

Nicolas Montavont

2000/2001

Rapport de D.E.A.Informatique

LA MOBILITE DANS LES RESEAUX IP

Université Louis Pasteur de Strasbourg



Maître de stage : Thomas Noël – Maître de conférences

REMERCIEMENTS

Je remercie Thomas Noël, mon maître de stage, pour m'avoir encadré durant mon stage de DEA, et m'avoir fait bénéficier de ses conseils et de son expérience.

Un grand merci à tous les membres de l'équipe de recherche Réseau du LSIIT qui m'ont initié au monde de la recherche.

Enfin, une pensée pour Sébastien et Christelle pour leur soutien.

TABLES DES MATIERES

Introduction.....	1
1 LA MOBILITE	4
1.1 Introduction.....	4
1.2 Terminologie et architecture	4
1.2.1 Modèle OSI.....	4
1.2.2 Nouveaux équipements	5
1.2.3 Adressage	6
1.3 MIPv4 [1].....	7
1.3.1 Découverte des agents de mobilité.....	7
1.3.2 Enregistrement auprès de l’agent mère.....	7
1.3.3 Communication	9
1.4 MIPv6 [2].....	11
1.4.1 Fonctionnalités requises	11
1.4.2 Découverte des routeurs d’accès.....	11
1.4.3 Enregistrement	12
2 HANDOFF ET MICRO MOBILITE.....	15
2.1 Définitions.....	15
2.2 Fast handoff.....	18
2.2.1 Scénario.....	20
2.3 Bi casting.....	22
2.3.1 Bi casting par l’agent mère.....	23
2.3.2 Bi casting réalisé par l’utilisation d’un tunnel	25
2.4 Architecture hiérarchique.....	25
2.4.1 Mobile IP Hiérarchique.....	26
2.4.2 Bi casting dans une architecture hiérarchique.....	28
2.5 Protocoles intégrant le paging.....	28
2.5.1 Cellular IP	28
2.5.1.1 Détail du protocole.....	30
2.5.1.2 Traitement du handoff.....	31
2.6 Conclusion.....	31
3 Présentation du simulateur réseau NS-2.....	34
3.1 Introduction	34
3.2 Architecture et implémentation.....	36
3.2.1 Implémentation du simulateur.....	36
3.2.1.1 Composants de la topologie	37
3.2.1.2 La gestion des files d’attente.....	38
3.2.1.3 Les agents.....	38
3.2.1.4 Le routage.....	40
3.2.1.5 Les réseaux locaux (LAN)	41
3.2.1.6 La mobilité dans NS.....	42
3.3 Moteur du simulateur	47
3.4 Point de vue de l’utilisateur.....	48
3.5 Statistiques et visualisation	50

Table des matières

3.5.1	Système de suivi.....	50
3.5.2	NAM.....	53
3.6	Extension pour la mobilité	57
3.6.1	NOAH	57
3.6.2	CIMS	57
3.6.2.1	Scénario.....	58
3.6.2.2	Tests sur Cellular IP	59
3.6.3	Mobiwan.....	64
3.6.3.1	Génération et manipulation de grandes topologies	66
3.6.3.2	Mobile IPv6 et IPv6	69
3.6.3.3	Mobilité dans un WAN	73
3.7	Conclusion.....	75
4	Gestion d'interfaces multiples.....	77
4.1	Description du projet de gestion d'interfaces multiples sans fil	77
4.1.1	Les objectifs	77
4.1.2	Etat de l'art	78
4.1.3	Cadre du projet et partenaires.....	79
4.2	Présentation de l'architecture MIMP	79
4.3	Normes prises en considérations	81
4.3.1	IEEE 802.11	82
4.3.2	HiperLAN/2	83
4.3.3	Bluetooth	84
4.3.4	Comparatif des trois normes et portée des solutions technologiques.....	85
4.4	Introduction de l'architecture MIMP dans NS-2.....	86
4.4.1	Planification	86
4.4.2	Implémentation d'un nouvel objet	87
4.4.3	Conclusion.....	88
	Conclusion.....	89
	Références bibliographiques	91
	Références Internet	93
	Glossaire	97

LISTE DES ILLUSTRATIONS

Figure 1-1 : modèle des couches OSI.....	5
Figure 1-2 : architecture de base	6
Figure 1-3 : configuration DHCP.....	8
Figure 1-4 : enregistrement v4	8
Figure 1-5 : dés-enregistrement auprès de l’agent mère	9
Figure 1-6 : routage triangulaire : du correspondant au mobile.....	9
Figure 1-7 : routage triangulaire : du mobile au correspondant.....	10
Figure 1-8 : communication v6.....	12
Figure 2-1 : niveaux du handoff.....	17
Figure 2-2 : initialisation du handoff.....	20
Figure 2-3 : enregistrement MIP	21
Figure 2-4 : enregistrement MIPv6	22
Figure 2-5 : bi casting par l’agent mère	24
Figure 2-6 : explicite multicast pour le bi casting.....	25
Figure 2-7 : architecture hiérarchique	26
Figure 2-8 : architecture Cellular IP.....	29
Figure 3-1 : projet VINT	34
Figure 3-2 : structure d’un lien.....	38
Figure 3-3 : structure d’un nœud unicast	39
Figure 3-4 : exemple de composition d’une application.....	40
Figure 3-5 : structure d’un LAN	41
Figure 3-6 : composants d’un nœud mobile (sauf pour DSR)	43
Figure 3-7 : architecture d’un scenario “wired-cum-wireless”	45
Figure 3-8 : structure d’un nœud point d’attache pour MIP	46
Figure 3-9 : diagramme de NAM.....	54
Figure 3-10 : fenêtre d’animation NAM	55
Figure 3-11 : animation de paquets dans NAM	55
Figure 3-12 : simulation souhaitée.....	56
Figure 3-13 : Simulation observée	56
Figure 3-14 : topologie des tests	59
Figure 3-15 : plate-forme de test Cellular IP.....	60
Figure 3-16 : paquets perdus pour un flux CBR	62
Figure 3-17 : numéro de séquence TCP pendant un handoff (mobile récepteur)	63
Figure 3-18 : numéro de séquence TCP pendant un handoff (mobile émetteur).....	64
Figure 3-19 : aires de mobilité locale et de mobilité globale	65
Figure 3-20 : topologie attendue	67
Figure 3-21 : nouvelle structure d’un nœud filaire	70
Figure 3-22 : nouvelle structure d’un nœud mobile ou d’un point d’accès	71
Figure 3-23 : mouvement à l’intérieur d’une grille.....	74
Figure 4-1 : vue générale de l’architecture intégrant MIMP.....	81
Figure 4-2 : topologie utilisée dans IEEE 802.11	83
Figure 4-3 : modèle de référence de HiperLAN/2	84

Liste des illustrations

Figure 4-4 : débit et portée de différentes modes de connexion	86
Figure 4-5 : gestion d'interfaces multiples dans NS-2.....	87

LISTE DES TABLEAUX

Tableau 1 : : format du fichier de sortie .tr.....	51
Tableau 3 : fonctionnalité de l'implémentation CIMS	58
Tableau 4 : comparatif des trois normes	85

INTRODUCTION

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 [19] comme GPRS [20] 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 [21][22][23][24]), offrant un débit plus élevé. Cela devrait permettre d'étendre le domaine d'application.

Les réseaux IP ont été mis en place initialement par l'interconnexion d'hôtes fixes reliés par un réseau filaire. L'objectif était d'offrir une communication rapide à haut débit. De nos jours, on essaye de plus en plus de rendre ces équipements IP mobiles. Différentes technologies de communication sans fil (IEEE 802.11, HiperLAN/2) sont en train de voir le jour. Elles offrent des débits suffisants pour la connexion d'hôtes mobiles à des réseaux IP. Cependant, la gestion des déplacements n'est pas encore suffisamment optimisée pour permettre à ces hôtes mobiles d'exécuter certaines applications en temps réel comme par exemple la vidéo.

La mobilité dans l'Internet a été introduite par l'organisme de standardisation IETF¹ 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

¹ IETF : Internet Engineering Task Force, <http://www.ietf.org>

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.

L'objet de ce stage de DEA est l'étude de terminaux mobiles communicants dans des réseaux IP. L'objectif est de faire un état de l'art complet de toutes les solutions émergentes pour résoudre les problèmes liés à la mobilité des hôtes. Cet état de l'art servira pour des travaux futurs sur l'intra mobilité ; l'intra mobilité est la mobilité interne à un équipement, c'est-à-dire la gestion du changement d'interface communicante. Le stage a duré cinq mois, de février à fin juin 2001. Il s'est décomposé en deux parties équivalentes qui étaient l'étude des solutions pour la mobilité, notamment par le suivi des groupes de travail Seamoby et MobileIP de l'IETF, et l'étude du simulateur réseau NS-2.

Le rapport est structuré de la manière suivante : tout d'abord une première partie est consacrée à l'étude du protocole Mobile IP [1][2] qui gère le déplacement des hôtes mobiles dans les réseaux IP. Les versions 4 (actuellement utilisé dans l'Internet) et 6 (en train d'être mise en place) seront étudiées. Ensuite, les optimisations à Mobile IP envisagées seront exposées dans la partie 2. Ces optimisations concernent l'amélioration des techniques de déplacement pour minimiser les impacts sur les applications. Ces améliorations sont essentiellement présentées par des groupes de travail de l'IETF. Puis, le simulateur réseau NS-2 (Network Simulator) sera étudié. NS-2 est le simulateur le plus utilisé dans la communauté de recherche en réseau et permet de simuler différentes caractéristiques de protocoles en cours de réalisation. On s'intéresse de très près à NS car nos travaux futurs concerneront dans un premier temps l'intégration d'une nouvelle architecture dans NS-2. Le projet de cette nouvelle architecture nommée MIMP² sera présenté dans une quatrième et dernière partie.

² MIMP : Multiple Interfaces Management Protocol, protocole de gestion d'interfaces multiples.

PARTIE 1

LA MOBILITE

1 LA MOBILITE

1.1 Introduction

Le terme de mobilité définit la situation intermédiaire entre le nomadisme et les réseaux ad hoc. Le nomadisme représente le déplacement d'un équipement IP³ [29] 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'opposé, 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** [1][2]. 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 [29].

1.2 Terminologie et architecture

Dans cette partie sont décrits les principaux termes utilisés dans Mobile IP [1][2]. De plus, une architecture type y sera exposée pour une meilleure vision des fonctions de chaque équipement. Pour de plus amples définitions, voir le glossaire.

1.2.1 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 OSI⁴ (Interconnexion des Systèmes Ouverts) en 7 couches [26]. Ces sept couches sont décrites dans la Figure

³ IP : Internet Protocol

⁴ OSI : Open Systems Interconnexion – Interconnexion des systèmes ouverts

1-1.

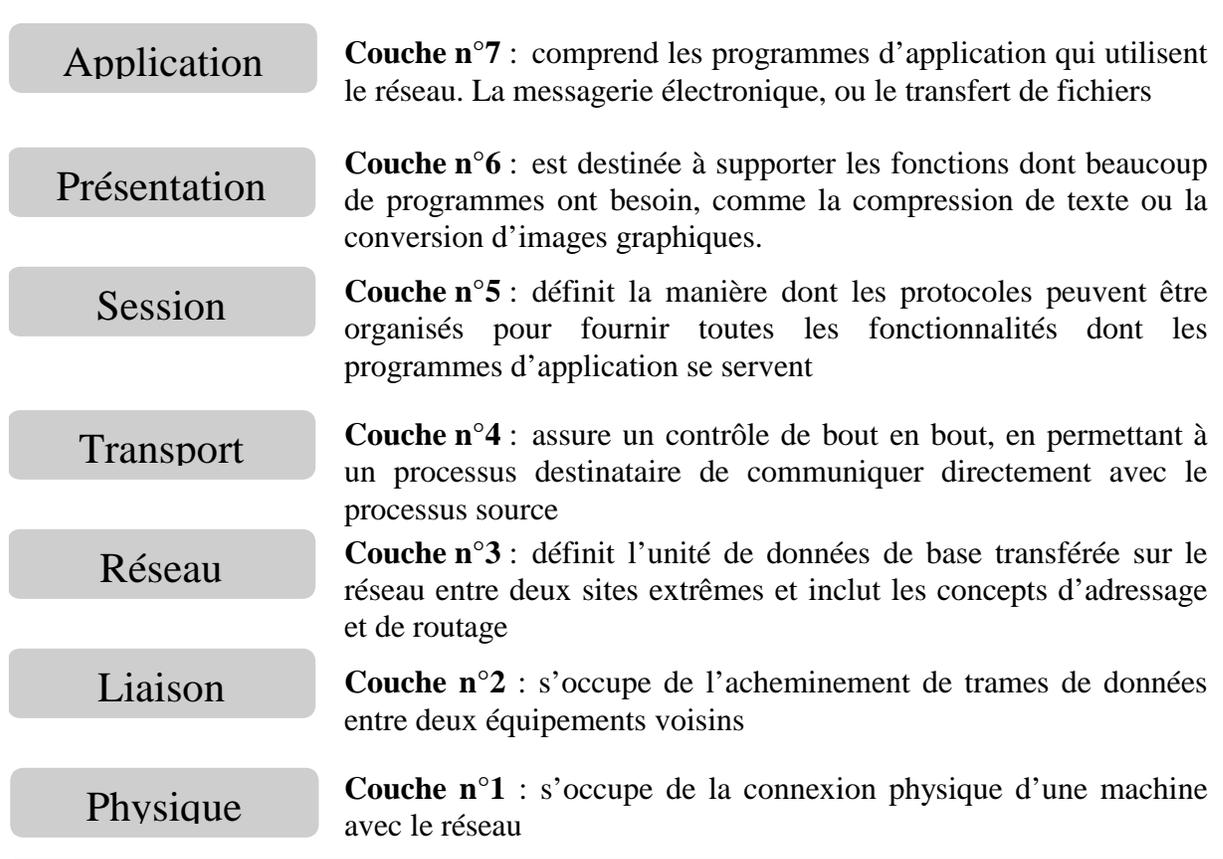


Figure 1-1 : modèle des couches OSI

L'ensemble de ce rapport s'appuie sur l'architecture OSI de la Figure 1-1. On s'intéressera plus particulièrement aux couches liaison et réseau. La couche liaison car elle 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. L'objet de ce rapport est principalement de constater et de proposer des solutions aux impacts causés par la mobilité sur le fonctionnement de ce protocole. On parlera aussi quelque fois des couches transport et application pour tester les effets des protocoles de la couche réseau sur les couches supérieures.

1.2.2 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 [35][36] 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 :

- **Noeud 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 noeud mobile (dans le réseau mère du mobile)
- **Agent visité** : routeur d'accès avec une interface sur le lien courant du noeud mobile (dans un réseau visité). Cet agent n'existe que dans Mobile IPv4.
- **Point d'accès** : équipement intermédiaire entre le réseau filaire et le noeud mobile qui offre la connexion aux noeuds mobiles qui lui sont rattachés

L'architecture de base des ces équipements est présentée dans la **Figure 1-2**.

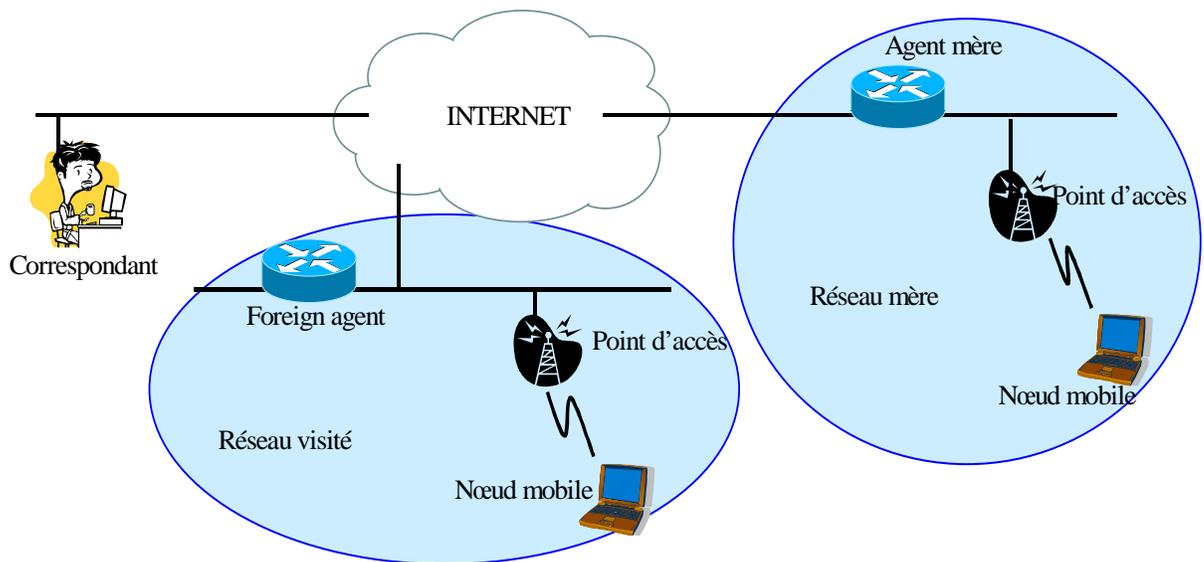


Figure 1-2 : architecture de base

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 nœuds 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.

1.2.3 Adressage

Un noeud mobile peut avoir plusieurs adresses simultanément. L'adresse qu'un noeud

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 TCP⁵ [30] 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 ré-initialisée 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.

A présent étudions plus spécifiquement le fonctionnement des protocoles MIPv4 [1] et MIPv6 [2].

1.3 MIPv4 [1]

1.3.1 Découverte des agents de mobilité

Comme on l'a souligné plus haut, une caractéristique propre au mobile est de pouvoir se déplacer en cours d'une communication. 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** [1] 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.

1.3.2 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 DHCP⁶ [32] (voir Figure 1-3).

⁵ TCP : Transmission Control Protocol

⁶ DHCP : Dynamic Host Configuration Protocol

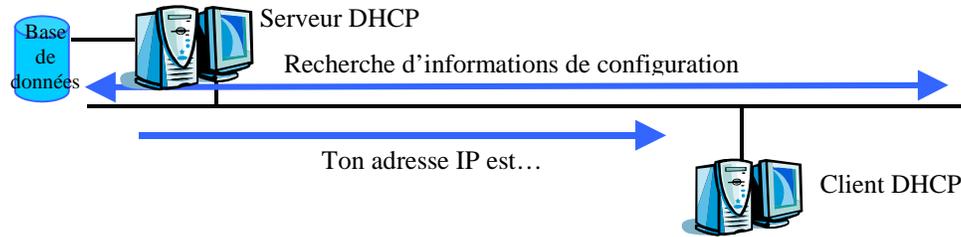


Figure 1-3 : configuration DHCP

Une fois que le nœud mobile a une adresse temporaire valide, il émet un message *Registration Request* (étape 1 dans la **Figure 1-4**) en indiquant la correspondance entre son adresse principale et son adresse temporaire et éventuellement d'autres options. Ce message passe par l'agent visité qui le transmet à l'agent mère du mobile s'il accepte les requêtes du nœud mobile. L'agent mère doit acquiescer le *Registration Request* pour bien confirmer la réception (message UDP⁷ [31]) 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 mettent à jour leur cache d'association pour ce nœud mobile.

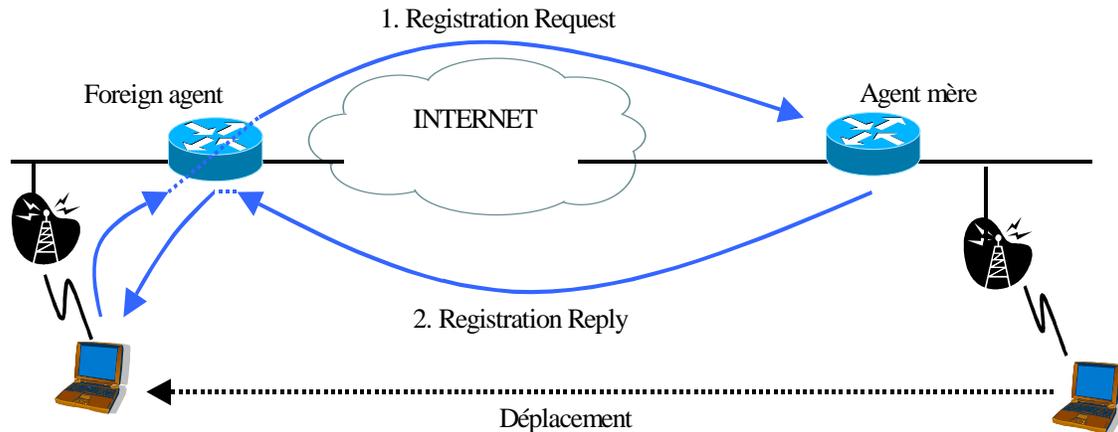


Figure 1-4 : enregistrement v4

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**

⁷ UDP : User Data Protocol

1-5 jusqu'à ce qu'il reçoive un *De-Registration 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.

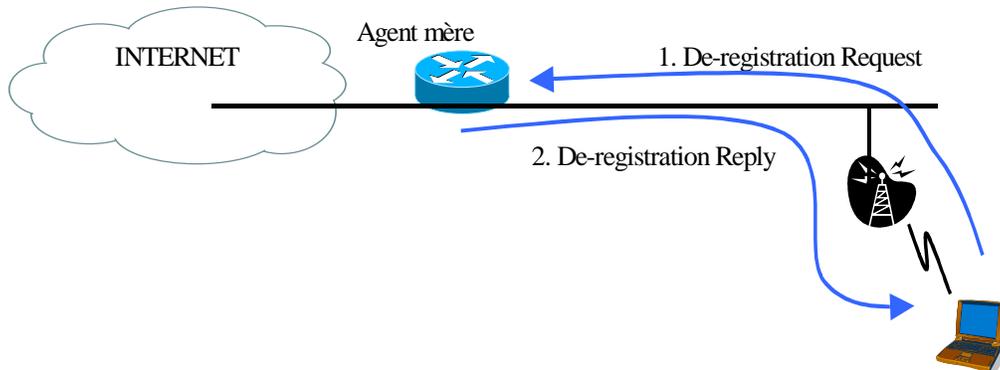


Figure 1-5 : dés-enregistrement auprès de l'agent mère

1.3.3 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 1-6**).

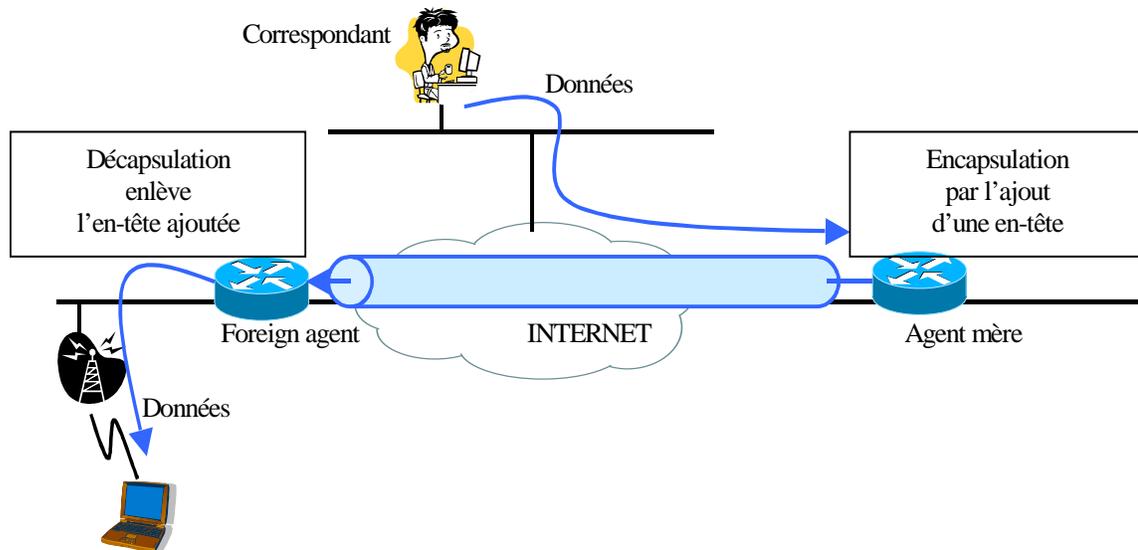


Figure 1-6 : routage triangulaire : du correspondant au mobile

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 (ingress filtering [37]). 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 1-7).

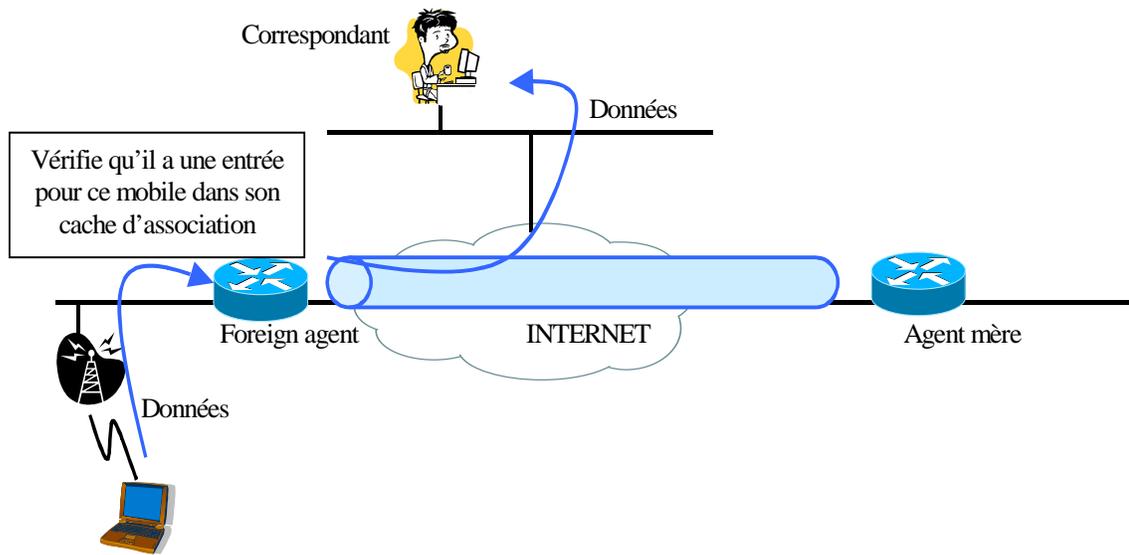


Figure 1-7 : routage triangulaire : du mobile au correspondant

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 [27][28] sur le réseau principal afin de faire correspondre l'adresse IP du mobile avec son adresse MAC [27]. 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 tunneller 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 1-4** et **Figure 1-6**.

1.4 MIPv6 [2]

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.

1.4.1 Fonctionnalités requises

Dans MIPv6, 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 œuvre 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).

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.

1.4.2 Découverte des routeurs d'accès

Le protocole de **découverte des voisins** [33] 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 Advertisement* 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 noeuds 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 [34]. 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.

1.4.3 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 « piggy-backing⁸ ». En réponse à une telle requête, l'agent mère envoie un *Binding Acknowledgement* 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 transité 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 1-8**).

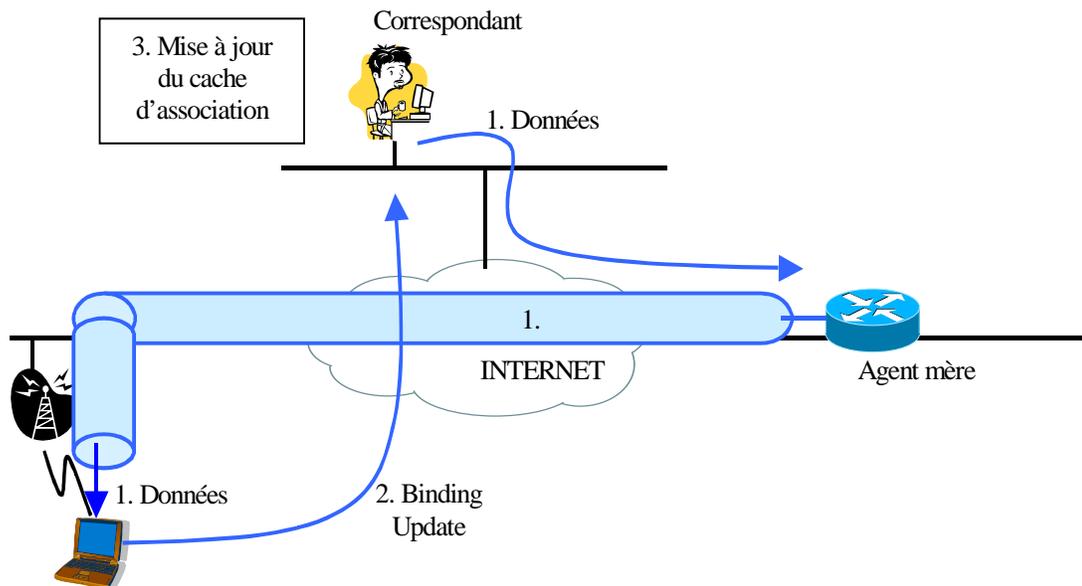


Figure 1-8 : communication v6

⁸ Piggy-backing : fait de mettre des informations de control dans des paquets de données

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.

