Gestion de la mobilité dans les réseaux ambiants

Laurentiu Sorin Paun

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

No attribué par la bibliothèque

T H È S E

pour obtenir le grade de

DOCTEUR DE L'INPG

Specialité : « Informatique : Systèmes et Communications »

préparée au laboratoire LSR – IMAG
dans le cadre de l'École Doctorale
« Mathématiques, Sciences et Technologies de l'Information »

présentée et soutenue publiquement par

Laurentiu Sorin PAUN

Le 22 Novembre 2005

Gestion de la mobilité dans les réseaux ambiants

Directeur de thèse : M. Andrzej DUDA

Co-encadrant : M. Franck ROUSSEAU

JURY :

M. Jacques CHASSIN DE KERGOMMEAUX,	Président
M. Christian BONNET,	Rapporteur
M. Thomas NOEL,	Rapporteur
M. Andrzej DUDA,	Directeur de thèse
M. Franck ROUSSEAU,	Co-encadrant

À mes parents.

Je tiens tout d'abord à remercier Andrzej Duda, Professeur à l'Institut National Polytechnique de Grenoble, mon directeur de thèse, pour son encadrement, ses nombreux conseils et son soutien. Son influence sur mon parcours post-universitaire a été décisive : il fut à l'origine de ma découverte du monde de la recherche et du monde des réseaux sans-fil. J'adresse aussi mes remerciements à Franck Rousseau, Maître de Conférence à l'Institut National Polytechnique de Grenoble, mon co-directeur de thèse, pour toute sa coopération et toutes les discussions qui ont beaucoup contribué à l'avancement de mon travail de thèse.

Mes remerciements s'adressent aussi à Jacques Chassin de Kergommeaux, Professeur à l'Institut National Polytechnique de Grenoble, pour m'avoir fait l'honneur de présider ce jury. Je remercie également Christian Bonnet, Professeur à l'Institut Eurecom Sophia-Antipolis et Thomas Noel, Maître de Conférence à l'Université Louis Pasteur de Strasbourg, d'avoir accepté le rôle de rapporteur de cette thèse et évalué mon travail de manière approfondie et constructive.

Au delà du jury, je tiens à remercier tous ceux qui ont permis à ces travaux d'aboutir, par leurs conseils, leurs contributions et leurs encouragements :

Merci à tous les enseignants-chercheurs de l'équipe Drakkar (Gilles, Martin, Pascal, Jean-Luc et Dominique) ainsi qu'aux doctorants de l'équipe et du laboratoire pour les échanges professionnels et amicaux que nous avons eus au cours des dernières années. Je mentionne particulièrement mes collégues de bureau Justi, Mhamed, Oualid, Pawel et Elena, ainsi que Paul, Cristina, Hoang, Binh, Vincent et Yann avec qui j'ai travaillé de près au cours de ma thèse. Merci aussi aux autres personnes très symphatiques du laboratoire que j'ai eu l'occasion de côtoyer aux cours de ces dernières années de laboratoire : enseignants-chercheurs, doctorants, post-doctorants, stagiaires et personnel administratif, pour leur gentillesse et pour les bons moments que nous avons passés ensemble.

Je voudrais remercier aux enseignants et enseignants-chercheurs avec qui j'ai pu gagner mes premières expériences pédagogiques pendant ma thèse : Pascal Sicard, Alain Cartade, Franck, Andrzej et Sébastien Viardot . Leur professionnalisme et leurs remarques précieuses ont beaucoup apporté à mon travail pédagogique.

Merci à mes enseignants de Polytechnique Bucarest et de l'UFRIMA de l'UJF Grenoble qui ont su, par leurs qualités pédagogiques, techniques et humaines, attirer mon intérêt pour l'informatique. J'exprime, en particulier, mon profond respect à Nicolae Tapus et Radu Chisleag de l'Université Polytechnique de Bucarest qui m'ont soutenu et encouragé pour continuer mes études à Grenoble. Merci à Jean-Claude Fernandez qui a conduit mes pas vers le laboratoire LSR-IMAG et encore une fois à Andrzej et Franck qui m'ont accueilli dans l'équipe Drakkar.

Enfin, je garde une place toute particulière pour ma famille et mes amis proches, pour tout ce qu'ils ont pu (et continuent à) m'apporter tout au long de ces années.

Sorin

Table des matières

Table des figures

Liste des tableaux

1. Introduction

Le réseau Internet connaît aujourd'hui un succès extraordinaire. Conçu il y a 30 ans, ses principes ont remarquablement résisté aux changements des technologies et de son utilisation dans le temps. En effet, pendant cette période, le monde de technologies informatiques et de télécommunications n'a cessé de changer. Nous dressons un bilan du contexte actuel :

- On observe une présence accrue, en nombre et pourcentage, d'ordinateurs portables. Dotés d'une richesse multimédia et d'une inter-connectivité étendue, ils sont suffisamment performants pour concourir les machines fixes. En plus des ordinateurs portables classiques, un nouveau genre d'équipements est apparu - *dispositifs intelligents* ou *dispositifs enfouis*. Équipés de microprocesseurs et mémoire, ils sont capables de produire, stocker, manipuler et échanger l'information. Aujourd'hui, la variété de ces dispositifs est impressionnante : ded tablettes PC, assistants personnelles et téléphones portables jusqu'aux cartes à puce, capteurs ou actionneurs.

- Les techniques de communications sans fil ont évolués. La largeur de la bande passante et le faible prix d'accès a permis le développement et le déploiement des nouvelles technologies sans fil. Ainsi, on a accès au GPRS[1] et à l'UMTS[2] dans les réseaux de télécommunications, aux réseaux locaux sans-fil IEEE 802.11[3][1] et aux réseaux personnels de dispositifs enfouis utilisant Bluetooth[5]. Toutes ces nouvelles technologies, complétées par les communications traditionnelles filaires, nous permettent d'être sous la couverture continue de plusieurs types de réseaux.

- L'Internet a connu une croissance record[6, 7], mesurée en nombre de nœuds connectés et aussi en terme de popularité - utilisateurs et services proposés. Nous considérons que le développement futur de l'Internet continuera d'être alimenté principalement par les trois aspects mentionnés : les ordinateurs portables, les dispositifs intelligents et les réseaux sans fil.

Ces tendances nous conduisent à penser que le monde de l'informatique et des communications se dirige vers ce qu'on appelle *l'informatique ubiquitaire*. Ce terme a été introduit par Mark Weiser dans [8] pour désigner un environnement où les ordinateurs sont omniprésents[2]. Si on transfère cette vision de l'informatique dans le monde des communications, on obtient l'inter-connectivité partout, tous le temps, avec tout le monde. Nous partageons l'avis de Jim Waldo, qui considère dans [9] que les réseaux futurs seront caractérisés par

[1] Ce type de réseau est connu aussi sous le nom Wi-Fi (*Wireless Fidelity*)[4], d'après le nom de la certification d'interopérabilité pour les produits qui implémentent le standard IEEE 802.11.

[2] Ce qui signifie *existant partout*.

deux propriétés principales. En plus de l'omniprésence (par défaut, chaque dispositif sera connecté à un réseau), la deuxième propriété est que les réseaux seront invisibles. En étant toujours présents, nous nous apercevons leur existence (ou plutôt leur inexistence) seulement dans les rares situations où ils sont indisponibles. En partie, l'omniprésence et l'invisibilité de réseaux seront causées par le sans-fil, au moins dans la bordure de l'Internet, pour l'accès des terminaux au réseau. Faire part d'un réseau ne nécessitera pas d'être relié d'une façon visible, par un câble, mais seulement d'être près de lui.

Dans ce monde d'ordinateurs et réseaux ubiquitaires, une question qui se pose est l'interaction de l'utilisateur avec son environnement. Nous pensons qu'un nouveau mode d'interaction sera privilégie. Celui-ci, appelé *pervasive computing*[10] par David Tennenhouse, redéfinit le rôle de la présence humaine, qui n'est plus principalement d'agir et interagir directement avec l'environnement, mais plutôt de superviser son déroulement automatique. Dans l'environnement ubiquitaire des réseaux entourant l'utilisateur, cela demande que la connexion des terminaux au réseau, ainsi que le transfert vers d'autres réseaux doivent s'exécuter automatiquement, sans que l'utilisateur l'aperçoive.

1.1. Problématique et motivations

Leonard Kleinrock[3], connu comme l'inventeur de l'Internet[12], présentait il y a 35 ans sa vision sur ce que ce réseau devrait devenir [13]. En grandes lignes, sa vision était que l'Internet sera omniprésent, toujours disponible, toujours actif, et que chacun pourra connecter son terminal à l'Internet d'une manière transparente, comme dans une prise d'électricité. Si ses trois premières prévisions s'avèrent plus ou moins déjà réalisées, c'est la dernière qui reste encore inachevée.

La principale difficulté vient du fait que les protocoles de l'Internet ont étés conçus sans prendre en considération que les utilisateurs et leurs machines peuvent changer leur point d'attachement dans les réseaux. En effet, à l'époque où les protocoles Internet se développaient, et jusqu'au milieu des années 90, la plupart des utilisateurs utilisaient l'Internet à partir des ordinateurs fixes de leurs institutions. Aujourd'hui, la plupart des utilisateurs sont devenus nomades. La prolifération des réseaux sans-fil favorise le nomadisme, car ils permettent d'être connecté même pendant les déplacements d'un point à un autre. Le nombre croissant des terminaux et réseaux sans fil superposés permet également de changer de terminal pour avoir plus de fonctionnalités ou de choisir entre plusieurs réseaux disponibles pour avoir une meilleure qualité de service.

Le point important dans la vision présentée plus haut est que l'utilisateur s'attend à un fonctionnement transparent de ses applications et services réseau, en dépit de sa mobilité. Cependant, dans la plupart des cas, les terminaux et les applications ne peuvent pas continuer à fonctionner sans des opérations supplémentaires comme la reconfiguration des adresses IP ou autres paramétrages réseau. Cette reconfiguration provoque l'interruption

[3] Il a introduit la théorie de la communication par échange de paquets[11], qui est à la base des réseaux informatiques d'aujourd'hui. Il a écrit les premières spécifications de l'ARPANET et a supervisé sa mise en place et l'échange du premier paquet de données.

des connexions actives et nécessite le redémarrage de certaines applications. Sans être liés toujours à la mobilité, des interruptions et déconnexions de réseau peuvent intervenir aussi, à cause de l'indisponibilité temporaire de la connexion ou de l'arrêt volontaire du terminal.

1.2. Cadre et objectifs de la thèse

Le travail de cette thèse s'est déroulé au sein de l'équipe Drakkar, qui mène des activités de recherche sur différents aspects de protocoles réseaux et multimédia, avec un thème central lié aux réseaux mobiles sans fil. Elle fait partie du laboratoire LSR (Logiciels Systèmes Réseaux) et de l'Institut IMAG (Informatique et Mathématiques Appliquées de Grenoble). Plus précisément, mon travail s'inscrit dans un des projets-clé de l'équipe, labellisé « Réseaux ambiants : vers un espace de communication ubiquitaire », qui propose un espace de communication entourant l'utilisateur. Le concept de réseau ambiant repose sur plusieurs mécanismes de bas niveau comme la qualité de service, la mobilité, la découverte de service, et la facilité de configuration.

Plus spécifiquement, cette thèse se propose d'offrir des services pour la mobilité des machines lors des déplacements des utilisateurs. Nous pensons que la mobilité est une fonction essentielle des réseaux ambiants et doit être intégrée au cœur des protocoles réseau. Un bon nombre d'extensions aux protocoles Internet ont été conçues pour gérer d'une manière spécifique les contraintes liées à la mobilité des hôtes. Cependant, ces propositions n'ont pas réussi à remplacer le cœur de la pile TCP/IP, en partie parce qu'ils se sont avérés inefficients, en diminuant significativement les performances en terme de latence et bande passante des connexions.

On peut agir à différents niveaux dans la pile des protocoles, mais d'une manière générale nous avons privilégié les solutions au niveau IP, étant donné que c'est autour de ce protocole de base que l'Internet est construit. En plus de sa version actuelle IPv4, nous nous sommes appuyés aussi sur la future version IPv6, qui permet de développer des mécanismes de mobilité d'une manière plus aisée. En traitant la mobilité des hôtes au niveau IP, notre but a été de faire en sorte que les changements de réseau soient transparents pour les applications et pour l'utilisateur.

Cette transparence dépend en partie du temps pris pour le changements de réseau. Dans 802.11, la technologie d'accès sans-fil la plus répandue aujourd'hui, ce temps est de quelques centaines de millisecondes, pouvant même dépasser une seconde. Un de nos buts a été de réduire ce temps ; après une analyse des spécifications du standard et de ses mécanismes utilisé dans le transfert d'une entre deux point d'accès sans-fil, nous proposons une extension qui permet, dans certaines conditions, de réduire le délai à seulement quelque millisecondes.

Mais le changement de réseau est plus qu'une affaire de changement physique du point d'accès. Le plus souvent le déplacement d'une hôte d'un réseau à un autre demande la reconfiguration des plusieurs paramètres au niveau IP, parmi lequel le plus important est le changement de l'adresse IP. Les protocoles d'auto-configuration comme DHCP en IPv4 et les mécanismes correspondants de l'IPv6 ont le mérite de ne pas impliquer l'utilisateur,

mais en revanche le temps pris par la reconfiguration est d'environ une seconde. En plus, la reconfiguration de l'adresse IP ne résout pas tout, car ce changement même d'adresse IP pose un problème aux protocoles de transport. Conçu dans un temps où les hôtes de l'Internet n'étaient pas mobiles, le protocole TCP ne permet pas le changement de l'adresse IP à la volée, en cours d'une connexion ; comme résultat, il traduit cela dans une erreur, envoyée aux application ou à l'utilisateur.

Dans Mobile IP, la solution « officielle » de l'IETF, la réponse à cette contradiction est de permettre aux hôtes mobiles de garder la même adresse IP, indifféremment du réseau où ils se trouvent. Ceci est fait avec le prix d'indirections et inefficiences dans l'acheminement de paquets dans l'Internet. Notre approche a été de privilégier un routage optimal des datagrammes dans l'Internet. Nous considérons qu'une solution unique pour les problèmes de la mobilité, comme Mobile IP se veut, est difficile à concevoir ; à la place, nous proposons deux types de solutions : une pour la mobilité locale des hôtes, à l'intérieur d'un seul domaine, et une pour la mobilité globale, à travers l'Internet.

Dans la mobilité locale, le transfert physique des hôtes d'un réseau à l'autre peut se faire très rapidement. Pour ne pas rajouter un délai supplémentaire avec la reconfiguration des adresses IP, l'approche que nous avons choisi est celle de Mobile IP : permettre aux hôtes de garder inchangées leurs adresses. En échange, nous considérons qu'on ne doit se baser que sur l'infrastructure IP du domaine local pour assurer un rouage optimal des datagrammes, sans aucune interaction avec d'autres éléments à l'extérieur du domaine.

Au contraire, pour la mobilité globale, le routage optimal à travers l'Internet nous oblige de concevoir une solution dans laquelle les hôtes mobiles changent leurs adresses IP, en conformité avec le réseau ou ils s'y trouvent. Nous privilégions pour cette situation une solution basé sur le principe général « de bout en bout », dans laquelle les deux hôtes impliquées (et seulement eux) dans une connexion participent au transfert de cette connexion au nouveau point d'attachement de l'hôte mobile.

1.3. Organisation du manuscrit

L'organisation du manuscrit est représenté schématiquement dans la figure 1.1. Le rôle de cette introduction a été de donner une vision d'ensemble du contexte actuel de l'Internet, tout en soulignant la problématique de la mobilité d'hôtes, les motivations qui nous ont emmené à concevoir des nouvelles solutions ainsi que les objectifs de ces solutions.

Le chapitre 2 présente les types de mobilité dans les réseaux. et les fonctionnalités requises par la mobilité. Dans le chapitre 3 nous présentons les protocoles Internet ainsi que les implications de la mobilité vis-à-vis de ceux-ci.

Les deux chapitres suivants font l'état de l'art des solutions proposées pour gérer la mobilité. Nous présentons dans le chapitre 4 les propositions qui opèrent au niveau IP. Ensuite, dans le chapitre 5 on exemplifie comment la mobilité peut être gérée aux couches supérieures – transport et session – et comment certaines applications ont été conçues pour gérer elles-mêmes la mobilité. Une comparaison de toutes les solutions présentées et un

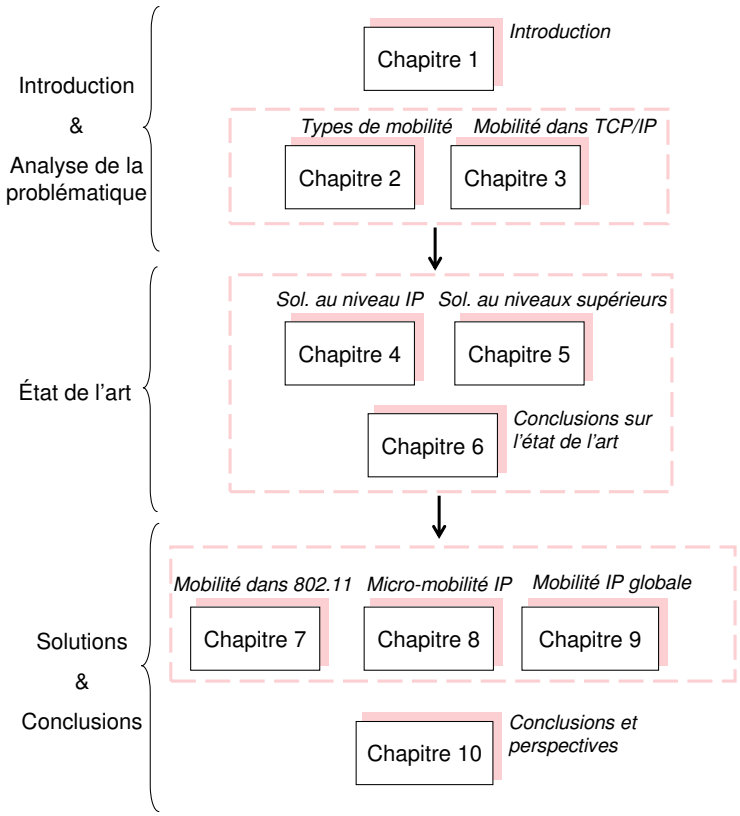


FIG. 1.1.: Organisation du manuscrit

tableau récapitulatif sont présentés dans le chapitre 6.

Notre analyse des réseaux sans fil de type IEEE 802.11 et de son mécanisme de transfert d'un point d'accès à un autre est présenté dans le chapitre 7. Notre extension de transfert direct ainsi que son application pour distribuer la charge du réseau sans-fil sont présentés en fin du chapitre.

Dans le chapitre 8, nous présentons un protocole de micro-mobilité qui utilise des routes d'hôte dynamiques à l'intérieur d'un domaine local. Il intègre les mécanismes de mobilité dans les réseaux 802.11 présentés auparavant, dans le but de réduire au maximum le délai du handoff. À la fin du chapitre, nous présentons un prototype développé pour valider notre protocole et mesurer ses performances.

Le chapitre 9 s'attaque au cas de la mobilité globale. Nous essayons de comprendre les implications de ce type de mobilité et proposons une solution simple, basés sur des adresses IP virtuelles et sur la translation locale d'adresses, qui masque aux couches supérieures le changement d'adresse IP.

Finalement, un chapitre de conclusions finira ce mémoire, en dressant un bilan de la thèse et en présentant quelques perspectives d'études futures.

Première partie .

Analyse de la mobilité dans les réseaux

2. Mobilité dans les réseaux

Dans le français le terme *mobilité* est défini comme le caractère, la capacité ou la facilité d'un objet ou d'une personne à être déplacé ou de se déplacer par rapport à un lieu, position ou ensemble d'objets de même nature. L'action de changer de position et le résultat de cette action sont appelés *mouvement* ou *déplacement*. Dans le domaine de réseaux, la *mobilité* se traduit par la possibilité que certaines *entités* peuvent être déplacées entre des *points d'attachement* différents. Nous énumérons quelques exemples, illustrés dans la figure 2.1 : :

- (1) Un terminal est physiquement déplacé à un autre endroit et reconnecté à l'Internet par le biais d'un nouveau réseau ;
- (2) Un utilisateur décide d'utiliser un nouveau terminal ;
- (3) Un terminal connecté simultanément à plusieurs réseaux change l'interface active ;
- (4) Parallèlement au déplacement de l'utilisateur, des données personnelles et applications portables sont migrées sur un autre terminal.

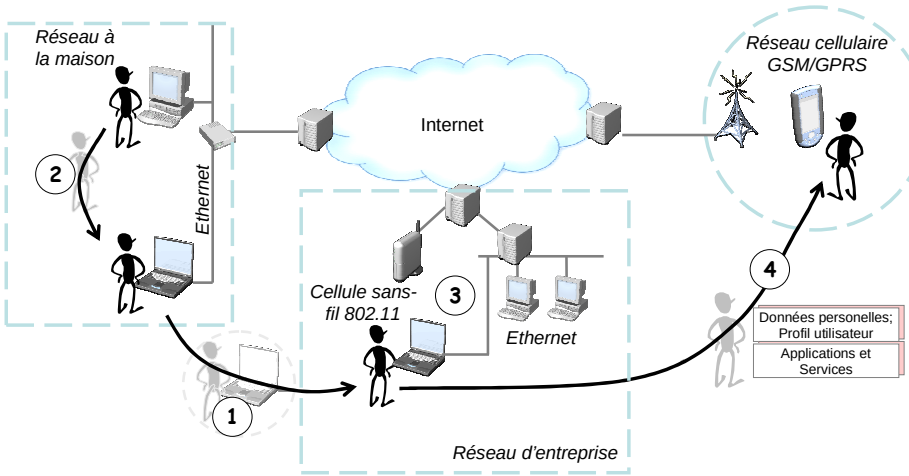


FIG. 2.1.: Types de mobilité

Dans ces situations, l'identité de l'entité mobile ne change pas, mais son support d'attachement au réseau change. Les correspondants qui sont en train de communiquer avec l'entité mobile le font en envoyant des paquets adressés à un point d'attachement bien précis. C'est

pour cela qu'un déplacement est si problématique.

2.1. Types de mobilité

On distingue plusieurs types de mobilité en fonction des entités qui sont impliquées. Généralement, on définit une connexion réseau comme une liaison établie par deux entités qui se trouvent aux deux bouts de la connexion et qui s'envoient des données. En fonction du niveau d'abstraction, ces entités peuvent désigner les machines, les applications ou même les utilisateurs. Dans ce qui suit, nous présentons les caractéristiques de ces différents types de mobilité.

2.1.1. Mobilité de terminaux

On commence par discuter du premier cas, *la mobilité de terminaux*, qui est aussi celui rencontré le plus fréquemment. En fonction de la portée et de la durée du déplacement, on distingue deux sous-catégories de la mobilité des terminaux : *la portabilité* et *la mobilité continue* (voir la figure 2.2).

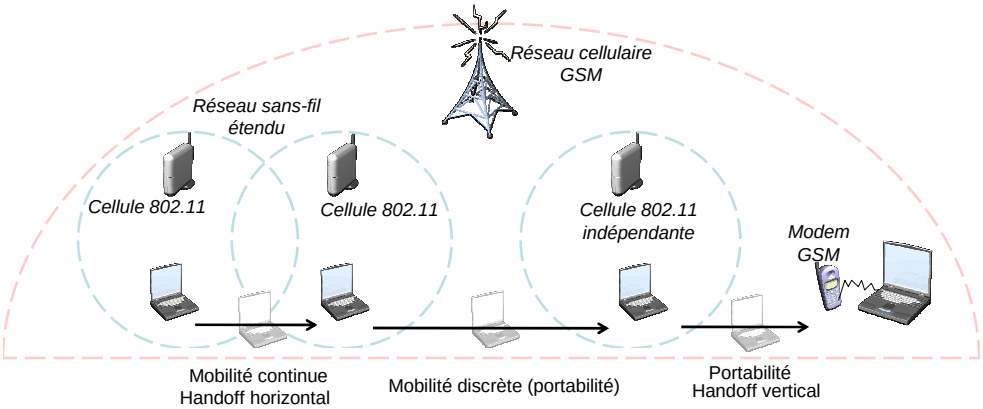


FIG. 2.2.: Mobilité de terminaux

Portabilité

Dans le premier cas, *la portabilité*, un terminal est déplacé entre deux points d'attachement au réseau distants. En profitant de la présence accrue des terminaux portables, on a pris l'habitude de les emporter et de les connecter au réseau dans différents endroits, profitant des îlots de couverture sans-fil dans les campus universitaires, les hôtels ou les gares. Cependant, ceci n'est pas un modèle de mobilité propre aux réseaux sans-fil, puisque c'est identique à la connexion de l'ordinateur portable par un réseau filaire. En même temps,

être connecté par sans fil ne signifie pas nécessairement mobile, car des machines fixes peuvent être connectés par sans-fil pour éviter le câblage (par exemple dans des bâtiments historiques).

La portabilité relève plutôt du nomadisme et sa principale caractéristique (et en même temps contrainte) est que le terminal n'utilise pas le réseau pendant qu'il est déplacé. Il subit donc une séquence *déconnexion – déplacement – reconnexion*.

Mobilité continue

Dans le deuxième cas, celui de *la mobilité continue*, les utilisateurs se déplacent au long d'une couverture réseau sans-fil contiguë, fournie par un ou plusieurs points d'accès. Ces points d'accès peuvent être équipés d'une même technologie et former des cellules adjacentes. L'autre cas est quand plusieurs technologies sans-fil créent des zones de couverture superposées.

Quand un terminal est connecté à un réseau sans-fil, il peut être déplacé sans problème dans le rayon de son point d'accès courant, les seules conséquences notables apparaissant au niveau des erreurs de transmission si le terminal s'éloigne. Par contre, une opération plus complexe est le transfert d'un terminal d'un réseau à un autre, désigné par le terme anglais *handoff*[1]. Le handoff entre deux cellules d'un même type de réseau est appelé *handoff horizontal*, tandis que le *handoff vertical* se fait entre des points d'accès d'une technologie sans-fil différente.

La continuité de la connexion au réseau malgré le changement de point d'accès est la caractéristique principale d'un handoff. Elle est exprimée par le niveau de pertes des paquets ou encore par le temps pris par le transfert de l'association. Ces pertes et délais apparaissent à cause du délai pris par la réassociation physique au nouveau point d'accès, mais aussi parce que la nouvelle localisation doit être apprise par les équipements réseau qui acheminent les paquets vers l'hôte mobile. Souvent dans le cas d'un handoff vertical mais parfois aussi dans les handoffs horizontaux, un changement d'adresse IP est nécessaire, ce qui induit d'autres délais supplémentaires et provoque même l'interruption des connexions réseau.

Dans le cas idéal où le handoff ne provoque pas d'interruption des services réseau, on parle d'un *handoff transparent* aux applications. Cependant, il existe des cas, par exemple ceux des applications intéressées par la position et le contexte, où il est préférable de ne pas cacher le handoff et de le notifier aux applications.

2.1.2. Mobilité de réseaux

Un cas particulier de la mobilité de terminaux est quand un sous-réseau entier se déplace, ses hôtes pouvant garder leur topologie inchangée à l'intérieur de ce réseau. Imaginons le cas de passagers d'un train ou d'un avion ou les équipements embarquées dans une voiture.

[1] Ou encore *handover* ou *roaming*

Malgré l'immobilité des hôtes par rapport au lien local, le réseau lui-même change ses liens avec les réseaux voisins et donc change sa position et ses interconnexions dans l'Internet. Au cas où les adresses IP du réseau restent inchangées, ce déplacement pourrait rester invisible pour ses hôtes. Cependant, dans la plupart des cas, le réseau doit opérer une renumérotation d'adresses, ce qui a un impact sur les terminaux. Dans ce cas, une place privilégiée pour implanter des fonctionnalités liées à la mobilité est dans le routeur de bordure du réseau mobile.

2.1.3. Mobilité de personnes

La plupart des utilisateurs se servent de plus d'un terminal pour communiquer à travers des application et services réseau. Il n'existe pas encore un seul terminal qui offre à la fois faible poids, petite taille, grande autonomie d'énergie, forte puissance de calcul et capacités multimédia étendues. Les utilisateurs font un compromis et utilisent différents terminaux en fonction de l'endroit et de la situation où nous nous trouvons. Par conséquent à la pluralité des terminaux et services de communication, l'utilisateur possède plusieurs *identifiants*, en fonction de l'application utilisée : *adresses e-mail* professionnelles et personnelles, *numéros de téléphone* (téléphone portable, téléphones à la maison et au bureau), et d'autres *noms d'utilisateur* pour d'autres applications Internet comme la messagerie instantanée et la téléphonie sur Internet.

Une fonction importante à accomplir dans le cadre de la mobilité de personnes est qu'un utilisateur puisse communiquer indépendamment de son terminal. Cela demande que tous les identificateurs énumérés plus haut soient regroupés pour que la personne puisse être localisée et les communications redirigées vers son terminal et application actifs. Les connexions réseau doivent donc être établies à un niveau d'abstraction plus élevé, entre des personnes, et non plus entre des terminaux ou d'applications.

2.1.4. Mobilité de services et applications

La mobilité de personnes décrite précédemment implique que les services et applications réseau sont disponibles et peuvent être utilisés d'une manière similaire, indépendamment du terminal courant de l'utilisateur. Pour satisfaire ceci, des applications entières ou des parties de code logiciel doivent être transférés dans certains cas d'une machine à une autre, même en cours d'exécution. On appelle ce transfert *mobilité de services et applications*.

Un exemple particulier de la mobilité de composants logiciels est le profil personnel de services utilisateur. La portabilité du profil de services signifie que les applications réseau fournissent les mêmes services, associés aux préférences de l'utilisateur, quelque soit son terminal actif. Le but est de créer un environnement personnel virtuel que l'utilisateur emporte partout pour accéder à ses données personnelles, à ses communications réseau et à ses applications favorites. Bien sur, l'environnement personnel doit aussi intégrer d'une manière transparente des services locaux – on veut avoir accès à des données personnelles, mais aussi accéder aux vidéoprojecteurs et imprimantes de l'endroit visité.

2.1.5. Mobilité de sessions

Une *session réseau* est une abstraction qui regroupe une ou plusieurs connexions réseau et qui fournit des services pour gérer globalement l'état de ces connexions. La mobilité au niveau d'une session doit permettre à ses connexions de rester actives et suivre les deux points finaux, en dépit de leur mobilité. La mobilité au niveau session est plus générale que la mobilité des hôtes ou la mobilité des personnes, puisque ces deux classes y en sont des instances particulières. Ainsi, comme on peut voir dans la figure 2.3, le déplacement d'un hôte peut être vu comme le déplacement d'une session regroupant toutes les connexions qui ont cette machine comme point final. De l'autre coté, la mobilité des personnes implique le déplacement des connexions ouvertes par une personne vers son nouveau terminal.

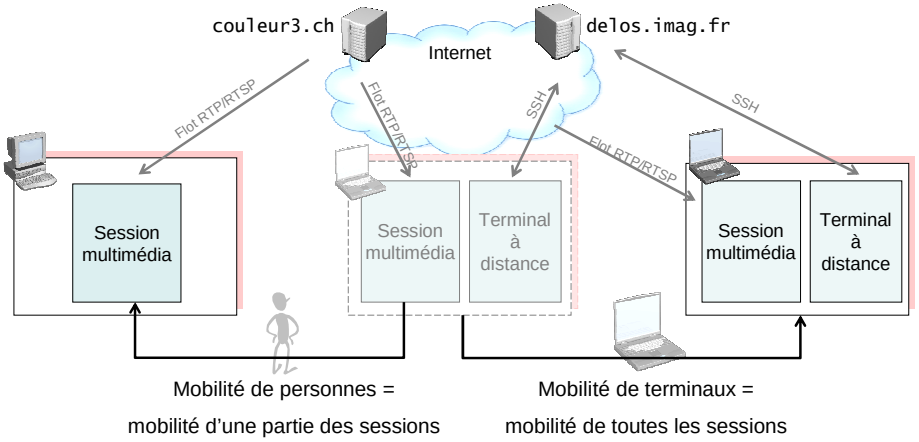


FIG. 2.3.: Mobilité au niveau session

Dans la pile de protocoles Internet il n'y a pas du support explicite pour regrouper plusieurs connexions dans une session, alors que ce niveau existait dans le modèle OSI (*Open Systems Interconnection*). En plus de l'initialisation d'une session et de la négociation des divers paramètres pour l'identifier, le niveau session du modèle OSI permet la définition de points de synchronisation de la session. Ceci fournit le support pour l'interruption et le redémarrage de la session à partir de ces points. Vu l'absence du niveau session dans les protocoles Internet, certaines applications ont été conçues dès le départ pour s'établir elles-mêmes une association de longue durée qui peut comprendre plusieurs connexions simultanées ou enchaînées dans le temps. On cite comme exemples des sessions HTTP pour la navigation web ou des sessions de transfert de fichiers sur les plates-formes P2P (*Peer to Peer*).

2.2. Fonctionnalités requises par la mobilité

Nous présentons dans cette section les démarches nécessaires pour assurer le bon fonctionnement de la mobilité des machines dans les réseaux Internet. Nous avons illustré ces opérations dans la figure 2.4. Ainsi, l'association à un nouveau point d'attachement à l'In-

ternet peut demander à l'hôte mobile de fournir des éléments d'authentification, ainsi que la reconfiguration de certains paramètres liés au nouveau réseau. Ensuite, une machine mobile qui remplisse des fonctions de serveur devrait pouvoir être retrouvée par ses clients. De plus, on veut assurer le transfert de toutes les connexions actives de l'hôte mobile vers sa nouvelle localisation sur le réseau.

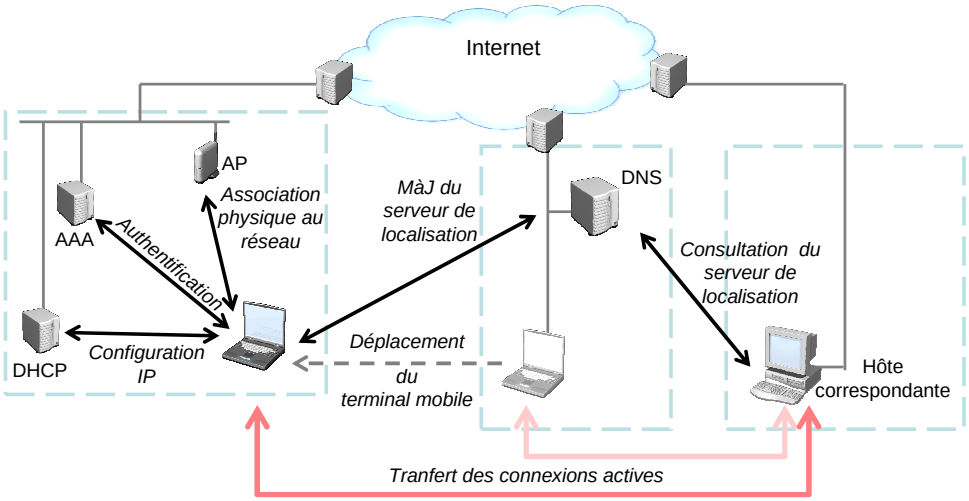


FIG. 2.4.: Opérations au cours de la mobilité

Nous précisons que la mobilité des personnes peut être synonyme du déplacement de son terminal actif. Cependant, la particularité qu'elle introduit est un niveau d'indirection supplémentaire, entre la personne et son « point d'attachement au réseau », c'est-à-dire son terminal actif. Les démarches demandées par la mobilité d'une personne sont en grandes lignes similaires à celles de la mobilité d'hôtes : authentification, mise à jour des informations qui servent à sa localisation, et le transfert du contexte utilisateur entre son ancien et son nouveau terminal. L'analyse de la suite d'opérations que nous allons présenter dans la suite est faite principalement du point de vue de la mobilité des hôtes, mais le même type de démarches est valable par exemple pour le transfert d'une session réseau suite au changement de terminal d'un utilisateur.

2.2.1. Association au réseau

La première étape est l'association physique du terminal au nouveau point d'attachement à l'Internet. Cette étape dépende fortement de la technologie d'accès à l'Internet : réseau local filaire Ethernet ou sans-fil Wi-FI, accès distant par modem classique ou en utilisant un téléphone GPRS. L'association au nouveau réseau peut demander une authentification de la part du terminal mobile. Par exemple, dans les réseau Wi-Fi, si le point d'accès utilise WEP (*Wired Equivallent Privacy*) ou WPA (*Wi-Fi Protected Access*), l'association physique au réseau sans-fil peut être conditionné par l'authentification du terminal, qui doit fournir une clé sécrète ou un mot de passe.

Après que l'association physique soit accomplie, d'autres interactions supplémentaire avec des serveurs AAA (*Authentication, Authorization, and Accounting*) peut encore avoir lieu pour que l'hôte mobile ait de droit d'utiliser en partie ou en totalité les services réseau ou pour que son activité réseau puisse être comptabilisée.

2.2.2. Configuration au niveau IP

Si l'hôte mobile se déplace entre deux sous-réseaux différents (par exemple si elle fait un handoff vertical, ou encore si elle change le réseau filaire pour une connexion sans-fil), une démarche importante qu'un hôte mobile doit accomplir est la reconfiguration de son interface réseau au niveau 3. Pour compléter sa connexion au nouveau sous-réseau, l'hôte doit être reconfiguré avec une nouvelle adresse IP, masque du sous-réseau, passerelle, serveur de noms, etc. Les travaux de recherche dans ce domaine ont abouti par l'apparition de protocoles de configuration automatique de ces paramètres, sans aucune intervention nécessaire de la part de l'utilisateur. Les protocole qui sont actuellement utilisés sont le DHCP (*Dynamic Host Configuration Protocol*)[14] pour l'IPv4 (et son correspondant et DHCPv6[15] pour la version 6 du protocole IP) et la configuration automatique de type *zeronconf*[16] (et sa variante intégrée dans le protocole de base IPv6[17]).

2.2.3. Mise à jour des informations de localisation

Correctement configuré avec une adresse IP propre au nouveau sous-réseau, l'hôte est maintenant en mesure d'initier des connexions avec des autres machines, sur le lien local ou à travers l'Internet. Par contre, si l'hôte mobile remplit la fonction d'un serveur, une démarche supplémentaire qu'il doit accomplir est de faire possible sa localisation par les clients potentiels.

Pour qu'un hôte mobile puisse être localisé par ses correspondants, il doit être identifié par autre chose que son adresse IP, si celle-ci change. Chaque fois qu'il se déplace et acquit une nouvelle adresse IP, il doit mettre à jour la correspondance entre son identifiant et son adresse courante dans un répertoire. Après que ce répertoire ait été mis à jour, les hôtes correspondants pourraient le consulter, apprendre la nouvelle adresse et initier des connexions vers l'hôte mobile. Le protocole Internet qui pourrait servir à ce but est le DNS, mais il présente néanmoins quelques inconvénients vis-à-vis de la mise à jour des correspondances nom – adresse IP, qu'on présentera dans le chapitre suivant.

2.2.4. Transfert des sessions réseau

Les démarches présentées plus haut peuvent suffire pour le modèle de mobilité qu'on a appelé auparavant *portabilité*. Cependant, si un terminal mobile se déplace pendant qu'il a des connexions réseau actives, un objectif additionnel apparaît : transférer les session réseau vers la nouvelle adresse de l'hôte mobile. Notifier les correspondants de l'hôte mobile pour que ceux-ci envoient leurs paquets vers la nouvelle adresse de l'hôte mobile ne résout pas

en totalité le problème. La cause vient des protocoles du niveau transport et de certaines applications, qui ne permettent pas le changement à la volée d'adresses IP (locale ou de la machine correspondante) au cours d'une connexion. Ceci est en effet un des problèmes les plus difficiles à résoudre pour la mobilité d'hôtes – nous l'expliquerons plus en détail dans le chapitre suivant.

2.3. Multi-accès et renumérotation

Le changement d'adresse IP d'une hôte peut avoir lieu dans d'autres cas, soit que cela soit liée à un déplacement physique de la machine en cause. Deux exemples de telles situations sont la renumérotation et le multi-accès.

La renumérotation signifie qu'une machine est obligée de reconfigurer son adresse IP, pour diverses raisons. Par exemple, cela intervient quand une machine utilise DHCP pour configurer automatiquement l'interface réseau et, suite à une expiration de bail, on lui attribue une nouvelle adresse. La renumérotation peut aussi apparaître dans un sous-réseau qui change son préfixe, ce qui se répercute dans les adresses de toutes les hôtes (nous expliquerons cela dans le chapitre suivant, quand nous discutons la structure des adresses IP).

Un autre cas de *pseudo-mobilité* est causé par le *multi-accès*[2] – terme qui désigne une machine qui possède plusieurs interfaces réseau et qui les utilise simultanément ou alternativement. Théoriquement, le multi-accès simultané peut être utile pour augmenter la robustesse et le débit des connexions. Mais utiliser plusieurs interfaces et chemins concurrents pour acheminer des paquets entraîne souvent un taux élevé de désordre dans l'arrivée de paquets, fait qui réduit la performance au lieu de l'augmenter. En pratique, le multi-accès se résume à l'usage successif ou alternatif des interfaces réseau. Même dans ce cas, les connexions réseau ne peuvent pas être tout simplement redirigées d'une interface à l'autre. La raison est celle expliqué dans la section précédente : les différentes interfaces ont des adresses IP différentes et les protocoles de niveau transport de permettent pas le changement à la volée des adresses IP pendant le temps de vie d'une connexion.

