'un même domaine ont révélé son incapacité à supporter ce type de mobilité. Pour cela, des travaux ont donné naissance à d'autres approches de gestion de *micromobilité* qui seront présentées dans le chapitre 3.

2.5 Handoff (ou Handover)

Dans cette partie sera décrit la terminologie liée au handoff pour expliciter au mieux ce que chaque protocole essaie de résoudre. Les termes employés sont ceux utilisés par le groupe de travail de l'IETF [Nicolas 2001].

Le handoff est le processus enclenché quand un mobile actif (en cours de communication) change son point d'attache à l'Internet. On peut découper un handoff de la manière suivante : handoff de niveau 3 (couche IP) et handoff de niveau 2 (couche liaison),

d'après le modèle OSI. Le handoff de la couche 2 est l'opération effectuée par un nœud mobile qui change de point d'accès sans fil, c'est-à-dire que c'est le passage d'un point d'accès à un autre. Ce handoff peut engendrer ou non un handoff de la couche supérieure selon le lien filaire des points d'accès (si elles sont sur le même lien réseau ou non). Plus généralement, on distingue trois types de handoff :

- Handoff intra-routeur d'accès : handoff généré par le changement d'interface réseau du routeur d'accès par laquelle il communique avec le mobile. L'adresse IP du mobile ne change pas.
- Handoff à l'intérieur d'un réseau d'accès : Opération effectuée quand le nœud mobile change de routeur d'accès en restant dans le même réseau d'accès. Ce handoff est invisible pour un point extérieur au sous-réseau et l'adresse du mobile ne change toujours pas, mais le chemin pour l'atteindre est modifié.
- Handoff entre réseaux d'accès : Déplacement du mobile hors du réseau d'accès ; cette fois le nœud mobile a besoin d'acquérir une nouvelle adresse IP.

Les travaux de recherche concerne surtout l'amélioration du dernier type de handoff, c'est-à-dire celui qui nécessite des messages d'enregistrement et qui peut provoquer une rupture de communication. On verra par la suite que l'utilisation des informations de niveau 2 (couche liaison du modèle OSI) peut s'avérer fort utile pour anticiper un handoff de niveau 3 (couche IP du modèle OSI). Ces trois types de handoff sont illustrés dans la Figure 4 ci-dessous [Nicolas 2001].

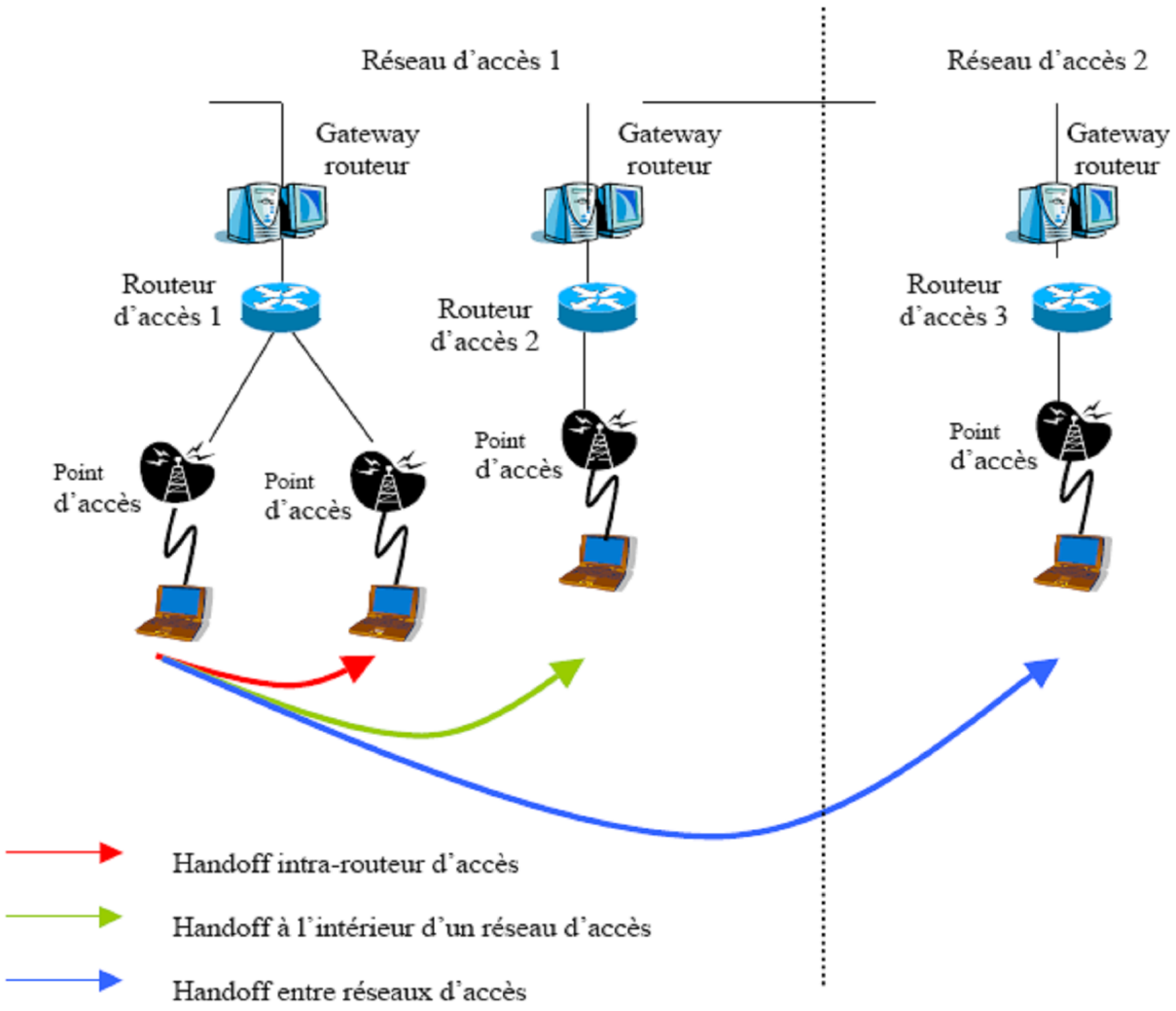


Figure 4. : Niveau du Handoff

On parlera par la suite de la latence du handoff pour exprimer les performances sur celui-ci. La latence du handoff est le laps de temps entre le dernier moment où le mobile peut recevoir et émettre des paquets IP à travers l'ancien routeur d'accès et le premier moment où il peut recevoir et émettre des paquets à travers le nouveau routeur d'accès.

C'est donc le temps pendant lequel un nœud mobile ne peut ni recevoir, ni émettre un trafic IP. Par ailleurs, le processus de handoff peut être amélioré soit en réduisant le nombre de paquets perdus, soit en diminuant la charge de la signalisation, soit encore en rendant le processus le plus rapide possible. On parlera alors de :

- Smooth handoff : handoff qui a pour but principal de minimiser la perte de paquets, sans condition sur le délai de forwarding des paquets.
- Fast handoff : handoff qui a pour but principal de minimiser les délais, sans conditions sur le nombre de paquets perdus.
- Seamless handoff : la définition absolue est un handoff où il n'y a pas de changement dans la capacité, la sécurité ou la qualité du service. En pratique, une dégradation est tout de même observée ; La définition pratique est que les applications ou les utilisateurs ne remarquent pas ces dégradations.

De plus en plus on parle de micro mobilité dans le domaine de la recherche sur la mobilité des hôtes. La micro mobilité est la gestion fine de la mobilité, souvent dans un domaine limité en taille. Des protocoles distincts peuvent être utilisés pour gérer la micro mobilité et la mobilité globale (entre aires ou domaines). Un protocole de micro mobilité raffine la gestion de la mobilité en fonction des besoins spécifiques dans l'aire ou le domaine en question.

Par la suite, on sera encore amener à définir quelques termes spécifiques à chaque protocole. Dans un souci de clarté, dans le cas de MIPv4, on appelle ancien agent visité (AV) l'agent visité auquel le nœud mobile est attaché et qu'il est sur le point de quitter pour le nouveau AV. Dans le cas de MIPv6, on parle d'ancien routeur d'accès (RA) et de nouveau RA [Nicolas 2001].

2.5.1 Fast handoff

Le support du fast handoff [Tsirtsis 2001] [Nicolas 2001], qui réduit le retard et la perte de paquet pendant un handoff, est un apport important aux protocoles de micro-mobilité. Un certain nombre de choix dans la structure des protocoles influencent les performances du handoff, comme le contrôle du handoff, le buffering et les techniques de forwarding, le comportement radio, la détection et la prédiction de mouvement et l'accouplement et la synchronisation entre les couches IP et radio. L'objectif sous-jacent à l'introduction du fast handoff est d'essayer de combiner tous ces choix dans un même protocole.

La solution de *fast handoff* est la minimisation de la latence du handoff. Le principe est d'établir une nouvelle adresse temporaire avant de rompre la liaison du nœud mobile avec son ancien AV/RA. Ensuite, quand le mobile se rattache au nouvel agent de mobilité, il peut continuer ses communications avec sa nouvelle adresse déjà déterminée. Si l'enregistrement anticipé échoue, le mobile peut toujours réaliser une opération de handoff « traditionnel ». On verra de plus que le *fast handoff* met en place un système de *forwarding* des paquets entre l'ancien et le nouveau routeur d'accès.

L'établissement de la nouvelle adresse temporaire avant que le nœud mobile ne se déplace implique une anticipation sur le mouvement du mobile. Cette anticipation peut être faite à partir des messages échangés au niveau physique ou simplement par la remontée d'informations pertinentes de niveau 2 (mesure d'intensité du signal…). L'objectif est de réaliser le handoff de niveau 3 avant que celui ne la couche 2 n'ait fini.

Le fait de faire interagir plus fortement les deux couches peut réduire au minimum la latence du handoff mais peut avoir un effet néfaste sur l'applicabilité générale de la solution. Comme nous allons le voir, beaucoup de propositions exposées par les groupes de travail Mobile IP et *Seamoby* de l'IETF discutent de *seamless handoff* où les données sont échangées entre les anciens et nouveaux points d'accès pendant le *handoff*. La plupart de ces approches utilisent une signalisation assez complexe, du *buffering* et des procédures de synchronisation. La détection de mouvement au niveau 3 joue un rôle important dans les performances du *handoff*. Le retard induit par la reconnaissance et l'enregistrement à un nouveau point d'accès peut avoir un impact significatif sur la livraison des données et la mobilité. Le schéma du *handoff* basé sur la mesure de l'intensité du signal peut fournir de meilleures solutions ; C'est le cas lorsque le *handoff* de la couche 3 est déclenché par un événement de la couche 2. Cependant, étant donné la grande diversité des équipements sans fil, il est difficile de définir les opérations et les interactions de ces protocoles radio dans une mobilité globale, sans tomber dans des définitions spécifiques. Il est donc nécessaire de définir une API radio qui capture l'essence de chaque technologie sans fil sans exposer des détails spécifiques. Cette API faciliterait la couche 2 « a déclenché » le *handoff* indépendamment de la technologie radio.

Dans un souci d'efficacité, l'anticipation peut concerner plus d'un agent de mobilité. On peut même se retrouver dans des cas où on a un chaînage de AV/RA lorsque le nœud mobile se déplace rapidement.

Le nouveau routeur d'accès (agent visité dans le cas de MIPv4) doit stocker les hôtes voisins qui sont capables de venir dans son sous-réseau pour défendre les adresses allouées. Néanmoins, ce type d'entrée a besoin d'être gardé moins longtemps qu'une entée pour un hôte appartenant réellement au sous-réseau. C'est pourquoi le nouveau routeur d'accès utilise un cache des voisins pour stocker les nœuds mobiles susceptibles d'entrer dans le sous-réseau.

Le protocole de *fast handoff* fonctionne dans les cas du handoff initialisé par le mobile et dans le cas du *handoff* initialisé par le réseau, la différence étant dans l'ordre des messages. Cinq nouveaux messages ont été définis en plus de ceux de MIP, dont trois entre routeur d'accès et nœud mobile :

- *Router Solicitation For Proxy* : envoyé par le nœud mobile pour demander à son ancien AV/RA un *Proxy Router Advertisement*. Le nœud mobile doit indiquer sa destination (par exemple en donnant l'adresse physique du nouveau point d'attache).
- *Proxy Router Advertisement* : envoyé par l'ancien AV/RA à un nœud mobile pour l'informer sur son nouveau sous-réseau potentiel ; L'ancien AV/RA peut indiquer au mobile qu'il ne connaît pas le sous-réseau potentiel, que le nouveau point d'attachement appartient en fait au même sous-réseau ou avec une adresse ou un préfixe réseau. Ce message peut être non sollicité (cas du handoff contrôlé par le réseau) ou en réponse à un *Router Solicitation For Proxy*.
- *Neighbor Advertisement* : envoyé par le nœud mobile pour informer le nouveau AV/RA qu'il est arrivé dans le nouveau sous-réseau.

Les deux autres messages du protocole sont échangés entre les agents de mobilité :

- *Handover Initiate* : envoyé par l'ancien routeur d'accès au nouveau pour demander une adresse temporaire ou pour en valider une.
- *Handover Acknowledgement* : envoyé par le nouveau AV/RA à l'ancien en réponse à un *Handover Initiate* pour valider ou rejeter une adresse.

2.5.2 Scénario

On considère deux agents de mobilité, un ancien auquel le nœud mobile est attaché et un nouveau vers lequel le mobile se déplace. Ces agents de mobilité sont des agents visités (AV) dans le cas de MIPv4 et des routeurs d'accès (RA) dans le cas de MIPv6. On admet que le nouveau AV/RA a été découvert par anticipation. Dans ce qui suit, on considère uniquement le cas où le nouveau AV/RA est connu par l'ancien, où l'attachement du nœud mobile au nouveau sous-réseau est accepté par le nouveau AV/RA et où l'adresse temporaire proposée est valide. Dans le cas contraire, le *fast handoff* échoue et le nœud mobile peut réaliser un handoff comme décrit dans MIP. De plus, on admettra que les routeurs d'accès utilisent l'auto-configuration d'adresse sans état, comme c'est la plupart du temps le cas aujourd'hui (le cas d'auto-configuration d'adresse avec état sera développé juste après).

Dans le cas du *handoff* contrôlé par le mobile, le nœud mobile envoie un *Router Solicitation For Proxy* (étape 1a dans la Figure 5) quand il détecte qu'un *handoff* va avoir lieu. Comme dit précédemment, le nœud mobile inclus des informations permettant l'identification du nouveau AV/RA. L'ancien AV/RA répond avec un *Proxy Router Advertisement* (étape 1b) qui contient une nouvelle adresse temporaire. Dans le même temps, l'ancien AV/RA envoie un *Handover Initiate* au nouveau AV/RA avec cette nouvelle adresse pour validation (étape 1b).

Dans le cas du *handoff* contrôlé par le réseau, une entité spécifique du réseau décide quand le mobile a besoin de se rattacher à un nouveau point d'accès. Quand le mobile semble se déplacer, l'ancien AV/RA envoie à la fois un *Proxy Router Advertisement Unsolicited* (étape 1b) avec une nouvelle adresse pour le mobile et un *Handover Initiate* (étape 1b) au nouveau AV/RA.

Puis, le nouveau AV/RA confirme la validité de la nouvelle adresse dans un *Handover Acknowledgement* (étape 2) à destination de l'ancien AV/RA.

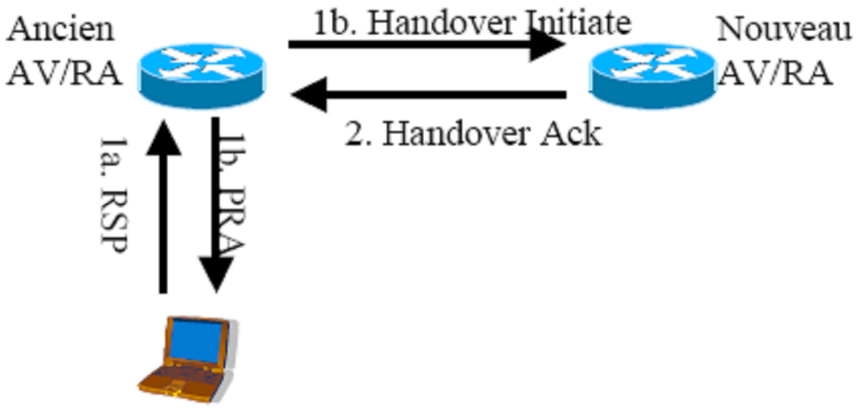


Figure 5. : Initialisation du Handoff

Ensuite, le nœud mobile doit enregistrer sa nouvelle adresse avec son agent mère (et éventuellement avec ses correspondants dans MIPv6). Cet enregistrement diffère quelque peu entre les versions 4 et 6 de MIP. Dans MIPv4, le nœud mobile envoie un *Enregistrement Request* (étape 3 dans la Figure 6) à son agent mère à travers l'ancien agent visité ou le nouveau suivant sa localisation. Le *Enregistrement Reply* (étape 4) est envoyé au nouveau agent visité qui le transmet à la nouvelle adresse temporaire du mobile et à l'ancien agent visité [Nicolas 2001].

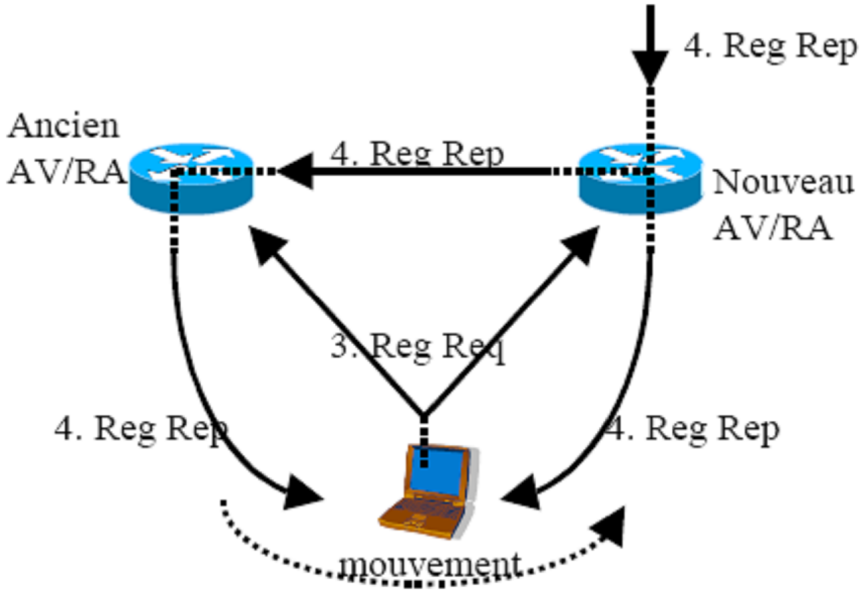


Figure 6. : Enregistrement MIP

Dans le cas de MIPv6, le nœud mobile envoie un *Binding Update* (étape 3 dans la Figure 7) [Nicolas 2001] à l'ancien routeur d'accès juste avant son déplacement pour lui indiquer son mouvement. Ce message déclenche le *forwarding* des paquets entre les routeurs d'accès, paquets que le nouveau routeur d'accès met en buffer. Le *Binding Acknowledgement* (étape 4) est envoyé par l'ancien routeur d'accès à l'ancienne adresse temporaire du mobile et au nouveau routeur d'accès, qui le fait suivre à la nouvelle adresse temporaire du mobile. Si le mobile reçoit cet acquittement, après son déplacement, il doit envoyer un *Binding Update* (étape 5) à son agent mère et ses correspondants à travers le nouveau routeur d'accès. S'il ne reçoit pas l'acquittement, il doit renvoyer un *Binding Update* à l'ancien routeur d'accès et un autre à son agent mère et ses correspondants.

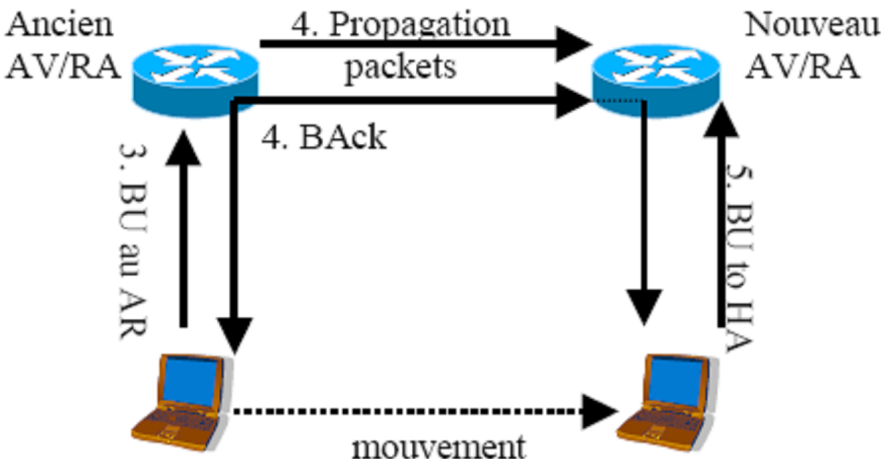


Figure 7. : Enregistrement MIPv6

Finalement, le nœud mobile envoie un *Neighbor Advertise* au nouveau routeur d'accès pour lui indiquer son arrivée. C'est alors que le nouveau routeur d'accès transmet les paquets mis en buffer au mobile.

Dans le cas de configuration d'adresse à état, l'ancien routeur d'accès doit envoyer un *Handover Initiate* avant le *Proxy Router Advertisement*. Le message *Handover Initiate* est utilisé pour demander une nouvelle adresse temporaire pour le mobile et non pour en valider une. Le message *Handover Acknowledgement* contient une adresse valide qui peut être transmise au mobile dans le *Proxy Router Advertisement*.

Si la procédure de *fast handoff* décrite ci-dessus termine sans erreur, l'établissement du service au nouveau point d'accès est plus rapide. Un transfert de contexte entre les deux routeurs d'accès, en utilisant le tunnel crée, pourrait encore améliorer la rapidité de l'établissement de service au nouveau point d'attachement. Par contexte on entend les caractéristiques utilisées par le nœud mobile pour sa communication, comme la compression d'en tête ou des informations de sécurité qu'il partagera avec l'ancien routeur.

Bien que ce protocole permette de rendre le *handoff* plus rapide, des paquets peuvent être perdus. Pour résoudre ce problème, on peut utiliser le bi casting vers les deux AV/RA comme décrit dans la section suivante [Nicolas 2001].

2.5.3 Bi casting

Le bi casting [Nicolas 2001] est la duplication du même trafic destiné au mobile à son ancienne et à sa nouvelle localisation. Cette méthode est utilisée pour réduire le nombre de paquets perdus pendant un *handoff* pour obtenir un *smooth handoff*. On parle alors de *handoff* proactif. Cette solution peut être utilisé en plus du protocole de *fast handoff* décrit ci-dessus pour envoyer des copies du trafic aux potentielles localisations du nœud mobile. Avec certes une augmentation de la charge du réseau, cette solution tend à procurer un *seamless handoff*. De plus, le bi casting est une bonne solution contre l'effet « ping-pong » ; quand un nœud mobile se déplace entre deux agents de mobilité plusieurs fois et fréquemment, MIP nécessite que le nœud mobile crée une nouvelle adresse temporaire et l'enregistre à chaque attachement. Le Bi casting permet au nœud mobile d'être enregistré avec les deux AV/RA simultanément.

Pour réaliser le bi casting, le nœud mobile doit être enregistré avec plus d'un agent de mobilité. Il doit donc être capable de gérer plusieurs adresses temporaires. D'un autre côté, les agents de mobilité doivent être capable de gérer des associations multiples. Pour demander le bi casting, le nœud mobile charge un bit spécifique dans son message d'enregistrement indiquant sa demande. Quand un agent de mobilité reçoit ce type de message, il ajoute une entrée pour le mobile sans en enlever.

Aucun nouveau message n'est nécessaire puisque toutes les requêtes et réponses liées au bi casting peuvent être mis en *piggy-backing* dans les messages déjà utilisé dans MIP. Si le nœud mobile sollicite le bi casting auprès d'un AV/RA qui ne supporte pas d'association simultanée, le AV/RA ignore l'option et le nœud mobile réalise un handoff comme spécifié dans MIP.

Dans cette section, on décrit le bi casting à partir de deux agents de mobilité différents ; tout d'abord on traitera le bi casting généré par l'agent mère. Cette solution est plus un cas d'école pour introduire les mécanismes du bi casting qu'une réelle opportunité pour un nœud mobile puisqu'elle présente des problèmes de mise à l'échelle. Ensuite on verra le bi casting effectué grâce à un tunnel entre les deux AV/RA. On considèrera dans tous les scénarios qui vont suivre que le bi casting est réalisé en complément du fast handoff explicité dans la section précédente.

2.5.3.1 Bi casting par l'agent mère

Après avoir procédé aux messages du *fast handoff* (*Router Solicitation For Proxy*, *Proxy Router Advertisement*, *Handover Initiate*, *Handover Acknowledgement*), le nœud mobile a juste à s'enregistrer en chargeant le bit indiquant l'association simultanée dans son message. Dans le cas de MIPv4, le nœud mobile envoie un *Enregistrement Request* (avec le bit chargé) à son agent mère. Si l'agent mère accepte la requête, il enregistre deux associations pour ce mobile et enverra désormais l'ensemble du trafic aux deux adresses temporaires du mobile. Les deux agents visités propagent le trafic dans leur sous-réseau respectif bien que le nœud mobile n'en écoute qu'un (voir Figure 8) [Nicolas 2001].

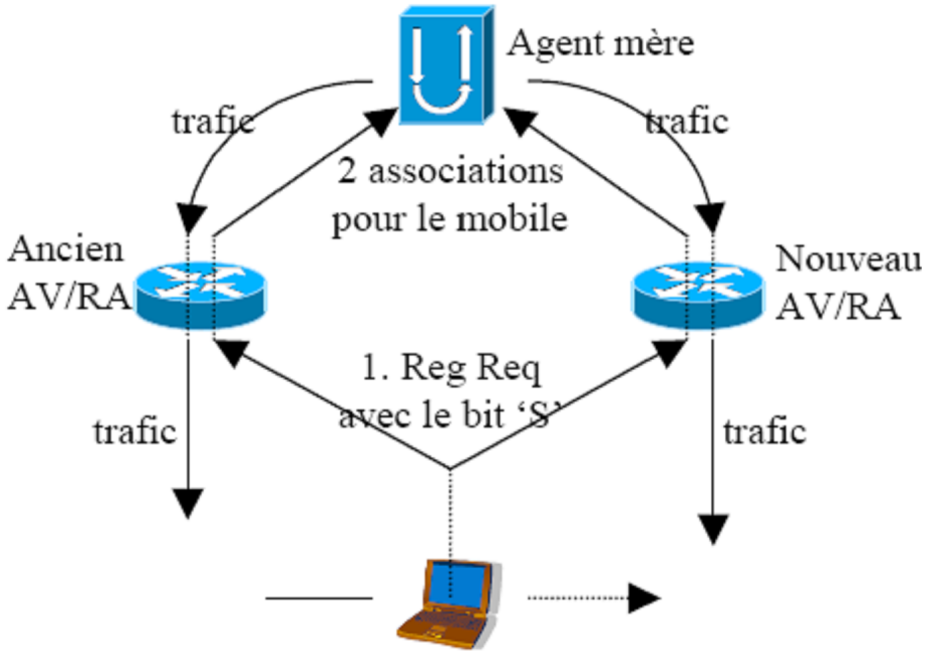


Figure 8. : Bi casting par l'Agent mère

Bien que le bi casting semble être efficace et réduise le nombre de paquets perdus, cette manière de le mettre en nœud n'est pas réalisable ; effectivement, l'ensemble du trafic pour le nœud mobile est dupliqué sur tout le chemin allant de l'agent mère au mobile alors que les paquets prennent exactement le même chemin, excepté quelques derniers sauts. Pour résoudre ce problème, on peut penser à envoyer les paquets en multicast. Cependant, la gestion de groupe doit être légère, dynamique et l'ajout ou le retrait d'une adresse destinatrice doit se faire rapidement. La technique du Small Group Multicast (SGM) répond parfaitement à ces besoins ; SGM est basé sur le multicast explicite : les datagrammes multicast sont routés d'après les informations de routage unicast et chaque datagramme contient la liste des adresses destination. A chaque saut, le routeur identifie si pour chaque destination le datagramme peut prendre la même direction. Si une adresse indique que le datagramme doit prendre une autre direction que les autres, le routeur duplique le paquet. Sinon il forward simplement le paquet sans aucune modification (voir Figure 9) [Nicolas 2001].

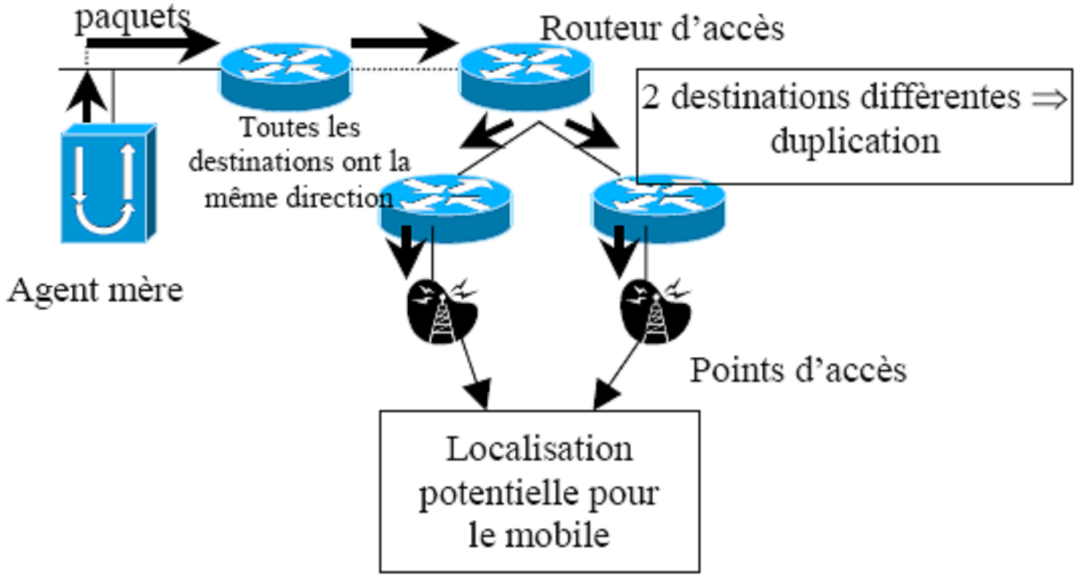


Figure 9. : Explicite Multicast pour le Bi casting

D'autres méthodes pour réduire la charge du réseau tout en faisant du bi casting existent. L'une d'entre elles est de réaliser le bi casting à partir d'un autre point du réseau. Le bi casting à partir de l'ancien routeur d'accès est décrite dans la sous-section suivante.