PARTIE 2

HANDOFF ET MICRO MOBILITE

2 HANDOFF ET MICRO MOBILITE

L'accès sans fil risque de devenir de plus en plus fréquent dans les prochaines années. Les utilisateurs mobiles demanderont évidemment des niveaux de qualité de service identiques à ceux offerts aux utilisateurs de postes fixes. D'un autre côté, le protocole utilisé actuellement pour gérer la mobilité IP (MIP) [1][2] fait cruellement ressentir ses limites dans un tel environnement. Une telle vision présente donc un certain nombre de défis techniques pour la mobilité IP en terme de performance et de mise à l'échelle.

C'est donc pour cela que récemment un certain nombre de protocoles de micro-mobilité visant à améliorer les performances de MIP ont été présentés dans le groupe de travail de l'IETF⁹. Ces protocoles de micro-mobilité sont conçus pour des environnements où les nœuds mobiles changent leur point d'attache à l'Internet si fréquemment que le tunneling de MIP s'avère insuffisant : surcharge de la signalisation, perte de paquets, livraison des données aux applications retardées. Ces retards sont directement liés au temps d'aller-retour des messages d'enregistrement. Les protocoles de micro-mobilité aspirent à gérer les déplacements locaux (à l'intérieur de domaines) des nœuds mobiles sans interagir avec MIP, c'est-à-dire cacher au reste de l'Internet les mouvements des nœuds mobiles à l'intérieur d'un domaine. Cela a l'avantage de réduire le retard et la perte des paquets pendant un déplacement et élimine l'enregistrement entre un nœud mobile et son agent mère qui peut être éloigné.

Par ailleurs, la signalisation engendrée par la gestion de la mobilité s'accroît avec le nombre d'utilisateurs. Dans les réseaux cellulaires, l'enregistrement et les techniques de pagination¹⁰ sont employés pour réduire au maximum la signalisation et optimiser les performances de la gestion de la mobilité. Actuellement, MIP supporte l'enregistrement mais pas la pagination. Une caractéristique importante des protocoles de micro-mobilité (cf. section 2.1) est leur capacité à réduire la signalisation liée aux migrations fréquentes des mobiles et de réduire la consommation des équipements en tenant compte du mode opérationnel de cet équipement (actif ou inactif). Le support d'une « connectivité passive » à l'Internet (par une localisation approximative) s'avère impérative puisqu'il permet de réduire la charge notamment sur les interfaces aériennes.

Dans ce qui suit, on commencera par définir les termes liés à la micro-mobilité et au handoff. Ensuite, on verra les solutions émergentes visant à résoudre les problèmes décrits ci-dessus : fast handoff [4][5], bi-casting [4][5], MIP hiérarchique [11][12], puis une approche plus proche des réseaux cellulaires avec Cellular IP [14][15][16][17] qui intègrent la pagination.

2.1 Définitions

Dans cette partie sera décrit la terminologie liée au handoff [19] pour expliciter au

⁹ IETF : Internet Engineering Task Force, <http://www.ietf.org>

¹⁰ Pagination : signalisation mise en œuvre pour la localisation des nœuds mobiles. Le protocole de localisation est dépendant du type de liaison utilisée.

mieux ce que chaque protocole essaie de résoudre. Les termes employés dans ce rapport sont ceux utilisés par le groupe de travail de l'IETF, principalement décrits dans [3].

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 présenté partie 1 section 1.2.1. 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 présenté dans la première partie) 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 2-1** ci-dessous.

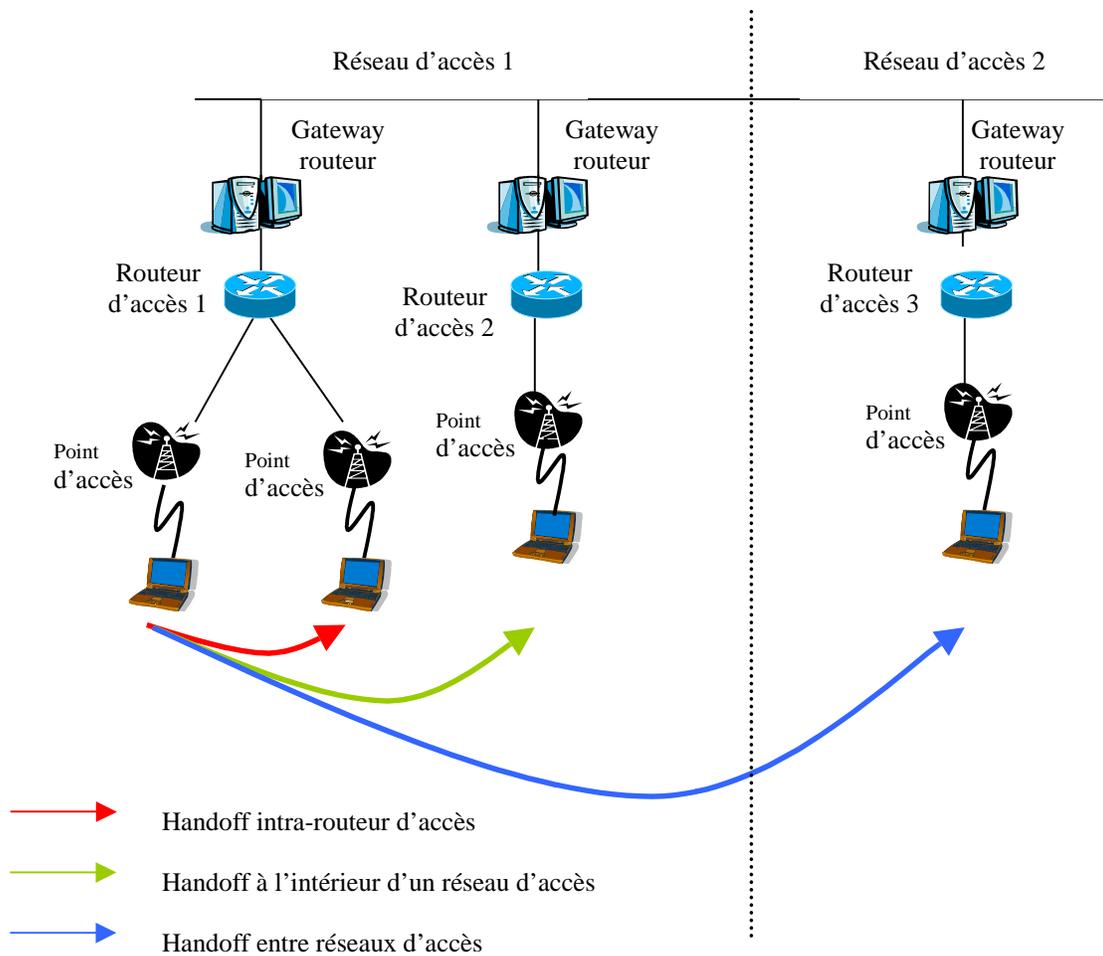


Figure 2-1 : niveaux du handoff

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 une aire ou 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.

Une plus ample terminologie est donnée dans le glossaire. Par la suite, on sera encore amené à définir quelques termes spécifiques à chaque protocole. Dans un souci de clarté, dans le cas de MIPv4 [1], 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 [2], on parle d'ancien routeur d'accès (RA) et de nouveau RA.

2.2 Fast handoff

Le support du fast handoff [4][5], 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 de 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 IP¹¹ et Seamoby¹² 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

¹¹ Mobile IP : groupe de travail de l'IETF cherchant à normaliser des protocoles pour optimiser Mobile IP.

¹² Seamoby : groupe de travail de l'IETF cherchant à normaliser des protocoles de micro-mobilité.

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. Ce point sera mis plus en avant dans la partie 3 du rapport puisqu'il constitue l'objet de mes futurs travaux.

Dans un souci d'efficacité, l'anticipation peut concerner plus d'un agent de mobilité [4]. 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 entrée 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.

2.2.1 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'autoconfiguration d'adresse sans état, comme c'est la plupart du temps le cas aujourd'hui (le cas d'autoconfiguration 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 2-2**) quand il détecte qu'un handoff va avoir lieu. Comme dit précédemment, le nœud mobile inclut 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.

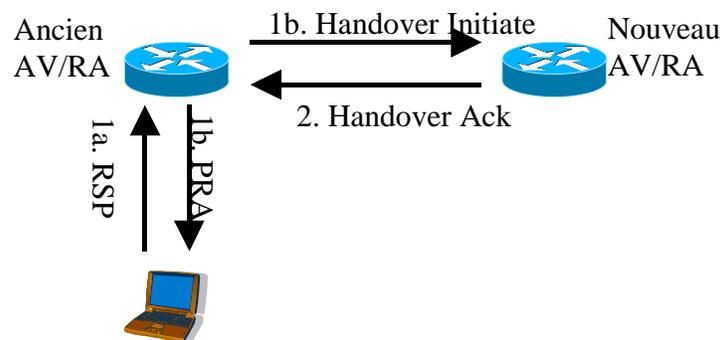


Figure 2-2 : initialisation du handoff

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 2-3**) 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.

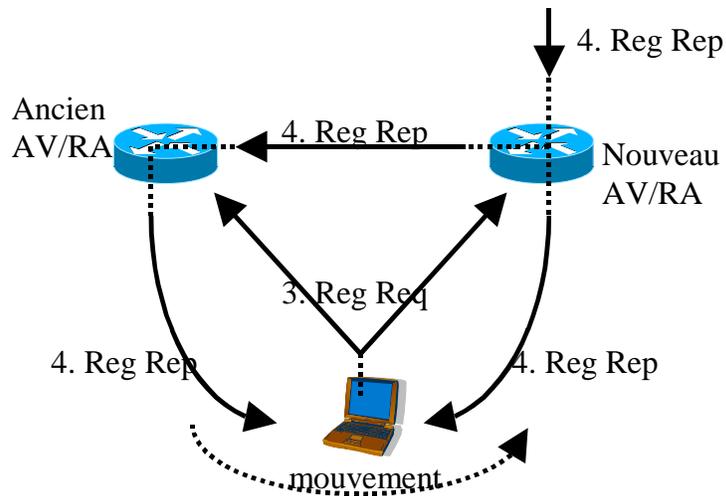


Figure 2-3 : enregistrement MIP

Dans le cas de MIPv6, le nœud mobile envoie un *Binding Update* (étape 3 dans la **Figure 2-4**) 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'acquiescement, il doit renvoyer un *Binding Update* à l'ancien routeur d'accès et un autre à son agent mère et ses correspondants.

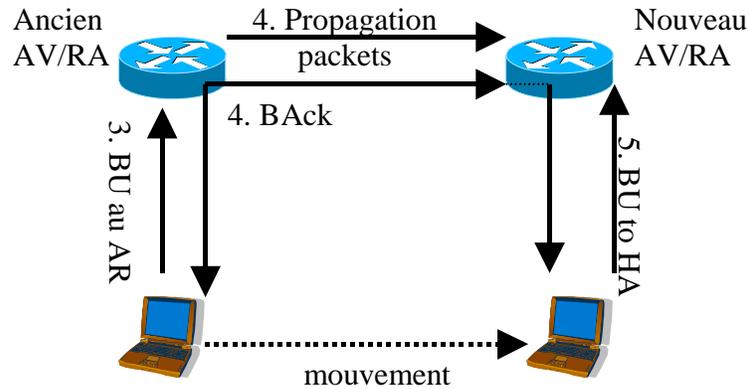


Figure 2-4 : enregistrement MIPv6

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 créé, 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 partagea avec l'ancien routeur. Les détails du transfert de contexte ne font pas l'objet de ce présent rapport. De plus amples informations peuvent être trouvées dans [6][7][8].

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.

2.3 Bi casting

Le bi casting [4] 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 utilisée en plus du protocole de fast handoff décrit ci-dessus pour envoyer des copies du trafic aux potentielles localisations du nœud mobile [4][5]. 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 » [4] ; 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 utilisés 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.

2.3.1 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 2-5**).

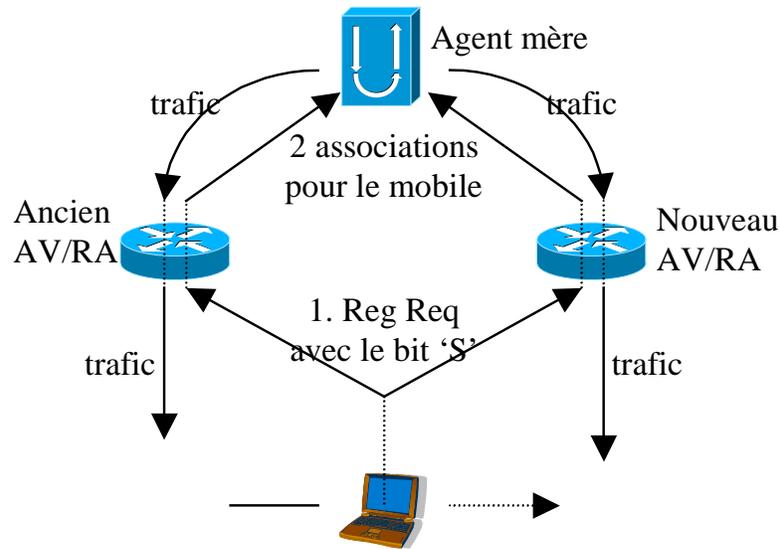


Figure 2-5 : bi casting par l'agent mère

Bien que le bi casting semble être efficace et réduise le nombre de paquets perdus, cette manière de le mettre en œuvre 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 [9]. Cependant, la gestion de groupe doit être légère, dynamique et l'ajout ou le retrait d'une adresse destinataire doit se faire rapidement. La technique du **Small Group Multicast** (SGM) [10] 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 2-6**).

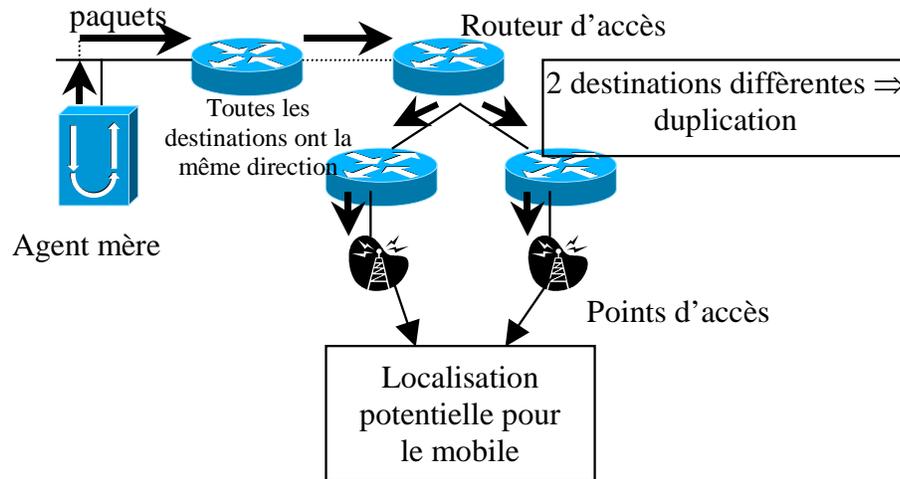


Figure 2-6 : explicite multicast pour le bi casting

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.

2.3.2 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 (section 2.2) 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 expiré 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.

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.

2.4 Architecture hiérarchique

L'objectif de mettre en place une hiérarchie 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** [11][12]. Un domaine est défini comme étant une aire de mobilité locale [3]. Généralement un domaine est indépendant des sous-réseaux et sa taille est choisi par l'opérateur réseau.

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 [11], il s'agit d'un **agent visité passerelle** et dans MIPv6 [12] c'est un **Mobility Anchor Point**. Ces points d'ancrage sont des agent visités dans MIPv4 et des routeurs d'accès dans MIPv6 au sommet de plusieurs agent visités (resp. routeurs d'accès), avec une adresse IP publique (voir **Figure 2-7**).

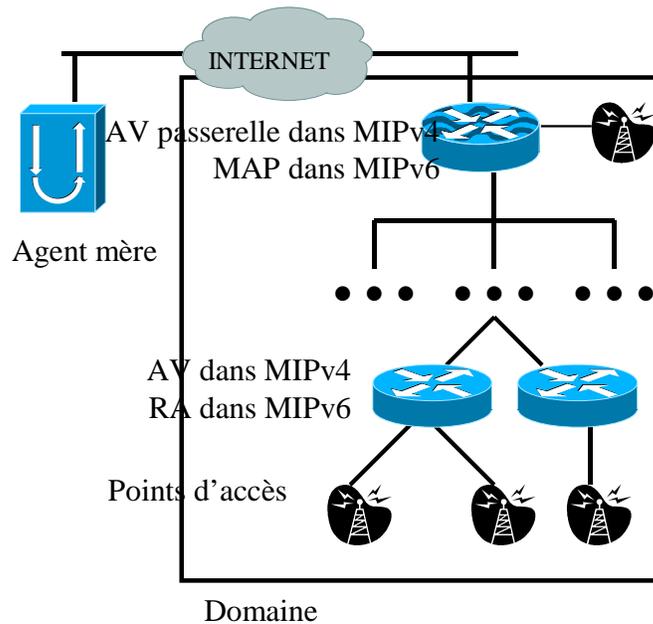


Figure 2-7 : architecture hiérarchique

Cette définition permet de mettre en œuvre une hiérarchie avec plus d'un niveau. Lorsqu'un nœud mobile entre dans un domaine, un tunnel est créé 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 bi-directionnel, il lui faut positionner un bit 'T' dans son message d'enregistrement.

Dans la prochaine sous-section, le fonctionnement de la hiérarchie et l'échange de messages occasionnés seront explicités. Ensuite dans une dernière sous-section, on verra comment réaliser le bi casting dans une telle architecture.

2.4.1 Mobile IP Hiérarchique

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.

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 2-7** 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.

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 tunnelle 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 soucis de mise à l'échelle ou pour des raisons de gestion, tous les nœuds mobiles ne peuvent pas acquérir leur propre adresse régionale. Dans le mode étendu, l'adresse temporaire régionale est celle du Mobility 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 une 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 une enregistrement locale à un AV/RA n'implémentant pas MIP hiérarchique aura la possibilité de faire une enregistrement telle que décrite dans MIP [1][2].

2.4.2 Bi casting dans une architecture hiérarchique

Comme il a été montré dans la section 2.3, 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 régionales. 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.

Seuls les messages définis dans la section 2.3 sont nécessaires au nœud mobile pour demander le bi casting. 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.

2.5 Protocoles intégrant le paging

Typiquement, les hôtes connectés à l'Internet (par exemple, des ordinateurs de bureau connectés à un réseau local) restent en ligne pendant plusieurs heures bien que la plupart du temps ils ne communiquent pas. Être "toujours connecté" de cette manière aboutit à être accessible à tout instant et de pouvoir accéder aux ressources Internet. Les utilisateurs mobiles connectés à l'Internet sans fil s'attendent à un service semblable. Malheureusement, le maintien de l'information de localisation par les mobiles pour rester continuellement accessible, exigeraient des mises à jour de localisations fréquentes, ce qui consommerait de la bande passante précieuse et l'énergie de la batterie. Ce surplus de signalisation et cette consommation électrique d'hôte mobile peut être réduite par l'introduction de la pagination. Généralement, les hôtes mobiles fonctionnent sur des batteries de faible autonomie, c'est pourquoi il s'avère impératif d'éviter que les hôtes mobiles inoccupés n'aient à transmettre des messages de mise à jour de localisation fréquemment. Cela exige l'appui explicite de protocoles réseau, comme la capacité de suivre la localisation des mobiles de manière approximative et la capacité de paginer des hôtes mobiles inactifs. Des hôtes mobiles inactifs ne doivent pas enregistrer leur déplacement dans la même aire de pagination pour ne signaler que leur changement d'aire de pagination. La pagination a été mise en oeuvre dans un certain nombre de protocoles de micro-mobilité dont Cellular IP [14][15][16][17] et Hawaii [18]. Cellular IP est présenté ci-dessous.

2.5.1 Cellular IP

Le protocole Cellular IP [14][15][16][17] 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 nœuds mobiles 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 [14] ou 6 [17] (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 nœuds cellular IP, qui sont des points d'accès et des nœuds mobiles implémentant le protocole Cellular IP (voir **Figure 2-8**). 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. Bien que cette structure soit bien adaptée pour le fast handoff, des traitements spécifiques pour le seamless handoff ont été mis en place tel que le **semi-soft handoff** [14][16] ou le **indirect semi-soft handoff** [17], qu'on détaillera plus tard.

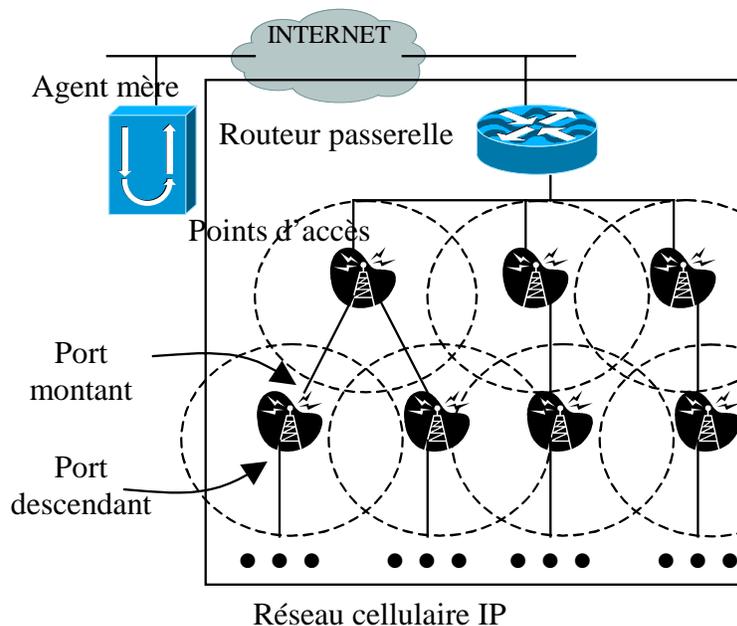


Figure 2-8 : architecture Cellular IP

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 de le 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.

La localisation des nœuds 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 contrôle est effectué lors de l'entrée des nœuds mobiles dans le réseau cellulaire IP et seuls les messages de contrôle peuvent créer des nouvelles entrées dans les caches pour assurer une sécurité.

2.5.1.1 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 2-8**). 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 2-8**). Les points d'accès envoient aussi des beacons¹³ 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 une enregistrement mère. Dans le cas de MIPv4, l'adresse indiquée dans cette enregistrement est celle du routeur passerelle, alors que dans MIPv6, c'est une adresse temporaire locale créée à partir du préfix réseau.

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* 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

¹³ Beacon : message de synchronisation envoyé périodiquement par les points d'accès, contenant des informations pour l'identification.

pagination est faite par l'envoi 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 toutes les nouvelles points d'accès sur le chemin allant du nœud mobile au routeur passerelle. Les opérations spécifiques au handoff sont détaillés dans la sous-section suivante.

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.

2.5.1.2 Traitement du handoff

Le changement de point d'accès, appelé **hard handoff**, 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.

2.6 Conclusion

On vient de voir un ensemble de protocoles de gestion des handoffs IP. Le but final est d'atteindre le seamless handoff, c'est-à-dire que les déplacements ne se fassent pas

ressentir dans les applications en cours. Nous avons vu que le fast handoff permettait de réduire le temps d'attache au nouveau routeur d'accès, que le bi casting permettait de réduire la perte des paquets et que le modèle hiérarchique réduisait la signalisation par une gestion locale de la mobilité. Bien que le bi casting semble être une bonne solution et très adaptable, sa réalisation est difficile à grande échelle.

L'ensemble de cette partie a fait l'objet d'un article qui est en cours de soumission [25].

On peut remarquer qu'aucun de ces protocoles ne traitent les problèmes inter-technologies, c'est-à-dire le passage d'une technologie à une autre sans forcément se déplacer. La gestion d'interface sera l'objet de la partie 4, gestion qu'on désirerait intégrer dans un premier temps dans le simulateur réseau NS-2 présenté dans la partie 3.

PARTIE 3

PRESENTATION DU SIMULATEUR RESEAU NS-2

3 Présentation du simulateur réseau NS-2

3.1 Introduction

Le simulateur réseau NS (Network Simulator) est un simulateur à événements discrets orienté objet, basée sur le simulateur réseau REAL¹⁴. Au départ, la version 1.0 de NS a été développée au Laboratoire National de Lawrence Berkeley¹⁵ (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 [46]

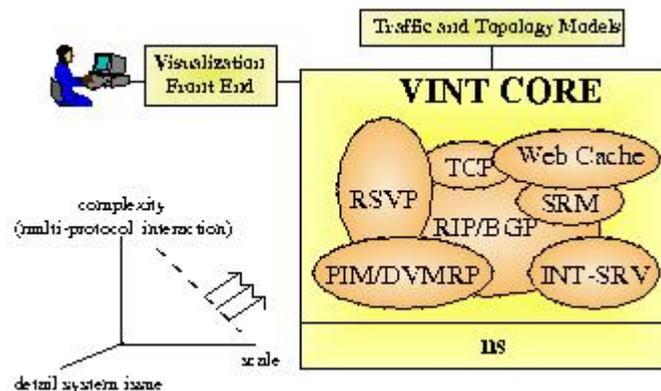


Figure 3-1 : projet VINT

Le projet VINT¹⁶ (Virtual InterNetwork Testbed) est dirigé par l'université de Californie du sud et est financé par le DARPA¹⁷ en collaboration avec Xerox PARC¹⁸ 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

¹⁴ Simulateur réseau REAL : <http://www.cs.cornell.edu/skeshav/real/overview.html>

¹⁵ LBNL : <http://www-nrg.ee.lbl.gov/>

¹⁶ Projet VINT : <http://www.isi.edu/nsnam/vint/index.html>

¹⁷ DARPA : <http://www.darpa.mil>

¹⁸ Xerox PARC : <http://www.parc.xerox.com/parc-go.html>

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 devient 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 NAM¹⁹ 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 de faire varier le 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 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.

Dans ce qui va suivre, on étudiera tout d'abord la structure interne du simulateur, son

¹⁹ « Network Animator », outil d'animation pour visualiser les résultats de simulations et des traces des paquets

implémentation de manière générale et le moteur du simulateur. Ensuite, une partie sera consacrée à son utilisation pour mieux s'apercevoir des fonctionnalités, avec un premier aperçu de scripts de simulation. Enfin, on exposera les outils d'interprétation et de visualisation offerts avec le simulateur. Enfin, on s'intéressera plus particulièrement à la mise en œuvre de la mobilité ; trois extensions au modèle de base ont été implémentées par des équipes de recherche. Après avoir expliquer en quoi consistent ces apports, quelques tests seront exécutés sur ces extensions dans une dernière partie.

3.2 Architecture et implémentation

L'architecture réseau de NS-2 est fortement basée sur le modèle des couches OSI. Ce modèle a brièvement été expliqué dans la partie 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.

La dernière version de NS-2 est sortie le 7 juin 2001 dans sa version 8 (ns-2.1b8). Les sources sont disponibles sur le site ISI²⁰ dans la section « nsnam » [<http://www.isi.edu/nsnam/ns>]. Les sources se présentent sous deux formes : l'une dite « tout en un » 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 OTcl²¹ ; Le paquage inclus 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 créé 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...).

3.2.1 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. 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.

²⁰ ISI : Information Sciences Institute

²¹ OTcl : extension objet du langage interprété Tcl

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'événements (voir 3.). 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. Ces éléments sont décrits dans la sous section suivante.