2.4. Reprise des déconnexions réseau

Le déplacement d'un terminal mobile est souvent accompagné de périodes de déconnexion. Le déploiement des réseaux sans-fil s'est traduit surtout par la présence des réseaux sans-fil locaux qui offrent des îlots de connectivité plus ou moins larges. L vision du réseau sans-fil global, toujours disponible, est encore loin de s'accomplir. Autre facteur duquel les utilisateurs doivent tenir compte aujourd'hui est que les technologies sans-fil consomment de l'énergie, et à ce sujet les batteries des équipements portables sont généralement limitées. Les déconnexions se produisent plutôt d'une façon inattendue, puisque l'utilisateur peut sor-

[2] En anglais *multihoming*.

tir involontairement de la portée d'un point d'accès. Finalement, la durée de la période de déconnexion est inconnue, de quelques secondes à quelque heures ou même plus.

Les systèmes d'exploitation des machines portables sont aujourd'hui dotés d'un mécanisme de suspension de l'exécution qui permet de définir un point de synchronisation, de sauvegarder l'état interne du système et de le mettre dans un état d'hibernation où les ressources d'énergie sont conserves. Après des périodes arbitraires de inactivité, l'opération peut reprendre à partir de ce point de synchronisation.

Malheureusement, la partie réseau de ces systèmes ne comprend pas un support pour que les sessions réseau puissent survivre pendant les interruptions. Les protocoles Internet détectent l'inactivité et la traduisent en un échec de connexion, et la session est interrompue. L'absence d'un niveau session explicite dans la pile de protocoles Internet fait que cette notification d'échec d'une connexion arrive à la couche application qui est forcée d'abandonner la connexion en cause et de re-ouvrir une nouvelle. Même si certaines applications savent gérer ces interruptions d'une manière transparente, la plupart les considèrent comme une anomalie et présentent en conséquence une erreur à l'utilisateur. Un mécanisme d'*hibernation* serait donc utile pour pouvoir traiter les déconnexions apparues au cours du mouvement d'un terminal portable, ainsi que les arrêts volontaires sur des périodes plus longues.

2.5. Conclusion

Nous avons commencé ce chapitre par la définition de la mobilité dans les réseaux. Nous avons ensuite présenté les différentes types de mobilité, entre lesquels la mobilité des terminaux et la mobilité des personnes. À un niveau d'abstraction plus élevé nous pouvons discuter de la mobilité d'une session de connexions réseau, concept qui permet de regrouper la mobilité des terminaux ou des personnes.

Dans la deuxième partie du chapitre, nous avons précisé quelles sont les fonctionnalités demandées par la mobilité des hôtes. Nous avons remarqué que des problèmes similaires, liés au changement de l'adresse IP et à l'impossibilité de transférer la session réseau, apparaissent aussi pour les hôtes multi connectés ou dans le cas d'une renumérotation. Pour mieux comprendre ces problèmes, nous allons analyser dans le chapitre suivant les protocoles de l'Internet, en mettant en évidence leur inconvénients ou leurs lacunes vis-à-vis de la mobilité d'hôtes.

3. Mobilité dans la pile TCP/IP

La suite de protocoles utilisé dans l'Internet est communément désignée sous le nom TCP/IP[18, 19]. Ce nom vient de ses deux principaux composants, le *Transmission Control Protocol* au niveau transport et *Internet Protocol* au niveau réseau. TCP a été développé dans les années 1970[20, 21] sous le nom *Internet Transmission Control Program* avant d'être divisé en TCP et IP en au début des années '80 et ensuite utilisé dans l'ARPANET (*Advanced Research Projects Agency Network*)[1].

Les protocoles Internet sont structurés en plusieurs *couches*. Le niveau *liaison* s'occupe de l'accès au lien local et de la communication effective entre les machines qui y sont connectées directement. Le niveau *réseau* est lui-même formé d'un ensemble de protocoles - IP, ICMP, ARP, etc. - qui collaborent à unifier les différents liens physiques et fournir le service « *réseau global unifié* » offert par IP. Au niveau *transport*, il y a les protocoles TCP et UDP (*User Datagram Protocol*). Enfin, le niveau *application* utilise les services de transfert de données de bout en bout fournies par les protocoles de transport TCP et UDP.

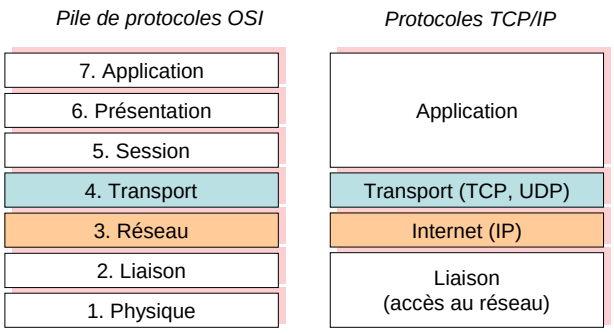


FIG. 3.1.: Les piles de protocoles OSI et TCP/IP

Nous montrons ces couches et le parallélisme avec la pile de protocoles OSI[2] dans la figure 3.1.

[1] Le premier réseau reliant des ordinateurs, conçu par l'agence de recherche du ministère américain de la défense et qui a constitué la première brique de l'Internet.

[2] Le modèle de référence OSI[22] mis au point par l'ISO (*International Standard Organisation*) est la référence théorique pour les protocoles de communication. Il est structuré en sept couches classées par ordre d'abstraction croissant, les protocoles spécifiques utilisés dans chacune des couches coopérant pour assurer une communication efficace.

3.1. L'acheminement de datagrammes vers les machines mobiles

Le protocole IP[23] est considéré comme étant le plus important d'Internet, car il unifie tous les réseaux physiques dans un seul réseau global. Il masque ainsi l'hétérogénéité des différentes technologies physiques et se charge d'acheminer les données entre n'importe quelle paire d'hôtes, indifféremment du sous-réseau ou celles-ci se trouvent.

Chaque point d'attachement à l'Internet a une adresse IP unique, par laquelle l'hôte qui y est connecté est joignable par les autres machines. Chaque paquet de données (*datagramme*) envoyé via l'Internet contient l'adresse IP destination du paquet, qui sera utilisé dans le processus d'acheminement à travers l'Internet.

L'acheminement est fait par les routeurs, des machines ayant plusieurs interfaces réseau, chacune reliée à un sous-réseau différent. Les routeurs maintient une table de routage qui contient des correspondances entre des préfixes des adresses IP et le nœud suivant auquel le routeur doit délivrer le message. La séquence de consultation de tables de routage et de retransmission du datagramme est répétée sur plusieurs routeurs, jusqu'au moment où le paquet arrive à sa destination.

Dans l'IPv4, les adresses IP sont codées sur 32 bits ; la nouvelle version IPv6 utilise des adresses IP d'une longueur de 128 bits. On a donc un nombre immense[3] d'adresses IP possibles. En conséquence, les tables de routage de routeurs ne peuvent pas contenir la totalité des adresses des hôtes connectés à l'Internet. Pour que le processus de routage soit viable à l'échelle actuelle et future de l'Internet, les adresses IP ont été structurées dès le début d'une façon hiérarchique : un partie pour l'adresse du sous-réseau et le reste pour identifier les hôtes du sous-réseau respectif.

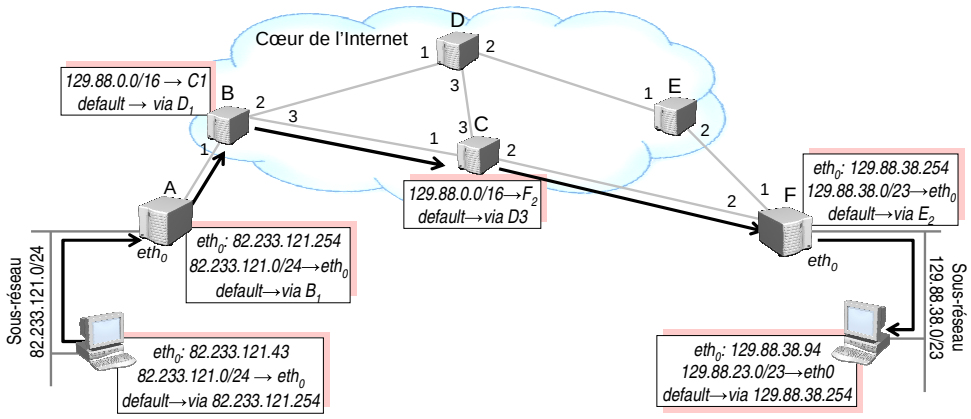


FIG. 3.2.: Le niveau réseau : les adresses IP et l'acheminement de datagrammes

Comme on peut voir dans la figure 3.2, seulement la partie sous-réseau des adresses IP est présente dans les tables de routage. Si la structuration des adresses permet la simplification des tables de routage, elle introduit néanmoins une restriction très importante du point de

[3]En réalité, moins que 2^{32} ou 2^{128} car certaines plages d'adresses sont réservés pour un usage spécial[24].

vue de la mobilité d'hôtes. En effet, une adresse IP ne peut être utilisée que dans le sous-réseau dont elle fait partie, car les datagrammes qu'elle reçoit sont acheminées en utilisant l'adresse du sous-réseau. Chaque fois qu'un hôte se déplace, elle doit utiliser une nouvelle adresse conforme au nouveau point d'attachement, c'est-à-dire qui contient le numéro du nouveau sous-réseau.

3.2. Mobilité dans les réseaux locaux commutés

Nous avons montré dans la section précédente comment les datagrammes circulent entre deux sous-réseaux distants, par l'intermédiaire des routeurs. Cependant, la couche 3 et le protocole IP ne s'occupent pas de transmettre physiquement le paquet d'une machine à une autre sur un même lien. C'est la couche liaison qui se charge de transmettre le datagramme entre la machine émettrice et son routeur local, entre chaque paire de routeurs intermédiaires et finalement entre le dernier routeur et la machine destination.

Les équipements actifs

Le protocole typique au niveau liaison, au moins dans le cas des réseaux filaires, est l'*Ethernet*[25]. Son principe général est que toutes les machines connectées à un même lien physique partagent le medium de communication. La topologie la plus simple est celle d'un lien Ethernet totalement partagé - chaque message émis est reçu par l'ensemble des machines raccordées au lien et la bande passante disponible est partagée entre les machines. Cependant, des équipements réseau *actifs*[4] sont souvent interposés dans les réseaux Ethernet. Un des intérêts de la présence de ces équipements est de regrouper différentes technologies physiques pour former un seul lien au niveau 2. À l'inverse, si on a beaucoup de machines sur un même lien physique, les équipements actifs permettent de diviser le sous-réseau en plusieurs liens pour isoler les collisions potentielles et profiter de la totalité de la bande passante à l'intérieur de chaque lien.

Des exemples de tels équipements actifs sont les ponts, qui séparent deux réseaux physiques distincts, et les commutateurs - des ponts multi-ports qui relient plus de deux réseaux (voir la figure 3.3). Lorsqu'un pont ou un commutateur reçoit une trame sur l'une de ses interfaces, il analyse les adresses physiques de la machine émettrice et destination. Si jamais le pont ne connaît pas l'émetteur, il sauvegarde son adresse dans une table afin de se souvenir de quel côté du réseau se trouve l'émetteur. Ainsi le pont ou le commutateur pourrait savoir si l'émetteur et le destinataire sont situés du même côté, cas où ils ignorent la trame. Par contre, si la machine destination est sur un autre lien, le pont ou le commutateur transmettent la trame sur ce lien. Une troisième situation est possible dans le cas d'un commutateur qui ne connaît pas le lien de l'hôte destinataire ; dans ce cas, la trame va être diffusé sur tous les liens.

[4] Il s'agit des équipements qui analysent les adresses physiques des trames diffusées sur le réseau et opèrent en conséquence.

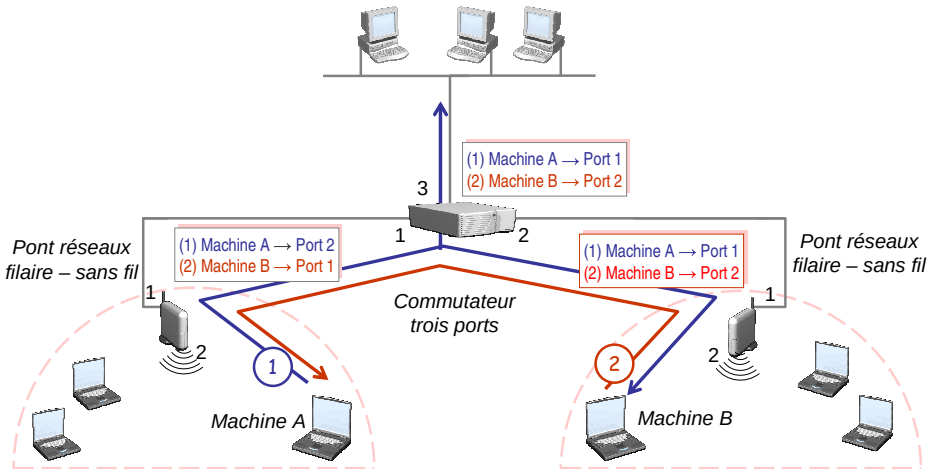


FIG. 3.3.: Les équipements actifs au niveau liaison

La plupart des réseaux sans-fil sont déployés en utilisant un point d'accès qui fonctionne comme un pont entre la cellule sans-fil et un réseau filaire réunissant les autres points d'accès et machines. Grâce à cela, toutes les hôtes font partie d'un seul sous-réseau au niveau IP. Au moment du handoff d'un terminal entre deux cellules, la seule opération qui doit être accomplie est la mise à jour de la nouvelle localisation de l'hôte mobile au niveau des deux point d'accès et autres équipements actifs. Cette mise à jour est automatiquement effectué par une trame spéciale[5] envoyé par le nouveau point d'accès.

3.3. DNS : Résolution de noms des machines mobiles

Les 32 bits des adresses IP sont représentés habituellement sous une forme décimale, comme une suite de quatre nombres de 0 à 255. Par exemple la machine qui héberge le site web de l'équipe Drakkar à l'IMAG a l'adresse `129.88.38.94`. Les adresses de la version IPv6, d'une longueur de 128 bits, sont à leur tour représentées usuellement par huit groupes de quatre caractères hexadécimales. L'adresse IPv6 de la même machine est `2001:660:5301:26:210:83ff:fe35:3404`.

Pour faciliter les choses, la même machine a le nom `delos.imag.fr`[6]. On observe que les noms d'hôte sont eux aussi organisés d'une façon hiérarchique, sous la forme d'une succession de chaînes de caractères. La raison de cette hiérarchisation est liée à la taille de l'Internet et à l'espace pratiquement infini de noms d'hôte, et elle permet d'administrer d'une façon distribué l'espace de noms. Contrairement à la structuration des adresses IP, il n'y a pas de contraintes liées à la mobilité : une machine peut théoriquement garder son nom si elle se déplace temporairement sur un nouveau sous-réseau.

[5] Cependant, cette procédure standard est spécifiée dans la recommandation IEEE 802.11f[26], mais sans qu'elle soit présente dans toutes les implémentations

[6] Et plusieurs autres alias, parmi lesquels `drakkar.imag.fr`.

Le mécanisme de translation entre noms et adresses IP s'appelle DNS (*Domain Name System*)[27, 28] et fonctionne d'une façon distribuée, suivant la hiérarchie de noms d'hôte. L'hiérarchie DNS est structuré sous la forme d'un arbre, avec un *domaine racine*, des *domaines de niveau supérieur* et des *domaines de niveau inférieur*. Chaque domaine est géré par une organisation qui fournit et administre un ou plusieurs serveurs qui gardent les translations entre les noms faisant partie du domaine respectif et les adresses IP correspondantes.

Pour illustrer le processus de résolution de noms, nous montrons dans la figure 3.4 l'exemple d'un utilisateur situé sur une machine distante et qui veut consulter le site web `http://drakkar.imag.fr`. La machine distante commence par contacter son serveur DNS local, en l'interrogeant sur le nom `drakkar.imag.fr`. Cette requête est de type *récursive*, c'est-à-dire que ce serveur DNS local doit résoudre cette requête avant de répondre à l'hôte qui l'a initié. Pour pouvoir répondre à la requête, le serveur DNS distant va effectuer à son tour plusieurs résolution de nom, mais celles-ci seront de type *itératif*.

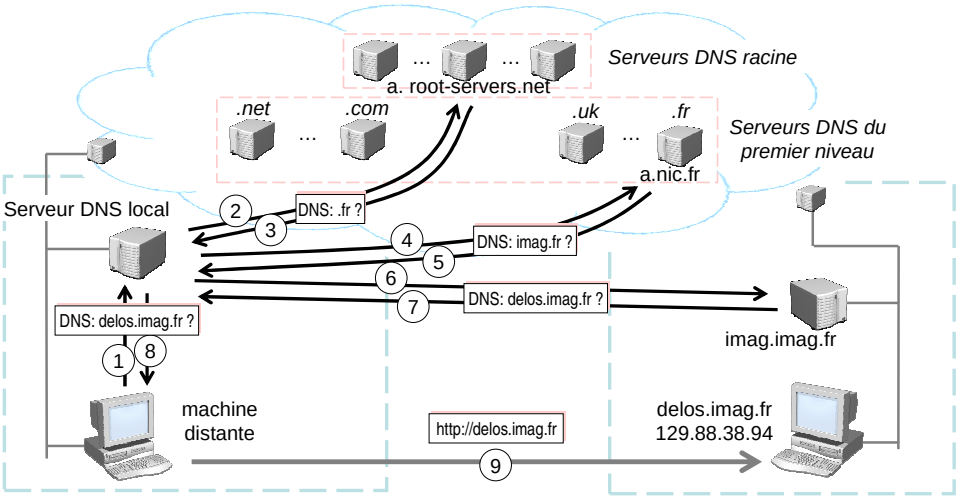


FIG. 3.4.: La résolution de noms par DNS

Pour minimiser le nombre de requêtes DNS dans la partie supérieure de l'arbre DNS et optimiser ainsi le temps de réponse pour, chaque association nom - adresse possède une certaine durée de vie, indiquée dans le champ TTL (*Time To Live*). Les serveurs DNS utilisent cette information pour déterminer combien de temps ils peuvent garder la réponse dans un *cache* et répondre directement aux requêtes subséquentes.

Une valeur par défaut de 24 heures pour le TTL est habituelle actuellement dans les caches des enregistrements DNS[29]. Pendant ce temps, une éventuelle mise à jour de l'association nom - adresse IP ne sera pas propagée vers les serveurs DNS distants qui la garde dans le cache. Des durées de vie plus petites, de l'ordre de quelques minutes ou secondes sont rencontrés pour les noms de serveurs web populaires qui changent périodiquement l'association nom - adresse IP pour distribuer la charge du serveur virtuel sur plusieurs machines physiques. À la limite, une valeur nulle dans le champ TTL interdira la mise en cache de réponses sur les serveurs DNS distants mais entraînera une surcharge des échanges de requêtes et réponses DNS relatifs au couple *nom – adresse* en cause.

Un TTL de 0 ne résout pas totalement le problème de la propagation des mises à jour, car la valeur TTL n'est pas transmise au niveau application qui avait demandé la résolution du nom. Il est courant pour certaines applications Internet d'avoir un propre cache pour les translations *nom d'hôte – adresse IP* [7].

3.4. Mobilité des connexions au niveau transport

Quand les protocoles TCP/IP ont été conçus, on a implicitement considéré que les machines sont statiques et ne changeront pas le point d'attachement et l'adresse IP pendant la durée de vie d'une connexion. En conséquence, au lieu d'identifier les hôtes par d'autres moyens, les protocoles du niveau transport et les applications ont été développés en utilisant les adresses IP comme des identificateurs stables des extrémités d'une connexion.

Dans la composition du trafic global Internet, TCP (*Transmission Control Protocol*)[32] est majoritaire : il représente 80% des flots de données, 90% des datagrammes et 95% des octets échangés[33]. Puisqu'il offre des garanties pour le transport des données à destination, TCP est utilisé par toutes les applications qui ont besoin de la fiabilité dans l'échange de données sur l'Internet.

Néanmoins, il présente un problème majeur vis-à-vis de la mobilité, car une connexion TCP est identifiée par un quadruple *<adresse IP source, port source, adresse IP destination, port destination>*. Une fois qu'une connexion TCP est établie, chacune des deux extrémités va envoyer et recevoir des données seulement vers et à partir d'une machine située à une adresse IP fixe.

Cela entre en conflit avec la mobilité d'hôtes : si une machine se déplace vers un sous-réseau différent, elle ne peut pas garder l'ancienne adresse IP, car les paquets qui y sont adressés seront acheminés à l'ancien sous-réseau. À la place, la machine doit changer son adresse pour pouvoir réceptionner les données au nouvel endroit où elle se trouve. Mais dans ce cas elle ne pourra pas continuer les connexions TCP déjà ouvertes, car ces connexions n'acceptent pas de données envoyées ou reçues d'une ou à une adresse IP différente ; en conséquence, ces connexions TCP seront interrompues.

Le même problème apparaît également dans le cas des hôtes ayant un multi-accès au réseau. Les différentes interfaces sont en général connectées à des sous-réseaux différents et donc configurées avec des adresses IP différentes. Dans cette situation également, il est impossible, du point de vue du protocole TCP, de changer d'une manière transparente les interfaces utilisées pour envoyer ou recevoir les datagrammes.

L'autre protocole de transport, UDP[34], est un protocole simple, sans connexion et non fiable, qui ne garantit pas la livraison des paquets à destination ou leur arrivée dans l'ordre qu'ils ont été transmis.

[7] Par exemple 15 minutes dans le cas des anciennes navigateurs web *Mozilla/Netscape* et 30 minutes pour les anciennes versions de leur concurrent *Microsoft Internet Explorer*)[30, 31]. Les versions actuelles en corrigé en partie cela, dans le sens que MS-IE ne garde plus un cache, et la durée de vie des caches des navigateurs Mozilla a été réduit à 1 minute

Par rapport au TCP, dans UDP chaque paquet est envoyé indépendamment des autres, et la même socket UDP peut être utilisée pour envoyer et recevoir des paquets vers et de n'importe quelle adresse IP. Ceci permet des changements dans les adresses IP des machines source et destination et il semble faciliter la mobilité des hôtes. En pratique les choses ne sont pas si simples : des applications construites autour de UDP utilisent une seule socket pour plusieurs connexions avec des correspondants différents. Ces applications utilisent alors l'adresse IP source des datagrammes reçues pour démultiplexer les données de ces connexions virtuelles. Ce type d'applications doivent être informées en avance sur un changement d'adresse IP distante pour continuer à garder un état correct sur leurs échanges de données. Néanmoins, cette notification devrait suffire pour pouvoir envoyer et recevoir des paquets de la nouvelle adresse, sans la surcharge introduite par le redémarrage d'une nouvelle connexion dans le cas du TCP.

3.5. Conclusion

Nous avons vu dans ce chapitre que les problèmes vis-à-vis de la mobilité des hôtes dans l'Internet sont issus principalement du conflit entre les deux rôles d'une adresse IP : adresse d'hôte utilisé comme indicateur de routage, ainsi qu'identificateur d'hôte dans une connexion TCP. Pour trouver une réponse à cette incohérence et pour avoir une différenciation nette entre ces deux rôles, nous allons approfondir dans ce qui suit les concepts présents dans les réseaux, pour pouvoir extraire ensuite les vraies notion d'identificateur et adresse d'une hôte.

Une première énumération des concepts réseau a été faite par J. Shoch, qui propose dans [35] la terminologie suivante : un *nom* identifie la cible d'une connexion, une *adresse* identifie où cette cible se trouve et un *chemin* identifie la route à suivre pour y arriver[8]. Cette terminologie utilise un nombre minimum d'objets fondamentaux et leur attribue des identifiants, mais sa déficience principale est l'omission d'une distinction nette entre les objets et les noms. Une contribution importante est la classification plus claire qui est faite par J. Saltzer dans [36]. Il propose 4 types d'objets dans le réseau et précise leur corrélation :

- Les *services*, qui représentent les fonctions de haut niveau disponibles par le biais du réseau et les *utilisateurs*, c'est-à-dire les clients qui les utilisent ;
- Les *nœuds*, qui sont les machines sur lesquelles tournent les services et les applications des utilisateurs ;
- Les *points d'attachement au réseau*, représentant les points d'entrée et de sortie sur le réseau où les nœuds sont attachés ;
- Les *chemins* entre les points d'attachement au réseau, qui peuvent traverser plusieurs nœuds intermédiaires et plusieurs liens physiques.

[8] Les termes précis utilisé par l'auteur sont en anglais *name, address* et *route*.

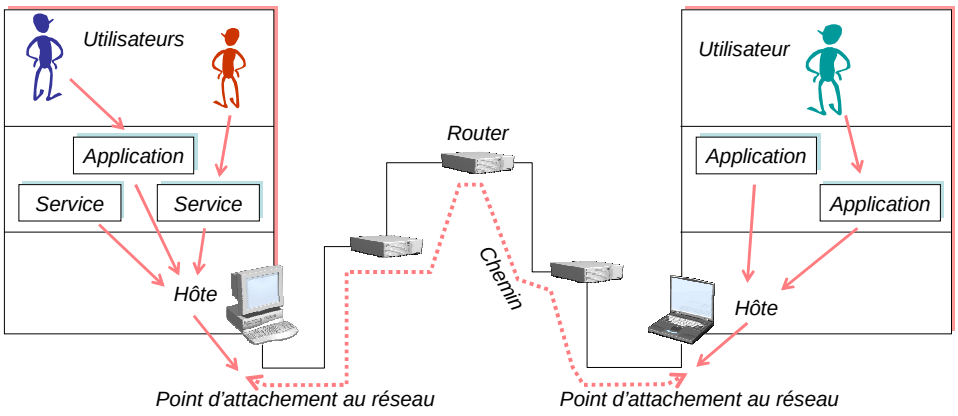


FIG. 3.5.: Les concepts et entités présents dans les réseaux

Chacun de ces quatre types d'objets peut être identifié par un ou plusieurs *noms*, qui peuvent avoir diverses formes. La dépendance entre les quatre catégories d'objets présentées plus haut peut être décrite en terme d'associations temporaires. Ainsi, un service peut être exécuté sur un ou plusieurs nœuds et peut être déplacé d'un nœud à un autre sans perdre son identité. D'une manière identique, un nœud peut être connecté à un ou plusieurs points d'attachement, et peut être déplacé d'un point à un autre sans perdre son identité de nœud.

Nous estimons que cette association entre un service et un nœud ou entre un nœud et un point d'attachement au réseau représente le vrai concept d'*adresse* – l'adresse d'un service est un nœud, comme l'adresse d'un terminal est le point d'attachement au réseau ou elle est connectée. Si on cherche à appliquer ce modèle à la pile de protocoles TCP/IP, la définition d'une adresse IP est d'être l'identifiant du point d'attachement au réseau. En conséquence, elle représente une adresse possible pour un terminal. Les noms d'hôte sont traduits par l'intermédiaire du système DNS dans des adresses IP mais leur rôle actuel est plutôt d'être un autre nom des points d'attachement au réseau, dans une forme plus adéquate pour les utilisateurs humains. Cependant, rien n'empêche qu'on pourrait utiliser les noms d'hôte pour identifier les terminaux mobiles. Néanmoins, il faut faire un choix et avoir un seul emploi pour les noms d'hôte. Aussi, il est impératif soit d'utiliser des valeurs TTL appropriés dans les enregistrements DNS, soit de choisir un autre système de résolution de noms qui fournit une cohérence globale des mises à jour chaque fois qu'une machine change d'adresse.

Dans la partie suivante de ce document, nous allons présenter diverses solutions qui ont été proposées pour gérer la mobilité d'hôtes. Ces solutions choisissent d'utiliser différents espaces de noms pour les différentes entités réseau. Nous allons voir dans les chapitres suivants comment l'unicité des identifiants est assurée et comment la résolution nom – adresse est réalisée dabs chacune de ces solutions. À la fin, nous allons présenter dans un tableau comparatif les choix faits par chacune des propositions discutées.

Deuxième partie .

État de l'art

4. Solutions au niveau IP

À partir du début des années '90, une multitude des solutions ont été proposées pour faire face aux défis posés par la mobilité des hôtes dans les réseaux IP. La solution « officielle » proposée à l'IETF n'a pas réussi d'obtenir le consensus nécessaire pour lui permettre un déploiement à grande échelle. D'autres propositions intervenant à différents endroits de l'infrastructure Internet ont été étudiées, chacune avec ses avantages et ses faiblesses. En conclusion, au moment actuel, aucune solution ne s'est imposée pour répondre d'une manière satisfaisante à tous les problèmes de la mobilité évoqués dans les chapitres précédents.

Nous avons classifié les différentes propositions en fonction de la couche protocolaire où elles opèrent et apportent des extensions. Une analyse de chaque solution sera présentée dans la suite, avec un accent mis sur les propositions actionnant au niveau IP.

4.1. Mobile IP

Le développement des extensions du protocole IP pour la mobilité des hôtes a débuté au sein de l'IETF (*Internet Engineering Task Force*) au début des années 90. Les extensions du protocole IP sont regroupées dans le protocole appelé *Mobile IP*, le même nom que le groupe de travail qui les a introduit[1]. L'Internet a continué d'évoluer et des nouvelles problématiques et contraintes sont apparues. En réponse, des nouvelles fonctionnalités et améliorations ont été proposées et ajoutées au standard spécifié initialement dans le RFC 2002[37]. Actuellement, les documents les plus récents qui spécifient les extensions pour la mobilité des hôtes sont le RFC 3344[38] pour IPv4 et le RFC 3775[39] pour sa version IPv6.

Mobile IP se veut une solution qui intervient exclusivement au niveau IP et qui fournit la transparence vis-à-vis des couches supérieures, y compris le protocole TCP. L'autre point important pris en compte dès le début dans la conception de Mobile IP, au moins pour sa version v4, a été la compatibilité avec les hôtes correspondants. Nous discutons dans cette section le protocole Mobile IPv4, les spécificités de Mobile IPv6 étant présentés dans la section 4.2.4.

[1] L'ancien groupe *mobileip* est maintenant divisé en deux groupes indépendantes, un pour chaque de deux versions IPv4 et IPv6.

L'architecture de Mobile IP

Dans Mobile IP, un hôte mobile a toujours associé une adresse IP de base qui reste inchangée. Celle-ci correspond au sous-réseau d'origine de l'hôte mobile. Quand l'hôte se connecte dans un sous-réseau différent, il dispose d'une adresse temporaire, propre au nouveau point d'attachement. Il continue cependant d'utiliser son adresse IP fixe dans la communication avec ses correspondants. Dans le schéma d'opération présenté dans la figure 4.1, les paquets destinés à l'hôte mobile sont toujours adressés à son adresse de base. Un nœud spécial dans le sous-réseau d'origine, appelé *agent mère*, intercepte les paquets et les remet à l'emplacement actuel de l'hôte mobile.

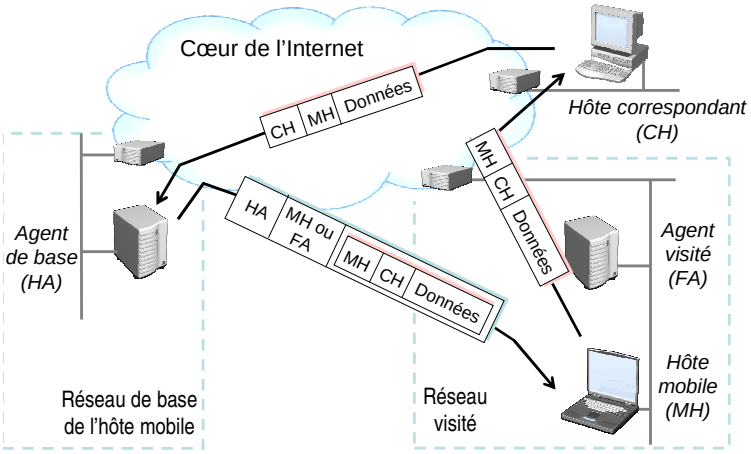


FIG. 4.1.: L'architecture de Mobile IP

L'hôte mobile peut utiliser deux méthodes différentes pour recevoir les paquets qui lui sont adressés pendant qu'ils se trouve connecté sur un autre sous-réseau qui celui d'origine. Dans la première, l'hôte mobile reçoit sa propre adresse IP sur le réseau visité, par exemple par un service de configuration automatique comme DHCP[14]. Chaque paquet reçu dans une première phase par l'agent-mère est empaqueté dans un nouveau datagramme qui est transmis directement à cette adresse. L'hôte mobile reçoit ce datagramme et extrait le paquet original, contenant son adresse de base. La deuxième méthode est motivée par le fait qu'il est difficile de réserver et gérer d'une manière efficace un espace d'adresses pour les machines mobiles qui visitent un domaine. Mobile IP introduit alors la notion d'*agent visité*, un autre nœud spécial qui se trouve dans le réseau visité et qui représente l'hôte mobile auprès du son agent-mère. Dans ce cas, les paquets envoyés à l'adresse de base de l'hôte mobile et interceptés par l'agent de base sont encapsulés et retransmis à l'agent visité. Celui-ci récupère les paquets originaux et les remets à l'hôte mobile sur le lien local.

Dans le sens inverse, les paquets envoyés par l'hôte mobile à ses correspondants ne passent pas par l'agent-mère du réseau d'origine et sont routés normalement à travers Internet en fonction de leur adresse destination. Parce que les hôtes correspondantes ignorent les mécanismes de Mobile IP, l'hôte mobile doit utiliser son adresse de base comme adresse

source de paquets envoyés. Cette différence entre les deux chemins suivis par les paquets reçus et les paquets envoyés par l'hôte mobile s'appelle *routage triangulaire*.

Les interactions entre l'hôte mobile et les agents Mobile IP

Un hôte mobile peut déterminer si elle est sur son réseau d'origine ou pas par un mécanisme de découverte d'agents de mobilité. Mobile IP utilise les paquets ICMP[40] existants de type *Router Advertisement* et *Router Solicitation* et y rajoute des extensions spécifiques. Les paquets envoyés par les agents visités contiennent notamment les adresses temporaires à utiliser et plusieurs autres informations comme le temps de validité de message ou la disponibilité de l'agent. Les agents de base n'ont pas d'adresse temporaire à diffuser, le but de leurs messages est que l'hôte mobile se rende compte de son reconnexion sur le réseau d'origine et cesse d'utiliser Mobile IP.

Pour qu'une hôte mobile puisse demander à son agent de base une redirection de messages vers une adresse temporaire, Mobile IP utilise deux messages spéciales (*Registration Request* et *Registration Reply*), les deux envoyés par UDP sur le port numéro 434. Le processus d'enregistrement vise notamment à notifier à l'agent de base le couple d'adresses fixe - temporaire nécessaire pour la redirection de ses paquets. L'éventuel agent visité a un rôle passif dans la procédure d'enregistrement et il ne fait que passer les messages de demande de l'hôte mobile à l'agent de base et les réponses en sens inverse.

Le processus d'enregistrement d'une adresse temporaire doit être authentifié et sécurisé, puisque sinon quelqu'un d'autre peut demander à un agent de base une redirection vers son adresse et intercepter tous les paquets destinés à une certaine adresse. Pour l'authentification, la méthode préconisé est de signer les messages conformément à l'algorithme HMAC-MD5 (*Hashed Message Authentication Code with Message Digest version 5*)[41], utilisant une clé de 128 bits échangée auparavant.

L'acheminement des datagrammes

L'agent de base doit utiliser deux fonctions spéciales du protocole ARP (*Address Resolution Protocol*)[42] - *proxy ARP* et *gratuitous ARP* - pour intercepter sur le réseau d'origine les datagrammes destinées à l'hôte mobile. Même dans le cas où la fonction d'agent de base est remplie par l'unique routeur du sous-réseau, ces fonctionnalités d'ARP restent nécessaires pour capturer les paquets envoyés par les autres machines sur le lien local. Pour cela, dès qu'il reçoit une demande d'enregistrement d'un hôte mobile, l'agent de base diffuse un paquet *ARP Reply*. Le rôle de ce message est de mettre à jour les caches ARP des autres machines et de faire correspondre son adresse physique à l'adresse fixe de l'hôte mobile. Ensuite, il répond à toutes les requêtes ARP subséquentes sur l'adresse fixe de l'hôte mobile. À son reconnexion sur le réseau d'origine, l'hôte mobile doit diffuser un paquet *ARP Reply* avec son propre adresse physique pour remettre à jour les caches ARP et se désenregistrer de l'agent de base.

Finalement, on précise comment la redirection des paquets de l'agent de base vers l'empla-

cement actuel de l'hôte mobile est réalisée. La technique utilisée, qui consiste à encapsuler le datagramme IP originale dans un en-tête IP supplémentaire, s'appelle *tunnelling*. Le datagramme obtenu comme résultat de l'encapsulation est envoyé par l'agent de base à l'adresse temporaire de mobile. Plusieurs algorithmes d'encapsulation sont spécifiés par Mobile IP. Celui qui doit être implémenté par toutes les entités impliquées (c'est-à-dire l'hôte mobile, l'agent de base et si le cas l'agent visité) est l'encapsulation IP-en-IP[43]. Cet algorithme est préféré aux autres car il est le seul à permettre la fragmentation des paquets sur les liens si leur taille dépasse la valeur de MTU (*Maximum Transmission Unit*).

4.2. Optimisations et extensions de Mobile IP

Les objectifs principaux de Mobile IP ont été atteints par le protocole de base présenté plus haut : la mobilité d'une hôte est transparente pour les hôtes correspondantes et pour les protocoles de niveau supérieur. Cependant, plusieurs points obstacles bloquent son acceptation générale et son déploiement à l'échelle de l'Internet :

- Le filtrage à l'entrée effectué par les routeurs de bordure ;
- Le routage indirect via l'agent de base ;
- La surcharge induite par l'encapsulation des paquets ;
- Le temps de handoff potentiellement important[44, 45].

Tous ces problèmes ont attiré l'attention du groupe de travail de l'IETF et ont reçu des réponses dans plusieurs optimisations et extensions du protocole de base, qui seront présentées dans ce qui suit.

4.2.1. Mobile IP avec encapsulation au retour

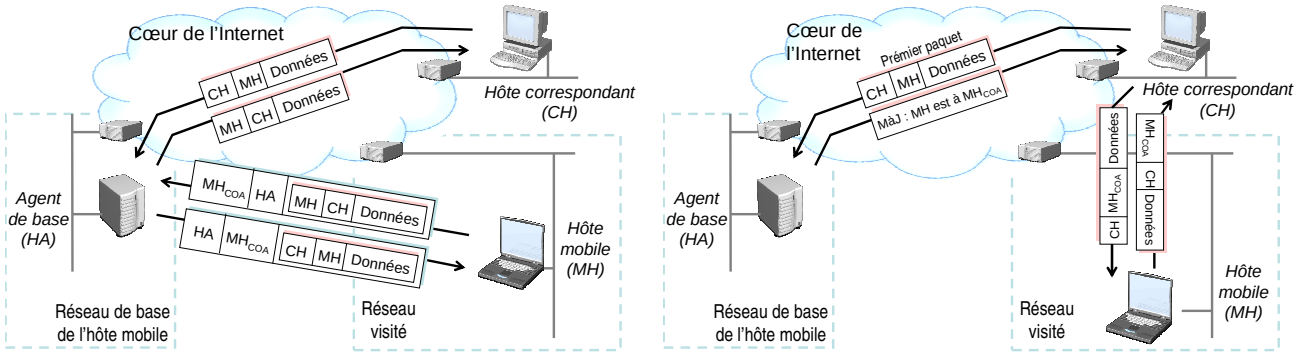
Un document assez récent[46] de l'IETF recommande que les routeurs de bordure des domaines administratives fassent du filtrage à l'entrée (*ingress filtering*) pour faciliter l'identification des attaques provenant de leur domaine et protéger ainsi le reste de l'Internet. Cette proposition, classifié BCP (*Best Current Practice*), est implémentée actuellement par la plupart des routeurs. Elle spécifie que les datagrammes sortant du domaine doivent avoir une adresse source topologiquement correcte, c'est-à-dire appartenant au domaine. Comme résultat, les paquets utilisés dans les attaques de type DoS (*Denial Of Service*) devraient permettre la localisation de l'attaqueur. Le problème vis-à-vis de Mobile IP est immédiat : dans un domaine visité qui applique le filtrage à l'entrée, l'hôte mobile ne peut pas utiliser son adresse de base pour envoyer des datagrammes à ses correspondants. Cela empêche toute opération de la version standard de Mobile IP dans un tel domaine.

D'une manière similaire, une autre recommandation IETF qui pose un problème pour Mobile IP est le filtrage à la sortie (*egress filtering*)[47]. Ce document propose que les routeurs de bordure doivent filtrer les paquets venant de l'extérieur mais qui ont une adresse

source appartenant au domaine. Ce filtrage empêche l'hôte mobile d'utiliser en Mobile IP son adresse de base pour envoyer des paquets vers son domaine d'origine.

Utiliser directement l'adresse temporaire comme adresse source dans les datagrammes envoyées par l'hôte mobile aux correspondants n'est pas une solution viable. En effet, les hôtes correspondantes ne connaissent que l'adresse de base de l'hôte mobile et leur couche transport va ignorer les datagrammes venues de l'adresse temporaire.