2.5.3.2 Bi casting réalisé par l'utilisation d'un tunnel

Le tunnel mis en place entre l'ancien et le nouveau AV/RA par la procédure de *fast handoff* peut être utilisé pour faire du bi casting. Dans le cas de MIPv6, tant que l'entrée pour le nœud mobile dans l'ancien routeur d'accès n'a pas expirée ou tant que l'agent mère du mobile n'a pas reçu de *Binding Update* avec la nouvelle adresse temporaire, le trafic du nœud mobile arrive à l'ancien routeur d'accès. Une fois que le mobile envoie un *Binding Update* avec le bit 'B' (indiquant la demande de bi casting) chargé à son ancien routeur d'accès, ce dernier peut commencer le bi casting : l'ancien routeur d'accès forward les paquets à l'ancienne et à la nouvelle adresse temporaire [Nicolas 2001].

Comme on vient de le voir, le *fast handoff* associé au bi casting tend à obtenir un *seamless handoff*. Cependant, de nombreux changements doivent être fait dans les équipements (agent mère, routeurs d'accès…). Finalement, la méthode de bi casting peut encore être enrichi dans un modèle hiérarchique. Le modèle hiérarchique, ainsi que la réalisation du bi casting dans MIP hiérarchique sont présentés dans la section suivante [Nicolas 2001].

Chapitre 3 :

Les protocoles de mobilité : étude comparative

Chapitre 3 : Les protocoles de mobilité : Etude comparative

3.1 Le protocole Mobile IP

L'IETF étudie actuellement un protocole nommé Mobile IP. Parmi les fonctionnalités offertes par ce protocole, je citerai le maintient des communications existantes entre le noeud mobile et ses correspondants, même pendant les déplacements du mobile. Le routage le plus direct possible entre le mobile et ses correspondants ainsi que le support de l'acheminement des paquets multipoint entre le(s) mobile(s) et le reste des participants à une communication de groupe. Pour remplir l'ensemble de ces fonctions l'IETF a identifié quatre acteurs :

- Le mobile lui-même.
- Les correspondants de ce dernier.
- Un routeur situé dans le réseau administratif du mobile appelé, Agent mère.
- Un routeur situé dans le réseau visité par le mobile appelé, Agent relais. Ce dernier est utilisé uniquement dans le cas de la mobilité IPv4.

Quand un mobile se déplace, à l'aide des techniques de transmission sans fil, il est amené à s'attacher à des points d'accès divers situés généralement dans des (sous-) réseau distincts. Cette contrainte entraîne pour le mobile un changement d'adresse IP. En effet, un équipement IP est généralement identifié par une adresse appartenant au réseau sur lequel il se trouve. Ce changement d'adresse entraîne généralement la rupture des communications de niveau transport. A travers le protocole Mobile IP, l'IETF permet de masquer ce changement aux applications utilisées entre le mobile et ses correspondants [Thomas 2006].

3.1.1 Fonctionnalité de protocole Mobile IP

3.1.1.1 Fonctionnalité pour le mobile

Les fonctionnalités définies par l'IETF pour le mobile sont les suivantes [Thomas 2006] :

- Etre capable de prévenir ses correspondants de son changement de position, afin que ces derniers acheminent leurs paquets vers la nouvelle position du mobile.
- Etre capable de prévenir son agent mère, afin que celui-ci achemine les demandes de communications des futurs correspondants du mobile. En effet, les futurs correspondants du mobile ne peuvent pas connaître la position courante du mobile. Ces derniers font initialement l'hypothèse que le mobile se situe dans son réseau administratif.
- Les mécanismes de mise à jour des correspondants et de l'agent mère doivent être sécurisés afin d'éviter des tentatives d'usurpation d'identité.

3.1.1.2 Fonctionnalité pour les correspondants

Les fonctionnalités définies par l'IETF pour les correspondants du (des) mobile(s) sont les suivantes [Thomas 2006] :

- Etre capable d'apprendre les changements de positions des mobiles avec lesquels le correspondant est en communication.
- Etre capable d'acheminer les paquets vers la position courante d'un mobile.
- Etre capable d'authentifier les mises à jour d'un mobile.

3.1.1.3 Fonctionnalité pour les agents mères

Les fonctionnalités définies par l'IETF [Thomas 2006] pour les agents mères sont les suivantes :

- Etre un routeur situé dans le réseau administratif du mobile.
- Etre capable d'apprendre les changements de positions des mobiles qu'il gère.
- Etre capable d'authentifier les mises à jour d'un mobile.
- Etre capable d'intercepter les paquets destinés à un mobile lorsque celui-ci n'est pas dans son réseau mère.
- Etre capable d'acheminer les paquets destinés à un mobile vers sa position courante.

Nous étudions plus spécifiquement le fonctionnement des protocoles MIPv4 et MIPv6.

3.1.2 Mobile IPv4

3.1.2.1 Découverte des agents de mobilité

Une caractéristique propre au mobile est de pouvoir se déplacer en cours d'une communication [Nicolas 2001]. Pour cela, un nœud mobile doit pouvoir détecter ses déplacements, c'est-à-dire détecter le changement de sous-réseau, ce qui nécessite l'obtention d'une nouvelle adresse temporaire. Le protocole de Découverte des Agents met en place un échange de messages permettant cette détection : les agents de mobilité envoient périodiquement des messages annonçant leur disponibilité sur le lien par l'émission de messages *Agent Advertisement* contenant l'information nécessaire pour l'identification du sous-réseau. Cette information peut être le préfixe réseau par exemple. Par ailleurs, un nœud mobile ne désirant pas attendre un tel message peut explicitement en demander un par l'émission d'un *Agent Solicitation* (cas où l'agent tombe en panne par exemple). Ces messages sont authentifiés et sont envoyés en broadcast ou multicast.

3.1.2.2 Enregistrement auprès de l'agent mère

Lorsque le nœud mobile détecte qu'il a changé de sous-réseau (à travers les messages explicités ci-dessus), il doit acquérir une nouvelle adresse temporaire et s'enregistrer auprès de son agent mère et du agent visité du réseau visité. L'acquisition de cette nouvelle adresse se fait grâce au protocole DHCP, Figure 10 [Nicolas 2001].

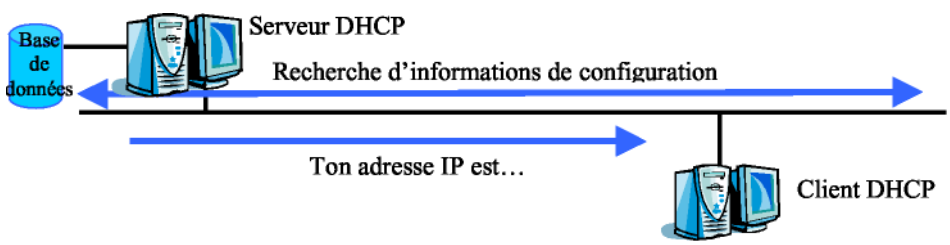


Figure 10. : Configuration DHCP

Une fois que le nœud mobile a une adresse temporaire valide, il émet un message *Registration Request* (étape 1 dans la Figure 11) en indiquant la correspondance entre son adresse principale et son adresse temporaire et éventuellement d'autres options. Ce message passe par le agent visité qui le transmet à l'agent mère du mobile s'il accepte les requêtes du nœud mobile. L'agent mère doit acquitter le *Registration Request* pour bien confirmer la réception (message UDP) et pour informer le nœud mobile de l'acceptation ou du refus de la requête par un *Registration Reply* (étape 2). A réception du *Registration Request*, aussi bien l'agent mère que le agent visité met à jour leur cache d'association pour ce nœud mobile.

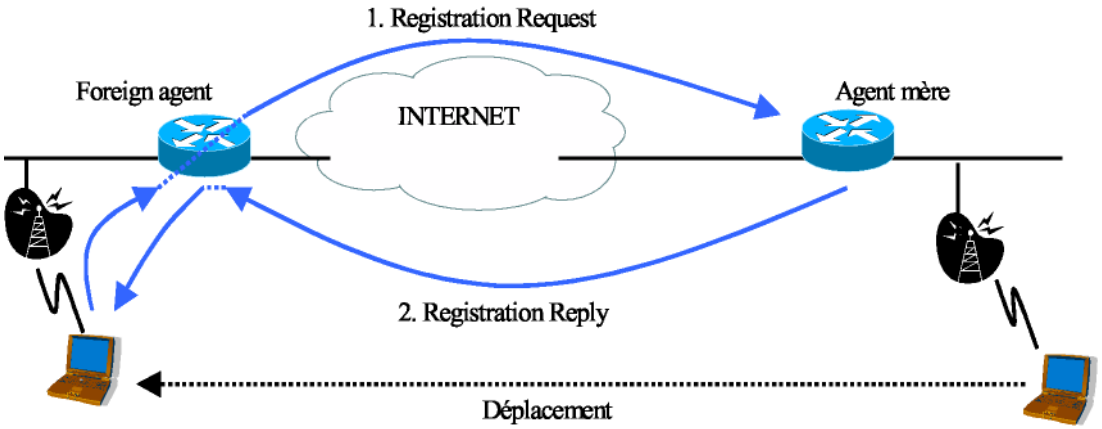


Figure 11. : Enregistrement IPv4

Ensuite, tant que le nœud mobile reste dans le même sous-réseau étranger, il doit uniquement envoyer un Registration Request à intervalle régulier pour éviter que son entrée dans le cache d'association des agents de mobilité n'expire. Par contre, à chaque nouveau déplacement dans un autre sous-réseau étranger, il devra reprendre les mêmes opérations que celles décrites ci-dessus.

Si le nœud mobile retourne dans son sous-réseau mère, il doit se *dés-enregistrer* auprès de son agent mère. Il envoie alors un *De-Registration Request* (étape 1 dans la Figure 12) jusqu'à ce qu'il reçoive un *De-Regitration Reply* (étape 2) qui spécifie que l'agent mère a bien reçu le message et qu'il a supprimé l'entrée pour ce nœud mobile [Nicolas 2001].

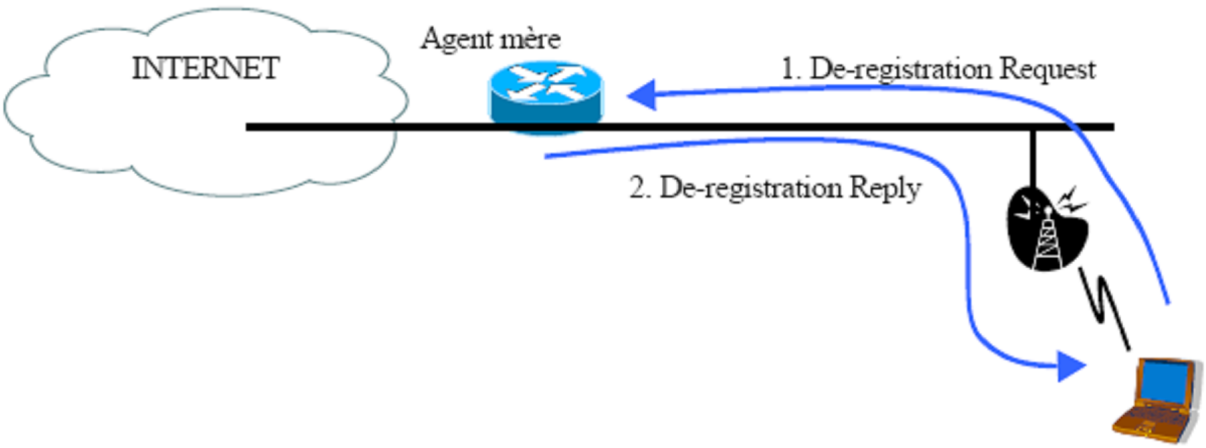


Figure 12. : Dés-enregistrement auprès de l'Agent mère

3.1.2.3 Communication

La communication entre un nœud mobile et un correspondant quelconque sur Internet est très spécifique et requiert plusieurs mécanismes des agents de mobilité. Comme un nœud correspondant d'un nœud mobile ne connaît que l'adresse principale du nœud mobile, les paquets à destination du nœud mobile sont toujours envoyés dans le sous-réseau mère du nœud mobile. Si le nœud mobile ne s'est pas déplacé, les paquets lui seront « livrés » de la même manière qu'un nœud fixe, c'est-à-dire sans opérations supplémentaires. Par contre, si le nœud mobile est dans un sous-réseau visité, son agent mère devra capturer tous les paquets destinés au nœud mobile et les lui transmettre à son adresse temporaire, grâce à son cache d'association (comme illustré Figure 13) [Nicolas 2001].

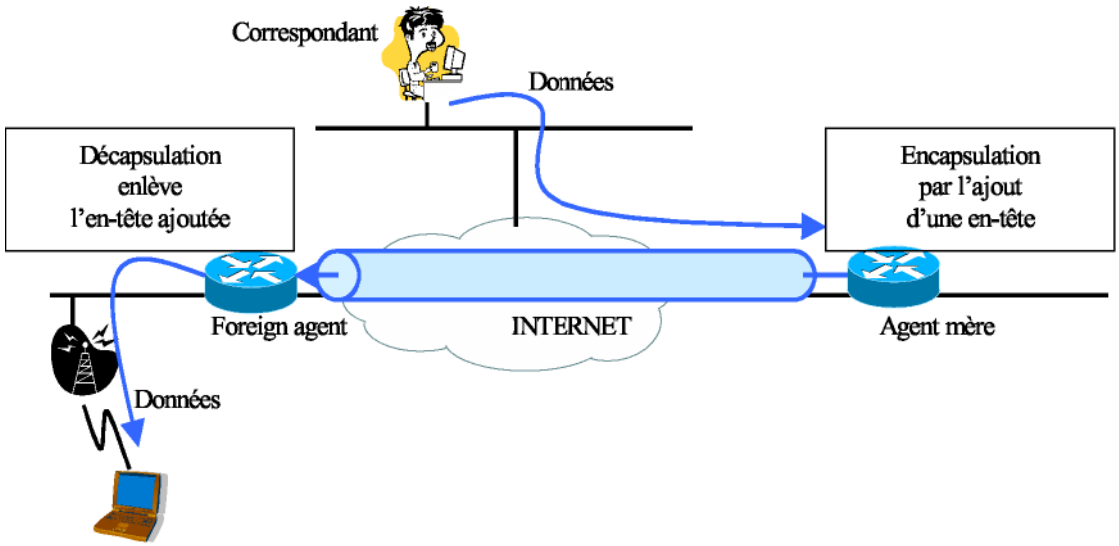


Figure 13. : Routage triangulaire : du correspondant au mobile

De l'autre côté, les paquets envoyés par le nœud mobile ont l'adresse du correspondant comme adresse destination et l'adresse principale du mobile comme adresse source. Ceci présente une entorse au modèle de l'Internet puisque l'adresse source des paquets envoyés par le nœud mobile ne correspond pas au préfixe du sous-réseau visité. Les paquets devront alors obligatoirement passer par l'agent visité pour éviter qu'ils ne soient détruits. Par contre, une fois que les paquets ont été routés hors du sous-réseau visité, ils vont directement du nœud mobile au correspondant sans passer par le réseau mère. C'est ce qu'on appelle le routage triangulaire (voir Figure 14) [Nicolas 2001].

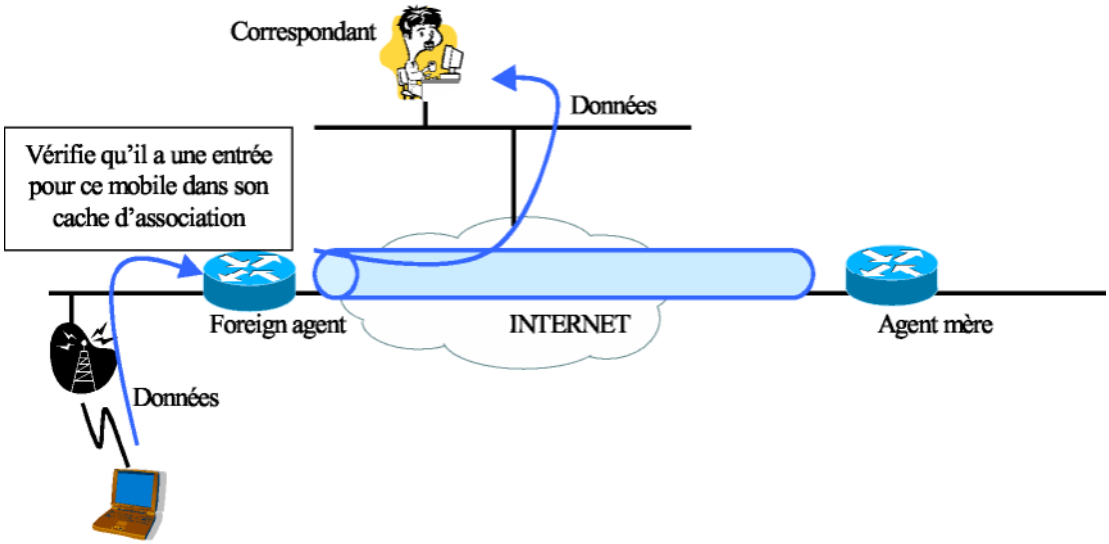


Figure 14. : Routage triangulaire : du mobile au correspondant

Etudions plus en détail les opérations nécessaires pour effectuer ce routage triangulaire : tout d'abord, lorsque le nœud mobile se déplace dans un sous-réseau visité, il doit en informer son agent mère à travers un *Registration Request*. A la réception de ce message, si l'agent mère accepte la requête, en plus de créer ou de mettre à jour l'entrée pour ce nœud mobile, il envoie une requête ARP sur le réseau principal afin de faire correspondre l'adresse IP du mobile avec son adresse MAC. Ainsi il peut intercepter les paquets à destination du mobile. Ensuite, l'agent mère doit faire suivre ces paquets à la position courante du mobile. Pour cela, il encapsule chaque paquet en ajoutant un en-tête de destination rempli avec l'adresse temporaire courante du mobile comme adresse destination et avec son adresse comme adresse source avant de les tunnellés à l'agent visité. Enfin, chaque paquet est décapsulé par l'agent visité (suppression de l'en-tête) et délivré au nœud mobile. Ces opérations sont décrites dans les Figure 11 et Figure 13.

3.1.3 Mobile IPv6

Une nouvelle version du protocole IP est en train d'émerger depuis quelques années : il s'agit de la version 6 du protocole IP. Ce protocole inclut entre autres la mobilité en standard. L'objectif de MIPv6 est d'offrir une communication directe entre un nœud mobile et ses correspondants (élimination du routage triangulaire) et éviter les ruptures des communications pendant les déplacements. Bien que MIPv6 reprenne des mécanismes de MIPv4, de nombreuses fonctionnalités supplémentaires ont été mises en place [Nicolas 2001].

3.1.3.1 Fonctionnalités requises

Dans MIPv6 [Nicolas 2001], le agent visité décrit dans MIPv4 n'existe plus. Par contre, l'agent mère est encore un routeur d'accès du sous-réseau principal du nœud mobile. Son rôle est le même que dans le cas de MIPv4, à savoir capturer les paquets à destination du mobile et les lui tunneller à sa localisation courante.

Par contre, les correspondants doivent mettre en nœud certains mécanismes supplémentaires : tout d'abord, ils doivent disposer d'un cache d'association tout comme l'agent mère ; dans ce cache sera stockée la correspondance entre l'adresse principale d'un nœud mobile avec lequel il a une communication et son adresse temporaire courante. Il devra donc être capable de traiter des messages de registration envoyés par un nœud mobile. De plus, il devra être capable d'effectuer le routage directement vers le mobile (*routing header*). Ceci constitue un apport important dans le fonctionnement de la mobilité puisque les paquets

des correspondants n'auront pas à passer par le réseau mère systématiquement. Mais toutes ces fonctionnalités supplémentaires ne sont faites qu'au niveau de la couche IP ; l'adresse identifiant la communication au niveau applicatif sera toujours l'adresse principale du nœud mobile, la couche IP cachant l'adresse temporaire source (ou destination selon qu'on se situe sur le nœud mobile ou le correspondant) [Nicolas 2001].

D'un autre côté, un nœud mobile doit toujours conserver la liste des correspondants auxquels il envoie un message de registration (pour les mises à jour éventuelles) et doit être capable de décapsuler lui-même les paquets qui lui sont transmis ; au niveau application, un nœud mobile utilise toujours son adresse principale, c'est pourquoi la couche IP doit pouvoir décapsuler l'en-tête indiquant l'adresse temporaire. Cette opération était exécutée par le agent visité dans MIPv4 [Nicolas 2001].

3.1.3.2 Découverte des routeurs d'accès

Le protocole de découverte des voisins [Nicolas 2001] offert par IPv6 joue un rôle important dans MIPv6. Il permet entre autres à des équipements situés sur le même lien physique de se découvrir mutuellement, de découvrir leurs adresses niveau 2 et de localiser les équipements de routage. Le processus de découverte des routeurs d'accès se déroule de manière similaire au protocole de découverte des agents ; tout routeur d'accès émet périodiquement des *Router Advertisment* contenant la liste des préfixes sur le lien. Un nœud mobile peut éventuellement en demander un explicitement, à travers un *Router Solicitation*.

Les routeurs d'accès offrant des fonctionnalités pour la mobilité émettent des *Router Advertisement* quelque peu modifié (pour avertir les mobiles de leur capacité). En outre, l'information contenue dans ces *Router Advertisement* permet aux nœuds mobiles de créer une adresse temporaire (auto configuration offerte par IPv6). Ensuite il leur faudra vérifier l'unicité de celle-ci grâce au protocole de détection de duplication d'adresse. La découverte des voisins ainsi que la découverte de l'adresse de niveau 2 d'un équipement voisin s'avère aussi très utile dans la mobilité, notamment pour effectuer des registrations plus rapides. L'utilisation des ces données sera détaillée plus tard dans le rapport car le protocole MIPv6 ne prend pas encore en compte ces données [Nicolas 2001].

3.1.3.3 Enregistrement

De la même manière que dans MIPv4, lorsqu'un nœud mobile se déplace hors de son sous-réseau mère, il doit en informer son agent mère. Le nœud mobile signale la correspondance entre son adresse principale et son adresse temporaire courante dans un message *Binding Update*. Ce message peut éventuellement être envoyé en « *piggybacking* ». En réponse à une telle requête, l'agent mère envoie un *Binding Ackowledgement* pour indiquer s'il peut répondre à la requête du mobile. Pour le moment, tout se passe comme dans MIPv4. Cependant, le mobile a par la suite la possibilité d'informer ses correspondants de sa position courante ; Lorsqu'il reçoit un paquet d'un correspondant, il détermine si le paquet a transigé par le réseau mère en regardant si le paquet contient un *routing header* ou s'il a été tunnelé par l'agent mère (encapsulation). S'il ne contient pas de *routing header*, le nœud mobile en déduit que le correspondant émetteur n'a pas d'entrée dans son cache d'association pour lui. Il peut alors lui envoyer un *Binding Update* pour qu'il lui envoie les paquets directement, sans plus passer par son sous-réseau mère (Figure 15) [Nicolas 2001].

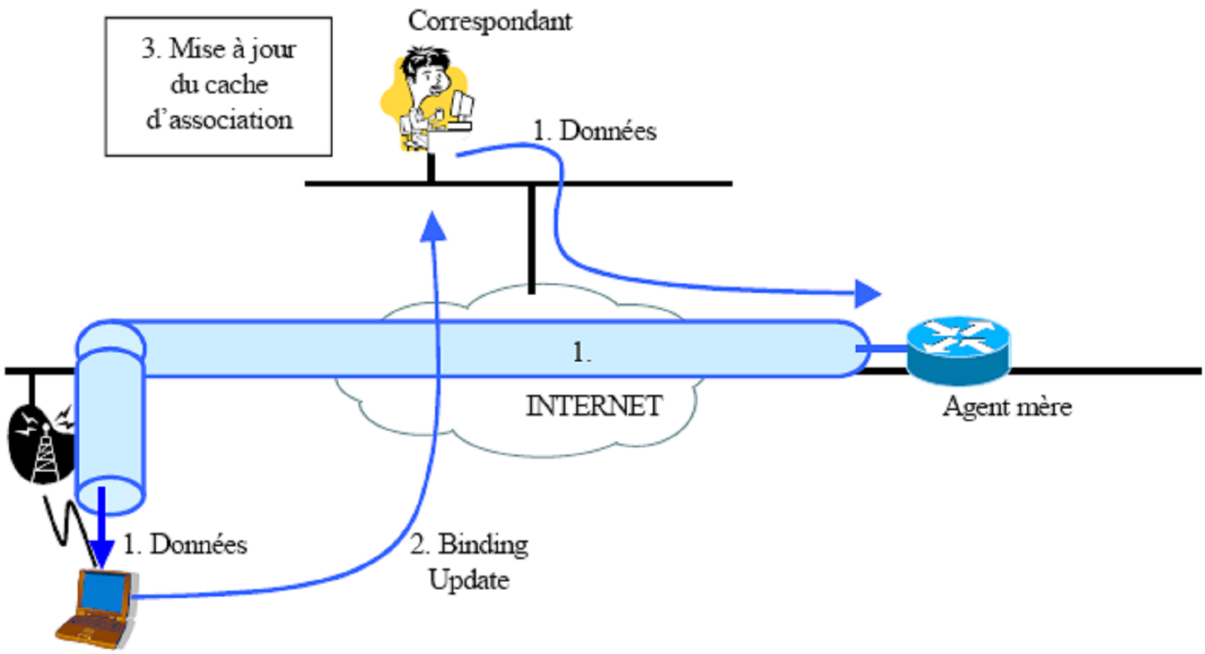


Figure 15. : Communication IPv6

Le fait que les correspondants aient la possibilité d'envoyer les paquets directement au mobile offre une meilleure résistance au facteur d'échelle et fiabilité. La communication entre nœuds mobiles et correspondants engendre moins de charge sur le réseau et est plus rapide. Comme l'agent mère est peu sollicité pour la retransmission des paquets, il y a beaucoup moins de risque de congestion au niveau de l'agent mère et une panne de l'agent mère aura un effet moindre.

Un nœud mobile peut détenir plus d'une adresse temporaire à un instant donné. Celle enregistrée auprès de l'agent mère (une seule) est dite principale. L'utilisation de plusieurs adresses temporaires peut être utile pour améliorer les performances lors d'un déplacement lorsque par exemple deux cellules de points d'accès se recouvrent fortement ; le nœud mobile peut alors acquérir une nouvelle adresse temporaire tout en utilisant son ancienne le temps de l'opération.

3.1.4 Comparaison de Mobile IPv4 avec Mobile IPv6

La conception de Mobile IPv6 s'est basée sur les expériences acquises du développement de Mobile IPv4 et sur les nouvelles opportunités offertes par le protocole Ipv6, telles que le nombre plus important d'adresses et les mécanismes d'auto configuration.

L'utilisation des options destination d'IPv6, qui fournissent des informations au nœud destinataire final, permet aux informations de contrôle de Mobile IPv6 d'être transportées dans l'entête des paquets IP contrairement à Mobile IPv4 où un paquet UDP spécifique doit être utilisé pour chaque type de message de contrôle. L'optimisation de routage est intégrée dans le protocole Mobile IPv6 puisqu'elle est assurée, comme l'enregistrement avec l'agent mère, par des messages de mise à jour d'associations. Le protocole IPv6 permet aux nœuds mobiles exécutant Mobile IP de communiquer à travers des routeurs filtrants en utilisant l'adresse temporaire comme adresse source. L'adresse mère est indiquée dans une option de destination, appelée option adresse mère, du paquet IPv6. Enfin, Mobile IPv6 ne requiert pas le déploiement d'agents étrangers. Les nœuds mobiles utilisent les mécanismes d'auto configuration IPv6 fonctionnant dans tous réseau IPv6 visité [Khouaja 2006].

3.2 Protocoles de Micro-mobilité :

3.2.1 Le protocole d'architecture hiérarchique

L'objectif de mettre en place une hiérarchie [Khouaja 2006] est de cacher certains mouvements des nœuds mobiles ; si un mobile se déplace à l'intérieur d'un domaine, il lui incombe uniquement de faire un enregistrement régional, sans indiquer quoique ce soit à l'extérieur du domaine. Par contre, lorsque le mobile change de domaine, il lui faudra faire un enregistrement global. Un domaine est défini comme étant une aire de mobilité locale. Généralement un domaine est indépendant des sous-réseaux et sa taille est choisie par l'opérateur réseau [Nicolas 2001].

La mise en place d'une hiérarchie permet de minimiser le trafic entre l'agent mère et les agents de mobilité, sans pour autant introduire de signalisation supplémentaire entre le nœud mobile et les agents de mobilité. Le modèle hiérarchique offre aussi des délais pour l'enregistrement plus petits. Ce modèle est bien adapté pour réaliser le fast handoff et le bi casting.

La définition d'une architecture hiérarchique passe par l'introduction d'un nouveau type de nœud qui agit comme un point d'ancrage pour les nœuds mobiles ; Dans MIPv4, il s'agit d'un agent visité passerelle et dans MIPv6 c'est un *Mobility Anchor Point*. Ces points d'ancrage sont des agents visités dans MIPv4 et des routeurs d'accès dans MIPv6 au sommet de plusieurs agents visités (resp. routeurs d'accès), avec une adresse IP publique (voir Figure 16) [Nicolas 2001].