3.2.1.1 Composants de la topologie

La topologie 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 3-3**). 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 permette 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 définit cinq variables clés (représentée dans la **Figure 3-2**) :

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

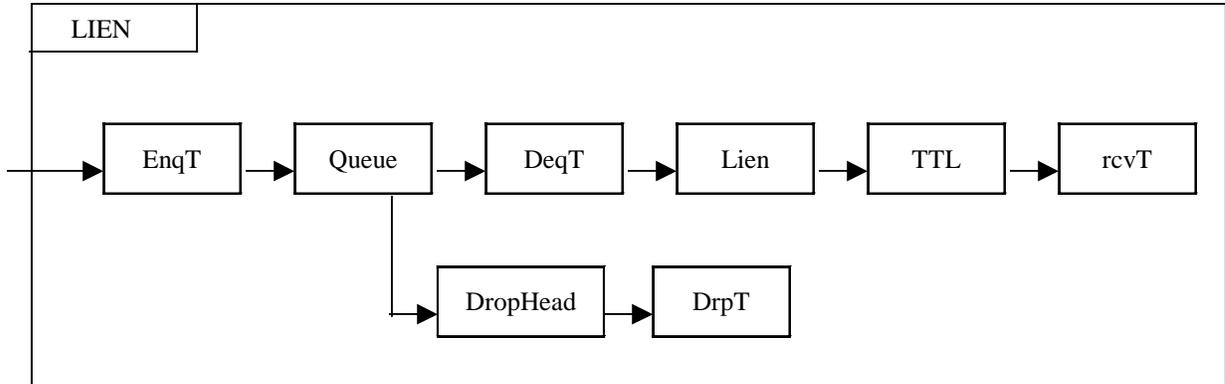


Figure 3-2 : structure d'un lien

Note sur la Figure 3-2 : 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 mis en place autour de la topologie pour faire interagir les nœuds entre eux.

3.2.1.2 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.

3.2.1.3 Les agents

A un niveau supérieur, on retrouve les agents (*classe Agent*) 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 [30], UDP [31], RTP...). La **Figure 3-3** montre les interactions entre ces différents composants dans NS.

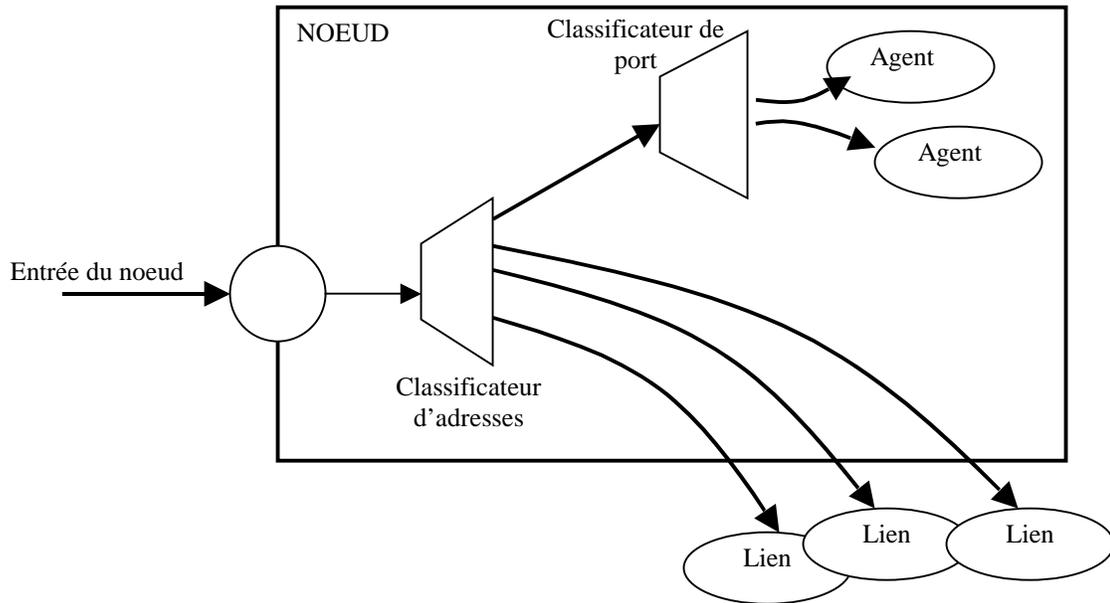


Figure 3-3 : structure d'un nœud unicast

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/OFF²² 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 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 3-4**.

²² Trafic ON/OFF : un nœud qui génère du trafic ON/OFF est un nœud qui alternativement émet des paquets et stoppe son émission.

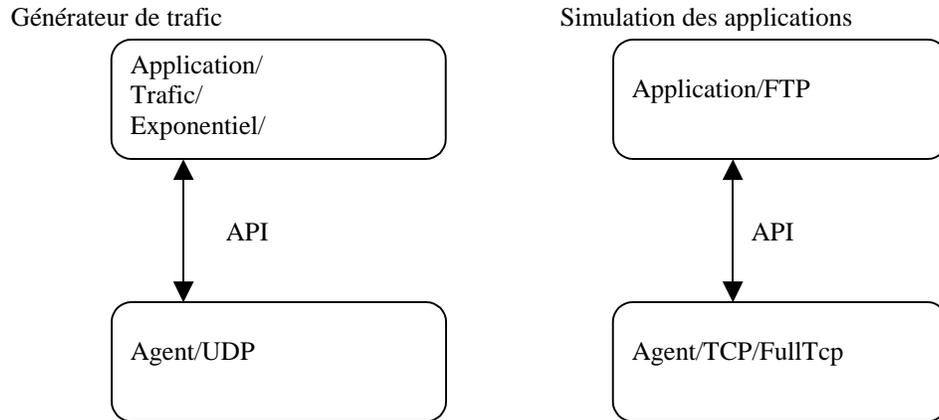


Figure 3-4 : exemple de composition d'une application

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.

3.2.1.4 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 :

- Routage statique : routage utilisé par défaut suivant l'algorithme SPF de Dijkstra [43]. 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 [44] (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'envoi 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.

3.2.1.5 Les réseaux locaux (LAN)

Pour permettre des simulations à plus grande échelle, NS permet d'utiliser la notion de LAN²³. 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 3-5**).

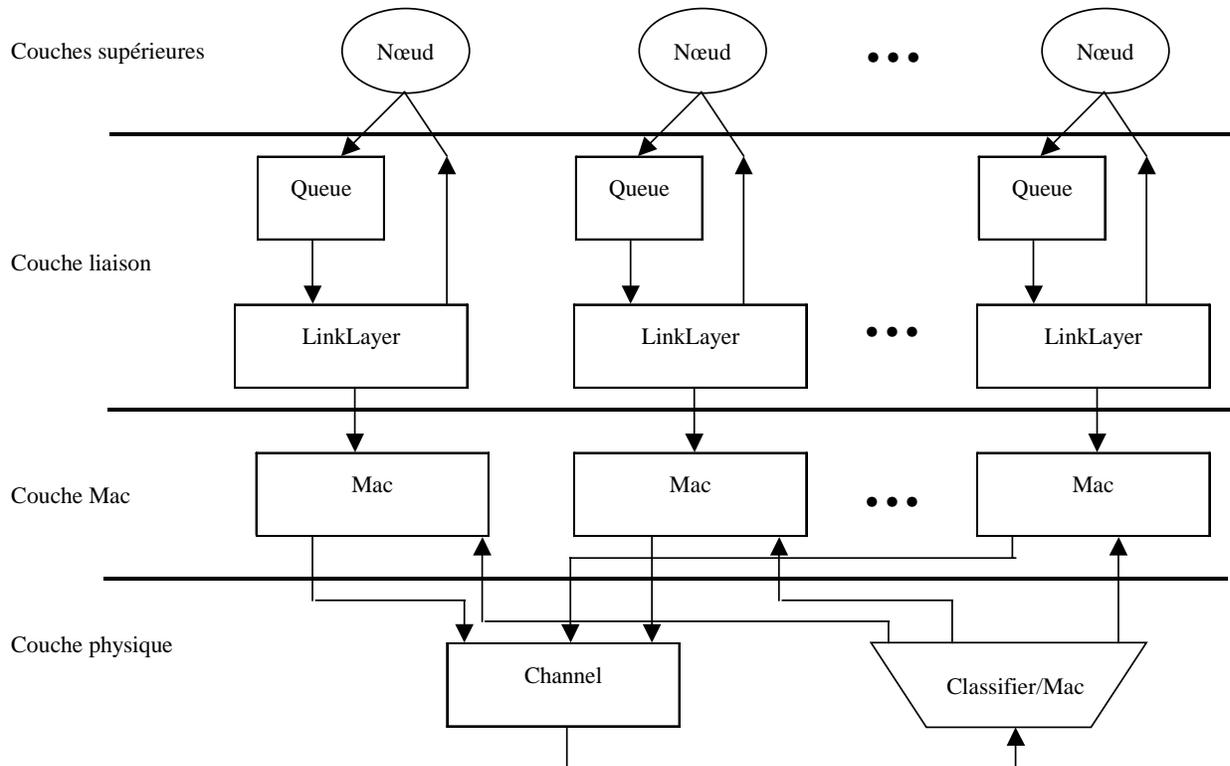


Figure 3-5 : structure d'un LAN

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 réplique 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 créé 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.

²³ Local Area Network : réseau local

3.2.1.6 La mobilité dans NS

Dans un premier temps, la mobilité a été introduite dans NS-2 par les chercheurs de l'université Cartegie Mellon²⁴ de Pittsburgh (CMU) 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 » [44]. A chaque nœud, une copie séparée de TORA tourne pour chaque destination. TORA opère au-dessus de IMEP²⁵ 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 nœud mobile est créé 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 3-6**.

²⁴ CMU : <http://www.cmu.edu>

²⁵ IMEP : Internet MANET Encapsulation Protocol

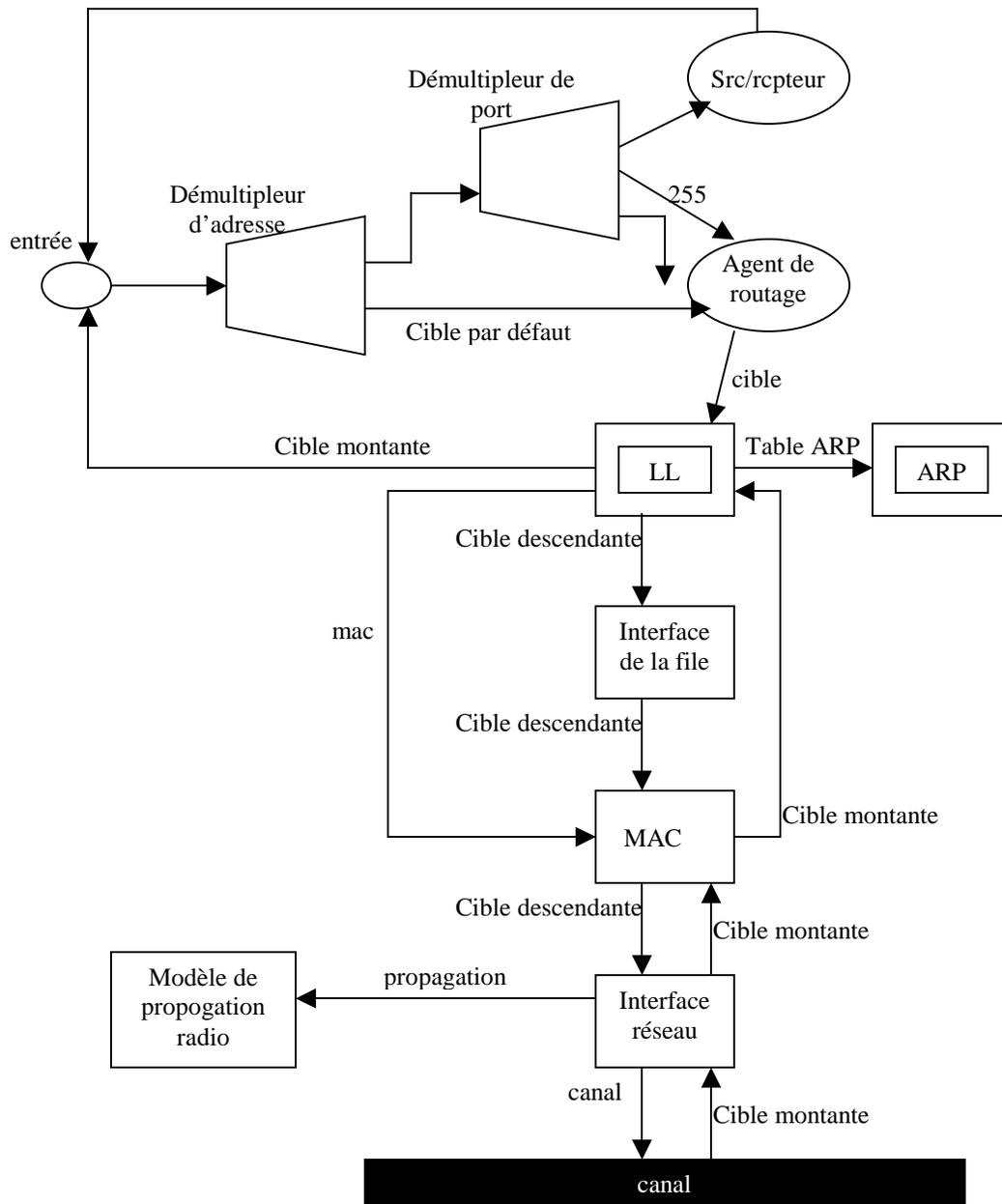


Figure 3-6 : composants d'un nœud mobile (sauf pour DSR)

La **Figure 3-6** 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 démultiplexeur 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, IBM²⁶ a développé une extension Bluehoc²⁷ qui implémente la norme Bluetooth dans la version 6 de NS-2. Cette norme sera brièvement présentée dans la partie 4.
- 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}. Ces interfaces sont soumises aux collisions et enregistrent l'intensité du signal, la longueur d'onde...
- Le modèle de propagation radio utilisé est l'atténuation espace-Friss ($\frac{1}{r^2}$) pour les courtes distances et Two Ray Ground ($\frac{1}{r^4}$) pour les grandes distances. L'antenne simulée est une antenne omnidirectionnelle.

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

²⁶ IBM : <http://www.ibm.com/>

²⁷ Bluehoc : <http://oss.software.ibm.com/developerworks/opensource/bluehoc/>

hoc uniquement. C'est pourquoi une première extension a été ajoutée au modèle, toujours fondée sur le travail des chercheurs de CMU, 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 créé pour assurer la liaison entre le réseau filaire et le réseau sans fil. Ce nœud est appelé *BaseStationNode* (représenté dans la **Figure 3-7**). 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 3-7**.

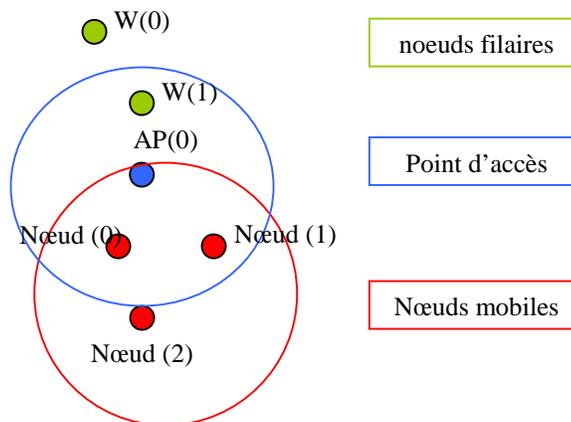


Figure 3-7 : architecture d'un scénario "wired-cum-wireless"

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 mères, des agents visités et des hôtes mobiles qui se déplacent de l'un à l'autre, comme présenté dans la partie 1. Les agents mères et les agents visités sont grossièrement des points d'accès comme ceux décrits plus haut. Ils sont définis dans

²⁸ Sun microsystem : <http://www.sun.com>

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 3-8**. 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 3-8**, sans l'encapsuleur et le décapsuleur.

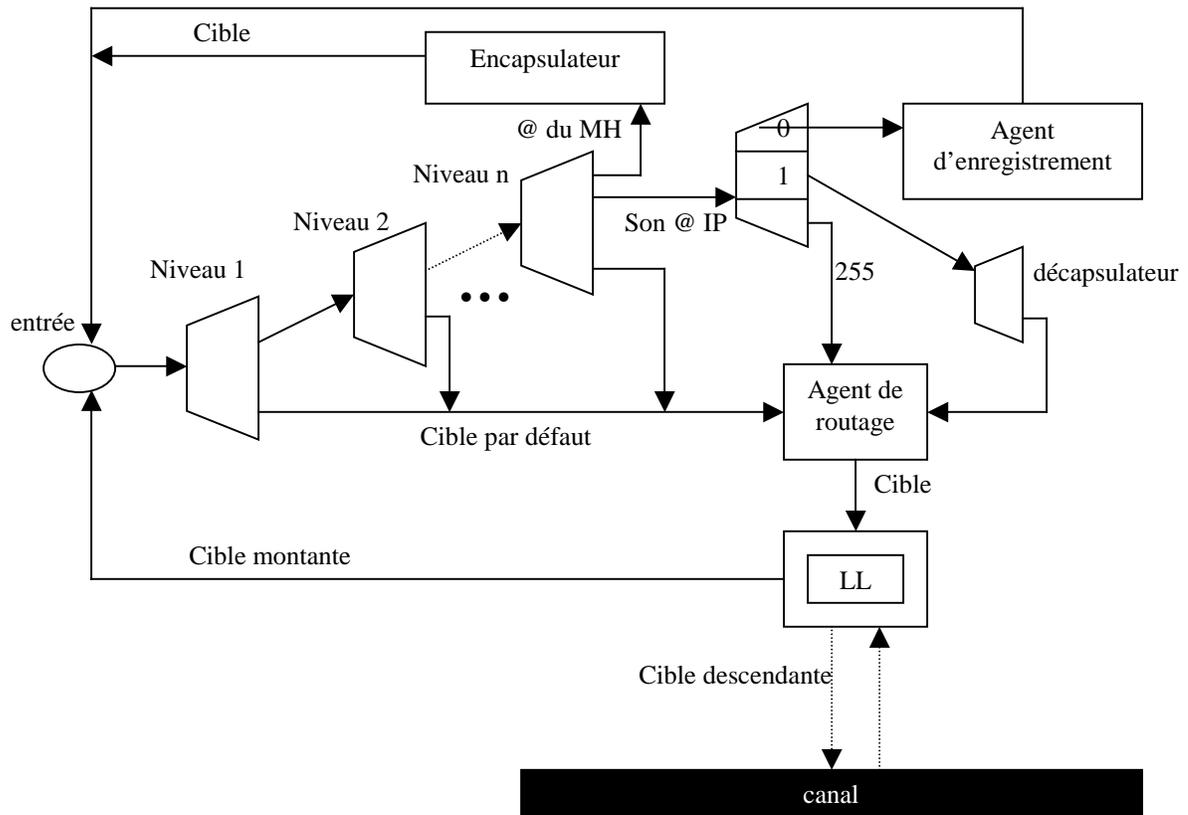


Figure 3-8 : structure d'un nœud point d'attache pour MIP

Pour s'apercevoir de la portée et du niveau de simulation qu'il est possible de 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 [38] dans NS-2 de base.

3.3 Moteur du simulateur

Le moteur de simulation est extensible, configurable et programmable. La mise en oeuvre actuelle est un chaînage simple d'évènements (un seul événement est exécuté à un instant donné). Il ne supporte pas l'exécution partielle d'évènements, ni la préemption.

Les événements sont décrits par des renvois et une fonction manipulatrice. Le type du **planificateur d'évènements** utilisé pour conduire la simulation peut être choisie parmi quatre actuellement disponibles : une liste liée simple (par défaut), le tas, la file d'attente de calendrier et un type spécial appelé temps réel. Chaque méthode est implémentée avec une structure de données différente :

- Le planificateur « **liste liée simple** » fournit une liste d'évènements tenue dans l'ordre chronologique, du plus premier au dernier. Pour insérer ou supprimer une entrée, il faut alors balayer toute la liste pour trouver l'entrée appropriée. L'entrée en tête est toujours exécutée en première. Les entrées avec le même temps simulé sont extraites selon leur ordre dans la liste.
- Le code du **planificateur de tas** est emprunté au simulateur MARS-2.0 (qui lui-même a emprunté le code de NETSIM DE MIT). Cette implémentation est plus efficace que le planificateur de liste liée quand le nombre d'évènements est grand. Les temps d'insertion et de suppression sont en $O(\log n)$ pour n évènements.
- Dans la mise en oeuvre du **planificateur de file d'attente de calendrier**, les événements avec le même "mois/jour" de différentes "années" sont enregistrés dans un même "jour".
- Le planificateur en **temps réel** essaie de synchroniser l'exécution d'évènements en temps réel. Il est actuellement mis en oeuvre comme une sous-classe du planificateur de liste. La capacité temps réel dans ns est toujours en développement, mais elle est employée pour introduire un réseau simulé dans une topologie réelle pour expérimenter de manière simple des

topologies réseau, la traversée de trafic... Les travaux réalisés jusqu'à présent fonctionnent pour des taux de trafic de données relativement lents. Or le simulateur doit être capable de suivre le taux d'arrivée des paquets du réseau réel. Cette synchronisation entre le réseau simulé et le réseau réel n'est actuellement pas mise en œuvre.

3.4 Point de vue de l'utilisateur

Afin de mettre en application toute la structure présentée ci-dessus, voici quelques exemples de scripts Tcl. Dans cette partie ne sera exposée que les commandes principales pour mieux comprendre le fonctionnement du simulateur. Pour une plus ample prise en main de NS-2 se référer au tutorial de Marc Greis (<http://www.isi.edu/nsnam/ns/tutorial/>).

Comme le simulateur est un interpréteur interactif de scripts, deux possibilités s'offrent à l'utilisateur : soit lancer le simulateur et taper les scripts dans l'interface du simulateur, soit écrire le script dans un fichier « .tcl ». La première solution peut être appréciée au départ pour se familiariser avec les commandes. La seconde vaut pour des simulations de plus grande ampleur et doivent ensuite être exécutées par la commande :

```
ns <fichier>
```

Comme il a été dit plus haut, avant toute action, il faut instancier un objet *Simulator* qui permettra de créer et gérer tout autre objet. Sa création est assurée par :

```
set ns_ [new Simulator]
```

Ensuite, il faut définir la topologie, c'est-à-dire les nœuds et les liens. Voici la description de deux nœuds et d'un lien les reliant :

```
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

Le lien ainsi créé est un lien « duplex » (communication en double sens), avec une bande passante de 10Mb, un délai de 10ms et une file d'attente « DropTail » (élimination de la queue). Pour créer une plus grande topologie, on peut créer les nœuds avec une boucle itérative. Chaque nœud sera alors stocké dans un tableau accessible par un indice. Le code correspondant à la création d'un tableau « n » de sept nœuds est le suivant :

```
for {set i 0} {$i < 7} {incr i} {
    set n($i) [$ns node]
}
```

Ensuite il faut définir les transferts possibles entre chaque nœud. Pour cela, il faut attacher des agents aux différents nœuds et les connecter ensemble. Voici la création d'un agent CBR :

```
set cbr0 [new Agent/CBR]
$ns_ attach-agent $n0 $cbr0
```

On peut également régler les paramètres des paquets envoyés (taille et période) :

```
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
```

De l'autre côté, il faut créer un agent récepteur sur l'autre nœud et enfin les connecter :

```
set null0 [new Agent/Null]
$ns_ attach-agent $n1 $null0
$ns_ connect $cbr0 $null0
```

Pour voir les effets des tests sur les protocoles de transport, on peut attacher un agent UDP ou TCP à un nœud générateur de trafic. Voici la mise en place d'un trafic CBR par un agent UDP :

```
set udp0 [new Agent/UDP]
$ns_ attach-agent $n0 $udp0
set cbr0 [new Agent/Traffic/CBR]
$cbr0 attach-agent $udp0
$udp0 set packetSize_ 536
```

Puis en admettant qu'un agent « Null » null0 a déjà été attaché au nœud 1, on peut connecter les agents :

```
set null0 [new Agent/Null]
$ns_ attach-agent $n1 $null0
$ns_ connect $udp0 $null0
```

Ensuite on définit les évènements du scénario, c'est-à-dire quand le trafic commence, se termine :

```
$ns_ at 0.5 "$cbr0 start"
$ns_ at 4.5 "$cbr0 stop"
```

Enfin il faut lancer la simulation par la commande :

```
$ns_ run
```

Bien entendu toutes ces commandes constituent la base des simulations. Pour simuler des ruptures de liens pour voir comment les protocoles réagissent à une panne, on peut indiquer au simulateur quand « détruire » une liaison et quand la remettre en place. On peut alors mettre un protocole de routage dynamique en place :

```
$ns_ rtmodel-at 1.0 down $(n1) $(n2)
$ns_ rtmodel-at 2.0 up $(n1) $(n2)
$ns_ rtproto DV
```

Ces quelques lignes indiquent que le lien entre les nœuds 1 et 2 ne seront pas utilisables du temps 1.0 au temps 2.0. Le protocole de routage qui se chargera de trouver une nouvelle route est celui à Vecteur de Distance.

La simulation de la mobilité dans NS-2 passe par la déclaration de nœuds mobiles. Celle-ci se passe comme pour les autres nœuds en précisant toutefois la configuration des paramètres décrits dans la section 3.2.1.6. L'attache des agents pour simuler un protocole de transport ou une application est identique. Cependant, on peut définir la position initiale des nœuds mobiles ainsi que leur mouvement de manière très précise :

```
set topo [new Topography]
$topo load_flatgrid 500 500
```

pour définir la grille, frontière de la simulation, puis :

```
$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$ns_ at 10.0 "$node_(0) setdest 20.0 18.0 13.0"
```

Les deux premières lignes indiquent la position initiale du mobile dans la grille. La dernière ligne signifie que le nœud 0 au temps 10.0 va se déplacer en direction de la position (x=20, y=18) sur la grille à une vitesse de 13 mètres par seconde.

Dans un souci de clarté des scripts, surtout pour les grandes simulations, les spécifications de mouvement ou les définitions des agents de trafic peuvent se faire dans des fichiers séparés.

Bien entendu, étant fait pour évaluer des protocoles, NS-2 permet de conserver une trace de l'échange de paquets. Les possibilités offertes sont décrites dans la section suivante. La génération de fichier de sortie fait partie du code NS-2 par des appels de procédures durant différentes actions dans les simulations. C'est pourquoi, pour visualiser une animation ou simplement enregistrer tous les évènements dans un fichier il faut le spécifier au début du script. Les commandes utilisées pour créer un fichier de sortie sont :

```
set tracefd [open sortie.tr w]
$ns_ trace-all $tracefd
```

et pour envoyer les évènements à NAM :

```
set tracenam [open out.nam w]
$ns_ namtrace-all $tracenam
```

Le détail des sorties générées par ces lignes est donné dans la section suivante.

3.5 Statistiques et visualisation

NS-2 fournit plusieurs types de support pour analyser les résultats d'une simulation. D'une part, NS-2 inclut 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. Cet outil est brièvement décrit dans la sous-section 3.5.2.

3.5.1 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 est réalisé 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 :

Action	Temps	Nœuds		Paquet	Taille	Flag	ID flux	Adresses		N°= de séq.	uid
		Source	Dest.					Source	Dest.		
+	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 :