Pour permettre le fonctionnement de Mobile IP dans ces conditions, il est nécessaire d'utiliser l'encapsulation même pour les paquets envoyés par l'hôte mobile (figure 4.2a). Le datagramme originale est encapsulé dans un en-tête IP ayant comme adresse source l'adresse temporaire de l'hôte mobile (ou celle de l'agent visité, si c'est le cas). Puisque les hôtes correspondantes ne sont pas tenus d'implémenter Mobile IP et ne sauront probablement pas comment décapsuler les paquets, il nous faut une autre destination pour les datagrammes encapsulés. La seule à disposition est l'agent de base, qui connaît les deux adresses de l'hôte mobile. En conséquence, pour des domaines visités qui pratiquent le filtrage à l'entrée et pour les communications avec des hôtes du domaine d'origine pratiquant de filtrage à la sortie, cette extension[48] de Mobile IP propose que tous les datagrammes envoyés par l'hôte mobile doivent passer par l'agent de base. Évidemment, cette encapsulation sur le chemin de retour double les inefficacités de Mobile IP liées au routage indirect.



(a) Mobile IP avec encapsulation au retour

(b) Mobile IP avec optimisation de routage

FIG. 4.2.: Extensions de Mobile IP

4.2.2. Mobile IP avec optimisation de routage

L'idée de base de cette extension[49] de Mobile IP est que les hôtes correspondants d'un hôte mobile puissent être informés de l'adresse temporaire de celle-ci. Une fois que les hôtes correspondants apprennent cette adresse temporaire, elle encapsulent les datagrammes originales et les envoient directement à l'adresse temporaire de l'hôte mobile, sans utiliser les services de l'agent de base (voir la figure 4.2b).

Les mises à jour de l'adresse de l'hôte mobile doivent être notifiées d'une manière authentifiée aux machines correspondantes (d'une manière similaire aux enregistrements demandés à l'agent de base dans le protocole de base). Puisque cette authentification se base sur un secret commun partagé par l'expéditeur et le destinataire d'un message, les auteurs de cette extension proposent que les mises à jour soient notifiées aux machines correspondantes par l'agent de base. En effet, celui-ci est bien situé, car il est le premier à recevoir un paquet envoyé par un correspondant qui ne connaît pas l'adresse temporaire de l'hôte mobile. Les auteurs considèrent aussi qu'il est plus facile d'établir une relation de sécurité avec l'agent de base qu'avec chaque hôte mobile d'un domaine.

Quand l'hôte mobile se déplace d'un domaine visité à un autre, il doit informer son agent de base sur les machines avec lesquelles il est en train de communiquer. Si l'agent de base est éloigné de l'hôte mobile, le temps de transmission peut être important et les hôtes correspondants ne recevront pas immédiatement les mises à jour. Des datagrammes peuvent donc être adressés pendant cette période de temps à l'ancienne adresse temporaire de l'hôte mobile. Les auteurs de cette extension proposent que l'hôte mobile puisse informer son précèdent agent visité sur sa nouvelle adresse temporaire. Cela permet aux datagrammes déjà envoyés par les correspondants est reçues à l'ancienne adresse d'être retransmises vers le nouvel emplacement de l'hôte mobile.

Cette solution résout l'inconvénient du routage indirect par l'agent de base, potentiellement éloigné du chemin direct entre l'hôte mobile et son correspondant. En revanche, son désavantage est qu'elle implique des changements dans les hôtes correspondants. Ils doivent demander/recevoir de notifications de changements d'adresse et encapsuler/décapsuler les datagrammes échangés avec l'hôte mobile.

4.2.3. Mobile IP hiérarchique

Cette proposition vise à résoudre le même problème présenté précédemment : le temps d'un handoff au sens Mobile IP peut être potentiellement élevé quand la mobilité d'une hôte a lieu dans une zone qui est éloignée de son réseau d'origine.

Dans cette extension de Mobile IP, il y a plusieurs machines remplissant la fonction d'agent visité. Ceux-ci sont organisés dans une structure hiérarchique. Le premier dans la hiérarchie et le plus proche de l'agent de base s'appelle *agent visité racine*. De l'autre coté, l'agent le plus proche de l'hôte mobile s'appelle *agent visité local*.

Lorsqu'un hôte mobile arrive dans un domaine qui implémente l'extension Mobile IP hiérarchique[50], il envoie une demande d'enregistrement à son agent visité local, qui se trouve sur le même lien. Cet agent relaye le message d'enregistrement à l'agent suivant au niveau supérieur dans la structure hiérarchique. Le cycle continue jusqu'au moment ou le message passe de l'agent visité racine à l'agent de base. Les paquets adressés à l'hôte mobile sont encapsulés par l'agent de base ou par les hôtes correspondants et transmis à l'agent visité racine. Celui-ci décapsule les paquets, récupérer le datagramme originale, le re-encapsule et le retransmet à l'agent suivant de niveau inférieur. L'opération se répète jusqu'au moment ou le paquet est décapsulé par le dernier agent visité et remis à l'hôte

mobile.

Quand un handoff a lieu à l'intérieur du domaine, l'hôte mobile génère un nouveau message d'enregistrement et l'envoie à son nouvel agent local. Celui-ci relaie le message dans la structure hiérarchique. À un certain niveau (dans le cas le plus défavorable au niveau racine) ce message est traité par un agent visité qui détient déjà un enregistrement pour l'hôte mobile, mais pointant vers un agent de niveau inférieur différent. Cet agent visité est appelé *agent de croisement*, car il se trouve à la jonction entre l'ancien et le nouveau chemin entre l'agent de base et l'hôte mobile. L'agent de croisement change son enregistrement de l'hôte mobile pour qu'il reflète la nouvelle position de celle-ci. Il n'est plus nécessaire de relayer le message plus haut dans l'hiérarchie, car les enregistrements des agents de niveaux supérieures demeurent inchangées.

Même s'il peut paraître plus complexe et inefficace que le protocole de base Mobile IP à cause des opérations répétées d'encapsulation, cette extension du protocole permet d'optimiser le temps de transfert. Puisque l'agent visité de croisement se trouve plus proche du nœud mobile que l'agent de base, le délai de la mise à jour causée par le handoff est réduit significativement.

4.2.4. Mobile IPv6

Mobile IPv6[39] est basé sur les mêmes principes de base que son correspondant pour IPv4. Cependant, il comporte un nombre des améliorations, possibles grâce aux fonctionnalités supplémentaires présentes dans IPv6.

Également, puisqu'il n'y a que peu de systèmes qui utilisent le protocole IPv6, le protocole Mobile IPv6 bénéficie d'un avantage important, car il ne vise la compatibilité avec les machines existantes. Cet avantage majeur est utile surtout pour optimiser le routage ; rappelons que la difficulté rencontrée dans l'extension similaire de Mobile IPv4 était justement l'incompatibilité avec les hôtes correspondantes n'implémentant pas les mécanismes en cause. Un autre élément important est que la protection de mises à jour envoyées aux hôtes correspondantes par l'hôte mobile ne demande ni d'avoir établi une association de sécurité auparavant, ni l'existence d'une infrastructure d'authentification. À la place, une méthode appelée *return routability*[2] est utilisé pour s'assurer que la vraie hôte mobile envoie le message de mise à jour. On estime donc que l'optimisation de route sera utilisée à l'échelle globale, entre toutes les hôtes mobiles et leurs correspondants.

L'hôte mobile peut acquérir son adresse temporaire dans le domaine visité par le mécanisme standard d'auto-configuration de IPv6[17]. L'espace d'adressage de IPv6 est très large et on n'a plus besoin d'un agent visité pour représenter plusieurs hôtes mobiles par une seule adresse. Pour cela, on à supprimé les agents visités de l'architecture de IPv6, une différence très importante par rapport à la version IPv4 où leur présence était préférée. En revanche, on élimine la possibilité que les agents visités coopèrent pour minimiser la perte des paquets lors d'un handoff (voir la section 4.2.2). À la place, Mobile IPv6 se sert du fait qu'une

[2]Un mécanisme qui garantir l'acheminement d'un message de réponse au bon destinataire.

machine peut avoir plusieurs adresses IPv6 par interface. Les terminaux peuvent ainsi garder ouverte l'ancienne connexion et continuer à recevoir des paquets à cette adresse même après qu'il est configuré avec une nouvelle adresse.

Les correspondants d'un hôte mobile envoient les datagrammes en utilisant l'adresse temporaire de l'hôte mobile sur le réseau visité. L'adresse fixe est incluse dans le nouvel en-tête de routage IPv6. En sens inverse, l'hôte mobile utilise une autre fonctionnalité de IPv6, l'option de destination de type *home address*, pour s'identifier. Utiliser ces mécanismes IPv6 à la place de l'encapsulation réduit la surcharge observée dans Mobile IPv4, tout en permettant aux niveaux supérieures de ne voir que l'adresse fixe de mobile et donc continuer à fonctionner de manière transparente.

4.3. Protocoles pour la micro-mobilité

4.3.1. La micro-mobilité

Mobile IP présente un délai important lors des handoffs, à cause des mises à jour. En effet, le temps aller-retour peut être potentiellement important entre le domaine visité, le domaine d'origine, et les hôtes correspondants. Plusieurs études[51, 52] ont indiqué que la mobilité des terminaux est souvent localisée à l'intérieur d'un seul domaine. Pour ces cas, on a besoin d'une solution qui n'expose pas la mobilité locale d'un hôte aux autres hôtes à l'extérieur du domaine. Ceci élimine le trafic de signalisation dans le reste de l'Internet et en même temps évite la latence induite par cette signalisation à distance.

On définit la micro-mobilité et les protocoles qui s'intéressent par cette principale propriété : le fait que la mobilité d'une hôte a lieu à l'intérieur d'un domaine et reste invisible pour les hôtes externes. Une fois qu'un datagramme destiné à l'hôte mobile arrive à la bordure du domaine, le protocole de micro-mobilité doit assurer par des moyens et fonctions spécifiques la remise du datagramme à l'hôte mobile à l'intérieur du domaine. Les objectifs des protocoles de micro-mobilité sont d'assurer un handoff rapide avec peu de pertes de paquets, tout en préservant la transparence vis-à-vis des hôtes correspondantes et couches supérieures, pour garder ouvertes les connexions TCP actives.

4.3.2. Types de protocoles de micro-mobilité

Au niveau IP, plusieurs approches sont possibles pour assurer l'acheminement local des datagrammes destinés à l'hôte mobile. Nous les avons classifié en solutions basées sur un *proxy*, solutions utilisant des *routes d'hôte* et solutions utilisant le *multicast*.

Architectures basées sur un proxy

Dans ce type de solution, l'hôte mobile se présente aux hôtes correspondants avec une autre adresse que la sienne, celle d'un *proxy*. Les datagrammes destinés à cette adresse arrivent dans le domaine de mobilité et sont interceptés par ce nœud. Le rôle de proxy pourrait être joué par le routeur de bordure ou par un autre nœud spécial, avec une adresse IP stable, qui va représenter tous les hôtes mobiles du domaine.

Ensuite, la remise des paquets du proxy à l'hôte mobile se fait en utilisant une de ces deux techniques possibles : l'encapsulation au niveau 3 où bien la translation d'adresses (NAT).

Solutions utilisant des routes d'hôte

Une autre approche pour gérer la micro-mobilité à l'intérieur d'un domaine est d'utiliser des entrées contentant l'intégralité d'une adresse IP dans les tables de routage du domaine. L'hôte mobile peut garder alors la même adresse IP indifféremment de sous-réseau ou elle est attachée. Les datagrammes qui y sont destinées vont être acheminés pas à pas jusqu'à son emplacement actuel. Ce type de routage est similaire à celui des réseaux ad-hoc MANET (*Mobile Ad Hoc Network*) [53], où l'acheminement de paquets se fait en plusieurs étapes et s'adapte lorsque les hôtes de déplacent et la topologie du réseau change.

Quand l'hôte mobile se déplace et change de sous-réseau, il suffit qu'il envoie un message de notification pour la mise à jour des tables de routage dans les routeurs du domaine. Le désavantage par rapport aux solutions de type proxy est qu'ici les mises à jour doivent être faites dans tous les routeurs du domaine. Cependant, dans une variante optimisée, ces mises à jour peuvent s'arrêter au routeur qui est au croisement entre l'ancien et le nouveau chemin entre l'hôte mobile et le routeur de bordure.

Nous allons discuter plus en détail ce type de solution dans la section suivante, et on va l'illustrer par deux propositions : HAWAII et Cellular IP.

Solutions utilisant le multicast

Le *multicast* [54] est un mode d'adressage des datagrammes IP qui permet à une machine source d'envoyer le même datagramme à un groupe de plusieurs machines destination. L'application du multicast pour la mobilité est immédiate : on attribue une adresse de multicast à l'hôte mobile et on fait en sorte que les routeurs des sous-réseaux où il est connecté joignent le groupe multicast correspondant.

Mysore et Bharghavan ont proposé dans [55] un protocole dans lequel chaque hôte mobile a assigné une adresse fixe multicast de classe D qui sert d'identificateur d'hôte. La seule opération à réaliser reste alors de propager les mises à jour dans l'infrastructure des routeurs. Ces mises à jour prennent la forme d'un message de rattachement au groupe multicast correspondant à l'adresse de l'hôte mobile. Cette approche est proche de celle proposé par Helmy dans [56], qui utilise une adresse IP unicast pour identifier l'hôte mobile, mais l'ache-

minement des paquets à son routeur courant se fait par multicast. Dans les deux propositions, l'arbre de distribution multicast doit être recrée rapidement et efficacement à chaque déplacement de l'hôte mobile. Cette contrainte et aussi le fait que le nombre d'adresses multicast de classe D est limité réduise,t l'application de ce type de protocole à des domaines restreints.

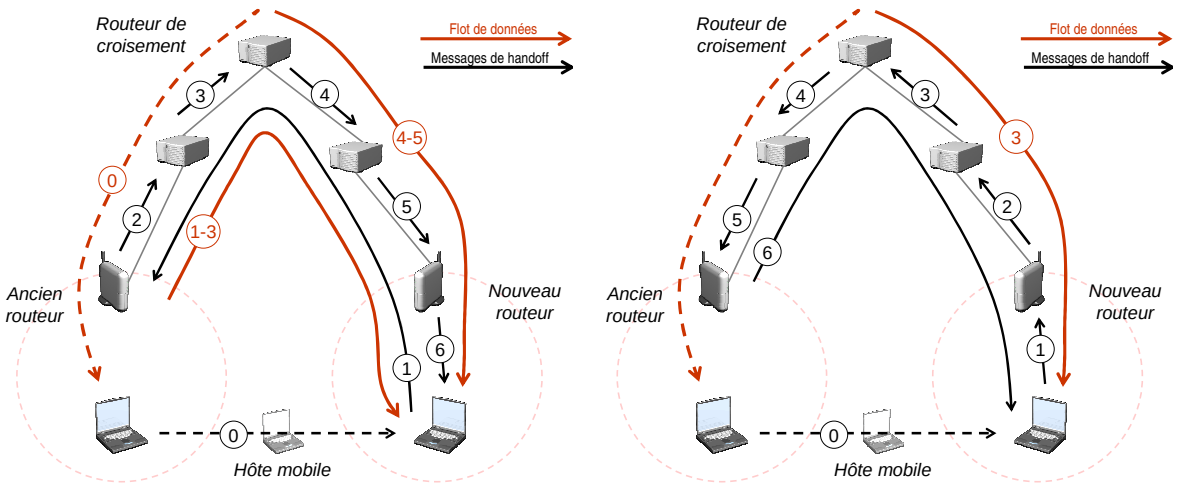
4.3.3. HAWAII

La première proposition d'un protocole de micro-mobilité présenté ici est HAWAII (*Handoff-Aware Wireless Access Internet Infrastructure*) [57]. Tous les routeurs du domaine de micro-mobilité doivent implémenter le protocole HAWAII. Parmi eux on identifie le routeur de bordure, appelé *routeur racine* du domaine. Chaque hôte mobile se voit attribué par configuration automatique une adresse fixe appartenant au domaine et qui sera utilisée dans la communication avec les hôtes correspondants. Les routeurs de l'extérieur du domaine acheminement les paquets destinés à cette adresse vers le routeur racine. L'adresse est ensuite utilisée dans les tables de routage des routeurs du domaine pour acheminer pas à pas les datagrammes vers le sous-réseau actuel de l'hôte mobile.

Des entrées dans les tables de routage sont crées à l'aide de messages de signalisation initiés par l'hôte mobile à sa première connexion dans le domaine. Le message que l'hôte mobile envoie lors de son entrée dans le domaine est relayé par chaque routeur du domaine sur le chemin direct entre l'hôte mobile et le routeur racine (voir la figure 4.3). Chaque routeur qui le reçoit établie ou met à jour une route spécifique pour l'hôte mobile. Cette route indique que les messages destinées à l'hôte mobile doivent être acheminées vers le routeur voisin dont le routeur courant a reçu le message de mise à jour. Lors d'un handoff de l'hôte mobile vers un nouveau sous-réseau, plusieurs schémas de propagation de la mise à jour sont possibles :

MSF (*Multiple Stream Forwarding*) et SSF (*Single Stream Forwarding*) sont deux schémas dans lesquels les datagrammes destinés à l'hôte mobile arrivent à son ancien routeur avant d'être redirigées vers le nouvel emplacement. Dans MSF, le message de mise à jour initié par l'hôte mobile se propage de l'ancien routeur vers le nouveau routeur. Le routeur situé au croisement des chemins entre le routeur racine du domaine et respectivement l'ancien et le nouveau routeur de l'hôte mobile est appelé routeur de croisement. Au moment où le message de signalisation arrive à ce routeur, on a pour une courte période deux flots de données (voir la figure 4.3a) : en plus du flot direct de données il y a aussi les paquets redirigées par les routeurs sur l'ancien chemin. Cette direction de propagation de la mise à jour peut produire des boucles temporaires dans le routage et une arrivée en désordre de paquets. L'autre schéma, SSF, évite les boucles temporaires et l'arrivée en désordre de paquets en envoyant la notification de mise à jour du nouveau routeur vers l'ancien. Ce schéma demande d'installer des tables de routage plus sophistiquées, qui prennent en compte dans la décision de routage l'interface sur laquelle le paquet est arrivé. Une description plus détaillée peut être trouvé dans une proposition[58] soumise à l'IETF.

Seule la direction de propagation de la mise à jour du nouveau routeur de l'hôte mobile vers l'ancien ne crée pas des boucles temporaires de routage. On l'utilise aussi dans les deux



(a) MSF (Multiple Stream Forwarding)

(b) UNF (Unicast Non-Forwarding)

FIG. 4.3.: Le schémas de handoff dans HAWAII

autres schémas : UNF (*Unicast Non-Forwarding*) et MNF (*Multicast Non-Forwarding*). UNF, illustré dans la figure 4.3b, est le plus simple des quatre schémas et mène à la situation où pendant un court intervalle des datagrammes sont envoyés à l'ancien routeur et perdus. Pour éviter cela, le schéma MNF spécifie que le routeur de croisement peut envoyer pendant ce temps les paquets sur les deux interfaces (technique appelée par les auteurs *dual-cast*).

4.3.4. Cellular IP

L'autre protocole de micro-mobilité qu'on détaille est Cellular IP[59]. Il adopte une approche similaire à celle de HAWAII, avec un domaine de mobilité locale séparé du reste de l'Internet par un routeur racine. Cependant, les deux protocoles diffèrent par le mode de mise à jour de tables de routage. Au lieu d'utiliser des messages explicites de signalisation, les nœuds de Cellular IP observent l'interface d'arrivée des datagrammes et rajoutent ou mettent à jour une entrée qui fait correspondre l'adresse IP source du datagramme à l'interface respective. Ainsi, tous les routeurs sur le chemin entre une hôte mobile et le routeur racine s'en servent des paquets de données envoyés par l'hôte mobile pour apprendre dans quelle direction celle-ci se trouve. L'hôte mobile a aussi la possibilité d'envoyer explicitement des messages de signalisation (par exemple des paquets vides destinés au routeur racine), à sa mise en route ou quand il y a pas de trafic de données. Le routeur racine envoie lui aussi des paquets de signalisation appelés *beacons* qui sont diffusés dans tous le domaine pour indiquer à tous les nœuds l'interface vers la sortie du domaine.

Cellular IP spécifie deux schémas de handoff : la première, appelée *hard handoff*, occasionne la perte d'un certain nombre des paquets. En échange, ce schéma est simple, rapide et implique peu de messages de signalisation. Le fonctionnement de ce schéma de handoff est illustré dans la figure 4.4. Une fois que l'hôte mobile est attaché à un nouveau sous-réseau,

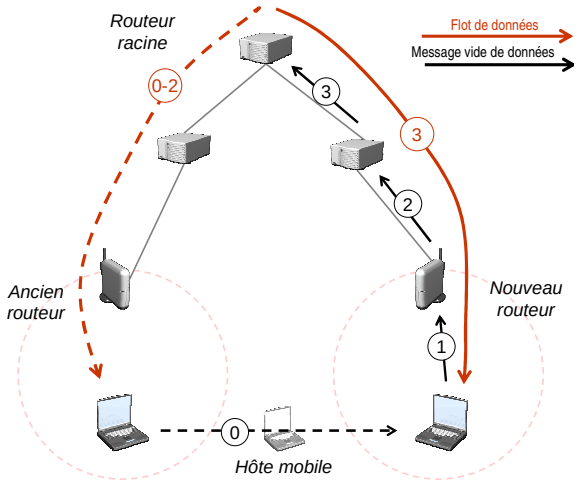


FIG. 4.4.: Le handoff dans Cellular IP

il envoie un message vide de données vers le routeur racine. En se propageant vers celui-ci, le paquet crée une nouvelle route pour l'hôte mobile en cause. Entre l'envoi du paquet par l'hôte mobile et son traitement par le routeur de croisement, il est possible que des paquets ont été déjà acheminée par celui-ci vers l'ancien emplacement de l'hôte mobile et seront perdus. Il faut aussi noter que par rapport à HAWAII, le paquet de mise à jour est envoyé à la destination de routeur racine, et non plus à l'ancien routeur, ce qui garanti toujours un routage optimal entre le routeur de bordure et les hôtes mobiles.

L'autre schéma de handoff est appelé *semi-soft handoff* et son but est de minimiser la perte de paquets. Comme dans le schéma de handoff MNF de HAWAII, le routeur de croisement utilise le *dual-cast* pour envoyer les paquets destinées a l'hôte mobile sur les deux chemins (ancien et nouveau) et minimiser ainsi les chances qu'il y ait de paquets perdus.

4.4. Niveau 3.5 : Identificateurs d'hôte

L'inefficacité des indirections dans le routage de datagrammes de Mobile IP, ainsi que l'incapacité de solutions de micro-mobilité de fonctionner à l'échelle globale sont les deux motivations qui ont conduit à cette classe de solutions. Leur point commun est l'introduction d'une nouvelle couche entre les protocoles IP et TCP : la couche *identificateur d'hôte*. Puisque ce niveau est situé entre les couches 3 et 4, il est appelé également *niveau 3.5*. Dans les solutions basées sur cette modèle, une connexion au niveau transport est établie entre deux hôtes, avec des identifiants qui ne changent pas. Par contre, les hôtes s'échangent des datagrammes à partir de leur points d'attachement, qui eux peuvent changer au cours d'une connexion.

Ce type de solution a comme but un routage direct et optimal de datagrammes dans l'Internet. La translation entre l'identificateur d'une machine et l'adresse IP du point d'attachement temporaire se fait directement sur les deux machines impliquées (voir la figure

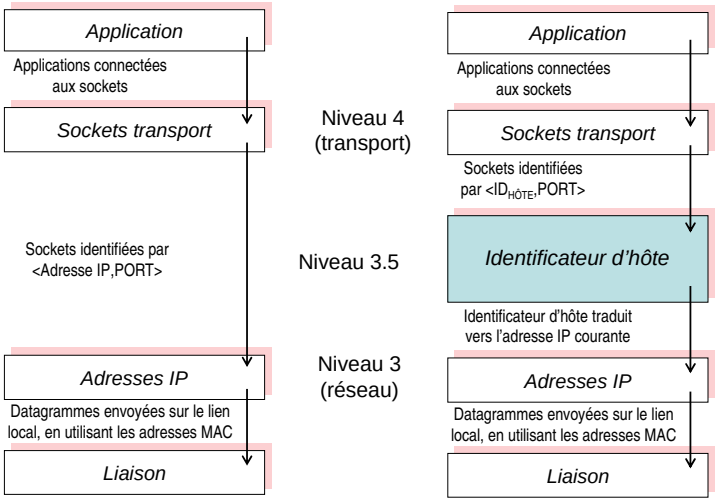


FIG. 4.5.: Le niveau 3.5 : identificateurs d'hôte

4.5). Ceci est conforme au principe *end-to-end*[3], principe de base dans la construction et fonctionnement de l'Internet, énoncé pour la première fois par J. Saltzer[60]. Dans les solutions basés sur la couche 3.5, seules les deux machines au bouts d'une connexion sont impliqués dans l'envoi de datagrammes de données. Les seuls échanges qui impliquent une machine tierce sont la mise à jour et la consultation d'un répertoire externe de translation entre les identifiants d'hôte et les adresses IP, ce qui pourrait être nécessaire à l'initiation ou à la reprise d'une connexion.

Dans le reste de cette section nous présentons quatre solutions qui proposent un niveau 3.5 dans leur architecture (LIN6, VIP, VNAT et HIP). Les particularités qui différentient les différentes propositions sont liées à l'espace de noms choisi pour les identificateurs d'hôte et à la translation entre un identificateur et une adresse IP.

4.4.1. LINA et LIN6

Une des propositions qui s'inscrit dans cette classe vient du projet japonais WIDE (*Widely Integrated Distributed Environment*)[61]. Son nom est LINA (*Location Independent Network Architecture*)[62] et son implémentation LIN6[63] a été faite en utilisant des propriétés spécifiques à l'architecture d'adressage de l'IPv6.

Dans LINA, un hôte mobile possède un ou plusieurs identificateurs qui y sont assignés par une autorité administrative, qui veille à leur unicité. Une fois connecté au réseau, l'hôte est configuré avec une ou plusieurs adresses IP pour chacune des interfaces réseau. Un correspondant peut spécifier directement l'identificateur de l'hôte mobile pour initier une connexion à celui-ci. Néanmoins, il peut utiliser une adresse IP à la place de l'identificateur d'hôte s'il veut se connecter à un certain point d'attachement sans être intéressé de l'identité

[3] En français *de bout en bout*

de l'hôte qu'y se trouve.

La particularité de LINA est que l'identifcateur d'hôte est inséré dans l'adresse réseau des datagrammes envoyés, afin d'éviter le rajout d'un en-tête supplémentaire. En IPv6 il est possible d'implémenter ceci en utilisant certaines propriétés de l'architecture d'adressage. Les adresses IPv6 doivent respecter le modèle AGUA (*Aggregatable Global Unicast Address*)[64], dans lequel les premières 64 bits d'une adresse spécifient le préfix du sous-réseau et les 64 bits suivants représentent l'identificateur d'une interface réseau. À son tour, cet identificateur, unique dans un sous-réseau, doit respecter le format EUI-64 (*Extended Unique Identifier on 64 bits*)[65]. La première partie de 24 bits indique le OUI (*Organizationally Unique Identifier*) (un numéro attribué par l'IEEE à chaque fabricant de cartes réseau) et le reste de 40 bits reste la partie configurable par la machine.

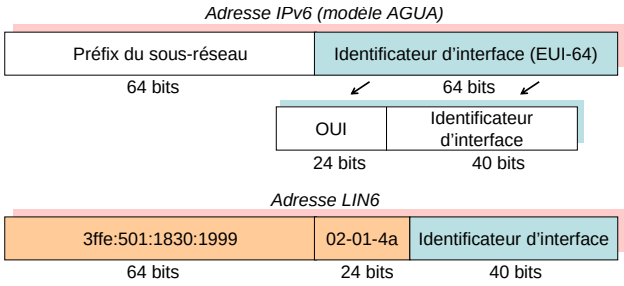


FIG. 4.6.: La structure des adresses LIN6

LIN6 propose d'utiliser cet espace de 40 bits et une valeur spéciale `02-01-4a` dans la partie OUI pour attribuer des identificateurs d'hôte aux nœuds mobiles (voir la figure 4.6). En rajoutant comme préfixe la valeur spéciale `3ffe:501:1830:1999` on obtient l'*identificateur généralisé LIN6*, qui a une longueur de 128 bits et peut être utilisé à la place des adresses IP. À part l'identificateur généralisé, chaque nœud mobile est configure avec une ou plusieurs *adresses LIN6* (appelés par les auteurs *embedded locators*) qui en plus de l'identificateur d'hôte sont complétées cette fois par le préfixe réel du sous-réseau ou le nœud mobile se trouve. Les adresses IPv6, ainsi formées, seront utilisées pour le routage des datagrammes vers la position actuelle de l'hôte mobile.

L'association entre l'identificateur d'hôte d'un nœud et les adresses LIN6 de ses points d'attachement se fait par l'intermédiaire d'un ou plusieurs répertoires, nommés MA (*Mapping Agents*) dans l'architecture. Jugeant que le DNS n'est pas approprié pour fournir un support efficace pour les mises à jour, LINA propose d'utiliser à ce but un ou plusieurs serveurs où sera gardé l'association entre l'identificateur et les adresses LIN6 d'un nœud. Pour apprendre quel est le MA d'un nœud mobile, un correspondant doit effectuer une résolution DNS inverse de l'identifiant généralisé. Les MA sont mis à jour par les nœuds mobiles LIN6, chaque fois qu'ils changent d'adresse. Ils doivent également notifier leurs correspondants sur le changement d'adresse, par un mécanisme similaire aux *Binding Updates* de Mobile IPv6.

Chaque fois qu'une application s'exécutant sur un nœud compatible LIN6 initie une connexion, le module LIN6 intercepte la requête et détermine si la destination est un identificateur généralisé ou une simple adresse IP, en se basant sur le préfixe du sous-réseau. Si le préfixe

utilisé est celui de LIN6, une requête DNS est faite pour trouver le MA, l'interroger et translater l'identificateur généralisé en une adresse LIN6. Le datagramme envoyé sur le réseau contient cette adresse LIN6 dans le champ destination, ainsi que l'adresse LIN6 du nœud local dans le champ source. À destination, le même module exécute les opérations en sens inverse : il détermine si l'adresse IP source est une adresse LIN6, cette fois-ci en regardant la partie OUI de l'adresse. De cette façon, une connexion TCP entre deux applications est toujours réalisée entre les identificateurs généralisés des deux nœuds et donc insensible aux modifications de leurs adresses LIN6.

4.4.2. VIP

Virtual IP (VIP) [66] a comme caractéristique principale l'utilisation d'adresses IP virtuelles comme identificateurs d'hôtes. L'idée de base est simple : utiliser les noms d'hôte dans leur forme complète FQDN[4] pour identifier de manière unique les nœuds. Toutefois, puisque les applications ouvrent des connexions entre deux adresses IP, VIP doit traduire les FQDN dans une suite de 32 bits, le format d'une adresse IP. À leur tour, ces suites de bits qu'on appelle *IP virtuels* sont traduits dans les adresses IP réelles qui correspondent aux points d'attachement courants des deux machines. Si les adresses IP changent suite à un déplacement d'une de deux machines, VIP garde la cohérence entre le nom d'hôte et l'adresse IP courante par des mises à jour dynamiques et sécurisées DNS.

Nous avons montre dans la section 3.3 que les noms d'hôte sont structurés hiérarchiquement. Les organisations qui administrent les différents espaces de noms dans le système DNS assurent l'unicité de chaque nom d'hôte. Les adresses IP virtuelles devraient elle aussi être uniques, et en plus être différentes des adresses IP réelles, pour ne pas entrer en conflit avec des machines qui n'utilisent pas VIP. Pour ces raisons, chaque nœud choisit aléatoirement un *IP virtuel* la classe réservée E[24] des adresses IP, qui n'est pas utilisé actuellement dans l'Internet. En échange, le nombre limité de 2^{28} adresses possibles n'offre pas une forte garantie de non-collision entre des adresses IP virtuelles construites aléatoirement. Pour résoudre ce problème, VIP inclut une phase de négociation des *IP virtuels* à l'initiation de communication. Cette phase permet d'avoir des adresses IP virtuelles uniques localement[5].

VIP fonctionne par l'interception des requêtes de résolution de noms d'hôte faites par les applications qui initient une connexion. Ces requêtes sont redirigées vers un daemon DNS local (voir la figure 4.7). Celui-ci continue la résolution du nom en interrogeant le serveur DNS réel pour obtenir l'adresse IP courante de la machine destination. En parallèle, une autre requête DNS est envoyée, cette fois-ci pour le nom d'hôte précédé d'un préfixe spécial, pour trouver l'IP virtuel proposée par la machine destination. Si la réponse du serveur DNS informe que ce nom spécial n'existe pas, cela signifie que la machine destination n'est pas compatible VIP. Dans ce cas, la connexion est établie directement entre les adresses phy-

[4] *Fully-Qualified Domain Name*

[5] Par l'unicité locale, on comprend que les deux machines n'ont pas des connexions ouvertes avec des autres machines qui utilisent les mêmes adresses IP virtuelles. Par contre, une même adresse IP virtuelle peut être utilisée par une machine tierce, mais qui ne communiquent pas avec aucune de nos deux machines (il y a pas d'unicité globale).

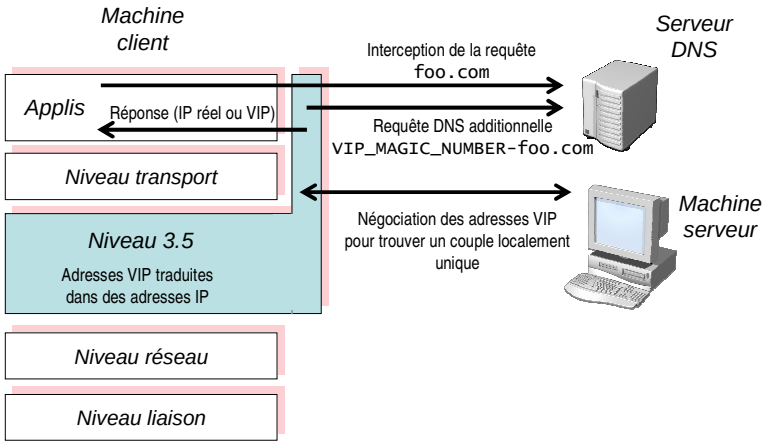


FIG. 4.7.: La négociation des adresses IP virtuelles dans VIP

siques des deux machines et ne pourra pas survivre à un éventuel changement d'adresse. Sinon, l'IP virtuel reçu comme réponse est retourné à l'application. Le daemon notifie à son tour son correspondant sur son propre IP virtuel proposé. Une négociation peut avoir lieu ensuite et d'autres IP virtuels proposés si une de deux machines refuse l'IP virtuel proposé par son correspondant (parce qu'il n'est pas unique localement).

Au moment où une de deux machines se déplace et son adresse IP réelle change, elle met à jour son serveur DNS. On suppose que chaque hôte est en mesure d'effectuer de manière dynamique et sécurisée la mise à jour des informations la concernant. Ensuite, elle envoie un message d'invalidation à l'ensemble de machines avec lesquelles elle a des connexions ouvertes. Ce message d'invalidation informe ses destinataires sur le changement d'adresse et demande de refaire la résolution de nom pour apprendre la nouvelle adresse IP.

Dans l'implémentation de leur prototype VIP, les auteurs ont utilisé une ou plusieurs interfaces réseau virtuelles, qui sont configurées avec les adresses VIP locales. Dans la table locale de routage, tous les paquets envoyés vers une adresse de type VIP sont relayés vers ces interfaces. Les datagrammes sont traités ensuite par un service d'encapsulation qui rajoutent un nouveau entête IP contenant les adresses IP réelles correspondantes aux deux machines. Le paquet repasse une nouvelle fois par la table locale de routage et est envoyé cette fois sur l'interface réseau réelle. Une suite similaire d'opérations a lieu sur la machine destination, où les datagrammes sont décapsulés et ensuite redirigés à l'entrée de l'interface virtuelle correspondant à l'adresse VIP destination. Ceci permet aux applications d'avoir des connexions stables, qui sont ouvertes et utilisent les IP virtuels invariables.

4.4.3. VNAT

VNAT[67] propose aussi d'utiliser des adresses IP virtuelles pour identifier les machines. Son nom (*Virtual NAT*) vient du fait qu'il utilise le mécanisme standard NAT (*Network Address Translation*)[68] pour translater les adresses IP virtuelles dans des adresses IP réelles.

Une autre difference par rapport à VIP est que la protocole VNAT ne se base pas sur une structure externe pour faire la translation identifiant – adresse IP. En échange, VNAT fournit des fonctions pour établir une association de sécurité entre les deux machines, qui est ensuite utilisée pour notifier d'une manière authentifiée les éventuelles mises à jour des adresses IP.

Quoique les adresses IP virtuelles puissent être choisies au hasard, les auteurs précisent qu'un choix soucieux peut apporter des avantages importants. Les auteurs considèrent que les deux extrémités d'une connexion doivent partager le même couple d'adresses virtuelles, donc les deux hôtes doivent s'informer réciproquement sur leur adresse IP virtuelle. Les messages de notification doivent être échangés en début de chaque connexion et peuvent être très pénalisants, essentiellement dans le cas des connexions de courte durée qui ne seront jamais transférées vers d'autres points d'attachement. Ce délai peut être évité en utilisant tout simplement les adresses IP réelles initiales comme identificateurs virtuelles des deux machines. Cela évite aussi la surcharge de la translation d'adresses avant qu'un changement d'adresse ait lieu.

VNAT ne contient pas un mécanisme qui permet aux deux hôtes de se retrouver en cas de déplacement simultané. Aussi, il n'y a pas de mécanisme de résolution de conflit d'adresses, cas où une adresse IP est réutilisée et deux connexions différentes partageant une extrémité commune entrent en conflit[6]. Les auteurs ne proposent pas un mécanisme de négociation d'une autre adresse virtuelle, et suggèrent tout simplement d'interrompre une des deux connexions.

4.4.4. HIP

Une approche intéressante a été proposé dernièrement par R. Moskowitz dans [69] et étudié dans le groupe de travail HIP (*Host identity Protocol*) de l'IETF . Les extensions proposées[70] et le protocole associé[71] peuvent être utilisés pour la gestion de la mobilité des hôtes. HIP (*Host Identity Protocol*) fournit un identificateur d'hôte pour séparer l'usage des adresses IP dans la pile de protocoles TCP/IP. Cet identificateur est appelé HI(*Host Identity*) et prend la forme d'une clé publique. Cette clé peut être générée automatiquement par une machine ou, pour mieux assurer son unicité, elle peut être attribuée par une organisation.

Pour être compatible avec les implémentations actuelles des protocoles TCP/IP, les formats utilisés pour identifier les machines devraient être sous la forme de 32 ou 128 bits (la longueur des adresses IP dans IPv4 et respectivement IPv6). Pour cela, HIP utilise deux autres formes de l'identificateur : le LSI(*Local Scope Identity*) sur 32 bits et le HIT (*Host Identity Tag*) sur 128 bits, qui peuvent être dérivés de la clé publique en appliquant des fonctions *hash*. Le niveau virtuel HIP les traduit ensuite dans les adresses IPv4 ou IPv6 réelles des machines par une translation d'adresses NAT.

Le caractère cryptographique des identificateurs HIP est le point-clé de l'architecture.

[6] Si on prend en compte la présence des numéros de port dans l'identification d'une connexion TCP, il est peu probable qu'un tel conflit ait lieu, mais pas impossible.

L'utilisation d'un protocole simple d'authentification par clé publique est utilisée au début d'une connexion entre deux machines HIP pour assurer la sécurité des messages échangés ultérieurement. HIP est conçu pour être intégré avec IPSec (*Internet Protocol Security*)[72] et l'échange initial crée également des associations de sécurité (SA) relatives aux clés publiques utilisées. L'avantage majeur d'utiliser IPSec est que les données UDP ou TCP sont encapsulés dans un en-tête IPSec. Celui-ci est ensuite utilisé comme indicateur de la clé publique et précise au destinataire la hôte source du paquet.

5. Solutions aux niveaux supérieurs

Les solutions qui gèrent la mobilité d'hôtes au-dessus du niveau IP ont l'avantage de modifier que les hôtes finales impliqués dans une connexion, indépendamment de l'infrastructure IP qui reste inchangée. Cet argument est aussi celui des solutions opérant au niveau 3.5 présentées dans la section précédente. Un des arguments pour des solutions au niveau supérieurs de la pile de protocoles Internet est que les réseaux sur lesquelles les machines mobiles se connectent peuvent avoir des caractéristiques différentes au niveau du débit, de la latence et des taux d'erreur. Les adeptes de ces solutions estiment que les niveaux supérieurs doivent être informés des nouvelles caractéristiques du réseau, pour changer les paramètres des connexions TCP ouvertes vis-à-vis de la stratégie des retransmission ou de contrôle de congestion ou tout simplement recommencer une nouvelle connexion TCP qui s'adapte aux nouvelles conditions.

5.1. Solutions au niveau transport

La plupart des solutions qui agissent au niveau transport concernent le protocole TCP. Une première classe de propositions lui ajoutent des extensions pour permettre à chacune des deux hôtes à l'extrémité d'une connexion de changer d'adresse IP. Puisque les systèmes d'exploitation implémentent souvent les fonctions TCP dans leur noyau, les modifications au niveau transport sont difficilement applicables à toutes les machines. À cause de ce fait, une deuxième classe de propositions rajoute un proxy dans la connexion TCP qui sépare la communication en deux connexions TCP indépendantes l'une de l'autre. L'hôte mobile et le proxy coopéreront pour continuer une connexion TCP existante à partir d'une adresse IP différente, par contre l'hôte correspondante ne sera pas concernée par ce changement. Finalement, nous présentons à la fin de cette section un nouveau protocole de transport proposé pour remplacer TCP et gérer mieux la mobilité des hôtes.