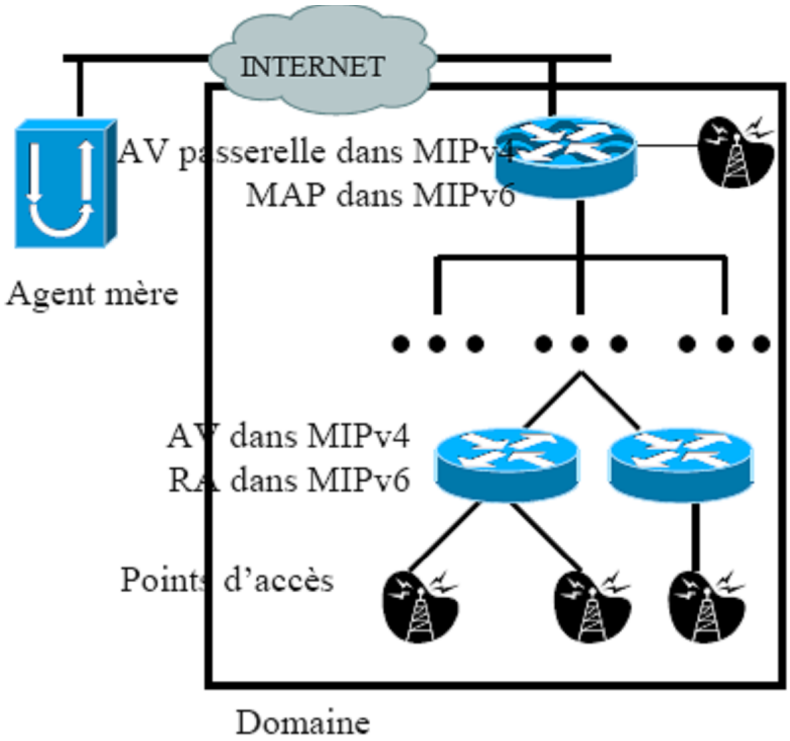


Figure 16. : Architecture Hiérarchique

Cette définition permet de mettre en nœud une hiérarchie avec plus d'un niveau. Lorsqu'un nœud mobile entre dans un domaine, un tunnel est crée entre lui et le point d'ancrage. Dans MIPv4, ce tunnel est par défaut unidirectionnel de l'agent visité passerelle jusqu'au nœud mobile. Si le nœud mobile désire établir un tunnel bidirectionnel, il lui faut positionner un bit 'T' dans son message d'enregistrement.

Dans la prochaine sous-section, on présente le fonctionnement de la hiérarchie et l'échange de messages dans telle architecture.

3.2.1.1 Le protocole Mobile IP Hiérarchique

Dans IP Mobile classique [Soliman 2000], du point de vue d'un terminal, l'architecture du réseau se compose seulement de trois régions : le réseau d'origine (*Home Network*) du terminal, le *Backbone Internet* et un réseau visité (*Foreign Network*). Les deux principales entités communicantes sont le FA dans le réseau visité et le HA dans le réseau d'origine. Lorsqu'un terminal se déplace localement sur un réseau en changeant son point d'attachement en local, la question est de savoir s'il est utile de propager cette information jusqu'au home agent, d'autant plus que si le HA est très loin du nouveau point d'attachement du mobile, cela va nuire fortement aux performances du relais.

Les architectures à base de proxy font un découpage plus fin du réseau. Elles mettent en place une hiérarchie de domaines. Au sein de chaque domaine se trouve un agent de mobilité qui sait quel agent de niveau inférieur joindre pour contacter le terminal mobile à l'intérieur de son domaine. Un mobile s'enregistre auprès d'un agent de mobilité de niveau le plus bas, celui-ci va s'enregistrer à son tour auprès de son agent de mobilité de niveau supérieur et ainsi de suite jusqu'au HA. Un agent de mobilité de niveau *n* n'est pas informé des déplacements du terminal mobile au sein des régions de niveau inférieur. En procédant de cette manière, les requêtes d'enregistrement n'ont plus besoin d'atteindre le HA mais restent confinées à l'intérieur d'une région localisée. On améliore ainsi les performances générales.

Les paquets à destination du mobile sont routés d'agents de mobilité en agent de mobilité au moyen tunnels IP. A chaque niveau, ils sont décapsulés puis encapsulés pour atteindre le niveau suivant. Par conséquent, même si on réussit à diminuer le temps d'interruption de service au moment du relais en limitant la propagation de l'information d'enregistrement, l'acheminement des paquets se complexifie et peut même être moins performant s'il y a trop de niveaux de hiérarchie. L'étape de découpage hiérarchique du réseau ne doit donc pas être négligée. De plus, la présence de tunnels est souvent problématique pour la QoS. Les mécanismes de réservation de ressources le long d'un chemin, tels que la signalisation RSVP, n'ont plus visibles par les routeurs traversés lorsqu'ils sont encapsulés dans un tunnel et ces mécanismes ne peuvent donc plus se faire.

Les principaux protocoles utilisant une architecture à base d'agents proxy sont IPv4 Mobile hiérarchique et IPv6 Mobile hiérarchique.

Un agent proxy fonctionne comme un point d'ancrage, chaque point d'ancrage est annoncé dans les *Agent Advertisement* envoyés par les AV/RA. Dans MIPv4, les *Agent Advertisement* contiennent toutes les adresses des agent visités entre le nœud mobile et l'agent visité passerelle. Dans MIPv6, en plus de l'adresse du *Mobility Anchor Point,* il y a le préfixe du domaine *du Mobility Anchor Point*, la distance au nœud mobile ainsi que les préférences du *Mobility Anchor Point* [Nicolas 2001].

Dans MIPv4, quand un nœud mobile entre pour la première fois dans un domaine, il doit s'enregistrer avec son agent mère en indiquant l'adresse de l'agent visité passerelle comme adresse temporaire. Ensuite lorsque le nœud mobile se déplace à l'intérieur du domaine (comme celui de la Figure 15 par exemple), il a juste besoin de faire des enregistrements locaux en chargeant un bit spécifique dans ses *Enregistrement Request*. Ces

Enregistrement Request ont l'adresse de l'agent visité passerelle comme adresse de destination et l'adresse du agent visité courant (ou 0 si le agent visité n'a pas annoncé son adresse) comme adresse temporaire [Nicolas 2001].

Dans MIPv6, la plupart des opérations sont les mêmes. Ceci dit, un nœud mobile a le choix entre deux modes lors de ses enregistrements dans un domaine : il s'agit des modes basique et étendu. Ces deux modes diffèrent dans le nombre d'adresses du nœud mobile. Dans le mode basique le nœud mobile a deux adresses : une adresse temporaire régionale basée sur le préfixe du Mobility Anchor Point et une adresse temporaire locale. Dans ce schéma, le Mobility Anchor Point agit comme un agent mère. Il intercepte les paquets à destination de l'adresse temporaire régionale et les tunnels à l'adresse temporaire locale correspondante. Ces opérations sont totalement transparentes à l'agent mère qui n'a besoin d'aucune modification.

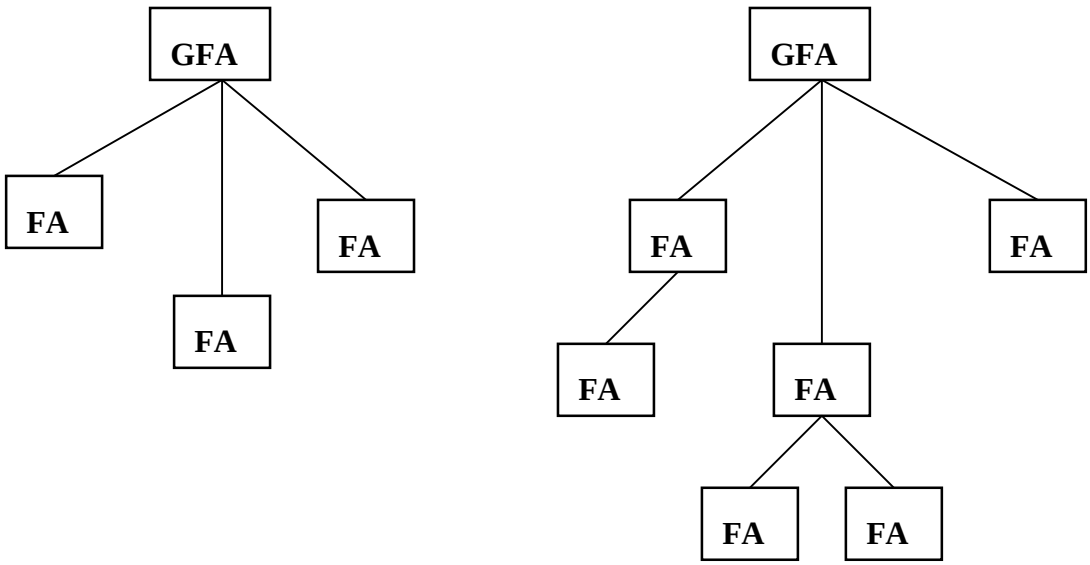
Cependant, dans un souci de mise à l'échelle ou pour des raisons de gestion, tous les nœud mobiles ne peuvent pas acquérir leur propre adresse régionale. Dans le mode étendu, l'adresse temporaire régionale est celle du Mobilty Anchor Point. Le Mobility Anchor Point tient une table des associations entre l'adresse principale des nœuds mobiles et leur adresse local. Quand le Mobility Anchor Point reçoit des paquets à destination d'un nœud mobile, il doit les décapsuler et les ré-encapsuler à l'adresse locale. Ceci implique que chaque paquet doit contenir l'adresse principale du nœud mobile. Le Mobility Anchor Point joue donc le même rôle qu'un agent mère. Le mode étendu supporte aussi bien les nœuds mobiles que les réseaux mobiles.

Aussi bien dans MIPv4 que dans MIPv6 le nœud mobile doit faire un enregistrement mère quand il change de domaine. Ce changement de domaine est détecté par le mobile dans les annonces des agents de mobilité. Effectivement, chaque AV/RA sous un point d'ancrage annonce l'adresse IP du point d'ancrage. Le nœud mobile compare l'adresse annoncée avec celle de son point d'ancrage courant. Si elles diffèrent, c'est qu'il a changé de domaine.

Toute cette procédure permet tout de même à un nœud mobile n'implémentant pas MIP hiérarchique d'appliquer MIP. De manière symétrique, un nœud mobile qui demande à faire un enregistrement local à un AV/RA n'implémentant pas MIP hiérarchique aura la possibilité de faire un enregistrement tel que décrit dans MIP [Khouaja 2006].

3.2.1.2 Mobile IPv4 hiérarchique et enregistrement

L'enregistrement régional de IPv4 Mobile [Gustafsson 2002] est une extension optionnelle, encore à l'état d'étude à l'IETF, pour le protocole IPv4 Mobile. Il offre la possibilité pour un mobile de faire un enregistrement de son FA localement. Sur le principe, ce protocole est assez proche de IP Mobile hiérarchique mais contrairement à HMIP [Soliman 2000], ce protocole ne propose que deux niveaux de hiérarchie d'agents visités. Un appendice dans le document d'étude parle néanmoins de la possibilité d'avoir plusieurs niveaux de hiérarchie (Figure 17).



a) Regional registration b) HMIPv4

Figure 17. : Mobile IPv4 hiérarchique et enregistrement

Un domaine IPv4 Mobile supportant également les enregistrements locaux comporte tout en haut de la hiérarchie au moins une passerelle d'agent visité (*Gateway Foreign Agent*, GFA) qui est un agent mobile visité avec quelques fonctions supplémentaires. Sous le GFA, il y a un ou plusieurs agents visités régionaux (*Regional Foreign Agent*, RFA). L'adresse temporaire *(care-of-address*) enregistrée auprès du home agent est l'adresse IP du GFA. Par conséquent, lorsqu'un terminal change d'adresse temporaire locale – que ce soit une adresse temporaire de l'agent mobile visité ou une adresse temporaire co-localisée – sous la même passerelle GFA, il effectue un enregistrement local auprès du GFA. Le home agent n'intervient pas dans cette phase. Par contre, si le terminal mobile change de GFA, il doit faire un enregistrement de l'adresse temporaire COA de son nouvel agent GFA auprès de son home agent. La figure 18 suivante, présente les deux cas d'enregistrements de l'adresse temporaire COA.

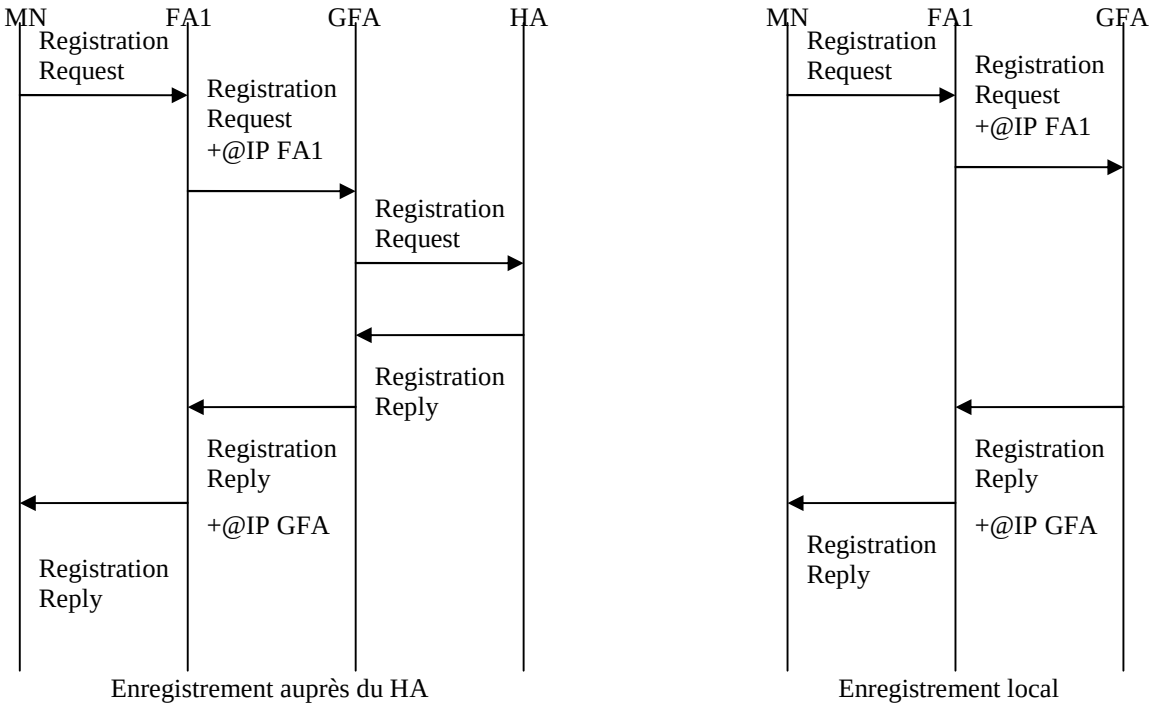


Figure 18. : Les enregistrements dans IPv4 Mobile avec enregistrement régional

Un des avantages de IPv4 Mobile avec enregistrement régional (*regional registration*) est qu'il nécessite peu de modifications par rapport à IPv4 Mobile aussi bien dans l'architecture du domaine que dans la partie logicielle de gestion du protocole. En ce qui concerne les messages d'annonce des agents de mobilité (*Agent Advertisements*) deux nouvelles extensions apparaissent :

- Un drapeau I pour indiquer que le domaine supporte la gestion d'enregistrement local,
- Les FA-NAI (*Foreign Agent Network Address Identifier*) qui sont optionnels et permettant au terminal de détecter le mouvement.

Les messages d'annonce (*Agent Advertisements*) peuvent contenir l'adresse IP du GFA. Si ce n'est pas le cas, un terminal peut utiliser l'extension GFA d'adresse IP.

Trois nouvelles extensions aux messages de IP Mobile registration sont définies :

- L'extension GFA d'adresse IP qui permet à un mobile de se voir assigner dynamiquement une adresse temporaire CoA de GFA ; le GFA ajoute une extension d'adresse IP GFA au message d'enregistrement (*Registration Request*) avant de la relayer au HA ; c'est cette adresse qui sera enregistrée par le HA ;
- L'extension hiérarchique (*Hiérarchical Foreign Agent*)
- L'extension *Relay Protection Style*.

Enfin, deux nouveaux messages d'enregistrement sont définis : *Regional Registration Request et Regional Reply*. Ces messages permettent à un mobile d'enregistrer son adresse temporaire CoA locale auprès de son GFA sans faire intervenir le HA. Ceci n'est possible que si le mobile connaît les extensions apportées IPv4 Mobile avec enregistrement régional (*Regional Registration*). Autrement, même si le réseau les implémente mais pas le mobile, le mobile effectuera toujours ses enregistrements auprès de son home agent.

3.2.1.3 Mobile IPv6 hiérarchique HMIPv6

Une amélioration de IPv6 Mobile est IPv6 Mobile hiérarchique [Bellier 2000] qui a tenté d'améliorer les performances de IPv6 Mobile en micromobilité. Un FA sera installé au

niveau de la passerelle du réseau visité formant ainsi un agent de mobilité nommé MAP (*Mobility Anchor Point*) correspondant au GFA dans HMIPv4. Le MAP va se charger de la procédure d'enregistrements régionale, en cachant ainsi au HA tous les déplacements au sein du même réseau visité. Le nœud mobile aura en plus de l'adresse permanente (home address), une adresse transitoire COA qui sera rattachée à la passerelle MAP, et une adresse colocalisée (Collocated COA) attribuée au niveau du réseau visité. Ainsi le HA garde la correspondance entre l'adresse permanente (Home Address) et la COA (MAP), et le MAP garde la correspondance entre la CCOA et la COA (MAP).

La procédure d'enregistrement est identique à celle de IP Mobile, la seule différence est que l'enregistrement avec le HA se fait uniquement si le nœud mobile change de MAP, sinon l'enregistrement au sein du réseau visité se fait auprès du MAP qui joue le rôle d'un HA localisée. Les paquets à destination du nœud mobile sont transmis directement du nœud correspondant vers le nœud mobile puisque celui-ci dispose de la localisation mise à jour du nœud mobile, et cela grâce à IPv6 Mobile. Ainsi, c'est le nœud mobile qui va décapsuler les paquets IP.

Contrairement au HMIPv4 où les paquets sont envoyés d'abord au HA qui va les encapsuler et les envoyer au GFA qui lui, va les décapsuler puis les ré encapsuler pour les envoyer au FA, où se trouve le nœud mobile. Ce dernier va décapsuler les paquets et va enfin les remettre au nœud mobile. Il est clair que la procédure de routage de HMIPv6 est plus performante que celle de HMIPv4.

3.2.1.4 Bi casting dans une architecture hiérarchique

Comme il a été déjà montré, le bi casting à partir de l'agent mère n'est pas scalable et peut générer une congestion. Le modèle hiérarchique permet de faire le bi casting à partir du point d'ancrage. Quand un nœud mobile se déplace à l'intérieur d'un domaine, il peut demander le bi casting dans ses messages d'enregistrements régionaux.

Cette demande est transmise au point d'ancrage qui ajoute une entrée pour le nœud mobile (association simultanée). Ensuite le point d'ancrage propage le trafic à l'ancienne et à la nouvelle localisation du nœud mobile.

L'échange de messages est donc le même. Le modèle hiérarchique permet donc de réaliser le bi casting plus rapidement et de manière efficace. Si un problème de mise à l'échelle se pose (par un nombre trop important de nœuds mobiles) il suffit d'ajouter des points d'ancrage [Nicolas 2001].

3.2.2 Le protocole Cellular IP

Le protocole Cellular IP [Campbell 2002] [Nicolas 2001] utilise aussi une architecture hiérarchique ; l'Internet globale est divisé en domaine constituant des aires de micro mobilité. Chaque niveau de mobilité est traité par un protocole différent, adapté aux besoins du niveau.

Cellular IP, inspiré des systèmes cellulaires, se propose de résoudre la gestion de la mobilité à l'intérieur d'un domaine. Ce protocole a été conçu pour répondre rapidement à un grand nombre de nœud mobile qui migrent fréquemment. Par contre, c'est MIP qui gère la mobilité entre domaines. Cellular IP interagit donc avec MIP, que ce soit dans sa version 4 ou 6 (avec des changements mineurs).

Dans une aire de micro mobilité, l'information de routage est totalement distribuée, c'est-à-dire qu'aucun des nœuds ne garde une information globale sur la topologie du réseau. Ceci rend la solution robuste. De plus, Cellular IP peut être utilisé dans un domaine allant d'un bureau à un réseau s'étendant sur une ville et peut supporter un grand nombre d'hôtes. Le but de Cellular IP est de procurer un *seamless handoff* à l'intérieur d'un domaine et de cacher les mouvements des nœuds mobiles au reste de l'Internet.

Cellular IP fonctionne avec une hiérarchie composée d'un routeur passerelle et de nœud cellular IP, qui sont des points d'accès et des nœuds mobiles implémentant le protocole Cellular IP (voir Figure 19). Le routeur passerelle assure la liaison entre le réseau cellulaire IP et le reste d'Internet. Il filtre, contrôle et propage les paquets en provenance et à destination du réseau cellulaire IP. Les points d'accès sont connectés ensemble par un réseau filaire et ont une interface sans fil pour communiquer avec les nœuds mobiles [Nicolas 2001].

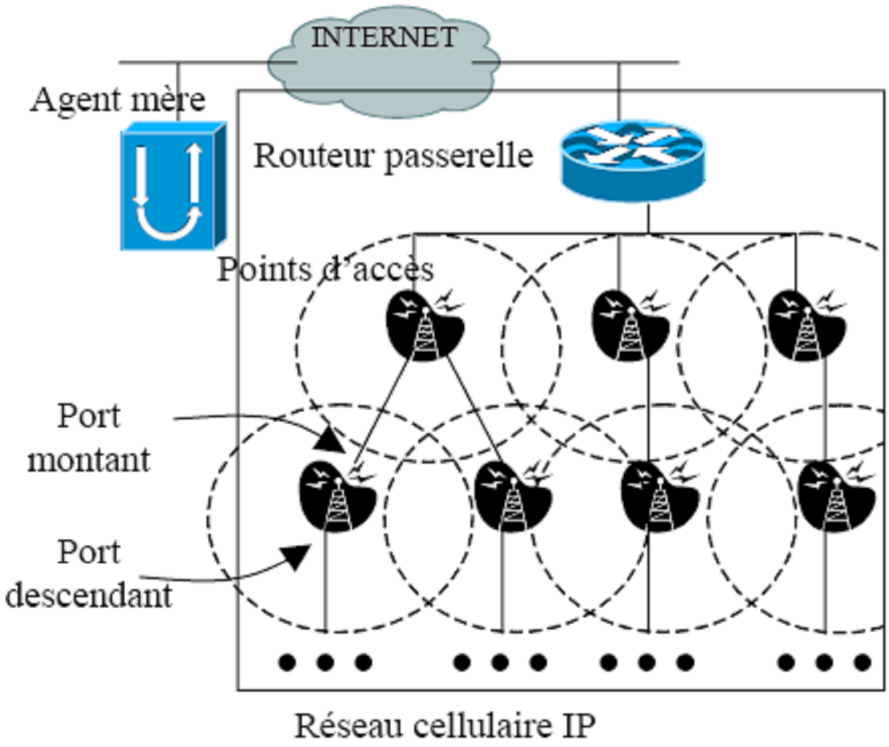


Figure 19. : Architecture Cellular IP

La gestion de la localisation est très différente de celle utilisée par MIP. L'adresse IP n'est plus utilisée pour localiser un équipement mais uniquement pour l'identifier. De ce fait, il n'est plus nécessaire de faire des encapsulations ou des conversions d'adresse. La localisation des nœuds mobiles est contenue dans deux différents caches dans les points d'accès. L'utilisation de deux caches permet de différencier le traitement des nœuds mobiles actifs de ceux inactifs. On dira qu'un hôte est actif quand il échange des paquets de données avec au moins un correspondant. Le cache de pagination est utilisé pour localiser les hôtes mobiles inactifs et le cache de routage pour la localisation des hôtes actifs. Ces caches sont des *mappings* entre l'adresse IP du nœud mobile et l'interface du point d'accès utilisé pour atteindre ce nœud. Ces *mappings* sont construits par le chemin inverse des paquets envoyés par le nœud mobile (aussi bien les paquets de données que ceux de control). Le routage se fait donc en saut-par-saut sur le plus court chemin [Nicolas 2001].

La localisation des nœud mobiles inactifs est approximative : les points d'accès sont organisées en aires de pagination et cherchent uniquement à savoir dans quelle aire se situe le nœud mobile. La gestion de la localisation est décrite plus tard. Dans un réseau cellulaire IP donné, chaque aire de pagination a un identifiant unique. Cet identifiant est annoncé par toutes les points d'accès mais toutes les points d'accès n'ont pas de cache de pagination.

Dans ce schéma, un nœud mobile n'a pas de point d'attache dédié, il utilise le meilleur à tout moment. Il n'y a donc pas d'authentification entre les points d'accès. Cependant, un nœud est effectué lors de l'entrée des nœud mobiles dans le réseau cellulaire IP et seuls les

messages de nœud peuvent créer des nouvelles entrées dans les caches pour assurer une sécurité.

3.2.2.1 Détail du protocole

Trois messages de contrôle sont nécessaires pour le fonctionnement du protocole : *Route Update*, *Paging-Update* et *Paging-Teardown*. L'utilisation de ces messages est décrite au fil de ce paragraphe. Le routeur passerelle envoie périodiquement des messages aux points d'accès pour leur indiquer leur port montant (*uplink port*), c'est-à-dire le port utilisé pour atteindre le routeur passerelle (voir Figure 19). Ceci rend le protocole *plug-and-play* puisque aucune configuration préalable n'est requise sur les points d'accès. Tous les autres ports des points d'accès sont des ports descendants (voir Figure 19). Les points d'accès envoient aussi des beacons contenant entre autres l'identificateur du réseau cellulaire IP, l'adresse IP du routeur passerelle et un identifiant d'aire de pagination sur leur interface radio. Quand un nœud mobile entre pour la première fois dans un réseau cellulaire IP, il envoie une authentification et des informations utilisateur dans un *Paging-Update* à destination du routeur passerelle. Si le routeur passerelle accepte la demande du nœud mobile, celui-ci doit faire un enregistrement mère. Dans le cas de MIPv4, l'adresse indiquée dans cet enregistrement est celle du routeur passerelle, alors que dans MIPv6, c'est une adresse temporaire locale créée à partir du préfix réseau [Nicolas 2001].

Comme on l'a dit plus haut, un nœud mobile peut être dans deux états. Lorsqu'il est inactif, il doit mettre à jour les caches de pagination à chacune de ses entrées dans une nouvelle aire de pagination ou juste avant que son entrée n'expire (cas où le nœud mobile ne s'est pas déplacé). Cette mise à jour est assurée par l'émission d'un *Paging-Update*.

Un *Paging-Update* [Nicolas 2001] détruit aussi les entrées dans les caches de routage. Un nœud mobile peut par ailleurs demander explicitement la destruction de son entrée dans un cache de pagination pour éviter tout conflit. La suppression explicite d'une entrée dans le cache de pagination est faite par l'envoie d'un *Paging-Teardown*.

Dans son état actif, un nœud mobile envoie des paquets de données, en reçoit ou fait les deux. Les paquets de données envoyés par le nœud mobile mettent à jour les caches de routage sans que le mobile n'ait besoin d'envoyer d'autres paquets de control. Par contre, quand un nœud mobile ne fait que de recevoir des données, il lui faut envoyer périodiquement un *Route-Update* pour éviter que son entrée dans les caches de routage n'expirent. Ceci dit, que le nœud mobile soit en émission ou en réception, il doit envoyer un *Route-Update* chaque fois qu'il change de point d'accès. Ce paquet crée une nouvelle entrée dans tous les nouveaux points d'accès sur le chemin allant du nœud mobile au routeur passerelle.

Quand un flux de données arrive pour un nœud mobile inactif, le premier paquet est utilisé pour faire la pagination (pagination implicite). Le paquet est transmis suivant les informations contenues dans les caches de pagination. Si jamais un point d'accès reçoit un paquet pour un nœud mobile dont elle n'a aucune entrée (pas de cache de pagination), elle duplique le paquet sur tous ses ports descendants. Quand le nœud mobile reçoit ce premier paquet, il envoie un *Route-Update* pour créer une entrée dans les caches de routage et devient actif.

3.2.2.2 Traitement du handoff

Le changement de point d'accès, appelé hard handoff [Nicolas 2001], est automatiquement géré par le protocole. Cependant, des paquets peuvent être perdus pendant le temps que le Route- Update atteigne le point d'accès qui doit réaliser le changement de route. Quand un nœud mobile peut interagir simultanément avec deux points d'accès, il peut faire un *semi-soft handoff*. Quand un nœud mobile décide de se déplacer à un nouveau point d'accès, il envoie un *Route-Update* avec un flag spécifique à travers le nouveau point d'accès

et retourne écouter l'ancien. Quand le *Route-Update* atteint le premier point d'accès qui doit modifier et non créer l'entrée pour le nœud mobile, une nouvelle entrée est créée sans remplacer l'ancienne. Les paquets de données sont alors envoyés à l'ancienne et à la nouvelle localisation du nœud mobile. Quand le nœud mobile décide par la suite de se rattacher au nouveau point d'accès, il envoie un *Route-Update* pour détruire l'entrée avec l'ancien point d'accès.

Pour les nœuds mobiles ne pouvant pas être connectés simultanément à deux points d'accès, la technique de indirect *semi-soft handoff* se rapproche du *semi-soft handoff*. Quand le nœud mobile décide de changer de point d'accès, il envoie un *Route-Update* avec un flag 'I' à travers son ancien point d'accès avec l'adresse du nouveau point d'accès dans le champ de l'adresse destination. Ce paquet est transmis au routeur passerelle qui l'envoie au nouveau point d'accès. A réception de ce paquet, le nouveau point d'accès envoie un *Route-Update* avec l'adresse du nœud mobile comme adresse source et on se retrouve alors dans la même situation que dans le *semi-soft handoff* [Nicolas 2001].

3.2.3 Le protocole Hawaii

Contrairement à Cellular IP, HAWAII [Ramjee 2000] [Reinbold 2001] ne remplace pas IP mais s'appuie sur lui dans son fonctionnement. Chaque station du réseau doit donc pouvoir fournir les services d'un routeur IP classique plus certaines fonctionnalités de gestion de la mobilité.

La gestion de la mobilité se fait de façon très similaire à Cellular IP : chaque station maintient un cache de routage qui lui permet de déterminer le traitement à appliquer aux paquets qu'elle reçoit. Le handover est traité suivant plusieurs mécanismes. Ces mécanismes offrent des performances similaires à celles de Cellular IP mais peuvent être sélectionnés en fonction des priorités du gestionnaire de réseau vis-à-vis de la perte de paquet, de la latence et du *ré-ordonnancement*. HAWAII présente un support natif de la connectivité passive et du *paging*. Ce comportement du mobile est modélisé dans une machine à états finis. Dans HAWAII, les stations faisant partie d'une *paging* area sont toutes membres du même groupe IP multicast. Ceci permet de distribuer une requête de *paging* à toutes les stations d'un même groupe en l'adressant à ce groupe.