```
s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It
message -Il 32 -If 0 -Ii 0 -Iv 32
s -t 1.511681090 -Hs 1 -Hd -1 -Ni 1 -Nx 390.00 -Ny 385.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255 -It
message -Il 32 -If 0 -Ii 1 -Iv 32
s -t 10.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It tcp -
Il 1000 -If
2 -Ii 2 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
r -t 10.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It tcp -
Il 1000 -If
2 -Ii 2 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
r -t 100.004776054 -Hs 1 -Hd 1 -Ni 1 -Nx 25.05 -Ny 20.05 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma a2 -Md 1 -Ms 0 -Mt 800 -Is 0.0 -Id 1.0 -It
tcp -Il 1020 -If 2 -Ii 21 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 1 -Po 0
s -t 100.004776054 -Hs 1 -Hd -2 -Ni 1 -Nx 25.05 -Ny 20.05 -Nz 0.00 -Ne
-1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id 0.0 -It ack -
Il 40
-If 2 -Ii 22 -Iv 32 -Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
```

Tableau 2 : format du fichier de sortie dans la mobilité .tr

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 ethernet, adresses ethernet...
- Au 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.

3.5.2 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 [45] 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++ d'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 créé 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 CONSER²⁹ et SAMAN³⁰.

NAM interprète un fichier de trace contenant des événements réseau indexés par le temps de différentes manières, comme présenté dans la **Figure 3-9**. Ces événements 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 événements interprétés.

²⁹ Projet CONSER : <http://www.isi.edu/conser/index.html>

³⁰ Projet SAMAN : <http://www.isi.edu/saman/index.html>

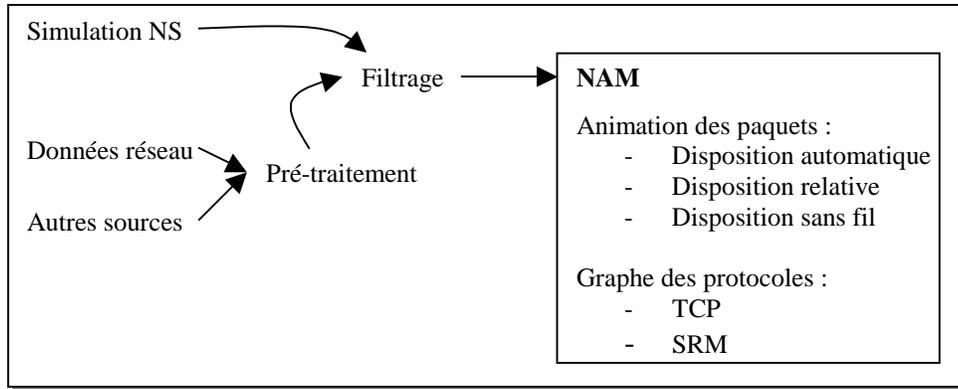


Figure 3-9 : diagramme de NAM

NAM est exécuté avec comme paramètre le fichier enregistré. Lorsqu'on exécute NAM, une fenêtre de travail NAM est créée. Il est possible de faire tourner plusieurs animations avec une seule instance, ce qui permet de mieux comparer certains protocoles. La **Figure 3-10** représente une fenêtre d'animation NAM avec le détail de la fonction de chaque bouton. On peut entre autre régler le pas de la simulation (de 8 μ s à 800ms), zoomer sur des zones de la simulation, et manipuler la lecture : on peut mettre pause à tout moment ce qui donne un « arrêt sur image », revenir, avancer sur les étapes de la simulation ce qui permet d'examiner des occurrences particulières.

La taille des objets dépend de leurs caractéristiques : l'épaisseur des liens dépend du débit de lien et la taille des paquets dépend de leur longueur en bits et de la bande passante sur le lien. La couleur des paquets peut être utilisée pour plusieurs raisons ; dans ce cas, elle différencie deux flux de données différents (celui du nœud 0 et celui du nœud 1). Les paquets se déplacent de nœuds en nœuds le long des liens et sont mis en file d'attente quand un lien est saturé (la file d'attente à côté du nœud 2 correspond à la saturation du lien entre les nœuds 2 et 3). Les points bleus et rouges que l'on voit dans la partie inférieure de la fenêtre représentent les paquets détruits. Ces paquets sont détruits car la file d'attente est remplie et le lien entre 2 et 3 n'a pas assez de débit pour écouler tous les paquets émis par les nœuds 0 et 1.

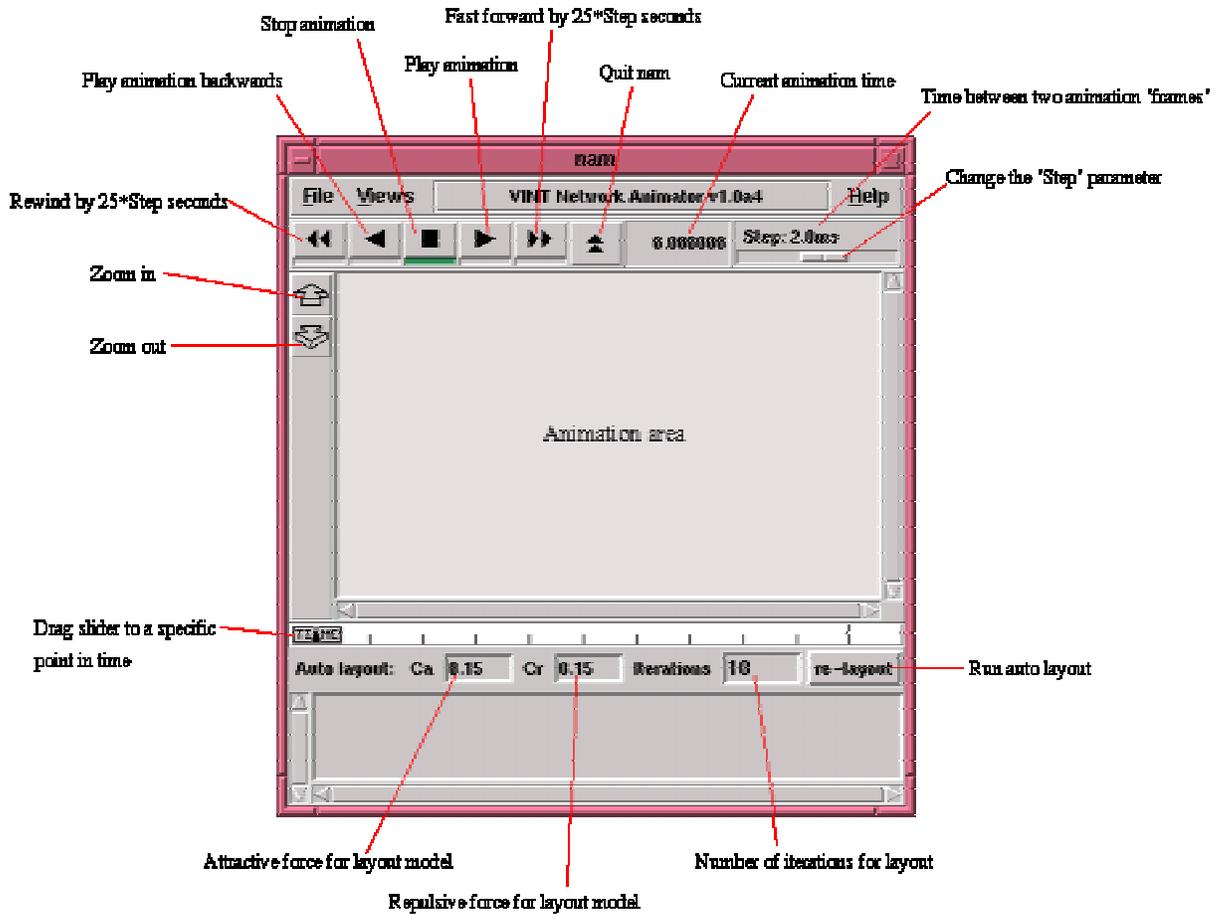


Figure 3-10 : fenêtre d'animation NAM

L'animation de paquets est représentée **Figure 3-11**.

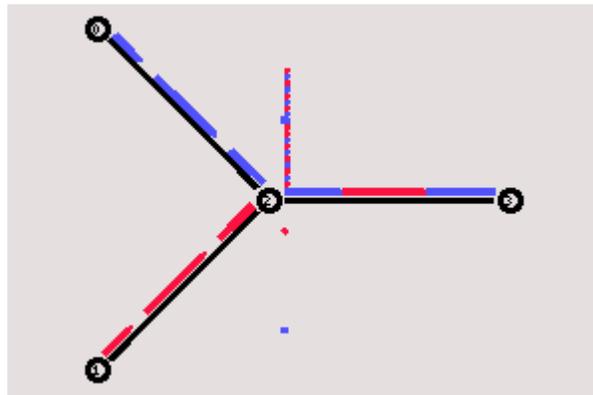


Figure 3-11 : animation de paquets dans NAM

► Visualisation de la mobilité

De la même manière que s'est introduite la mobilité dans NS-2, la visualisation de la mobilité dans NAM s'est faite par l'ajout d'extensions. Les nœuds mobiles apparaissent comme les autres nœuds à la différence près qu'ils ne sont pas reliés entre eux. Dans un premier temps l'affichage des paquets circulant entre eux n'était pas supporté, seuls l'affichage de leurs mouvements l'était. A l'heure actuelle, les paquets échangés entre nœuds mobiles sont représentés par des petits points lorsque les mobiles sont à portée l'un de l'autre. Les points d'accès représentant la frontière entre le réseau filaire et le réseau sans fil sont rattachés au réseau filaire par un lien et communiquent avec les nœuds mobiles. L'émission des beacons sur l'interface radio est simulée par l'apparition périodique de cercle autour du point d'accès, indiquant dans le même temps leur portée.

Cependant, l'affichage n'est pas encore parfait et est en cours d'amélioration. Effectivement, dans chaque script il faut préciser les positions initiales des nœuds mobiles dans la fenêtre NAM, alors que cette information pourrait être déduite de la déclaration des positions initiales des nœuds dans la simulation. De plus, le dernier nœud mobile affiché est systématiquement au même endroit dans la fenêtre, quelque soit la position qu'on indique. Par contre, s'il vient à se déplacer, il part du bon endroit. Un autre défaut relativement gênant concerne toujours la disposition des nœuds dans la fenêtre NAM. Maintenant qu'il est possible de faire des simulations avec à la fois des nœuds mobiles et fixes, l'affichage ne peut pas être contrôlé totalement : on ne peut actuellement pas donner explicitement la position des nœuds filaires dans la fenêtre d'animation. Même les points d'accès, pour lesquels la position joue un rôle important puisqu'elle définit une zone d'accessibilité, ne peuvent pas être affichés au bon endroit.

Afin de mieux situer ce problème, prenons en considération l'exemple suivant : soit un nœud mobile qui se déplace entre deux points d'accès situés sur la même ligne horizontale mais suffisamment espacée pour que leurs cellules se recouvrent à peine (voir **Figure 3-12**). Dans ce scénario, on désire tester comment s'opère le passage du nœud mobile entre ces deux points d'accès lorsqu'il va et vient de l'un à l'autre. La visualisation apportée par NAM est complètement inefficace comme le montre la **Figure 3-13**.

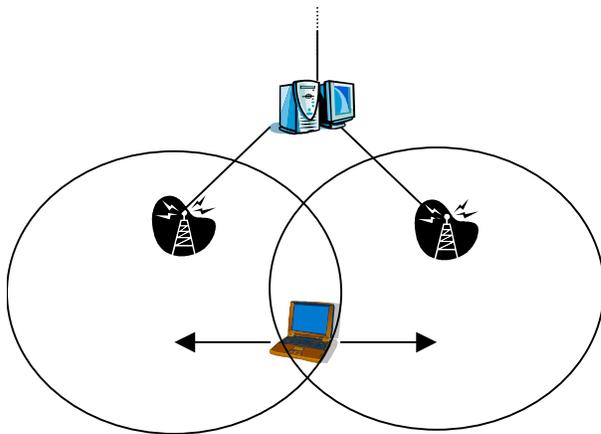


Figure 3-12 : simulation souhaitée

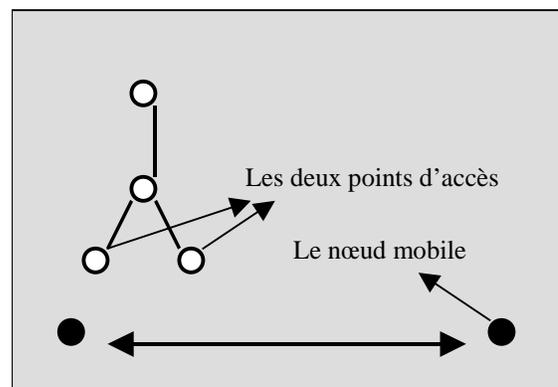


Figure 3-13 : Simulation observée

3.6 Extension 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 Manheim, 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. Malheureusement, le temps m'a manqué pour faire des tests concluants sur la mobilité dans NS-2 à travers ces modèles dans le cadre du stage de DEA. Ces tests seront effectués durant les prochains mois.

3.6.1 NOAH

Cette première extension présentée joue un rôle important dans la gestion de la mobilité dans NS-2. Effectivement, on verra que l'extension CIMS présentée ci-dessous utilise l'agent 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). L'extension est disponible à l'url <http://www.icsi.berkeley.edu/%7Ewidmer/mnav/ns-extension/>.

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... étudiés dans la section 3.2.1.6). 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 (voir section 3.2.1.6) 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 inclus 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).

3.6.2 CIMS

CIMS³³ est une extension de NS-2 basée sur les versions 6 (ns-2.1b6) ou 7 (ns-2.1b7) disponibles à l'url <http://www.comet.columbia.edu/micromobility>. 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

³¹ AT&T : <http://www.att.com>

³² ACIRI : <http://www.aciri.org>

³³ CIMS : Columbia IP Micro-Mobility Suite

collaboration avec, laboratoires et entreprises.

CIMS v1.0 inclus les implémentations NS-2 de Cellular IP, Hawaii et Mobile IP Hiérarchique qui ont été présentés dans la deuxième partie de ce rapport. 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 de propagation de flux multiples. Pour l'instant, la 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. Le tableau 3.3 présente les supports de chacune des implémentations de manière détaillée.

	Cellular IP	Hawaii	MIP Hiérarchique
Couche OSI	Couche 3	Couche 3	Couche 3.5
Nœuds impliqués	Nœuds IP cellulaires	Tous les routeurs	Agents visités
Id. du nœud mobile	Adresse principale	Adresse temporaire	Adresse principale
Nœuds intermédiaires	Commutateur niveau 2	Commutateur niveau 2	Routeurs niveau 3
Mise à jour	Paquets de données	Messages de signalisation	Message de signalisation
Pagination	Non	Non	Explicite
Tunneling	non	Non	oui
Handoff déclenché par la couche 2	optionnel	optionnel	Non
Messages MIP	non	oui	oui

Tableau 3 : fonctionnalité de l'implémentation CIMS

L'université de Colombie fourni un scénario type pour effectuer des tests. L'exemple décrit ci-dessous concerne une topologie constitué de nœuds Hawaii. Ils proposent d'utiliser cette même topologie en changeant simplement le type des points d'accès en nœuds cellulaire IP puis en point d'accès hiérarchique pour implémenter MIP hiérarchique. Cependant l'extension a encore des bugs et son installation requiert beaucoup de retouches, c'est pourquoi les tests comparatifs n'ont pas été fait. On verra tout de même dans la partie qui suit, le scénario qu'ils proposent. Ensuite, on détaillera des tests que l'équipe COMET a réalisés sur le protocole Cellular IP.

3.6.2.1 Scénario

Le script du scénario suivant est disponible sur le site Internet de CIMS. À la fin de la simulation, une fenêtre de NAM apparaît (**Figure 3-14**) qui donne la topologie du réseau.

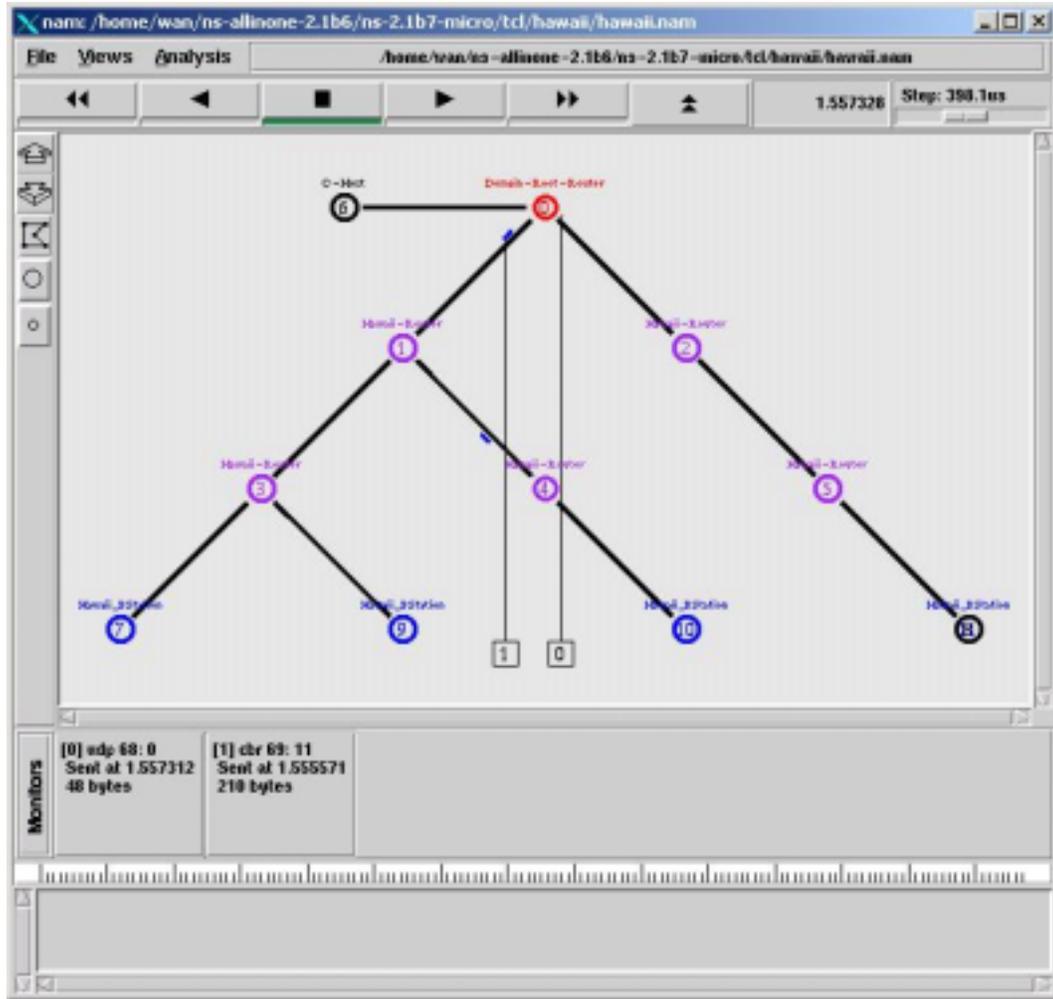


Figure 3-14 : topologie des tests

Cette simulation crée un réseau d'accès sans fil où un unique nœud mobile se déplace entre les point d'accès Hawaii 10 et 11 en bas à droite dans la **Figure 3-14**.

Dans ce scénario, le premier handoff arrive à peu près à 1.53 secondes dans la simulation. En ralentissant le minuteur d'affichage on peut observer le flux de message de contrôle (paquets rouges représentant la messagerie de contrôle, indiquée par "0" dans la **Figure 3-14**) et son impact sur la propagation des paquets de données.

3.6.2.2 Tests sur Cellular IP

Dans cette section, des tests sur les performances du handoff dans des réseaux d'accès Cellular IP sont étudiés. Ces tests portent sur les impacts quantitatifs du modèle du handoff dans les réseaux cellular IP (perte de paquets, retard dans la livraison des paquets...).

► Environnement de Simulation

L'environnement de simulation IP Cellulaire employé pour les résultats annoncés est montré dans la **Figure 3-15**. il s'agit d'un nœud mobile qui se déplace d'un point d'accès (à gauche dans la figure) à une autre (à droite dans la figure). Ces points d'accès ont des zones de couverture qui se recouvrent. Le routeur passerelle de Cellular IP peut atteindre les deux points d'accès et est connecté à un LAN. Un autre nœud fixe est aussi connecté au LAN qui va interagir avec le nœud mobile.

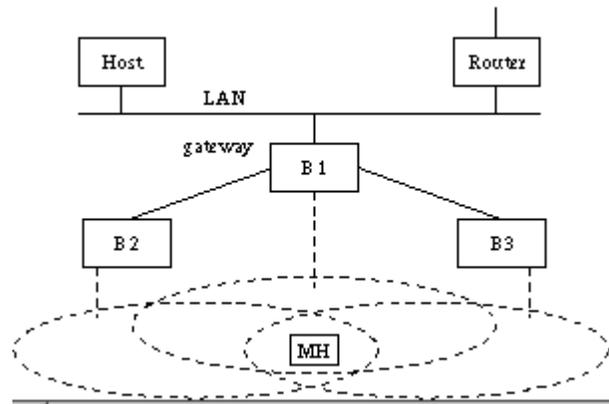


Figure 3-15 : plate-forme de test Cellular IP

Les hypothèses de l'environnement de simulation NS-2 sont les suivantes : tout d'abord, on considère que les nœuds mobiles utilisent "une interface sans fil idéale", c'est-à-dire que les paquets sont transmis sur l'interface sans fil sans retard, sans erreur et sans perte. La congestion sur l'interface radio n'est pas modélisée non plus, ni les beacons transmis par le routeur passerelle cellulaire IP. Le réseau est configuré quand la session de simulation est initialisée et la topologie reste constante pendant toute la durée de la simulation. Finalement, la disposition des points d'accès cellulaire IP est telle que les cellules de recouvrement se chevauchent pour que le mouvement des nœuds mobiles d'une cellule à une autre soit immédiat. Cela ne limite pas la capacité du simulateur à étudier la perte de paquet pendant le handoff puisque la perte est principalement due à des erreurs de routage.

► Performance du handoff

Latence du handoff

L'impact du handoff sur des sessions en cours est généralement caractérisé par la latence du handoff (voir définition partie 2 section 2.1). Bien que la latence du handoff ne détermine pas entièrement les performances du point de vue des applications, c'est une bonne indication des performances du handoff. On peut décomposer ce temps de latence en deux : le temps de rendez-vous et le temps pour le protocole. Le temps de rendez-vous est le temps pris par le nœud mobile pour s'attacher au nouveau point d'accès après avoir quitté l'ancien. Ce temps est lié à la couche physique et au protocole d'accès au médium. Le temps pris pour le protocole est le temps pris pour restaurer le trafic des sessions du

nœuds mobiles au nouveau point d'accès une fois qu'il a reçu un beacon du nouveau point d'accès. Les tests porteront essentiellement sur le temps pris pour le protocole, le temps de rendez-vous étant généralement très faible.

On considèrera donc dans le cadre de ces tests que la latence du handoff est la différence entre le moment où le nœud mobile quitte l'ancien point d'accès et le moment où il reçoit le premier paquet au nouveau (en considérant le temps de déplacement nul).

Perte de Paquet

En plus de la latence du handoff, la qualité de service au niveau applicatif est aussi affectée par la perte de paquet pendant le handoff. Pour déterminer la perte de paquet pendant un handoff, un nœud quelconque de l'Internet transmet un flot de paquets périodiquement au nœud mobile. Avant qu'un handoff ne soit amorcé, les paquets sont acheminés le long de l'ancienne route. Dans la simulation, on suppose que le nœud de croisement (B1 dans la **Figure 3-15**) connaît d'avance lequel des paquets du flot sera le dernier pour atteindre le nœud mobile à son ancienne localisation. On suppose que le nœud de croisement marque ce paquet. En recevant le paquet marqué, le nœud mobile exécute un handoff et transmet immédiatement un paquet de mise à jour à travers le nouveau point d'accès. Les paquets acheminés par le nœud de croisement après le paquet marqué, mais avant l'arrivée du paquet de mise à jour, sont envoyés à l'ancien point d'accès et sont donc perdus. Cet intervalle de temps est égal à la somme du temps pris par le paquet marqué pour atteindre le nœud mobile et le temps pris pour le paquet de mise à jour pour atteindre le nœud de croisement. La perte de paquets dû au handoff est donc lié au temps d'aller-retour entre les anciennes et nouvelles localisation et le nœud de croisement.

La perte de paquet occasionnée lorsqu'une source émet à un débit constant de bits (CBR) dans cet environnement de simulation est montré dans la **Figure 3-16**.

La courbe représente en ordonnée le nombre moyen de paquets perdus pendant le handoff et en abscisse le temps entre les arrivées de paquet en secondes. Ces résultats sont réalisés avec des nœuds mobiles et des points d'accès implémentant le handoff de base décrit dans Cellular IP (ce qu'on a appelé hard handoff dans la partie 2 section 2.5.1.2).

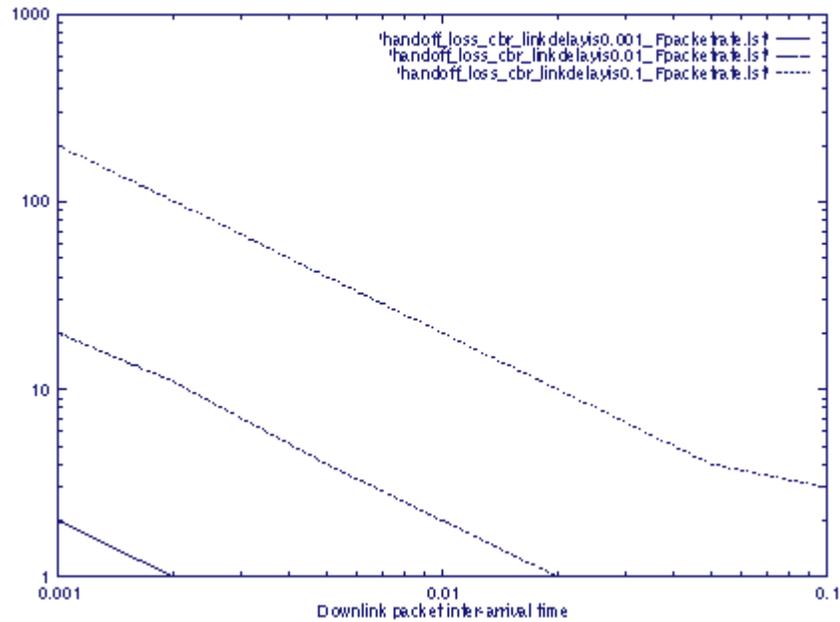


Figure 3-16 : paquets perdus pour un flux CBR

Les performances de handoff sont dépendantes des conditions de trafic. Dans un réseau fortement chargé, la latence du handoff et la perte de paquet seront plus importants. Les applications en temps réel (voix sur IP par exemple) sont sensibles au retard dans la livraison des paquets et ne peuvent typiquement pas tolérer le retard associé à la retransmission des paquets perdus. Pour ces applications, le nombre de paquets perdus caractérise les performances du handoff. D'autres applications, cependant, utilisent le contrôle de flux de bout en bout pour répondre aux conditions réseau et retransmettent des paquets et/ou réduisent le taux de transmission si des erreurs ont lieu. Dans ce qui suit, on analyse l'impact d'un handoff sur un trafic TCP. TCP représente aujourd'hui le type de trafic le plus typique dans l'Internet puisqu'il supporte le World Wide Web, le transfert de fichier, l'établissement de connexion à distance...

Comportement TCP

Tout d'abord, on va observer le comportement d'une session TCP pendant un handoff dans la simulation précédente. Le premier exemple TCP est le téléchargement de données par un nœud mobile. La taille des paquets TCP est de 1000 octets et l'utilisateur mobile a jusqu'à 5 Mbps de la largeur de bande en sortie, c'est-à-dire que le taux de paquet partant est de 625 paquets/seconde. Le temps de transmission de paquets entre des nœuds dans la configuration simulée est 2 ms, ce qui abouti à une latence du handoff de 4 ms. **Figure 3-17** expose les numéros de séquence des paquets de données arrivés et des acquittements observés au routeur passerelle pendant le handoff. Elle présente en ordonnée les numéros de séquence TCP côté client (paquets de données et acquittements) et le temps en secondes en abscisse.

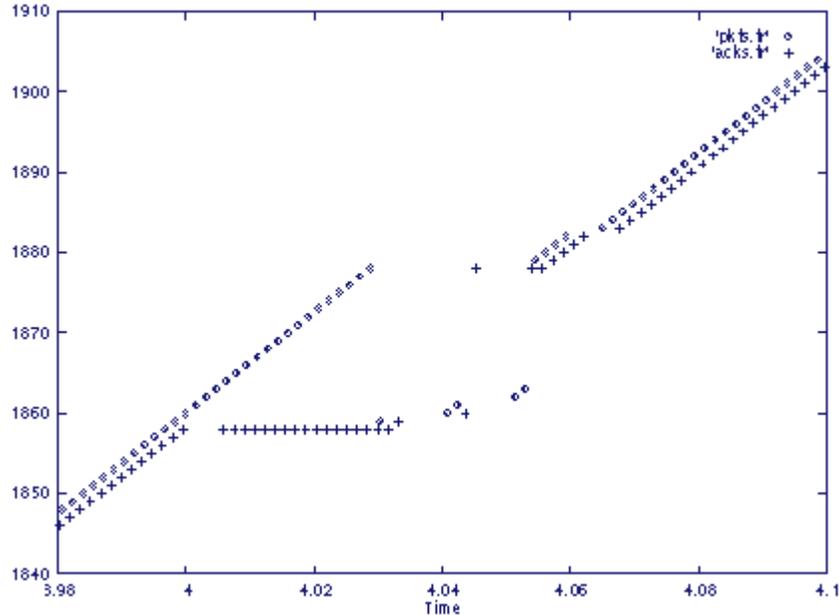


Figure 3-17 : numéro de séquence TCP pendant un handoff (mobile récepteur)

D'après la **Figure 3-17**, le handoff est initialisé par le nœud mobile à 4 secondes dans la simulation. Trois paquets consécutifs sont perdus comme indiqué par les trois acquittements manquants consécutifs. Une fois que le processus de handoff est achevé, les paquets continuent à parvenir au nœud mobile. Ces paquets sont, cependant, reçus dans le mauvais ordre et impliquent la duplication des acquittements par le récepteur. Cette duplication est indiquée par la ligne horizontale des numéros de séquence des acquittements. Les acquittements dupliqués informent l'émetteur TCP des pertes et causent la retransmission des paquets perdus. Le premier paquet retransmis arrive environ 20 ms après le handoff. Le taux TCP maximum est à nouveau atteint à 4.07 secondes.

Le handoff est interprété par l'émetteur dans le réseau IP filaire comme une congestion qui cause la réduction du taux de transmission. Le contrôle du flux TCP dans NS-2 permet l'augmentation ou la réduction progressive de la fenêtre d'émission selon les erreurs produites. Si jamais une perte ou une congestion est observée, l'émetteur réduit immédiatement la fenêtre. La simulation montre que la fenêtre de transmission maximum est remise en place environ 70 ms après que le handoff ait été initialisé.

Dans l'expérience suivante, le nœud mobile est la source du trafic TCP, tout le reste de la simulation est exactement identique au cas précédent. Dans ce cas, le handoff provoque la perte des acquittements et non la perte des paquets de donnée. Les résultats de la simulation sont montrés dans la **Figure 3-18**.

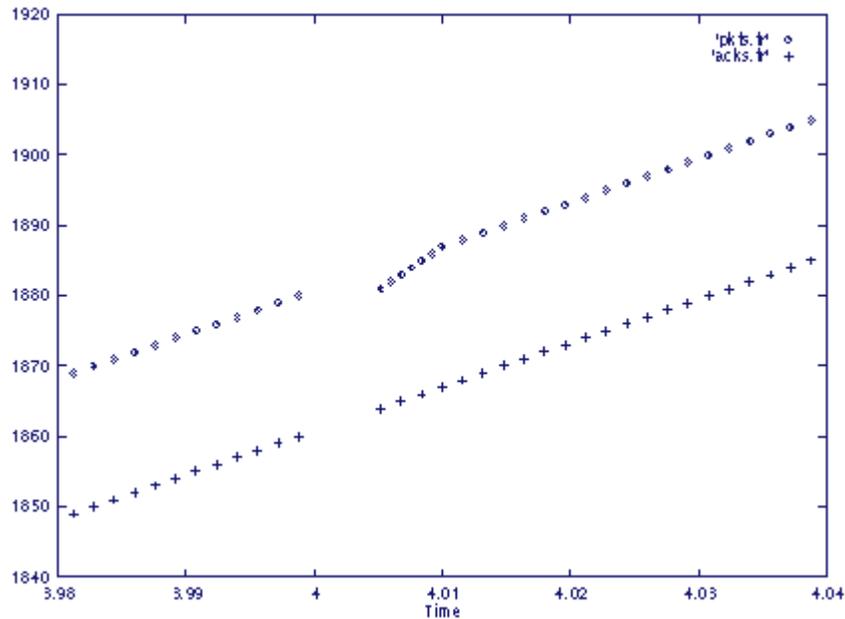


Figure 3-18 : numéro de séquence TCP pendant un handoff (mobile émetteur)

Avant que le handoff ne soit initialisé, l'expéditeur TCP au niveau du nœud mobile utilise une fenêtre d'émission maximale de 20 paquets, ce qui se remarque par la différence des numéros de séquence entre les paquets de données et les acquittements. À 4 seconde (temps simulé) le nœud mobile exécute un handoff et arrête de recevoir les acquittements pendant environ 4 ms, ce qui représente la latence du handoff. Pendant le handoff, l'expéditeur n'envoie pas de paquets puisque sa taille de fenêtre est entièrement utilisée et il a besoin d'acquittements pour continuer l'émission. Dans cette expérience, le handoff est initialisé quand la session TCP est stabilisée et les acquittements parviennent au nœud mobile continûment. D'après les caractéristiques TCP, le premier acquittement que le nœud mobile reçoit acquitte non seulement le paquet de données dont le numéro de séquence correspond, mais aussi tous les paquets de données de numéro de séquence inférieur qui n'avaient pas été acquittés. L'émission reprend donc très vite, et les répercussions sur la session TCP sont très faibles dans le cas présent.

Nous observons donc un impact différent du handoff selon que celui-ci arrive dans une phase de début (petite fenêtre d'émission, acquittement reçu irrégulièrement) ou dans une phase avancée stable et selon que le nœud mobile est la source ou le récepteur du flux TCP.

3.6.3 Mobiwan

Le projet de recherche Mobiwan³⁴ a pour but d'intégrer la mobilité des nœuds dans des grandes aires de réseaux IPv6 dans les simulations NS-2. L'objectif principal est de simuler une mobilité locale (dans un domaine administratif ou sur un site) et une mobilité