Quant à UDP, l'autre protocole de niveau transport, il n'a pas reçu une attention considérable par rapport à la mobilité d'hôtes. UDP est utilisé généralement par des applications multimédia (diffusion de flots audio et vidéo) pour lesquelles le temps requis par TCP pour gérer les retransmissions et l'ordonnancement des paquets n'est pas tolérable. Le plus souvent, ces applications qui utilisent UDP fournissent elles-mêmes des fonctions spécifiques pour la mobilité. Il est aussi employé dans des applications comme DNS, dont la durée de vie des communications par UDP est souvent réduite à une séquence requête - réponse entre un client et un serveur. Ceci minimise la possibilité qu'un changement d'adresse ait lieu dans ce temps et même si c'est le cas, un simple renvoi des messages résout le problème.

5.1.1. Extensions pour TCP

E-TCP

Une première extension du TCP a été introduite par Huitema et s'appelle E-TCP (*Extended TCP*)[73]. Les deux hôtes aux extrémités d'une connexion incluent dans chaque segment TCP envoyé un identificateur unique du flot TCP, appelé *PCB-ID* et qui a une taille de 32 bits.

À l'initiation d'une connexion, les deux hôtes rajoutent leurs PCB-ID locaux aux segments *SYN* envoyés. Ensuite, les hôtes peuvent s'échanger des paquets E-TCP, dans lesquels la paire de numéros de port est remplacée par leur PCB-ID local. C'est le PCB-ID qui permet ensuite à l'hôte destinataire de regrouper tous les paquets du même flot, même s'ils sont issus d'adresses IP source différentes.

L'auteur reconnaît le fait que sa proposition introduit une brèche dans la sécurité de la connexion, car quelqu'un d'autre pourrait observer le réseau et s'introduire dans la communication en fabriquant des paquets avec le même PCB-ID. Cependant, il soutient que cette brèche est inhérente dans le contexte de la mobilite et explique qu'on peut obtenir une sécurité accrue en utilisant une solution de type IPSec.

Migrate TCP

Récemment, le projet *Migrate* de MIT a proposé extension similaire de TCP pour identifier les extrémités d'une connexion. Migrate TCP[74] utilise les segments *SYN* échangés à l'initiation d'une connexion pour y rajouter une option appelé *Migrate-Permitted* et négocier un identificateur unique de la connexion. Cette négociation est sécurisé par une clé secrète, partagée par le protocole Diffie-Hellman[75]. L'identificateur de la connexion est formé par les 64 bits les plus représentatifs du hash SHA-1 (*Secure Hash Algorithm*)[76] calculé sur les numéros de séquence initiaux et sur la clé secrète partagée.

Suite à un déplacement qui implique un changement d'adresse, l'hôte mobile peut reprendre une connexion en cours en envoyant un segment SYN qui contient l'identificateur négocié auparavant (voir la figure 5.1). L'hôte destinataire interprète ce segment comme une demande de reconnexion et l'ancienne connexion est reprise.

Proxy TCP : MSOCKS et I-TCP

MSOCKS[77] est une proposition basée sur une architecture proxy et utilise une technique appelé *TCP-Splice*[78] pour diviser une connexion TCP en deux.

Le protocole MSOCKS est construit autour du protocole SOCKS[79] pour la traversée de pare-feux. MSOCKS y rajoute un identificateur logique qui permet de garder la trace des connexions ouvertes entre l'hôte mobile et le proxy. Le proxy MSOCKS construit un nouvel

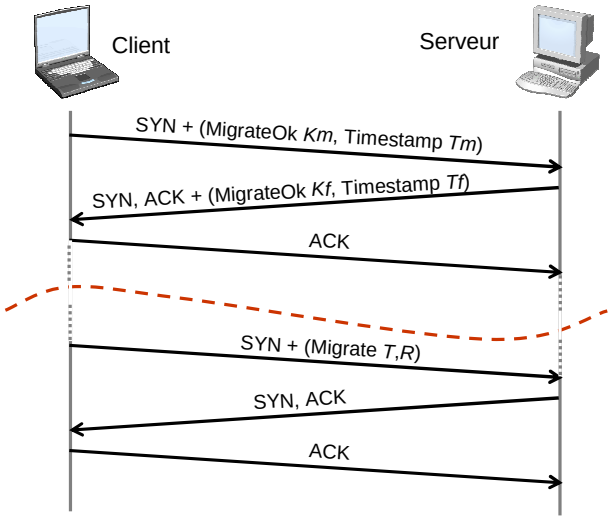


FIG. 5.1.: Le fonctionnement de Migrate TCP

identificateur chaque fois qu'il reçoit une requête *BIND* ou *REQUEST*. L'identificateur est envoyé à l'hôte dans la réponse qui confirme la réussite de la connexion. Lorsque l'hôte change d'adresse et veut reprendre une connexion déjà ouverte, il envoie une requête *RECONNECT* au proxy MSOCKS en spécifiant l'identificateur de la connexion respective. Lors de la réception de cette requête, le proxy détache sa connexion à l'hôte mobile de l'ancienne adresse de celle-ci et la rattache à la nouvelle adresse.

La technique TCP-Slice utilisé par MSOCKS pour couper en deux une connexion TCP permet de réaliser la reconnexion transparente de l'hôte hôte mobile à ses correspondants. TCP-Slice prend soin de retransmettre dans le bon ordre les datagrammes véhiculés entre les deux connexions et garantit la sémantique d'une connexion TCP habituelle.

Une autre proposition similaire à MSOCKS est I-TCP[80], où le rôle du proxy TCP est joué par une entité appelée MSR (*mobile support router*). Cependant, la façon dont le proxy TCP redirige les données entre les deux connexions TCP est simpliste et ne respecte pas la sémantique TCP originale. Par exemple, le proxy envoie un acquittement à l'hôte correspondante sans s'assurer que l'hôte mobile ait effectivement reçu le segment de données respectif.

5.1.2. Le protocole SCTP

Nous achevons cette section avec la présentation de SCTP (*Stream Control Transmission Protocol*), un troisième[1] protocole de niveau transport qui a été proposé récemment par l'IETF. À la base, SCTP est un protocole de transport orienté connexion qui garanti la réception de paquets de données. Il a eu comme but principal d'apporter des extensions liées

[1] SCTP se veut l'équivalent de IPv6 au niveau transport et on prévoit qu'à terme il va remplacer les protocoles TCP et UDP.

au multi-accès et multi-flot qui manquaient au TCP. Actuellement, une nouvelle extension est en cours de développement pour ajouter un nouveau mode de transport plus simple qui n'offre pas des garanties de livraison. Ainsi, cette extension fera de SCTP un possible remplaçant pour l'autre protocole actuel de transport - UDP.

SCTP fournit un moyen pour que chaque hôte puisse envoyer à son correspondant la liste de ses adresses IP possibles. Ceci est fait au début de la connexion et ensuite il est possible d'envoyer et de recevoir des paquets entre n'importe quelles adresses faisant partie des listes échangées par les deux extrémités de la connexion. Une seule paire d'adresses définit le chemin principal utilisé à un instant donné, et toutes les autres forment des chemins de réserve.

Cette fonctionnalité de *multi-accès* offerte par SCTP permet d'envisager une solution pour gérer la mobilité, puisqu'il permet aux deux hôtes d'utiliser plus d'une adresse IP. Néanmoins, dans le protocole de base décrit dans [81] l'ensemble des adresses IP pouvant être utilisés ne peut être spécifié qu'en début de la connexion. Une extension de SCTP a été proposée pour qu'un hôte puisse introduire une nouvelle adresse en cours de connexion, par exemple après un changement d'adresse. L'extension qui permet la mise à jour dynamique de la liste d'adresses est *ADD-IP* et le protocole SCTP enrichi par cette extension est connu sous le nom mSCTP (*Mobile SCTP*)[82].

ADD-IP ajoute deux nouveaux types de messages de contrôle : *ASCONF* et *ASCONF-ACK*. Ces messages permettent d'ajouter, d'enlever ou de changer une adresse dans l'ensemble des adresses IP des deux extrémités de la connexion. Ensuite, on peut notifier à l'hôte correspondant un changement de l'adresse utilisé dans le chemin principal, pour utiliser une des adresses rajoutées par ADD-IP.

5.2. Le niveau session

Nous avons introduit dans la section 2.1 le concept de session - une association de longue durée entre deux applications qui inclut plusieurs connexions réseau, simultanées ou successives. Dans la suite de protocoles TCP/IP il n'y a pas un niveau session explicit qui se charge de l'établissement d'une session entre deux points en regroupant une ou plusieurs connexions qui partagent un contexte commun. En conséquence, certaines applications ont implémenté elles-mêmes le concept session. Un bon exemple est celui des applications web qui utilisent des *cookies* dans les requêtes et les réponses HTTP (*HyperText Transfer Protocol*)[83] pour permettre à une session de continuer en dépit des terminaison de connexions passageres. Un autre exemple est celui des applications de partage de fichiers P2P, qui reprennent le transfert d'un fichier du point où celui-ci a été interrompu.

Dans cette section nous discutons plusieurs propositions basées sur un même concept : un niveau session virtuel interposé entre le niveau transport et les applications. Ce niveau se charge de l'établissement d'une ou plusieurs connexions, de leur migration et de leur continuation si pour une raison quelconque elles sont interrompues. Ces actions peuvent rester transparentes aux applications pour permettre la compatibilité avec les logiciels existants.

5.2.1. La couche session Migrate

Le projet Migrate de MIT est un des premiers[2] qui introduit le niveau session. Il fournit une interface API aux applications pour décrire les points finaux de la session par des identificateurs uniques et stables, différents des adresses IP des points d'attachement où les deux machines se trouvent. Ensuite, les applications fournissent la composition de la session : une ou plusieurs connexions entre les deux points. Un mode de fonctionnement transparent aux application est aussi possible, dans lequel chaque connexion ouverte par une application est considérée comme une session. Dans ce cas, les identificateurs des points finaux assumés par défaut sont les noms d'hôte correspondants aux adresses IP utilisées dans la connexion.

L'API de Migrate permet aux applications d'utiliser un espace de noms et un système de résolution de noms quelconque qui n'est pas restreint seulement aux noms d'hôte et au DNS. Ce mécanisme est séparé du déroulement ultérieur de la session. L'application spécifie les noms et le système de résolution désirés et le niveau session se charge ensuite de la continuation des connexions ouvertes entre les deux points.

Au début de la session, il y a une phase de négociation qui tente de créer un canal de contrôle entre les deux point distants. Si les deux point distants sont compatibles Migrate et la création du canal de contrôle réussit, un échange de clés cryptographiques a lieu. Ces clés seront ensuite utilisées pour authentifier les mises à jour échangées ultérieurement.

Suite à un changement d'adresse, une hôte doit informer son correspondant en lui envoyant un message de mise à jour. La réception de ce message peut s'avérer impossible dans le cas où l'autre point a changé d'adresse IP en même temps. Dans ce cas les deux hôtes doivent refaire la résolution des noms, en utilisant les identificateurs et le système de résolution spécifiés au début par l'application. Si le contact réciproque ne réussit toujours pas, Migrate suppose qu'une interruption plus longue a lieu et suspend la connexion. On sauvegarde le contexte de la session et on libère une partie des ressources système et réseau, ce qui pourrait être utile pour les autres connexions.

L'implémentation de Migrate est réalisée par une couche intermédiaire interposée entre les applications et le système d'exploitation. Son opération est illustrée dans la figure 5.2. Migrate intercepte la demande d'ouverture d'une socket. Au lieu de connecter directement l'application à la socket réseau, Migrate ouvre une paire de deux sockets. L'une est virtuelle et connectée à l'application et l'autre est une socket réseau habituelle connectée au point distant spécifié au départ par l'application. Les données arrivées sur chacune de deux sockets sont redirigées vers sa paire. Ce mécanisme d'indirection permet de découpler au besoin la socket connectée à l'application de la socket réseau connectée à la destination. Si l'hôte correspondante change d'adresse IP, il suffit de créer une nouvelle socket réseau qui sera connectée à la nouvelle destination réseau. Cette nouvelle socket est couplée à la socket application remplaçant la socket réseau de l'ancienne destination. Une suite similaire d'opérations devra être mise en place sur l'hôte destinataire.

Le problème majeur dans l'implémentation de ce niveau d'indirection par sockets est lié

[2]Un autre projet qui propose un niveau session similaire s'appelle *rocks (Reliable Sockets)* et a été développé en même temps à l'Université de Wisconsin.

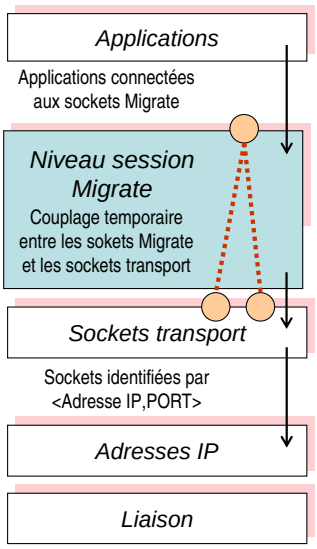


FIG. 5.2.: Le niveau session de Migrate

aux tampons des sockets réseau. Ces tampons se trouvent généralement implantés dans le noyau de systèmes d'exploitation. Les tampons peuvent cacher des données qui soit n'ont pas été envoyées à l'hôte distante, soit n'ont pas été consommées par l'application. La difficulté vient du fait qu'on a pas accès à l'état de ces tampons et donc on ne peut pas savoir si l'information se trouve ou pas dans les tampons des sockets. Pour éviter le risque de perdre les données qui sont restées dans le tampon lorsque Migrate remplace la socket réseau, le niveau d'indirection rajoute des tampons supplémentaires d'envoi et de réception (voir la figure 5.3). Après un remplacement d'une socket réseau, il suffit que les deux hôtes s'échangent sur le canal de contrôle les numéros des derniers octets reçus. Ensuite, la transmission peut recommencer à partir de ce point précis.

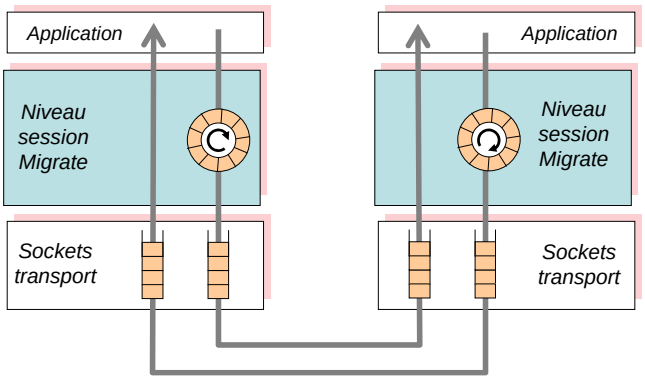


FIG. 5.3.: Le tampons supplementaires des sockets Migrate

Le cas des « connexions » UDP est plus simple de ce point de vue, puisque UDP ne garanti pas la réception des données par l'hôte correspondant. Il n'est pas nécessaire dans

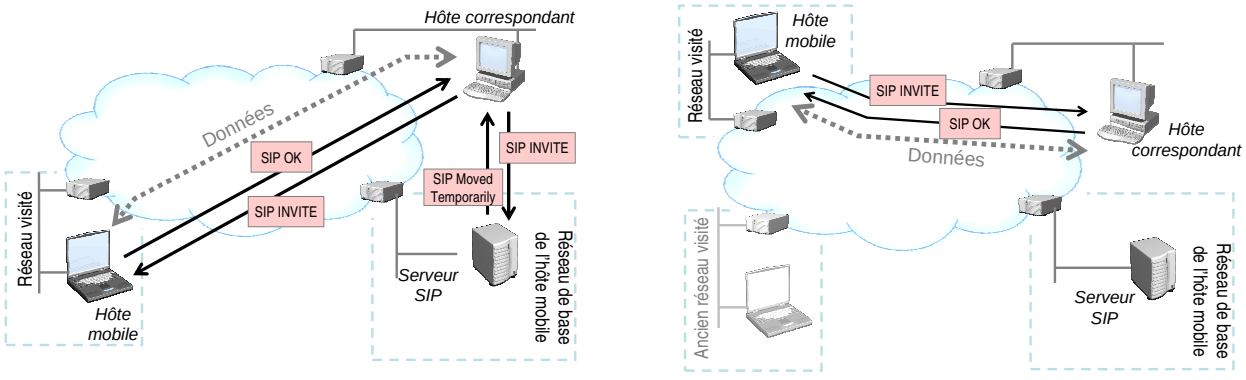
ce cas d'utiliser des tampons supplémentaires, car on suppose que les applications utilisant UDP ont leur propres stratégies de retransmission lors des pertes de données.

5.2.2. Le protocole SIP

SIP (*Session Initiation Protocol*)[84] est un protocole qui fournit une solution simple l'établissement d'une session multimédia entre deux utilisateurs. Il comporte également un support pour la mobilité de personnes et des machines[85], car l'établissement de la session inclut la localisation des utilisateurs et implicitement de leurs terminaux.

Dans l'architecture SIP, chaque utilisateur fait partie d'un domaine d'origine. Un serveur SIP présent dans chaque domaine contient des informations mises à jour sur l'emplacement actuel de chaque utilisateur du domaine. La partie la plus simple des mécanismes de mobilité fournis par SIP est la localisation de l'utilisateur par ses correspondants, appelée *pre-call mobility*. Chaque fois qu'une personne se déplace (avec son terminal ou en le changeant), le serveur SIP du domaine d'origine est informé sur l'adresse IP où elle peut être jointe. Les correspondants utilisent un nouveau type d'enregistrement DNS apprendre l'adresse du serveur SIP d'un certain domaine. Ensuite, les correspondants contactent ce serveur, en lui envoyant une requête *INVITE*.

Les serveurs SIP peuvent fonctionner soit comme un proxy, soit comme un serveur de redirection. Les deux cas sont illustrées dans la figure 5.4a. Les serveurs SIP proxy retransmet le message *INVITE* vers l'adresse IP courante de la personne recherchée. En échange, un serveur SIP de redirection répond directement au correspondant à l'origine du message *INVITE* en l'informant sur la localisation de la personne recherchée.



(a) Pre-call Mobility

(b) Mid-call Mobility

FIG. 5.4.: La mobilité dans SIP

Si un des participants à une session multimédia change d'adresse au cours d'une session ((*mid-call mobility*)), il doit envoyer un message *INVITE* à son correspondant (figure 5.4b). Il précise l'identificateur de la session en cours et remplie le champ *Contact* avec la nouvelle adresse. Pour rediriger le flot de donnes, il doit préciser l'adresse de transport complète, y

compris le numéro de port, dans le champ SDP (*Session Description Protocol*). Finalement, il doit mettre à jour son enregistrement dans le serveur SIP du domaine d'origine, pour que les nouveaux appelants puissent le joindre.

SIP est plutôt une solution pour la mobilité gérée au niveau application, puisqu'il a été conçu pour initier une session et échanger les informations de contact utilisées ensuite par les applications multimédia. Ces application utilisent le protocole RTP (*Real-Time Transport Protocol*)[86] au-dessus de l'UDP pour 'échanger des données. Contrairement au TCP, RTP n'utilise pas les adresses IP pour maintenir une association entre deux points distants et donc n'est pas sensible aux changements d'adresse. Les extrémités d'une connexion sont identifiées par une valeur de 32 bits choisie aléatoirement, appelé SSRC (*Synchronization Source*).

Il est fort probable que SIP va jouer un rôle important dans l'Internet de demain, vu son rôle d'initiation de session entre les applications multimédia mobiles. Cependant, il n'est pas compatible dans sa forme de base avec le transport de paquets par TCP, à cause de la dépendance de celui-ci des adresses IP utilisées. Une proposition[87] pour résoudre cela utilise des messages de notification *SIP INFO* et des agents *SIP-EYE* implantés dans les terminaux. Le rôle de ces agents est d'intercepter les connexions TCP et utiliser l'encapsulation IP pour maintenir des adresses IP stables au bouts d'une connexion TCP.

6. Conclusions sur l'état de l'art

Nous avons présenté dans les deux chapitres précédents les différentes propositions qui essayent de gérer la mobilité d'hôtes dans l'Internet. Pour résoudre l'incompatibilité entre le rôle d'indicateur d'acheminement d'une adresse IP et celui d'identifiant d'hôte qui lui est prêté dans TCP, les solutions existantes ont été emmenées à traiter d'une façon différente certains aspects. Dans la suite, nous énumérons et analysons ces aspects, en précisant pour chacun les choix possibles.

6.1. Choix de conception

Adressage et routage. Un de plus importants points de conception est de changer ou pas le fonctionnement habituel du niveau réseau.

- Les adresses IP ont une connotation topologique, dans le sens qu'une certaine adresse IP ne peut être employée que dans le sous-réseau dont elle fait partie. Ceci est une propriété fondamentale sur laquelle est fondée le routage dans l'Internet et qui permet à celui-ci de fonctionner à l'échelle actuelle.

- À l'inverse, si on permet à une hôte de garder et utiliser la même adresse IP à plusieurs endroits, il faut changer le mécanisme d'acheminement des paquets utilisé par les routeurs. Ainsi, l'acheminement de paquets pourrait se faire en fonction de l'intégralité de l'adresse IP. Pour ce faire, les adresses IP de toutes les hôtes mobiles devraient être connues par tous les routeurs. Le désavantage direct de ce mode d'opération est qu'il n'est pas applicable à l'échelle de l'Internet sans provoquer une explosion dans les tables de routage. De plus, suite à la mobilité d'une hôte, une mise à jour doit être réalisée dans tous les routeurs qui pourront être concernés. Ceci restreint l'application de cette aproche à la gestion de la mobilité dans des domaines de petite taille et avec un nombre limité de terminaux mobiles.

Architecture end-to-end ou tierce partie. Cette question se réfère à l'implication d'un élément additionnel, autre que les deux machines aux extrémités d'une connexion, dans les mécanismes de gestion de la mobilité.

- La majorité de solutions qui gèrent la mobilité des hôtes à grande échelle choisissent d'utiliser l'adresse IP comme indicateur de routage de datagrammes. Les deux hôtes qui participent à la connexion s'informent réciproquement sur les changements d'adresse qui ont lieu et envoient les datagrammes aux adresses mises à jour.

- Dans d'autres propositions, les hôtes gardent inchangées leur adresses mais on ajoute à l'infrastructure de routage un élément ayant une fonction de redirection des datagrammes vers les machines mobiles.

Nommage et résolution de noms. Pour garder l'identité d'une hôte en dépit de ses mouvements, on a besoin d'un nom stable pour identifier les machines, indépendamment de leur adresse IP qui peut changer. Ensuite, si les machines identifient leurs correspondants par un nom, on a besoin aussi d'un dispositif de translation entre ce nom et l'adresse IP du point d'attachement courant de l'hôte mobile. Les questions qui se posent sont :

– Quel espace de noms choisir ?

– Comment ce nom est attribué aux hôtes mobiles pour assurer son unicité ?

– Qui retient l'association nom - adresse IP ? Est-elle visible aux machines correspondantes ou la translation se fait automatiquement dans l'infrastructure Internet ?

La plupart de solutions supposent l'existence d'un répertoire qui contient des couples *identifiant - adresse actuelle* pour les machines mobiles. Au moment de l'initiation d'une connexion ou après un déplacement, ce répertoire permet de découvrir la position actuelle de l'hôte cible. Par la suite, l'adresse IP reçue en réponse sera utilisée directement par la machine correspondante pour toutes les datagrammes envoyés à l'hôte mobile. L'indirection est réduite donc à un seul échange en début de connexion. En cas du déplacement subséquent de l'hôte mobile au cours de la connexion, celle-ci devra mettre à jour le répertoire avec sa nouvelle adresse et éventuellement notifier ses correspondantes pour éviter toute incohérence.

Transparence vis-à-vis de couches supérieures. Dans leur présentation, nous les avons groupé les différentes solutions en fonction de la couche où elles opèrent : couche réseau, couche transport, couche session :

- L'avantage d'une solution qui implique seulement la couche IP est qu'ainsi la mobilité d'hôtes est masquée aux niveaux supérieurs. Ainsi, la couche transport et les applications réseau ne subissent aucune perturbation et les connexions établies entre deux machines peuvent persister si le point d'attachement change.

- D'autres solutions proposent des modifications et améliorations dans les protocoles de transport pour accepter le fait que l'adresse IP d'une machine peut changer en cours d'une connexion. Ainsi, on peut modifier ou apporter des nouvelles options dans le protocole TCP pour permettre des changements dans les adresses IP des extrémités. En complément, d'autres protocoles alternatifs de transport ont été conçus pour pouvoir fonctionner avec plus d'une adresse IP et permettre un changement d'adresse à la volée.

- D'autres propositions ne touchent ni le protocole IP, ni le TCP, mais proposent un

niveau *session*. Celui-ci permet de suivre le déroulement d'une ou plusieurs connexions TCP et définir des points d'interruption. Dans le cas d'une interruption causé par la mobilité ou par l'indisponibilité temporaire d'une des deux extrémités, l'état de la connexion est sauvegardé. Par la suite, on peut recommencer une connexion à partir du point d'interruption de la précédente, dès que cela devient possible. L'avantage de cette approche est qu'elle gère de la même façon la mobilité et les autres types d'interruption.

6.2. Classification des solution présentées

Nous présentons dans le tableau 6.1 une comparaison entre toutes les solution analysées dans cette partie.

Dans Mobile IP, ainsi que dans les protocoles de micro-mobilité (Hawaii et Cellular IP), on vise la transparence de la mobilité vis-à-vis des hôtes correspondants et vis-à-vis des couches supérieures. Ceci est fait en permettant au terminal de garder inchangée son adresse IP. Dans Mobile IP, cela est obtenu avec le prix d'indirections et inefficiences dans l'acheminement des paquets. Plusieurs extensions de Mobile IP ainsi que sa variante Mobile IPv6 résolvent en partie les inefficiences du procole original en informant les hôtes correspondants de la vraie adresse de l'hôte mobile.

Dans les protocoles de micro-mobilité, on actionne toujours au niveau IP, mais cette fois on se focalise sur l'infrastructure de routage. Les différentes propositions ont comme point commun le changement de l'acheminement standard de datagrammes et l'utilisation des services spécifiques dans les routeurs présents dans l'infrastructure. Nous avons présenté en détail HAWAII, un protocole où l'acheminement des datagrammes est fait en fonction de l'intégralité d'un adresse IP, ainsi qu'une autre proposition similaire, Cellular IP.

Les solution que nous avons présenté ensuite sont construites sur le principe général *de bout en bout* et s'avèrent plus efficaces. Au niveau IP, la séparation entre les deux rôles d'une adresse IP peut être faite par l'introduction d'une couche IP virtuelle. Ainsi, dans VIP et VNAT les connexions transport sont ouvertes entres des adresses IP virtuelles et stables. Par un mécanisme de translation d'adresses ou d'encapsulation IP, les paquets de données sont ensuite transformés en datagrammes qui contiennent les adresses IP réelles et qui peuvent être acheminées dans l'Internet. Cette approche propose de faire possible la continuation des connexions TCP en rendant transparent le changement d'adresse au dessus du niveau IP. Ces solutions assurent un routage optimal, sans modifier l'infrastructure IP existante, mais néanmoins elles supposent la participation et la coopération des deux machines.

D'autres solutions proposent d'actionner là où il y a le problème, c'est-à-dire dans les protocoles de niveau transport. Nous considérons que le déploiement des modifications dans la couche transport sera lent et difficile dans l'Internet. Ces propositions introduisent des nouvelles options de TCP qui permettent de notifier à l'hôte correspondante un changement d'adresse. Le nouveau protocole SCTP permet quant à lui d'utiliser plusieurs adresses IP aux deux extrémités, et cela même simultanément pour les machines multi-connectés.

	Mobile IP	Mobile IPv6	Hawaii & CIP	LIN6	VIP	VNAT	Migrate TCP	SCTP	Migrate Session	SIP
Adresse(s) IP des hôtes mobiles	de base & courante	de base & courante	Adresse unique	LIN6 & courante	VIP & courante	Première adresse & suivantes	Adresse courante	Plusieurs adresses en même temps	Adresse courante	Id. personnel & adresse courante
Routage	Indirect	Direct	Direct	Direct	Direct	Direct	Direct	Direct, plusieurs chemins	Direct	Direct
Modification des datagrammes	Encapsulation	Options IPv6	Non	Non	Encapsulation	Non (NAT sur les hôtes)	Non (à part options TCP)	Non	Non	Non
Déroulement du handoff	Hôte mobile ↔ Agent de base	Hôte mobile ↔ Hôtes corresp.	Hôte mobile ↔ Routeurs du domaine	Hôte mobile ↔ Répertoires	Hôte mobile ↔ Serveur DNS	Hôte mobile ↔ Hôtes corresp.	Hôte mobile ↔ Hôtes corresp.	Hôte mobile ↔ Hôtes corresp.	Hôte mobile ↔ Hôtes corresp.	Hôte mobile ↔ Hôtes corresp.
Passage à l'échelle	Oui	Oui	Non, mobilité locale	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Couche d'opération	IP	IP	IP	IP	IP	IP	TCP	SCTP (transport)	Session	Application
Eléments additionnels	Agent de base, agents visités	Agent de base (partiel.)	Routeurs du domaine	Répertoires	Serveurs DNS dynamiques	Non	DNS dynamique	Non	DNS dynamique (déplacement simultané)	Serveurs SIP
Transparence vis-à-vis des hôtes correspondantes	Oui	Non	Oui	Non	Non	Non	Non	Non	Non	Non
Changement de l'infrastructure IP	Non	Non	Oui (routes d'hôte)	Non	Non	Non	Non	Non	Non	Non
Avantages	Transparence aux niv. superieurs et aux hôtes corresp.	Routage direct, envoi des màj aux hôtes corresp.	Routage direct	Routage direct, pas de messages de contrôle	Routage direct	Routage direct, usage des adresses IP courantes	Routage direct	Gère aussi le multi-accès	Gère en même temps les déconnexions longues	Supporte la mobilité de services, personnes, et terminaux
Désavantages	Agents (HA, FA), temps de handoff long, routage triangulaire	IPv6 pas encore déployé à grande échelle	Ne peut fonctionner à l'échelle de l'Internet	Implantable que dans IPv6	Encapsulation négociation des adresses VIP	Echec si collision d'adresses, et déplac. simultané	Couche TCP modifié sur toutes les hôtes, UDP n'est pas pris en compte	SCTP n'est pas performant, déploiement difficile	Tampons additionnels, difficilement implantable	Conçu que pour les applications multimédia RTP

TAB. 6.1.: Comparatif des solutions présentées

À la fin, nous avons présenté deux autres solutions qui opèrent cette fois au dessus des niveau IP et transport. La réintroduction du niveau session dans la pile TCP/IP est très intéressante parce qu'en plus de la mobilité, elle prend aussi en charge les problèmes dus aux interruptions. Cependant, l'implémentation d'une telle couche reste très difficile, à cause de l'impossibilité d'interagir avec les implémentations de la couche transport du noyau, et en particulier avec les tampons de sockets réseau. Quant à lui, le protocole SIP gère, en plus de la mobilité des machines, la mobilité des personnes. Toutefois, il reste une solution envisageable que pour les applications multimédia qui s'échangent des données par RTP et UDP. Pour les connexions TCP, il a besoin des éléments additionnels et d'utiliser l'encapsulation des datagrammes IP. Cette classe de proposition fait référence au niveau session de la pile OSI. Elle offre aux applications des nouvelles primitives et services pour pouvoir recommencer une nouvelle connexion TCP du même point ou la précédente s'est terminée.

Troisième partie .

Contribution

7. Mobilité locale dans les réseaux sans fil 802.11

Grâce au rôle unificateur de son protocole de base, l'IP, l'Internet est composé aujourd'hui d'une multitude de types de réseau. Parmi ceux-ci, on remarque la présence de plus en plus importante des réseaux sans-fil. Parmi les différentes technologies d'accès sans-fil à l'Internet, la plus populaire est sans aucun doute l'IEEE 802.11. Nous avons utilisé ce type de plate-forme sans-fil pour nos prototypes et expérimentations de la micro-mobilité IP, qui seront présentés dans le chapitre 8. Dans ce chapitre nous étudions les réseaux locaux de type 802.11 et principalement leurs mécanismes de mobilité.

7.1. Le protocole 802.11

Le standard 802.11 de l'IEEE pour les réseaux locaux sans-fil définit une partie des couches bases de la pile OSI[22] : la couches physique et la couche liaison de données.

7.1.1. La couche physique

Pour la couche physique, le standard initial publié en 1997[3] proposait trois techniques de transmission : FHSS (*Frequency Hopping Spread Spectrum*), DSSS (*Direct Sequence Spread Spectrum*) et IR (*InfraRed*). Les deux premières fonctionnent dans la bande de fréquences de 2,4 Ghz et peuvent offrir un débit maximal de 1 ou 2 Mbps. La version 802.11b publié en 1999[88] a retenu que le deuxième type de transmission physique et y apporte des améliorations dans HR/DSSS (*High Rate Direct Sequence Spread Spectrum*) pour obtenir des débits pouvant aller jusqu'à 11 Mbps. La version 802.11a[89] choisit quant à elle d'utiliser une autre technique de transmission appelée OFDM ; elle change également la bande de fréquences utilisée à 5 Ghz, avec des débits de transmission jusqu'à 54 Mbps. Le même débit maximum caractérise aussi la dernière version du standard, 802.11g[90], apparue en 2001 et qui utilise la modulation OFDM dans la bande de 2,4 Ghz pour rester compatible avec les équipements 802.11b existants. Une synthèse de ces technologies est présentée dans la table 7.1.

La bande de fréquences utilisée, que ce soit 2,4 Ghz pour 802.11b et 802.11g ou 5Ghz pour 802.11a est repartie sur plusieurs plages de fréquences, appelées canaux. Ainsi, 14 canaux sont disponibles dans la bande de 2,4 Ghz; ils se recouvrent partiellement, d'où l'utilisation habituelle des seuls trois canaux isolés 1, 6 et 11 (voir la figure 7.1).

Standard	Technologie de la couche physique	Bande de fréquences	Débit maximum
802.11	Modulation FHSS (étalement de spectre avec sauts de fréquence)	Bande de 2,4Ghz (2.400-2.4835 GHz) 75 canaux de 1 Mhz	1 ou 2 Mbps
	Modulation DSSS (étalement de spectre à séquence directe)	Bande de 2,4Ghz (2.400-2.4835 GHz) 14 canaux de 22 Mhz se recouvrant	
	IR (infrarouge)		
802.11b	HR/DSSS (High Rate / Haut Débit) basé sur la modulation CCK (Complementary Code Keying – jeu complémentaire de clefs codées)	Bande de 2,4Ghz (2.400-2.4835 GHz) 14 canaux de 22 Mhz se recouvrant	1, 2, 5.5 ou 11 Mbps
	DSSS original pour la compatibilité avec le 802.11 original		
802.11a	Modulation OFDM (multiplexage par division en fréquences orthogonales)	Bande de 5 Ghz (5.15-5.825 Ghz) 12 canaux de 20 Mhz indépendantes	6, 9, 12, 18, 34, 36, 48 ou 54 Mbps
802.11g	Modulation OFDM	Bande de 2,4Ghz (2.400-2.4835 GHz) 14 canaux de 22 Mhz se recouvrant	1, 2, 5.5 ou 11 Mbps ; 6, 9, 12, 18, 24, 36, 48 ou 54 Mbps
	Modulation DSSS avec CCK pour la compatibilité avec le 802.11 b		

TAB. 7.1.: Les couches physiques des protocoles 802.11 a,b et g

Couche	Standard	Objet du standard
	802.1	Norme générale. Le fonctionnement inter-réseaux (réseaux pontés). Séparation des deux couches OSI *Physique* et *Liaison* en trois sous-couches *LLC, MAC et PLS*
LLC – Link Layer Control	802.2	Définition et spécifications de la couche contrôle de liaison
Contrôle d'accès au medium (*MAC - Media Access Control*) et Couche physique *(PLS - Physical Layer Signaling)*	802.3	Les réseaux locaux en bus logique (Ethernet) avec la méthode d'accès CSMA/CD
	802.4	Les réseaux locaux en bus à jeton (*Token Bus LAN*)
	802.5	Le réseaux locaux en anneau à jeton (*Token Ring LAN*)
	802.6	Les réseaux métropolitains (*MAN*)
	802.11	Les réseaux locaux sans fil (*Wireless LAN*)
	802.12	Les réseau basés sur la priorité de la demande
	802.15	Réseaux personnels sans fil (*Wireless PAN*)
	802.16	Les réseaux métropolitains sans fil (*Wireless MAN*)
	802.17	Les réseaux *Resilient Packet Ring (Token Ring* amélioré)
	802.22	Les réseaux sans fil régionaux (*Wireless RAN*)

TAB. 7.2.: Les protocoles 802

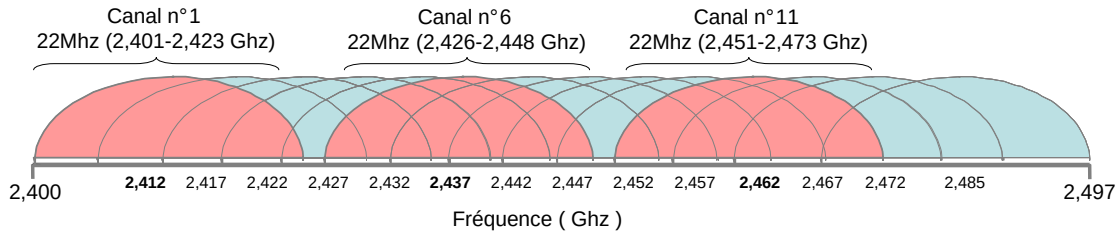


FIG. 7.1.: Les canaux de transmission dans 802.11b

7.1.2. La couche d'accès au médium

En ce qui concerne la couche de liaison de données de 802.11, elle se compose de deux sous-couches : le contrôle de la liaison logique LLC (*Logical Link Control*) et le contrôle d'accès au medium MAC (*Medium Access Control*). Le standard 802.11 utilise le protocole LLC 802.2[91] et l'adressage sur 48 bits, tout comme les autres réseau locaux 802, simplifiant ainsi le pontage entre les réseaux sans fil et filaires (un récapitulatif des standards 802 est présenté dans la table 7.2). Par contre, la couche MAC est propre aux réseaux sans-fil. Elle reste très proche de la couche Ethernet 802.3[25] dans sa conception, étant conçue pour faire coexister plusieurs machines sur un même support partagé.

La sous-couche MAC de 802.11 définit deux modes d'opération fondamentalement différentes : DCF (*Distributed Coordination Function*) et PCF (*Point Coordination Function*). La dernière fournit un support optionnel au dessus du DCF et utilise les services d'une entité centralisée qui a le rôle de coordonnateur des communications dans la cellule. PCF été conçu pour offrir les garanties nécessaires pour le trafic sensible, en temps réel. Resté toutefois optionnel, PCF est peu implémenté, la quasi-totalité du matériel existant utilisant le schéma de base DCF.

DCF et CSMA/CA

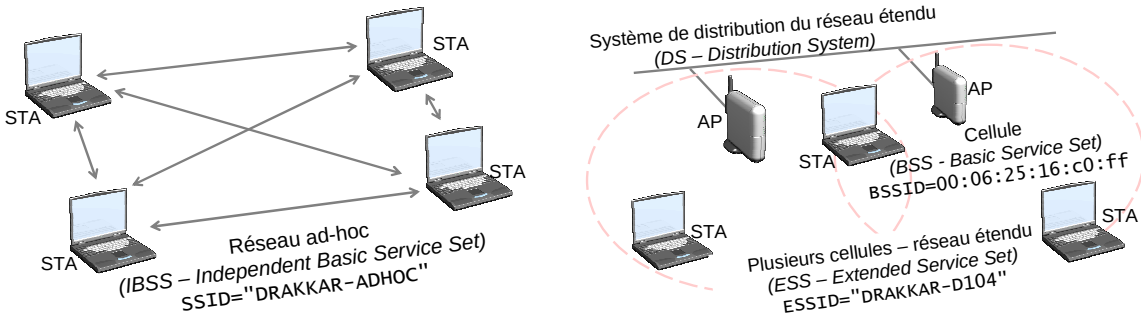
La technique d'accès au medium que DCF utilise s'appelle CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Elle partage son principe de base avec le standard 802.3 (Ethernet) : CS (*Carrier Sense*), ce qui signifie « écoute avant de transmettre ». Dans les réseaux Ethernet, les collisions sont détectées durant la transmission (CD - *Collision Detection*). En échange, dans 802.11 les collisions sont plutôt évitées (CA - *Collision Avoidance*) parce que le médium sans-fil n'offre pas la possibilité d'écouter est de transmettre en même temps.

Dans une cellule sans-fil, une station émet sur le canal seulement si celui-ci est détecté libre. Il pourrait y avoir toujours une collision si deux stations écoutent le medium, le détectent libre et commencent à transmettre en même temps. Pour empêcher cela, une propriété fondamentale du protocole CSMA/CA est la période de temps aléatoire entre l'écoute du medium et le début de la transmission, appelée période de contention (*contention window*).