3.2.4 Le protocole TeleMIP

TeleMIP [Subir 2000] [Reinbold 2001] est un protocole simple et bien adapté au cas des réseaux CDMA. TeleMIP définit un nouvel agent de mobilité que nous appellerons *TeleMIP Mobility Agent* (TMA). Un réseau TeleMIP est composé d'un ensemble de sous-réseaux (composés d'une machine centrale agissant comme FA local et d'un ensemble de stations de base qui lui sont connectées) et d'un série de machines faisant fonction de TMA. Chaque FA est connecté à au moins un TMA du réseau global. Lorsqu'un mobile se connecte au réseau, il obtient deux adresses temporaires du FA local : l'une est enregistrée auprès d'un TMA et restera valide tant que le MN restera dans le domaine (cela lui servira de COA), l'autre est strictement locale et ne sert à identifier le mobile que tant qu'il reste dans le sous-réseau. Le fonctionnement de TeleMIP est très simple : lors- qu'un paquet arrive dans le réseau pour un mobile, il porte la COA du mobile comme adresse destination. Il est intercepté par le TMA concerné qui consulte sa table de correspondance entre adresses locales et COA. Ceci lui permet alors de faire suivre le paquet jusqu'au réseau local où se trouve le mobile à ce moment, le FA local n'ayant plus qu'à délivrer le paquet au MN via la bonne station de base.

3.2.5 Le protocole EMA

EMA [O'Neill 2000] [Reinbold 2001] vise à définir un système générique de gestion de la mobilité dans les domaines d'accès mobile. Dans ce système, les auteurs de EMA discutent la possibilité d'utiliser TORA [Parks 1997] [Parks 1999], un protocole de gestion de réseaux ad-hoc, pour le cas des réseaux mobiles mais ne restreignent pas leur approche à ce seul cas particulier. Les réseaux ad-hoc constituent en effet une forme extrême de réseau mobile. Dans ce type de réseau, on ne suppose l'existence d'aucune architecture fixe : les stations sont toutes équivalentes et se déplacent les une par rapport aux autres. Des protocoles très particuliers doivent alors gérer les interactions entre celles-ci pour assurer l'acheminement des paquets dans une topologie sans cesse mouvante. EMA ne fait aucune supposition sur une technique d'accès radio particulière et propose une gestion du handover transparente pour le protocole de routage tournant au dessus. Au moment de la connexion au réseau, un MN obtient une adresse du sous-réseau dans lequel il se trouve, permet- tant ainsi un routage par préfixe tant qu'il y reste dans ce sous-réseau. Lorsqu'un mobile sort du sous-réseau auquel il s'est d'abord connecté, EMA prévoit l'injection de routes spécifiques pour atteindre celui-ci.

3.3 Comparaison entre les déférents protocoles

Dans cette section [Reinbold 2001] [Campbell 2002], nous comparons effectivement les différents protocoles de mobilité IP vis-à-vis des différents critères que nous avons définis précédemment.

3.3.1 Handover

Nous examinons trois caractéristiques essentielles du mécanisme de handover :

- Temps de latence : le temps nécessaire pour que le réseau soit stabilisé après un handover
- La perte de paquet : le nombre de paquets potentiellement perdus à cause du handover
- La signalisation : la quantité de message de signalisation qui doivent circuler dans le réseau pour gérer chaque handover

Pour cette comparaison [Reinbold 2001], nous utiliserons un modèle de réseau simplifié. Nous supposerons que *n1* est le nombre moyen de sauts entre un MN et le *gateway* du réseau d'accès. Le temps pour parcourir cette distance sera de *t1* msec. De la même façon, *n2* représente le nombre moyen de sauts entre le MN et la station de base avec laquelle il était connecté avant le handover. Nous supposerons que cette station peut être jointe en un temps moyen de *t2 msec*. Enfin, *t3* représente le temps moyen pour atteindre la *crossover base station*, la station de base se trouvant à l'intersection du chemin menant du MN au geteway et du chemin menant du MN à son ancienne station de base.

Nous pouvons considérer qu'en moyenne $t1 \geq t2 \geq t3$. Nous appellerons *ntora* le nombre moyen de messages échangés pendant la stabilisation d'un réseau TORA (ce nombre dépend évidemment de la topologie du réseau considéré). *t0* sera le temps nécessaire pour atteindre la HA lors du processus d'enregistrement dans Mobile IP. Enfin, τ est le taux d'émission moyen d'un MN durant le handover.

Dans Cellular IP, la mise à jour des tables de routage se fait via l'émission par le MN d'un paquet spécifique qui est retransmis de proche en proche jusqu'au gateway.

Ce dernier doit alors envoyer un acquittement et le handover ne se termine qu'au moment où le MN le reçoit. Le temps de latence de ce mécanisme est donc de *2t1* avec *n1* mises à jour dans les différentes machines du réseau. Cellular IP propose deux types de handoff différents. Avec le semi-soft handoff, nous pouvons considérer que les pertes de

paquets seront nulles. D'autre part, le hard handoff génèrera *τt3* pertes de paquets en moyenne. En effet, c'est seulement à partir du moment où le paquet d'update atteint la station crossover que le routage est à nouveau correctement réalisé.

Le mécanisme de gestion du handover dans HAWAII est constitué d'un échange de message entre les deux stations de base concernées. La latence du mécanisme sera donc 2t2. Par contre, les pertes de paquets seront différentes suivant le *path setup scheme* utilisé. Dans le cas du *forwarding path setup scheme*, les pertes seront de τt2car il faut que l'ancienne station de base soit atteinte par le paquet de mise à jour pour que le routage se fasse à nouveau correctement. Dans le cas du *non-forwarding path setup scheme*, les pertes seront de τt3 pour les mêmes raisons que dans le cas de du hard handoff de Cellular IP.

TeleMIP gère le handover via l'envoi d'un message au TMA dans le cas d'un changement de sous-réseau et par un enregistrement classique Mobile IP dans le cas d'un changement de TMA. Si nous considérons que le TMA est situé à la même distance du MN que le *gateway*, la latence est, dans le premier cas, de 2t1 et dans le second *2t0*.

Les pertes n'ont cependant lieu que pendant le temps nécessaire à atteindre le TMA, elles sont donc de τt1.

EMA défini des mécanismes de gestion du handover qui permettent d'éviter toute perte de paquet si le handover peut être anticipé (comme c'est le cas dans les réseaux dont la technologie radio est basée sur CDMA). La latence de ce protocole est cependant longue puisqu'elle est constituée du temps nécessaire à effectuer un three way handshake entre les deux stations de base concernées et du temps mis par le routage du réseau TORA à se stabiliser.

Il apparaît clairement qu'un compromis doit être fait et qu'aucune des propositions n'est totalement satisfaisante, réunissant latence minimale, pertes nulles et nombre d'updates réduit. Dans cette optique, HAWAII peut sembler offrir un bon compromis.

3.3.2 Connectivité passive et Paging

Deux propositions seulement incluent ces importantes fonctionnalités : HAWAII et Cellular IP [Reinbold 2001]. Toutes les deux utilisent le concept de location area que l'on retrouve dans les réseaux de type GSM. En effet, les stations sont regroupées sur une base géographique en paging area et la localisation d'un MN dans une de ces zones se fait via le processus de *paging*.

La différence principale entre Cellular IP et HAWAII est située au niveau de l'algorithme de *paging*. Dans Cellular IP, le Gateway effectue un *paging* dès qu'un paquet arrive pour un MN en mode passif. Le processus de *paging* est pris en charge par le Gateway et un ensemble de machines dédiées gardant en mémoire l'information nécessaire. Ces machines ont donc toute la charge du *paging* et sont définies statiquement. A contraire, HAWAII défini un algorithme de répartition dynamique de la charge du *paging* sur les machines du réseau, en tendant à repousser cette charge vers les stations de base. La machine qui effectue un *paging* est choisie dynamiquement sur base de la charge effective des stations du réseau.

3.3.3 Support du trafic interne au réseau d'accès

La plus grande part des communications GSM aujourd'hui est constituée d'échanges de données entre utilisateurs du même réseau. Cette section [Reinbold 2001] compare la manière dont les différentes propositions supportent ce type de trafic.

Avec Cellular IP, tout le trafic venant du MN doit passer par le Gateway, même si le destinataire se trouve dans le même réseau. Ce système, loin d'être optimal du point de vue du

routage, impose un surplus inutile de charge de travail au Gateway et aux stations environnantes. HAWAII est un protocole dont le niveau de fonctionnement est au dessus de IP, on peut donc supposer que le trafic destiné à l'intérieur du réseau sera routé directement. L'utilisation de TORA permet à un réseau EMA de faire un routage efficace du trafic interne pour autant que le réseau implémente toutes les fonctionnalités de TORA. Dans le cas de TeleMIP, tous les paquets provenant d'un MN doivent passer par son TMA. Ceci implique les mêmes restrictions que dans le cas de Cellular IP.

3.3.4 Qualité de Service

Le support de Qualités de Service [Reinbold 2001] dans un réseau mobile est évidemment lié aux mécanismes de handover, de contrôle de trafic et de *paging*. . . Cependant, la possibilité pour un protocole d'interagir efficacement avec les mécanismes QoS existants, comme RSVP, est également très importante. Dans cette perspective, un avantage important de l'approche de la micromobilité est que les réservations ne doivent pas être refaites sur l'entièreté du chemin chaque fois que le MN change son point d'attache dans un domaine. En effet, le chemin hors du domaine reste inchangé et ne doit donc pas faire l'objet d'une nouvelle réservation. Nous nous intéresserons ici à la manière dont un réseau d'accès peut gérer les modifications de chemin à l'intérieur du domaine.

Seul HAWAII présente une intégration directe de RSVP dans sa spécification. Ce protocole fonctionnant au dessus de IP. Pour un MN receveur, rien ne sera fait sur la partie inchangée du chemin au moment du handover. Les réservations ne seront refaites que sur les nouvelles parties du chemin (grossièrement à partir du *crossover* jusqu'au MN). Si le MN est émetteur, il suffit qu'il envoie un message PATH après le handoff pour que RSVP s'adapte suivant les mécanismes classiques de ce protocole.

Les deux particularités de HAWAII qui rendent ceci possible sont :

– L'utilisation d'une seule et unique adresse COA pendant tout le séjour d'un MN dans le domaine,

– Le fonctionnement de HAWAII au dessus de IP, ce qui rend naturel le déploiement de RSVP dans HAWAII.

Cellular IP, TeleMIP et EMA ne mentionnent pas explicitement une intégration de mécanismes de QoS. Cependant, ils profitent des avantages liés à la micro-mobilité. Si l'intégration de ces mécanismes semble assez simple dans Cellular IP et TeleMIP, il n'en est pas de même pour EMA. En effet, des modifications majeures devraient être faites dans TORA pour pouvoir assurer une qualité de service. Dans le cas de TeleMIP les réservations pourraient même profiter de la gestion très localisée du handover au niveau du sous-réseau.

L'utilisation de RSVP est souvent associée à la perspective Integrated Services [Zheng 2001], une autre approche bien connue est celle de *differentiated services* [Zheng 2001]. En dépit de son aspect plus simple et plus directement praticable, cette dernière approche n'est absolument pas envisagée dans la description des différents protocoles.

3.3.5 Niveau de fonctionnement des stations

Les réseaux GSM actuels fonctionnent avec des millions d'utilisateurs connectés simultanément et nous pouvons considérer que les futurs réseaux mobiles devront supporter au moins une charge équivalente à celle-là. Par exemple, le routeur CISCO 7206 fonctionnant en GGSN GPRS peut supporter jusqu'à 90000 contextes d'utilisateurs simultanément [Reinbold 2001]. Ceci est à mettre en relation avec les problèmes de gestion des tables de routage de plus en plus grandes dans l'Internet aujourd'hui : gérer dynamiquement une table de quelques centaines de milliers d'entrées devient un véritable problème.

Dans Cellular IP et HAWAII [Reinbold 2001] [Campbell 2002], le *gateway* est une machine extrêmement chargée. Elle doit maintenir en permanence une table contenant une entrée par mobile effectivement connecté au réseau. La gestion de cette table, sous les contraintes que nous avons mentionnées plus haut, est particulièrement difficile. HAWAII assume pour sa part que le réseau est composé de routeurs IP classiques capables de gérer la mobilité. Les stations doivent donc non seulement effectuer les fonctions de routage classiques mais aussi assurer leur rôle dans HAWAII ; elles sont donc particulièrement chargées.

Pour Cellular IP, les stations du réseau doivent assurer les fonctions de «switchs évolués », ce qui les rends plus simples et plus légères. Cependant, les mécanismes évolués de Cellular IP comme le semi-soft handoff ou le paging, augmentent sensiblement les fonctionnalités devant être supportées par les stations, les rendant dans les faits plus proches de vrais routeurs que de switchs.

TeleMIP [Reinbold 2001] ne définit pas de réel mécanisme de routage. Cependant, la multiplicité des MA permet d'utiliser des algorithmes de *load balancing* pour alléger la charge de chaque machine, en leur permettant par exemple de ne conserver en mémoire qu'une table de *forwarding reduite*. Il semble qu'avec des topologies simples, les MA peuvent fonctionner comme de simples switchs. On peut cependant s'attendre à ce que les futurs réseaux d'accès présentent des topologies diverses et complexes.

EMA [Reinbold 2001] utilise TORA pour gérer le routage dans le réseau d'accès, en prennent soin de permettre un routage par préfixe et un routage par adresse. Ceci semble un bon compromis pour diminuer la taille des tables en mémoire. Cependant, TORA étant un protocole prévu pour les réseaux ad-hoc, de nombreuses routes sont définies pour parvenir à une destination (tout un graphe pour chaque destination en fait). Chaque station située sur une de ces routes doit maintenir de l'information spécifique à celles-ci (sa hauteur relative à la destination). Dans des réseaux de large taille, la multiplicité des routes et la gestion de ces informations redondantes deviendra un problème pour des stations aux ressources limitées. Ce problème est rendu d'autant plus critique que la redondance est largement moins nécessaire dans un réseau d'accès sans fil que dans un réseau ad-hoc puisque la plus grande part du réseau d'accès est câblée et fixe. Dans un réseau ad-hoc, la disponibilité de plusieurs routes par destinations est extrêmement utile pour faire face aux multiples changements de topologies et aux pertes de connectivité. Sur cette base, on peut voir que les tables devant être maintenues en mémoire par les stations d'un réseau EMA peuvent devenir plus importantes que dans les autres propositions et que la charge supplémentaire qui en résulte semble inutile en raison des spécificités des réseaux d'accès sans fil.

Chapitre 4 :

Le modèle proposé

Chapitre 4 : Le Modèle proposé

Introduction

Dans les réseaux mobiles, les utilisateurs mobiles (nœuds) changent librement leurs points de service pendant qu'ils sont connectés. Dans cet environnement, la gestion de mobilité est une technologie essentielle en vue de définir un mécanisme permettant de maintenir les sessions ouvertes lors des déplacements d'une part et d'une autre part de déterminer la nouvelle position du nœud dans la topologie (localisation et routage). Ceci se fait généralement avec coût de messages de signalisation.

Le mobile hiérarchique IPv6 (HMIPv6) [Soliman 2004] [Pack 2004] [Pack 2007] (voir Figure 20) est un mobile IPv6 amélioré pour réduire au minimum le coût de signalisation en utilisant un agent de mobilité nommé MAP (*Mobility Anchor Point*). Le MAP va se charger de la procédure d'enregistrement régionale, en cachant du HA tous les déplacements au sein du même réseau visité. Le nœud mobile aura en plus de l'adresse permanente (*home address*), une adresse transitoire COA ou régionale (*RCoA*) qui sera liée à la passerelle MAP, et une adresse co-localisée (*Collocated COA*) (*LCoA : on-link CoA*) attribuée au niveau du réseau visité. Ainsi le HA (Home Agent) garde la correspondance entre l'adresse permanente (*Home Address*) et la RCOA (MAP), et le MAP garde la correspondance entre la LCOA et la RCOA (MAP). Le MAP est prévue pour limiter la quantité des messages de signalisation de mobile IPv6 hors du domaine local.

Le MAP [Pack 2005] [Pack 2007] intercepte tous les paquets destinés vers le nœud mobile et les transmit directement à l'adresse courante du MN. Quand le MN change son adresse courante dans un domaine local de MAP, il doit seulement enregistrer un nouveau CoA avec le MAP. Par conséquent, seule le RCoA doit être inscrit aux nœuds correspondants (le CNs) et l'agent mère (Home Agent). Le RCoA ne change pas tant que le MN se déplaçant dans le même domaine du MAP. Ceci rend la mobilité du MN transparente au CNs avec lequel le MN communique. Les frontières d'un domaine de MAP sont définies à l'aide des routeurs d'accès (Access Routers ARs), diffusants l'information de MAP au MNs joints.

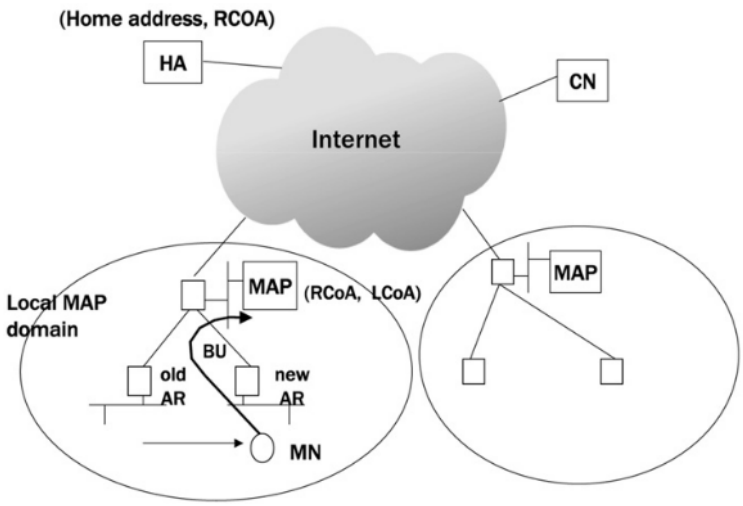


Figure 20. : Architecture de Mobile hiérarchique IPv6

4.1 Solutions existantes :

4.1.1 Architecture à plusieurs MAPs (SHMIPv6) :

Le MAP agit comme un agent local pour le nœud mobile (MN).

L'utilisation d'un seul MAP [Ma'en 2007] garde un grand nombre de paquets en attente, ce qui engendre un retard et plusieurs paquets seront perdus. Par conséquent, la communication sera affectée (interrompue ou découpée). En outre, plusieurs MAPs [Pack 2005] [Chung 2007] sont nécessaires pour la gestion du trafic, cette proposition appelée Smart Hierarchical Mobile IPv6 (SHMIPv6). Dans ce protocole (voire Figure 21) l'ensemble des MAPs est utilisé pour assurer la gestion des informations du trafic de tunnel et la gestion de l'information d'enregistrement dans chaque Agent mère (HA) pour diminuer et empêcher la surcharge du trafic. Le nombre des MAPs dépend du nombre de MNs et du nombre des paquets transférés dans le domaine.

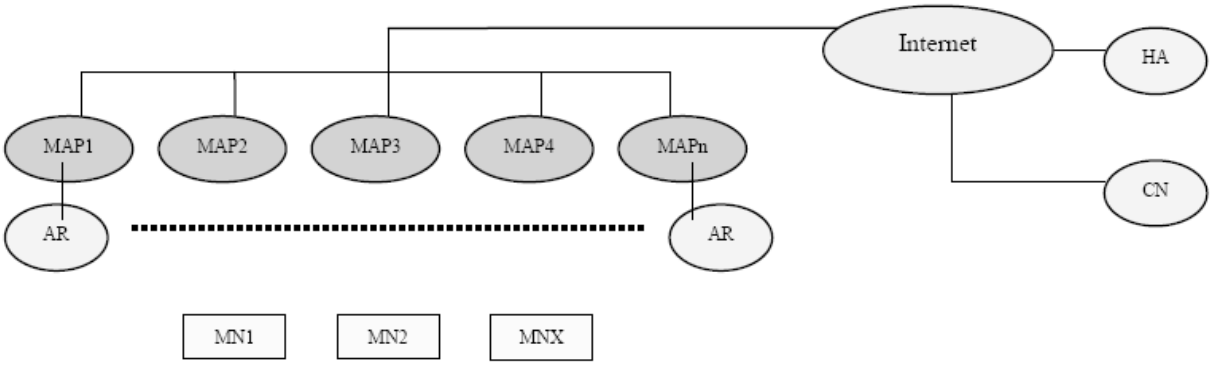


Figure 21. : Architecture de SHMIPv6

Chaque MAP attaché avec un routeur d'accès (AR) partage l'information de trafic entre l'ensemble des MAPs suivant une décision de sélection du MAP, l'adresse temporaire (RCoA) est la même dans tous les MAPs du domaine. [Ma'en 2007]

L'étude de SHMIPv6 montre que :

- L'utilisation de plusieurs MAPs dans le domaine augmente la vitesse effective de transfert de données.
- Le nombre des paquets perdus diminue dans le cas d'exécution d'un handoff.
- Le pourcentage des paquets perdus est diminué à chaque multiplicité de MAPs quel que soit le nombre des nœuds mobiles (MN) dans le domaine.

A première vue et suivant les avantages mentionnés, on peut dire que la solution de SHMIPv6 est bonne dans un environnement où le nombre des nœuds mobiles est très important dans le domaine, et ces nœuds exécutent un nombre important de handoff. Dans le cas où le nombre des nœuds mobiles est petit, l'utilisation de plusieurs MAPs dans le domaine devient irraisonnable à cause de la complexité de gérer les MAPs et l'implémentation coûteuse, au temps qu'un seul MAP peut gérer le trafic des messages transmis.

Dans le cas où le MN essaye de choisir un MAP dans le domaine visité parmi les autres, le MN doit prendre la décision selon sa vitesse et le temps d'occupation. Donc un algorithme de sélection du MAP est nécessaire pour assurer le bon choix du MAP, et de maintenir l'équilibre de charge.

Particulièrement, quand SHMIPv6 est implémenté dans un réseau de mobiles à grande échelle, il est important qu'un nœud mobile choisit le MAP le plus approprié parmi les MAPs disponibles, afin de réduire le coût total de signalisation. Le coût comprend le coût de mise à jour (BU) et le coût de transmission du paquet.

4.1.2 Architecture multi-niveaux

4.1.2.1 Architecture HMIPv6 à plusieurs niveaux

L'architecture HMIPv6 à multi-niveaux [Minji 2004] vise à réduire le nombre de messages de signalisation de et vers les réseaux extérieurs et à éviter le problème de surcharge de MAP en particulier. Dans cette architecture (figure 22), il existe multiple niveaux de MAPs (deux ou plus). Un nœud mobile choisit un niveau approprié de MAPs selon sa vitesse de mobilité. Le MN à grande vitesse effectue toujours plus de transferts, donc il doit choisir un MAP plus haut pour réduire le nombre de messages de mises à jour à son agent mère et CNs.

L'algorithme de sélection de MAP [Xu 2003] [Pack 2005] assure l'équilibre de charge entre les différents niveaux de MAP.

La vitesse d'un nœud mobile dans la zone d'accès précédente est calculée à partir du temps de séjour de MN dans la zone d'accès précédente, ainsi que la distance standard de la zone d'accès courante. Le temps relevé quand un MN entre dans la zone d'accès précédente et le moment où il entre dans la zone d'accès courante, sont employés pour calculer le temps de séjour. La distance standard est une distance constante définie à l'avance qui remplace la distance de déplacement réel d'un MN dans la zone d'accès. Pour réduire l'estimation des erreurs lorsque la vitesse d'un MN est calculée, la précédente vitesse calculée ($speed_{p\_h}$) est également considérée comme historiquement calculée.

La vitesse historiquement calculée ($speed_h$) est calculée comme suit :

$$speed_h = \alpha \times speed + (1-\alpha) + speed_{p\_h}$$

Où le paramètre α est une constante positive moins de 1.

Cependant, puisque la CN d'un MN affecte également le nombre de signalisation, le gain d'exécution de l'algorithme de sélection de MAP basé à la vitesse est limité.

Dans HMIPv6 avec hiérarchie de multiples niveaux [Minji 2004] [Pack 2006], (voir Figure 22) un message de signalisation ou de mise à jour *Binding Update* (BU) est envoyé au MAP de racine (RMAP) par un ou plusieurs MAPs intermédiaires (IMAPs). Quand le message de BU arrive à la première MAP, le MAP vérifie sa table de trace pour voir si le MN est déjà inscrit à lui ou pas. Si le MN est déjà enregistré dans la table de trace, le message de mise à jour locale est accompli au MAP. À savoir, dans ce cas, le MAP produit un message réponse de BU et envoie le message au prochain MAP plus bas dans la hiérarchie. Cependant, s'il n'est pas le cas, le MAP fait suivre le message de mise à jour BU au prochaine MAP de plus haut niveau. Ce processus est répété dans chaque MAP de la hiérarchie jusqu'à ce qu'un MAP ait le MN inscrit dans sa table de trace.

Quand le niveau hiérarchique multiple est employé, plusieurs MAPs peuvent être organisés sous forme d'arbre, de sorte qu'il soit possible de fournir des services scalable et de servir un plus grand nombre de MNs.

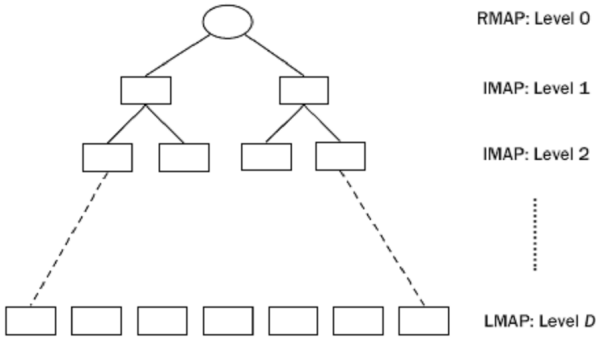


Figure 22. : Architecture abstraite pour HMIPv6 multi-niveaux

Dans le cas où quelques nœuds (c.-à-d., MAPs intermédiaires) sont tombés en panne, seulement les sous-arbres enracinés par ces noeuds sont affectés d'échecs. Par conséquent, le HMIPv6 à multi-niveaux peut être une solution plus fiable. Cependant, les résultats HMIPv6 à multi-niveaux en coût de traitement est plus élevé que HMIPv6 à un seul niveau quand un paquet est livré à un nœud mobile, parce que le paquet passe par plusieurs MAPs intermédiaires, les procédures d'encapsulation/décapsulation sont répétées à chaque MAP.

Donc, l'utilisation de plusieurs niveaux est inutile, car la taille du réseau devient très importante par le grand nombre de nœuds intermédiaires et les nœuds de réseaux s'occupent d'échanger et de traiter plus des paquets de signalisation et de contrôler que des paquets des données.

4.1.2.2 Architecture à trois niveaux

Dans cette proposition (voir Figure 23) [Zheng 2006], les agents de mobilité sont classés en trois niveaux : un MAP supérieur, un MAP inférieure, et des agents mères du noeuds mobiles (MNs), qui constituent une architecture à trois niveaux. Le concept de MAP à multi-niveaux a été adopté afin de tirer l'avantage de mobilité et de trafic de MNs afin de réduire le nombre de messages de mise à jour (BU) en dehors des domaines de MAP. D'ailleurs, le niveau de l'agent mère a été ajouté pour permettre la sélection adaptative entre le HMIPv6 et le MIPv6 standard afin que le MNs réalise la réduction de signalisation. Un autre avantage d'ajouter ce troisième niveau est que la capacité globale est augmentée. Lorsque les MAPs supérieurs et inférieurs atteignent leurs pleines capacités, les nouveaux MNs peuvent être servis par leurs agents mère.

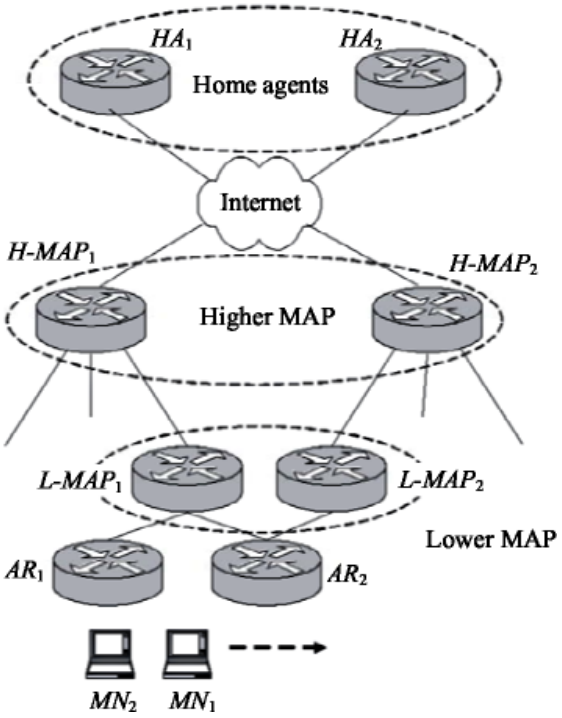


Figure 23. : L'architecture à trois niveaux

Extension de l'option de MAP [Zheng 2006]

L'option courante de MAP est prolongée pour exécuter l'opération de contrôle de charge dans l'architecture à trois niveaux proposée à la gestion de mobilité. La Figure 24 Montre le contenu de la nouvelle option de MAP, qui inclut les champs additionnels suivants :

(1) *level* : indiquer le niveau d'un MAP. « 1 » indique le plus haut MAP et « 0 » indique le MAP inférieure. Avec une longueur de 8 bits réservés au champ de niveau, représente au maximum 256 niveaux d'architecture hiérarchique.

(2) *Sat* : degré courant de saturation d'un MAP. Elle est définie comme rapport de tout le nombre de CNs qu'un MAP sert actuellement (*Nt*) au nombre maximum de CNs allouer (*Na*) qu'un MAP peut manipuler. Ce champ est employé pour équilibrer la charge de MAP d'un niveau, La valeur de Sat pour un MAP est calculée comme suit:

$$Sat = (Nt / Na) \times 255$$

(3) *Range* : distance standard des routeurs d'accès dans le domaine de MAP, qui est employée pour remplacer la distance réelle de mobilité d'un MN dans un domaine. Un MN utilise ce champ et son temps de séjour pour estimer sa vitesse de mobilité. Il est évident que dans le même domaine de MAP, tous les routeurs d'accès aient la même distance standard.

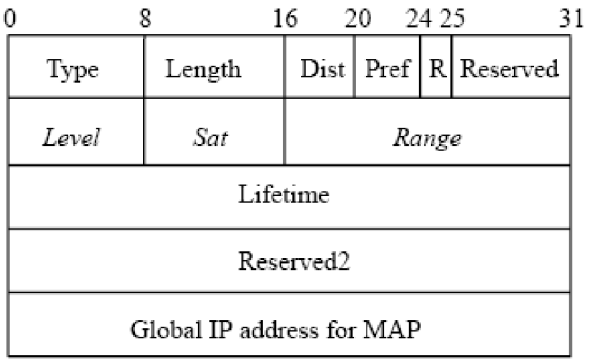


Figure 24. : le contenu de la nouvelle option de MAP

L'utilisation de trois niveaux est plus raisonnable dans une architecture multi-niveaux pour réduire la complexité de la hiérarchie. Dans cette architecture il existe toujours un algorithme de sélection de MAP suivant leur charge quelque soit le niveau. Le MN choisit un MAP selon sa vitesse de mobilité dans le niveau approprié.