³⁴ Mobiwan : Mobility in Wide-Area Network : mobilité dans des grandes aires de réseaux

globale (à travers les frontières de domaines ou de sites) dans des grandes topologies Internet. En particulier, ce projet a l'ambition de simuler Mobile IPv6, Mobile IPv6 hiérarchique, et divers protocoles de micro-mobilité pour des topologies allant jusqu'à des centaines de nœuds. Pour cela, il est nécessaire de pouvoir simuler une grande topologie Internet représentative de la réalité et un modèle de mobilité qui permettra la mobilité d'un nœud dans n'importe quelle partie de cette topologie. Cette mobilité est montrée dans la **Figure 3-19** où les flèches bleues montrent une mobilité dans un site et la flèche rouge une mobilité globale entre sites ou domaines.

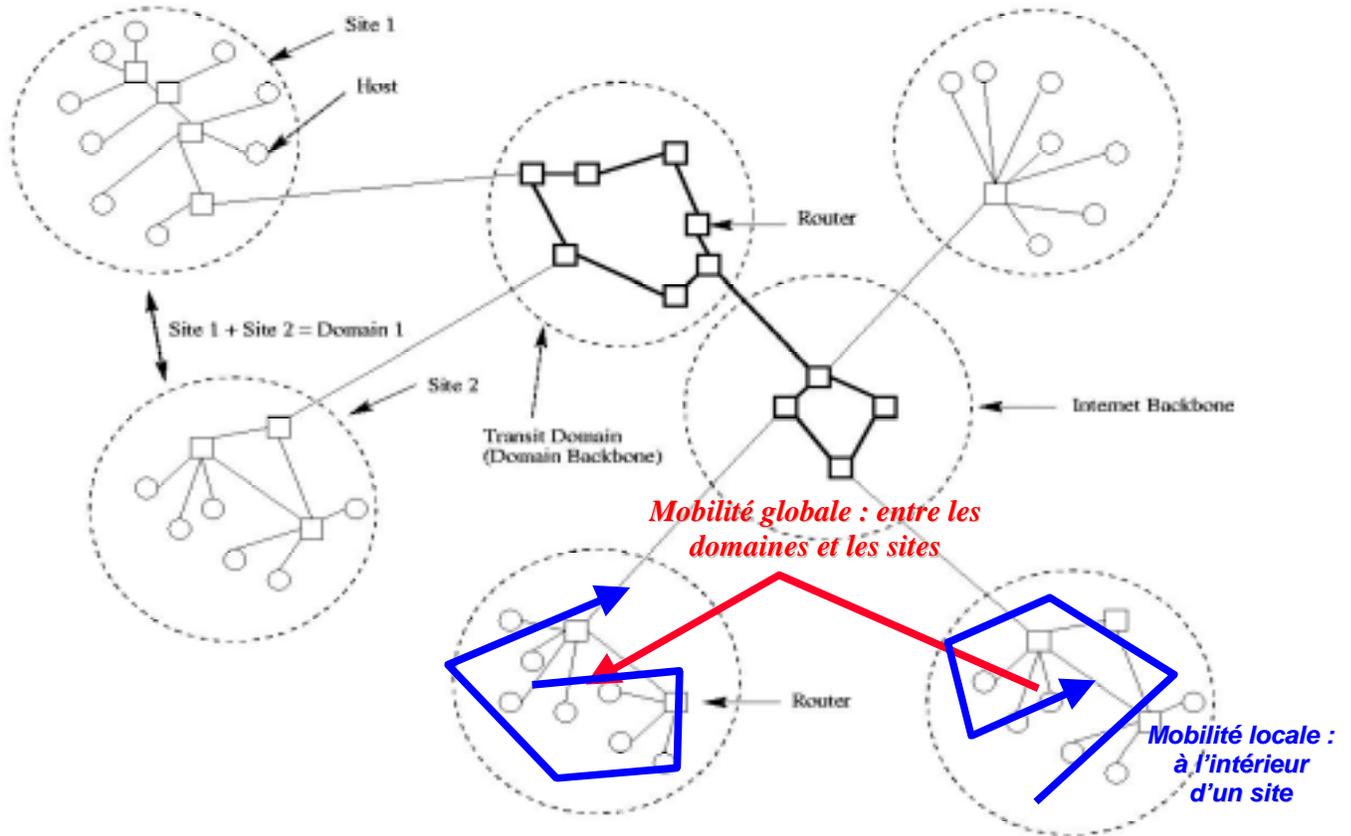


Figure 3-19 : aires de mobilité locale et de mobilité globale

L'extension Mobiwan a été implémentée pour la version 6 de NS-2 (NS-2.1b6) par Thierry Ernst de l'INRIA Rhône Alpes³⁵ dans le cadre du projet PLANETE³⁶, en collaboration avec le laboratoire Motorola³⁷ de Paris. Il est prévu d'étendre le code pour qu'il tourne sous la version 8 de NS-2 (NS-2.1b8).

Le projet se découpe en deux parties : une première partie concerne la génération et la

³⁵ INRIA : <http://www.inrialpes.fr>

³⁶ projet PLANETE : <http://www.inrialpes.fr/planete>

³⁷ Motorola Paris : <http://www.motorola.fr>

manipulation de grandes topologies. Ces topologies doivent pouvoir être créées et contrôlées le plus simplement possible. Le code fourni pour simuler des grandes topologies peut servir pour simuler d'autres protocoles qui n'impliquent pas forcément des nœuds mobiles. Une deuxième partie consiste en l'implémentation du protocole MIPv6 dans NS-2. Comme on l'a vu dans la présentation de Mobile IPv6 dans la partie 1 section 1.4, MIPv6 nécessite que les nœuds non mobiles implémentent certaines fonctionnalités spécifiques. Il a donc fallu rajouter cette implémentation à l'ensemble des nœuds dans NS-2. Cependant, l'ensemble du protocole IPv6 n'a pas été mis en œuvre. L'implémentation de MIPv6 hiérarchique est prévu pour dans quelques mois. Puis, le but final du projet est de pouvoir utiliser simultanément l'apport de ces deux parties pour simuler la mobilité v6 dans des grandes aires de réseaux.

3.6.3.1 Génération et manipulation de grandes topologies

Comme il a été dit plus haut, la génération de grande topologie peut être utilisée pour toutes sortes de topologie. Des tests ont été réalisés avec plus de 3000 nœuds, ce qui devient intéressant pour l'étude de la mise à l'échelle de protocoles. Pour conduire des grandes simulations, il faut :

- Une génération automatique d'une grande topologie hiérarchique qui est une représentation réelle de ce qu'on peut trouver dans l'Internet.
- Une topologie hiérarchique qui expose la notion de domaines, de sites et de sous-réseaux.
- Une configuration et une manipulation simple de la topologie pour identifier la position et les fonctions d'un nœud dans la hiérarchie, pour configurer un nœud avec les caractéristiques appropriées à ses fonctions. Par exemple, on peut configurer tous les routeurs frontières avec MIPV6 hiérarchique et tous les points d'accès avec une fonction d'agent mère.
- Une configuration de scénario de simulation facile, par exemple choisir tous les routeurs d'un site particulier comme correspondants d'un nœud mobile.

La configuration et la manipulation de grandes topologies réseau sont automatique dans NS-2 de base. Les topologies générées sont de bonnes représentations de ce qu'on trouve dans l'Internet et permettent de définir des fonctionnalités différentes aux nœuds selon leur position dans la topologie.

Le module de NS-2 qui crée la topologie est GT-ITM [47]. GT-ITM est un générateur de topologie hiérarchique au format SGB. Il crée des topologies constitués de deux sortes de domaines : d'une part les domaines terminaux (*stub domain*) qui supporte uniquement le trafic généré ou à destination d'un nœud dans ce domaine. D'autre part, la topologie est constitué de domaine de transit (*transit domain*) qui supporte en plus la propagation de trafic. Ce sont des domaines qui relient d'autres domaines. Un domaine terminal correspond à un campus ou à un réseau local (LAN) alors qu'un domaine de transit correspond plutôt à une grande aire de réseau (WAN³⁸). La **Figure 3-20** illustre ces notions de domaine et explicite légèrement la notion de fonction de nœuds qui sera

³⁸ WAN : Wide Area Network : grande aire de réseau

développé plus tard.

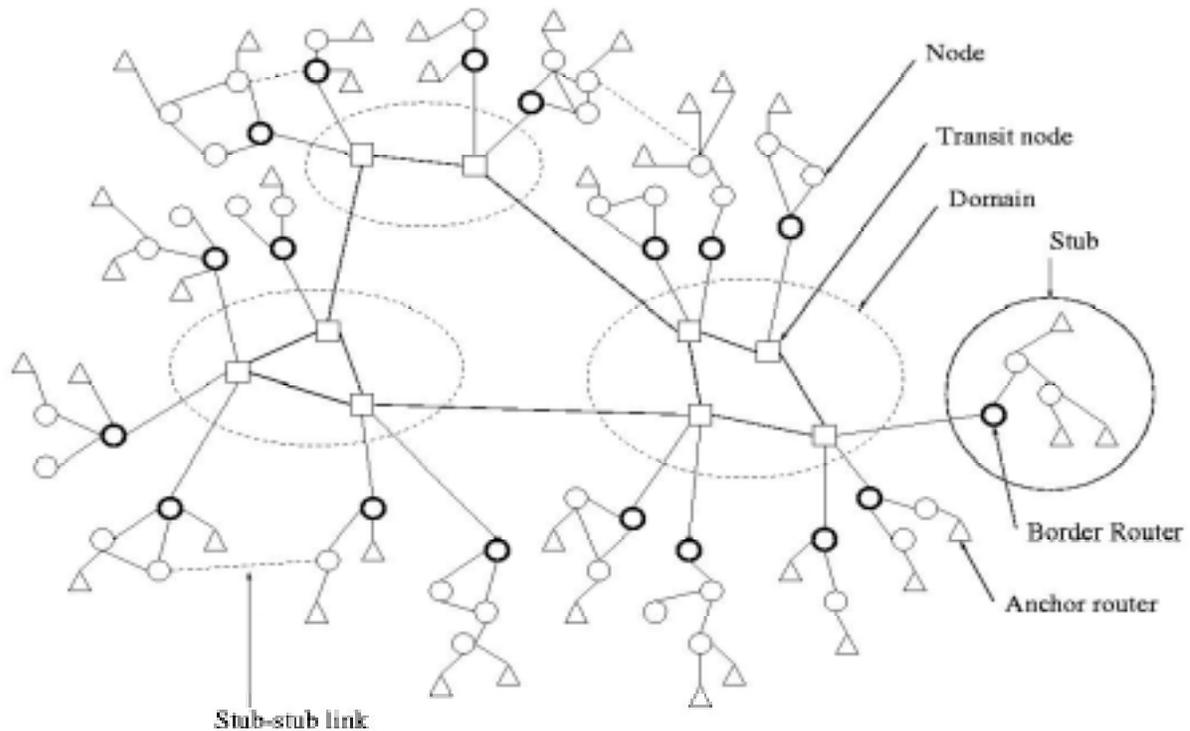


Figure 3-20 : topologie attendue

Cependant, le traducteur du format SGB de la topologie créée par GT-ITM au format NS-2 perd la plupart des informations, en particulier la hiérarchie des nœuds. Bien que le traducteur produise des adresses hiérarchiques, l'utilisateur n'est pas capable de déterminer si un nœud est un nœud de transit ou un nœud terminal.

La topologie générée par GT-ITM est tout de-même un bon squelette pour les besoins de cette extension. Il manque la notion de site, qu'il faudra ajouter. On peut définir un site dans une topologie GT-ITM comme étant un domaine terminal avec des points d'accès en plus.

Les outils ajoutés par l'extension pour réaliser ces tâches sont le traducteur TOPOGEN et la librairie TOPOMAN qui sont décrits ci-dessous. La forte hiérarchie mise en place implique aussi un changement dans l'adressage, que nous allons étudier par la suite.

► TOPOGEN

TOPOGEN traduit la production GT-ITM (format SGB) en un format NS-2 approprié pour TOPOMAN. La topologie GT-ITM n'est pas modifiée. TOPOGEN classe les nœuds selon leur position dans la topologie (quel domaine de transit dans la topologie, quel site dans le domaine de transit et quel sous-réseau dans le site) et leur fonction. La sortie de TOPOGEN donne le nombre de domaines administratifs dans la topologie, ajoute des

nœuds de transit au domaine indiqué et pour chaque nœud de transit, il détermine le nombre de sites. Pour chaque site, il ajoute un nœud frontière et d'autres nœuds. Bien sûr, il produit aussi la liste des liaisons entre les nœuds

➤ **TOPOMAN**

TOPOMAN est une bibliothèque de procédures Tcl/OTcl pour manipuler et configurer une topologie produite. TOPOMAN construit automatiquement la topologie NS. Il charge la topologie TOPOGEN décrite ci-dessus, définit l'adressage NS-2 automatiquement et crée des nœuds et des liaisons. TOPOMAN permet non seulement de créer et de configurer la topologie automatiquement, mais il permet aussi d'afficher et de jouer avec la topologie avant qu'elle ne soit effectivement créée. Ceci est très utile : pour ajuster la topologie, l'utilisateur n'a pas besoin de lancer NS-2, ce qui impliquerait la création de tous les objets et de la table de routage.

Le chargement de la topologie NS-2 peut être exécutée à la main avec des instructions TOPOMAN, ou par l'utilisation du traducteur TOPOGEN. TOPOMAN permet de classifier les nœuds selon leur fonction dans la topologie. À ce jour, les fonctions sont :

- Nœud de Transit : nœuds qui peuvent générer et recevoir des paquets, mais aussi uniquement en propager.
- Routeur frontière : nœuds connectant un site à un domaine de transit.
- Routeur de Site : routeur autre que le routeur de frontière dans un site.
- Point d'accès : routeur connectant les nœuds mobiles à un nœud filaire.
- Nœud mobile : fonction identique à celle définie dans la section 3.2.1.6

L'utilisateur peut ajouter des nouveaux types de nœuds, toujours en utilisant les appels à TOPOMAN. TOPOMAN tient une information détaillée sur la topologie (nombre de domaines, de sites, de nœuds dans un site, etc.). TOPOMAN peut être étendu pour retourner n'importe quelle sorte d'information nécessaire à l'utilisateur. Pour le moment, TOPOMAN peut rendre la liste des nœuds frontière dans la topologie, le préfixe d'adresse pour des nœuds dans le topologie, etc.

➤ **Outil pour les scénario : SCEN TOOL**

SCEN TOOL est un outil de configuration et de génération de scénario de mobilité qui utilise TOPOMAN.

➤ **Adressage**

Chaque nœud de transit correspond à un sous-domaine. Un domaine administratif est donc une accumulation d'un ou plusieurs nœuds de transit. Chaque nœud de frontière est la porte entre un site et un sous-domaine. Tous les routeurs dans le même site (routeurs frontière, routeurs de site, points d'accès) ont le même préfixe, c'est-à-dire qu'ils ont tous le préfixe du site. Chaque routeur dans le site a aussi son propre préfixe de sous-réseau.

TOPOMAN organise les adresses NS-2 selon cette information, sans qu'on ait à se soucier de quoi que ce soit. Des procédures TOPOMAN permettent de connaître à quel site un nœud appartient.

L'adresse de NS-2 de base est divisée en 3 niveaux par défaut. Il était techniquement impossible de faire apparaître le domaine et le sous-domaine dans une adresse de NS-2 de base, c'est pourquoi cette information est tenue par TOPOMAN de manière interne.

L'utilisation des nœuds mobiles dans NS-2 de base passe par le système d'adressage suivant:

- Le premier niveau identifie le site ; un nœud de transit a son propre site (raison technique). Tous les nœuds avec le même site ont le même identifiant de site.
- Le deuxième niveau identifie le sous-réseau dans le site ; on obtient ainsi le préfixe sous-réseau. Tous les routeurs frontières, les routeurs de site et les points d'accès ont le même identifiant de sous-réseau (unique).
- Le troisième niveau identifie l'hôte dans sous-réseau, c'est-à-dire le système terminal. Les routeurs ont *hostid* = 0, les nœuds mobiles ont *hostid* ≠ 0.

L'utilisateur peut désirer conserver ce niveau hiérarchique d'adressage, notamment lorsqu'il effectue des simulations sans nœuds mobiles. Cependant, l'utilisation d'un niveau supplémentaire d'adressage peut être préféré pour réduire la consommation de mémoire de NS-2 et réduire le temps de calcul des tables de routage (par une information plus explicite). Par exemple, une simulation avec une topologie de 500 nœuds et les mêmes paramètres prendra 37 minutes au lieu de 70 et consommera 100 MO de mémoire au lieu de 150.

Les 4 niveaux d'adressage dans NS-2 signifient :

- Le premier niveau identifie le sous-domaine dans la topologie (c'est-à-dire les nœuds de transit). Cela donne le préfixe de sous-domaine (le préfixe de domaine est enregistré intérieurement)
- Le deuxième niveau identifie le site dans le sous-domaine (i.e. *stub*). Les nœuds de transit sont dans le *stubid* 0. Cela donne le préfixe de site.
- Le troisième niveau identifie le sous-réseau dans le site. Cela donne le préfixe du sous-réseau. Tous les routeurs frontières ont *stubid* ≠ 0.
- Le quatrième niveau identifie l'hôte dans le sous réseau, c'est-à-dire le système terminal. Les routeurs ont *hostid* = 0 et les nœuds mobiles ont *hostid* ≠ 0.

Pour effectuer ce niveau hiérarchique d'adressage, il faut aussi compiler NS-2 avec les fichiers *route.cc* et *route.h* modifiés. Il est impossible d'utiliser à la fois l'adressage à 3 niveaux et à 4 niveaux dans le même fichier binaire (cela aurait exigé de changer plus de code dans Tcl, alors que là il ne s'agit que d'une modification de fichiers...). Pour les simulations sans nœuds mobiles, l'adressage à 4 niveaux de hiérarchie n'est pas nécessaire.

3.6.3.2 Mobile IPv6 et IPv6

L'implémentation des protocoles Mobile IPv6 et IPv6 est mise en oeuvre par un jeu de nouveaux agents et de nouveaux classificateurs NS-2. Ce modèle sera prochainement étendu pour soutenir Mobile IPv6 hiérarchique. Toutes les particularités d'IPv6 n'ont pas été mises en oeuvre, seules celles nécessaires à Mobile IPv6 ont été implémentées (par

exemple DHCPv6, découverte des voisins, etc.).

Les nœuds qui ont besoin d'implémenter Mobile IPv6 sont les nœuds mobiles, les points d'accès et les correspondants des nœuds mobiles. Les correspondants doivent implémenter MIPv6 pour réaliser entre autre l'encapsulation, la décapsulation, le traitement des en-têtes de routage. Ces changements ont été apportés dans la configuration des nœuds NS-2 (classe *Node*).

Quant au nœud mobile et au point d'accès (ou à l'agent mère), l'extension s'appuie sur la classe *MobileNode* créée par CMU (voir section 3.2.1.6). L'agent de routage ad hoc est remplacé par l'agent Réseau (*Network Agent*). La structure des nœuds Mobile IPv6 est montrée dans la **Figure 3-21** et la **Figure 3-22**.

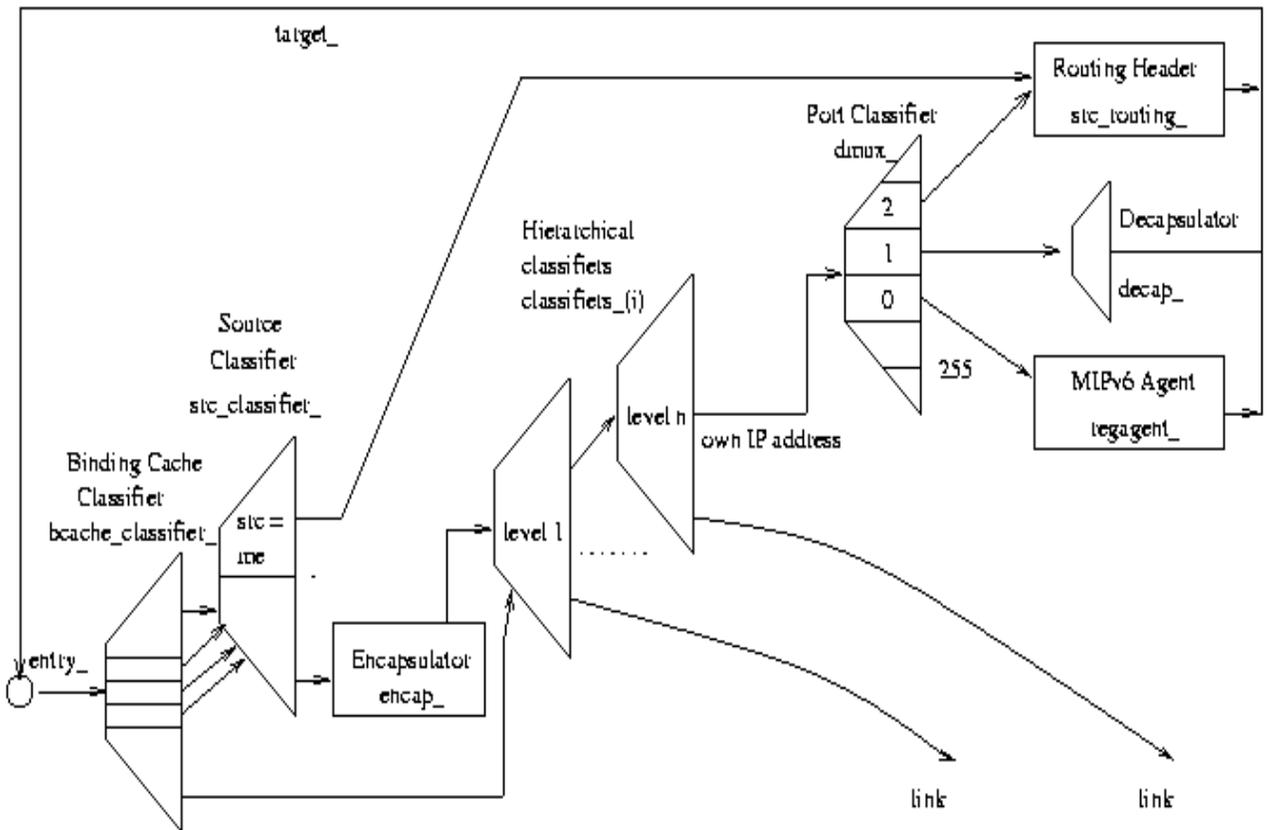


Figure 3-21 : nouvelle structure d'un nœud filaire

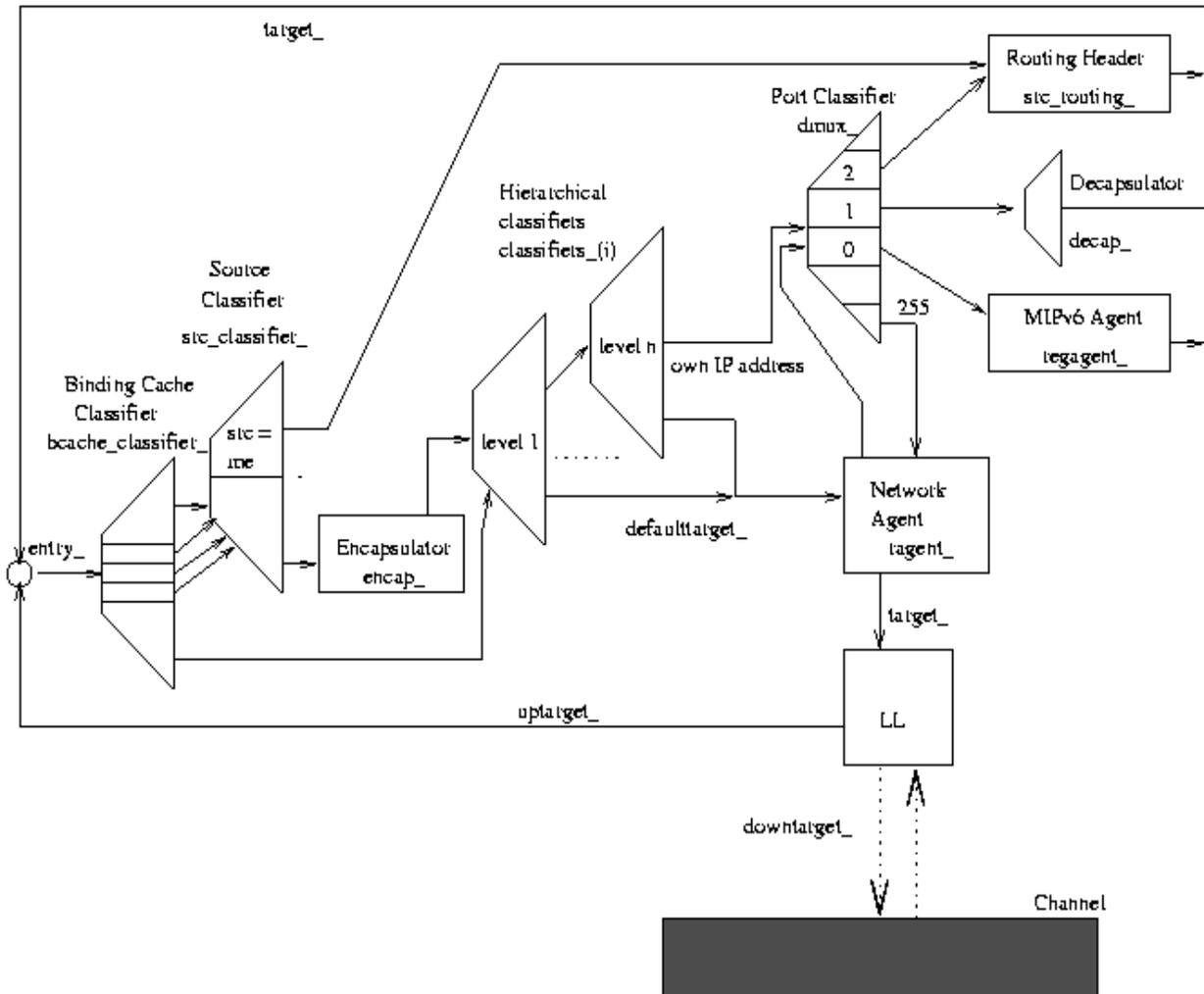


Figure 3-22 : nouvelle structure d'un nœud mobile ou d'un point d'accès

L'implémentation du protocole respecte les spécifications qui ont été décrites dans la partie 1, c'est pourquoi l'algorithme de MIPv6 n'est plus décrit ici.

➤ Agent Réseau

Cet agent est utilisé par les points d'accès et les nœuds mobiles. Il remplace l'agent de routage ad hoc dans la classe *MobileNode*. Cet agent instancie principalement la variable de prochain saut dans l'en-tête de paquet NS-2, vérifie le TTL³⁹ et transmet en les *Router Advertisement* et *Router Solicitation* broadcast aux agents MIPv6.

Quant au nœud mobile, l'agent Réseau-MN (*NetworkMN*) permet aux paquets envoyés à l'adresse temporaire d'être expédiés au port démultiplexeur. Il contrôle aussi les paquets

³⁹ TTL : Time To Live : durée de vie du paquet

entrants et informe l'agent du nœud mobile de la source du paquet en faisant appel au *MNAGENT* (ce qui aboutit à une nouvelle entrée dans la liste de correspondance).

► Agents Mobile IPv6

Le protocole Mobile IPv6 lui-même est mis en oeuvre par un ensemble d'agents NS-2. La classe de base est la classe *MIPv6Agent*, et les classes dérivées sont la classe *MNAgent*, la classe *CNAgent* et la classe *BSAgent*. Les agents MIPv6 se servent des particularités existantes :

- *Router Advertisement* et *Router Solicitation* : Actuellement traités directement par les agents MIPv6 (*MNAgent* et *BSAGENT*).
- La liste des points d'accès : Cette liste est maintenue par les nœuds mobiles grâce aux *Router Advertisement* envoyées par les points d'accès. Les entrées sont supprimées quand la durée de vie expire. Le nœud mobile écoute les *Router Advertisement* envoyés par les points d'accès. Quand il reçoit un *Router Advertisement* d'un point d'accès inconnu, il l'enregistre dans la liste des points d'accès avec une certaine durée de vie. Le nœud mobile obtient alors une nouvelle adresse temporaire et utilise ce nouveau point d'accès comme point d'accès par défaut.
- Obtention d'une nouvelle adresse temporaire : Les adresses ont deux composants : l'identifiant réseau (*network_id*) qui identifie le sous-réseau et l'emplacement dans la topologie et l'identifiant de nœud (*node_id*) qui identifie le nœud. L'identifiant réseau de l'adresse temporaire est donc le préfixe de l'adresse du point d'accès et identifie le sous-réseau visité ; Tandis que l'identifiant du nœud est l'identifiant de l'adresse mère du nœud mobile modulo 128. Le modulo 128 permet d'avoir 128 mobiles simultanément dans une simulation. L'objet ARP du nœud mobile est modifié pour répondre à des demandes pour l'adresse temporaire principale du nœud mobile.
- Liste de mise à jour des associations : Cette liste maintenue par les nœuds mobiles a deux buts : le maintien de la liste des correspondants actifs du nœud mobile et le maintien de l'information sur l'émission des *Binding Update*. Ne maintenir qu'une seule liste évite d'avoir à gérer deux listes distinctes et complémentaires dont le but commun est de savoir à qui envoyer un *Binding Update*.
- Une entrée indique l'adresse de destination, un flag spécifiant s'il faut envoyer un *Binding Update* à ce nœud ou non (non utilisé dans la mise en oeuvre actuelle), l'adresse temporaire, le temps, le numéro de séquence et la durée de vie du dernier *Binding Update* envoyé, plus des informations complémentaires (type de nœud pour distinguer entre point d'accès, agent mère et correspondant).
- La liste des correspondants inscrit les nœuds potentiels auxquels un *Binding Update* peut être envoyé. Ces nœuds sont automatiquement inséré par l'agent Réseau, ou explicitement par l'utilisateur à partir de

l'interpréteur OTcl, ou automatiquement par le MNAGENT lui-même.

- La liste des mises à jour des associations indique aux nœuds quel *Binding Update* a réellement été envoyé. Les entrées dans la liste peuvent être activées ou non.
- Optimisation de routage : il est possible de mettre en place la communication directe entre le nœud mobile et un correspondant, c'est-à-dire indiquer au correspondant la position courante du nœud mobile.
- Propagation par l'ancien point d'accès : il est possible de mettre en place la propagation du trafic destiné au nœud mobile au nouveau point d'accès. Par défaut, le nœud mobile envoie un *Binding Update* à l'ancien router d'accès. Par contre, aucune mécanisme de dé-enregistrement n'a été mis en place (car non spécifier pour le moment), donc les paquets peuvent tourner en boucle si le nœud mobile se déplace rapidement entre deux mêmes points d'accès.
- Emission des *Binding Update* : le nœud mobile envoie un *Binding Update* chaque fois qu'il obtient une nouvelle adresse temporaire ou que les entrées dans son cache sont sur le point d'expirer. Il envoie ce message à toutes les entrées activées dans sa liste de mise à jour en même temps.
- Cache d'association : quand un nœud reçoit un *Binding Update*, il met à jour son cache d'association. La table de routage doit alors être mise à jour pour rediriger les paquets dans la bonne direction. Dans NS-2, la table de routage est implémenté par un ensemble de classificateurs. Comme le montre la **Figure 3-21**, des nouveaux classificateurs ont été ajoutés pour déterminer si un paquet doit être redirigé ou pas.
- Collecte des statistiques : l'utilisateur peut charger l'historique des événements (attache aux points d'accès, entrée/sortie dans les listes, émission/réception des *Binding Update*...) pour effectuer des statistiques sur les simulations.

3.6.3.3 Mobilité dans un WAN

Pour tester la mobilité dans des grandes aires de réseaux (WAN), il faut différencier la mobilité locale de la mobilité globale. Un site est un réseau régi par une politique administrative commune. Tous les nœuds dans un site sont identifiés par le même préfixe IP. La longueur du préfixe détermine la taille maximale du site. Le site peut aussi être défini par un secteur géographique limité. Les nœuds mobile se déplacent librement dans la frontière géographique du site et sont rattachés aux différents routeurs du site. Un exemple d'un site peut être une cité universitaire.

Pour la mobilité WAN, nous avons besoin de :

- Moyens pour définir la notion de sites dans une topologie de réseau NS-2.
- Moyens pour qu'un nœud mobile puisse rester dans un site limité à un certain secteur géographique et ensuite se déplacer dans un autre site.

- Un modèle de mobilité qui distingue la mobilité locale d'une mobilité global

► Mobilité intra et inter site

La mobilité intra-site est la mobilité dans un seul domaine d'administration ou la mobilité dans un seul site (un campus, une organisation). La mobilité intra-site est traitée par l'implémentation de NS-2 existante (celle du CMU exposée section 3.2.1.6). La mobilité inter-site signifie le déplacement à travers les frontières de domaines, c'est-à-dire une mobilité entre deux domaines d'administration ou entre deux sites.

La mobilité inter-site est assurée par des nouvelles particularités qui permettent aux nœuds mobiles de se déplacer d'un site à un autre. Tous les sites sont associés à la même grille géographique, mais à un canal radio distinct. Tous les points d'accès dans le même site écoutent le même canal radio et sont associées à la même grille géographique. Le nœud mobile change son emplacement géographique dans la grille. Un mobile franchissant la frontière d'un site change donc le canal radio qu'il écoute. Pour conserver une certaine simplicité, le nœud mobile ne change pas ses coordonnées géographiques quand il passe d'un site à un autre. Ceci permet de continuer à employer le modèle de mobilité CMU existant, notamment pour les mouvements aléatoires. En effet, le canal sur lequel le nœud mobile écoute est transparent au modèle de mobilité.