En dépit de ce temps aléatoire d'attente, une collision peut toujours avoir lieu si les temps d'attente choisis par deux stations sont les mêmes. Ainsi, pour s'assurer de l'envoi correct d'une trame, le protocole 802.11 demande que toutes les stations envoient un acquittement à chaque trame reçue. Ceci est valable que pour les trames envoyées en *unicast* ; pour celles envoyés en *multicast* et *broadcast* il y a pas d'acquittement prévu. Si l'émetteur d'une trame ne reçoit pas l'acquittement attendu, il va réessayer l'envoi plusieurs fois. Si l'échec persiste, cela est perçu comme une indication des erreurs de transmission et détermine le passage à un débit inférieur (5.5, 2 ou 1 Mbps pour 802.11b).

7.1.3. Les modes infrastructure et ad-hoc

Du point de vue de l'architecture, 802.11 définit deux modes d'opération : le mode infrastructure BSS (*Basic Service Set*) et le mode ad-hoc IBSS (*Independant Basic Service Set*). La topologie du mode ad-hoc est très simple et l'ensemble des stations communique directement, par paires, sans aucune fonction de relais de messages. Le mode infrastructure est beaucoup plus répandu que le mode ad-hoc et il définit un élément central, le point d'accès - AP (*Access Point*). Tous les messages passent par le point d'accès qui les relaie localement vers leur destination.



(a) Mode ad-hoc

(b) Mode infrastructure

FIG. 7.2.: L'architecture d'un réseau sans-fil

Généralement les réseaux sans-fil 802.11 sont déployés en regroupant plusieurs points d'accès rapprochés, pour former une zone de couverture étendue composée des cellules de couverture contiguës. Ce réseau étendu est appelé ESS (*Extended Service Set*) et les points d'accès qu'il contient coopèrent entre eux pour acheminer les messages entre les cellules desservies.

Nous allons maintenant introduire quelques éléments de terminologie présents dans le standard 802.11. Chaque station faisant partie d'une cellule sans-fil est appelée STA, y compris le point d'accès. Chaque réseau indépendant (de type BSS ou IBSS) comporte un identificateur appelé SSID (*Service Set ID*), qui est une chaîne de maximum 32 caractères. Dans le cas d'un réseau étendu, cet identificateur est appelé ESSID (*Extended Service Set ID*) ; les cellules qui le forment par sont alors identifiées par le BSSID, qui est l'adresse physique du

point d'accès de la cellule respective.

7.1.4. La structure des trames 802.11

La figure 7.3 présente le format standard d'une trame 802.11. Les données sont placées dans le champ *Frame Body*, d'une longueur variable. Les stations source et destination ainsi que les points d'accès utilisés pour relayer la trame sont identifiées par leurs adresses physiques, sur 6 octets, dans les champs de type *Address*.

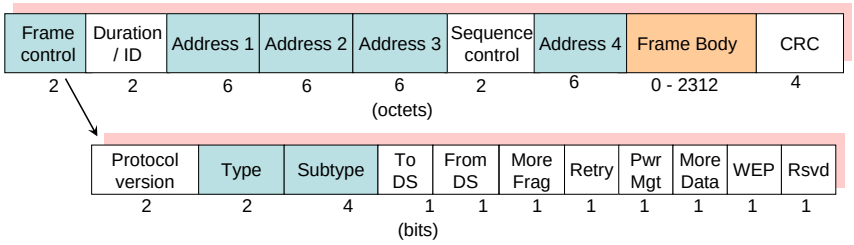


FIG. 7.3.: La structure d'une trame 802.11

Il y a trois types de trames qui sont envoyées parmi les stations d'un réseau sans-fil 802.11. Ce type, codé dans le champ *Frame Control*, catégorise les trames en :

- Trames de données ;
- Trames de contrôle, utilisées pour coordonner l'accès au médium. Dans cette catégorie entrent les trames d'acquittement - ACK (*Acknowledgement*) - ou les trames RTS (*Request to Send*) et CTS (*Clear to Send*), dont le rôle est d'éviter les collisions avec des stations plus éloignées. À cause de leur caractère de contrôle, cette catégorie de trames est prioritaire pour l'accès au medium.
- Trames de management. Celles-ci ont la même priorité d'accès au medium que les trames de données. Leur rôle est l'échange des informations relatives strictement au protocole 802.11 (synchronisation, scanning, authentification, association) entre les stations du réseau sans-fil.

7.2. Le handoff 802.11

Dans cette section nous dressons une analyse du processus d'association et de handoff dans le protocole 802.11, en précisant particulièrement le rôle des trames de management qui sont utilisées à ce but.

7.2.1. L'association au réseau

Une station doit s'associer auprès d'un point d'accès pour pouvoir envoyer et recevoir des trames. Cette procédure d'association a lieu au moment de la première connexion de la station au point d'accès et se répète quasiment à l'identique chaque fois que la station de reconnecte au même réseau sans-fil. La connexion à un réseau sans-fil comporte deux phases :

- Découverte des points d'accès présents dans le voisinage et choix de l'AP cible ;
- Authentification et association auprès du point d'accès choisi.

Le scan

Premièrement, la station doit trouver les points d'accès potentiels auxquels elle peut se connecter. Ceci se réalise par une phase de découverte qui s'appelle *scan*. Pendant un scan, la station change sa fréquence radio sur chaque canal de communication et essaye de capter des messages provenant des APs. Le standard spécifie deux types de scan : *actif* et *passif*. Dans le scan actif, la station change sa fréquence sur chaque canal, diffuse des trames *Probe Request* et attend des trames de réponse *Probe Response*. Dans le mode passif, la station change aussi les canaux de communication mais elle n'envoie aucune trame et écoute uniquement le médium pour intercepter des trames *Beacon* qui sont envoyées périodiquement par les points d'accès.

Une fois que le scan est fait, la station possède la liste des points d'accès qui lui sont accesibles. Les informations contenues dans les trames *Probe Response* ou *Beacon* envoyées par les points d'accès sont utilisées par la station pour choisir celui auquel elle va essayer de s'associer. En général, la station va choisir le point d'accès qui a le meilleur indicateur SNR (*Signal to Noise Ratio*)[1].

L'authentification et l'association

Après que la station a choisi un point d'accès candidate pour s'associer avec, elle va initier la procédure d'authentification. Les versions initiales du protocole 802.11 spécifient deux méthodes : *Open System Authentication* et *Shared Key Authentication*. La première est très simple : la station envoie une première trame et le point d'accès lui répond en lui indiquant l'acceptation ou le rejet. Le mécanisme de control d'accès peut se basee que sur l'adresse MAC de la station, présente dans sa demande d'authentification. La deuxième méthode suppose l'existence d'un secret partagé entre la station et le point d'accès. Ce secret est représenté par une clé WEP (*Wired Equivallent Privacy*), utilisée aussi pour le chiffrement éventuel de trames de données. Un échange supplémentaire de deux trames (*défi - réponse*)

[1] Ce paramètre indique la puissance du signal reçu à la réception d'une trame, rapporté au bruit présent sur le canal de communication.

est rajouté dans l'authentification, dans lequel la station doit déchiffrer un texte fourni par le point d'accès.

Le contrôle d'accès basé sur les adresses MAC des stations peut être facilement contourné avec des outils logiciels qui permettent de reconfigurer l'adresse MAC des interfaces sans-fil[92]. D'un autre coté, la clé WEP est aussi utilisé pour chiffrer les trames de données et peut être compromise en analysant un nombre suffisant de trames[93]. À présent d'autres méthodes d'authentification plus sures sont apparues. Ainsi, WPA (*Wi-Fi Protected Access*)[94] permet de changer dynamiquement la clé WEP et WPA2 (*Wi-Fi Protected Access version 2*)[95], basé sur le standard 802.11i[96], fait usage des protocoles EAP[97] et 802.11X[98]. Le dernier permet notamment d'interagir avec des serveurs RADIUS (*Remote Authentification Dial-In User Service*)[99] et de s'authentifier à l'aide d'un mot de passe ou du certificat utilisateur.

Après que la station est authentifiée auprès du point d'accès, elle initie le dernier échange de trames de cette phase d'initialisation. La station envoie une trame *Association Request* et le point d'accès répond avec une trame *Association Response*. Si la réponse est positive, la station et le point d'accès peuvent commencer à s'échanger entre eux des trames de données.

7.2.2. Le transfert d'association

Un handoff a lieu quand une station s'éloigne de son point d'accès courant et entre dans la couverture d'un autre AP dont la qualité du signal est meilleure. On peut identifier trois phases différentes dans le déroulement d'un handoff (voir la figure 7.4). La première est celle où la station s'aperçoit que le signal de l'AP courant est en baisse. La deuxième et la troisième sont en grandes lignes les mêmes que la connexion initiale au réseau : la découverte des points d'accès et la réassociation.

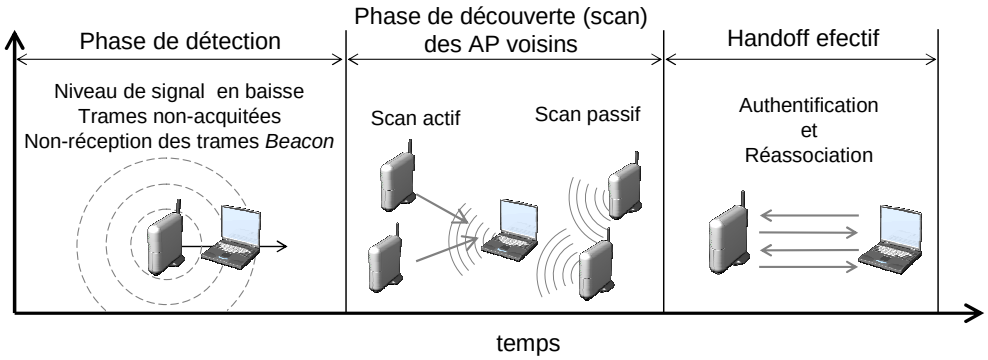


FIG. 7.4.: Les phases du handoff dans 802.11

En liaison directe avec le déplacement de la station dans le rayon de couverture d'un point d'accès, la puissance du signal du point d'accès perçu par la station peuvent se dégrader et causer des erreurs dans la transmission des trames sur le medium sans-fil. En général, la puissance du signal est en fonction de la distance, mais la topologie du site et

les interférences avec les autres équipements (une source très importante des interférences est représentée par les fours à micro-ondes, qui fonctionnent dans la même gamme de fréquences). Les erreurs de transmission dues à la puissance du signal reçu se traduisent dans la perte des trames de données ou des acquittements, mais ces pertes de paquets peuvent aussi apparaître à cause d'une congestion de trafic dans la cellule.

Le standard ne spécifie pas le moment où le client détecte la nécessité d'initier un scan, mais la plupart des implémentations présentes dans les équipements 802.11 le font au moment ou la qualité du lien[2] descend en dessous d'une certaine valeur. En initiant le scan, la station essaie de découvrir d'autres AP qui appartiennent au même réseau (c'est-à-dire qui ont le même ESSID) et dont la puissance du signal reçu est meilleure.

Le processus de réassociation est similaire au celui de l'association initiale. Il comporte deux ou quatre trames d'authentification en fonction de la méthode utilisée (*open system* ou shared key) et deux trames d'association appelées cette fois *Reassociation Request* et *Reassociation Response*. L'élément nouveau para rapport à l'association initiale est la spécification de l'ancien point d'accès de la station dans la trame *Reassociation Request*.

Cet élément nouveau peut être utilisé pour un échange supplémentaire de messages entre les deux poins d'accès (l'ancien et le nouveau), conformément aux spécifications d'un document récent de l'IETF - la recommandation 802.11f[3] [26]. Ainsi, le nouvel AP envoie à l'ancien un message *IAPP MOVE Notify* et la réponse consiste dans un message *IAPP MOVE Response*. Le but de ce messages est l'échange des informations regardant la station mobile, comme par exemple des informations de sécurité permettant une authentification plus rapide au nouveau point d'accès.

Le même document recommande au nouveau point d'accès point d'accès d'envoyer un trame de mise à jour au niveau 2. Cette trame contient comme adresse source dans son en-tête LLC l'adresse MAC de la station mobile. Son rôle est la mise à jour des tables des équipements actifs de niveau 2, comme les ponts et commutateurs, pour que les trames destinées à la station soient acheminées vers sa nouvelle localisation.

7.3. Le handoff 802.11 en pratique

Comme nous l'avons mentionné auparavant, le standard 802.11 ne spécifie pas un mécanisme pour implémenter la phase de détection d'un handoff. Si pour la phase de scan le standard contient les algorithmiques génériques, on ne spécifie pas comment choisir l'un ou l'autre et quelles sont les valeurs des différents *timers* présents dans les algorithmes. En conséquence, on s'attend à avoir des comportements différents pendant les phases de détection et scan, dont les durées peuvent aussi varier.

Nous analysons dans cette section les différences de comportement des implémentations

[2] La qualité du lien est un indicateur composite qui varie en fonction de la puissance du signal reçu, corrélée avec le taux d'erreurs et de retransmissions.

[3] 802.11f consiste principalement dans la spécification du protocole de communication inter-points d'accès - IAPP (*Inter Access Point Protocol*)

du standard 802.11, en capturant les trames transmises pendant les trois phases du handoff et mesurant les durées de ces phases.

7.3.1. Outils et méthodologie

Nos expérimentations ont été réalisés en utilisant des cartes réseau sans-fil basées sur les chipset *Intersil Prism*, plus exactement les cartes *Farallon SkyLine 11* et *LinkSys WPC11*. Les cartes ont été utilisées sur des machines fixes et portables Intel Pentium III avec le système d'exploitation Linux et la version du noyau 2.4.20.

Le pilote HostAP

Pour réaliser nos expériences, nous avons utilisé le pilote HostAP développé par Jouni Malinen. HostAP[100] est un pilote Linux pour les cartes réseau 802.11 basées sur les chipsets Prism 2/2.5/3 qui permet notamment de faire fonctionner la carte réseau comme un point d'accès.

Ce mode de fonctionnement peut transformer une machine Linux quelconque, douée d'une carte réseau sans-fil Prism, dans un point d'accès logiciel. En plus, le pilote HostAP offre l'accès à plusieurs statistiques et paramètres de configuration. Ainsi, on peut établir une politique de contrôle d'accès basée sur les adresses MAC des stations et on peut forcer la dissociation d'une station. On a accès à des statistiques sur le trafic de la cellule et sur la qualité du signal reçu des stations.

HostAP permet bien sur d'utiliser la carte réseau dans le mode normal, c'est-à-dire comme station 802.11. Comme pour le point d'accès, on a accès à des statistiques regardant le trafic entre la station et le point d'accès, le niveau de la qualité du signal des trames reçues de point d'accès. On peut initier des scans pour découvrir les points d'accès voisins et leur puissance du signal. Quand il fonctionne comme une station, HostAP peut fonctionner dans un mode appelé *host roaming*, dans lequel les décisions regardant le handoff sont prises par l'utilisateur et non plus au niveau du *firmware* de la carte. Normalement, un scan est initié quand la qualité du signal descend en dessous d'un certain seuil, mais dans le mode *host roaming* on peut imposer que les scans soient initiés seulement par l'utilisateur. Lors d'un scan, on peut influencer le choix de l'AP choisi pour le handoff, par la spécification d'un paramètre appelé *AP preferé*. Si celui-ci est présent dans la liste des résultats d'un scan, il est choisi comme cible du handoff, sans que sa qualité du signal par rapport aux autres AP soit prise en compte.

Pour analyser le comportement et la durée de chaque phase d'un handoff, nous avons utilisé une autre fonctionnalité du pilote HostAP et du chipset Prism. Si la carte est passée dans le mode *Monitor*, on peut intercepter toutes les trames transmises sur un certain canal 802.11. Ceci inclut les trames de contrôle et de management qui normalement sont visibles seulement au niveau du firmware.

Outils dans l'espace utilisateur

Pour qu'on puisse interagir avec la carte réseau à partir de l'espace utilisateur, on a utilisé les fonctionnalités des *Linux Wireless Extensions*[101] et *Linux Wireless Tools*[102], partie d'un projet développé principalement par Jean Tourhilles. *Wireless Extensions* est une interface API générique qui permet à différents pilotes et différentes cartes réseau de présenter à l'espace utilisateur des paramètres de configuration et statistiques propres aux réseaux sans fil 802.11

Nous avons développé un outil qui qui utilise l'API de Wireless Extensions et fourni une interface graphique à l'utilisateur pour :

- Être averti de l'apparition de plusieurs événements comme l'initiation d'un scan, l'association ou la dissociation d'un point d'accès ;
- Demander les résultats du dernier scan et la qualité du lien avec le point d'accès courant ;
- Commander l'initiation d'un scan actif ou passif et spécifier le paramètre *AP préféré*.

Avec cet outil, nous avons pu initier des scans et des handoff et mesurer leur durée telle qu'elle est aperçue par l'utilisateur. Conjointement, dans nos expérimentations, nous nous somme servi d'une autre machine avec une carte 802.11 en mode Monitor. Cette machine a eu le rôle d'intercepter les trames échangées pour pouvoir analyser plus en détail la composition des temps de scan et handoff.

Méthodologie employée

Nous avons eu à disposition plusieurs méthodes pour initier un scan et éventuellement provoquer un handoff :

- Nous avons simulé le mouvement physique de la station par rapport à son point d'accès courant en modifiant la puissance de transmission du point d'accès logiciel HostAP. La qualité du signal reçu par la station se dégrade et dès qu'elle atteint le seuil fixé, le firmware de la carte initie un scan. Un autre point d'accès qui lui garde la puissance maximale est trouvé dans le résultat du scan et la station effectue vers lui.
- Nous avons passé le pilote HostAP en mode *host roaming*, où les décisions regardant le scan et le handoff sont prises par l'utilisateur. Ainsi, on peut à tout moment initier un scan. Conjointement, on peut forcer le handoff en spécifiant un point d'accès préféré qui sera choisi comme cible du handoff sans que la puissance du signal compte dans la décision.
- Une autre méthode est d'éteindre le point d'accès courant. Après une période de détection où l'indicateur de la qualité du lien courant baisse à cause des beacons attendus

et non reçus ou à cause des trames non acquittées par le point d'accès, un scan va être initié. Nous avons utilisé cette méthode pour analyser et mesurer la période de détection, entre la dernière trame échangée entre la station et son AP et le moment de début du scan.

7.3.2. Résultats

La phase de détection

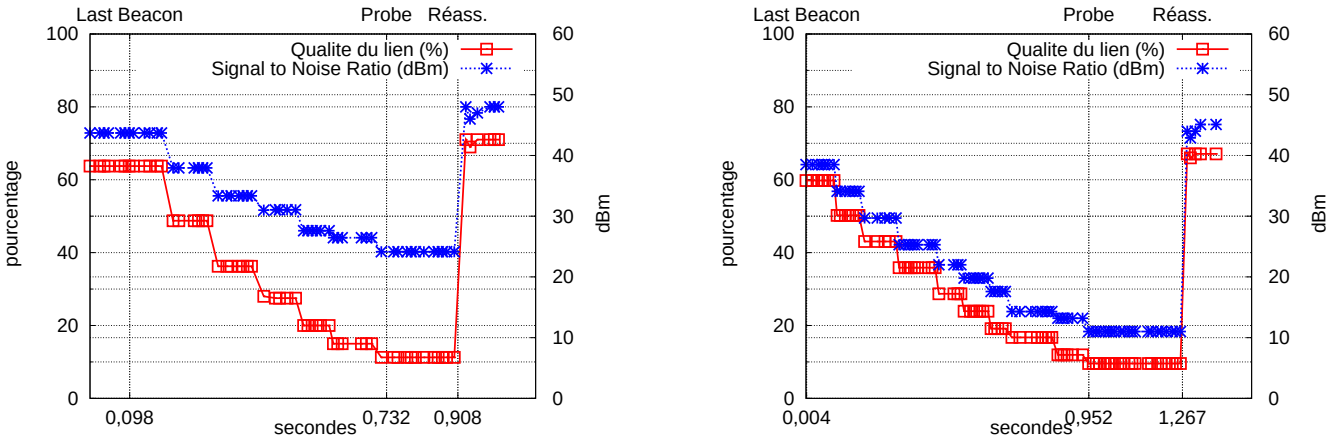
Dans la phase de détection, chaque constructeur d'équipements 802.11 utilise son propre algorithme pour calculer la qualité du lien avec le point d'accès courant et décider quand un scan est initié. Cet indicateur de qualité varie principalement en fonction de la puissance du signal observé à la réception de chaque trame envoyée par le point d'accès courant. Dans nos expérimentations, nous avons pu observer un autre facteur qui influence : la non-réception des trames attendues de la part du point d'accès courant.

Si l'éloignement de la station de son point d'accès se fait graduellement, les trames seront reçues par la station avec un signal de plus en plus faible et un scan sera initié avant que la station sorte de la cellule courante. Par contre, s'il n'y a plus d'échange de trames entre l'AP et la station, c'est le pourcentage des trames attendues et non reçues qui joue un rôle déterminant dans le calcul de la qualité de la ligne. Nous avons observé, par exemple, que les beacons attendus à des intervalles réguliers influencent en grande mesure la qualité du lien.

L'importance de ce paramètre dans le calcul de la qualité du signal diffère parmi les cartes que nous avons testées (Cisco et LinkSys). Par exemple, nos manipulations ont montré un taux de baisse d'environ 25% pour chaque beacon attendu et non réceptionné. Également, le seuil exact qui détermine quand un scan est initié diffère parmi les cartes testées, mais en général cette valeur est aux environs de 10%.

Nous avons représenté dans la figure 7.5 le comportement et la durée de la phase de détection pour deux cartes réseau différentes. À un instant donné on éteint le point d'accès courant d'une station ; nous avons observe la variation de la qualité du lien sur la station. Nous remarquons qu'il descend à chaque 100ms, c'est-à-dire à l'instant où un *beacon* est attendu[4]. Ainsi, la durée de la phase de détection mesurée se situe entre 500 et 800 millisecondes. Un comportement anormal observé à cette occasion concerne la puissance du signal reçu qui descend elle aussi, quoiqu'il n'y ait plus de trames qui soient reçues (elle devrait rester constante et montrer la valeur de la dernière trame reçue).

[4]Les *beacons* sont diffusés par le point d'accès aux intervalles réguliers. La valeur de cet intervalle est paramétrable ; les implémentations testées utilisent par défaut 100 millisecondes.



(a) Carte *Farallon SkyLine 11*

(b) Carte *LinkSys WPC 11*

FIG. 7.5.: La phase de détection

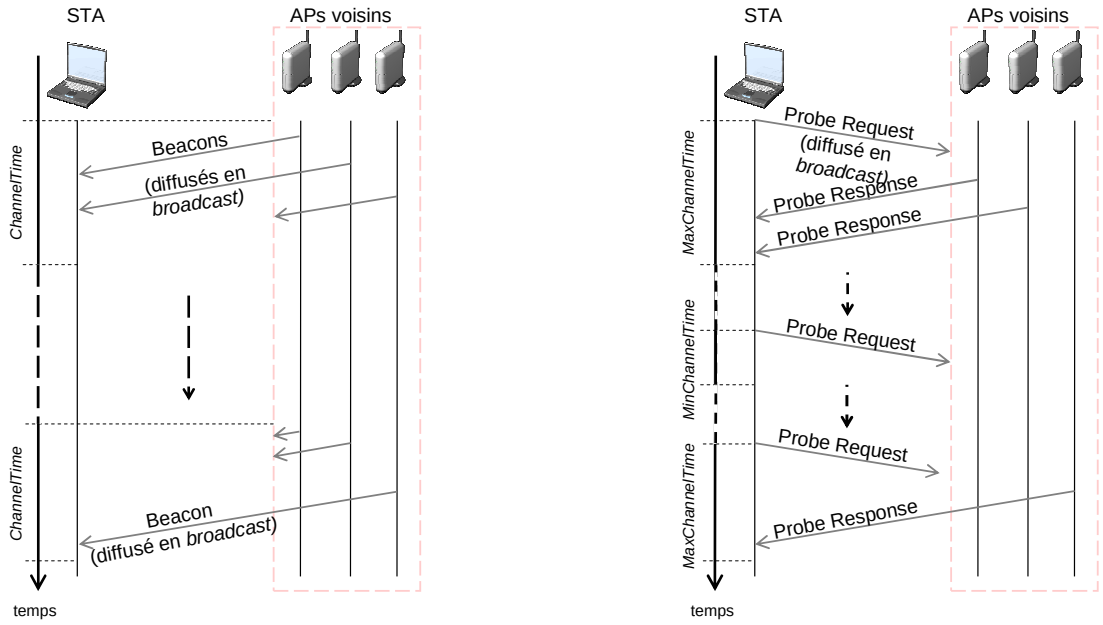
Le scan

Les deux types de scan spécifiés dans le standard 802.11 peuvent être utilisés dans la pratique, mais la plupart des implémentations présentes dans les stations 802.11 choisissent le mode actif parce qu'il est plus rapide. Le mode passif n'est présent que dans les implémentations les plus récentes et reste toutefois optionnel.

Le principal inconvénient du scan passif est le durée importante du temps pendant laquelle les stations doivent rester écouter sur chaque canal (figure 7.6a). Ce temps doit être plus long que l'intervalle d'envoi des beacons ; en plus, cet intervalle est inconnu jusqu'au moment où un premier beacon est reçu. La station ne peut changer de canal juste au moment ou un premier beacon ait été reçu, parce que d'autres points d'accès peuvent opérer sur le même canal, avec des fréquences d'envoi de beacons différentes. Comme l'ensemble des canaux[5] doit être parcouru, et que l'intervalle d'envoi des beacons peut être plus grand que la valeur de 100ms par défaut, le temps d'un scan actif est souvent de l'ordre de quelques secondes.

Dans le scan actif, la station participe activement au processus de découverte, dans le sens où elle diffuse sur chaque canal une trame qui demande aux points d'accès de lui répondre (voir la figure 7.6b). Comme pour le scan passif, le facteur important dans le scan actif est le temps d'attente sur chaque canal. Nous avons représenté dans la figure 7.7 l'algorithme de fonctionnement d'une station qui effectue un scan actif. On peut voir que le temps d'attente sur un canal est contrôle par deux *timers*. Le premier est le temps minimum d'écoute sur un canal (*MinChannelTime*). Si pendant ce temps la station ne détecte aucune activité sur

[5] 13 canaux peuvent être légalement utilisés en Europe contre 11 aux États-Unis



(a) Scan passif (b) Scan actif

FIG. 7.6.: Les deux types de scan definis par le protocole 802.11

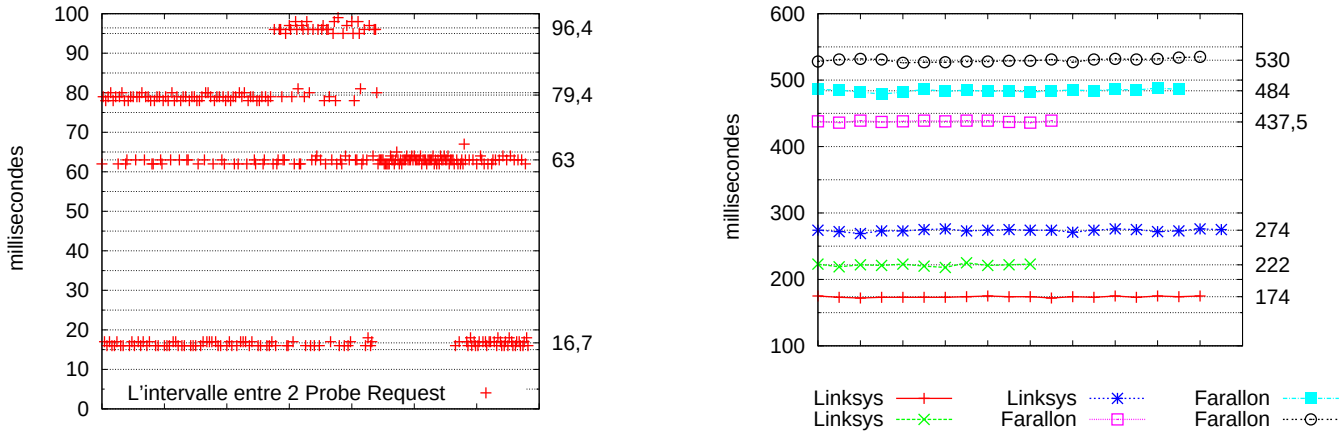
le canal, celui-ci est déclaré inactif et elle passe au canal suivant. Par contre, si on détecte une activité quelconque sur le canal, la station est obligé d'attendre les éventuelles trames de réponse pour un temps plus grand, le *MaxChannelTime*.

```
Pour chaque (canal dans la liste de canaux) {
        Change la fréquence sur le canal;
        Diffuse une trame Probe Request sur le canal;
        Initialisation d'un compteur;
        Tant que (compteur < MinChannelTime) {
                Attendre la réception de trames Probe Response;
                Incrémente le compteur;
        }
        Si (il y a eu de l'activité sur le canal)
                Tant que (compteur < MaxChannelTime) {
                        Attendre la réception de trames Probe Response;
                        Incrémente le compteur;
                }
}
```

FIG. 7.7.: L'algorithme de fonctionnement du scan actif

Les valeurs de ces deux paramètres diffèrent largement parmi les implémentations des stations 802.11. Pour les cartes que nous avons utilisées, on ne connaît pas les valeurs exactes, mais nous avons pu les déduire en capturant les trames *Probe Request* lors des scans en me-

surant l'intervalle existant entre elles. Les résultats, présentés dans la figure 7.8a, montrent pour les cartes douées d'un chipset Prism une value de 16,7 ms pour *MinChannelTime* et 63 ms pour *MaxChannelTime*. Les quelques valeurs de 79,4 ms et 96,4 ms représentent en effet le temps du scan de deux et trois canaux successives, les trames *Probe Request* intermédiaires étant probablement perdues.



(a) La durée de scan sur un canal

(b) La durée totale de scan

FIG. 7.8.: Le délai du scan actif

Ces scans ont été effectués en utilisant comme clients les cartes Farallon et LinkSys, dans un voisinage de deux points d'accès configurés pour opérer sur le canal 10 et respectivement 13. Nous montrons dans la figure 7.8b les temps totaux des scans, comme perçus par l'utilisateur, entre l'instant où il commande un scan et l'instant où les résultats sont disponibles. Les résultats obtenus se situent autour de trois valeurs moyennes : 437,5 ms, 484 ms et 530 ms pour la carte LinkSys et 174 ms, 222 ms et 274 ms pour la carte Farallon.

Cette différence entre les temps obtenus pour les deux cartes vient du fait que la carte LinkSys considère tous les 13 canaux permis en Europe ; par contre, la carte Farallon fonctionne d'après l'ancienne norme française et inspecte seulement 4 canaux (de 10 à 13). Sachant que les canaux adjacentes se chevauchent (par exemple une trame envoyée sur le canal 10 sera captée sur les canaux 8, 9, 10, 11 et 12), nous avons illustré dans la figure 7.3 la décomposition des temps totaux de scan dans une somme de temps d'écoute de chaque canal. Les valeurs résiduelles restantes représentent les délais de la communication entre l'espace utilisateur et la carte sans-fil.

En plus de nombre de canaux examinés par la station, le voisinage des points d'accès représente un autre facteur important dans les temps des scans. Leur nombre ainsi que les canaux sur lesquels ils opèrent déterminent si le temps d'attente sur un canal est soit celui minimum, soit celui maximum. Même si les points d'accès appartiennent à d'autres réseaux (avec un ESSID différent), leur activité sera détectée par la station qui atteindra pendant

Durée de scan (ms)	Temps d'attente sur chaque canal	Délai supplémentaire
174	2×63 + 2×16,7	14
222	3×63 + 1×16,7	16
274	4×63	22
437,5	4×63 + 9×16,7	35
484	5×63 + 8×16,7	35
530	6×63 + 7×16,7	35

TAB. 7.3.: La décomposition du temps de scan

MaxChannelTime sur le canal respectif. Les temps des scans que nous montrons dans la figure 7.9 sont pour une seule station équipée d'une carte LinkSys située dans un voisinage d'un, deux, ou trois points d'accès.

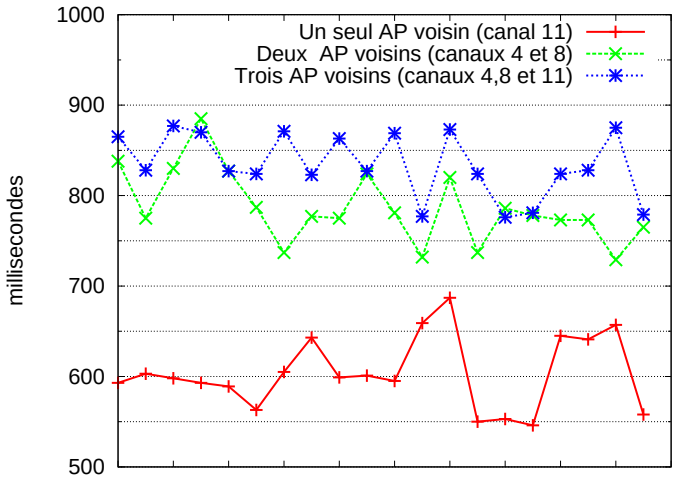


FIG. 7.9.: Le temps de détection en fonction du nombre des points d'accès voisins

La réassociation

Un handoff est initié si un point d'accès avec une meilleure puissance du signal est découvert lors du scan ou si un AP préféré est spécifié et présent dans les résultats du scan.

Nous avons mesuré la durée de cette phase de réassociation en capturant et analysant les trames échangées. Nos mesures ont relevé que les deux trames d'authentification et les deux trames de réassociation[6] ont besoin seulement de 4 ms en total pour être échangées. Si on utilise l'authentification WEP, les deux trames supplémentaires échangées agrandissent ce temps à environ 6 ms.

[6]Nous n'avons pas utilisé le protocole IAPP entre les points d'accès

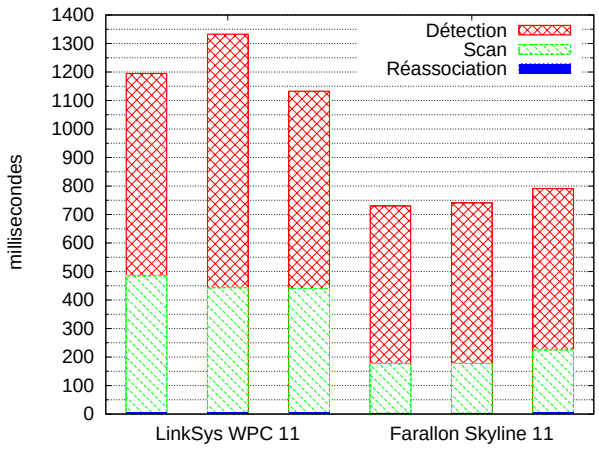


FIG. 7.10.: Comparatif des délais des trois phases du handoff

Pour conclure cette section, nous montrons dans la figure 7.10 les parts occupés par les différentes phases dans le temps total d'un handoff. On observe l'importance des deux phases de détection et de scan et le fait que la partie de réassociation représente une partie infime du temps total.

7.4. Optimisation du temps de handoff

7.4.1. Le handoff en situation de charge

Les mesures de handoff et scans que nous avons présentées précédemment ont été réalisées dans les conditions où il n'y avait pas de trafic de données dans la cellule.

Si le trafic de données dans la cellule est important, la possibilité d'apparition des collisions et donc de pertes de trames devient importante. En premier lieu, les résultats des scans peuvent ne pas être justes, parce que deux types de trames présentes dans ce processus, les *Probe Request* et *Beacon*, sont diffusées en *broadcast* et donc il n'y a pas d'acquittement pour confirmer leur réception. Nous avons réalisé une expérience (voir la figure 7.11) dans laquelle une station STA_1 initie un scan qui devrait permettre de découvrir un point d'accès voisin AP ; cet AP est impliqué dans un échange important de données avec une autre station STA_2. L'échange de données est simulé par un trafic descendant unidirectionnel $\mathsf{AP} \rightarrow \mathsf{STA}_2$, avec des trames de taille variable. Dans la figure 7.11 on présente le pourcentage de l'apparition de cet AP dans les résultats du scan, en fonction de la charge du trafic qu'il support et de la taille des trames présentes dans le trafic. Les résultats obtenus nous montrent qu'en situation de trafic important, une partie des trames *Probe Request* collisionnent et sont perdues.

Les trames échangées pendant les phases d'authentification et de réassociation, même

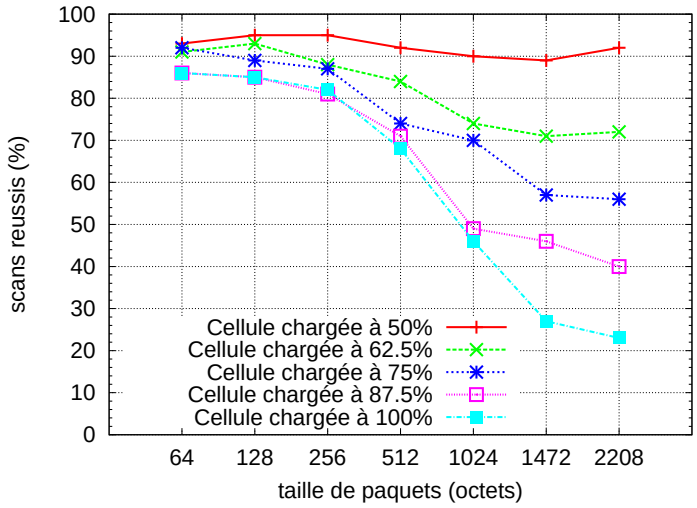


FIG. 7.11.: Taux de réussite des scans en fonction du volume de trafic

si elles sont envoyées en unicast, peuvent aussi entrer en collision. Au bout d'un nombre de retransmissions, leur émetteur va renoncer à leur envoi et signalera une erreur dans le déroulement du handoff. Un autre facteur qui peut empêcher les procédures de scan et handoff de se dérouler correctement sont les queues présentes dans le pilote HostAP et dans le firmware des cartes.

Nous avons observé le déroulement d'un handoff dans le cas où la station ou le point d'accès sont en train d'émettre un gros trafic de données. Nous avons modifié la taille des queues du pilote et du noyau Linux pour que le délai de transmission des trammes devienne très important. Ce délai est mesuré à l'aide de l'outil *ping* et des paquets ICMP *Echo Request* et *Echo Response*. Les résultats observés dans la figure 7.12 nous montrent que les trames de management utilisés dans le handoffs ont le même délai de transmission ; dès que celui-ci dépasse 100ms, la réassociation échoue.

Nous ne nous sommes pas proposé d'intervenir à un si bas niveau et de modifier le fonctionnement du pilote ou du firmware, mais les résultats obtenus nous incitent à proposer quelques améliorations possibles dans ce contexte :

- Favoriser l'envoi de trames de management sur le medium, en leur accordant un priorité plus haute, comme c'est le cas pour les trames de contrôle ;
- Donner la priorité aux trames de management dans les queues du pilote et de la carte, pour qu'elles ne soient pas retardées par l'envoi de trames de données ;
- Une amélioration pourrait être réalisée aussi au niveau des trames *Probe Request*, qui sont envoyées en broadcast et donc sans acquittement demandé :
  - La station pourrait envoyer plusieurs trames *Probe Request* sur chaque canal, pour

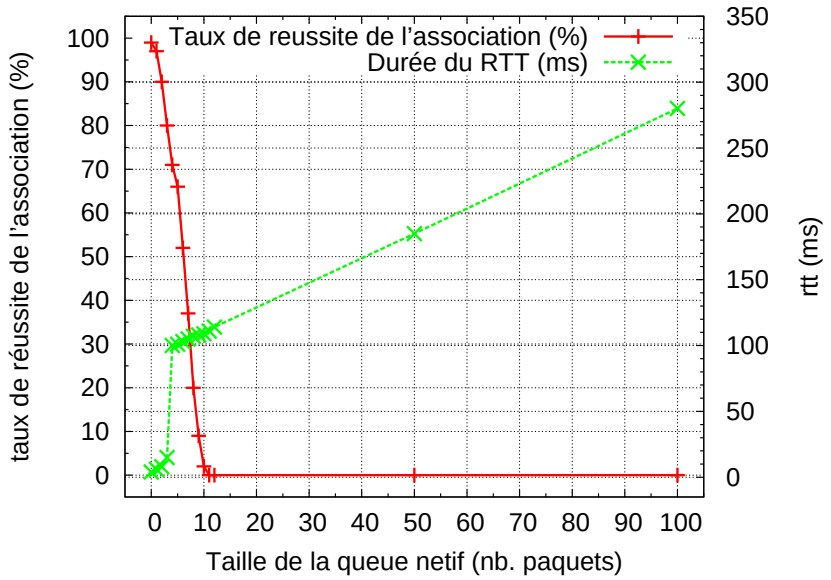


FIG. 7.12.: Taux de réussite de l'association dans une cellule chargée

augmenter les chances de réussite de transmission.

– Si la station connaît déjà une partie des points d'accès voisins, on peut introduire des scans ciblés, pour les AP connus. Ainsi, la station pourrait envoyer la trame *Probe Request* en *unicast* à destination d'un seul AP, uniquement sur le canal d'opération de celui-ci. Ceci réduirait le temps de ce scan à la valeur de *MaxChannelTime*.

7.4.2. Nouvelle extension : réassociation directe

Les temps très importants enregistrés par les phases de détection et de scan nous ont emmené à développer une nouvelle extension de type *Wireless Extensions*. Nous l'avons appelée *join request* (demande d'association), car elle permet à l'utilisateur de spécifier l'adresse MAC et le canal d'opération d'un certain AP et de demander directement l'association à celui-ci.