L'ajout d'un troisième niveau des HAs rend la micromobilité des MNs explicite pour leurs HAs, la sélection adaptative entre MIPv6 et HMIPv6 à ce niveau complique la gestion de handover local, même si en profite de HA de servir les MNs quand les MAPs atteignent leurs pleines capacités.

4.2 Topologie proposée :

Notre solution s'appuie sur les solutions précédentes avec des modifications afin de cumuler les avantages et essayer d'éviter les inconvénients de ces solutions.

Particulièrement quand HMIPv6 est utilisé dans le réseau mobile à grande échelle, des multiples MAPs sont employés pour fournir des services mobiles d'Internet scalables et robustes. Dans de tels environnements, il est important qu'un nœud mobile choisisse le MAP le plus approprié parmi les MAPs disponibles.

L'utilisation de plusieurs MAPs (protocole SHMIPv6) augmente la capacité globale de protocole HMIPv6 standard. Dans cet environnement le MN sélectionne toujours un MAP de domaine suivant sa charge et sa disponibilité. Une architecture multi-niveaux des MAPs réduit le coût de sélection d'un MAP de domaine.

L'utilisation de plusieurs niveaux (plus de deux niveaux) accroît le coût de traitement des paquets de signalisation

L'idée (voir Figure 25), c'est d'organiser l'ensemble des MAPs (protocole SHMIPv6) en deux niveau (niveau 0 et niveau 1), le niveau N0 est le niveau inférieur, le niveau N1 est le niveau supérieur, les MAPs de N0 (MAP1N0, MAP2N0,…, MAPnN0) et les MAPs de N1 (MAP1N1, MAP2N1,…, MAPnN1), chaque MAP du niveau inférieur est attaché par un nombre fixe des ARs ;

Un sous-réseau de domaine regroupe un nombre fixe des MNs reliés au MAP local de sous-réseau avec ARs. Quand les MNs effectuent un mouvement local au niveau de sous-réseau (mouvement intra-domaine), les messages de signalisation sont transmis au MAP local de sous-réseau du niveau inférieur (N0), lorsque les MNs effectuent un mouvement entre les sous réseaux, l'enregistrement se fait au MAP local de niveau supérieur (N1).

Les MNs à grande vitesse de mouvement s'accordent avec les MAPs de niveau supérieur N1, les MNs à petite vitesse de mouvement reliés aux MAPs de niveau inférieur via les routeurs d'accès (ARs).

Le domaine regroupe des déférents sous-réseaux, si le MN change son domaine, il utilise le protocole Mobile IP pour sélectionner le MAP de niveau supérieur (N1) qui correspond au domaine visité.

Avantages :

- Plusieurs MAPs dans le domaine sont nécessaire pour la gestion de trafic au niveau du MAP et la réduction du temps de traitement des paquets ;
- Durée d'attente plus courte au niveau du MAP, contrairement à HMIPv6, un seul MAP par domaine cause une durée d'attente supérieure ;
- Le nombre de MAPs n'est pas très important comme dans le protocole SHMIPv6. il est fixé suivant la taille de réseau, la vitesse de mouvement et le nombre des nœuds mobiles ;
- Un nombre fixe de MAPs par domaine diminue le coût de sélection du MAP ;

- Le multi-niveau réduit le nombre de messages de signalisation et de mise à jour BU (Binding Update) hors des domaines du MAP;
- Chaque ensemble de MNs est relié à un niveau de MAPs bien défini selon la vitesse de mobilité.
- Réduire le nombre de paquets perdus ;

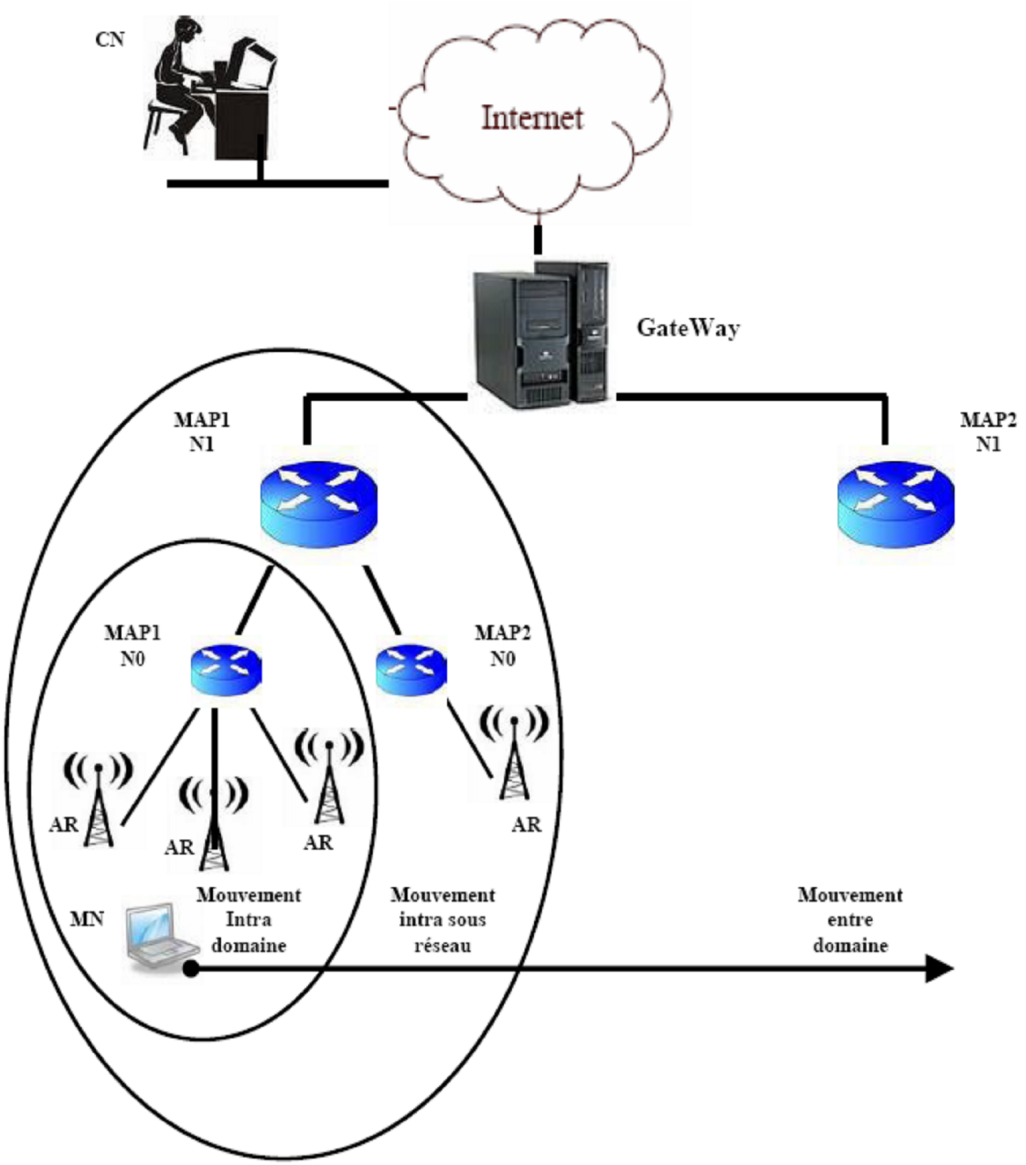


Figure 25. : Protocole HMIPv6 à deux niveaux HMIP2L

Conclusion :

Dans ce chapitre nous avons présenté les différentes solutions proposées pour améliorer le protocole de mobilité HMIPv6. Ces solutions présentent plusieurs avantages comme elles possèdent des inconvénients.

Nous avons essayé d'insister sur les avantages de ces solutions afin de proposer une solution optimale réalisable autour des ces travaux.

Notre solution propose une organisation hiérarchique multi-niveaux optimale (deux niveaux au maximum) de multiples MAPs pour diminuer le coût de sélection de MAP et de réduire le nombre de messages de signalisation et de mise à jour hors des domaines de MAP.

Chapitre 5 :

Simulation et analyse des résultats

Chapitre 5 : Simulation et Analyse des résultats

5.1 Introduction

Le simulateur réseau NS (Network Simulator) est un simulateur à événements discrets orienté objet, basé sur le simulateur réseau REAL [Nicolas 2001]. Au départ, la version 1.0 de NS a été développée au Laboratoire National de Lawrence Berkeley (LBNL) par le groupe de recherche réseau. Son développement fait maintenant partie du projet VINT sous lequel la version 2.0 est sortie.

Le projet VINT [VINT 1996]

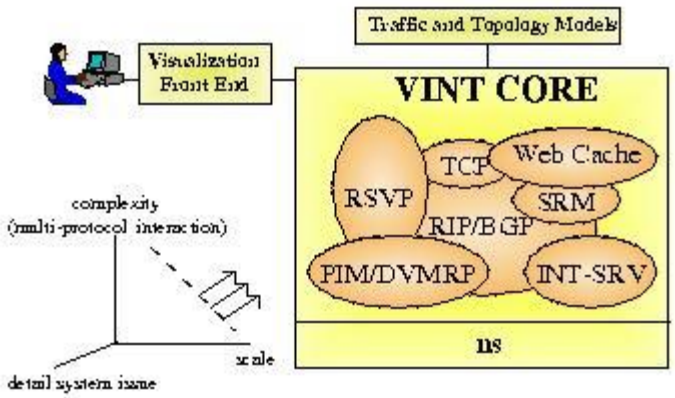


Figure 26. : projetVINT

Le projet VINT [Campbell 2000] (Virtual InterNetwork Testbed) est dirigé par l'université de Californie du sud et est financé par le DARPA en collaboration avec Xerox PARC et LBNL. Le but de ce projet est la construction d'un simulateur réseau qui offre des outils et des méthodes innovatrices dans un environnement proche de la réalité. Ce simulateur essaie de répondre aux questions de mise à l'échelle (simulation de grandes topologies) et d'interaction entre protocoles dans des services intégrés à l'Internet (problèmes d'hétérogénéité).

L'objectif n'est pas de concevoir un nouveau simulateur réseau, mais d'unifier les efforts de toutes les personnes travaillant dans le domaine de la simulation de réseau. La plupart des simulateurs réseau se focalisent seulement sur des protocoles simples et simulent des protocoles indépendamment. Ils ne se préoccupent pas des interactions avec d'autres composants de l'architecture. Le résultat est donc souvent limité, primitif et applicable uniquement à des cas particuliers. Ceci engendre aussi un manque de comparabilité dans chaque simulateur.

En outre, les divers composants d'un réseau (type d'application, charge, topologie...) sont de plus en plus impliqués dans les protocoles. A mesure que l'Internet devienne de plus en plus complexe en termes de mise à l'échelle, de nombre de protocoles et au niveau matériel et système d'exploitation, il est plus dur de concevoir ou de prouver des protocoles. Les coûts pour créer un tel simulateur sont si importants qu'ils sont au-delà de la portée de n'importe quelle société commerciale ou université.

Enfin, les simulateurs courants fournissent rarement des outils pour la visualisation et l'interprétation des résultats. Le projet VINT se fonde sur NS pour le simulateur et NAM pour la visualisation. Le projet VINT propose d'augmenter la synergie dans la communauté de simulation. Chaque apport d'un groupe de chercheurs doit pouvoir

profiter à tous et augmenter la structure initiale au fur et à mesure. Ce simulateur est notamment inspiré de NETSIM de MIT, de MARS de l'université du Maryland, de REAL de UC Berkeley, de NEST de la Colombie et NS-1 de LBNL.

Le projet VINT a pour objectif l'évaluation à la fois de la justesse et des performances de protocoles allant des protocoles de routage aux protocoles de transport et de session dans des grandes aires de réseaux Internet et ceci à tous les niveaux. Le projet s'est notamment concentré sur les points suivants :

- Repousser les capacités de mise à l'échelle aussi loin que possible.
- Offrir une simulation réseau composable pour modéliser la modularité de l'Internet et pour supporter des nouveaux modules de différents collaborateurs.
- Utiliser diverses techniques d'abstraction pour permettre la variation du niveau d'abstraction.
- Mettre en place des techniques de visualisation pour mieux interpréter les résultats, selon le niveau de granularité.
- Proposer une interface d'émulation pour permettre aux nœuds d'interagir avec le simulateur.
- Créer des bibliothèques vastes et extensibles de topologie de réseau et de générateurs de trafic.

Le projet VINT est utilisé par plusieurs groupes de travail comme l'IRTF ou l'IETF pour évaluer les protocoles en cours de normalisation. NS est ainsi couramment utilisé dans la communauté des chercheurs pour des comparaisons de simulation dans les publications.

NS-2 est bien adapté pour simuler la circulation de paquets dans des réseaux commutés. Il est principalement utilisé pour tester des algorithmes de file d'attente, des contrôles de congestion, des protocoles de transport, le multicast et la mobilité.

5.2 Architecture et implémentation

L'architecture réseau de NS-2 [NS2 2007] est fortement basée sur le modèle des couches OSI. Ce modèle a brièvement été expliqué dans la section 2.3.1. Il s'agit de la décomposition de la pile réseau en couches. Tout au long de cette section on retrouvera donc les éléments de ces couches avec plus ou moins de détails.

Les sources sont disponibles sur le site ISI dans la section «nsnam» sur le site Internet officiel [NS-2].

Les sources se présentent sous deux formes : l'une dite «tout en un ; All-in-One» qui contient le code NS-2 et d'autres composants utilisés (comme OTcl, NAM…), soit par morceaux, c'est-à-dire qu'on peut choisir uniquement les composants dont on a besoin. Le package comprend aussi des exemples de script ainsi que des modèles de mouvement pour les nœuds mobiles ou de génération de trafic.

NS-2 est un simulateur à évènements à temps discrets orienté objet. Il est développé en C++ et en OTcl; Le paquet inclue une hiérarchie de classe compilée d'objets écrits en C++ et une hiérarchie de classe interprétée d'objets écrits en OTcl. Ces deux hiérarchies sont étroitement liées ; quand l'utilisateur crée un nouvel objet par l'interpréteur OTcl, un objet correspondant appelé objet reflet est aussi crée dans la hiérarchie compilée. On dit que ces objets sont des « objets fendus ». Bien entendu, les objets peuvent être accédés aussi bien en OTcl qu'en C++ grâce à la mise en place de procédures d'appel entre OTcl et C++.

Le langage OTcl est un langage interprété qui ne demande pas de compilation. Il est principalement utilisé pour concaténer des objets, accéder aux objets à partir de l'interpréteur et configurer des simulations (début et arrêt des évènements, perte réseau,

rassemblement de statistiques). Son utilisation est rapide et assez conviviale. D'un autre côté, C++ est utilisé pour créer les classes de base et pour traiter un grand nombre de données (tel que calcul des tables de routage, mouvement des mobiles…).

5.2.1 Implémentation du simulateur

Au plus bas niveau, il y a six classes qui définissent l'ensemble de la structure du programme et fournissent les méthodes élémentaires [NS2 2007] [NS-2]. Il s'agit des classes *Tcl, TclObject, TclClass, TclCommand, EmbeddedTcl, InstVar*. Elles définissent entre autres les méthodes utilisées par C++ pour accéder à l'interpréteur, la hiérarchie, les principales commandes de haut niveau et les méthodes pour accéder aux variables C++ et OTcl.

La simulation est configurée, contrôlée et gérée à l'aide des interfaces fournies par la classe OTcl Simulator. Cette classe fournit des procédures pour créer et gérer la topologie, initialiser le format des paquets et choisir le planificateur d'évènements.

Elle stocke intérieurement des références à chaque élément de la topologie. Un script devra donc toujours commencer par l'instanciation d'une variable de cette classe. L'utilisateur crée ensuite la topologie à travers OTcl en utilisant les classes *node* et *link*, composants essentiels de la topologie.

5.2.1.1 Composants de la topologie

La topologie NS-2 [NS2 2007] [NS-2] est essentiellement composée de nœuds et de liens. La définition des nœuds se fait dans un premier temps à travers l'instance de Simulator puis à travers l'instance de la classe *Node*. La fonction d'un nœud est de recevoir des paquets, les examiner et les mapper à ses interfaces sortantes appropriées (voir Figure 27). Cette classe est composée d'un classificateur et de méthodes pour configurer un nœud. Les méthodes proposées sont des fonctions de contrôle, de gestion d'adresse et de port, de gestion d'agents et de repérage des voisins. Le classificateur est la partie du nœud qui traite chaque segment des paquets reçus. Il en existe donc plusieurs, chacun étant spécifique au champ examiné (ex : le classificateur d'adresse est utilisé pour supporter la propagation des paquets). Nous verrons plus loin l'implémentation de deux nœuds spéciaux (sous-classe de *Node*) qui permettent la mise en œuvre des sous-réseaux locaux de manière simple et la mise en œuvre de la mobilité.

Les liens constituent la deuxième partie de la topologie. Les liens entre les nœuds sont définis dans la classe *Link* et *SimpleLink* plus précisément lorsqu'il s'agit de relier deux nœuds. Plusieurs types de liaisons sont supportés, comme le point à point, le broadcast ou les liaisons sans fil pour la mobilité. Cette classe définie cinq variables clés (représentée dans la Figure 25) :

- La file d'attente de connecteurs
- La tête de file
- Le lien
- La durée de vie (TTL)
- L'élimination de la tête

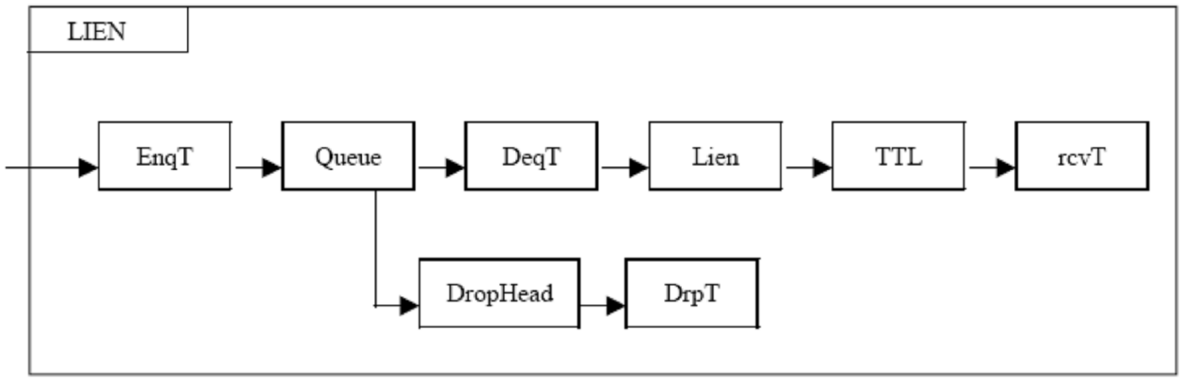


Figure 27. : Structure d'un lien

Note sur la Figure 27: *EnqT, DeqT, DrpT* et *rcvT* sont des procédures qui manipulent la file d'attente (enfilement, défilement, destruction et réception).

Une liaison est définie par une séquence de connecteurs (classe *Connector*) qui sont rangés dans une file d'attente. Ces connecteurs font suivre les paquets qui leur sont envoyés dans une seule direction ; Le paquet est alors délivré au voisin cible ou il est détruit.

A présent, voyons les structures mises en place autour de la topologie pour faire interagir les nœuds entre eux.

5.2.1.2 La gestion des files d'attente

La gestion des files d'attente et la simulation des délais sur les liens sont implémentés dans les classes Queue et *LinkDelay* respectivement. Les files d'attente actuellement disponible dans NS sont :

- FIFO
- RED buffer management
- CBQ (priorité et circulaire)
- Plusieurs variantes de file d'attente juste (Fair Queue)

Pour simuler un quelconque délai dans la réception ou l'émission d'un paquet, la file d'attente correspondante est simplement bloquée.

5.2.1.3 Les agents

A un niveau supérieur, on retrouve les agents (classe Agent) [NS2 2007] [NS-2] qui jouent un rôle important dans les simulations. Les utilisateurs créent de nouvelles sources ou récepteurs à partir de la classe Agent. Ils font partie intégrante d'un nœud et sont les points terminaux vis à vis des paquets couche réseau ; leur rôle est de générer et réceptionner des paquets suivant les protocoles de transport (TCP, UDP, RTP…). La Figure 28 montre les interactions entre ces différents composants dans NS.

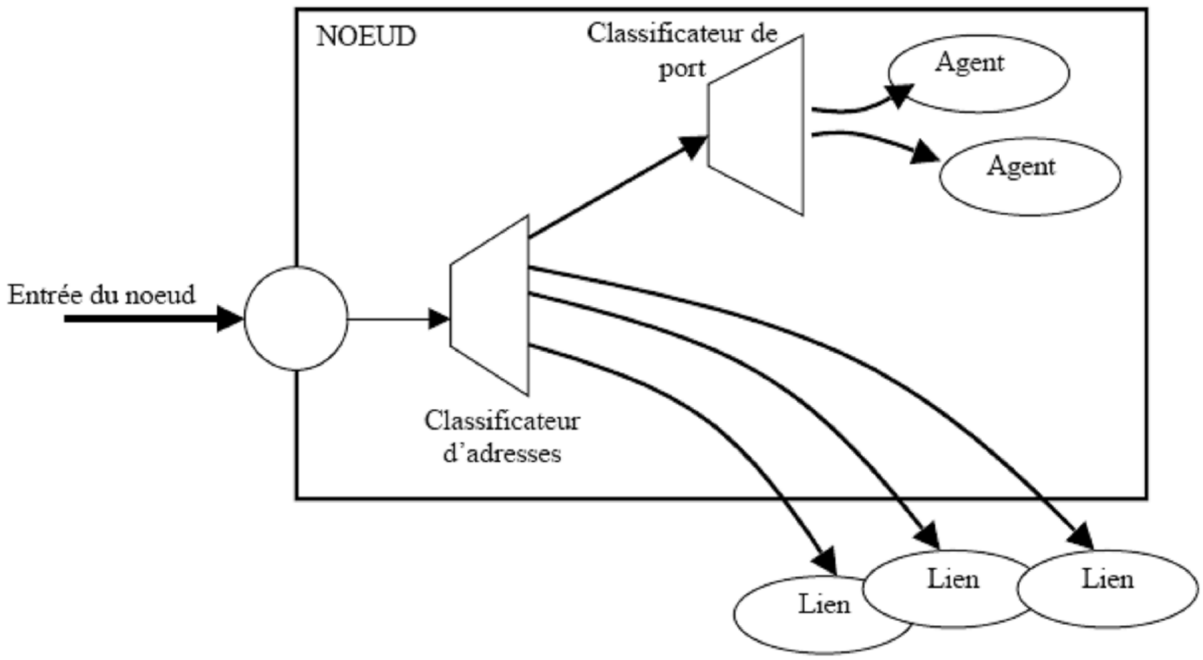


Figure 28. : Structure d'un nœud unicast

La génération de trafic dans NS peut se faire de deux manières différentes et est décrit dans la classe Application. Il est possible de générer des paquets par un générateur de trafic (classe *TrafficGenerator*) ou par la simulation d'applications existantes (classe prenant le nom de l'application), toutes ces classes étant dérivées dans la classe *Application*. Les générateurs de trafic peuvent être de quatre types :

- Classe *Exponential* : génère un trafic ON/OFF à intervalle de temps régulier.
- Classe *Pareto* : génère un trafic ON/OFF à intervalle de temps aléatoire.
- Classe *CBQ* : débit de bit constant.
- Classe *Trace* : permet de lire la génération de trafic dans un fichier.

Tous ces générateurs de trafic [NS2 2007] [NS-2] peuvent être associés au protocole de transport UDP. Actuellement, seules les applications existantes FTP, Telnet et récemment HTTP sont disponibles dans NS pour simuler des applications TCP. Le schéma entre ces deux méthodes de génération de trafic est présenté dans la Figure 29.

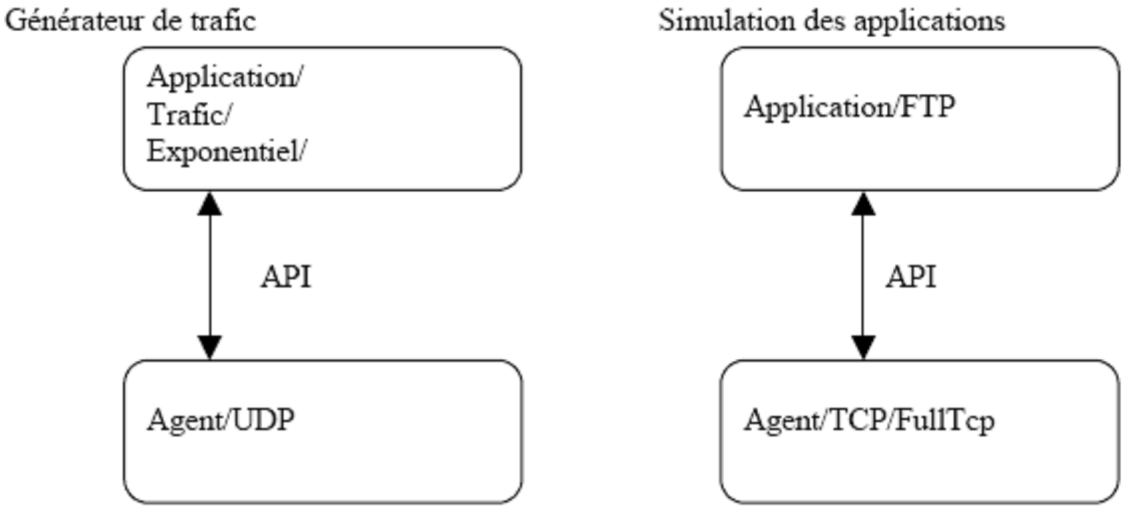


Figure 29. : Exemple de composition d'une application

La simulation de HTTP diffère quelque peu de FTP et Telnet ; Alors que FTP et Telnet simulent la circulation de données en donnant simplement la taille des paquets et le débit, HTTP échange réellement des données. Ceci peut par exemple servir à voir l'impact des caches pour le web. Des serveurs et clients HTTP ont donc été mis en place ainsi qu'un cache de proxy qui fonctionnent au-dessus d'UDP ou TCP.

5.2.1.4 Le routage

Lors d'une simulation, l'utilisateur doit spécifier un protocole de routage, c'est-à-dire indiquer au simulateur comment construire les routes entre les nœuds. NS offre trois types de routage dans un réseau filaire [NS2 2007] [NS-2]:

- Routage statique : routage utilisé par défaut suivant l'algorithme SPF de Dijkstra. Il est exécuté au début de la simulation une fois pour toutes.
- Routage session : routage identique au routage statique mais ré-exécute l'algorithme à chaque changement de topologie.
- Routage dynamique : une valeur est assignée à chaque route et un tableau stocke toutes les routes les plus courtes. Il est possible de faire du routage asymétrique ou multi-chemin. C'est le protocole à vecteur de distance (Distant Vector) qui est utilisé.

Il est possible d'indiquer le protocole de routage uniquement à un sous-ensemble des nœuds constituant la topologie (par défaut il s'applique à la totalité des nœuds). Il existe aussi une implémentation de protocoles de routage multicast (PIM-SM, PIM-DM) mais leur étude et leur utilisation ne font pas l'objet de ce rapport.

NS permet de provoquer des erreurs dans les simulations pour tester la robustesse des protocoles. Pour cela, il existe un modèle d'erreur implémenté dans la classe *ErrorModel* qui simule les erreurs au niveau liaison par l'envoie des paquets à des agents destructeurs.

L'unité d'erreur peut être spécifiée en terme de paquets, bits ou temps. Maintenant que les modules de base ont été présentés, nous allons voir comment l'on peut spécifier des caractéristiques au niveau des nœuds pour étoffer les simulations. Nous verrons alors comment simuler les réseaux locaux et les nœuds mobiles.

5.2.1.5 Les réseaux locaux (LAN)

Pour permettre des simulations à plus grande échelle, NS permet d'utiliser la notion de LAN [NS2 2007] [NS-2]. Cette nouvelle entité a été introduite en tant que nouveau type de nœud. Les composants d'un LAN sont la couche liaison, la couche MAC et la couche physique (voir Figure 30).

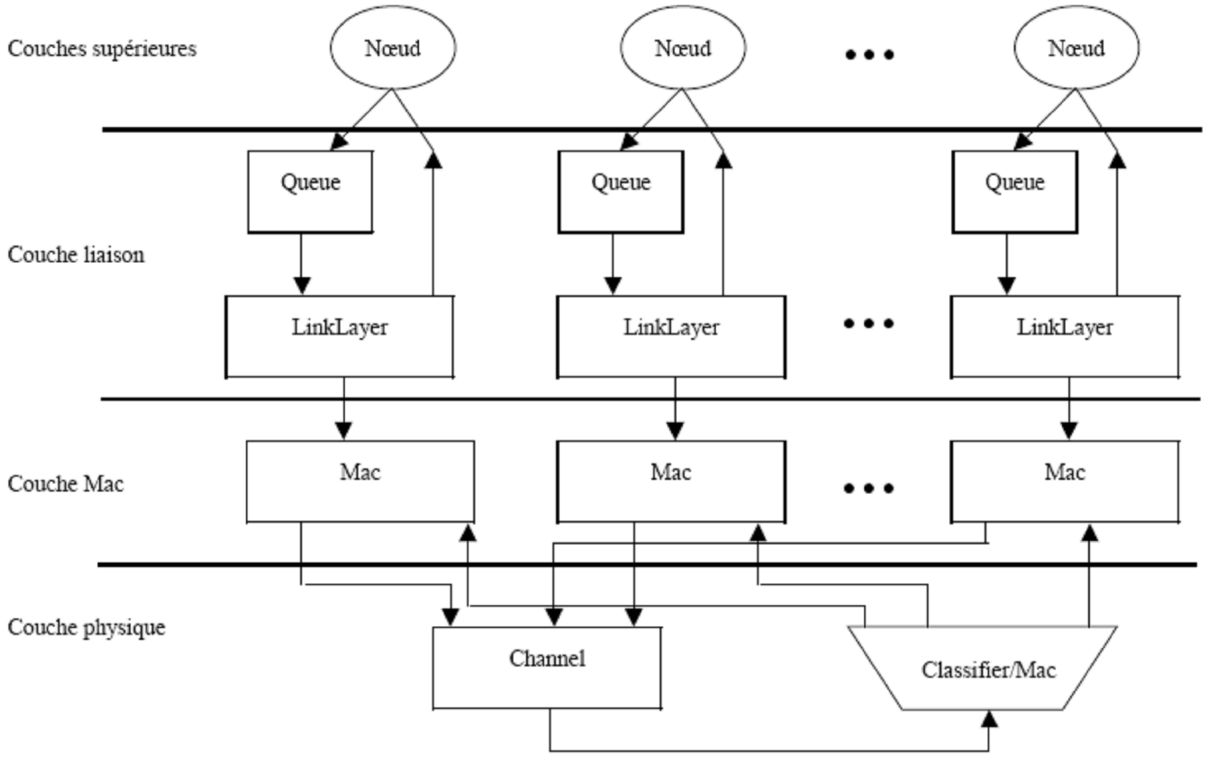


Figure 30. : structure d'un LAN

Au niveau de la couche physique, l'objet *Channel* simule le médium partagé et supporte des mécanismes d'accès au médium des objets *Mac* (phase d'émission). L'objet *Classifier/Mac* est responsable de la livraison et de la réplication des paquets pour des objets *Mac* (phase réception). Les détections de collisions se font au niveau de la couche MAC où sont implémentés les protocoles d'accès au médium (CSMA…). La couche liaison est composée de deux objets : Queue qui simule l'interface de file d'attente et *LinkLayer* qui implémente un protocole de couche de données. Un LAN contient en plus un seul objet *LanRouter* qui est crée automatiquement que le nœud LAN est initialisé. Tous les nœuds d'un LAN ont un pointeur sur cet objet qui joue le rôle de routeur.