► Mouvements

Un nœud mobile peut se déplacer par deux moyens : géographiquement et topologiquement. Des mouvements géographiques correspondent à la mobilité locale, c'est-à-dire la mobilité dans un site couvert par un secteur géographique. Le nœud mobile se déplace alors dans une grille comme le montre la **Figure 3-23**. Ces mouvements peuvent être générés aléatoirement ou par l'utilisation de la commande *setdest*.

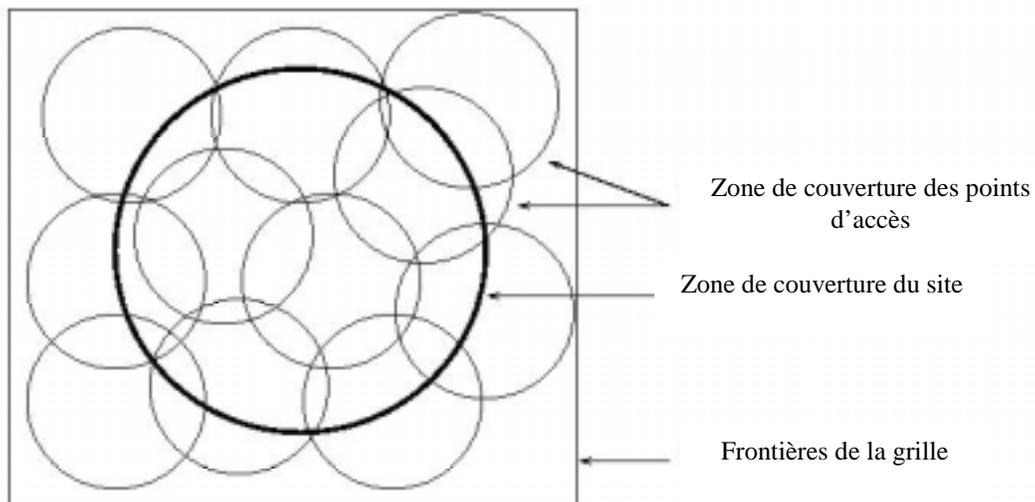


Figure 3-23 : mouvement à l'intérieur d'une grille

Les mouvements topologiques correspondent à la mobilité globale, c'est-à-dire d'un

site à un autre. Suite à un tel mouvement, le nœud mobile écoute un nouveau canal, celui correspondant au site auquel il entre. Le nœud mobile ne change pas en réalité ses coordonnées géographiques suite à un tel mouvement, mais il change son emplacement dans le topologie. Il entend donc les *Router Advertisement* envoyés par les points d'accès situé dans le nouveau site, c'est-à-dire les points d'accès ayant un préfixe d'adresse différent.

► Tests et commentaires

La mobilité globale n'a pas été évaluée avec plus qu'un nœud mobile. En réalité, au moins un changement est nécessaire pour réaliser des tests avec plus qu'un nœud mobile : pour limiter le nombre d'événements à manipuler, les points d'accès dans le site venant d'être quitté arrêtent d'émettre des *Router Advertisement*. Il faut donc changer la procédure `proc enter-site` pour effectuer des simulations avec plus d'un nœud mobile.

Par ailleurs une petite partie du code concerne le support du multicast pour les nœuds mobiles et les points d'accès (NS-2 existant ne permet pas à ces nœuds d'envoyer ou de recevoir du multicast). L'implémentation ajoutée permet uniquement à un nœud mobile d'émettre à un groupe mais pas de recevoir en tant que membre de groupe.

3.7 Conclusion

Cette partie ne constitue qu'une brève présentation de l'outil de simulation de réseau NS-2. d'autres fonctionnalités offertes comme les liaisons satellites, des techniques d'émulation pour utiliser un réseau réel, n'ont pas été exposées. NS-2 est un produit universitaire en cours de développement et non un produit commercial ; C'est pourquoi il comporte quelques erreurs aussi bien à la compilation que lors des tests spécifiques. Il existe une mailing liste ou les problèmes des utilisateurs et souvent les solutions leurs sont données.

La rapide évolution des versions s'avère être à la fois un inconvénient et une qualité. Un inconvénient car toutes ces extensions ajoutées par des groupes de chercheurs tournent pas forcément sur les mêmes versions. L'utilisation de ces extensions demande donc beaucoup de patience puisque chaque version a ces problèmes d'installation. Une qualité car NS-2 devient rapidement de plus en plus robuste, complet et son domaine d'application augmente.

PARTIE 4

GESTION D'INTERFACES MULTIPLES

4 Gestion d'interfaces multiples

4.1 Description du projet de gestion d'interfaces multiples sans fil

4.1.1 Les objectifs

L'objectif principal de ce projet que nous appellerons MIMP est de mettre en place des solutions technologiques permettant aux terminaux mobiles IPv6 d'avoir un accès simultané ou successif à plusieurs technologies sans fil. Les nouvelles fonctionnalités développées permettront une gestion intelligente de la mobilité des équipements IPv6. Une interface générique interne au terminal permettra l'accès aux informations spécifiques de chaque technologie d'accès sans fil ce qui facilitera la gestion des handoffs et le choix d'une interface radio. Lorsque plusieurs interfaces permettront simultanément une connexion au réseau, l'équipement choisira le support en fonction de différents critères comme la qualité de services et le coût d'accès au support. Les applications adapteront également leur comportement en fonction de ces critères.

Ces solutions technologiques permettront à terme de s'intégrer dans des solutions de mobilité globale dépassant le cadre de ce projet. Ainsi, la mobilité du point de vue de l'utilisateur peut se résumer en trois points: un service de télécommunication omniprésent (à la maison, dans la rue, dans la voiture, au bureau, dans le train, à l'aéroport, au café) ; un objet communicant simple pour toutes les applications (voix, données, images...) ainsi qu'un service continu, efficace et bon marché (la communication doit rester ouverte quels que soient l'endroit, le réseau support, le fournisseur de service, l'application utilisée). Ces trois exigences peuvent se traduire par la mise en place d'un service de mobilité globale (universelle) avec les contraintes suivantes :

Des réseaux supports présents partout et tous différents : courte et moyenne portée (Bluetooth), moyenne portée et très haut débit (IEEE 802.11 [38] et HiperLAN/2 [40]), longue portée et bas à moyens débits (GSM [20], GPRS [20], UMTS [21][22][23][24]). Ces normes seront décrites dans la section 4.2. Tous ces supports sont fournis en général par des opérateurs différents (global, local, public, privé).

Un terminal mobile unique avec le support de divers service de communication (téléphonie, messagerie, web...) et différents services d'accès (privé, public, abonnement...).

Une gestion de la mobilité à tous les niveaux : lors du changement du point d'attache, lors du changement de sous-réseaux, lors du changement du sous-réseau support ou lors du changement du service d'accès.

Afin de permettre l'émergence d'un tel service de mobilité globale, le projet se focalise sur la fédération de différents supports de communication sans fil au sein d'un terminal. Cela permettra une meilleure prise en compte de la mobilité des équipements et des utilisateurs. Les mécanismes développés porteront sur la détection d'une nouvelle

ressource de transmission sans fil, la décision de basculement sur cette nouvelle ressource de transmission, la mise en place et l'optimisation des procédures de basculement, le basculement du trafic sur la ou les ressources de transmission disponibles et l'adaptation du trafic aux niveaux de la qualité et de la sécurité en fonction des caractéristiques de la nouvelle ressource de transmission sans fil. Ces mécanismes, limités au terminal, doivent permettre à terme de gérer la mobilité dans les réseaux IPv6 de manière globale, efficace et transparente pour le client et ce sur les plans de la technique, du service, de la qualité de service et de la sécurité.

4.1.2 Etat de l'art

Initialement, le monde de l'Internet à travers son organisme de standardisation, l'IETF, 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 le protocole Mobile IP [1][2] qui a été présenté dans la partie 1. Les travaux récents qui étudient l'utilisation des réseaux cellulaires (Cellular IP [14][15][16][17], Hawaii [18]...) montrent les différents problèmes posés pour la gestion des différentes mobilités des terminaux et la prise en compte rapide des déplacements. La plupart de ces travaux sont proposées 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 (voir partie 2).

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 plusieurs projets qui étudient l'Internet nouvelle génération, mais à notre connaissance il n'en existe pas (ou très peu) 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.

Dans le domaine applicatif, l'adaptation dynamique aux conditions d'exécution en général et aux variations de la qualité de connexion en particulier a été considérée dans le cadre des applications multimédia gérant la qualité de service. Néanmoins, ces travaux proposent des mécanismes d'adaptation dynamique adaptés à une application ou à un type d'applications particulier. Développer des nouvelles applications adaptatives requiert donc la conception répétée de ces mécanismes. En outre, les systèmes existant actuellement manquent d'une approche intégrée à l'adaptation des couches de plus bas niveau ce qui permettrait d'effectuer de meilleurs choix lors des adaptations.

4.1.3 Cadre du projet et partenaires

Le projet MIMP représente l'objet de ma thèse qui débutera en septembre 2001. La thèse s'effectue dans le cadre d'un contrat de recherche financé par France Télécom⁴⁰. Elle fait partie d'un projet plus vaste intitulé projet CYBERTÉ⁴¹. Le projet CYBERTÉ est financé par le RNRT [42]. Il est orienté vers des thèmes de recherche et développement précompétitifs. Les partenaires du projet regroupent les centres de Recherche et Développement France Télécom pour les réseaux cellulaires, Cisco⁴² pour les réseaux locaux sans fil, l'IRISA⁴³ pour le développement d'applications s'adaptant aux environnements mobiles et l'ENST⁴⁴ Bretagne pour la gestion de la qualité de service. Le projet CYBERTÉ intègre 4 sous-projets qui sont :

- Sous-projet 1 : description d'une architecture multi-interfaces (architecture MIMP). Cette architecture est décrite dans la section suivante.
- Sous-projet 2 : gestion des profils de qualité de service multiples ; l'objectif est de définir les profils de qualité de service utilisés pour mesurer les capacités de transmission et les mécanismes d'adaptation pour différentes interfaces d'un système.
- Sous-projet 3 : définition d'une architecture générique facilitant le développement d'applications adaptant leurs besoins en qualité de service aux conditions d'accès au réseau. L'objectif est de proposer des mécanismes permettant aux applications, d'une part de réagir aux variations de l'environnement indiquées par les mécanismes d'alerte fournis par l'interface développée dans le sous-projet 1 et, d'autre part, d'informer la couche réseau de leurs besoins à l'aide des profils définis dans le sous-projet 2.
- Sous-projet 4 : maquette et démonstrations

4.2 Présentation de l'architecture MIMP

L'objectif principal du projet est de doter les terminaux mobiles IPv6 d'une nouvelle architecture de communication capable d'accéder à plusieurs technologies de transmission sans fil. Les terminaux mobiles bénéficiant de cette nouvelle architecture pourront s'adapter dynamiquement aux contraintes de qualité de services en remontant aux applications des informations sur les différentes interfaces de communication sans fil disponibles.

L'architecture permettra de mettre en œuvre des solutions pour :

- Offrir une meilleure gestion de la bande passante. Ce mécanisme doit permettre à certaines applications de choisir l'interface qui leur convient le mieux. Ce choix peut être effectué suivant différents critères (débit, coût...).
- Offrir les fonctionnalités de sélection de l'adresse source. En effet, à l'heure

⁴⁰ France Télécom : <http://www.francetelecom.fr>

⁴¹ Projet CYBERTE : projet RNRT décrit à l'url http://www.telecom.gouv.fr/rnrt/suivi/res_01_9.htm

⁴² Cisco : <http://www.cisco.com>

⁴³ IRISA : <http://www.irisa.fr>

⁴⁴ ENST Bretagne : <http://www.enst-bretagne.fr/>

actuelle une application ne choisit pas l'adresse avec laquelle elle émet des paquets. Ce choix se fait simplement en fonction de l'interface d'émission. Dans le cas de la mobilité IPv6, le choix de l'adresse source est particulièrement important. En effet, un mobile peut changer rapidement de point d'attachement sur l'Internet. Ce changement impose également un changement d'adresse réseau, dans de nombreux cas, il peut être utile de pouvoir prendre en compte ce changement sans que pour autant les applications en soient averties, notamment dans le cadre des communications multicast.

- Disposer d'outils de gestion des interfaces. Il est également très important de pouvoir piloter correctement les interfaces réseau dont dispose un équipement. Aussi, l'architecture offrira un mécanisme de gestion et de pilotage du comportement à associer à une interface.
- Prendre en compte les interfaces de "backup" afin de ne pas interrompre des communications en cours suite à la panne de l'un des réseaux utilisés par l'équipement mais également d'automatiser le basculement des communications de manière transparente pour les applications vers un réseau (à travers une interface de backup) de secours.
- Optimiser les handoffs. La nature même des supports sans fil nécessite un traitement particulier. Or à l'heure actuelle, la majorité des drivers de carte réseau sans fil cherche à masquer ces particularités. Il est important de développer des échanges de contrôle avec les drivers des interfaces (mise en œuvre de l'interaction entre les couches 2 et 3 dont on a parlé dans la partie 2). En effet un mobile IPv6 est amené à se déplacer. Ce déplacement peut-être plus ou moins rapide. Jusqu'à présent la prise en compte du déplacement d'un mobile est relativement lente. Elle dépend généralement de la prise en compte de la diffusion des préfixes réseau annoncés par les routeurs. Cette diffusion peut prendre plusieurs secondes. Afin de mieux gérer cette mobilité, une prise en compte des informations reçues et émises par les drivers des interfaces permettra de réduire ces temps. De plus un mobile peut se déplacer à l'aide d'une même technologie sans fil. Mais nous souhaitons également lui offrir la possibilité de se déplacer de manière transparente à l'aide de plusieurs technologies. Il est donc nécessaire d'étudier des mécanismes de handoffs à différents niveaux.

L'architecture MIMP qui doit réaliser toutes ces fonctionnalités est présentée dans la **Figure 4-1**.

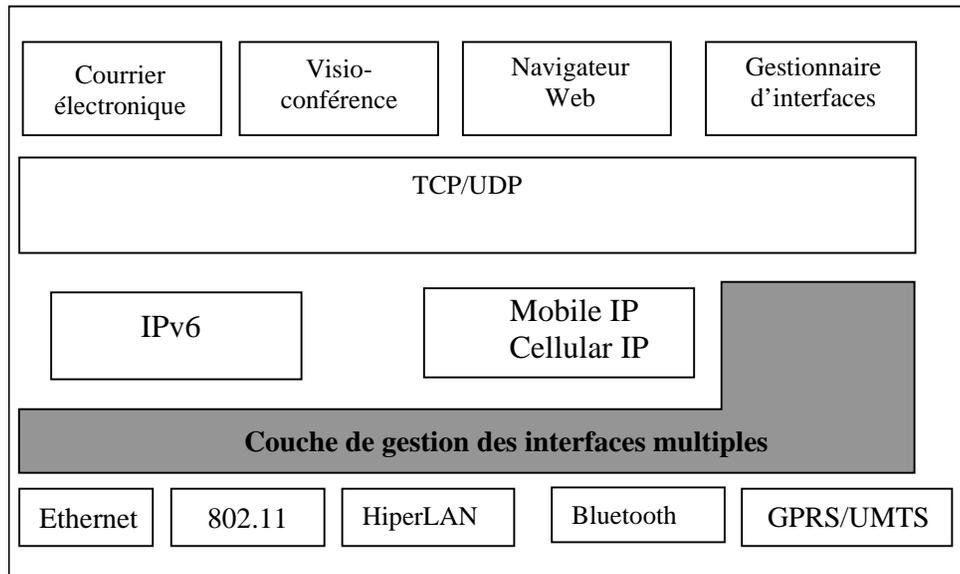


Figure 4-1 : vue générale de l'architecture intégrant MIMP

Comme le montre la **Figure 4-1**, une gestion globale de la mobilité est proposée à travers la définition d'une couche d'abstraction qui permettra d'une part de définir une API⁴⁵ utilisable à la fois par les applications mais également par la couche Réseau IPv6, et d'autre part de tirer au mieux parti des spécificités de chacun des supports de communication sans-fil sous-jacents.

Grâce à des mécanismes de gestion unifiés et génériques, cette architecture permettra d'intégrer tout type d'interface radio existante ou à venir. Dans un premier temps, on a prévu d'étudier quelques-unes de ces normes radio pour identifier les informations pertinentes nécessaires aux couches supérieures. Dans la section 4.3 quelques-unes de ces normes sont brièvement présentées. Cependant il ne faut pas perdre de vue que beaucoup de normes sont encore en cours de standardisation (UMTS, HiperLAN/2...). C'est pourquoi l'architecture de communication MIMP doit être modulaire et paramétrable pour intégrer efficacement les normes et les technologies qui verront le jour ultérieurement.

Le développement durant ma thèse se déroulera en deux étapes ; tout d'abord l'intégration de cette architecture se fera dans le simulateur réseau NS-2 qui a été présenté dans la partie 3. Une première idée de la réalisation de l'intégration de l'architecture MIMP dans NS-2 sera présentée dans la quatrième section. Cette implémentation fera l'objet de premiers tests. Ensuite, l'implémentation réelle sera effectuée sur le système d'exploitation FreeBSD avec la pile IPv6 livrée en standard.

4.3 Normes prises en considérations

Dans cette section seront décrites brièvement les principales normes de

⁴⁵ API : Application Programmable Interface – interface programmable d'application

communication qui serviront de points de repère pour la conception de l'architecture générique. Il s'agit de la norme IEEE⁴⁶ 802.11 développée principalement aux Etats-Unis, de la norme HiperLan/2 qui est en cours de développement et de Bluetooth, produit européen qui s'intègre de plus en plus dans les appareils grand public. Suivant leurs spécificités, ces normes représentent une alternative au réseau filaire ou simplement l'abstraction de câblage. Elles proposent des technologies de communication sans fil associée à des protocoles d'accès au canal radio. Elles peuvent définir une méthode de communication soit entre un nœud mobile et un point d'accès, soit entre deux nœuds mobiles. Dans le projet, on s'intéresse uniquement à la communication entre un nœud mobile et un point d'accès.

4.3.1 IEEE 802.11

La norme 802.11 [38] est la plus utilisée dans les réseaux sans fil à ce jour. Elle joue le même rôle que 802.3 [39] pour Ethernet. La norme définit trois fonctions de base pour établir la communication :

- Une interface de propagation radio pour qu'il y ait une interopérabilité entre les fournisseurs.
- Une méthode de codage et de modulation : *Frequency Hopping Spread Spectrum* ou *Direct Sequence Spread Spectrum*.
- Une couche MAC qui gère l'accès au canal : *CSMA*⁴⁷ ou *Virtual Carrier Sense*.

La bande de fréquence utilisée est dans les 2,4 GHz et se divise en quatre canaux. Un point d'accès dispose donc de quatre canaux différents pour communiquer avec les stations. Une station utilisera toujours le point d'accès qui offre le meilleur signal. Le débit maximal est de 11 Mbps.

Au niveau topologique, on parle d'ensemble de service basique (BSS⁴⁸) quand deux nœuds mobiles (communication ad hoc) ou un nœud mobile et un point d'accès se sont authentifiés et ont établi une communication (voir **Figure 4-2**). Un ensemble de service étendu (ESS⁴⁹) est une série de BSS contenant chacun un point d'accès, connectés ensemble par un réseau filaire (voir **Figure 4-2**).

⁴⁶ IEEE : <http://www.ieee.org>

⁴⁷ CSMA : Carrier Sense Multiple Access

⁴⁸ BSS : Basic Service Set

⁴⁹ ESS : Extended Service Set

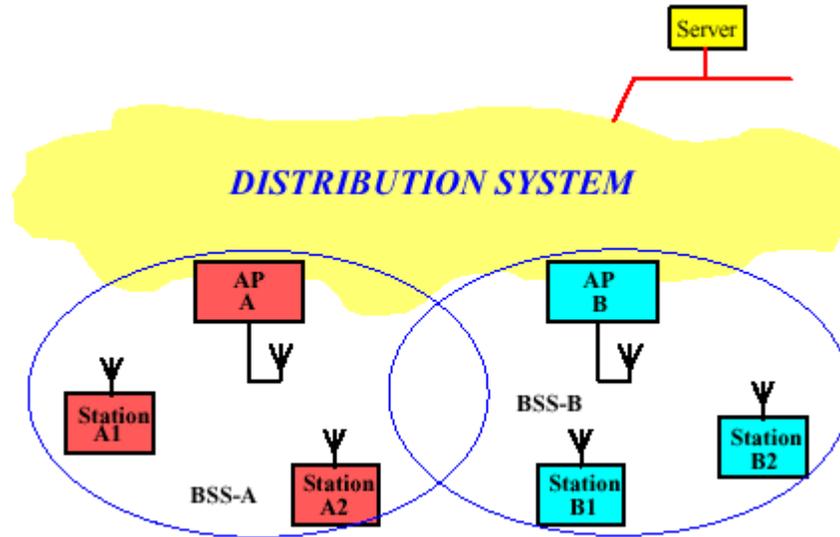


Figure 4-2 : topologie utilisée dans IEEE 802.11

Généralement, les points d'accès sont placés de manière à ce que leurs cellules de couverture se recouvrent. Ce recouvrement permet les déplacements sans coupure des nœuds mobiles entre les points d'accès. La norme 802.11 ne définit que les principes de base du déplacement (active et passive scanning, processus de ré association...).

4.3.2 HiperLAN/2

Pour le moment la norme HiperLAN/2 [40] offre uniquement la communication entre les nœuds mobiles et les points d'accès. La communication directe entre nœuds mobiles est encore en développement. La topologie proposée est plus ou moins la même que celle pour IEEE 802.11, à savoir des points d'accès reliés par un réseau filaire, fournissant une connexion sans fils aux hôtes mobiles. Cette norme fournit un haut débit, 54 Mbps au niveau de la couche physique et jusqu'à 25 Mbps au niveau de la couche IP. Le protocole MAC d'accès au canal radio utilise une forme de division temporelle dynamique. L'allocation de fréquence (dans la bande des 5GHz) est dynamique, comme on peut le trouver dans les réseaux GSM ; au démarrage, les points d'accès sondent les différents canaux et choisissent le plus approprié en fonction de ceux utilisés par les points d'accès voisins.

HiperLAN/2 inclut une gestion de qualité de service. Toute communication est en mode orienté connexion, c'est-à-dire que deux équipements communicants doivent d'abord s'échanger des informations de sécurité ou au moins d'authentification avant toute transmission de données. Chaque connexion peut être affectée d'une qualité de service spécifique (largeur de bande passante, taux d'erreur par bit...). HiperLAN/2 fournit aussi des mécanismes d'authentification et de cryptage pour régler les problèmes de sécurité.

La pile du protocole HiperLAN/2 possède une architecture flexible pour permettre une adaptation facile et une intégration dans diverses variétés de réseaux fixes, comme

Ethernet ou un réseau cellulaire troisième génération. Le modèle de référence du protocole HiperLAN/2 est décrit dans la **Figure 4-3**.

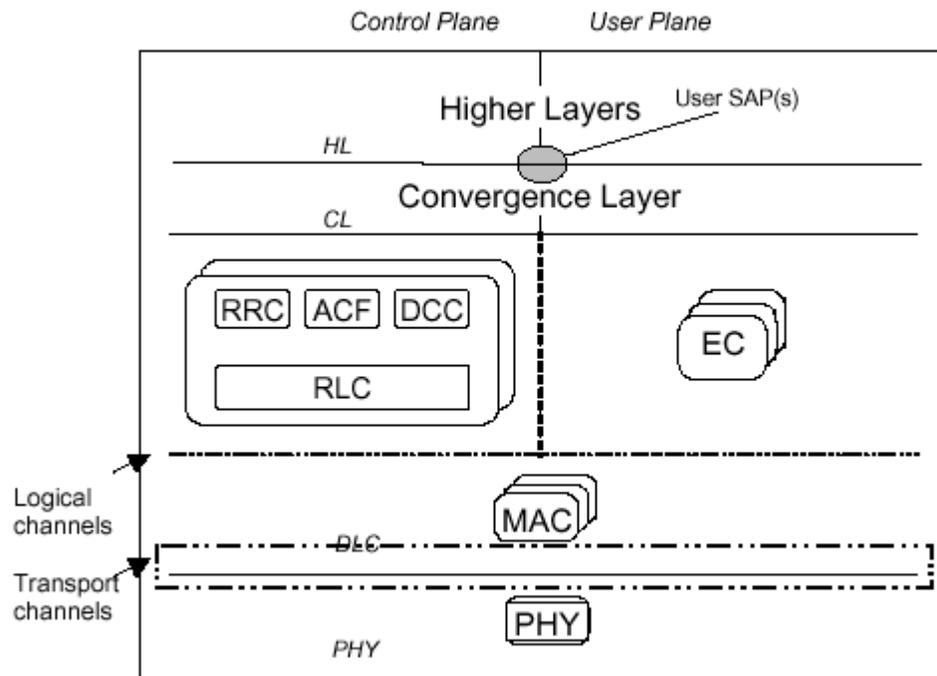


Figure 4-3 : modèle de référence de HiperLAN/2

Le déplacement des nœuds mobiles entre les points d'accès peut se faire de deux manières dans HiperLAN/2 : soit par ré association, soit par une signalisation dans le réseau fixe. Le processus de ré association est assez coûteux puisqu'il nécessite que le nœud mobile recommence tout le processus d'association (authentification, autorisation...). Dans la deuxième solution, le nouveau point d'accès reçoit les informations nécessaires sur le nœud mobile par le réseau filaire qui interconnecte les points d'accès.

4.3.3 Bluetooth

Bluetooth [41] ne s'inscrit pas dans la même gamme que les deux normes présentées ci-dessus. Bluetooth définit une liaison courte distance, économique mais à faible débit, pour remplacer le câblage. Les points forts de cette norme sont sa robustesse, sa faible complexité, sa faible consommation de courant et son prix.