Cette extension a comme résultat immédiat le changement de la fréquence de la station sur le canal indiqué et la transmission d'une trame d'authentification à destination de l'AP respectif. Si la station et le point d'accès sont suffisamment proches pour que la trame soit correctement reçue, la séquence d'authentification et de réassociation continue et la station s'associée au nouveau point d'accès. Nous rappelons que le délai pris par cette dernière phase de handoff est particulièrement faible, de moins de 10 millisecondes. Nous avons observé, tout de même, un délai supplémentaire d'encore 10 ms entre l'issue de la commande de réassociation directe dans l'espace utilisateur et le moment où la première trame est

envoyée sur le medium (voir la figure 7.13).

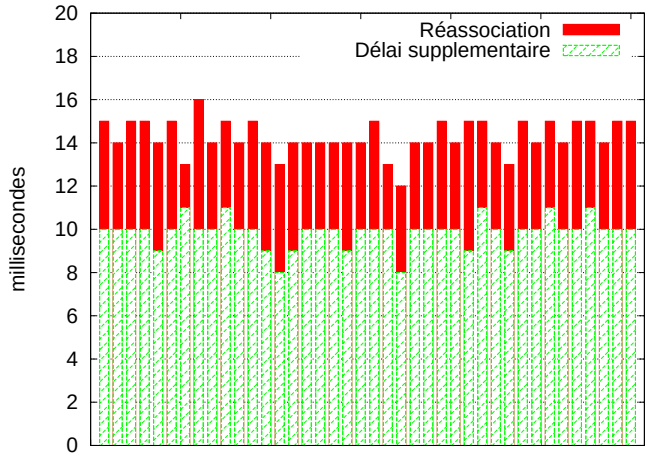


FIG. 7.13.: Délai de la réassociation directe

Le grand avantage de cette extension est qu'on peut complètement découpler la phase de la réassociation des phases de détection et de scan. Malgré cela, une difficulté importante apparaît : l'utilisateur doit connaître l'adresse MAC du point d'accès ciblé, ainsi que le canal sur lequel il fonctionne. En plus, la station doit savoir aussi le moment où elle est suffisamment proche de l'AP pour que l'échange de trame réussisse.

Nous considérons qu'une station doit toujours continuer à effectuer des scans, mais la périodicité de ces scans doit être en fonction du degré de mobilité de la station. Ainsi, une station qui ne se déplace pas a besoin d'un seul scan pour sa première association ; une station plus mobile devra effectuer des scans périodiquement. Le moment choisi pour effectuer ces scans périodiques doit être choisi aussi en fonction du trafic de données actuel de la station, en priorité quand que le trafic de données est nul ou moins important.

Nous avons précisé auparavant que les réseau sans-fil sont en général déployés en utilisant des canaux qui n'interfèrent pas (par exemple les canaux 1,6 et 11). Si lors d'un scan on écoute que deux ou trois canaux au lieu de 11 ou 13 on réduit considérablement l'interruption, et de 500 ms on peut arriver à 150–200 ms. Plus encore, on peut scanner q'un seul de ces trois canaux à la fois, avec des interruptions plus brèves, de l'ordre de 50 ms, plus acceptables pour les applications multimédia. Une amélioration supplémentaire que nous proposons à ce but est un nouvel échange entre le point d'accès et ses stations associés, par lequel l'AP notifie à ses clients la liste de canaux des points d'accès faisant partie d'un réseau sans-fil.

7.5. Équilibration de la charge dans des cellules 802.11 superposées

7.5.1. Analyse des réseaux 802.11 deployés

Nous sommes parti d'une série d'observations issues de plusieurs analyses des réseaux 802.11 déployés. Celles-ci ont révélé le fait que les utilisateurs des réseaux locaux sans-fil ont tendance à se regrouper dans certains endroits bien particuliers pour diverses raisons de situtaion :

- À courte distance des prises d'électricité dans les salles de conference ;
- À proximité des machines à café dans les universités et les bureaux ;
- À côté des portes d'enregistrement et de départ dans les aéroports ;

Ainsi, dans [103], les auteurs ont étudié les 177 points d'accès et plus d'un millier de stations clientes d'un réseau sans-fil d'une entreprise commerciale. 40% des points d'accès n'ont jamais eu plus de dix clients associés en même temps, alors que ceux situés dans les salles de réunion ou à proximité de cafétérias ont eu plus de trente stations associées simultanément. L'analyse faite au niveau de chaque station a relevé le fait que seulement un pourcentage de 20% des utilisateurs est responsable pour plus de 40% du trafic total observé.

Dans [104], on présente l'analyse d'un campus universitaire, avec plus de deux milles utilisateurs et 476 points d'accès repartis dans une centaine de bâtiments différents. Le plus grand nombre d'utilisateurs connectés à un seul point d'accès est le plus souvent dans des amphithéâtres de cours. Par contre, le trafic le plus important a été détecté pour des points d'accès avec relativement peu de stations associées, mais situées dans les résidences universitaires. Le fait que la charge des points d'accès ne soit pas en corrélation directe avec le nombre de stations associées est confirmé aussi dans [105]. L'analyse des quatre points d'accès installés dans une salle, à l'occasion de la conférence ACM SIGCOMM'01, montre que pour un nombre égal d'utilisateurs associés simultanément, les débits des quatre AP diffèrent considérablement, avec des écarts de l'ordre de 37%.

7.5.2. Distribution de stations dans le standard 802.11

Pour faire face à la forte affluence de stations ou au fort trafic dans certaines zones, on déploie souvent plusieurs points d'accès au même endroit. Les fréquences utilisées n'interfèrent pas et le but de ce type de déploiement est d'augmenter le débit total offert. Cependant, si la position de ces points d'accès n'est pas exactement la même, on n'arrive toujours pas à une distribution égale dans le nombre des stations associées. De plus, les analyses présentées ci-dessus nous ont montré le fait que le trafic d'une cellule n'est pas forcement en relation directe avec le nombre de stations présentes dans la cellule.

Le standard 802.11 contient un mécanisme qui pourrait être utile pour les situations quand le trafic d'une cellule s'approche de sa limite maximale. En effet, comme nous avons pu le voir dans la section 7.2.2, l'indicateur de la qualité du lien entre une station et son point d'accès courant peut diminuer s'il y a des pertes dans la transmission de trames. Un fort trafic dans la cellule pourrait être synonyme de la perte de beacons, de trames de données ou de leur acquittement. Si la qualité du lien descend en dessous d'un seuil critique, la station détecte la nécessité d'effectuer un scan et éventuellement initier un handoff. Cependant, la baisse de cet indicateur se fait graduellement, et la bonne réception d'une trame le fait augmenter rapidement vers la valeur du niveau de signal.

Même si on arrive à l'initiation d'un scan, le choix du point d'accès ciblé pour le handoff est toujours faite en fonction du niveau du signal reçu des points d'accès. Si le signal du point d'accès courant est meilleur que les autres, il sera toujours choisi, ce qui empêche le transfert de la station vers un autre point d'accès moins chargé. La seule possibilité est que le point d'accès courant n'apparaît pas parmi les résultats du scan. Ceci reste quand même une exception, et les mesures présentées dans la section 7.4.1 montrent que cela arrive seulement si la charge de la cellule est très proche de sa capacité maximale (aux environs de 90–95%).

7.5.3. Notre proposition de distribution de charge

Nous proposons un protocole de distribution de charge qui permettra à une station de faire usage de la réassociation directe (voir la section 7.4.2) pour se transférer entre les points d'accès du son voisinage.

L'algorithme de distribution de stations entre les points d'accès doit se baser sur la charge des cellules à la place du niveau de signal reçu. Cependant, nous estimons que le niveau de signal reste une caractéristique importante. Une faible valeur de celui-ci introduit un taux d'erreurs et de retransmissions élevé, donc une utilisation sub-optimale du medium. Dans les implémentations des stations 802.11 que nous avons testé, la perte de trois paquets successifs détermine une baisse dans le débit de transmission de la station, de 11 Mbps à 5,5, 2 ou 1 Mbps. Une découverte récente[106] démontre le fait que cette baisse pénalise non seulement la station en cause, mais toutes les autres communication dans la cellule.

En conclusion, il y a un compromis à faire entre le niveau de signal et la charge de la cellule dans le choix d'un point d'accès. Il est difficile de calculer une formule exacte qui tienne compte de ces deux facteurs. Dans nos expérimentations, nous avons établi un niveau minimum de signal[7] pour lequel le taux d'erreur reste raisonnable. Si le niveau de signal d'un AP est en dessous de ce seuil, l'AP respectif ne sera pas choisi comme cible d'un handoff.

[7] En pratique, les niveaux de signal présentés par les stations dépendent fortement du matériel utilisé (AP et clients), donc nous ne pouvons pas préciser une valeur généralement valable.

Calcul de la charge

Chaque AP doit observer en permanence le trafic transité dans la cellule et sur la base de ces informations calculer continuellement sa charge. Celle-ci est exprimée par le trafic total observé dans la cellule, rapporté au trafic maximum possible. Pour le calcul du trafic, il existe plusieurs possibilités, en fonctions de la couche réseau inspectée. Dans nos expérimentations, nous avons choisi d'utiliser les informations fournies par la pilote HostAP des cartes sans fil Prism. Par le biais du fichier dans `/proc/net/hostap/wlan0/stats`, il offre des statistiques sur l'activité de l'interface sans-fil, y compris le nombre d'octets transmis et reçus sur l'interface. Utilisant ces informations, nous avons crée et installé des daemons sur les points d'accès logiciels HostAP, qui calculent chaque seconde le trafic de la cellule et déterminent ainsi son niveau de charge.

Distribution de l'information

Les daemons installés sur les points d'accès observent la charge courante et lorsque celle-ci change significativement (avec $\pm$ 5%), le point d'accès respectif diffuse un message de notification destiné aux autres AP. De ce fait, chaque point d'accès a une vue globale de la distribution de la charge dans le réseau sans-fil. Les messages de notification doivent contenir aussi les informations nécessaires aux stations pour le handoff : l'adresses MAC et le canal sur lequel l'AP fonctionne.

Distribution de stations et équilibration de charge

Le problème le plus difficile est la mise en place effective de la distribution des stations en vue d'une équilibration de la charge. Plusieurs questions se posent : Qui prend les décisions ? À quel moment ? Quelle station sera choisie pour le handoff, et quel sera l'AP cible ?

Nous avons considéré une approche distribuée, où la décision des éventuels transferts est prise au niveau des points d'accès. Ils sont en mesure d'observer l'évolution de leur niveau de charge en rapport avec les AP voisins. Ils ont aussi accès aux statistiques sur le trafic de chaque station associée. La seule information qui manque regarde le voisinage de chaque station associé, car des stations peuvent être dans le rayon d'un autre AP sans que les deux AP soient directement atteignables. Si une coopération totale entre les points d'accès et toutes les stations associées existait, on pourrait imaginer un très efficace algorithme de distribution de la charge.

Cependant, il est difficile d'envisager d'avoir le contrôle de toutes les stations associées. De ce fait, un protocole de distribution de la charge où les décisions sont centralisées au niveau des points d'accès, ou encore plus, par une seule entité, devient inefficace. Ceci parce que la seule possibilité de provoquer le transfert d'une station non-coopérante est de lui envoyer un message de de-association et de lui répondre négativement aux tentatives subséquentes d'association. Les tests que nous avons réalisés avec plusieurs types de cartes nous ont montré que la station réessaie la réassociation au même AP avant d'initier un scan

et choisir un autre point d'accès. Les délais du handoffs subits par ces stations sont alors de plus de 500 ms, pouvant attendre même une seconde.

Pour ces raisons, notre approche déplace la décision des handoffs au niveau des stations. Nous expliquons son fonctionnement par l'exemple illustré dans la figure 7.14. Nous avons simulé un trafic multimédia entre le point d'accès AP_1 et la station STA_1 : paquets UDP d'une taille de 128 octets, débit obtenu 512 kbps. À l'instant t_0, une deuxième station STA_2 s'associe dans la cellule et commence un gros téléchargement par *ftp*. La débit total dans la cellule dépasse le seuil maximal et l'AP_1 notifie les stations, en précisant qu'un autre point d'accès, AP_2, n'est pas chargé. Seule la station STA_1 sait interpréter le message et à l'instant t_1 décide d'initier un handoff vers l'AP_2. Pour ce faire, elle utilise la demande de réassociation directe, car un scan effectué auparavant indique l'AP_2 dans son rayon de couverture. Le handoff ne prend que 15 ms et l'évolution des deux flots de données est montrée la figure 7.15.

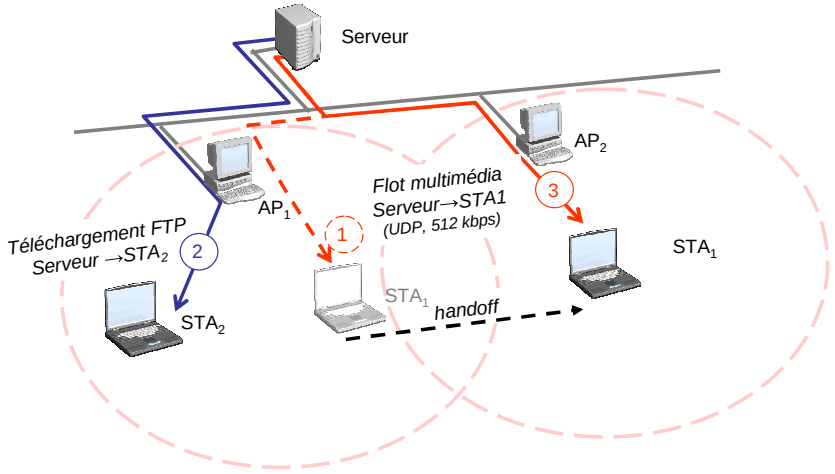


FIG. 7.14.: Configuration de test

7.6. Conclusion

Les réseaux sans fil, de de plus en plus rencontrés, offrent le cadre nécessaire pour la mobilité locale des terminaux portables. Nous avons commencé ce chapitre en détaillant le protocole 802.11 et ses mécanismes de mobilité : détection de la dégradation de la qualité du lien, scan, et réassociation. Ensuite, nous avons présenté des tests et des mesures des performances lors de scans et de handoffs. Pour cela, nous avons utilisé des cartes réseau sans fil Prism et le pilote Linux HostAP. Nous avons introduit une extension qui permet à une station de découpler les phases de scanning et de réassociation.

Le temps très faibles des handoffs faits par cette nouvelle extension de réassociation directe font que la pénalisation d'un transfert inter-cellules devienne peu importante. Nous avons proposé comme application la possibilité de faire des handoffs volontaires entre les

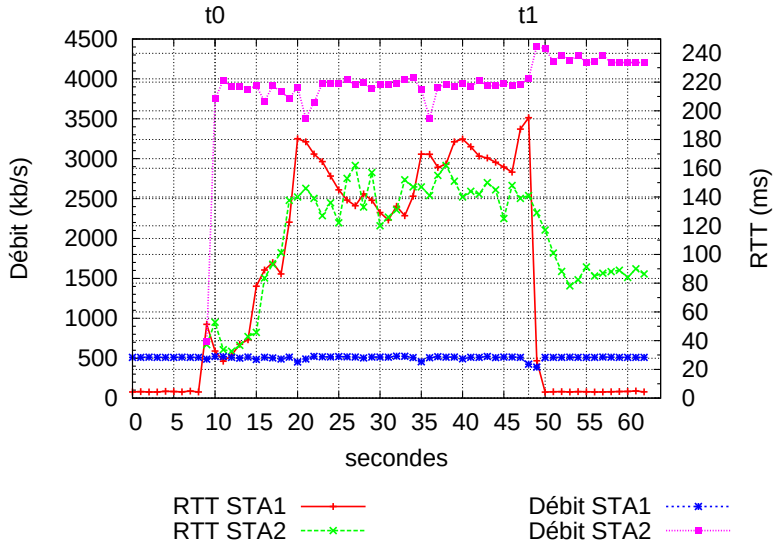


FIG. 7.15.: Équilibration de charge : évolution du débit et de la latence

cellules d'un réseau sans-fil étendu (ESS) pour équilibrer la charge des points d'accès déployés dans une même zone.

8. Gestion de la mobilité locale au niveau IP

Ce chapitre contient la présentation de notre protocole de micro-mobilité au niveau IP, conçu pour utiliser le support du handoff rapide 802.11 présenté dans le chapitre précédent. Le domaine de mobilité locale comprend un ensemble de points d'accès qui forment un réseau sans fil étendu 802.11. Ces points d'accès sont interconnectés par une structure des équipements réseau de niveau 2 et 3, y compris donc des routeurs. Ceci permet l'existence des cellules qui font partie de sous-réseaux IP différents.

Nous avons choisi d'implémenter un schéma d'acheminement local de datagrammes utilisant des routes d'hôte, similaires à celles employées dans HAWAII[57]. Ce type de schéma renonce aux tables de routage statiques usuelles qui contient des entrées basées sur les préfixes des adresses IP (c'est-à-dire la partie sous-réseau d'une adresse IP). Dans notre protocole, l'hôte mobile communique avec les routeurs et envoie des messages de mise à jour qui peuvent insérer, modifier ou effacer des entrées dans les tables de routage. Ces messages de mise à jour créent un chemin temporaire entre le point d'entrée du domaine et le sous-réseau courant de l'hôte mobile. Celui-ci peut ainsi garder inchangée son adresse IP. Avoir une adresse stable a comme avantage immédiat le fait que les protocoles de niveau transport et les applications continuent à fonctionner.

8.1. Le domaine de micro-mobilité

8.1.1. Entités présentes dans la topologie

On peut identifier plusieurs types de nœuds présents dans un domaine de mobilité locale. Nous allons les définir par la suite :

- *MR (*Mobility Router*)* – Les routeurs appartenant au domaine de mobilité locale qui participent activement à l'acheminement de datagrammes vers les hôtes mobiles. Les MR communiquent entre eux en s'échangeant des messages de contrôle et de mise à jour des tables de routage. Les mises à jour créent des chemins de routage pour chaque hôte mobile. Ils continuent cependant de remplir la fonction d'un routeur régulier et faire transiter des datagrammes des données destinés aux autres hôtes du domaine.
- *GMR (*Gateway Mobility Router*)* – Le routeur de bordure du domaine de mobilité. Il représente la limite supérieure du domaine et donc le point d'interconnexion de celui-ci avec le reste de l'Internet. Il a le même fonctionnement qu'un simple MR, la seule contrainte étant qu'il ne doit pas retransmettre les messages de contrôle sur le lien

montant vers l'Internet.

- *MR-AP (*Mobility Router - Access Point*)* – Les routeurs les plus proches des hôtes mobiles. Ils ont une fonction supplémentaire, car ils sont doués d'une interface réseau sans-fil qui les relie directement aux hôtes mobiles. Ils représentent donc le point d'accès des hôtes mobiles et en même temps leur passerelle réseau sur le lien local. Ils communiquent directement avec les hôtes mobiles et sont les premiers à recevoir les demandes de mise à jour de routes de la part de celles-ci.

  Dans un réseau sans-fil usuel, ces routeurs sont connectés à des équipements physiques de type point d'accès par un lien filaire Ethernet. Dans ce cas, les points d'accès fonctionnent comme des ponts de niveau 2, reliant la cellule sans-fil à l'infrastructure filaire. Dans notre implémentation, nous avons utilisé des points d'accès logiciels. Ces derniers sont matérialisés par une machine PC équipée d'un carte réseau Prism et du pilote Linux HostAP[100]. Ceci permet aux MR-AP d'avoir un accès direct aux statistiques et à l'interface de contrôle du lien sans-fil. De ce fait, la mobilité au niveau IP pourra être mieux intégrée avec le handoff 802.11.

- *MH (*Mobile Host*)* – Les hôtes mobiles équipés d'une interface sans-fil. Ils se déplacent entre les points d'accès MR-AP et se servent du protocole de micro-mobilité pour pouvoir garder inchangées leurs adresses IP.

8.1.2. Topologie du domaine

La topologie standard de notre domaine de micro-mobilité est présentée dans la figure 8.1. Elle est caractérisée par quelques particularités que nous avons imposé pour simplifier dans un premier temps la conception du notre protocole :

- Il y a *un seul routeur de bordure* dans le domaine ;
- La topologie est structurée d'une façon *hiérarchique* ;
- Tous les liens entre les routeurs sont des *liens point-à-point* ;
- La structure étant hiérarchique et formée des liens point-à-point, on peut classifier les interfaces des routeurs dans des liens montants (*uplink*) et descendants (*downlink*) ; *il n'y a pas de liens croisés (crossover link)* qui traversent l'arbre ;
- La topologie logique coïncide parfaitement avec la *topologie physique* des nœuds présents dans le domaine de micro-mobilité. Cela signifie que tous les nœuds physiques représentent des MR, GMR, MR-AP ou MH dans la structure logique.

Les contraintes que nous avons imposées aident à simplifier la conception du protocole de micro-mobilité. Néanmoins, des variations de cette topologie peuvent exister. Nous allons présenter ci-dessous les possibles transformations de la topologie standard. Nous allons voir ensuite si elles peuvent être compatibles avec notre protocole et quelles sont les fonctions

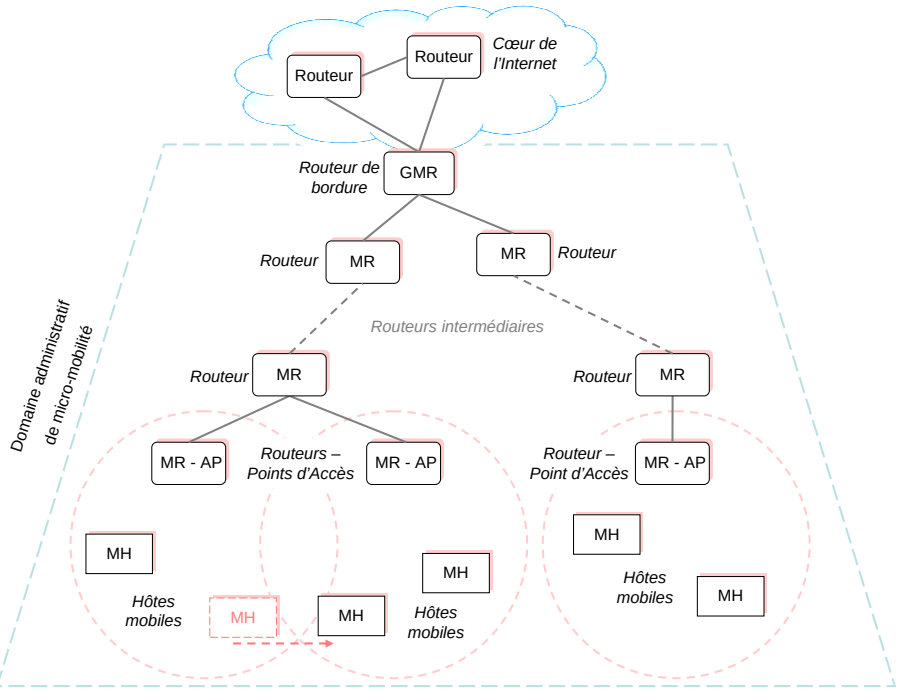
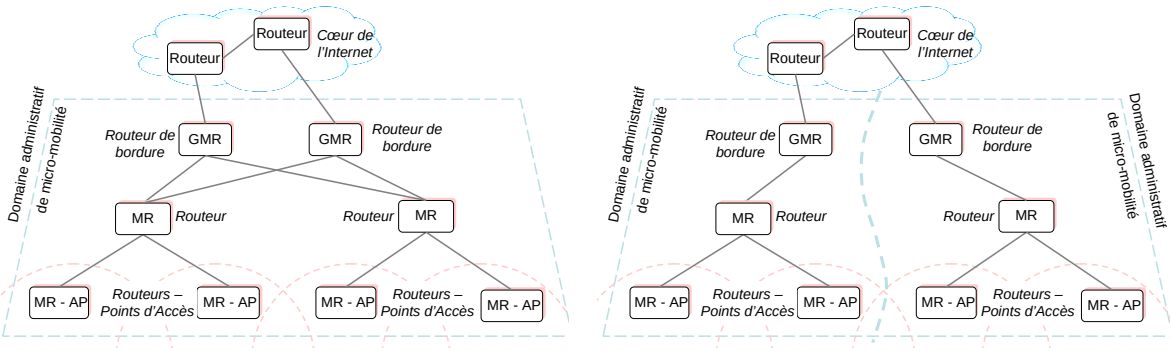


FIG. 8.1.: La topologie standard du domaine de micro-mobilité

nécessaires pour les supporter.

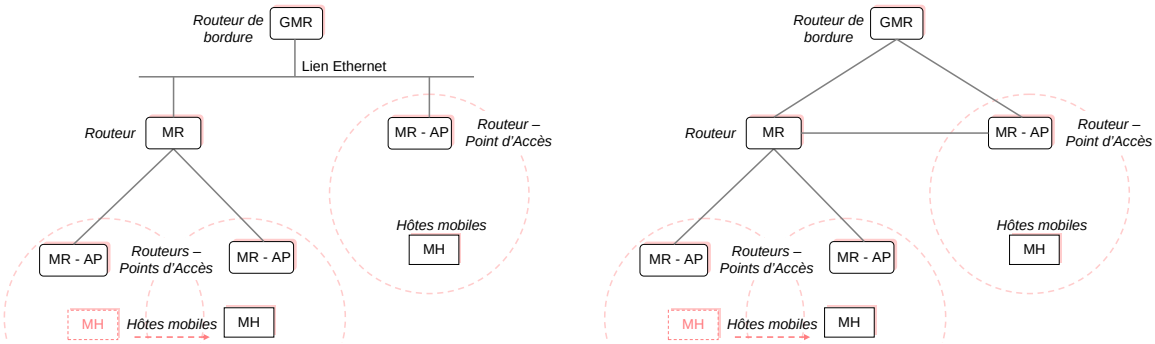
- Plus d'un routeur de bordure. Nous avons illustré dans la figure 8.2 deux cas possibles : un domaine avec deux GMR est celui où on peut atteindre toutes les MH à partir de chaque GMR (figure 8.2a) ; en cas contraire, le domaine peut être subdivisé en deux domaines séparés (figure 8.2b) ;



(a) Deux routeurs de bordure d'un même domaine (b) Deux routeurs de bordure qui divisent le domaine

FIG. 8.2.: Domaines de mobilité avec plusieurs routeurs de bordure

- La connexion physique des routeurs du domaine n'est pas réalisée avec seulement des liens point-à-point. Dans l'exemple illustré dans la figure 8.3a, plusieurs nœuds sont connectés par un réseau de type Ethernet : le GMR et ses deux descendants MR_1 et $MR - AP_3$. Cela crée un lien croisé et la boucle $MR_1 \leftrightarrow GMR \leftrightarrow MR - AP_3$.



(a) Plusieurs MR sur un même lien Ethernet

(b) Lien point-à-point traversant l'arbre

FIG. 8.3.: Topologies contenant des boucles

- Une boucle apparaît aussi dans le cas d'une lien croisé entre deux MR éloignés (figure 8.3b). Nous allons concevoir le protocole de micro-mobilité pour permettre la propagation de mises à jour et l'acheminement des datagrammes aux hôte mobiles dans une topologies avec boucles.

8.1.3. Organisation et configuration des nœuds

Le point suivant qu'on discute est lié à l'organisation et à la configuration des nœuds dans la topologie logique. Il y a deux parties différentes :

L'organisation et configuration des routeurs

Normalement, l'ensemble des routeurs qui forment l'arbre d'acheminement est stable, sans qu'il y ait des éléments qui apparaissent ou disparaissent fréquemment. La même supposition est valable pour l'organisation de ces routeurs dans la topologie logique. Toutefois, pour ne pas compromettre le fonctionnement de notre protocole en cas d'une modification mineure de la topologie, nous avons conçu l'organisation et la configuration de routeurs pour qu'elle soit dynamique.

Ainsi, nous avons introduit une phase de mise en place (*bootstrap*) au démarrage des daemons de mobilité s'exécutant sur les routeurs. Chaque routeur diffuse, à destination de l'adresse de *broadcast* IP de chaque de ses interfaces, une requête qui permet de découvrir les autres MR. Les réponses envoyées par ceux-ci contiennent intrinsèquement, dans l'adresse source du datagramme, l'information nécessaire pour configurer les MR voisins.

Cependant, cette découverte automatique de routeurs voisins ne fonctionne entièrement que pour la topologie simplifiée présentée dans la section 8.1.2. Dans le cas des topologies contenant plusieurs routeurs sur un même lien Ethernet ou des boucles (voir la figure 8.3), une configuration manuelle devient nécessaire. Celle-ci peut être utile pour enlever des liens dans la structure logique et éliminer ainsi les boucles.

La configuration de la communication entre les points d'accès et les hôtes mobiles

Dans la communication entre les routeurs d'accès sans-fil (MR-AP) et leurs clients mobiles (MH), la configuration doit être dynamique : les clients peuvent se connecter et se transférer fréquemment d'un point d'accès à un autre. Nous n'avons donc pas prévu une configuration manuelle ; à la place, nous proposons d'utiliser les mécanismes standard de configuration automatique des protocoles IPv4 et IPv6. Une fois que le MH apprend l'adresse IP de son point d'accès, ils peuvent s'échanger des informations propres au protocole de micro-mobilité.

8.2. Conception du protocole

Coexistence du routage traditionnel et des routes d'hôte

Dans le domaine de micro-mobilité, on pourrait imposer que toutes les routeurs du domaine n'utilisent que des routes d'hôte. Cependant, les entrées standard des tables de routage, basées sur les préfixes des sous-réseaux, permettent la présence d'hôtes qui n'utilisent pas le protocole de mobilité. Pour cette raison et dans le but d'élargir le champ d'application de notre protocole, nous avons préféré la coexistence de deux types de routes.

Configuration des adresses des hôtes mobiles

Normalement, les routeurs IPv6 diffusent sur le lien local des messages de configuration automatique, contenant le préfixe du sous-réseau ainsi que leurs adresse IP. Ces informations sont utilisées par les hôtes présents sur le lien pour s'auto-configurer une adresse globale IPv6 basée sur ce préfixe et pour utiliser le routeur respectif comme passerelle par défaut. La même chose est possible en IPv4, si le terminal mobile utilise DHCP et le routeur, configuré aussi comme serveur DHCP, lui répond en lui attribuant une adresse IPv4 locale.

Ceci peut interférer avec notre protocole et les hôtes mobiles qui l'utilisent, car son principe de base est que l'hôte mobile garde une seule et même adresse IP dans tout le domaine. Deux solutions sont envisageables :

– Soit les hôtes mobiles n'utilisent pas l'auto-configuration IPv6 et le DHCP ;

– Soit les hôtes mobiles acquièrent automatiquement une nouvelle adresse, mais cette-ci n'est utilisée que pour les nouvelles connexions. Ils gardent et se servent de l'ancienne adresse pour les connexions ouvertes auparavant.

Nous avons choisi la deuxième solution, pour des raisons d'optimisation de la communication avec les autres hôtes du domaine, et principalement avec ceux du lien local. L'hôte mobile se voit configurer à chaque fois une nouvelle adresse, topologiquement correcte, car correspondant au lien où est elle connectée actuellement. Elle gardera aussi les anciennes adresses, jusqu'au moment ou elles ne seront plus utilisées dans les connexions ouvertes. Nous allons détailler dans la section 8.3 la façon dont deux ou plusieurs adresses peuvent coexister.

Communication des hôtes mobiles avec les hôtes situés sur le même sous-réseau

Avoir une adresse appartenant au lien local simplifie la communication avec les autres hôtes situés sur le même sous-réseau, car l'envoi des messages se fait directement entre les hôtes concernées, sans passer par un routeur. Pourtant, le problème majeur que ce choix introduit apparaît seulement après que l'hôte mobile se déplace sur un autre sous-réseau. En effet, en se basant sur l'adresse destination et sur leur table locale de routage, les hôtes d'un même sous-réseau s'échangent des paquets directement sur le lien local. En cas du déplacement vers un autre réseau, l'hôte mobile et ses anciens correspondants locaux ne sont plus en mesure de se retrouver, car ils ne seront plus sur le même lien local.

Ce problème peut être résolu par une combinaison de deux mécanismes spéciaux du protocole ARP[42] : *gratuitous ARP* et *proxy ARP*. Après que l'hôte mobile se déplace, le routeur local répond aux requêtes ARP à la place de celui-ci et intercepte toutes les datagrammes qui sont destinés à l'hôte mobile. De son coté, l'hôte mobile élimine de sa table locale de routage l'entrée correspondante à l'ancien sous-réseau. De ce fait, il va passer par son nouveau routeur, pour envoyer des paquets aux hôtes de son ancien sous-réseau. Nous avons illustre ce fonctionnement dans la figure 8.4.

Communications avec les autres hôtes du domaine.

Une partie importante des communications de l'hôte mobile est faites avec les autres hôtes du domaine local où elle se trouve. Dans les communications avec le reste du domaine, à part les hôtes du lien local, le but et d'optimiser le plus possible l'acheminement des datagrammes dans le domaine. Ainsi, on doit éviter que toutes les datagrammes passent par un point unique, comme le routeur de bordure ou un des anciens MR-AP de l'hôte mobile.

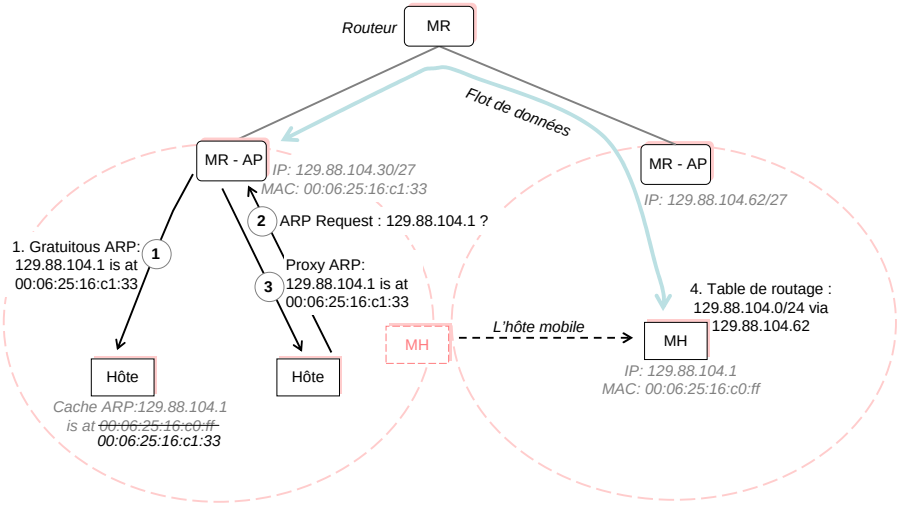


FIG. 8.4.: La communication avec les hôtes de l'ancien sous-réseau

8.3. Fonctionnement du protocole

Notre protocole de micro-mobilité a été conçu pour être le plus simple possible, et en même temps compatible avec toutes les configurations possibles d'un domaine de mobilité locale. Ainsi, il permet la compatibilité avec des topologies qui incluent des boucles ou plusieurs routeurs de bordure. Pour avoir toujours un chemin de routage directe, les mises à jour doivent sont propagées dans tout le domaine, est non seulement jusqu'à un routeur de croisement ou le routeur de bordure. Aussi, nous avons essayé de garder et utiliser le plus possible le mécanisme de routage classique. L'hôte mobile va se configurer toujours comme adresse principale une adresse topologiquement correcte, propre à son sous-réseau actuel. Elle gardera des anciennes adresses pour continuer les connexions en cours, cas où notre protocole de micro-mobilité va se superposer sur le routage traditionnel.

8.3.1. Première connexion dans le domaine

Pour sa première connexion dans le domaine, le mobile doit configurer les adresses IPv4 et IPv6 pour son interface sans-fil. Nous nous sommes appuyé sur les mécanismes d'auto-configuration déjà existantes : DHCP[14] pour l'IPv4 et l'auto-configuration pour l'IPv6[17]. Par des paquets transmis par le point d'accès local (MR-AP), l'hôte mobile configure ses adresses IP, que la passerelle par défaut pour les communications en dehors du lien local.

Nous précisons que des routes correspondantes aux différents sous-réseaux sont installés dans les routeurs du domaine. Ainsi, aucun autre message de configuration n'est nécessaire dans le reste du domaine, car les routeurs assurent l'acheminement des paquets adressés à l'hôte mobile vers son sous-réseau.

8.3.2. Décision du handoff et choix de l'AP cible

L'hôte mobile va commencer à évaluer l'opportunité d'un transfert vers une autre cellule si la qualité du lien sans-fil se dégrade ou si le débit dans la cellule s'approche d'un seuil maximal (conformément à la section 7.5.3 du chapitre précédent).

Pour établir une liste des points d'accès voisins vers lesquels un transfert est envisageable, la station doit effectuer de temps en temps de scans. Comme nous l'avons évoqué dans la section 7.4.2, ces scans doivent être faits en fonction du degré de mobilité de la station, et de préférence quand il n'y a pas de trafic de données.

Immédiatement après sa connexion, et chaque fois qu'un résultat d'un nouveau scan est disponible, le mobile informe le point d'accès courant sur les points d'accès de son rayon d'action. Parallèlement, l'ensemble des points d'accès du domaine se tiennent informés réciproquement du niveau de charge actuel de leurs cellules. Ainsi, après la réception d'une liste de points d'accès voisins d'une station, son point d'accès lui retourne le niveau de charge des points d'accès respectifs, ainsi que leurs adresses IP.

Disposant de ces informations, une station peut à un certain moment décider d'effectuer un handoff vers une cellule voisine[1]. En fonction de l'adresse IP du point d'accès choisi comme cible du handoff, l'hôte mobile détermine si la nouvelle cellule fait ou non partie d'un sous-réseau IP différent. Si oui, il doit préparer, parallèlement au handoff physique, le handoff au niveau IP.

8.3.3. Envoi et propagation des mises à jour

Si la nouvelle cellule appartient à un sous-réseau différent, l'hôte mobile se configurer des nouvelles adresses propres à ce sous-réseau. Ceci est fait immédiatement après le handoff physique et la réassociation sur le nouveau lien, par les mêmes mécanismes (DHCP, auto-configuration IPv6) évoqués plus haut.

Cependant, si l'hôte mobile a des connexions ouvertes qui utilisent son adresse courante, notre protocole de micro-mobilité va permettre à l'hôte de garder cette adresse dans le nouveau sous-réseau. Pour cela, on doit propager un message de mise à jour dans le domaine, pour que les paquets adressés à cette adresse soient redirigés vers le nouveau sous-réseau. Ce message doit en effet contenir non pas seulement l'adresse courante, mais toutes les adresses que l'hôte est en train d'utiliser. C'est le cas si l'hôte mobile a déjà effectué des handoffs et garde une ou plusieurs anciennes adresses comme adresses secondaires.

Une caractéristique importante de notre procédure de handoff au niveau IP est que l'hôte mobile connaît déjà l'adresse IP de son futur point d'accès. En conséquence, il peut préparer et initier le message mise à jour avant le handoff physique. Celui-ci sera exécuté seulement une fois que l'acheminement des paquets vers son nouvel emplacement est correctement mis en place. Nous présentons dans la figure 8.5 les messages échangés et les opérations réalisées lors d'un handoff :

[1] Possiblement suite à la suggestion de son point d'accès, voir la section 7.5.3

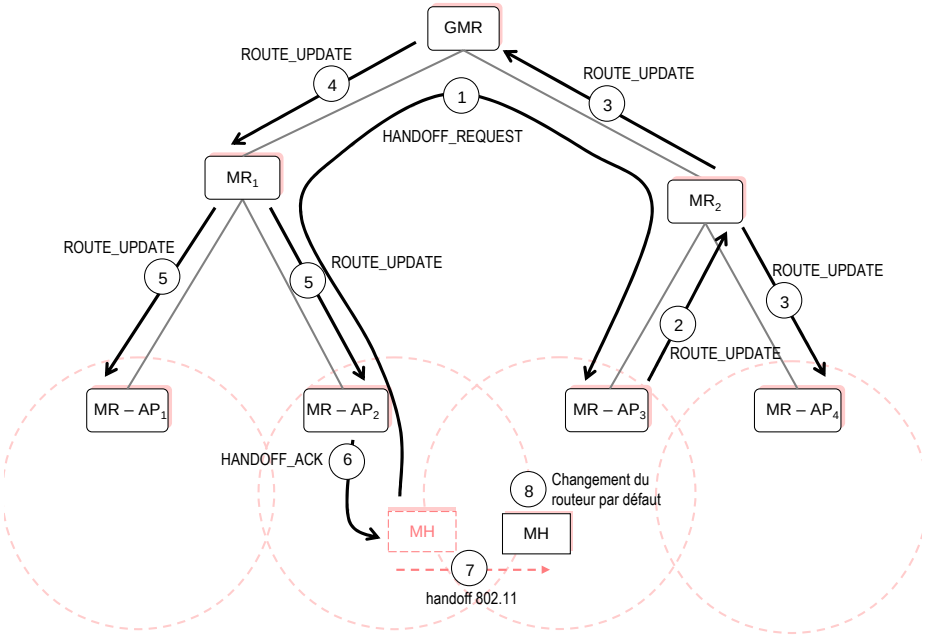


FIG. 8.5.: Les messages échangés lors du handoff

- 1. La demande d'un handoff au niveau IP se fait par un message HANDOFF_REQUEST, envoyé par le mobile à destination du nouvel AP ($MR - AP_3$). Ce message contient l'adresse MAC, ainsi que ses adresses IP en cours d'utilisation pour lesquelles il demande la mise en place des routes. Comme l'expérience des différents schémas de handoff de HAWAII et CIP nous l'a montré, seule la propagation des mises à jour à partir du nouvel AP garantit l'inexistence de boucles dans l'acheminement et une arrivée en ordre de datagrammes. C'est pour cela que l'hôte mobile envoie le message à destination du son nouvel AP ; toutefois, elle reste connecté à l'ancien AP jusqu'à la confirmation de la mise en place des routes.