5.2.2 La mobilité dans NS

Dans un premier temps, la mobilité a été introduite dans NS-2 par les chercheurs de l'université Cartegie Mellon de Pittsburgh (CMU) [NS2 2007] dans la volonté de simuler des réseaux ad hoc. Le lecteur doit particulièrement faire attention à la terminologie. Dans les deux premières parties, on a parlé de point d'accès comme étant un équipement de niveau 2 fournissant l'accès Internet aux nœuds mobiles. Dans la terminologie Ns-2, on parle de point d'accès non seulement pour spécifier un tel nœud mais aussi pour parler d'agent mère ou d'agent visité.

L'apport de la mobilité passe par l'ajout d'un nouveau type de nœuds définis dans la classe *MobileNode*, qui ne sont pas connectés entre eux. Les caractéristiques de la mobilité telles que le mouvement des nœuds, les mises à jour de localisation ou les limites de la topologie sont implémentées en C++. Par contre, les composants réseaux comme le nœud mobile lui-même (classificateur, couche liaison…) sont implémentés en OTcl.

Comme l'objectif était de simuler des réseaux entièrement mobiles, il a fallu mettre en place des protocoles de routage. Actuellement, il y a quatre protocoles de routage mis en œuvre dans NS-2 :

- Séquence de destination à vecteur de distance (DSDV) : des messages de routage sont échangés entre nœuds mobiles voisins. Les mises à jour peuvent être déclenchées (par une information sur un nœud voisin qui change la table de routage) ou périodique. Tout paquet à destination d'un nœud inconnu est mis en buffer pendant que des *Query* sont envoyés.
- Routage par source dynamique (DSR) : utilise un objet particulier de la classe *SRNode*. Toutes les entrées de l'objet SRNode pointent sur cet agent de routage. Ce modèle s'avère intéressant pour une future implémentation de protocole utilisant le piggy-backing.
- Algorithme de routage ordonné temporaire (TORA) : protocole de routage distribué basé sur l'algorithme « Link Reversal ». A chaque nœud, une copie séparée de TORA tourne pour chaque destination. TORA opère au- dessus de IMEP qui procure la liaison des messages de routage et informe le protocole de routage des changements des liens aux voisins à travers l'émission de beacons.
- Vecteur de distance sur demande (AODV) : combinaison des protocoles DSR pour la découverte et la maintenance des routes et DSDV pour le routage saut par saut, numéro de séquence et l'algorithme des beacons.

Lorsqu'un un nœud mobile est crée dans une simulation, le simulateur crée un objet *MobileNode*, un agent de routage et la pile réseau qui sera décrite plus loin. Ensuite ces composants sont interconnectés et la pile est connectée au canal. Ces composants sont illustrés dans la Figure 31 [NS2 2007].

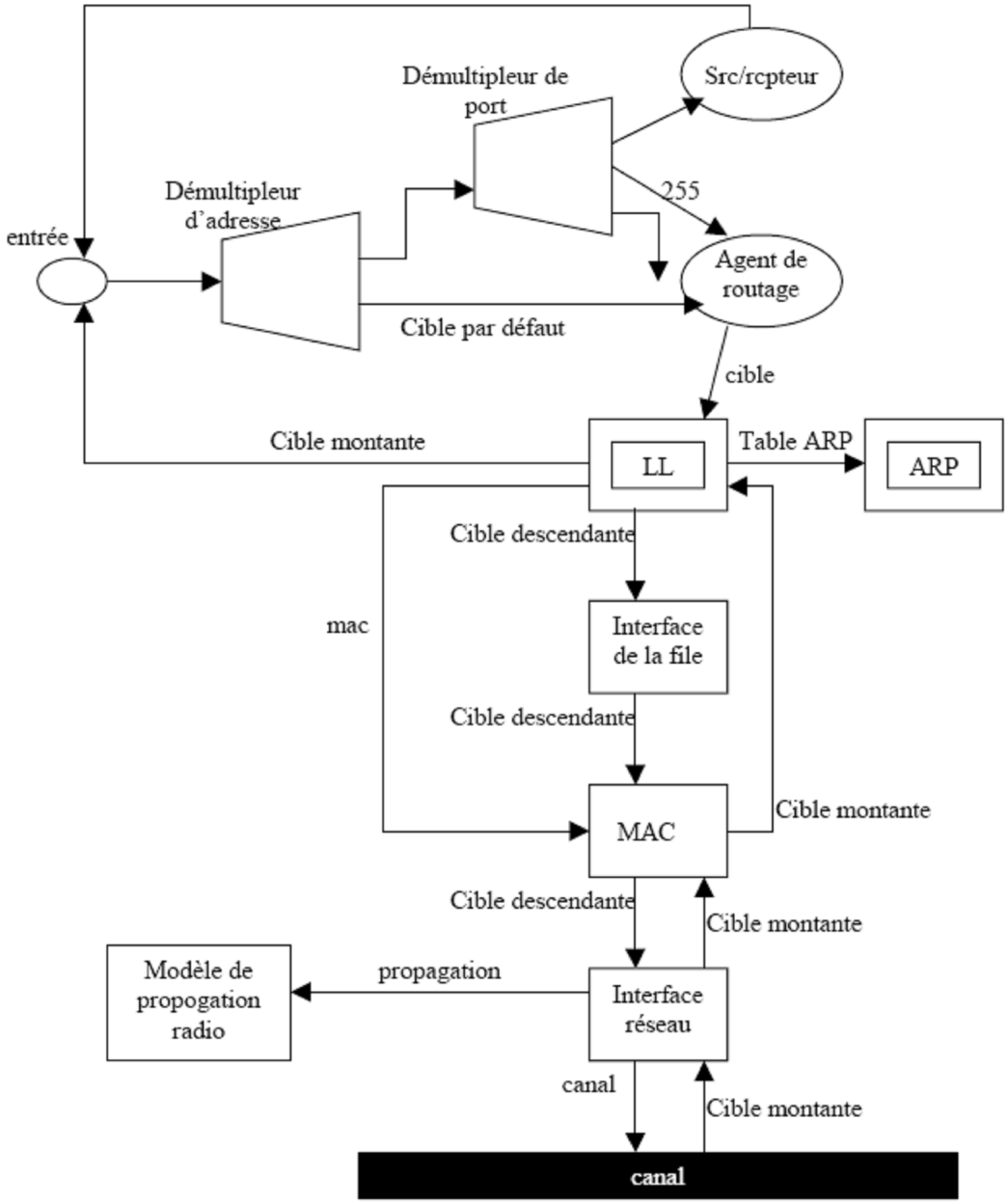


Figure 31. : Composants d'un nœud mobile (sauf pour DSR)

La Figure 31 vaut pour tous les protocoles de routage excepté pour DSR. Lorsque ce dernier est utilisé, les fonctionnalités du nœud mobile sont différentes ; Tous les paquets reçus par le nœud mobile sont dirigés vers l'agent DSR. C'est l'objet *SRNode*, dérivé de *MobileNode*, qui réalise cette redirection. Cet objet n'utilise pas de dé multiplexeur d'adresses ou de classificateur.

Une caractéristique forte des nœuds mobiles est bien sûr de pouvoir se déplacer. NS-2 a été conçu pour exécuter des déplacements en 3D, mais actuellement la troisième dimension n'est pas utilisée (Z=0). Il existe deux mécanismes pour l'utilisateur pour donner du mouvement aux nœuds mobiles :

- Indiquer le point d'origine, la destination et la vitesse explicitement pour chaque nœud mobile. Les mises à jour sont déclenchées chaque fois que l'on exige la position du nœud mobile à un moment donné. Cette solution est plutôt faite pour des petites simulations.
- Générer des mouvements aléatoires : à l'appel d'une procédure, le nœud mobile démarre à partir d'une position aléatoire et exécute des déplacements.

- Le nœud mobile exécute des mises à jour de routage pour changer de destination et de vitesse.
- Indépendamment des méthodes utilisées pour générer les mouvements des nœuds mobiles, il faut définir une topographie ; L'espace est considéré comme étant une grille dont il faut donner les frontières (valeurs de x abscisse
- et y ordonnée).

Voyons à présent plus en détail les composants réseaux d'un nœud mobile :

- L'objet *LinkLayer* est le même que celui utilisé par les nœuds unicast, avec un module ARP en plus. Ce module est implémenté dans le style BSD. Il traite des demandes de l'objet *LinkLayer* en utilisant une table de correspondance entre l'adresse IP et l'adresse MAC. Il se sert d'un buffer pour mettre les paquets pour lesquels il ne connaît pas de correspondance.
- L'interface de file d'attente a été augmentée d'un nouvel algorithme de priorité : *PriQueue*. Cette gestion de file d'attente donne priorité aux paquets de routage.
- La couche MAC implémente uniquement la norme 802.11 pour les nœuds mobiles. D'autres normes sont en train d'émerger mais ne font pas partie de la distribution de NS-2 pour le moment. Par exemple, IMB26 a développé une extension Bluehoc qui implémente la norme Bluetooth dans la version 6 de NS-2.
- On arrive aux interfaces réseaux. L'objectif est de simuler l'interface matérielle qu'un nœud mobile utilise pour accéder au canal. Ces interfaces sont implémentées dans les fichiers *wirelessPhy*.{cc,h}. Elles sont soumises aux collisions et enregistrent l'intensité du signal, la longueur d'onde…

Les limitations de ce modèle se sont vite fait ressentir. Effectivement, le modèle original de la mobilité permet des simulations de réseaux locaux sans fil et de réseaux ad hoc uniquement. C'est pourquoi une première extension a été ajoutée au modèle, toujours

fondée sur le travail des chercheurs de CMU [NS2 2007], qui permet de faire des simulations impliquant des nœuds filaires et des nœuds sans fil à la fois. Cette extension, appelée « ***wired-cum-wireless*** », utilise le modèle de base de la mobilité décrit ci-dessus.

L'objectif est donc de relier des réseaux locaux sans fil par un réseau filaire. Il se pose immédiatement un problème pour le routage ; Le routage des nœuds filaires se fait d'après la topologie grâce au concept de liaison, alors que dans la mobilité le concept de liaison n'existe pas. Un nouveau type de nœud est alors crée pour assurer la liaison entre le réseau filaire et le réseau sans fil. Ce nœud est appelé *BaseStationNode* (représenté dans la Figure 30). Ce nœud est un hybride entre un nœud hiérarchique et un nœud mobile. L'introduction de tel point d'accès a un impact sur l'adressage. Chaque domaine de nœuds mobiles a une adresse unique de domaine et un domaine est défini comme l'ensemble des nœuds mobiles qui sont attachés au point d'accès du domaine. Les nœuds mobiles doivent donc supporter le routage hiérarchique (excepté le *SRNode*). Un paquet destiné à un correspondant situé en dehors du domaine sans fil sera transmis au point d'accès, si toute fois il existe un chemin jusqu'à celui-ci. La hiérarchie de domaine est montrée dans la Figure 32.

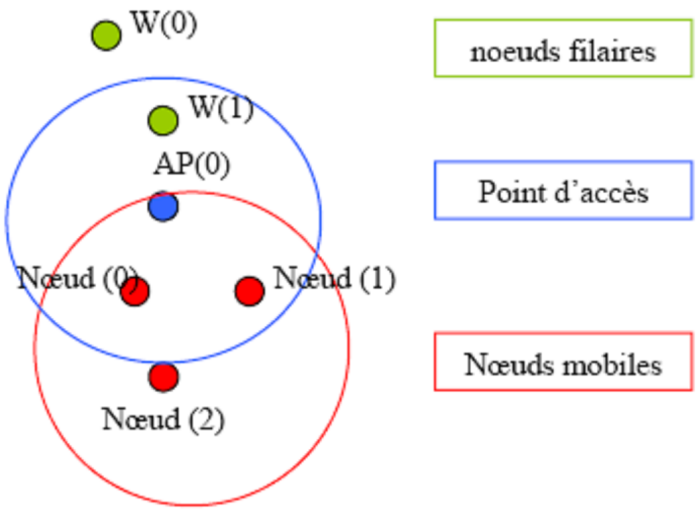


Figure 32. : architecture d'un scenario"wired-cum-wireless"

L'adressage hiérarchique fonctionne donc de la manière suivante : il est composé de trois niveaux et noté ainsi : 1.0.1. le premier chiffre indique le domaine, le deuxième indique le cluster et le dernier est l'identifiant du nœud. Prochainement, il est prévu d'étendre le niveau hiérarchique d'adressage à n, c'est-à-dire laisser l'utilisateur choisir le nombre de niveau suivant les besoins de sa simulation.

Une autre extension a été ajoutée dans NS-2 pour supporter l'implémentation de Sun Microsystem28 de Mobile IP. Cette extension est uniquement basée sur le modèle des nœuds filaires et non sur le modèle de la mobilité fait par CMU. Le scénario de Mobile IP consiste en des agents mère, des agents visités et des hôtes mobiles qui se déplacent de l'un à l'autre, comme présenté dans le chapitre de mobilité. Les agents mères et les agents visités sont grossièrement des points d'accès comme ceux décrits plus haut. Ils sont définis dans *MobileNode/MIPBS*. Ils contiennent un agent d'enregistrement qui envoie les beacons et effectue l'encapsulation et la décapsulation des paquets. Leur structure est décrite dans la Figure 33. L'hôte mobile est défini dans *MobileNode/MIPMH* qui a aussi un agent d'enregistrement qui réceptionne et émet des beacons. Leur structure est la même que celle décrite dans la Figure 31, sans l'encapsuleur et le décapsuleur.

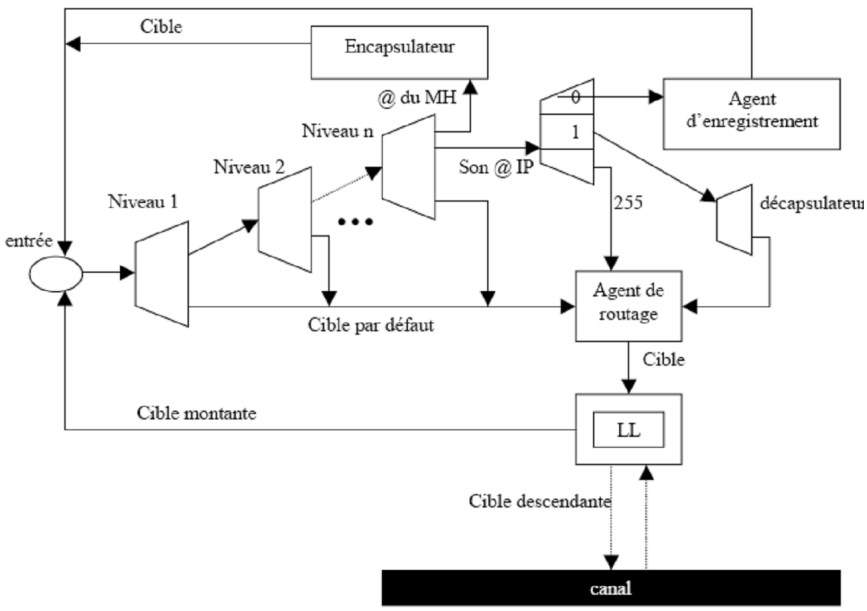


Figure 33. : structure d'un nœud point d'attache pour MIP

Pour s'apercevoir de la portée et du niveau de simulation possible à faire, voici la liste des configurations possibles pour un nœud mobile :

- Type d'adressage : plat ou hiérarchique
- Type du routage ad hoc
- Type de l'objet *LinkLayer*
- Type du protocole utilisé par la couche MAC
- Type de propagation radio
- Interface de la file d'attente
- Type de support physique
- Type de l'antenne
- Type du canal
- Activer ou non les fonctionnalités de routage filaire
- Activer ou non le protocole MIP sur le nœud
- Type de modèle d'énergie / énergie initiale
- Activer ou non un agent de trace sur le nœud
- Activer ou non un agent de trace de routage sur le nœud
- Activer ou non les traces des mouvements des mobiles

Pour le moment, certains de ces paramètres ne peuvent prendre qu'une seule valeur. Par exemple, la couche MAC n'est implémentée que par le protocole 802.11 dans NS-2 de base.

5.3 Statistiques et visualisation

NS-2 fournit plusieurs types de support pour analyser les résultats d'une simulation [NS2 2007]. D'une part, NS-2 inclus des classes pour suivre à la trace les fluctuations des paquets, pour calculer et enregistrer diverses statistiques sur l'ensemble des paquets ou uniquement pour un certain flux. L'utilisateur a le choix de mettre en place ce système de suivi pour chaque simulation. Ce système de suivi est présenté dans la sous-section suivante. D'autre part, NS-2 travaille de paire avec l'outil de visualisation NAM qui permet de visualiser l'ensemble de la topologie dans une fenêtre graphique.

5.3.1 Système de suivi

NS-2 propose deux types de monitoring :

- Traceur (*Trace*) : enregistre chaque paquet (arrivée, départ ou suppression) sur un lien ou dans une file d'attente. Ces objets sont configurés dans la simulation comme des nœuds dans la topologie de réseau. Ils sont définis dans plusieurs sous-classes pour des évènements précis.
- Moniteur (*Monitor*) : enregistre le décompte de différentes quantités comme le nombre d'arrivée de paquets, nombre de bit… Il traque ainsi la dynamique des paquets dans une file d'attente en faisant des moyennes. Quand un paquet arrive ou quitte un lien, il est passé à un objet spécial qui contient une référence sur l'objet *QueueMonitor*. Le contrôle basé sur un certain flux de paquets et non sur l'ensemble, fonctionne avec des classes

spécialisées. L'inspection et le mapping du flux sont réalisés par un objet classificateur. Le paquet passe ensuite par le moniteur de flux qui enregistre les états pour chaque flux.

Pour illustrer le suivi des paquets, nous allons voir un fichier de sortie d'une simulation. L'utilisateur peut demander au simulateur d'enregistrer chaque déplacement des paquets dans un fichier de sortie. Ce fichier sera formaté de la manière suivante :

		Nœuds					ID	Adresses		N°	
Action	Temps	Source	Dest.	Paquet	Taille	Flag	flux	Source	Dest.	de séq.	uid
+	1.84375	0	2	cbr	210	---	0	0.0	3.1	225	610
-	1.84375	0	2	cbr	210	---	0	0.0	3.1	225	610
R	1.84471	2	1	cbr	210	---	1	3.0	1.0	195	600
R	1.84566	2	0	ack	40	---	2	3.2	0.1	82	602
+	1.84566	0	2	tcp	1000	---	2	0.1	3.2	102	611
-	1.84566	0	2	tcp	1000	---	2	0.0	3.2	102	611
R	1.84609	0	2	cbr	210	---	0	0.0	3.1	225	610
+	1.84609	2	3	cbr	210	---	0	0.0	3.1	225	610
D	1.84609	2	3	cbr	210	---	0	0.0	3.1	225	610
-	1.8461	2	3	cbr	210	---	0	3.0	3.1	192	511
R	1.84612	3	2	cbr	210	---	1	3.0	1.0	196	603
+	1.84612	2	1	cbr	210	---	1	3.0	1.0	196	603
-	1.84612	2	1	cbr	210	---	1	3.0	1.0	196	603
+	1.84625	3	2	cbr	210	---	1	3.0	1.0	199	612

Tableau 1 : format du fichier de sortie .tr

Le Tableau 1 ci-dessus présente 14 entrées de trace de paquets, dont cinq opérations de mise en file (indiqué par "+" dans la première colonne), quatre opérations de défilement (indiqué par "-"), quatre événements de réception (indiqué par "r") et un événement de suppression ("d"). Le temps simulé (en secondes) auquel chaque événement est arrivé est inscrit dans la deuxième colonne. Les deux champs suivants indiquent les deux nœuds entre lesquels le paquet circule. Vient ensuite un nom descriptif pour le type de paquet, suivi de sa taille codée dans son en-tête IP. Le champ « flag » contient des flags qui ne sont pas utilisés ici.

Le champ d'après donne l'identificateur de flux IP. Les deux champs suivants indiquent les adresses source et destination du paquet, respectivement, comme utilisées dans NS-2. Puis il y a le numéro de séquence et un identificateur de paquet unique. Chaque nouveau paquet créé dans la simulation est assigné un nouvel identificateur unique.

Ce type de fichier de sortie peut bien entendu être utilisé pour tracer des courbes. Il est possible de demander au simulateur de ne répertorier que les paquets d'un certain type, par exemple que les paquets de contrôle appartenant au protocole pour une meilleure lisibilité.

Le support des traces pour la mobilité utilisait dans un premier temps les objets cmu-trace. Ce format de fichier est très proche de celui que nous venons de voir, bien qu'il présente quelques champs supplémentaires. Ces champs supplémentaires concernent

surtout des informations MAC (identifiant MAC des nœuds, temps attendu avant d'émettre sur le médium…) et les protocoles utilisés pour la mobilité.

Dans un effort de regrouper tous les formats de trace de simulation sans fil qui ont émergés et pour être le plus complet possible, un nouveau format de fichier de sortie a été mis en place. Ce nouveau format n'est valide que pour les simulations sans fil, il sera étendu par la suite à l'ensemble des simulations. Il est toutefois compatible avec l'ancien format. Le nouveau format est décrit dans le tableau suivant :

Ce format de fichier semble plus complexe, mais en fait il est beaucoup plus facile à lire que le précédent. Effectivement, chaque valeur notée dans le fichier est précédée de sa signification. En plus des informations contenues dans l'ancien format (type d'événement, temps, source, récepteur…) il contient des informations sur :

- Sur la position des nœuds (–Nx, -Ny, -Nz)
- Sur l'énergie des nœuds (-Ne)
- Sur le prochain saut (-Hs, -Hd)
- Au niveau MAC : type éthernet, adresses éthernet…
- An niveau applicatif : type de l'application, type du protocole, caractéristiques particulières suivant ces types

Ce nouveau format est donc beaucoup plus complet que l'ancien.

5.3.2 NAM

La conception de protocole demande une compréhension de plusieurs détails, dont le suivi des états d'un grand nombre de nœuds, une analyse de l'échange de messages et doit caractériser les interactions dynamiques pour des trafics concurrents. Habituellement, des traces de paquets sont utilisées pour accomplir ces tâches. Cependant, ces traces ont deux inconvénients majeurs : elles présentent un nombre important de détails, ce qui peut compliquer la compréhension des données, et elles sont statiques, ce qui cache une dimension importante du comportement des protocoles. Les outils de visualisation adressent ce problème en permettant à l'utilisateur de prendre en considération plusieurs informations très rapidement, d'identifier visuellement les modèles de communication et de mieux comprendre les interactions et les causalités.

NAM[1] [NS2 2007] est un outil d'animation basé sur Tcl/TK pour l'observation des traces de paquet. Il peut être installé sur un système de type unix ou sur Windows 95/98/NT ayant Microsoft Visual C++ installé. Les données utilisées par NAM peuvent provenir d'un simulateur ou de tests sur des réseaux réels (par exemple : utilisation de tcpdump). Il supporte l'affichage de la topologie, l'animation des échanges de paquets et des outils d'inspection de données divers. NAM a été crée par le laboratoire LBL et s'est considérablement développé durant les dernières années. Le développement de NAM est en collaboration avec le projet VINT. Actuellement, il est développé à ISI dans les projets CONSER et SAMAN.

NAM interprète un fichier de trace contenant des évènements réseau indexés par le temps de différentes manières. Ces évènements sont principalement les arrivées, départs et suppression de paquets, rupture de lien. Pour les simulations de réseau sans fil, la localisation et les mouvements des nœuds s'ajoutent aux évènements interprétés.

[1] NAM : Network Animator http://www.isi.edu/nsnam/nam/index.html

5.4 Extension NS-2 pour la mobilité

De nombreux laboratoires de recherche emploient NS-2 pour tester la réaction de nouveaux protocoles dans divers cas de figure. Dans cette section, trois extensions introduites par l'université de Mannheim, l'université de Colombie et l'INRIA respectivement seront présentés. Il s'agit généralement de fichier modifié ou de nouveaux fichiers introduits dans une version de NS-2. Le code est gratuit et disponible sur les sites respectifs.

5.4.1 NOAH

NOAH[1] (no-ad-hoc) joue un rôle important dans la gestion de la mobilité dans NS-2. Effectivement, l'extension de mobilité CIMS présentée ci-dessous utilise l'agent de routage NOAH. Cette extension a été implémentée par Jörg Widmer du laboratoire AT&T31 ACIRI32 à Berkeley pour NS-2 version 6 (ns-2.1b6) ou 7 (ns-2.1b7).

NOAH est un nouvel agent de routage sans fil qui supporte uniquement la communication entre les points d'accès et les nœuds mobiles (en contraste avec les agents DSDV, DSR…. Cet agent permet de faire des simulations dans lesquelles le routage multi-sauts entre les nœuds mobiles n'est pas désiré. En plus, l'agent NOAH n'envoie pas de paquets de routage. Cette extension consiste donc en l'amélioration de l'implémentation de Mobile IP existante dans NS-2 par le chevauchement des aires de couverture des points d'accès, la sélection intelligente des agents visités, l'amélioration du processus de handoff. Cette extension inclut en plus un modèle simple de propagation de distance : quand les paramètres du modèle de propagation radio CMU ne sont pas disponibles (qualité de réception…), le modèle simple de propagation de distance permet de spécifier la portée des points d'accès comme une distance (pas d'atténuation du signal). Par contre, quand l'information est disponible, le modèle exact sera utilisé.

Bien que cet agent soit très utilisé, la documentation sur cette extension n'est pas très explicite. D'après une brève étude du code, quelques fichiers ont été modifiés (sdist.{cc,h}, wireless-phy.{cc,h}, cmu-trace.cc, mip.h, mip-reg.cc).

5.4.2 CIMS

Columbia IP Micro-Mobility Suite (CIMS)[2] est une extension de NS-2 basée sur les versions 6 (ns-2.1b6) ou 7 (ns-2.1b7)

L'extension est disponible en deux versions, selon que l'agent NOAH décrit ci-dessus est déjà installé ou non. Elle a été développée par le groupe COMET de l'université de Colombie, en collaboration avec, laboratoires et entreprises.

CIMS v1.0 inclus les implémentations NS-2 de Cellular IP, Hawaii et Mobile IP Hiérarchique. L'implémentation de Cellular IP supporte le semi-soft handoff et la pagination IP. L'implémentation de Hawaii supporte les modèles de non-propagation unicast et la propagation de flux multiples. Pour l'instant, le pagination IP de Hawaii n'est pas supportée dans cette version de CIMS. De même, l'implémentation de MIP Hiérarchique ne supporte pas la pagination IP non plus.

[1] NOAH : http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/

[2] CIMS : http://www.comet.columbia.edu/micromobility

5.4.3 IST-CIMS

Le code développé de la mobilité a été établi sur la version 2.31 de Network simulator NS-2, qui vise la recherche de gestion de réseau et fournit un support substantiel pour la simulation du TCP, le routage, et les protocoles multicast de réseaux filaire, sans fil (local et satellite) et hybrides. Ce simulateur et pendant les dernières années a été maintenu par le groupe de l'institut de sciences de l'information (ISI) de l'université de Southern California.

L'extension de IST-CIMS [Estrela 2007] inclus dans le noyau de simulateur que seulement le protocole MIP de base ; tous autres protocoles et modules sont maintenus en tant que contributions externes, et qui sont typiquement attachées à la version particulière de NS2 sur lequel le code a été développé.

Concernant le module contribué par la thèse recherche de PhD de Pedro Estrela[1], c'est une extension de la version 2.31 du simulateur de base avec le modèle d'eTIMIP[2], caractéristiques de la mobilité des noeuds mobiles, le routage de base, le routage optimisé, les passerelles d'accès multiples de réseaux.

Le code inclut est une implémentation originale du protocole TIMIP, et des implémentations des protocoles de CIP, HAWAÏ et de HMIP. Ces réalisations ont été dérivées de CIMS v1.0, qui est développée pour la version NS2.1b7.

Tous les protocoles emploient le scénario NS-2 filaire-sans-fil hybride qui supporte des stations de base hybrides, composé d'interfaces filaires et sans fil. Dans ce scénario tous les noeuds sans fil doivent toujours avoir un agent de mobilité ad-hoc, ces protocoles emploient le module « no-ad-hoc » (NOAH) qui émule un simple infra-structure de l'interface 802.11 qui supporte soft-handovers.

Ce module générique, qui a été également amélioré dans la version NS2-26, a été également personnalisé pour simuler le fonctionnement de l'interface 802.11 avec des canaux multiples. Le code inclut également un modèle générique d'un récepteur UDP, prolongé et compatible avec le standard.

5.5 Scénario de la simulation du modèle proposé

Le scénario de la simulation est une description du modèle avec le langage tcl. Le script développé doit être exécuté dans le simulateur de réseau NS-2. Après que NS-2 interprète le script, il génère deux fichiers de trace : un fichier d'animation NAM et un fichier de trace normal (.tr). Le fichier de trace NAM est utilisé pour la visualisation de l'animation avec NAM. Le fichier de trace normal doit être interprété par un script AWK pour dessiner des graphes en utilisant des outils de traçage graphique comme Gnuplot, XGraph. La figure 34 suivante montre les déférentes étapes de la simulation pour un modèle donné.