Bluetooth opère dans la bande de fréquence de 2,4 GHz. Le débit symbolique est de 1 Mbps sur un canal à division temporelle. Le système se décompose en une unité radio, une unité de contrôle de liaison et une unité de gestion de liaison et de primitives.

Plusieurs équipements partageant le même canal sont appelés **piconet**. Chaque piconet est hiérarchisé en un maître et un ou plusieurs esclaves. Le mode de communication peut

être orienté connexion ou non. Lorsqu'il s'agit d'une communication point à point, la liaison est synchrone orientée connexion. Lorsqu'il s'agit d'une communication point à multipoint entre un maître et des esclaves, la liaison est asynchrone en mode non connecté.

4.3.4 Comparatif des trois normes et portée des solutions technologiques

Le **Tableau 4** décrit les caractéristiques principales des trois normes.

Caractéristique	802.11	HiperLAN	Bluetooth
Spectre de fréquence	2,4 GHz	5 GHz	2,4 GHz
Débit maximal	11 Mbps (version b)	54 Mbps	1 Mbps
Contrôle d'accès au média	CSMA/CA	TDMA/TDD	TDMA/TDD
Connexion	Sans	Avec	Avec ou sans
Multicast	Oui	Oui	Oui
Mode économique	Oui	Oui	Oui
Qualité de service	PCF	ATM/802.1p/RSVP/DiffServ	Gestionnaire de liaison
Réseau fixe sous-jacent	Ethernet	Ethnet, IP, ATM, UMTS...	Multiple

Tableau 4 : comparatif des trois normes

La **Figure 4-4** qui suit montre grossièrement les solutions actuelles pour l'interconnexion d'équipements en terme de débit, ainsi que la portée ou plutôt les lieux d'utilisation. Sont représentés les LAN qui offrent le débit le plus élevé, mais est limité à des postes fixes en intérieur. Bluetooth offre une mobilité locale de faible portée avec un débit moindre. Les LAN sans fil actuel (IEEE 802.11 généralement) offrent un débit allant jusqu'à 10 Mbps mais aucun protocole ne gère actuellement les déplacements de manière assez efficace pour que les équipements puissent se déplacer dans n'importe quelle partie de l'espace. Par contre les réseaux cellulaires offrent une totale liberté de mouvement (dans la limitation des zones couvertes) mais avec un débit trop faible pour du trafic IP.

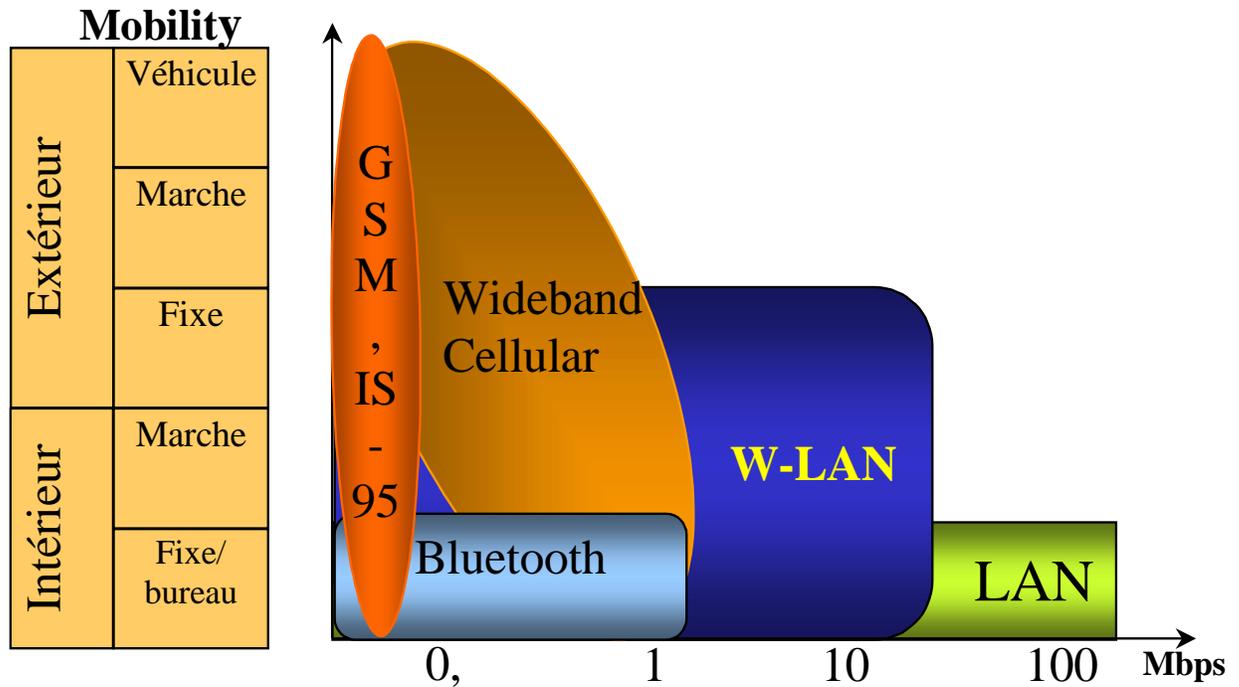


Figure 4-4 : débit et portée de différents modes de connexion

4.4 Introduction de l'architecture MIMP dans NS-2

4.4.1 Planification

La première étape dans la réalisation de l'architecture MIMP sera de l'implémenter dans le simulateur réseau NS-2 qui a été présenté dans la partie 3. On pourra ainsi s'apercevoir des performances qu'elle pourra procurer au sein des équipements mobiles. Actuellement, la norme 802.11 est implémenté en standard dans NS-2 et Bluetooth est une extension basée sur la version 6 de NS-2. Un premier travail pourrait donc concerner la gestion de ces deux interfaces sans fil et de l'interface utilisée par les nœuds fixes.

Dans NS-2 les protocoles d'accès au médium sont implémentés dans la classe *MAC*. La couche MAC est la couche intermédiaire entre le canal physique et la couche liaison. Les fichiers `mac.{h,cc}` contiennent les états et méthodes de configuration générales applicables à tout type de protocole MAC. La classe *MAC* est ensuite dérivée pour des implémentations plus spécifiques comme pour le protocole CSMA (classe *CsmaMac*) ou 802.11 (classe `mac-802_11`).

L'objectif est de prendre la meilleure interface disponible à un moment donné. Mais pour cela, il faut pouvoir comparer la qualité des interfaces radio. La meilleure interface est celle qui offre le meilleur débit par rapport à son coût et à l'application en cours. Il faudra donc ajouter aux implémentations des technologies un attribut coût qui entrera en compte lors de la décision. De même, il faudra pouvoir distinguer quelle application

nécessite un plus grand débit que telle autre. Ceci peut être fait dans NS-2 par la mesure du taux de trafic généré.

4.4.2 Implémentation d'un nouvel objet

La gestion de ces interfaces passe par l'ajout d'un nouvel objet. C'est cet objet qui devra détecter quelles interfaces sont disponibles, quel est leur coût et quels sont les besoins de la communication. Cet objet devra intégrer la structure interne d'un nœud mobile, se situant au même niveau que l'interface de file d'attente. L'objet gestionnaire et ses interactions avec les autres objets sont représentés dans la **Figure 4-5**.

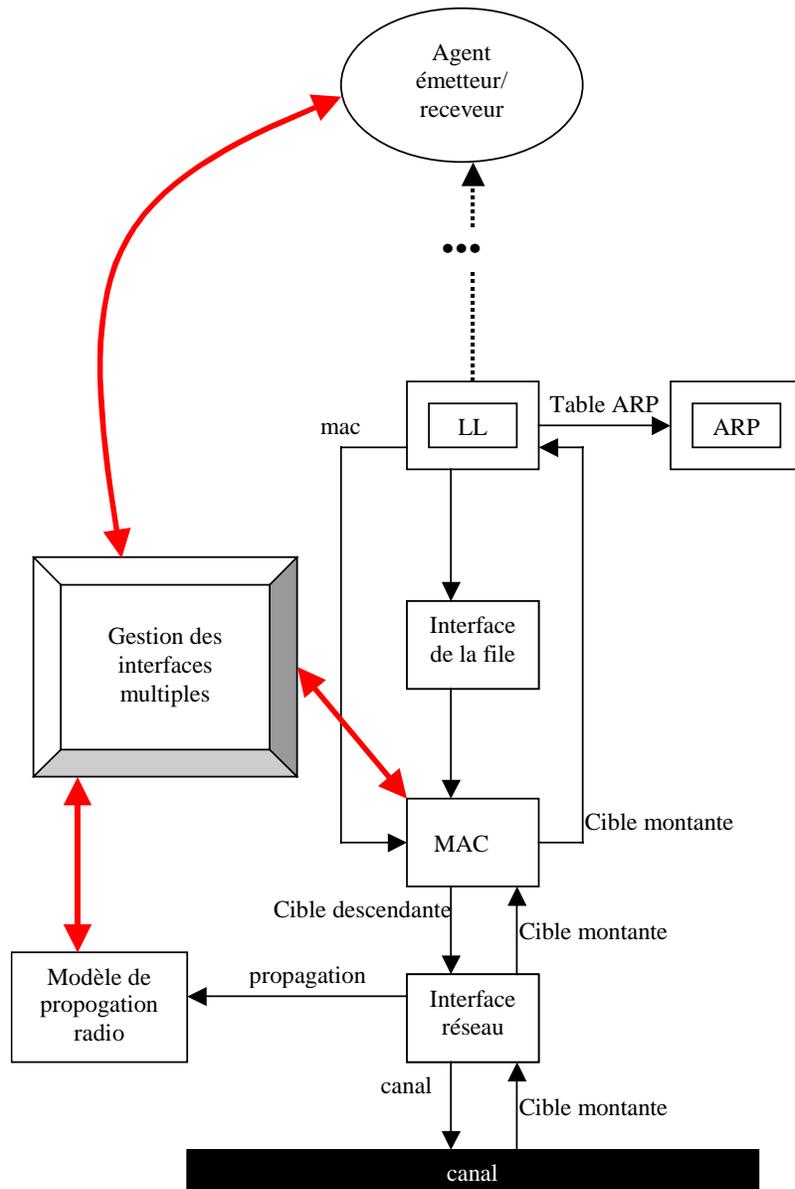


Figure 4-5 : gestion d'interfaces multiples dans NS-2

Le modèle de propagation radio enregistre l'intensité du signal à chaque paquet reçu. Effectivement les paquets voyageant sur les interfaces radio contiennent un indicateur d'intensité du signal. Si celui-ci est en dessous d'un certain seuil, il est marqué avec un flag d'erreur et le protocole de la couche MAC le détruira. Il serait très intéressant d'utiliser cette mesure d'intensité du signal au niveau du gestionnaire d'interfaces multiples. C'est pourquoi il faut établir un échange d'information entre ces deux objets.

D'autre part, le gestionnaire d'interface multiple devra communiquer avec l'objet MAC pour lui donner l'interface à utiliser. L'objet MAC devra donc utiliser tel ou tel protocole selon les décisions prises au niveau du gestionnaire.

L'analyse des besoins de l'application peut passer par l'agent émetteur/receveur selon le cas du nœud en question. Le traitement sera évidemment différent selon que cet agent est un émetteur ou un récepteur puisque les nécessités ne sont pas les mêmes (cf. agent TCP dans Cellular IP partie 3 section 3.6.2.2). Ceci nécessite cependant une petite intervention dans les fichiers correspondants afin de pouvoir extraire des informations comme le débit minimum et de pouvoir les communiquer au gestionnaire.

L'objet gestionnaire sera implémenté par une classe dérivée de la classe *Connector*. Elle contiendra donc les méthodes de communication avec les objets MAC, modèle de propagation radio et agent récepteur/émetteur ainsi que des méthodes de mesure et de décision. La classe *MAC* devra devenir une sous-classe d'une classe au-dessus d'elle qui permettra de changer de protocole d'après les données fournies par le gestionnaire. Cette nouvelle classe sera celle qui instanciera tel ou tel protocole MAC.

Il suffira d'ajouter quelques méthodes au modèle de propagation radio pour qu'il puisse communiquer les informations pertinentes à la classe gestionnaire. Par contre, il faudra définir une échelle de graduation pour que les agents puissent exprimer les besoins des applications. La classe *Application* se verra alors augmentée d'un nouvel attribut que l'agent du nœud devra communiquer au gestionnaire.

4.4.3 Conclusion

La courte durée de mon stage ne m'a malheureusement pas suffi pour commencer l'implémentation. C'est pourquoi les explications ci-dessus ne sont qu'une première vision de l'implémentation de l'architecture dans NS-2 et viendra probablement à être modifiée.

Un autre problème se pose pour l'implémentation de cette architecture ; pour pouvoir utiliser Bluetooth ou même les extensions Noah, Mobiwlan ou CIMS (présentées partie 3 section 3.6) il est nécessaire d'utiliser la version 6 de NS-2. Or aujourd'hui la version 8 est déjà disponible. Cette nouvelle version comporte non seulement des outils supplémentaires mais est beaucoup moins buggé. Il sera donc probable qu'il faille travailler sur plusieurs versions au départ de manière générique pour pouvoir utiliser le plus d'extensions possibles.

CONCLUSION

Dans ce rapport ont été développés les problèmes et les solutions liés à la mobilité dans l'Internet d'aujourd'hui à travers divers organismes de recherche. Un ensemble de ces propositions a été étudié dans les deux premières parties. Les solutions proposées ne sont en général qu'à leur phase de spécification et sont donc en constante évolution. Afin d'évaluer de tels protocoles, le simulateur réseau NS-2 est fréquemment utilisé. La présentation de ce simulateur a été faite dans la partie 3. Cette présentation s'est focalisée sur l'implémentation existante et à venir de la mobilité. La dernière partie du rapport concerne plutôt nos travaux à venir, à savoir l'implémentation d'une nouvelle architecture pour gérer plusieurs interfaces radio dans un même équipement.

La gestion de la mobilité

Le protocole Mobile IP est la solution actuelle pour résoudre les problèmes de rupture de communication durant les déplacements des nœuds mobiles dans des réseaux IP. Ce protocole permet donc aux nœuds mobiles de se déplacer de réseaux en réseaux sans rompre leurs sessions en cours. Le nœud 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, agent qui se charge d'intercepter les paquets dans le réseau principal du nœud mobile et de les lui transmettre à sa position courante dans l'Internet.

Mobile IP existe en deux versions, MIPv4 et MIPv6, selon la version du protocole IP utilisé. La version 6 n'est encore qu'un draft de l'IETF (en cours de normalisation) alors que la version 4 est un RFC de l'IETF (normalisée). MIPv6 offre beaucoup plus de fonctionnalités que MIPv4, comme l'optimisation de route ou la suppression d'un agent pour la mobilité (l'agent visité). Par contre, la mise en œuvre des optimisations offertes par MIPv6 nécessite que tous les nœuds qui communiquent avec un nœud mobile supportent des fonctionnalités comme la gestion de cache d'association ou la gestion d'en-tête de routage.

Bien que MIP permette à des nœuds mobiles de se déplacer tout en continuant leur communication, des pertes de paquets ou des déséquences peuvent avoir lieu. Ces pertes peuvent se faire ressentir au niveau des applications temps réel comme de la vidéo ou la voix sur IP. Pour résoudre ces problèmes, les groupes de travail Seamoby et MobileIP de l'IETF proposent des améliorations de la gestion de la mobilité et particulièrement du traitement du handoff (procédure de déplacement). Ces solutions 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 nouveaux routeurs d'accès lors d'un handoff. Cet échange d'information permet au nouveau routeur d'accès d'identifier le nœud mobile et de lui fournir un service plus rapidement. L'échange peut avoir lieu avant même que le nœud mobile ne se détache de son ancien routeur d'accès, par une anticipation du déplacement. Cette anticipation permet de réaliser des fast

handoffs. Le bi casting est une autre solution qui a été présentée. Le bi casting est la duplication du trafic destiné à un nœud mobile à son ancienne à sa nouvelle localisation.

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 nœuds mobiles à l'intérieur d'un domaine. Le nœud mobile doit juste communiquer son entrée dans un domaine à son agent mère (et éventuellement ses correspondants), et ensuite ses mouvements sont gérés par le domaine.

Cellular IP est un autre protocole de gestion de la micro mobilité (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. L'information de routage est totalement distribuée dans tous les nœuds IP cellulaires. Cellular IP est prévu pour fonctionner avec un grand nombre de mobile et gérer les handoffs de manière optimisée. Cellular IP interagit avec MIP pour ce qui est de la mobilité entre domaines.

Simulateur réseau NS-2

NS-2 est le simulateur réseau le plus utilisé dans la communauté de recherche en réseau. Il implémente en C++ et OTcl tout un ensemble de classes pour définir les concepts de nœuds, liaisons et agents. La structure d'un nœud est fortement inspirée par le modèle en couche OSI. La mobilité a été introduite dans NS-2 dans sa version 2 par le CMU. Au départ, il s'agissait uniquement de simuler des LAN ad hoc, c'est-à-dire tous les nœuds de la simulation devaient être mobiles. Plus tard est venue l'implémentation de MIPv4 qui a permis de faire des simulations sur des topologies avec à la fois des nœuds mobiles et des nœuds fixes. Plusieurs extensions sont notamment disponibles sur différents sites de recherche qui implémente des protocoles particuliers, comme Cellular IP ou MIPv6.

Perspectives

Ce stage de DEA avait pour objectif de faire l'état de l'art sur la gestion actuelle de la mobilité des hôtes. C'était le premier pas dans un projet plus vaste s'inscrivant dans un projet RNRT [42]. Etant donné l'émergence de diverses et nombreuses technologies de communication, il est probable que dans un proche futur un hôte ait plusieurs interfaces de communication. Il devient alors impératif qu'un tel équipement puisse gérer de manière intelligente ses interfaces de communication. Pour une gestion efficace des interfaces, on se propose de rajouter une couche dans l'architecture de la pile réseau d'un hôte. Cette couche gestionnaire d'interfaces devra prendre en compte les besoins des applications, déterminer quelles interfaces sont utilisables, quels sont leurs coûts (en terme de débit, d'abonnement...) et sélectionner l'interface la plus appropriée.

L'implémentation de cette nouvelle couche se fera dans un premier temps dans le simulateur réseau NS-2. Elle sera intégrée dans la structure d'un nœud et utiliser les modèle de propagation disponible dans NS-2. Ensuite elle intégrera la pile réseau d'un terminal.

REFERENCE BIBLIOGRAPHIQUES

- [1] C. Perkins, "Mobile IP", Internet Engineering Task Force Request For Comment 2002, octobre 1996.
- [2] D. Johnson, C. Perkins, "Mobility support in IPv6", Internet Engineering Task Force draft-ietf-mobileip-ipv6-13.txt, Novembre 2000.
- [3] J. Manner, M. Kojo, T. Suihko, P. Eardley, D. Wisely, R. Hancock, N. Georganopoulos, "Mobility Related Terminology", Internet Engineering Task Force draft-manner-seamoby-terms-01.txt, Mai 2001.
- [4] K. El-Malki, P. Calhoun, T. Hiller, J. Kempf, P.J. McCann, A. Singh, H. Soliman, S. Thalanany, "Low latency Handoffs in Mobile IPv4", Internet Engineering Task Force draft-ietf-mobileip-lowlatencyhandoffs-v4-00.txt, Février 2001.
- [5] 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.
- [6] O. Levkowitz, P. Calhoun, G. Kenward, H. Syed, J. Manner, M. Nakhjiri, G. Krishnamurthi, R. Koodli, K. Atwald, M. Thomas, M. Horan, P. Neumiller, "Problem Description: Reasons For Doing Context Transfers Between Nodes in an IP Access Network", Internet Engineering Task Force draft-ietf-seamoby-context-transfer-problem-stat-00.txt, Février 2001.
- [7] R. Koodli, C. Perkins, "A Context Transfer Framework for Seamless Mobility", Internet Engineering Task Force draft-koodli-seamoby-ctv6-00.txt, Février 2001.
- [8] G. Krishnamurthi, R. Chalmers, C. Perkins, "Buffer Management for Smooth Handovers in IPv6", Internet Engineering Task Force draft-govind-seamoby-buffer6-00.txt, Février 2001.
- [9] Y. Ezaki, Y. Imai, "Mobile IP6 handoff by Explicit Multicast", Internet Engineering Task Force draft-ezaki-handoff-xcast-00.txt, Novembre 2000.
- [10] "Small Group Multicast (sgm) BOF" agenda, <http://www.ietf.org/ietf/00jul/sgm-agenda.txt>.
- [11] E. Gustafsson, A. Jonsson, E. Perkins, "Mobile IP Regional Registration", Internet Engineering Task Force draft-ietf-mobileip-reg-tunnel-04.txt, Mai 2001.
- [12] H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier, "Hierarchical MIPv6 mobility management", Internet Engineering Task Force draft-ietf-mobileip-hmipv6-03.txt, Février 2001.
- [13] C. Perkins, P. Calhoun, "AAA Registration Keys for Mobile IP", Internet Engineering Task Force draft-ietf-mobileip-aaa-key-04.txt, March 2001.
- [14] A. Campbell, J. Gomez, C-Y. Wan, S. Kim, Z. Turanyi, A. Valko, "Cellular IP", Internet Engineering Task Force draft-ietf-mobileip-cellularip-00.txt, Décembre 1999.
- [15] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility"
- [16] A. Campbell, J. Gomez, S. Kim, A. Valko, C-Y. Wan, Z. Turanyi, "Design, Implementation, and Evaluation of Cellular IP", IEEE Personal Communications, Août 2000.
- [17] Z. Shelby, D. Gatzounas, A. Campbell, C-Y. Wan, "Cellular IPv6", Internet Engineering Task Force draft-shelby-seamoby-cellularip6-00.txt, Novembre 2000.
- [18] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, L. Salgarelli, "IP Micro-mobility support using HAWAII", Internet Engineering Task Force draft-ietf-mobileip-hawaii-00, Juin 1999.
- [19] S. Tabbane, "Réseaux mobiles", réseaux et télécommunications, édition Hermes, 1997.
- [20] X. Lagrange, "Réseaux GSM", cinquième édition, édition Hermes, Septembre 2000.
- [21] F. Muratore, "UMTS Mobile Communications for the Future", Janvier 2001.
- [22] H. Kaaranen, S. Naghia, L. Laitinen, A. Ahtiainen, "UMTS Networks: Architecture, Mobility and Services", 2001.

Références bibliographiques

- [23] J.P. Castro, "The UMTS Network and Radio Access Technology: Air Interface Techniques for Future Mobile Systems", Avril 2001.
- [24] UMTS, <http://www.umts-forum.org>.
- [25] N. Montavont, T. Noel, "Survey of IP Handoffs in Wireless Network", article en cours de soumission, Juin 2001.
- [26] "Data processing – Open System Interconnexion – Basic Reference Model", ISO IS 7498, 1984.
- [27] D.C. Plummer, "An Ethernet Address Resolution Protocol, Internet Engineering Task Force Request For Comment 826, Novembre 1982.
- [28] Finlayson, Mann, Mogul, Theimer, "A Reverse Address Resolution Protocol", Internet Engineering Task Force Request For Comment 903, Juin 1984.
- [29] Université de Californie du Sud, "Internet Protocol", Internet Engineering Task Force Request For Comment 760, Janvier 1980.
- [30] Université de Californie du Sud, "Transmission Control Protocol", Internet Engineering Task Force Request For Comment 761, Janvier 1980.
- [31] J. Postel, "User Datagram Protocol", Internet Engineering Task Force Request For Comment 768, Août 1980.
- [32] R. Droms, "Dynamic Host Configuration Protocol", Internet Engineering Task Force Request For Comment 2131, Mars 1997.
- [33] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6", Internet Engineering Task Force Request For Comment 2461, Décembre 1998.
- [34] S. Thomson, T. Narten, "IPv6 Stateless Address Autoconfiguration", Internet Engineering Task Force Request for Comments 2462, Décembre 1998.
- [35] P. Mockapetris, "Domain Names -- Concepts and Facilities", Internet Engineering Task Force Request for Comment 1034, Novembre 1987.
- [36] P. Mockapetris, "Domain Names -- Implementation and Specification", Internet Engineering Task Force Request for Comment 1035, Novembre 1987.
- [37] P. Furguson, D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", Internet Engineering Task Force Request for Comment 2827, Mai 2000.
- [38] Norme IEEE 802.11 : IEEE Std 802.11-1999.
- [39] Norme IEEE 802.3 : IEEE Std 802.3ad-2000.
- [40] Norme HiperLan2 : ETSI TR 101 031 v2.2.1, Janvier 1999.
- [41] Norme Bluetooth : Specification of the Bluetooth System, v1.1, Février 2001.
- [42] RNRT, Réseau National de Recherche en Télécommunication, <http://www.telecom.gouv.fr/rnrt>
- [43] M. Gondran, M. Minoux, "Graphes et algorithmes", édition Eyrolles, 1979.
- [44] D. Comer, TCP/IP : "Architecture, Protocoles Applications", édition InterEditions, 1992.
- [45] D. Estrin, M. Handley, J. Heidemann, S. McCanne, Y. Xu, H. Yu, "Network Visualization with the VINT Network Animator NAM", Mars 1999.
- [46] Université de Californie du Sud ISI, Virtual InterNetwork Testbed (VINT): methods and system, 1996.
- [47] K. Calvert, E.W. Zegura, GT-ITM : Georgia Tech Internetwork Topologie Models, Technical report, Georgia Institute of Technology, College of computing, 1996.

REFERENCES INTERNET

IPv6

IP Version 6 (IPv6) : <http://playground.sun.com/ipv6/>
Recherche G6 : <http://www.urec.fr/IPng/G6.html>
Projet6 – réseau IPv6 sous linux : <http://project6.ferrara.linux.it/>
IPv6 : la nouvelle génération internet : <http://www.ipv6.org/>
Home page du 6bone : <http://www-cnr.lbl.gov/6bone/>
Projet Kame : implémentation de la pile IPv6 pour FreeBSD :
<http://www.kame.net/>
Présentation d'IPv6 : <http://www.data.com/issue/991021/ipv6.html>

Documentation IPv6

Aperçu d'IPv6 : <http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html>
Tout sur IPv6 : <http://www.networkmagazine.com/issue/991021/ipv6.html>
Linux et IPv6 : <http://www.bieringer.de/linux/>

Equipes de recherche

IETF Mobile IP : <http://www.ietf.org/html.charters/mobileip-charter.html>
Cellular IP à l'université de Columbia : <http://comet.ctr.columbia.edu/cellularip/>
Projet CMU Monarch : <http://www.monarch.cs.cmu.edu/>