- 2. À la réception de ce message, le nouveau point d'accès introduit (ou modifie, si c'est le cas) dans sa table de routage des entrées pour les adresses IP de l'hôte mobile. Ces entrées font que les datagrammes destinés aux adresses respectives soient acheminées sur l'interface sans-fil de l'AP. Le nouveau point d'accès envoie un message ROUTE_UPDATE_REQUEST qui contient les adresses de l'hôte mobile à son routeur voisin MR_2.

- 3. Ensuite, chaque routeur qui reçoit un message ROUTE_UPDATE_REQUEST de la part d'un routeur voisin fait des modification similaires dans sa table de routage. Ainsi, toutes les datagrammes destinés aux adresses de l'hôte mobile seront acheminées vers le MR voisin d'où le message est arrivé. Ensuite, il renvoie une copie du message à ses MR voisins, sauf celui d'où il a reçu le message (pour ne pas introduire une boucle infinie dans la propagation du message).

- 4-5. Le routeur de bordure ne renverra pas le message sur le lien montant vers l'In-

ternet. Ceci se fait automatiquement, car le GMR n'a pas de MR voisins sur ce lien. Le message ROUTE_UPDATE_REQUEST continue de se propager alors vers la partie inférieure de domaine. Cette propagation s'arrête aux points d'accès, car ils n'ont pas d'autres MR voisins sur l'interface sans-fil.

- 6. À un certain instant, le message ROUTE_UPDATE_REQUEST est reçu par le point d'accès courant de l'hôte mobile ($MR - AP_2$). Ceci signifie que la mise en place des routes a été correctement faite, au moins pour la portion critique entre l'ancien et nouvel point d'accès. La propagation dans le reste du domaine pourrait ne pas être finie, mais le changement de route vers le nouvel point MR-AP est déjà mis en place. Le point d'accès courant envoie donc un message de confirmation HANDOFF_ACK à l'hôte mobile. Ensuite, il met à jour dans sa table de routage les entrées correspondant aux adresses de l'hôte mobile. Il envoie aussi une trame *gratuitous ARP* dans la cellule, pour mettre à jour les éventuelles mises en cache qui font référence à l'adresse IP du l'hôte mobile.

- 7-8. Au moment ou l'hôte mobile reçoit le message HANDOFF_ACK, il initie le handoff physique au niveau 802.11, en utilisant l'extension de réassociation directe introduite par nous dans le chapitre précédent (section 7.4.2). Il supprime de sa table locale de routage le sous-réseau courant et configure le nouveau point d'accès comme passerelle par défaut. Dès que la réassociation physique abouti, il peut continuer ses anciens connexions. Ensuite, il utilise les services locaux d'auto-configuration pour se reconfigurer des nouvelles adresses locales IPv4 et IPv6 comme adresses principales. Ces adresses principales sont utilisés dans les nouvelles connexions ouvertes ensuite.

8.4. Implémentation, expérimentations et mesures

Dans cette section, nous allons présenter la plate-forme expérimentale utilisée pour le développement et le test de notre prototype de micro-mobilité. Nous avons construit un domaine de mobilité formé par les machines suivantes :

- Une machine fixe, *tenerife*, équipée d'un processeur AMD AThlon à 1400 Mhz et de 256 MB de mémoire RAM, avec Redhat Linux 8.0 (noyau 2.4.20) ;
- Deux machines fixes, *lanai* et *kauai*, équipées de processeurs Intel Pentium III à 500 Mhz et de 64 MB de mémoire RAM, avec Redhat Linux 8.0 (noyau 2.4.20) ;
- Deux machines portables, *crete* et *marie* équipées de processeurs Intel Pentium III à 800 Mhz et de 128 MB de mémoire RAM, avec Redhat Linux 8.0 (noyau 2.4.20).

La plate-forme expérimentale est illustrée dans la figure 8.6. *tenerife* fait office de routeur de bordure (GMR) et *kauai* et *lanai* sont des routeurs d'accès (MR-AP). Les trois machines sont interconnectées par un réseau Ethernet à 100 Mbps. Les MR-AP d'accès sont équipés de cartes sans-fil LinkSys WPC-11 utilisant le pilote HostAP en mode *master* pour fonctionner

comme points d'accès logiciels 802.11. La machine portable *crete* utilise la même carte sans-fil et le pilote HostAP, mais en mode *managed*, pour fonctionner comme une station client 802.11. Finalement, *marie*, équipée elle aussi d'une carte sans-fil LinkSys avec le pilote HostAP en mode *monitor*, est utilisée pour capturer les trames transmises dans les deux cellules sans-fil.

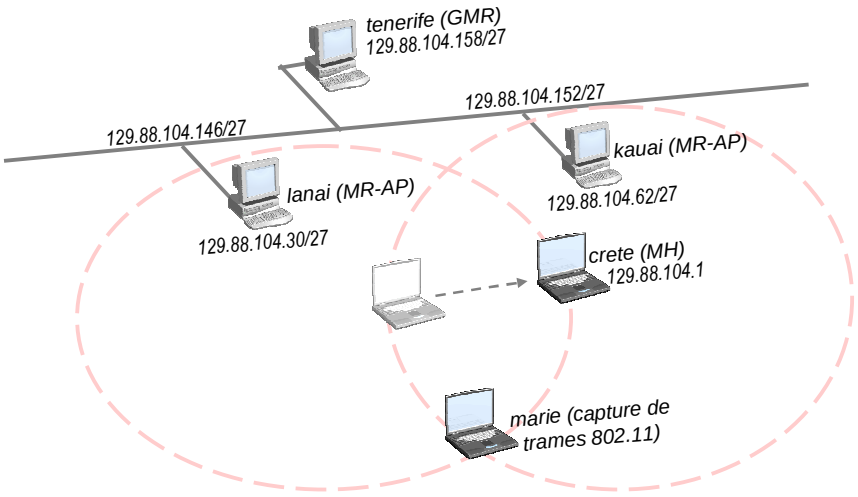


FIG. 8.6.: Configuration de la plate-forme de test

Sur les trois machines fixes ainsi que sur la machine client tournent des daemons de mobilité qui communiquent entre eux par des messages UDP envoyés sur un port bien connu (5500). Les messages envoyés sur le réseau sans-fil nécessitent un acquittement, car il existe la possibilité de perdre les messages. Pour garder la cohérence, même les messages envoyés dans l'infrastructure filaire vont être acquittés, quoique la perte des messages soit moins probable. Ces daemons de mobilité contiennent plusieurs fonctionnalités (voir la figure 8.7) :

- Sur *crete*, ils font usage du code logiciel présenté dans le chapitre précédent, section 7.3.1, pour interagir avec le pilote HostAP, recevoir des informations sur la qualité du lien courant et initier des scans et des handoffs au niveau 802.11 ;
- Sur *kauai* et *lanai* ils calculent la charge actuelle de la cellule, en termes de trafic total par seconde. Chaque fois que la charge de la cellule varie sensiblement ($\pm$ 5%), ils diffusent l'information actualisée dans l'infrastructure de routeurs. Quand l'autre MR-AP reçoit cette information, il informe sa station associée (*crete*) sur la distribution de charge dans le réseau sans-fil.
- Tous les daemons de mobilité interagissent avec les tables de routage et nécessitent d'être exécutés en mode privilégié (par l'utilisateur *root*). Pour modifier les tables de routage nous avons utilisé l'interface standard de contrôle *ioctl()* en lui donnant comme arguments une structure de type `struct rtentry` et les commandes `SIOCADDRT` (ajout d'une route) ou `SIOCDELRT` (suppression d'une route).

Pour l'IPv4, Le système Linux utilise une deuxième table de routage, nommée cache de routage. Le contenu de cette table peut être consulté en utilisant les commandes `route -C` ou `ip route list table cache`. La suppression ou la modification d'une entrée dans la table de routage devrait logiquement provoquer la mise à jour de la table cache. Nos premiers résultats nous ont montré que l'effacement de l'entrée du cache de routage n'intervient qu'après un délai de deux secondes. Ce délai vient des paramètres noyau pour les tables de routage, et plus particulièrement de `net.ipv4.route.min_delay` égal à 2 et de `net.ipv4.route.max_delay` égal à 10. Pour que les mises à jour des tables de routage soient immédiatement prises en compte dans l'acheminement des paquets, nous avons dû régler ces paramètres à 0.

- Pour optimiser la communication entre l'hôte mobile et les points d'accès, les daemons de ces machines doivent agir aussi sur les tables ARP. Nous avons utilisé la même interface *ioctl()* et les commandes `SIOCSIARP` et `SIOCDARP` pour l'IPv4. Par contre, pour l'IPv6, nous avons dû utiliser un autre mécanisme : des messages de type `RTM_NEWNEIGH` et `RTM_DELNEIGH` passés à *netlink*, une interface entre l'espace utilisateur et le noyau Linux.

8.4.1. Temps de vie et effacement des routes

Les entrées correspondant à des anciennes adresses d'un hôte mobile doivent avoir un temps de vie limité dans les routeurs du domaine. Pour cela, l'hôte mobile surveille l'état de ses sockets ouvertes, inspectant régulièrement le résultat fourni par l'outil standard Linux *netstat*. Dès qu'une ancienne adresse n'est plus utilisée, un message ROUTE_UPDATE_CANCEL contenant l'adresse respective est propagé dans le domaine.

L'effet de ce message est la suppression des entrées correspondantes dans les routeurs du domaine, ce qui permet d'avoir toujours une taille minimale de tables de routage de ceux-ci. En plus, au moment ou le message de d'invalidation des routes arrive au routeur d'accès dont l'adresse respective correspond, celui-ci pourra la réutiliser pour la ré-attribuer par DHCP à un autre hôte mobile.

8.4.2. Évitement de pertes de paquets

Les protocoles HAWAII et Celluar IP essayent d'éviter les pertes de paquets par un mécanisme de *double-cast* au routeur de croisement, ce qui signifie que celui-ci envoie pour une courte période de temps les paquets sur les deux chemins (ancien et nouveau) vers l'hôte mobile.

Notre approche est différente. Nous sauvegardons les paquets arrivant au nouveau point d'accès, jusqu'au moment où l'hôte mobile va s'associer à celui-ci. Dans notre prototype, nous avons utilisé un mécanisme très simple, présent dans les systèmes d'exploitation et qui fait partie du mécanisme standard ARP de découverte de l'adresse physique. ARP utilise une queue ou les paquets destinés à une machine voisine sont gardés jusqu'au moment où

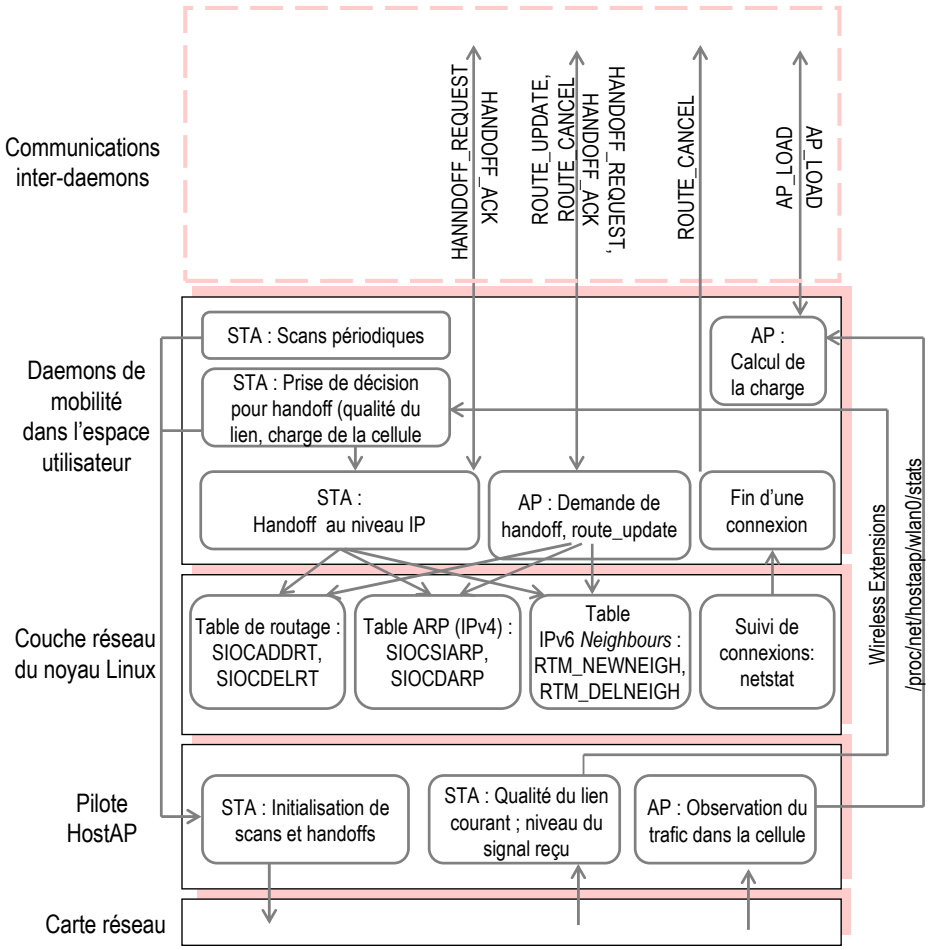


FIG. 8.7.: L'architecture des daemons de mobilité

la résolution d'adresse physique de cette machine aboutit. Une fois que le mobile achève le transfert physique vers la nouvelle cellule, le routeur finit la résolution et lui envoie les datagrammes mis en cache.

Le système d'exploitation Linux nous offre la possibilité de configurer quelques paramètres liés au mécanisme ARP de résolution d'adresses physiques. Ces paramètres sont accessibles par le système de fichiers dans le répertoire */proc* ou par l'outil de configuration de noyau *sysctl*. Les paramètres qui nous intéressent et leurs valeurs par défaut dans le noyau 2.4.20 sont :

- `net.ipv4(6).neigh.wlan0.mcast_solicit` et `ucast_solicit` : le nombre de fois qu'on envoie une trame *ARP Request* (en broadcast et respectivement unicast) pour résoudre une adresse IP vers une adresse physique (par défaut 3) ;
- `net.ipv4(6).neigh.wlan0.retrans_time` : l'intervalle (exprimée en dixièmes de se-

condes) entre deux retransmissions successives d'une trame *ARP Request*, si on ne reçoit pas de réponse (par défaut 100, c'est-à-dire une seconde);

- `net.ipv4(6).neigh.wlan0.unres_qlen` : le nombre maximum de paquets acceptés des couches supérieures et mises en cache pendant qu'on essaye la résolution ARP (par défaut 3).

Dans notre prototype, nous avons ajusté la taille du tampon à une valeur suffisante (100) pour mettre en cache tous les paquets pendant la durée d'un handoff. Ces modifications sont à faire sur les hôtes mobiles et les points d'accès, pour éviter la perte des paquets dans les deux directions du trafic. Les deux autres paramètres n'ont pas besoin d'être modifiés, car les daemons de mobilité agissent directement sur les tables ARP et y introduisent manuellement l'entrée nécessaire, immédiatement que la réassociation 802.11 est achevée.

8.4.3. Tests et mesures

Pour tester le fonctionnement des daemons et de notre protocole de mobilité, nous avons fait un premier test à l'aide de l'outil standard *ping*, avec l'option enregistrement de route (`ping -R`). L'hôte mobile *crete* est associé à *kauai* et exécute un ping vers le routeur de bordure *tenerife*. La trace d'exécution présentée dans la figure 8.8 est capturée lors d'un handoff de *crete* vers *lanai*. Elle nous montre que le changement et la propagation de route a bien lieu entre les différentes entités, aucun paquet n'étant perdu[2].

```
[root@crete ~]$ ping -R tenerife-104.imag.fr
PING tenerife-104.imag.fr (129.88.104.158) 56(124) bytes of data.
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=0 ttl=127 time=10.410 ms
RR:     crete.imag.fr (129.88.104.1)
        kauai-104.imag.fr (129.88.104.152)
        tenerife-104.imag.fr (129.88.104.158)
        kauai-wl.imag.fr (129.88.104.30)
        crete.imag.fr (129.88.104.1)

64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=1 ttl=127 time=7.656 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=2 ttl=127 time=6.714 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=3 ttl=127 time=6.821 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=4 ttl=127 time=6.726 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=5 ttl=127 time=5.677 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=6 ttl=127 time=6.422 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=7 ttl=127 time=7.019 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=8 ttl=127 time=6.708 ms (same route)
RR:     crete.imag.fr (129.88.104.1)
        lanai-104.imag.fr (129.88.104.146)
        tenerife-104.imag.fr (129.88.104.158)
        lanai-wl.imag.fr (129.88.104.62)
        crete.imag.fr (129.88.104.1)

64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=9 ttl=127 time=6.534 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=10 ttl=127 time=6.381 ms (same route)
64 bytes from tenerife-104.imag.fr (129.88.104.158): icmp_seq=11 ttl=127 time=7.198 ms (same route)

--- crete.imag.fr ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11510ms
```

FIG. 8.8.: Ping avec enregistrement de routes

[2]Nous avons vérifié le fonctionnement pour l'IPv6 en utilisant d'outre outils, car `ping6` n'a pas l'option `-R` pour enregistrer la route

Temps (ms)	Message	Observations
113	HANDOFF_REQUEST	Envoyé par crete Destinataire (IP) = lanai Destinataire (MAC) = kauai
122	HANDOFF_ACK	Reçu par crete Source (IP) = kauai Source (MAC) = kauai
139	Le handoff physique est achevée ; les daemons de mobilité sur crete et lanai reçoit des notifications et insère manuellement les entrées correspondantes dans leur tables ARP.	

TAB. 8.1.: Capture des messages échangées par l'hôte mobile lors du handoff

Pour vérifier plus en détail les performances de notre protocole de mobilité, nous avons simulé une situation dans laquelle l'hôte mobile participe à un trafic de données similaire à un flot multimédia. Nous avons utilisé un outil développé dans le langage C qui envoie des paquets UDP à une rate constante entre deux machines. Nous avons testé séparément deux cas : trafic de *crete* vers *tenerife* et vice-versa. Pendant le transfert de données, nous forçons le handoff de *crete* de son point d'accès courant *kauai* vers l'autre point d'accès, *lanai*. Le transfert physique est précédé de la demande de mise en place de la nouvelle route. Au moment où *crete* reçoit la confirmation de cette mise en place, elle effectue le handoff physique et change sa passerelle par défaut vers *lanai*.

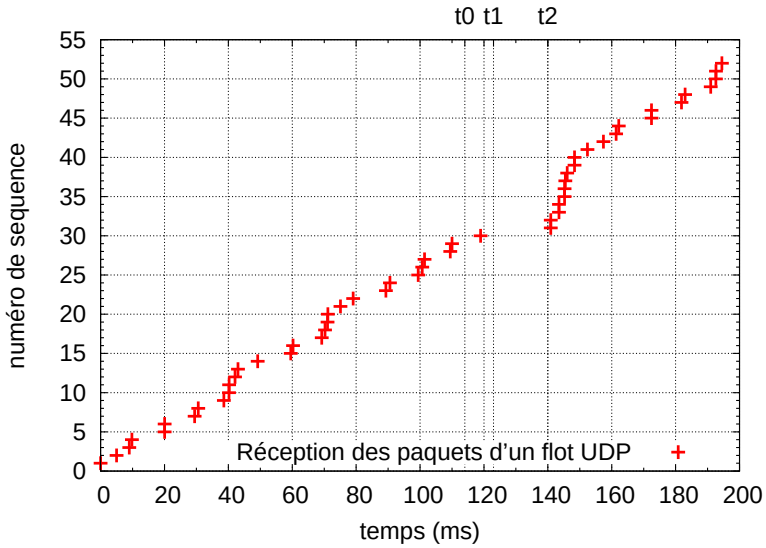


FIG. 8.9.: Capture du flot UDP lors du handoff

Nous présentons dans la figure 8.9 et le tableau 8.1 la séquences de messages capturés sur la machine *marie*, dans le cas du trafic descendant vers l'hôte mobile. Nous observons qu'il n'y a pas de paquets perdus. Il y a une interruption de 20 ms, entre le moment où le

changement de route intervient sur *tenerife* et le moment où *crete* achève le transfert vers le nouveau point d'accès *lanai*. La capture des messages échangés nous confirme que les datagrammes arrivés auparavant sur *lanai* sont mises dans un tampon et envoyées dès que la résolution ARP réussit, d'où l'intervalle plus petit entre les premiers paquets reçus après le handoff.

8.5. Conclusion

Nous avons commencé ce chapitre par la présentation des topologies possibles d'un domaine de micro-mobilité et nous avons défini les entités qui le composent. Ensuite, nous avons exposé les points importants dans la conception de notre protocole : compatibilité avec les diverses topologies possibles et routage optimisé pour les communications intra-domaine.

Notre protocole de micro-mobilité propage tout simplement les routes d'hôte dans tout le domaine, pour assurer un routage direct entre l'hôte mobile et tout le reste du domaine. De plus, les routes d'hôte coexistent avec le routage traditionnel basé sur les préfixes des sous-réseaux. Chaque fois qu'une hôte mobile change de sous-réseau, elle se configure des nouvelles adresses IP topologiquement correctes (appartenant au nouveau sous-réseau) qu'elle utilisera pour les connexions ouvertes ensuite. Les anciennes adresses et les routes d'hôte correspondantes ont la durée de vie de connexions ouvertes auparavant, et elles sont effacées au moment où ces connexions se termineront.

Nous avons testé le fonctionnement et la performance de notre solution par une approche expérimentale, en développant un prototype. Nous avons implémenté des daemons de mobilité qui s'exécutent sur les hôtes mobiles, les routeurs intermédiaires et les points d'accès. Les daemons de mobilité communiquent entre eux pour propages la mise à jour des routes d'hôte ; ils agissent sur les tables de routage pour insérer, modifier ou effacer ces entrées. Notre protocole est étroitement lié au handoff 802.11 et dès que la réassociation physique est achevée, les daemons de mobilité introduisent manuellement des entrées dans les tables ARP pour réduire encore plus le temps de handoff. Pour éviter la perte de paquets, nous avons utilisé une fonctionnalité standard du mécanisme ARP dans Linux pour garder les paquets au points d'accès cible pendant la durée du handoff. Les tests et mesures que nous avons fait ont montré de bonnes performances : aucune perte de paquets et un temps de handoff de l'ordre de dizaines de millisecondes.

9. Le niveau 3.5 : Identificateurs d'hôte

Nous présentons dans ce chapitre une solution pour la mobilité globale des hôtes, c'est-à-dire entre deux sous-réseaux qui peuvent être potentiellement distants, à travers l'Internet. Notre solution est basée sur l'introduction d'une nouvelle couche *d'identificateurs d'hôte* entre le niveau réseau et le niveau transport (d'où le numéro de couche *3.5*). Cette couche a comme but de rendre transparent au niveau transport le changement d'adresses IP pour permettre ainsi la continuation des connexions transport.

Cette architecture ne demande aucun changement ni dans les applications, ni dans les protocoles réseau ou transport existants. Étant basée sur le paradigme *end-to-end*, la couche 3.5 doit être implémentée sur les deux hôtes qui participent à une connexion ; en échange, on ne demande aucun autre changement dans des machines tierces. La compatibilité avec les machines qui n'implémentent pas la couche 3.5 es gardée dans le sens où des connexions transport basées sur la couche 3.5 peuvent co-exister avec des connexions transport régulières.

Après une présentation de l'architecture du niveau 3.5 et une discussion sur la forme et les caractéristiques nécessaires pour les identificateurs d'hôte, nous allons présenter en détail les mécanismes utilisés dans notre solution et son implémentation Linux.

9.1. Le niveau 3.5

Le concept d'identificateur d'hôte n'est pas nouveau ; il est discuté à l'intérieur de l'IETF depuis quelques années. Un des premières documents à ce sujet est « *Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture* » de N. Chiappa[107]. Dans sa proposition, Chiappa présente deux emplacements différentes pour le concept d'identificateurs d'hôte dans la pile de protocoles réseau :

- Couche 4.5, entre le niveau transport et les applications;
- Couche 3.5, entre le niveau réseau et le niveau transport.

Si on place les identificateurs d'une communication au niveau 4.5, ça correspond en effet au niveau *session* de l'OSI. L'inconvénient principal de l'introduction du niveau session dans les protocoles TCP/IP est qu'on doit dupliquer une bonne partie du niveau transport, en spécial les fonctionnalités nécessaires pour assurer la fiabilité et l'arrivée dans l'ordre pour les connexions TCP. Un autre désavantage est qu'une telle couche devrait interagir et donc

fournir de fonctions spécifiques à tous les protocoles de niveau transport (aujourd'hui TCP et UDP, mais demain SCTP, DCCP (*Datagram Congestion Control Protocol*) [108] et autres).

En échange, le positionnement des identificateurs d'hôte au niveau 3.5 a le mérite d'assurer l'indépendance des *tous* les protocoles transport face aux changement des adresses réseau. Ceci grâce au rôle du protocole IP, lien unique entre les différentes technologies réseau et les différents protocoles de transport (voir figure 9.1).

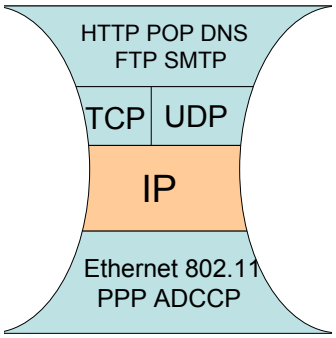


FIG. 9.1.: Choix de l'emplacement pour la couche d'identificateurs d'hôte

L'architecture que nous proposons introduit donc une couche supplémentaire, qui masque les adresses IP des interfaces réseau aux connexions transport de données. Cette couche se présente sous la forme de deux composants :

- Une *sous-couche de virtualisation*, située au-dessous de la couche transport, qui présente aux protocoles de niveau transport les identificateurs d'hôte dans le format attendu d'une adresse IP - 32 et respectivement 128 de bits pour les deux version IPv4 et IPv6 ;
- Une *sous-couche de translation*, située en dessus de la couche réseau, qui traduit les identificateurs d'hôte dans les adresses IP réelles présentes dans les datagrammes circulant sur le réseau.

Nous expliquons en détail les mécanismes de notre architecture dans la section 9.3, après avoir examiné les choix possibles pour les identificateurs d'hôte.

9.2. Choix des identificateurs d'hôte

Un point très important dans la construction de la couche 3.5 est le choix des identificateurs d'hôte. Ce choix est déterminant pour les autres parties de l'architecture, comme par exemple le protocole d'échange des identificateurs entre les deux hôtes, la présence des

identificateurs dans les datagrammes circulant sur le réseau, etc.

Nous avons identifié plusieurs propriétés qui caractérisent les identificateurs d'hôte :

- *Structure interne* : les identificateurs peuvent être opaques ou structurés ;
- *Taille* : identificateurs de longueur fixe ou variable ;
- *Unicité* : un hôte peut avoir un identificateur universellement unique ou l'unicité peut être restreinte que pour le couple des machines participantes à une connexion ;
- *Allocation* : chaque machine peut choisir aléatoirement son identificateur ou il lui est attribué par une autorité ;
- *Durée de vie* : des identificateurs stables même après une redémarrage de la machine ou des identificateurs éphémères qui changent pour chaque connexion.

Ces caractéristiques ne sont pas totalement indépendantes : par exemple, une structure interne stricte peut déterminer une taille fixe des identificateurs, permettre l'allocation hiérarchique et stable des identificateurs et leur assurer une unicité globale.

Plusieurs espaces de noms sont candidates pour fournir des identificateurs d'hôte. Les solutions que nous avons présentées dans la section 4.4 lors de l'état de l'art (LIN6, VIP, VNAT, HIP) utilisent comme identificateurs d'hôte des adresses IP, des noms d'hôte FQDN ou des clés cryptographiques asymétriques.

Le choix pour un type ou un autre doit se faire en fonction de propriétés offertes par l'espace de noms respectif par rapport aux caractéristiques nécessaires d'un identificateur (unicité, stabilité) et par rapport à l'architecture de solution (allocation des identificateurs, translation entre les noms d'hôte et les identificateurs, translation entre des identificateurs et les adresses IP, échange des identificateurs avec l'hôte pair, etc.).

HNAT représente un cas particulier, car il fonctionne en dessus de IPSec, ce qui motive en grande partie le choix de clés asymétriques comme identificateurs d'hôte. Les trois autres propositions mentionnées utilisent des adresses IP, ce choix étant déterminé par la transparence souhaitée vis-à-vis des protocoles transport.

Dans LIN6 l'identificateur d'hôte comporte 40 bits et il est attribué d'une façon unique et stable à chaque hôte par une autorité. Tirant profit de la structure et de la taille de 128 bits des adresses IPv6, l'identificateur est inséré dans les datagrammes circulant sur le réseau. Dans VIP, chaque hôte choisi son identificateur d'une façon aléatoire, dans l'espace réservé de la classe E des adresses IP. Pour apprendre l'identificateur de l'hôte pair, des requêtes DNS spéciales sont lancées, en parallèle avec la requête DNS initiale. En plus, pour assurer l'unicité des identificateurs, une phase de négociation peut avoir lieu en cas de conflit local entre les différents identificateurs utilises dans les connexions d'une même machine. La troisième solution, VNAT, utilise carrément les adresses IP initiales comme identificateurs et ne propose aucun mécanisme de négociation au cas d'un conflit local d'adresses (ce qui peut intervenir par exemple quand plusieurs hôtes correspondants d'une même machine se trouvent derrière le même routeur NAT).

9.2.1. Unicité locale et globale des identificateurs d'hôte

Le point clé de notre solution, qui la différencie des autres propositions, est notre choix sur l'unicité et la durée de vie des identificateurs. Nous sommes partis de l'observation suivante : les identificateurs des deux hôtes participants a une connexion sont tenus a être uniques que localement, sur chacune de deux machines. Ceci signifie que les deux pairs peuvent très bien utiliser chacun des identificateurs suivants, et n'ont pas besoin de s'échanger et de négocier la même paire d'identificateurs. En plus, pour minimiser la taille de l'espace de noms, les identificateurs utilisés pour designer les hôtes correspondants sont éphémères, leur durée de vie étant égale a celle des connexions ouvertes avec l'hôte respective. Par contre, l'identificateur local de chaque machine a une durée de vie plus longue, et on pourrait utiliser le même identificateur après un redémarrage de la machine.

Pour les mêmes raisons que les autres solutions, nous utilisons des identificateurs d'hôte de la forme d'une adresse IP. Pour que ces identificateurs n'interfèrent pas avec les adresses IP réelles d'interfaces, notre solution a été d'utiliser l'espace réservé d'adresses de la classe E. Même si nous n'avons pas implémenté la solution correspondante pour le cas d'IPv6, il est clair que nous pouvons utiliser un espace d'adresses correspondant à un préfix IPv6 réservé[109].

Pour une connexion donné, les identificateurs d'hôte restent stables, en dépit de la mobilité des hôtes d'un sous-réseau à un autre (c'est-à-dire, en dépit des changements intervenus dans l'adresse IP de l'interface réseau). En plus, les identificateurs d'hôte sont uniques pour une machine donnée, même dans le cas où la machine possède plusieurs interfaces réseau, qu'elle utilise simultanément ou non.

9.3. Mécanismes utilisés sur les hôtes

Dans cette section, nous allons présenter l'architecture de notre solution, sans toutefois entrer dans les particularités de l'implémentation Linux. Comme nous l'avons précisé, notre solution est composée de deux parties principales : une couche de virtualisation des adresses IP et une couche de translation des adresses IP.

9.3.1. La couche virtualisation

Notre but a été de rendre l'insertion de la sous-couche 3.5 transparente, tant aux applications qu'aux protocoles réseau et transport. La réalité de l'implémentation des protocoles dans les noyaux des systèmes d'exploitation actuels fait que la couche transport est étroitement liée à la couche réseau. Pour cette raison, nous avons du utiliser dans notre architecture deux autres mécanismes, qui se situent cette fois à la frontière entre le niveau transport et les applications : interception des requêtes DNS et interception des appels de fonctions de la bibliothèque système *socket*.

La couche de virtualisation doit fournir des adresses IP virtuelles qui sont utilisés comme

identificateurs stables pour les deux cotés de la connexion : local et distant. Le cas est différent pour l'initiateur actif de la connexion (le client) et pour l'hôte serveur.

Interception des requêtes DNS

Dans la plupart des cas, les applications client commencent par la résolution d'un nom d'hôte dans l'adresse IP correspondante avant d'ouvrir une connexion à un serveur. Nous profitons de l'existence de ces requêtes DNS pour les intercepter et retourner aux applications un identificateur d'hôte, sous la forme d'une adresse IP choisie aléatoirement dans la classe E, au lieu de l'adresse IP réelle reçue dans la réponse du serveur DNS. Le couple *(identificateur d'hôte - adresse IP réelle)* est envoyé à la couche de translation qui devra effectuer la translation inverse avant d'envoyer les datagrammes sur le réseau.

Interception des appels de fonctions de la bibliothèque socket

Toutefois, il est possible que les applications ouvrent une connexion à un hôte distant sans passer par une requête DNS, en utilisant directement l'adresse IP de l'hôte distant. Dans ce cas, l'unique possibilité de remplacer cette adresse IP par un identificateur stable est d'intercepter les appels d'ouverture de connexion (de la bibliothèque système *socket*). Ainsi, dans le cas d'une application écrite en C, l'appel de la fonction `connect` doit être intercepté et l'adresse IP réelle sera remplacée par un identificateur d'hôte. Comme dans le cas de l'interception de requêtes DNS, ce changement d'adresse est notifié à la couche translation.

Cas de l'hôte serveur

Dans le cas de l'hôte serveur, la situation est complètement différente, car l'application serveur ne connaît pas à priori ni le nom d'hôte DNS, ni l'adresse IP des clients distants. Les serveurs TCP ouvrent un socket d'écoute, attendent des demandes de connexion de la part des clients, et créent une nouvelle socket pour chaque connexion client. Dans le cas de connexions UDP, qui sera présenté dans la section 9.3.3, une seule et même socket sera utilisée pour tous les datagrammes entrants envoyés par des clients différents.

Dans les deux cas, la virtualisation des adresses IP de clients est faite automatiquement par le travail de la couche translation, qui remplace l'adresse IP source présente dans les datagrammes entrants avec un identificateur d'hôte.

Virtualisation de l'adresse IP locale

Pour la virtualisation de l'adresse IP locale utilisée dans les connexions transport, nous utilisons la même couche d'interception de fonctions de la bibliothèque système *socket*. Cette fois-ci la fonction qui nous intéresse est `bind`, par laquelle on spécifie l'adresse IP d'une des

interfaces réseau locales qui sera utilisé pour la connexion. Les appel de cette fonction sont interceptés et on remplace le paramètre réel avec l'identificateur local d'hôte. Il faut noter que cette adresse IP virtuelle qui identifie la machine locale est commune pour la totalité des connexions ouvertes sur la machine.

Une différence par rapport à l'interception de l'appel de `connect` est que dans la plupart de cas, l'appel de la fonction `bind` se fait avec le paramètre `INADDR_ANY` comme adresse IP locale. En plus, si pour les serveurs l'appel de la fonction `bind` est explicit, dans le cas des clients il est rare que les applications contiennent un appel vers cette fonction. À la place, au moment de l'envoi de données à une *socket* dont l'adresse source n'a pas été spécifie, la fonction `bind` est appelée automatiquement. En conclusion, pour être sûr que l'adresse IP locale est virtualisée dans ces situations, nous interceptons les appels de toutes les fonctions de connexion et d'envoi de données et faisons auparavant un appel explicit de la fonction `bind` en passant comme paramètre l'identificateur local d'hôte. Nous donnerons plus de détails au moment où nous présenterons l'implémentation de la couche virtualisation (section 9.5).

9.3.2. La couche translation

La couche translation est basée sur un moteur NAT (*Network Address Translation*) standard, la seule différence avec l'usage normal de NAT étant que les translations se font localement, sur chacune de deux machines participantes à une connexion, et aucun autre élément intermédiaire n'est pas nécessaire. Par rapport à la couche virtualisation, la couche translation a plutôt un rôle passif ; elle reçoit de règles de translation de la part de la couche de virtualisation et les met en pratique, traduisant les identificateurs d'hôte vers des adresses IP réelles et vice-versa. Ces règles peuvent être ensuite mises à jour suite à un changement de l'adresse IP d'une des hôtes participantes à la connexion, et sont effacées suite à la fermeture de la connexion respective.

Dans une partie de son opération, le rôle couche de translation n'est pas tout à fait passif : le moment où la machine serveur reçoit le premier datagramme d'une nouvelle connexion. La couche translation doit être capable d'identifier ces nouvelles connexions, choisir un identificateur d'hôte pour la machine distante et mettre en place une règle de translation qui actionnera sur les datagrammes subséquents faisant partie de la respective connexion.

9.3.3. Le cas de connexions UDP

Dans l'analyse que nous avons fait jusqu'ici, nous avons discuté le cas particulier des protocoles de transport orientes connexion, plus exactement le TCP, utilisé dans la plupart des données véhiculées dans l'Internet[33]. Toutefois, le cas des protocoles *sans connexion*, en instance de l'UDP, doit être pris en compte, car on dénote un usage croissant dans les applications multimédia sur Internet.

Même s'il n'y a pas un vrai concept de connexion dans UDP, les applications maintiennent elles-mêmes une notion de connexion. Cela signifie que les applications qui utilisent UDP

sont préparées pour la perte de datagrammes ou pour leur arrivée en doublons ou en désordre. Dans ces situations, les applications déclenchent des retransmissions, décident l'échec de la communication ou rouvrent une nouvelle connexion. En échange, il y a peu des applications qui prennent en compte la possibilité que hôte locale ou l'hôte distante aient pu se déplacer et changer d'adresse IP. Dans ce cas, les applications pourraient profiter du support pour la mobilité offert par notre couche d'identificateurs d'hôte.

Il faut aussi mentionner une particularité de sockets UDP, qui peuvent fonctionner de deux façons différentes. Habituellement elles sont utilisées en mode *déconnecté*, dans lequel une socket peut être utilise pour changer des datagrammes avec n'importe quel hôte distant. Il y a aussi un mode *connecté*, dans lequel on utilise la fonction connect pour spécifier que les données envoyées par la socket seront toujours destinées au même hôte distant. Même dans le mode « connecté », il y toujours une différence par rapport au TCP, dans le sens que pour UDP il n'y a pas de messages d'établissement ou de la fermeture de la connexion. Cela doit être pris en compte par la couche virtualisation et translation dans la détection d'une nouvelle « connexion ».

9.4. Transfert des connexions réseau

Dans les sections précédentes, nous n'avons pas mis l'accent sur la procédure de localisation des hôtes qui se trouvent au bout des connexions réseau. Il est même possible qu'il n'y ait pas de phase de localisation avant l'ouverture d'une connexion. En effet, une des raisons pour lesquelles nous avons utilisé l'interception de appels de la fonction connect a été justement la possibilité que les clients ouvrent des connexions sans passer par un nom d'hôte DNS.

Par contre, pour le transfert de la connexion, il faut que les deux hôtes possèdent des identificateurs uniques globalement et qu'il y ait une infrastructure tierce où les associations entre ces identificateurs et les adresses IP courantes soient régulièrement mises à jour. Même si notre architecture est censée à être *end-to-end*, un élément tierce est nécessaire au moins dans le cas où les deux hôtes au bout d'une connexion changent simultanément d'adresse IP, et aucun de deux hôtes n'est plus en mesure de retrouver son pair à l'adresse IP précédente.

Pour ce but, nous avons choisi l'infrastructure du système DNS et les serveurs DNS dynamiques qui peuvent être mises à jour d'une manière sécurisé. Ainsi, tout suite après l'ouverture d'une connexion, chaque hôte doit apprendre le nom de la machine distante qui lui envoie des paquets. Ensuite, suite a un déplacement, le l'hôte en cause doit notifier son correspondant du changement d'adresse.

Nous avons eu le choix entre deux types de sécurisation de ces mises a jour : en créant une association de sécurité entre les deux machines, ou en se basant sur une infrastructure externe. Les mécanismes de création d'une clé secrète commune, comme Diffie-Helmann, impliquent un échange consistant de paquets dans ce sens. La deuxième possibilité, que nous avons choisie pour notre solution, se base sur la confiance deux pairs dans une infrastructure externe. Nous croyons que la solution la plus appropriée est le DNS dynamique et

sécurisé, qui commence à être déployé a large échelle dans les serveurs DNS ainsi que dans les dernières versions des systèmes d'exploitations des clients.

9.4.1. Échange réciproque des identifiants

Dès qu'une connexion est ouverte, chacun des deux points finaux doivent apprendre le nom d'hôte, stable et unique, de son pair. La particularité de notre solution étant que les identificateurs d'hôte sont choisis aléatoirement par l'hôte local, il n'y a pas de correspondance entre les identificateurs d'hôte choisis d'un coté et de l'autre de la connexion. Les seules informations communes que les deux hôtes partagent déjà sont leurs adresses IP respectives. Notre idée est de trouver des noms d'hôte stables et uniques globalement, qui seraient éventuellement basés sur ces adresses IP de début. Plusieurs solutions existent :

- L'infrastructure DNS permet de faire des requêtes *inverses* (plus exactement, des requêtes de type PTR) qui retournent le nom d'hôte correspondant à une adresse IP. Une solution serait donc que des noms d'hôte stables associés à chaque hôte soient présentes dans les enregistrements PTR correspondantes à leurs adresses IP actuelles. Par contre, le fait que les enregistrements PTR soient gérés au niveau de sous-réseaux IP n'offre pas la certitude d'avoir le contrôle sur les serveurs DNS responsables de chaque réseau, ce qui est nécessaire pour la mise à jour régulière de ces enregistrements.