[1] Perdro estrela Home page http://tagus.inesc-id.pt/~pestrela/ns2/

[2] TIMIP Homepage", http://tagus.inesc-id.pt/~pestrela/timip

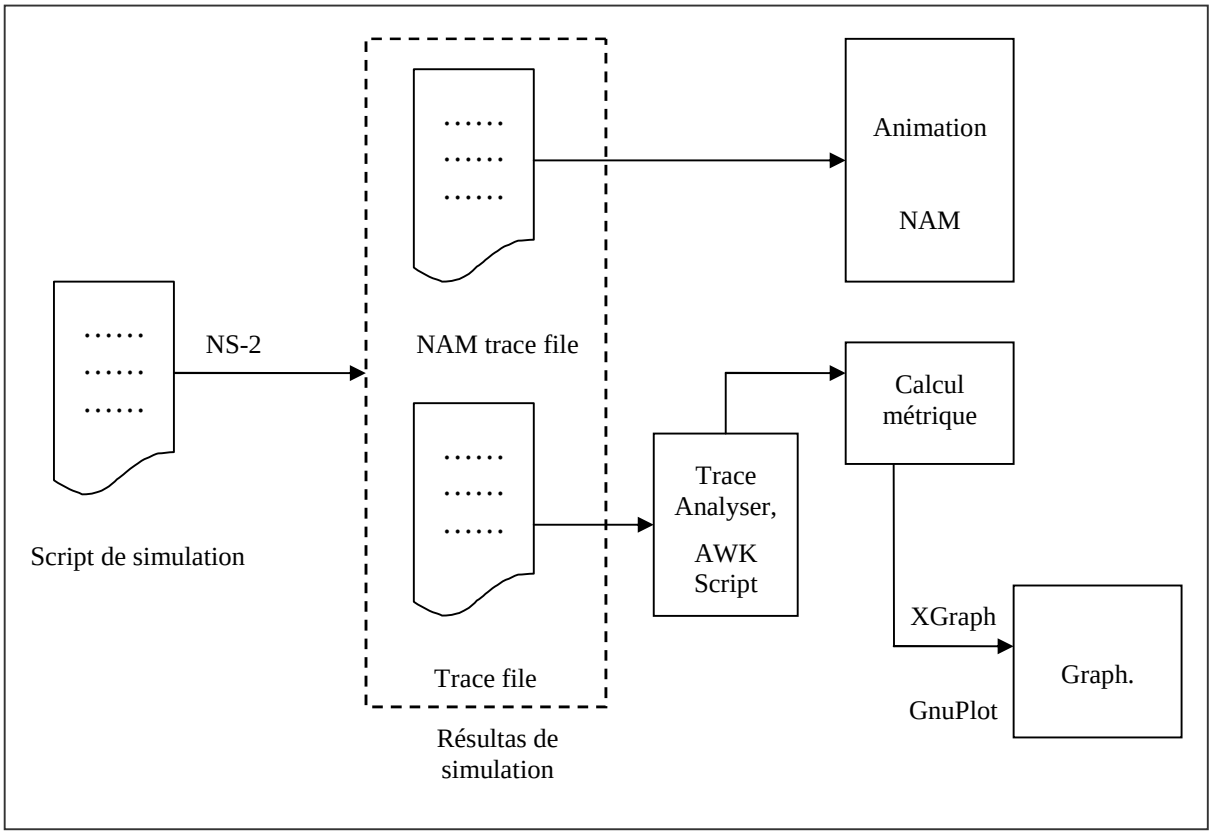


Figure 34. : Processus de simulation

5.5.1 Environnement de simulation

Le package ns-allinone-2.31 utilisé, c'est La version la plus récente de NS-2 qui supporte les protocoles de mobilité et intègre le protocole de routage (NOAH). Cette version de NS-2 patcheé par Pedro Estrela afin de supporter l'extension de mobilité IST-CIMS.

Linux mandriva 2007 c'est le système d'exploitation utilisé pour faire fonctionner NS-2. Des outils comme awk, gnuplot, xgraph utilisés dans la phase de l'analyse des résultats de simulation, ils seront présentés ultérieurement.

5.5.2 Génération de la topologie avec NS-2

Pour étudier la performance de notre proposition présentée dans le chapitre 4 ; nous avons développé un scénario sous NS-2. Le scénario montre un nœud correspondant CN qui émet des paquets vers le nœud mobile MN, via un réseau. L'infrastructure de ce réseau utilise un nœud comme agent étranger Gateway FA et 5 nœuds jouent le rôle des MAPs pour gérer la mobilité locale de domaine ; les MAPs sont regroupées et organisés en deux niveaux : les MAP1n1, MAP2n1 sont les MAPs de niveau supérieur ; les MAP1n2, MAP2n2, MAP3n2 sont les MAPs de niveau inférieur ; chaque MAP du niveau inférieur regroupe un ensemble de routeurs d'accès AR : le MAP1n2 relie deux AR : les nœuds 7 et 8 ; le MAP2n2 relie deux AR : les nœuds 10 et 11 ; et le dernier AR nœud 12 est relié avec le MAP3n2.

Le nœud mobile MN 9 se déplace de AR (nœud 8) vers AR (nœud 10), effectuant des handovers. Pendant son mouvement, le nœud mobile maintient la connexion avec le nœud 5 CN.

Le modèle proposé implémenté avec le protocole Mobile IP pour la gestion de la macromobilité et avec le protocole HMIP pour la gestion de la micromobilité. Le simulateur NS-2 possède des modules qui intègrent le protocole Mobile IP. Puisque le NS-2 ne possède pas des modules qui intègrent les protocoles de micromobilité, nous avons exploité

l'extension de mobilité de CIMS présentée ci-dessus. Cette extension nous a permis de réaliser des modèles de simulation sur NS-2 à base de ces protocoles.

5.5.2.1 Performance UDP et Performance TCP pour HMIP2L

L'étude de performances de notre modèle se fait à travers la génération et le routage des paquets UDP et des paquets TCP du nœud source (CN) au nœud destination (MN).

Une application CBR attachée au nœud CN pour générer des paquets UDP vers le MN, les paquets transmis par CN sont reçus par l'agent *Null* du nœud mobile (MN).

Les paquets transmis suivant le chemin vers le routeur d'accès (AR) dont le MN est relié, quand le MN commence à se déplacer vers un autre (AR) en effectuant des handoffs, le CN continue toujours de faire le routage des paquets vers MN à travers l'ancien AR. Le nombre des paquets perdus varie suivant la vitesse de mouvement de MN et suivant la distance du nœud mobile et son AR.

Le modèle est défini dans le script *mon-script.tcl* (voir Annexe 1)

Pour l'étude de performances de TCP, on utilise une application FTP au CN à gérer des paquets TCP au nœud mobile MN qui doit être équipé d'un agent *TCPsink* pour recevoir les paquets depuis CN.

5.5.2.2 Visualisation avec NAM

Pour visualiser le modèle dans NS-2, dans le *Shell* linux on exécute :

« ns mon-script.tcl »

Le résultat de la visualisation est un fichier Nam (hmipML.nam), (voir Figure 34).

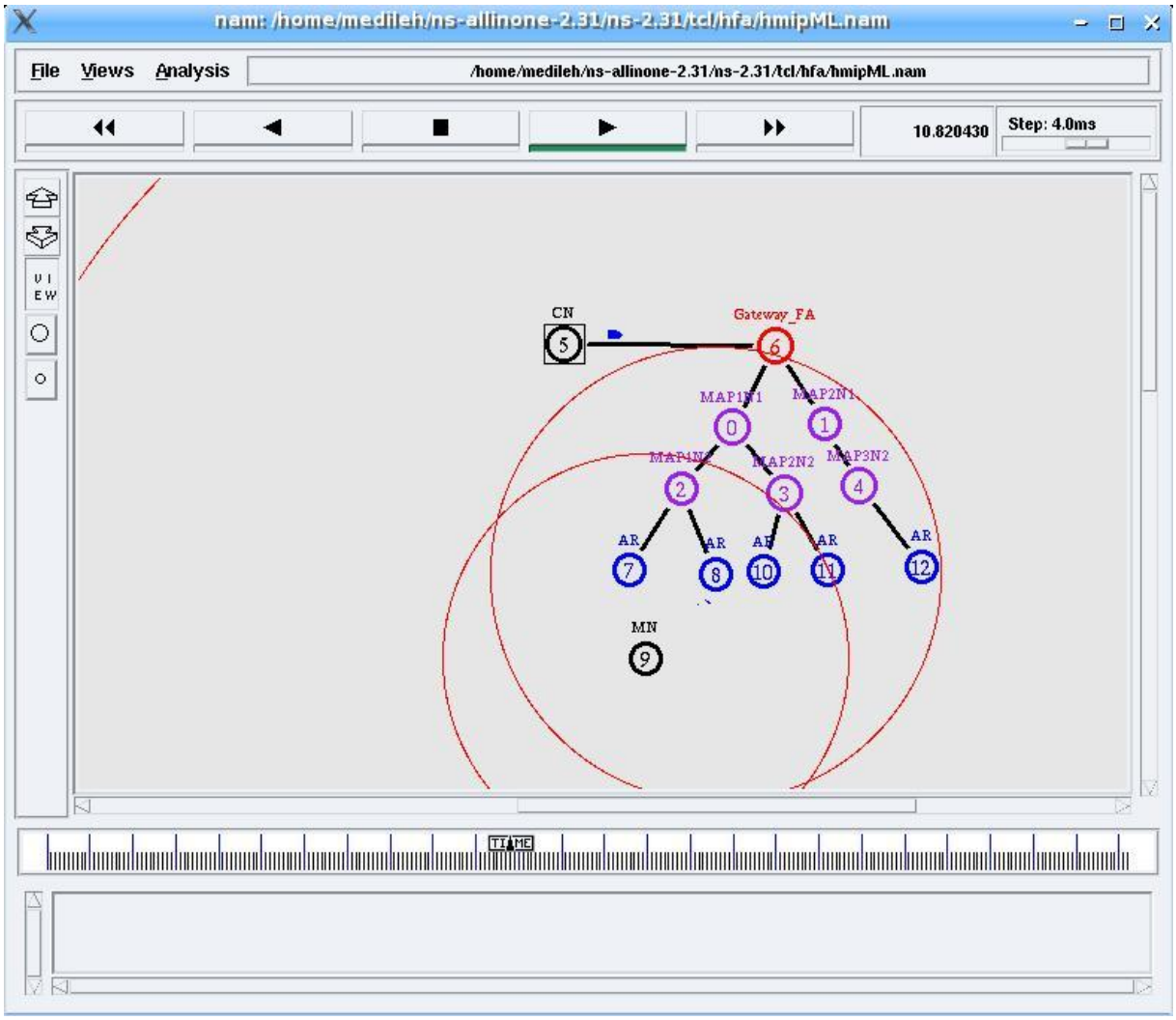


Figure 35. : Visualisation de modèle avec NAM.

5.5.3 L'analyse des résultas

Avec l'analyseur de fichiers de trace on peut obtenir des paramètres de performances tels que : paquets perdus (Packet loss), débit de la bande passante (Throughput bps), délai de retard (End to End delay)…

La perte de paquet : les paquets générés n'atteignent pas leurs destinations. Généralement les paquets se perdent avant que la destination ne puisse être atteinte.

$$\textbf{\textit{Paquet perdus = paquets transmis – paquets reçus}}$$

Débit de la bande passante Throughput : c'est le taux auquel le réseau envoie ou reçoit des données. C'est la capacité de canal de connexion du réseau qui est évaluée de bits par seconde (bit/s).

$$\textbf{\textit{Throughput}} = P_r / P_f \times \textbf{\textit{simTime}}$$

Avec :

P_r *: la somme des tailles des paquets reçus.*

P_f *: la quantité de paquets envoyés dans un certain intervalle de temps.*

simTime : temps de simulation.

Délai de retard : c'est le moment pris pour un paquet transmis à travers un réseau de source à la destination.

$$\boldsymbol{Delay = T_d - T_s}$$

Tel que :

$\boldsymbol{T_d}$ *: temps d'arrivée ou de réception du paquet par la destination.*

Ts*: temps d'envoi du paquet de la source.*

5.5.3.1 Scripts d'analyse des fichiers de trace

NS-2 permet de générer un fichier de trace (section 5.3.1). Dans notre travail on a développé des scripts d'analyse de fichier de trace avec AWK (voir Annexes) pour extraire des données nécessaires à l'étude de performance de notre modèle.

L'utilitaire AWK possède des instructions qui permettent de filtrer le fichier de sortie du simulateur colonne par colonne, l'extraction des données, le calcul des paramètres de performances et la réorganisation des résultats sous forme des valeurs numériques pour les visualiser sous forme des graphes.

5.5.3.2 Visualisation graphique des résultas

La figure 36 montre que dans le début de simulation, la bande passante a une valeur stable, quand l'agent UDP commence à générer des paquets vers la destination, le taux augmente rapidement, ce qui signifie que les paquets générés par le nœud source CN atteignent leurs destination MN ; durant le mouvement des MNs, la taille des données transmises vers la destination augmente aussi. Le graphe montre le taux throughput (bps) (Annexe 2) du système à chaque instant donné. Le système atteint un débit maximal de la seconde 3 à la seconde 10, c'est le taux maximal.

La valeur du débit change durant le processus de simulation suivant la distance qui sépare le nœud mobile de son routeur d'accès. Au moment où le MN commence à se déplacer vers l'autre routeur d'accès, le taux diminue jusqu'à il atteindre 0, et donc aucun paquet n'atteint sa destination.

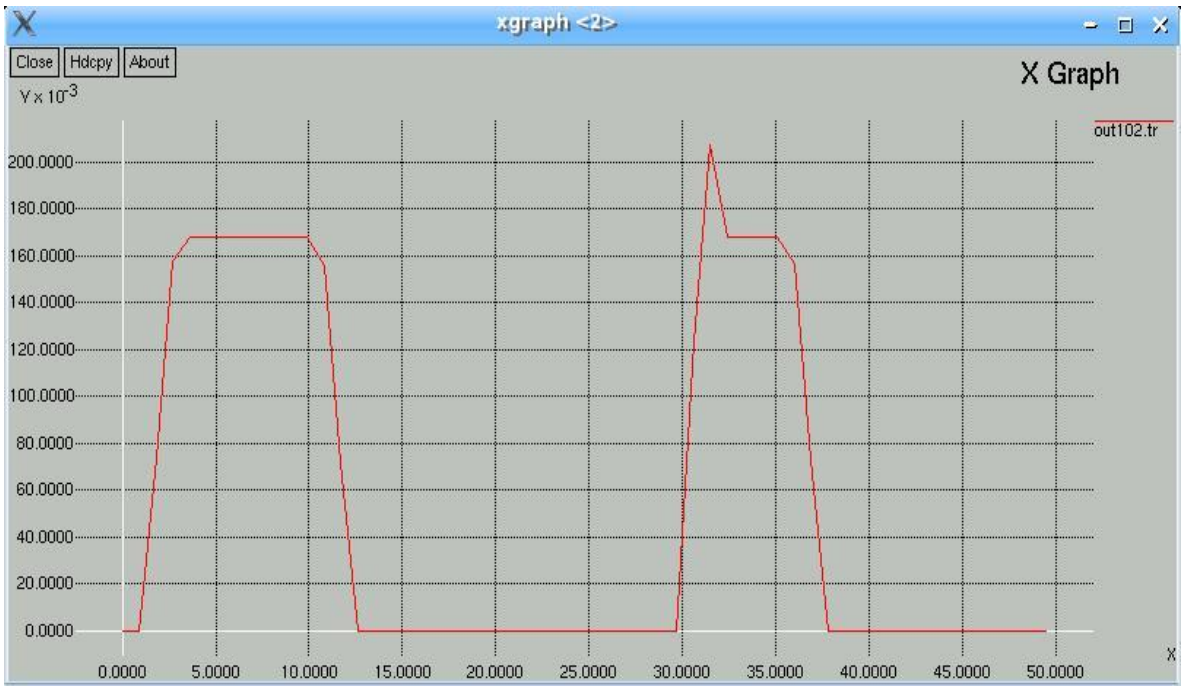


Figure 36. : Graphe bande passante (throughput) de système

Le graphe suivant (figure 37) montre la valeur calculée du débit entre les deux nœuds correspondants (CN, MN), la valeur maximale du taux est le débit de transfert entre le CN et MN quand le chemin est établi entre les deux nœuds. Si le MN effectue des handoff, le taux diminue. Ce débit est celui de l'échange entre les deux nœuds et non pas de tout le système.

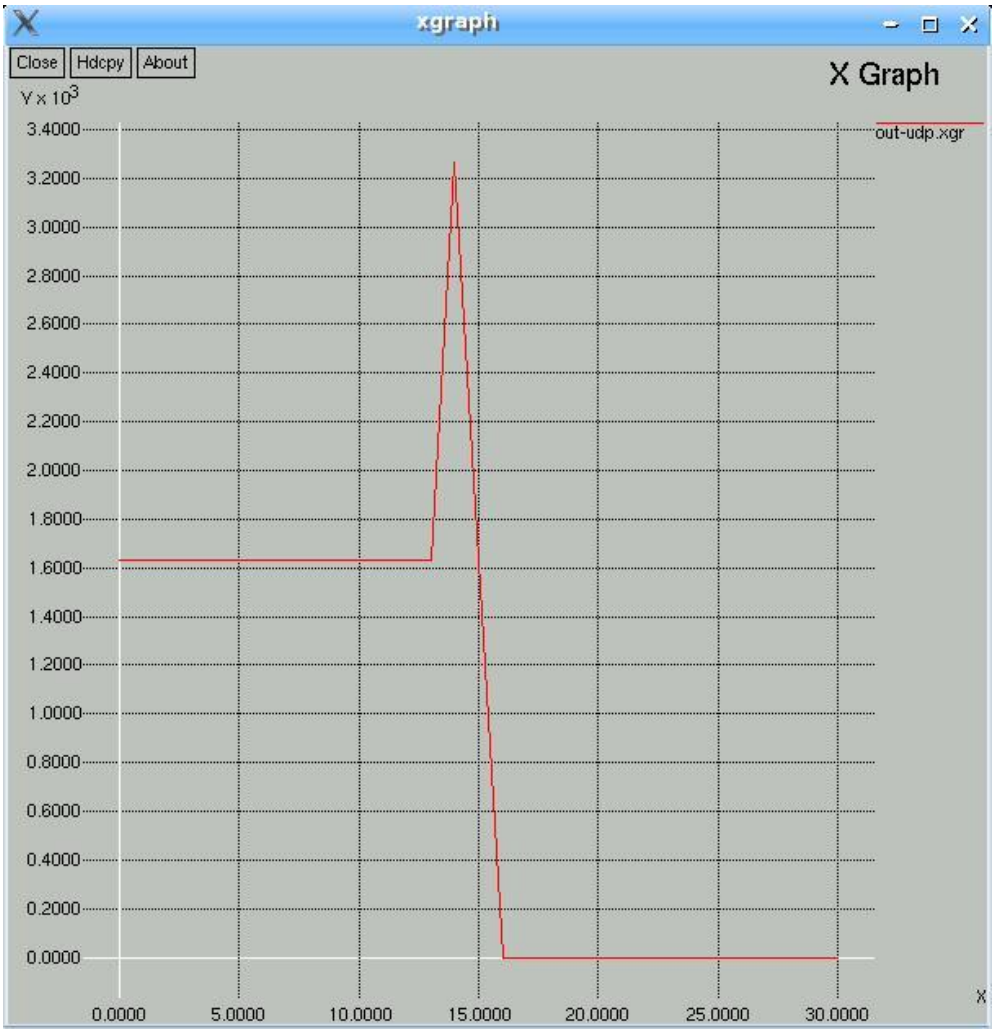


Figure 37. : bande passante throughput bit par second (bps)

Le graphe suivant (figure 38) montre le délai de retard de connexion (Annexe 3) quand le nœud mobile change sa position initiale et exécute des handoffs. Au début de simulation, le délai est court ; après quelques secondes et lorsque le MN commence son mouvement vers l'autre AR le délai croît, c-à-d des paquets reçus par le MN avec un peu de retard, mais finalement ils atteignent la destination, il arrive que quelques paquets sont perdus pendant le déplacement du MN.

Au moment où le MN rétablit la connexion avec l'AR le délai redevient minimal, lorsque le délai prend la valeur maximale les paquets sont complètements perdus et n'arrivent pas à atteindre la destination à partir de la source.

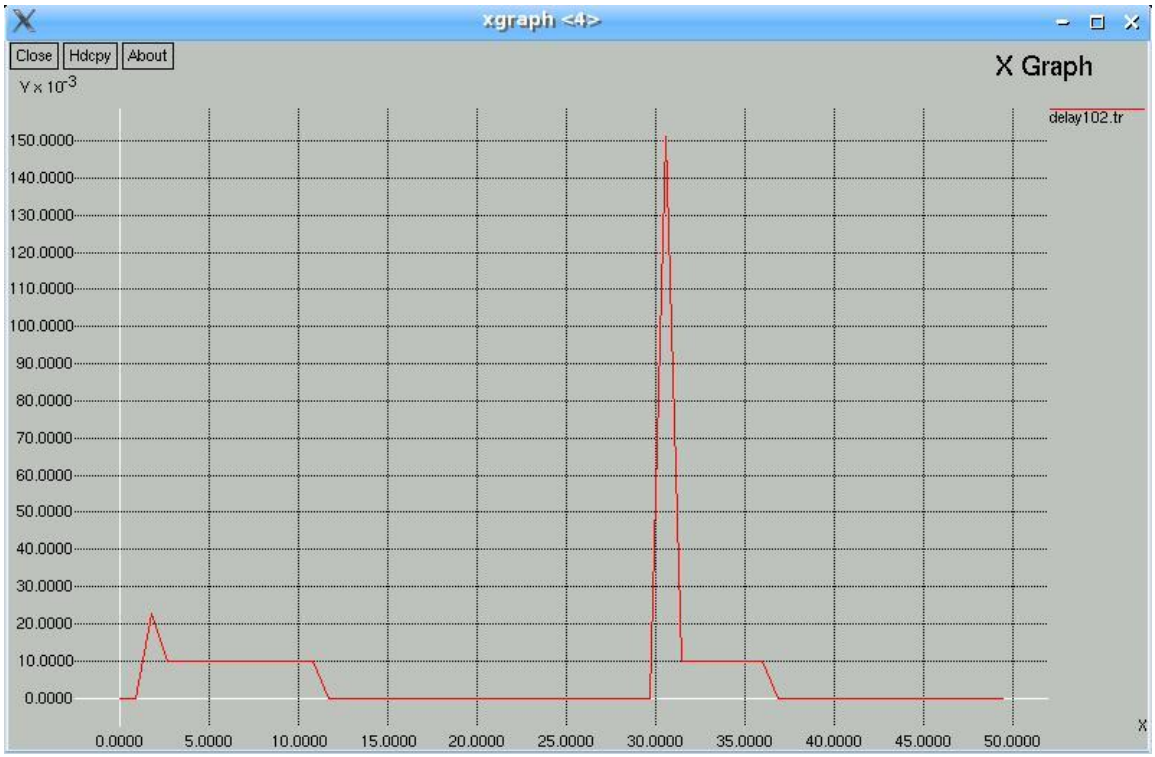


Figure 38. : Graphe délai de connexion (*delay*)

La perte des paquets (Annexe 4) est un paramètre critique et indispensable à étudier dans l'analyse de performance du modèle, c'est à dire moins de paquets sont perdus, plus, le modèle est performant. Le graphe suivant (figure 39) montre que depuis la seconde 0 à 30s aucune perte de paquets n'est signalée, tandis qu'après la seconde 30 la vitesse de perte atteint la valeur maximale.

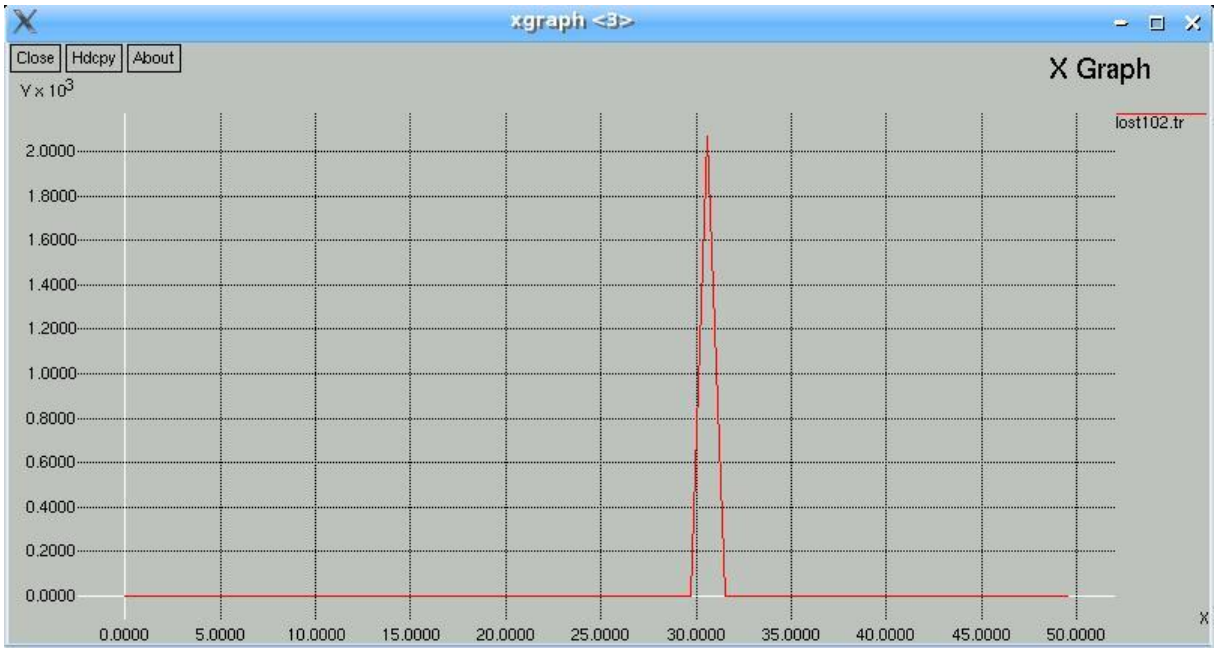


Figure 39. : Graphe vitesse de perte des paquets

Conclusion générale

Dans ce mémoire nous avons étudié les problèmes et les solutions liées à la mobilité dans l'Internet et dans les réseaux mobiles actuels. Un ensemble de propositions de différents organismes de recherches a été étudié. En général les solutions proposées sont à leur phase de spécification et d'évaluation.

Le protocole Mobile IP est la solution actuelle pour résoudre les problèmes de rupture de communication durant les déplacements des noeuds mobiles dans des réseaux IP. Ce protocole permet donc aux noeuds mobiles de se déplacer de réseau en réseau sans interrompre leurs sessions en cours. Le noeud mobile obtient une nouvelle adresse temporaire à chaque entrée dans un réseau visité. Cette adresse indique la position courante du nœud mobile dans l'Internet. Il devra la communiquer à son agent mère, qui se charge d'intercepter les paquets dans le réseau principal du noeud mobile et de les transmettre à sa position courante dans l'Internet.

Bien que MIP permet à des noeuds mobiles de se déplacer tout en continuant leur communication, des pertes de paquets peuvent avoir lieu. Les solutions proposées pour résoudre ces problèmes sont soit des améliorations directes à MIP, soit l'ajout de nouveaux protocoles de gestion d'une mobilité locale (limité à un domaine). Les améliorations apportées à MIP consistent en un échange d'informations entre les anciens et les nouveaux routeurs d'accès lors d'un handoff. Cet échange d'informations permet au nouveau routeur d'accès d'identifier le noeud mobile et de lui fournir un service plus rapidement.

MIP hiérarchique met en place une hiérarchie des réseaux. L'Internet est divisé en domaines. La mobilité à l'intérieur d'un domaine est appelée mobilité locale et la mobilité entre domaines mobilité globale. MIP hiérarchique cache les mouvements des noeuds mobiles à l'intérieur d'un domaine par un nœud de gestion de mobilité locale (*Mobility Anchor Point*). Le MAP agit comme un agent local pour le nœud mobile (MN).

L'utilisation d'un seul MAP dans le domaine garde un grand nombre des paquets en attente, ce qui cause un grand retard et une perte de plusieurs paquets, par conséquent la communication sera affectée (interrompue ou entrecoupée). Dans un domaine, le noeud mobile doit juste communiquer son entrée à son agent mère (et éventuellement ses correspondants), ensuite ses mouvements sont gérés par le domaine. Le nombre des messages de communication de mise à jour à chaque déplacement sera très important par la suite, les noeuds s'occupent de l'échange des paquets de contrôle que des paquets de données utiles. MIP hiérarchique est prévu pour un réseau avec un nombre de nœuds mobiles limité.

Cellular IP est un autre protocole de gestion de la micromobilité (mobilité locale à un domaine). Il gère aussi les mouvements à l'intérieur d'un domaine et les cachent au reste de l'Internet. Cellular IP s'inspire des systèmes cellulaires en utilisant la pagination (localisation des mobiles) et en faisant la distinction entre mobiles actifs et inactifs. Cellular IP est prévu pour fonctionner avec plusieurs mobiles et gérer les handoffs de manière optimisée. HAWAII ne remplace pas IP mais s'appuie sur lui dans son fonctionnement. Chaque station du réseau doit donc pouvoir fournir les services d'un routeur IP classique plus certaines fonctionnalités de gestion de la mobilité. TeleMIP et EMA sont des autres solutions pour la gestion de la micromobilité, ces protocoles sont étudiés dans ce mémoire à titre comparatif.

Ces solutions présentent des avantages. Elles possèdent aussi certains inconvénients. Dans notre solution HMIP2L, on a essayé de proposer une amélioration du protocole de mobilité HMIPv6. Nous avons tenté d'exploiter les avantages des différentes solutions afin de proposer une solution optimale et réalisable. Tandis que le protocole HMIP est une amélioration du protocole de mobilité, Mobile IP qui est déjà implémenté dans les systèmes d'exploitation, est le plus utilisé dans Internet et les réseaux IP pour la gestion de mobilité.

Les extensions de la mobilité dans NS-2 déjà réalisées nous a donné la possibilité d'élaborer des modèles de simulation de la mobilité pouvant s'exécuter sous NS-2. Notre solution propose une organisation hiérarchique multi-niveaux optimale (deux niveaux au maximum) de multiples MAPs pour diminuer le coût de sélection du MAP et réduire le nombre de messages de signalisation et de mise à jour hors les domaines du MAP.