micro-mobilité HAWAII IP (Lucent Bell Labs) : <http://www.bell-labs.com/user/ramjee/>
SUN : Mobile IP pour Solaris / Linux : <http://playground.sun.com/pub/mobile-ip/>
Groupe de recherche réseau du LBNL : <http://www-nrg.ee.lbl.gov/>
Groupe de recherche réseau de l'ULP Strasbourg : <http://www-r2.u-strasbg.fr/>
Groupe de recherche réseau de l'ENST : <http://www.enst-bretagne.fr>
At&t lab : <http://www.att.com>
ACIRI : <http://www.aciri.org>

Implémentations Mobile IP

LancasterIPv6 Mobilité (Linux) : <http://www.cs-ipv6.lancs.ac.uk/MobileIP/>
HUT Mobile IP (Linux) : <http://www.cs.hut.fi/Research/Dynamics/>
Mobile-IP par l'université de l'état du Portland (sous BSD) :
<http://www.cs.pdx.edu/research/SMN/index.html>
CMU Mobile-IP (sous BSD) : <http://www.monarch.cs.cmu.edu/software.html>
SUN Mobile-IP (sous Solaris) : <http://playground.sun.com/pub/mobile-ip/index.html>
Mobile-IP de l'université de Bucharest (NT) : <http://mip-nt.aii.pub.ro/>

Evènements dans la mobilité

Workshop de MANET :

<http://www.isr.umd.edu/Courses/Workshops/MANET/program.html>
IFIP Broadband Communications 99 : <http://www.ee.ust.hk/~bc99/>
Bibliothèque WWW : Mobile and Wireless Computing :
<http://gunpowder.stanford.edu/mobile/>

Tutoriels et résumé

Tutorial : Mobile IP : <http://www.computer.org/internet/v2n1/perkins.htm>
Résumé de Mobile IP standard :
<http://www.neda.com/mobileIpSurvey/html/mobileIP.html>
Intégration d'UMTS et de B-ISDN :
<http://www.ee.surrey.ac.uk/Personal/L.Wood/UMTS/>
Tutorial: Mobile IP : <http://computer.org/internet/v2n1/perkins.htm>
G6 IPv6 - Présentations Recherche Mobilité : <http://www.ipv6.u-strasbg.fr/G6Recherche/>

Sans fil

Communications sans fil et mobilité :
<http://http.cs.berkeley.edu/~randy/Courses/CS294.S96/CS294-7.S96.html>
IEEE P802.11, groupe de travail pour des LAN sans fil :
<http://grouper.ieee.org/groups/802/11/>
FranceNet - intégrateur Internet français : <http://www.fluxus.net/HomePage/>
IEEE P802.15 groupe de travail pour des aires de réseaux personnels sans fil :
<http://grouper.ieee.org/groups/802/15/>
IEEE EUI64 Standard : <http://grouper.ieee.org/groups/802/15/>
ETSI: TIPHON Homepage : <http://www.etsi.org/tiphon/tiphon.htm>
Qos dans IP: George Fankhauser : <http://www.tik.ee.ethz.ch/~gfa/>

Organisations

Programmes de R&D européen : <http://www.cordis.lu/en/home.html>
Commission européenne : <http://www.cordis.lu/en/home.html>
Home Page de l'IETF: <http://www.ietf.org/>
IPv6 Forum : <http://www.ipv6forum.com/>
W3C : <http://www.w3.org/>
Forum WAP : <http://www.wapforum.org/>
Homepage de 3GPP : <http://www.wapforum.org/>
Forum de l'internet sans fil mobile : <http://www.mwif.org/>
EUROCONTROL : <http://www.eurocontrol.fr/>
TAROT : <http://www.eurocontrol.fr/>
Renater : <http://www.renater.fr/>
IEE: Institution of Electrical Engineers : <http://www.iee.org.uk/>
AFNIC: Association Française pour le Nommage Internet en Coopération :
<http://g6.nic.fr/>
ETSI : <http://www.etsi.org/>
EIA: Electronic Industries Alliance : <http://www.eia.org/>
IMT-2000 : <http://www.itu.int/imt/>
IEEE Communications Society : <http://www.comsoc.org/index.html>

IEEE Standards : <http://standards.ieee.org/>
IEEE Computer Society : <http://www.computer.org/>
ITU : <http://www.itu.int/home/index.html>
SIGCOMM : <http://www.acm.org/sigcomm/>
ACM : <http://www.acm.org/>
ACTS Information Window : <http://www.uk.infowin.org/ACTS/>
CaberNet : <http://www.laas.research.ec.org/cabernet/>

Simulateur réseau

- **NS**

HomePage de NS : <http://www.isi.edu/nsnam/ns/index.html>
NAM: Network Animator : <http://www.isi.edu/nsnam/nam/index.html>
OTcl- MIT Object Tcl : <http://www.isi.edu/nsnam/otcl/index.html>
NS : Implementation d'OSPF : <http://networks.ecse.rpi.edu/~sunmin/rtProtoLS/>
WirelessNetwork Simulator (WiNS) : <http://http.cs.berkeley.edu/~gnguyen/ns/>
MetricomNetwork Simulation avec NS :
<http://http.cs.berkeley.edu/~rfromm/Courses/SP96/cs294-7/>
LBNL Network Simulator, ns version 1 : <http://www-nrg.ee.lbl.gov/ns/>

- **Topologie réseau**

Générateur de topologie pour des simulateurs réseau :
<http://www.cs.uoregon.edu/~zappala/topology/>
GT-ITM: modélisation de topologie inter réseau :
<http://www.cc.gatech.edu/projects/gtitm/>

- **Autres Simulateurs**

Simulation pour des mécanismes de routage : <http://www-nrg.ee.lbl.gov/collapse.html>
Simulateur réseau cnet (v1.4) : <http://www.cs.uwa.edu.au/pls/cnet/>
STCP: ATM et IP : <http://lrcwww.epfl.ch/~manthorp/stcp/>
Simulation de qualité de service IP : <http://bacon.gmu.edu/qosip/>
Software de simulation : <http://www.topology.org/soft/sim.html>
Simulateur REAL : <http://www.cs.cornell.edu/skeshav/real/overview.html>

Tcl/Tk

Programmation en Tcl et Tk : <http://www.beedub.com/book/2nd/>
Tcl/Tk Consortium : <http://www.tclconsortium.org/index.html>
Manuel Tcl8.0/Tk8.0 : <http://www.elf.org/tcltk-man-html/contents.htm>

C++

Stroustrup: The C++ Programming Language (Third Edition) :
<http://www.research.att.com/~bs/3rd.html>
The ISO/ANSI C++ Draft : <http://www.cygnus.com/misc/wp/>

Entreprise

Motorola : <http://www.motorola.fr>

Motorola Bluetooth : <http://www.mot.com/bluetooth/index.html>

Ericsson : <http://www.ericsson.de/>

France Telecom : <http://www.francetelecom.fr>

Cisco : <http://www.cisco.com>

Sun Microsystem : <http://www.sun.com>

GLOSSAIRE

Agent visité passerelle	Gateway FA (GFA)	Agent Visité qui a une adresse IP routable publique
Domaine	Domain	Collection de réseaux qui partagent une administration réseau commune
Agent Visité régional	Regional Foreign Agent	Agent Visité qui peut être la cible d'une demande de registration régionale
Clé de registration	Registration key	Clé utilisée par les nœuds mobiles et les agents de mobilité pour sécuriser certains messages de contrôle relatifs à MIP
Registration régionale	Regional registration	Un nœud mobile fait une registration locale dans un domaine visité en envoyant un Regional Registration Request au GFA et reçoit un Regional Registration Reply en retour
Registration-mère	Home registration	Registration, exécutée par l'agent mère et l'agent visité passerelle utilisé les spécifications du RFC 2002 (MIPv4)
Handoff soft	Soft handoff	Processus par lequel un terminal mobile CDMA est transféré entre une ou un ensemble de point d'accès à un(e) autre dans un réseau d'accès. Un handoff soft est en général très rapide (de l'ordre de 20ms) et à une faible probabilité de réduire la connexion temps réel
Handoff hard	Hard handoff	Processus par lequel un terminal mobile est transféré entre un fournisseur de service cellulaire et un autre, ou entre deux routeurs d'accès qui ne partagent pas directement une connexion de fournisseur de réseau. Un handoff hard a une grande probabilité de réduction de connexion et est assez lent (100ms ou plus)
Interface RP	RP interface	Interface entre un nœud d'une réseau radio et un nœud servant des paquets de données
Nœud servant des paquets de données (PDSN)	Packet Data Serving Node	Nœud responsable de l'établissement, du maintien et de la terminaison de la couche liaison au nœud mobile
Nœud d'un réseau radio (RNN)	Radio Network Node	Nœud responsable du relais du protocole de la couche liaison entre le nœud mobile et le PDSN correspondant
Tunnel GRE	GRE tunnel	Norme de tunnel (voir « sun's SKIP Firewall Traversal for Mobile IP », RFC 2356, 1998)
Data-connected	Data-connected	Connexion orientée données

Agent de mobilité	Mobility agent	Point d'accès sans fil au réseau
Anchor Agent Visité	Anchor Foreign Agent	FA avec une adresse IP publique qui joue le rôle de point d'ancrage quand un mobile se déplace à un nouvel agent visité. Jusqu'à ce que la registration soit complète, l'Anchor FA supporte une registration locale quand le nœud mobile change son point d'attache à l'agent visité voisin
Roaming	Roaming	Terme basé sur les opérateurs qui implique les accords entre différents opérateurs pour permettre à mobile d'avoir une connexion dans le réseau étranger
Handover ou handoff	Handover or handoff	Processus enclenché quand un mobile actif change son point d'attache au réseau sans qu'une reconfiguration de la part de l'utilisateur n'ait lieu
Handoff de la couche 2	Layer 2 handoff	Opération effectuée par un mobile qui change de point d'accès sans changer de routeur d'accès. Ce type de handoff est transparent au routage de la couche IP
Handoff entre routeur d'accès	Intra-AR handover	Handoff qui change l'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	Intra-AN handover	Opération effectuée quand le 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. L'adresse du mobile ne change toujours pas, mais le chemin pour l'atteindre est modifié
Handoff entre réseau d'accès	Inter-AN handover	Déplacement du mobile hors du réseau d'accès ⇒ obtention d'une nouvelle adresse IP pour le mobile
Handoff intra-technologie	Intra-technology handover	Handoff entre des équipements de même technologies
Handoff inter-technologie	Inter-technology handover	Handoff entre équipements de technologies différentes
Handoff horizontal	Horizontal handover	Handoff réalisé par le mobile sur une même interface c'est-à-dire que le mobile communique avec le réseau d'accès avec la même interface avant et après le handoff (généralement c'est un handoff intra-technologie)
Handoff vertical	Vertical handover	Handoff réalisé par le mobile entre deux de ses interfaces (généralement handoff inter-technologie)
Handoff proactif	Proactive handover	= fast handoff ?
Mode veille	Dormant mode	Etat du mobile dans lequel il restreint sa capacité à recevoir un trafic IP normal en réduisant le monitoring du canal radio. Cet état permet au mobile d'économiser son énergie et d'alléger la charge de signalisation sur le réseau
Mode veille de semi-écoute	Time-slotted dormant mode	Implémentation du mode de veille dans lequel le mobile alterne des périodes d'écoute et de non-écoute du trafic. Cette implémentation est généralement synchronisée avec le réseau

Pagination	Paging	Signalisation utilisée pour localiser un mobile en mode veille et établir la connexion du dernier saut lorsqu'un trafic arrive pour le mobile. Un mobile peut entrer dans ce mode quand il ne génère pas de trafic IP pour ses données (aucunes données à émettre et aucunes données à recevoir). Le protocole de localisation est dépendant du type de liaison radio utilisé. OU : procédure pour mettre un mobile idle dans un mode actif
Aire de pagination	Paging area	Ensemble de points d'accès radio qui effectuent une signalisation pour localiser le mobile. Le mobile n'informe de sa localisation que quand il change d'aire. Une aire de pagination ne correspond pas forcément un à sous-réseau
Canal de pagination	Paging channel	Canal radio dédié à la signalisation des mobiles en mode veille dans le but du pagination. Actuellement, le protocole utilisé est dépendant de la technologie radio
Canal du trafic	Channel traffic	Canal utilisé pour le trafic IP lié à un mobile actif. Pour certains protocoles radio, c'est le seul canal disponible
Mise à jour de localisation	Location updating	Procédure réalisée par le mobile par laquelle il informe le réseau d'accès qu'il s'est déplacé dans une nouvelle aire de pagination
Registrations d'aire de pagination	Paging area registrations	Signalisation d'un nœud mobile en mode veille au réseau quand le nœud mobile traverse une aire de pagination pour informer sa présence dans la nouvelle aire
Diff-serv	Diff-serv	Procure des services à un mobile inscrit et un traitement différents des paquets dans les routeurs. Diff-serv place la responsabilité sur le réseau qui gère la politique, les comptes...
Int-serv	Int-serv	Utilise la signalisation RSVP pour installer des détails sur le chemin de routage d'un trafic à destination d'un utilisateur
Handoff contrôlé par le mobile	Mobile-controlled handover	C'est le mobile qui a le plus important contrôle sur le processus de handoff
Handoff contrôlé par le réseau	Network-controlled handover	C'est le réseau qui a le plus important contrôle sur le processus de handoff
Handoff assisté par le mobile	Mobile –assisted handover	Les informations et les mesures effectuées au mobile sont utilisées pour décider de l'exécution du handoff
Handoff assisté par le réseau	Network-assisted handover	Handoff pour lequel le réseau d'accès collecte des informations qui peuvent être utilisées dans la décision de handoff
Handoff backward	Backward handover	Handoff initialisé par l'ancien router d'accès, ou alors par le mobile à travers l'ancien routeur d'accès

Handoff forward	Forward handover	Handoff initialisé par le nouveau routeur d'accès ou par le mobile à travers le nouveau routeur d'accès
Handoff planifié	Planned handover	Handoff proactif (prévu) dans lequel une partie de la signalisation peut être faite avant la connexion du mobile au nouveau routeur d'accès (construction d'un tunnel entre l'ancien et le nouveau routeur d'accès)
Handoff non planifié	Unplanned handover	Handoff réactif (non prévu) où aucune signalisation n'est faite avant que le mobile ne se soit déplacé de l'ancien au nouveau routeur
Handoff initialisé par le réseau	Network-initiated handover	Le réseau est le seul qui prend la décision que le mobile doit faire un handoff en gérant une liste de routeurs d'accès candidats
Handoff initialisé par le mobile	Mobile-initiated handover	Le mobile est le seul qui prend la première décision de faire un handoff - Décision de handoff prise sur réception d'un Neighbor Advertisement ou sur des informations de plus bas niveau comme le SIR
Handoff make-before-break	Make-before-break handover	Pendant un handoff make-before-break, le mobile peut communiquer simultanément avec l'ancien et le nouveau routeur d'accès
Handoff break-before-make	Break-before-make handover	Durant un handoff break-before-make, le mobile ne peut pas communiquer simultanément avec les deux routeurs d'accès
Smooth handoff	Smooth handoff	Handoff qui a pour but principal de minimiser la perte de paquets, sans condition sur le délai de propagation des paquets
Fast handoff	Fast handoff	Handoff qui a pour but principal de minimiser les délais, sans conditions sur le nombre de paquets perdus
Seamless handoff	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 \Rightarrow la définition pratique est que d'autres protocoles, applications ou utilisateurs ne remarquent pas ces dégradations
Latence du handoff	Handover latency	Le temps de latence du handoff est la différence 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 au nouveau routeur d'accès \Rightarrow c'est le temps pendant lequel il ne peut pas recevoir et émettre un trafic IP

Micro diversité (ref L1/L2)	Micro diversity	Terme employé pour le cas où par exemple deux antennes sur le même transmetteur émettraient le même signal prenant deux chemins différents pour parer aux pertes
Macro diversité (ref L1/L2)	Macro diversity	Cette diversité a lieu quand la duplication est mise en place sur plusieurs points d'accès pas forcément sur le même routeur d'accès. Ceci procure un support pour la couche réseau pour déplacer des frames radio entre points d'accès et un point central combiné
Diversité IP (ref L3)	IP diversity	Découpage et recombinaison des paquets à la couche IP
Mode actif	Active mode	Un mobile est en mode actif quand le réseau d'accès connaît le routeur d'accès du mobile et quand le mobile émet et reçoit un trafic IP. Le lien d'accès peut ne pas être actif, mais la couche radio peut établir une liaison sans utiliser la couche réseau. Le mobile a une adresse IP assignée
Mode idle	Idle mode	Mode dans lequel le réseau d'accès connaît l'aire de pagination du mobile, mais où le mobile n'a pas de lien d'accès et donc les paquets ne peuvent pas être délivrés au mobile sans une pagination du réseau d'accès
Etat détaché	Detached state	Etat dans lequel le mobile n'est ni en mode actif, ni en mode idle. Le nœud mobile n'a pas d'adresse IP dans le réseau d'accès
Mobilité utilisateur	User mobility	Réfère à la capacité d'un utilisateur d'accéder à des services à partir de différents host physiques
Mobilité personnelle	Personal mobility	Complète la mobilité utilisateur avec la capacité de suivre les positions de l'utilisateur et fournir l'emplacement actuel d'utilisateurs pour permettre à des sessions d'être lancées par l'utilisateur par n'importe qui sur n'importe quel autre réseau
Mobilité d'host	Host mobility	Réfère aux fonctions pour permettre au host mobile de changer leur point d'attache au réseau sans interrompre les communications de ce host
Macro mobilité	Macro mobility	Réfère à une mobilité dans une vaste aire. Ceci inclut le support de la mobilité et les procédures de registration d'adresse IP lorsque le host mobile se déplace entre différents domaines IP
Micro mobilité	Micro mobility	Réfère à une mobilité dans une petite aire. Habituellement c'est une mobilité dans un domaine IP
Adresse temporaire	Care-of Address	adresse temporaire du mobile attribuée dans le réseau visité
Agent mère	Home Agent	routeur Ipv4 avec une interface sur le même lien que le mobile (dans le réseau mère)

Ancien routeur d'accès	Old access router	Routeur d'accès qui offrait une connectivité au host mobile avant un handoff
Cellule radio	Radio Cell	Aire géographique dans laquelle un point d'accès procure une couverture radio
Domaine administratif	Administrative Domain	Collection de réseaux sous le même control administratif et regroupé ensemble pour des buts administratifs
passerelle d'accès réseau	Access Network Gateway	Routeur de réseau d'accès qui sépare un réseau d'accès d'autres réseaux IP. Un routeur d'accès et une passerelle de réseau d'accès peuvent être situé physiquement sur le même nœud
Hôte mobile	Mobile Host	Nœud mobile qui est un host terminal
Lien d'accès	Access Link	Lien du dernier saut entre un hôte mobile et un routeur d'accès. C'est le médium à travers lequel un point d'accès et un équipement de la couche 2 attaché au mobile peuvent communiquer au niveau de la couche de liaison
Micro flot	Microflow	Unité fondamentale d'un service IP = plus petite partie d'un trafic d'une mobile ayant un contexte distinct
Nœud mobile	Mobile Node (MN)	Nœud Ipv4 qui peut changer de points d'attachement sur l'Internet tout en maintenant les communications en cours (et en utilisant uniquement son adresse principale). Un MN peut avoir des fonctionnalités de routage
Nouveau routeur d'accès	New access router (nAR)	Routeur d'accès qui offrait une connectivité au host mobile après un handoff
Point d'accès	Access Point (AP)	Équipement de niveau 2 connecté à un ou plusieurs routeurs d'accès et qui offre une connexion sans fil au host mobile. Quelques fois on utilise les termes de point d'accès ou point d'accès transceiver. Il peut être une partie d'un routeur ou sur un équipement séparé
Réseau d'accès	Access Network (AN)	Réseau IP qui inclue au moins un routeur de réseau d'accès
Routeur d'accès	Access Router (AR)	Un routeur de réseau d'accès est situé sur une branche d'un réseau d'accès et est connecté à un ou plusieurs points d'accès. Un routeur d'accès offre une connexion IP aux host mobiles, fonctionnant comme un routeur par défaut pour le(s) mobile(s) qu'il dessert
Routeur d'accès candidat	Candidate Access Router	Routeur d'accès candidat sur lequel le host mobile pourrait se déplacer. Un schéma de handoff peut supporter plusieurs candidats

Routeur d'accès de service	Serving Access Router	Routeur d'accès offrant actuellement une connexion au host mobile. C'est habituellement le point de départ d'un host mobile quand il se dirige vers un nouveau routeur d'accès. Un mobile peut avoir plusieurs routeurs d'accès de service à un moment donné
Routeur de réseau d'accès	Access Network Router (ANR)	Routeur IP dans un réseau d'accès. Peut inclure des fonctionnalités spécifiques à un réseau d'accès comme la mobilité, QoS...
Seamless handoff	Seamless handoff	Un seamless handoff est théoriquement un handoff pour lequel il n'y a aucun changement dans la capacité de service, sécurité ou qualité. Pratiquement, c'est un handoff pour lequel un utilisateur ne remarque aucun changement
Transfert de contexte	Context transfer	Mécanisme pour établir les conditions suffisantes à un ou plusieurs routeurs d'accès pour entièrement supporter le microflot d'un mobile. Après ce transfert, un routeur d'accès est capable de gérer le trafic d'un mobile sans interruption