L'étude des performances de notre modèle a consisté à étudier la génération et le routage des paquets UDP et des paquets TCP du nœud source (CN) au nœud destination (MN). Le nœud mobile se déplace de l'ancien routeur d'accès vers le nouveau routeur d'accès. L'analyseur de fichier de trace généré par NS-2 nous a permis d'étudier et d'évaluer des paramètres de performance tels que : le nombre de paquets perdus (*Packet loss*), débit ou la bande passante (*Throughput bps*) et délai ou retard (*End to End delay*). Les résultats sont représentés sous formes de graphes.

En perspective à ce travail, nous espérons continuer l'évaluation de notre solution par l'élaboration d'autres modèles de simulation et l'étude d'autres paramètres de performance (gigue,…). L'analyse et la comparaison des résultats aux autres méthodes, nous donneront la possibilité d'estimer et de prouver le bien fondé de notre solution.

Annexes

Annexe (1) :

Fichier script : *mon_script.tcl*

Contient la description du modèle de simulation en langage Otcl :

mon_script.tcl 1

```
##########################     HFA-simulation    ##########################
#
#                        o W1                 WIRED NODES
#                        |
#                        o W2
#                       / \
#                      /   \
#--*--*--*--*--*--*- o      o base-stn nodes  --*-*-*-*-*-*-*-*-
#                    BS1      BS2
#                        o
#                   o    MH       o          WIRELESS NODE MOVING
#                MH                MH    FROM BS1 TO BS2.
#
#================================================================================
#       Code source de l'éxtension de mobilité CIMS de protocole HMIP     #
#       le code eté modifier afin de rélaiser le modèle de notre solution #
#       Modification: le 16/12/2008. par Mr. MEDILEH Saci                  #
#================================================================================

#Iniialisation des options pour le domaine de réseau san-fil.
set opt(chan)           Channel/WirelessChannel
set opt(prop)           Propagation/TwoRayGround
set opt(netif)          Phy/WirelessPhy
set opt(mac)            Mac/802_11
set opt(ifq)            Queue/DropTail/PriQueue
set opt(ll)             LL
set opt(ant)            Antenna/OmniAntenna
set opt(x)              500      ;# X & Y dimension of the topography
set opt(y)              500      ;# hard wired for now...
set opt(rp)             NOAH     ;# Le protocole de routage NOAH
set opt(ifqlen)         50               ;# max packet in ifq
set opt(seed)           0.0
set opt(stop)           25.2             ;# default simulation time
set opt(cc)             "off"
set opt(tr)             hmip2L.tr        ;# trace file (Wireless traffic)
set opt(cp)             ""
set opt(sc)             ""
set opt(ftp0-start)     6.0
set opt(BgTrafficStart) 0.1

# ======================================================================
#Déclaration du nombre des nodes filaire - san-fil dans le domaine#
# ======================================================================
set num_wired_nodes     7
set num_bs_nodes        5
set num_wireless_nodes 1

set opt(nn)             6         ;# total number of wireless nodes
#=======================================================================

Agent/TCP set sport_            0
Agent/TCP set dport_            0
Agent/TCP set packetSize_       1460
```

mon_script.tcl 2

```
Queue/DropTail/PriQueue set Prefer_Routing_Protocols    1

# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 0.2
Antenna/OmniAntenna set Gr_ 0.2
#======================================================================
# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

# ======================================================================
source ../lib/ns-wireless-mip.tcl
#================ mdif
set GLOBAL_rnd [new RandomVariable/Uniform]
$GLOBAL_rnd set min_            0.0
$GLOBAL_rnd set max_            1

# Registration interval in Mobile IP
Agent/MIPMH set reg_rtx_ 1.0
# ======================================================================
## Session Specific setting, command line arguments
#  usage: ns sample_msf <speed> <overlap> <simulation Time>
#
set speed(1) [ lindex $argv 0 ]
set overlap [ lindex $argv 1 ]
set stopTime [ lindex $argv 2 ]
if {$stopTime != ""} {
    set opt(stop) $stopTime
    if {$stopTime <= 1} {
        set opt(stop) 1.1; # since traffic for MH start at 1
    }
}

# Increase Queue in link to introduce delay instead of dropping
Queue set limit_ 200

# number of Packets received in MH's sink
set pktsNum 0
# Configuration de l'adressage hiérachique
# creation d'un nouveau instance de simulation
set ns [new Simulator]
$ns set-address-format hierarchical

# set mobile IP flag
```

mon_script.tcl 3

```
Simulator set mobile_ip_ 1

# ======================================================================
# enable HFA Routing
set HFA_Routing 1
# ======================================================================

set namtrace [open hmip2L.nam w]
$ns namtrace-all $namtrace
set tracefd     [open $opt(tr) w]
set trace [open /dev/null w]
$ns trace-all $tracefd
#======================================================================
# les noeuds dans le domaine
#======================================================================
AddrParams set domain_num_ 1
lappend cluster_num  11
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 1 1 1 1 1 1 2 1 1
AddrParams set nodes_num_ $eilastlevel
#======================================================================
#create intermediate host;  MAPs dans notre solution
#======================================================================
set W_(1) [$ns node 0.0.0]
set W_(2) [$ns node 0.1.0]
set W_(3) [$ns node 0.2.0]
set W_(4) [$ns node 0.3.0]
set W_(5) [$ns node 0.4.0]
#create corresponding host le CN
set host [$ns node 0.5.0]
#======================================================================
#Inclure le fichie de Biblio,extension de mobilité de Protocole HMIP
#setup a wired-Gateway-ForeignAgent
#======================================================================
source hfa-lib_ns-2.26.tcl

Simulator set node_factory_ MobileNode/GFA
Simulator set EnableHierRt_ 1
set W_(0) [$ns node 0.6.0]
$W_(0) color "red"

# Highlight mobility-unaware router
foreach i {1 2 3 4 5} {
    $W_($i) color "purple"
}

#======================================================================
#create common objects reqd for wireless sim.
if { $opt(x) == 0 || $opt(y) == 0 } {
        puts "No X-Y boundary values given for wireless topology\n"
}
set chan        [new $opt(chan)]
set prop        [new $opt(prop)]
set topo        [new Topography]
```

mon_script.tcl 4

```
# setup topography and propagation model
$topo load_flatgrid $opt(x) $opt(y)
$prop topography $topo

# Create God
create-god $opt(nn)

# Configure using NOAH routing in Wireless domain;
#cofiguration des noeuds
$ns node-config -mobileIP ON \
                -adhocRouting $opt(rp) \
                -llType $opt(ll) \
                -macType $opt(mac) \
                -ifqType $opt(ifq) \
                -ifqLen $opt(ifqlen) \
                -antType $opt(ant) \
                -propType $opt(prop) \
                -phyType $opt(netif) \
                -channelType $opt(chan) \
                -topoInstance $topo \
                -wiredRouting ON \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace ON

# Set tranmission Power so that overlapping area is desired
if {$overlap == ""} {
    set overlap 30
}

set power 0.29705643626340894

#===================================================================
# création du noeud mobile that would be moving between BS3 and BS4
#===================================================================
$ns node-config -wiredRouting ON;
## setup HFA Base Station nodes using NOAH
$ns node-config -rxPower $power -txPower $power
set BS(1) [$ns node 0.7.0]
$ns node-config -rxPower $power -txPower $power
set BS(2) [$ns node 0.8.0]
$ns node-config -wiredRouting OFF;
$ns node-config -rxPower $power -txPower $power
set MH_(1) [$ns node  0.8.1]
set GFAaddress [AddrParams addr2id [$W_(2) node-addr]]
makeHfaMH $MH_(1) $GFAaddress
$ns node-config -wiredRouting ON;
$ns node-config -rxPower $power -txPower $power
set BS(3) [$ns node 0.9.0]
$ns node-config -rxPower $power -txPower $power
set BS(4) [$ns node 0.10.0]
$ns node-config -rxPower $power -txPower $power
```

mon_script.tcl 5

```
set BS(5) [$ns node 0.11.0]

foreach i {1 2 3 4 5} {
    makeHfaBS $BS($i)
}

# Table to convert address between dot format and integer for all wired
# nodes
foreach i {0 1 2 3 4 5} {
    createAddrTable $W_($i)
}

#===================================================================
#Position pour chaque base-station nodes.
#===================================================================
$BS(1) set X_ 1.000000000000
$BS(1) set Y_ 1.000000000000
$BS(1) set Z_ 0.000000000000

$BS(2) set X_ 100.000000000000
$BS(2) set Y_ 100.000000000000
$BS(2) set Z_ 0.000000000000

$BS(3) set X_ 200.000000000000
$BS(3) set Y_ 200.000000000000
$BS(3) set Z_ 0.000000000000

$BS(4) set X_ 300.000000000000
$BS(4) set Y_ 300.000000000000
$BS(4) set Z_ 0.000000000000

$BS(5) set X_ 400.000000000000
$BS(5) set Y_ 400.000000000000
$BS(5) set Z_ 0.000000000000
# Set default mobile movement speed, cell overlapping size, simulation time.

}
# Set tranmission Power so that overlapping area is desired
if {$overlap == ""} {
    set overlap 30
if {$speed(1) == ""} {
    set speed(1) 20.000000000000

}

set ns_ [Simulator instance]

## Label the Special Node in NAM
$ns_ at 0.0 "$BS(1) label AR"
$ns_ at 0.0 "$BS(2) label AR"
$ns_ at 0.0 "$BS(3) label AR"
$ns_ at 0.0 "$BS(4) label AR"
$ns_ at 0.0 "$BS(5) label AR"
$ns_ at 0.0 "$W_(0) label Gateway_FA"
```

mon_script.tcl 6

```
$ns_ at 0.0 "$W_(1) label MAP1N1"
$ns_ at 0.0 "$W_(2) label MAP2N1"
$ns_ at 0.0 "$W_(3) label MAP1N2"
$ns_ at 0.0 "$W_(4) label MAP2N2"
$ns_ at 0.0 "$W_(5) label MAP3N2"
$ns_ at 0.0 "$MH_(1) label MN"
$ns_ at 0.0 "$host label CN"

# ======================================================================
# movement of the MH move towards other BS
#======================================================================

# number of handoff in this simulation session
set NumOfHO 110

set stime 4
set StayTime 2
# les cordonnés initials de MN
set spoint 100.0;#330.0
set dpoint 370.0;#370.0
#======================================================================
#calcule de la distance à chaque instant de handoff
#======================================================================
set dist [expr (sqrt(2 * (pow(($dpoint - $spoint), 2))))]
puts "************************* DISTANCE: $dist **********************"
set trip_time [expr ($dist / $speed(1)) + $StayTime]

$MH_(1) set Y_ $spoint
$MH_(1) set X_ $spoint

set rnd [new RandomVariable/Uniform]
$rnd set min_ 0.0
$rnd set max_ 1

for {set i 1} {$i < [expr $NumOfHO +1]} {incr i} {
    set rv [$GLOBAL_rnd value]
    set rv [$GLOBAL_rnd value]
    set rv [$GLOBAL_rnd value]
    set j [expr $i * $trip_time - $stime]
    set k [expr $i % 2]
    if { $k == 1 } {
         $ns at $j "$MH_(1) setdest $dpoint $dpoint $speed(1)"
         $ns at $j "puts =============>>>"
    } else {
         $ns at $j "$MH_(1) setdest $spoint $spoint $speed(1)"
         $ns at $j "puts <<<============="
    }
    puts "$i*****time: $j"
}

if { $opt(x) == 0 || $opt(y) == 0 } {
        usage $argv0
        exit 1
}
```

mon_script.tcl 7

```
if {$opt(seed) > 0} {
        puts "Seeding Random number generator with $opt(seed)\n"
        ns-random $opt(seed)
}

#
# Source the Connection and Movement scripts
#
if { $opt(cp) == "" } {
        #puts "*** NOTE: no connection pattern specified."
        set opt(cp) "none"
} else {
        puts "Loading connection pattern..."
        source $opt(cp)
}

if { $opt(sc) == "" } {
        #puts "*** NOTE: no scenario file specified."
        set opt(sc) "none"
} else {
        puts "Loading scenario file..."
        source $opt(sc)
        puts "Load complete..."
}
#===============================================================
# creation et configuration des liens entre
# les stations de base (AR) et les autres noeuds filaires
#===============================================================
$ns duplex-link $host   $W_(0) 10Mb 2ms DropTail
$ns duplex-link $W_(0) $W_(1) 10Mb 2ms DropTail
$ns duplex-link $W_(0) $W_(2) 10Mb 2ms DropTail
$ns duplex-link $W_(1) $W_(3) 10Mb 2ms DropTail
$ns duplex-link $W_(1) $W_(4) 10Mb 2ms DropTail
$ns duplex-link $W_(2) $W_(5) 10Mb 2ms DropTail
$ns duplex-link $W_(3) $BS(1)  10Mb 2ms DropTail
$ns duplex-link $W_(3) $BS(2)  10Mb 2ms DropTail
$ns duplex-link $W_(4) $BS(3)  10Mb 2ms DropTail
$ns duplex-link $W_(4) $BS(4)  10Mb 2ms DropTail
$ns duplex-link $W_(5) $BS(5)  10Mb 2ms DropTail

$ns duplex-link-op $host   $W_(0) orient right
$ns duplex-link-op $W_(0) $W_(1) orient left-down
$ns duplex-link-op $W_(0) $W_(2) orient right-down
$ns duplex-link-op $W_(1) $W_(3) orient left-down
$ns duplex-link-op $W_(1) $W_(4) orient right-down
$ns duplex-link-op $W_(2) $W_(5) orient right-down
$ns duplex-link-op $W_(3) $BS(1) orient left-down
$ns duplex-link-op $W_(3) $BS(2) orient right-down
$ns duplex-link-op $W_(4) $BS(3) orient left-down
$ns duplex-link-op $W_(4) $BS(4) orient right-down
$ns duplex-link-op $W_(5) $BS(5) orient right-down
#===============================================================
#       Couleur des paquets
```

mon_script.tcl 8

```
#===================================================================
$ns color 1 Blue
$ns color 5 Blue
$ns color 6 Blue
$ns color 7 Blue
$ns color 8 green
$ns color 22 Blue
# Color for Control packet
$ns color 0 Red
#===================================================================
# configuration de  UDP connection
#===================================================================
set udp0 [new Agent/UDP]
$ns attach-agent $host $udp0
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set interval_ 10ms
$cbr_(0) attach-agent $udp0
set null_(0) [new Agent/LossMonitor]
$ns attach-agent $MH_(1) $null_(0)
$ns connect $udp0 $null_(0)
$udp0 set fid_ 11
foreach i {11 12} {
$ns color $i Blue
}
foreach i {1 2 3} {
$ns color $i orange
}
#==========================endof UDP=============================
#===================================================================
# UDP traffic to the MH 1
#===================================================================
$ns at 1.0 "$cbr_(0) start"
$ns at [expr $opt(stop) - 0.5] "$cbr_(0) stop"
#===================================================================
# Tell all the nodes when the simulation ends
#===================================================================
for {set i 1} {$i < $num_wireless_nodes } {incr i} {
    $ns_ at $opt(stop).0000010 "$MH_($i) reset";
}
foreach i {1 2 3 4 5} {
    $ns_ at $opt(stop).0000010 "$BS($i) reset";
}
foreach i {1} {
$ns_ at $opt(stop).20 "$MH_($i) log-movement"
}
$ns_ at $opt(stop).21 "finish"
$ns_ at $opt(stop).20 "puts \"NS EXITING...\" ; "
#=============================================================================
# procedure de fin de simulation
# fermeture des fichier de trace
# excution des scripts d'analyse des résultas
# construction des graphes
#=============================================================================
proc finish {} {
```

mon_script.tcl 9

```
        global ns_ trace namtrace null_ cbr_ pktsNum mytrace
        global HawaiiRoutingMSF

    puts stderr "========================================="
    puts stderr "Result for M-level Hierarchical Mobile IP"
    puts stderr "========================================="
    foreach i {0} {
        puts stderr "Total number of packet lost:\
                [expr [$cbr_($i) set seqno_]-[$null_($i) set npkts_]]"
        puts stderr "Total packet sent:[$cbr_($i) set seqno_] \
                received:[$null_($i) set npkts_]"
    }
        $ns_ flush-trace
        close $namtrace
        close $trace
        exec rm -f out-udp.xgr
        exec awk -f fil-udp.awk hmipML.tr > out-udp.xgr
        exec xgraph out-udp.xgr &
        puts "running nam..."
        exec ../../../bin/nam hmipML.nam &
        puts "Finishing ns.."
        exit 0
}
#========================== fin de proc finish ============================
#==========================================================================
# information pour le fichier de trace
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp $opt(rp)"
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"
#==========================================================================
puts "Starting Simulation..."
$ns_ run
#========================== lancement de simulation =======================
#============================== fin de script =============================
##########################################################################
```

file:///home/medileh/Desktop/mon_script.tcl

Annexe (2) :

Fichier de script AWK *Throghput.awk*, pour analyser le fichier de trace hmip2L.tr

Throughput.awk 1

```
#==============================================================================
=====
# script AWK; Analyse le fichier de trace  pour calculer le throughput
# le taux de débit de la bande passante entre le CN & MN
#==============================================================================
=====
#==============================================================================
=====
# Intialisation des variables de l'analyseur
BEGIN {
        recv = 0
        simtime=30.0
        hdr_size = 0
        flow_t = "cbr"
        dst = 8
        flow =5
}
# parcoure de fichier de trace colonne par colonne
{
         # Trace line format: normal
                 event = $1
                 time = $2
                 if (event == "+" || event == "-") node_id = $3
                 if (event == "r" || event == "d") node_id = $4
                 flow_id = $8
                 pkt_id = $12
                 pkt_size = $6


         # Calculate total received packets' size
         if (event == "r" && node_id == dst) {
                 if (flow_t != "sctp") {
                         recv += pkt_size - hdr_size
                         printf("recv[%g] = %g --> tot:
%g\n",node_id,pkt_size-hdr_size,recv)
                 } else {
                         # Rip off SCTP header, whose size depends
                         # on the number of chunks in each packet
                         if (pkt_size != 448 && pkt_size != 864 && pkt_size !=
1280) pkt_size = 0
                         if (pkt_size == 448) pkt_size = 400
                         if (pkt_size == 864) pkt_size = 800
                         if (pkt_size == 1280) pkt_size = 1200
                         recv += pkt_size
                 }
         }
 }
#==============================================================================
# ecriture des valeurs numérique dans un fichier afin d'etre employé par XGraph
 END {
         printf("%10g %10s %10g\n",flow,flow_t,(recv/simtime)*(8/1000))
 }
#==============================FIN=============================================
```

file:///home/medileh/Desktop/Throughput.awk

Annexe (3) :

Fichier script AWK *dely.awk* extrait des valeurs numériques et calcule le délai retard.

delay.awk 1

```
#====================== delay.awk ====================================
#=============================================================================
=====
# script AWK; Analyse le fichier de trace  pour calculer le Delay
#=============================================================================
=====
#=============================================================================
=====
# Intialisation des variables de l'analyseur
BEGIN {
        for (i in send) {
                send[i] = 0
        }
        for (i in recv) {
                recv[i] = 0
        }
        delay = avg_delay = 0
        src = 5
        dst = 9
}
# parcoure de fichier de trace colonne par colonne
{
         # Trace line format: normal
         if ($2 != "-t") {
                 event = $1
                 time = $2
                 if (event == "+" || event == "-") node_id = $3
                 if (event == "r" || event == "d") node_id = $4
                 flow_id = $8
                 pkt_id = $12
         }
         # Trace line format: new
         if ($2 == "-t") {
                 event = $1
                 time = $3
                 node_id = $5
                 flow_id = $39
                 pkt_id = $41
         }

         # Store packets send time
         if  (node_id == src && send[pkt_id] == 0 &&  (event == "+" || event ==
"s")) {
                 send[pkt_id] = time
                 #printf("send[%g] = %g\n",pkt_id,time)
         }
         # Store packets arrival time
         if (node_id == dst && event == "r") {
                 recv[pkt_id] = time
                 #printf("\t\trecv[%g] = %g --> delay[%g] =
%g\n",pkt_id,time,pkt_id,recv[pkt_id]-send[pkt_id])
         }
 }
#=============================================================================
```

delay.awk 2

```
# ecriture des valeurs numérique dans un fichier afin d'etre employé par XGraph
#==============================================================================
 END {
        # Compute average delay
        for (i in recv) {
                if (send[i] == 0) {
                        printf("\nError %g\n",i)
                }
                 delay += recv[i] - send[i]
                #printf("%g %g\n",i, recv[i]- send[i]) >> "del.txt"
                 num ++
        }

        #printf("%10g ",flow)
        if (num != 0) {
                avg_delay = delay / num
        } else {
                avg_delay = 0
        }
        printf("%10g\n",avg_delay*1000)
 }
#================================FIN===========================================
```

file:///home/medileh/Desktop/delay.awk

Annexe (4) :

Fichier de script AWK *pakt_loss.awk* calcule le nombre des paquets perdus et reçus et le taux de perte.

pakt_loss.awk 1

```
#======================== pkt_loss.awk ===================================
# un script AWK analyse le fichier de trace pour calculer le nombre
# des paquets perdus a chaque instant pendant la comunication entre CN & MN
#=========================================================================
BEGIN {
        for (i in send) {
                send[i] = 0
        }
        for (i in recv) {
                recv[i] = 0
        }
        tx = 0
        drop = 0
        pkt_loss = 0
        src = 5
        dst = "_9_"
        flow = 11
}
#=============================================================
# parcoure de fichier de trace
#=============================================================
{
        # Trace line format: normal
        if ($2 != "-t") {
                event = $1
                time = $2
                if (event == "+" || event == "-") node_id = $3
                if (event == "r" || event == "d") node_id = $4
                flow_id = $8
                pkt_id = $12
        }
         # Trace line format: new
       if ($2 == "-t") {
                event = $1
                time = $3
                node_id = $5
                flow_id = $39
                pkt_id = $41
       }

        # Store packets send time
        if (flow_id == flow && node_id == src && send[pkt_id] == 0 && (event ==
"+" || event == "s")) {
                send[pkt_id] = 1
                #printf("send[%g] = 1\n",pkt_id)
         }
         # Store packets arrival time
         if (flow_id == flow && node_id == dst && event == "r") {
                recv[pkt_id] = 1
                #printf("\t\trecv[%g] = 1\n",pkt_id)
         }
 }
#=============================================================
# ecriture des valeurs dans un fichier pour XGRAPH
```

pakt_loss.awk 2

```
#=============================================================
END {
        #printf("%10g ",flow)
        for (i in send) {
                if (send[i] == 1) {
                        tx ++
                        if (recv[i] == 0) {
                               drop ++
                                #printf("pkt %g not recvd\n",i)
                        }
                 }
        }
        if (tx != 0) {
                pkt_loss = drop / tx
        } else {
                pkt_loss = 0
        }
        printf("%10g %10g %10g\n",tx,drop,pkt_loss*100)
}
#==================== FIN ====================================
```

file:///home/medileh/Desktop/pakt_loss.awk

Bibliographie

[Bellier 2000] Bellier L., Castelluccia C., “Hierarchical Mobile IPv6”, Internet draft, July 2000.

[Campbell 2000] A. T. Campbell, J. Gomez, S. Kim, A. Valko, C-Y. Wan, Z. Turanyi, “Design, Implementation, and Evaluation of Cellular IP”, IEEE Personal Communications, Août 2000.

[Campbell 2002] A. T. Campbell, J. Gomez, S. Kim, AND C-Y. Wan, “Comparison of IP Micromobility Protocols”, COLUMBIA UNIVERSITY, IEEE Wireless Communications, Feburary 2002

[Chaouchi 2006] Hakima chaouchi « Gestion de la mobilité dans les futurs réseaux IP », Université Pierre et Marie Curie Laboratoire d’Informatique Parie 6, article de recherche sur le site Internet : http://www-rp.lip6.fr/dnac/Hakima.pdf, visité septembre 2006.

[Chung 2007] WonSik Chung and SuKyoung Lee, “Cost-Effective MAP Selection in HMIPv6 Networks” IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

http://www.prism.uvsq.fr/~jbo/ICC07/DATA/S09S48P01.pdf

[Estrela 2007] P. Estrela, IST-CIMS MANUAL “IP Micro-Mobility Suite for NSv2.31”, Institut Supérieur Technique, Université technique de LISBON, Décembre 2007.

[Gustafsson 2002] Gustafsson E., Jonsson A., Perkins C.E. “Mobile IPv4 Regional Registation”, 22 octobre 2002.

[Khouaja 2006] Youssef Khouaja , Karine Guillouard , Philippe Bertin - France Télécom R&D ; Jean-Marie Bonnin - ENST Bretagne, « Gestion Hiérarchique de la Mobilité IPv6 Contrôlée par le réseau », article sur le site Internet : http://www-rp.lip6.fr/dnac/3.2-khouaja-article.pdf, visité septembre 2006.

[Ma'en 2007] Ma'en Alrashdan, Mahamod Ismail, Kasmiran Jumari, “A Study on Effective Transfere Rate Over Smart Hierarchical Mobile IPv6 (SHMIPv6)”, International Journal of Computer Science and Network security, (2007).

http://paper.ijcsns.org/07_book/200705/20070535.pdf

[Minji 2004] Minji Nam, S. Pack and Yanghee Choi, "A study on optimal hierarchy in multi-level hierarchical mobile IPv6 networks," IEEE Communications Society Globecom 2004.

http://mmlab.snu.ac.kr/publications/docs/Optimal%20HMIPv6_Globecom.pdf

[Nicolas 2001] Nicolas Montavont « La mobilité dans les réseaux IP », rapport de DEA informatique, Université Louis Pasteur de Strasbourg, 2001.

[NS2 2007] NS-2 ,"The ns Manual" formerly ns notes and documentation; 10 décembre 2007.

[NS-2] HomePage de NS : http://www.isi.edu/nsnam/ns/index.html, visité décembre 2008.

[O'Neill 2000] A. O'Neill, G. Tsirtsis, and S. Corson. "Edge Mobility Architecture", Internet draft, draft-oneill-ema-01.txt,(work in progress.) March 2000.

[Pack 2004] S. Pack, T. Kwon, and Y. Choi, "A Comparative Study of Mobility Anchor Point Selection Schemes in Hierarchical Mobile IPv6 Networks," in Proc. ACM

MobiWac 2004, September 2004.

http://mmlab.snu.ac.kr/publications/docs/p173- shpack.pdf

[Pack 2005] S. Pack et al., "An Adaptive Mobility Anchor Point Selection Scheme in Hierarchical Mobile IPv6 Networks," Technical Report, Seoul National University, 2005.

http://mmlab.snu.ac.kr/publications/docs/Adaptive_CC_Revision.pdf

[Pack 2006] S. Pack, , Choi, Y., and Nam, M. 2006. "Design and Analysis of Optimal Multi-Level Hierarchical Mobile IPv6 Networks." Wirel. Pers. Commun. 36, 2 (Jan. 2006), 95-112.

DOI= http://dx.doi.org/10.1007/s11277-006-0025-7

[Pack 2007] S. Pack, , Kwon, T., and Choi, Y. 2007. "A performance comparison of mobility anchor point selection schemes in Hierarchical Mobile IPv6 networks." Computer. Networks 51, 6 (Apr. 2007), 1630-1642.

DOI= http://dx.doi.org/10.1016/j.comnet.2006.09.002

[Parks 1997] V. D. Parks and M. S. Corson. A highly adaptative distributed routing algorithme for mobile wireless networks. In IEEE Proceedings of INFOCOM, April 1997.

[Parks 1999] V. Parks and S. Corson. Temporally-Ordered Routing Algorithm (TORA) version 1, functionnal specification. Internet draft, draft-ietf-manet-tora-spec-02.txt, (work in progress), October 1999.

[Ramjee 2000] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. "IP micro-mobility support using HAWAII". Internet draft, draft-ietf-mobileip-hawaii-01.txt, (work in progress), July 2000.

[Reinbold 2001] P. Reinbold and O. Bonaventure. "A Comparison of IP Mobility Protocol". Techni- cal Report Infonet-TR-2001-07, University of Namur, Infonet Group, June 2001. http://www.infonet.fundp.ac.be/doc/tr/.

[Soliman 2000] H. Soliman, K.Malki "Hierarchical Mobile IPv4/v6 and Fast Handovers", internet Draft, IETF, (work in progress), March 2000.

[Soliman 2004] H. Soliman, C. Castelluccia, K.Malki, L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", Internet Draft, IETF, draftietf-mipshop-hmipv6-03.txt (work in progress), (2004).

[Subir 2000] Subir Das et al. TeleMIP : Telecommunication-Enhanced Mobile IP Architecture for Fast Intradomain Mobility. IEEE Personal Communications, 7(4) :50–58, August 2000.

[Thierry 2001] Thierry Ernst «Le support des Réseaux Mobiles dans IPv6», "Network Mobility Support in IPv6", Thèse de doctorat, Université Joseph Fourier de Grenoble, soutenue le 29 octobre 2001.

[Thomas 2006] Thomas NOËL, « La mobilité IP », article de recherche sur le site Internet : http http://1999.jres.org/articles/noel-te-12-final.pdf, visité septembre 2006.

[Tsirtsis 2001] G. Tsirtsis, A. Yegin, C. Perkins, G. Dommety, K. El-Malki, M. Khalil, "Fast Handovers for Mobile IPv6", Internet Engineering Task Force draft-ietf-mobileip-fast-mipv6-00.txt, Février 2001.

[VINT 1996] Projet VINT , Virtual InterNetwork Testbed, methods ans system, Université de Californie du Sud ISI, 1996. http://www.isi.edu/nsnam/vint/index.html visité novembre 2008.

[Xu 2003] Yi Xu Henry C. J. Lee Vrizlynn L. L. Thing, "A Local Mobility Agent Selection Algorithm for Mobile Networks", IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2003, VOL 2, pages 1074-1079

http://icsd.i2r.a-star.edu.sg/publications/HenryLee_2003_icc.pdf

[Zheng 2001] WAN Zheng, “Internet QoS, Architectures and Mechanisms for Quality of Service”, Morgan Kaufmann Publisher, 2001.

[Zheng 2006] WAN Zheng, PAN Xue-zeng, CHEN Jian, CUI Yu-zeng, “A three-level mobility management scheme for hierarchical mobile IPv6 networks” ; Journal of Zhejiang University SCIENCE A.

www.zju.edu.cn/jzus; www.springerlink.com

Wan et al. / J Zhejiang Univ SCIENCE A 2006

http://www.zju.edu.cn/jzus/2006/A0612/A061225.pdf