- Dans le système DNS il est plus facile d'avoir le contrôle d'un certain domaine de noms d'hôte que d'un domaine des adresses IP. Une solution pourrait être de former un nom a partir de l'adresse IP respective, par exemple `129-88-38-99.dyndns.org`. Un problème présent dans cette solution est la réallocation des adresses IP dynamiques, qui fait que deux hôtes distincts qui utilisent successivement la même adresse IP peuvent se retrouver associés à un même nom. Ce problème apparaît aussi pour les connexions ouvertes par des clients qui se trouvent derrière un même routeur NAT. Dans ce cas, le serveur reçoit des paquets qui viennent de la même adresse IP, et ne pourra pas l'utiliser pour former des noms d'hôte distincts.

- Une solution pour contourner ces collision des noms d'hôte est de proposer explicitement au début de la connexion un nom d'hôte qu'on peut contrôler. Même si cela implique quelques datagrammes en plus au début de la connexion, nous avons choisi d'utiliser cette solution pour sa simplicité. Nous considérons que les quatre messages qu'il faudra échanger (notifications de nom + acquittements) n'influencent pas significativement la performance de la connexion.

9.4.2. Notification des correspondants

Chaque hôte participant à une connexion virtualisée doit surveiller l'état et les changements d'adresses de ses interfaces réseau, ainsi que l'état de sockets réseau. Ainsi, notre

couche peut déterminer quelles sont les connexions actives et à quel moment ces connexions doivent être migrées.

Ainsi, si son adresse IP change suite à un déplacement ou autre événement, l'hôte en cause doit effectuer trois actions :

- Il met à jour les règles de translation du moteur NAT local et remplace l'ancienne adresse IP avec la nouvelle ;
- Il ouvre une connexion sécurisée à son serveur DNS et fait correspondre à son nom d'hôte l'adresse IP courante ;
- Il notifie son changement d'adresse à tous les hôtes correspondants avec lequels il a des connexions ouvertes. Ensuite, les hôtes correspondants qui ont besoin de s'assurer de la l'authenticité de cette notification peuvent le faire en faisant une simple requête DNS sur le nom de l'hôte mobile.

Déplacement simultané

Une situation particulière est le déplacement simultané des deux hôtes, car dans ce cas le message de notification ne peut pas être envoyé a l'ancienne adresse de l'hôte correspondant. Pour résoudre ceci, nous avons prévu un acquittement pour le message de notification. Dans le cas ou cet acquittement n'est pas reçu, un déplacement simultané est présumé et une requête DNS doit être effectuée de deux cotés avant d'envoyer la notification.

9.5. Implémentation

L'architecture de notre implémentation est présentée dans la figure 9.2. Une partie des composants s'exécutent dans l'espace utilisateur, notamment le composant central de l'architecture, le daemon IPMapping. D'autres composants sont implémentés sous la forme de modules noyau, notamment la translation d'adresses.

Le daemon IP Mapping

Le daemon IPMapping représente le cœur de l'architecture. Sa responsabilité est de garder une vue toujours actualisée des correspondances *identificateurs virtuels* $\leftrightarrow$ *adresses IP réelles*. Dans ce but, il communique avec tous les autres composants de l'architecture :

- Il alloue des plages d'identificateurs virtuels uniques pour les composants de la couche virtualisation et choisit l'identificateur local d'hôte, tout en gardant l'unicité locale de ces identificateurs. Il transmet à la couche virtualisation les identificateurs disponibles et reçoit des notifications à chaque virtualisation d'une nouvelle connexion ;

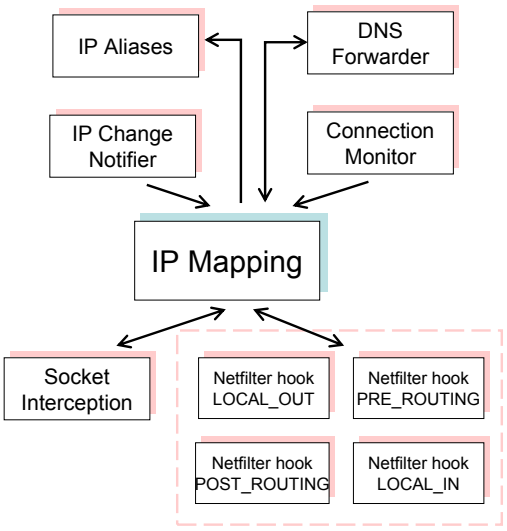


FIG. 9.2.: Architecture de l'implémentation Linux

- Reçoit des informations sur l'état des interfaces réseau locales et sur l'état des connexions : fermeture des connexions, changement de l'adresse IP locale, notification de migration de la part de l'hôte distant ;
- Demande à la couche translation la mise en place, la mise à jour ou l'effacement des règles de translation nécessaires ;
- Le daemon IPMapping s'occupe aussi du protocole de migration avec l'hôte correspondant.

Le daemon DNS Forwarder

Ce daemon est responsable de l'interception des requêtes de noms DNS faites par les applications locales. Nous avons tiré profit de la facilite de paramétrage du serveur DNS d'une machine : ainsi, l'activation du daemon se fait par la spécification de l'hôte locale (adresse IP `127.0.0.1`) comme serveur DNS. Dans les systèmes Unix, ceci est fait par le changement d'une ligne dans le fichier `/etc/resolv.conf`.

Le daemon consiste dans un application de type proxy DNS qui écoute sur le port standard DNS (53) et reçoit ainsi les requêtes DNS faites par les applications. Ensuite, il fait suivre la requête au vrai serveur DNS, attend et reçoit la réponse, et vérifie si l'hôte distante a déjà associé un identificateur virtuel. Si ce n'est pas le cas, il choisit un identificateur virtuel dans la plage fournie par le daemon IPMapping et l'informe en retour du couple *adresse IP virtuelle – adresse IP réelle* qui vient d'être crée. Pour terminer, il retourne à l'application l'identificateur virtuel à la place de l'adresse IP.

Le daemon IP Change Notifier

Ce composant est responsable pour la surveillance des changements dans l'état des interfaces réseau. Il consulte régulièrement la structure de données `struct ifreq`, définie dans `<linux/if.h>` pour apprendre des éventuelles changements dans les adresses IP des interfaces réseau. Toutes les changements sont notifiés immédiatement au daemon IPMapping qui modifiera en conséquence les règles de translation passées au moteur NAT local.

Le daemon IP Aliases

Comme son nom l'indique, ce composant est responsable de la mise en place et de la mise à jour des alias pour les interfaces réseau. Dans la construction de notre prototype Linux, il nous a été impossible d'exécuter un appel de la fonction `bind` avec une adresse IP qui n'appartient pas à une interface réseau locale. Le paramètre prévu a cet effet dans le noyau Linux, `ip_nonlocal_bind`, n'a plus d'effet dans les noyaux 2.6.

Pour dépasser cet obstacle, nous avons utilisé des alias IP, qui sont des adresses IP supplémentaires qu'on assigne aux interfaces permettant d'avoir, comme en IPv6, plusieurs adresses par interface. Le rôle de ce daemon est donc d'assigner l'identificateur virtuel choisi pour la machine locale comme un alias l'interface de sortie actuelle.

Le module d'interception des appels socket

Ce composant intercepte les appels de fonctions de la bibliothèque système *socket* et réalise la virtualisation des adresses IP en les remplaçant avec des identificateurs virtuels.

Son fonctionnement est base sur le fait que dans un système Linux, toutes les appels système passent par les entrées du fichier `/i386/kernel/entry.S` qui contient des pointeurs vers les fonctions system. Nous avons développé un module noyau qui remplace le pointeur respectif avec l'adresse de son propre code. Notre module est conçu pour intercepter plusieurs appels systèmes, notamment `bind`, `connect` et `close`, mais aussi `sendto` et `recvfrom` pour les datagrammes UDP.

Pour chaque appel de ces fonctions, on remplace le paramètre de l'appel avec une adresse IP virtuelle. Les adresses virtuelles sont gérées par le daemon IP Mapping, qui assure ainsi l'unicité locale des adresses virtuelles en tant qu'identificateurs d'hôte. Quelques situations particulières existent, surtout pour la virtualisation de l'adresse IP locale :

- Dans l'interception de l'appel de la fonction `connect`, on vérifie si un `bind` a été effectue sur la socket. Si ce n'est pas le cas, on appelle le `bind` avec une adresse IP virtuelle avant d'appeler la fonction système `connect`.
- Si l'appel de `bind` a été fait sans préciser une adresse IP locale mais `INADDR_ANY` à la place, on n'a pas de correspondance exacet *adresse IP locale – identificateur d'hôte*

local à passer au daemon IPM̃apping. Dans ce cas, la couche translation doit remplacer l'adresse IP virtuelle avec l'adresse IP réelle qui correspond à l'interface de sortie du datagramme.

Le daemon Connection Monitor

La durée de vie des identificateurs virtuels qui remplacent les adresses IP réelles des hôtes correspondants doit être égale à la durée de vie des connexions. Si le début des connexions TCP correspond toujours à l'appel de la fonction `connect`, la fin est un plus compliqué à determiner, car la fonction `close` n'est pas toujours appelé. Pour cette raison, un daemon appelé Connection Monitor est responsable pour la surveillance des états des sockets ouvertes, pour voir quand elles passent de l'état actif aux états de fin FIN_WAIT, TIME_WAIT ou CLOSE_WAIT.

Le modules de translation netfilter

L'implémentation de la couche translation est basée sur le moteur netfilter de Linux. Celui-ci offre une façon complexe et puissante de programmer les règles de translation : écrire des modules noyau et les insérer dans cinq *canevas* différentes à plusieurs stages sur le parcours des datagrammes dans le noyau Linux. Prenons l'exemple d'un paquet envoyé par l'hôte client à un serveur. La couche virtualisation s'est assurée qu'au début ce paquet a été construit en utilisant des adresses IP virtuelles. En arrivant à la couche IP, les actions suivantes sont exécutées sur le paquet en cause :

- Une translation de l'adresse destination est faite dans le canevas LOCAL_OUT ;
- Une fois que le processus de routage est passé, une translation de l'adresse source est effectué dans le canevas POST_ROUTING avant de envoyer le datagramme sur l'interface de sortie choisie;
- Le paquet qui a maintenant des adresses IP réelles dans l'en-tête IP circule sur le réseau Internet et arrive finalement à l'hôte destination ;
- Deux autres translation seront ensuite effectuées, pour que l'en-tête IP contienne des adresses IP virtuelles : une translation de l'adresse destination dans le canevas PRE_ROUTING et une translation de l'adresse source dans le canevas LOCAL_IN.

Pour chaque canevas de netfilter nous avons implémenté un module noyau séparé. Les actions exécutées par les deux premières sont évidentes : translation d'adresses réseau pour les paquets sortants en fonction des règles transmises par le daemon IPMapping.

En échange, l'opération des deux canevas pour les paquets entrants est plus complexe. Ces deux canevas sont responsables, en plus de la translation de paquets suites a des règles imposées, de la détection les nouvelles connexions. Elles doivent identifier les segments SYN

de début des connexions TCP et les premiers paquet UDP qui font partie d'une nouvelle « connexion ». La détection de nouvelles connexions est faite dans le premier canevas, PRE_ROUTING, qui informe le daemon central IPMapping. Celui-ci transmet l'information au canevas suivant, LOCAL_IN, qui réalisera la translation effective.

Communication entre les composants

Pour les communications entre les différents composants de notre architecture, nous avons fait usage de différents mécanismes standard présents dans le système Linux. Ainsi, les communications entre les daemons de l'espace utilisateur se font par des *named pipes*, ce qui est le cas aussi pour la communication être l'élément central de l'architecture et le module noyau de l'interception des appels *socket*. Par contre, nous n'avons pu utiliser le même moyen pour la communication avec les canevas de netfilter. À la place, nous avons utilise des sockets netlink, une nouvelle interface utilise en Linux pour la communication entre l'espace utilisateur et les modules noyau.

9.5.1. Tests et résultats

Nous avons réalisé une série des tests et expériences pour prouver le fonctionnement de notre solution et pour mesurer ses performances.

Le but de la première série de tests a été de prouver son fonctionnement, dans le cas d'une simple application client – serveur qui fonctionne en mode TCP ou UDP. La configuration de test est formés de deux machines portables, équipées des processeurs Intel Pentium III à 800 Mhz et de 128 MB de mémoire RAM, avec Linux Fedora-Core 3 (noyau 2.6.10).

Les informations affichées par l'application de test utilisé sur les deux machines confirment que les sockets réseau sont associées à des adresses IP virtuelles, fait confirmé aussi par le résultat de l'outil netstat. De l'autre coté, l'observation de paquets circulant sur le réseau, capturés à l'aide de l'outil tcpdump sur une autre machine connectée sur le même lien physique, nous montre que les adresses IP virtuelles ont été bien modifiées dans des adresses IP réelles (voir la figure 9.3).

Le fonctionnement correct ayant été prouvé, le deuxième groupe de tests a visé la mesure de performances d'une connexion entre deux machines qui implémentent la couche 3.5. Nous avons fait des mesures de débit et de latence en utilisant l'outil netperf pour mesurer le débit maximum et notre application client - serveur précédente pour mesurer la latence. Les résultats obtenus sont montrés dans la figure 9.4. On observe que la différence de débit est peu importante, étant due en grande partie aux translations effectués par le moteur netfilter. La latence introduite est aussi peu significative : l'interception des appels de la fonction socket et des requêtes DNS retarde le début de la connexion de seulement quelques dizaines de millisecondes.

La troisième série d'expériences a visé le comportement de la couche 3.5 pour masquer le changement d'adresses. Nous avons reconnecté l'interface réseau d'une de deux machines

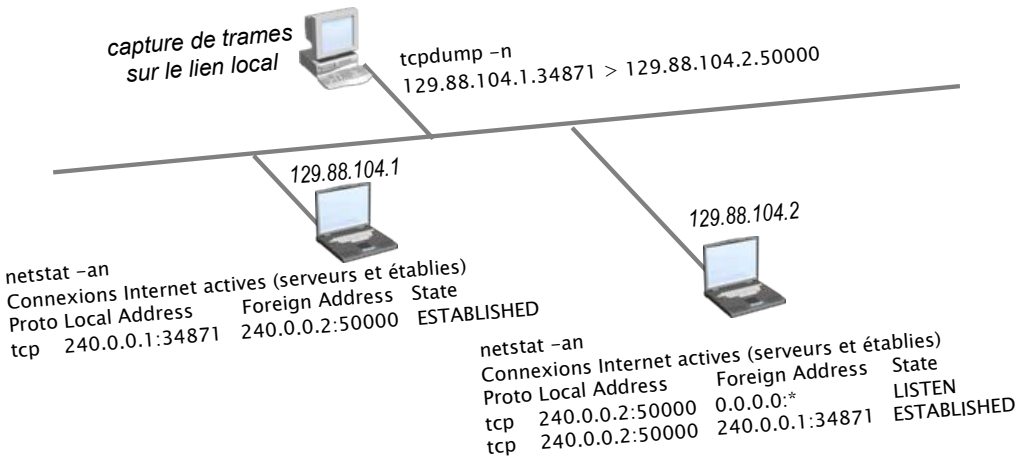


FIG. 9.3.: Fonctionnement de la couche 3.5

FIG. 9.4.: Surcharge introduite par la couche 3.5

à un autre sous-réseau, ce qui a résulté dans l'acquisition d'une adresse IP différente. La rapidité de la reprise de connexion dépend evidemment fortement de la distance entre les serveurs DNS et les deux machines. Dans notre cas toutes les machines étaient sur un même sous-réseau (RTT = 0.200 ms) , et nous avons pu observer que la durée de la reconnexion est inferieure à 10 millisécondes.

9.6. Conclusion

Les mécanismes présentés dans les deux chapitres précédents ont été conçus exclusivement pour la mobilité dans un domaine local restreint. Pour les compléter, nous avons présenté dans ce chapitre une solution pour la mobilité globale. Notre solution est construite sur le principe *end-to-end*, ce qui assure un acheminement optimal des datagrammes entre les deux machines participantes à une connexion. Le seul élément additionnel est représente par un ou plusieurs serveurs DNS, et son rôle se limite à l'authentification des mises à jour des adresses IP, suite à un changement de celles-ci.

Nous avons commencé le chapitre par une discussion sur le choix des identificateurs d'hôte et de leurs position dans la pile TCP/IP. Nous avons précisé la particularité de notre solution, dont la simplicité du fonctionnement vient du concept d'unicité locale des identificateurs d'hôte. Nous avons ensuite présenté l'opération des deux sous-couches virtualisation et translation, suivi des détails et particularités de l'implémentation Linux.

Nous avons testé le fonctionnement et la performance de notre solution par une approche expérimentale, en utilisant une simple application client – serveur qui fonctionne sous TCP et UDP et des outils réseau comme netstat, netperf et tcpdump. Les résultats obtenus confirment l'efficacité de notre solution.

10. Conclusions et perspectives

10.1. Bilan du travail réalisé

Le travail présenté dans cette thèse s'inscrit dans le vaste domaine de la gestion de la mobilité dans les réseaux. Pour mieux situer notre travail, nous avons présenté au début de ce document un panorama du domaine, en énumérant les différents types de mobilité, les entités impliquées et la problématique spécifique à chaque type de mobilité.

L'étendue du domaine nous a emmené à faire des choix précis sur notre thème particulier d'étude. Bien qu'il y ait un lien entre les types de mobilité, nous considérons qu'il n'est pas possible d'attaquer le problème de la mobilité dans son ensemble. Nous avons choisi de nous focaliser sur la mobilité d'hôtes et nous avons mis en évidence explicitement le concept de *session réseau*, le point commun qui fait la passerelle entre les différents types de mobilité.

Motivés par le rôle fédérateur du protocole IP, nous nous sommes appuyés plus particulièrement sur des solutions qui traitent la mobilité d'hôtes au niveau IP. En effet, nous considérons que les propositions qui opèrent au niveau TCP ou application ne représentent qu'une solution ponctuelle pour un problème particulier et ne gèrent pas toute la problématique de la mobilité. Nous avons considéré la réintroduction d'un niveau session très intéressante : elle permet le regroupage de plusieurs connexions réseau dans une seule session et pouvoir gérer d'une manière similaire la mobilité et les déconnexions prolongées. Malheureusement, *l'opacité* des sockets réseau dans les systèmes d'exploitation fait que l'implémentation d'une couche session soit difficile et en même temps faible en performances.

Lors de notre travail de recherche sur la mobilité locale au niveau IP, nous avons utilisé comme plate-forme expérimentale un réseau sans-fil de type 802.11. Nous avons observé que le temps pris par le handoff entre deux points d'accès est très important. Cela nous a emmené à consacrer une partie de la thèse pour l'analyse et la proposition des améliorations du handoff physique dans les réseaux 802.11.

Dans l'ensemble de travaux portés dans le cadre de cette thèse, nous pensons avoir apporté des réponses originales aux problèmes suivants :

Analyse des protocoles TCP/IP vis-à-vis de la mobilité d'hôtes. Dans la première partie de la thèse, nous avons analysé le fonctionnement des protocoles réseau par rapport à la mobilité. Un point important a été l'identification des concepts essentiels dans les réseaux et la présentation de leur emploi dans les protocoles de la pile TCP/IP. Notre conclusion a été qu'il faut faire une séparation nette entre les concepts d'identificateur et d'adresse.

Comparatif de l'état de l'art. Dans notre présentation des solutions existantes, nous avons suivi la structure de la pile de protocoles Internet, en situant chaque solution au niveau où elle opère. Nous avons identifié ensuite les différents choix présents dans l'architecture de chaque solution ; nous avons synthétisé ces choix dans un tableau comparatif de propositions examinées.

Analyse et mesures du handoff 802.11. Nous avons montré que le standard 802.11 manque de spécifications très précises pour la phase de handoff ; de ce fait, les diverses implémentations ont des comportements différents qui se reflètent dans des délais de handoff variables. Une contribution importante a été l'étude du handoff en situation de charge : nous avons montré qu'une partie importante de associations et de scans ne réussissent pas si le point d'accès ciblé est soumis à un trafic important de données. Nous avons présenté quelques pistes possibles dans la conception d'une solution à ce problème, mois nous ne sommes pas arrivés à la définition complète de cette solution.

Amélioration du temps de handoff 802.11. Après avoir analysé les phases du handoff 802.11, nous avons identifié les deux composants qui consomment le plus de temps : détection et scan. Nous considérons que dans certaines conditions, un découplage de phases de scan et réassociation peut être possible ; dans cette situation, notre extension de réassociation directe réduit le délai à seulement quelques millisecondes.

Distribution de la charge dans un réseau sans-fil. Comme application de la réassociation directe, nous avons conçu un protocole pour distribuer la charge dans un réseau sans-fil. Nous sommes partis des observations qui montrent que dans certains cas plusieurs points d'accès sont déployés au même endroit pour décongestionner un point d'accès régulièrement chargé. Cependant, la distribution de charge n'est pas spécifiée dans le standard mais seulement dans quelques implémentations propriétaires. Dans notre proposition, les stations et les points d'accès compatibles coopèrent pour s'informer sur leur voisinage et le niveau de charge dans les cellules voisines. Les stations peuvent alors faire usage de l'extension de réassociation directe pour se transférer vers une cellule moins chargée, avec une pénalisation minimale en terme d'interruption de la connexion.

Néanmoins, notre approche et simpliste, car chaque station est libre de prendre toute seule la décision et l'instant de transfert ; nous pensons que plusieurs facteurs doivent entrer dans la prise de décision de transfert et que la passage à l'échelle pourrait représenter également un problème pour l'efficacité de notre protocole.

Protocole de micro-mobilité. Nous pensons qu'une solution unique comme Mobile IP ne peut pas être efficace pour un type particulier de mobilité comme la micro-mobilité. En même temps, les schémas de handoff des deux protocoles HAWAII et CIP sont soit non-optimisés soit difficilement implantables. Dans un domaine local, une grande partie de communications se font avec des hôtes correspondants situés dans le même domaine. Notre approche a été donc de privilégier un routage optimal à l'intérieur du domaine.

Notre démarche a volontairement été applicative, et cela a permis de concrétiser le travail sous la forme d'un prototype. Ce prototype tire parti de nos mécanismes de mobilité au niveau 802.11 ainsi que des techniques standard présentes dans l'implémentation Linux. Ainsi, des entrées dynamiques sont introduites manuellement dans les tables de routage et dans les tables ARP des hôtes mobiles et des routeurs du domaine, dans le but de minimiser les phases de reconfiguration des hôtes mobiles. Nous avons présenté des tests et des mesures de performance du prototype. Néanmoins, les tests ont été faits avec un nombre restreint de machines ; nous considérons que qu'une analyse plus étendue, éventuellement complétée par la simulation, peut être utile pour compléter nos tests.

Mobilité globale en utilisant des identificateurs d'hôte et translation locale d'adresses. Dans la dernière partie de la thèse, nous avons examiné les implications de ce type de mobilité, pour laquelle nous proposons une solution simple. Nous avons montré que l'unicité des identificateurs doit être seulement locale, d'où l'élimination de la partie de négociation présente dans les autres propositions comme VIP et VNAT. Nous faisons usage des mécanismes simple et largement utilisé dans l'Internet comme la translation locale d'adresses, les enregistrements DNS dynamiques.

10.2. Perspectives d'études futures

L'activité de recherche concernant la mobilité dans les réseaux est devenue aujourd'hui très importante et les perspectives sont évidemment nombreuses. Nous avons signalé dans la section précédente quelques-unes des limites de nos propositions. Nous allons aborder maintenant ces points en considérant les évolutions possibles et la direction que nous souhaitons prendre.

- Analyse des handoffs dans les variantes 802.11 a et g où le comportement sera probablement diffèrent, à cause du nombre différent des canaux, ainsi que de la méthode d'accès au médium différente. Nous nous attendons à voir des délais différents pour les scans et un autre comportement pour les handoff en situation de charge ;
- Notre protocole de distribution de charge a des points communs avec le nouveau sous-groupe de travail IEEE 802.11k, qui a comme domaine d'intérêt des mesures et l'échange des informations entre les composants d'un réseau sans-fil. Nous devons voir comment on peut intégrer et continuer notre travail en fonction des extensions discutées au sein de ce groupe ;
- Notre protocole de micro-mobilité doit être testé à une échelle plus large, et comme nous avons mentionné plus haut, nous envisageons d'utiliser une simulation du protocole. Ainsi, comme plusieurs éléments d'implémentation se basent sur des fonctionnalités standard Linux, on doit s'assurer si les mêmes fonctionnalités sont présentes dans d'autres systèmes Unix ou dans Windows ou MacOS ;

- Pour notre protocole de mobilité globale, on peut faire la même observation : si la translation d'adresses et l'utilisation d'un proxy DNS peuvent être facilement appliquées à d'autres systèmes d'exploitation, il n'est pas de même pour l'interception des appels système. Une autre remarque est que nos tests ont été fait entre des machines proche l'une de l'autre, et les périodes de déconnexion pratiquement nulles. Nous pensons qu'il serait utile de tester le fonctionnement dans d'autres contextes et de voir comment le changement de type de réseau et les délais de déconnexion plus longs vont jouer sur le bon fonctionnement de notre protocole.

Annexes

A. Abréviations

AAA	*Authentication, Authorization, and Accounting*
ACK	*Acknowledgement*
AGUA	*Aggregatable Global Unicast Address*
AP	*Access Point*
API	*Application Programming Interface*
ARP	*Address Resolution Protocol*
ARPANET	*Advanced Research Projects Agency Network*
BSS	*Basic Service Set*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
CTS	*Clear to Send*
DCCP	*Datagram Congestion Control Protocol*
DCF	*Distributed Coordination Function*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
DoS	*Denial Of Service*
DSSS	*Direct Sequence Spread Spectrum*
ESS	*Extended Service Set*
ESSID	*Extended Service Set ID*
EUI-64	*Extended Unique Identifier on 64 bits*
FHSS	*Frequency Hopping Spread Spectrum*
FQDN	*Fully-Qualified Domain Name*
GPRS	*General Packet Radio Service*
HIP	*Host Identity Protocol*
HMAC-MD5	*Hashed Message Authentication Code with Message Digest version 5*
HR/DSSS	*High Rate Direct Sequence Spread Spectrum*
HTTP	*HyperText Transfer Protocol*
IAPP	*Inter Access Point Protocol*
IBSS	*Independant Basic Service Set*
IEEE	*Institute of Electrical and Electronic Engineers*

IETF	*Internet Engineering Task Force*
IMAG	Informatique et Mathématiques Appliquées de Grenoble
ICMP	*Internet Control Message Protocol*
IP	*Internet Protocol*
IPSec	*Internet Protocol Security*
IPv4	*Internet Protocol, version 4*
IPv6	*Internet Protocol, version 6*
IR	*InfraRed*
ISO	*International Standard Organisation*
LLC	*Logical Link Control*
LSR	Logiciels Systèmes Réseaux
MAC	*Medium Access Control*
MANET	*Mobile Ad Hoc Network*
MTU	*Maximum Transmission Unit*
NAT	*Network Address Translation*
OFDM	*Orthogonal Frequency Division Multiplexing*
OSI	*Open Systems Interconnection*
OUI	*Organizationally Unique Identifier*
P2P	*Peer to Peer*
PC	*Personal Computer*
PCF	*Point Coordination Function*
RADIUS	*Remote Authentification Dial-In User Service*
RTP	*Real-Time Transport Protocol*
RTS	*Request to Send*
SCTP	*Stream Control Transmission Protocol*
SDP	*Session Description Protocol*
SHA-1	*Secure Hash Algorithm*
SNR	*Signal to Noise Ratio*
SSID	*Service Set ID*
TCP	*Transmission Control Protocol*
TTL	*Time To Live*
UDP	*User Datagram Protocol*
UMTS	*Universal Mobile Telecommunications System*
WEP	*Wired Equivallent Privacy*
Wi-Fi	*Wireless Fidelity*
WPA	*Wi-Fi Protected Access*
WPA2	*Wi-Fi Protected Access version 2*

B. Bibliographie

[1] G. Brasche and B. Walke. Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service. *IEEE Communications Magazine*, 35(8):94–104, Août 1997.

[2] K. Richardson. UMTS Overview. *IEEE Electronics & Communication Engineering Journal*, 12(3):93–100, Juin 2000.

[3] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. ISO/IEC DIS 8802-11:1997, IEEE Std. 802.11-1997.

[4] Site web de Wi-Fi Alliance. http://www.wi-fi.org.

[5] Site web de Bluetooth SIG. http://www.bluetooth.com.

[6] K. G. Coffman and A. M. Odlyzko. Internet Growth: is there a "Moore's law" for Data Traffic? *Handbook of Massive Data Sets*, pages 47–93, 2002.

[7] L. Press. The State of the Internet: Growth and Gaps. In *Proceedings of the the 10th Annual Internet Society Conference (INET 2000)*, Yokohama, Japon, Juillet 2000.

[8] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3):94–10, Septembre 1991.

[9] J. Waldo. When the Network is Everything. *Communications of the ACM*, 44(3):68–69, Mars 2001.

[10] D. Tennenhouse. Embedding the Internet: Proactive Computing. *Communications of the ACM*, 43(5):43–43, Mai 2000.

[11] L. Kleinrock. Information Flow in Large Communication Nets. Quarterly Progress Report 62, MIT Research Laboratory of Electronics (RLE), Juillet 1961.

[12] B.M. Leiner, V.G. Cerf, D.D. Clark, R.E. Kahn, L. Kleinrock, D.C. Lynch, J. Postel, L.G. Roberts, and S.S. Wolff. The Past and Future History of the Internet. *Communications of the ACM*, 40(2):102–108, Février 1997.

[13] L. Kleinrock. Breaking Loose. *Communications of the ACM*, 44(9):41–46, Septembre 2001.

[14] R. Droms. Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard), Mars 1997. Mis à jour par RFC 3396.

[15] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315 (Proposed Standard), July 2003.

[16] S. Cheshire, B. Aboba, and E. Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927 (Proposed Standard), Mars 2005.

[17] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462 (Draft Standard), Décembre 1998.

[18] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC 1122 (Standard), Octobre 1989. Mis à jour par RFC 1349.

[19] R. Braden. Requirements for Internet Hosts - Application and Support. RFC 1123 (Standard), Octobre 1989. Mis à jour par RFCs 1349, 2181.

[20] V. Cerf and R. Kahn. A Protocol for Packet Network Intercommunications. *IEEE Transactions on Communications*, 22(5):637–648, Mai 1974.

[21] V. Cerf, Y. Dalal, and C. Sunshine. Specification of Internet Transmission Control Program. RFC 675, Décembre 1974.

[22] ANSI/ISO. Information Processing Systems - Basic Reference Model for Open Systems Interconnection (OSI). ISO/IEC 7498-1:1994.

[23] J. Postel. Internet Protocol. RFC 791 (Standard), Septembre 1981. Mis à jour par RFC 1349.

[24] IANA. Special-Use IPv4 Addresses. RFC 3330 (Informational), Septembre 2002.

[25] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. ISO/IEC DIS 8802-3:2002, IEEE Std. 802.3-2002.

[26] IEEE Standards for Information Technology. IEEE Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. IEEE Std. 802.11f-2003.

[27] P.V. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034 (Standard), Novembre 1987. Mis à jour par RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035.

[28] P.V. Mockapetris. Domain Names - Implementation and Specification. RFC 1035 (Standard), Novembre 1987. Mis à jour par RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035.

[29] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. In *Proceedings of the 1st ACM SIGCOMM Internet Measurement Workshop*, pages 153–167, San Francisco, CA, États-Unis, Novembre 2001.

[30] JH Software. DNS Caching and How It Affects Simple Failover. http://www.simplefailover.com/outbox/dns-caching.pdf. Livre Blanc.

[31] AOL. Site web d'informations sur le DNS. Time To Live (TTL). http://dns.info.aol.com/time.shtml.

[32] J. Postel. Transmission Control Protocol. RFC 793 (Standard), Septembre 1981. Mis à jour par RFC 3168.

[33] K. Thompson, G. J. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE/ACM Transactions on Networking*, 11(6):10–23, Novembre 1997.

[34] J. Postel. User Datagram Protocol. RFC 768 (Standard), Août 1980.

[35] J.F. Shoch. Inter-Network Naming, Addressing, and Routing. In *Proceedings of IEEE Computer Conference (COMPCON)*, pages 72–79, Washington, DC, États-Unis, Fall 1978.

[36] J.H. Saltzer. On the Naming and Binding of Network Destinations. In *Proceedings of the IFIP/TC6 International Symposium on Local Computer Networks*, pages 311–317, Florence, Italie, Avril 1982.

[37] C. Perkins. IP Mobility Support. RFC 2002 (Proposed Standard), Octobre 1996. Obsoleted by RFC 3220, Mis à jour par RFC 2290.

[38] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), Août 2002.

[39] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), Juin 2004.

[40] J. Postel. Internet Control Message Protocol. RFC 792 (Standard), Septembre 1981. Mis à jour par RFC 950.

[41] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104 (Informational), Février 1997.

[42] D.C. Plummer. Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826 (Standard), Novembre 1982.

[43] C. Perkins. IP Encapsulation within IP. RFC 2003 (Proposed Standard), Octobre 1996.

[44] N. A. Fikouras, K. El Malki, S.R. Cvetkovic, and M. Kraner. Performance Analysis of Mobile IP Handoffs. In *Proceedings of the Asia Pacific Microwave Conference (ACMP'99)*, volume 3, pages 770–773, Singapour, Décembre 1999.

[45] A. Fladenmuller and R. De Silva. The Effect of Mobile IP Handoffs on the Performance of TCP. *Journal on Mobile Networks and Applications (MONET)*, 4(2):131–135, Juin 1999.

[46] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice), Mai 2000. Mis à jour par RFC 3704.

[47] T. Killalea. Recommended Internet Service Provider Security Services and Procedures. RFC 3013 (Best Current Practice), Novembre 2000.

[48] G. Montenegro. Reverse Tunneling for Mobile IP, revised. RFC 3024 (Proposed Standard), Janvier 2001.

[49] C. Perkins and D. Johnson. Route Optimisation in Mobile IP. IETF Internet Draft draft-ietf-mobileip-optim-11, Septembre 2001. (Travaux en cours).

[50] P. Calhoun, G. Montenegro, C. Perkins, and Gustafsson E. Mobile IPv4 Regional Registration. IETF Internet Draft draft-ietf-mobileip-reg-tunnel-09, Juillet 2004. (Travaux en cours).

[51] G. Kirby. Locating the User. *Communications International Magazine*, 1995(10):25–28, Octobre 1995.

[52] C. Toh. The Design and Implementation of a Hybrid Handover Protocol for Multimedia Wireless LANs. In *Proceedings of First ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'95)*, pages 49–61, Berkeley, CA, États-Unis, Novembre 1995.

[53] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501 (Informational), Janvier 1999.

[54] S.E. Deering. Host extensions for IP multicasting. RFC 1112 (Standard), Août 1989. Mis à jour par RFC 2236.

[55] J. Mysore and V. Bharghavan. A New Multicasting-based Architecture for Internet Host Mobility. In *Proceedings of the 3rd ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '97)*, pages 161–172, Budapest, Hongrie, Septembre 1997.

[56] A. Helmy. A Multicast-based Protocol for IP Mobility Support. In *Proceedings of the ACM Second International Workshop on Networked Group Communication (NGC 2000)*, pages 49–58, Palo Alto, CA, États-Unis, Novembre 2000.

[57] R. Ramjee, K. Varadhan, L. Salgarelli, S.R. Thuel, S.Y. Wang, and T.F. La Porta. HAWAII: A Domain-based Approach for Supporting Mobility in Wide-Area Wireless Networks. *IEEE/ACM Transactions on Networking*, 10(3):396–410, Juin 2002.

[58] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. IP Micro-mobility Support Using HAWAII. IETF Internet Draft draft-ietf-mobileip-hawaii-01, Juillet 2000. (Travaux en cours).

[59] A.T. Campbell, J. Gomez, S. Kim, A.G. Valko, Chieh-Yih W., and Z.R. Turanyi. Design, Implementation, and Evaluation of Cellular IP. *IEEE Personal Communications*, 7(4):42–49, Août 2000.

[60] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Transactions in Computer Systems*, 2(4):277–288, Novembre 1984.

[61] Site web du projet WIDE. http://www.wide.ad.jp.

[62] M. Ishiyama, M. Kunishi, K. Uehara, H. Esaki, and F. Teraoka. LINA: A New Approach to Mobility Support in Wide Area Networks. *IEICE Transactions on Communication*, E84-B(8):2076–2086, Août 2001.

[63] M. Ishiyama, M. Kunishi, and F. Teraoka. An Analysis of Mobility Handling in LIN6. In *Proceedings of the Fourth Inernational Sympsium on Wireless Personal Multimedia Communications (WPMC'01)*, Aalborg, Danemark, Septembre 2001.

[64] R. Hinden, S. Deering, and E. Nordmark. IPv6 Global Unicast Address Format. RFC 3587 (Informational), Août 2003.

[65] IEEE Standards Association. Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority. http://standards.ieee.org/regauth/oui/tutorials/EUI64.html, Mars 1997.

[66] P. Yalagandula, A. Garg, M. Dahlin, L. Alvisi, and H. Vin. Transparent Mobility with Minimal Infrastructure. Technical Report CS-TR-01-30, University of Texas, Austin, TX, États-Unis, Juillet 2001.

[67] G. Su and J. Nieh. Mobile Communication with Virtual Network Address Translation. Technical Report CUCS-003-02, Department of Computer Science, Columbia University, New York, NY,États-Unis, Février 2002.

[68] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational), Janvier 2001.

[69] R. Moskowitz, P. Nikander, and T. Henderson. Host Identity Protocol. IETF Internet Draft draft-ietf-hip-base-03, Juin 2005. (Travaux en cours).

[70] P. Nikander, Y. Ylitalo, and J. Wall. Integrating Security, Mobility and Multi-Homing in a HIP Way. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'03)*, pages 87–99, San Diego, CA, États-Unis, Février 2003.

[71] P. Nikander, J. Arkko, and T. Henderson. End-Host Mobility and Multi-Homing with the Host Identity Protocol. IETF Internet Draft draft-ietf-hip-mm-01, Février 2005. (Travaux en cours).

[72] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), Novembre 1998. Mis à jour par RFC 3168.

[73] C. Huitema. Multi-homed TCP. IETF Internet Draft draft-huitema-multi-homed-01, Mai 1995. (Travaux en cours).

[74] A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 155–166, Boston, MA, Août 2000.

[75] W. Diffie and M. E. Hellman. Privacy and Authentication: An Introduction to Cryptography. In *Proceedings of the IEEE*, volume 67, pages 397–427, Mars 1979.

[76] NIST (National Institute of Standards and Technology). The Secure Hash Algorithm (SHA-1), Avril 1995. NIST FIPS PUB 180-1.

[77] D.A. Maltz and P. Bhagwat. MSOCKS: An Architecture for Transport Layer Mobility. In *Proceedings of the 17th IEEE Conference on Computer Communications (INFOCOM '98)*, volume 3, pages 1037–1045, San Francisco, CA, États-Unis, Mars 1998.

[78] D.A. Maltz and P. Bhagwat. TCP Splicing for Application Layer Proxy Performance. Research Report RC 21139, IBM TJ Watson Research Center, Mars 1998.

[79] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. RFC 1928 (Proposed Standard), Mars 1996.

[80] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS'95)*, pages 136–143, Vancouver, Canada, Mai 1995.

[81] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard), Octobre 2000. Mis à jour par RFC 3309.

[82] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. IETF Internet Draft draft-ietf-tsvwg-addip-sctp-11, Février 2005. (Travaux en cours).

[83] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), Juin 1999. Mis à jour par RFC 2817.

[84] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), Juin 2002. Mis à jour par RFCs 3265, 3853.

[85] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *ACM SIGMOBILE Mobile Computing and Communications Review*, 4(3):47–57, Juillet 2000.

[86] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), Juillet 2003.

[87] F. Vakil, A. Dutta, J-C. Chen, M. Tauil, S. Baba, N. Nakajima, and H. Schulzrinne. Supporting mobility for tcp with sip. IETF Internet Draft draft-itsumo-sipping-mobility-tcp-00, Juin 2001. (Travaux en cours).

[88] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Supplement to 802.11-1997, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band. IEEE Std. 802.11-1999.

[89] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification – Amendment 1: High-speed Physical Layer in the 5 GHz Band. ISO/IEC DIS 8802-11:1999/Amd 1:2000(E), IEEE Std. 802.11a-1999.

[90] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification – Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band. IEEE Std. 802.11g-2003.

[91] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 2: Logical Link Control (LLC). ISO/IEC 8802-2:1998, IEEE Std. 802.2-1998.

[92] J. Wright. Detecting Wireless LAN MAC Address Spoofing. http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf. White Paper.

[93] S. R. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, pages 1–24, Toronto, Canada, Août 2001.

[94] Wi-Fi Alliance. Wi-Fi Protected Access. http://www.wi-fi.org/OpenSection/protected_access.asp.

[95] Wi-Fi Alliance. Wi-Fi Protected Access version 2. http://www.wi-fi.org/OpenSection/protected_access.asp.

[96] IEEE Standards for Information Technology. Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification – Amendment 6: Medium Access Control (MAC) Security Enhancements. IEEE Std. 802.11i-2004.

[97] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard), Juin 2004.

[98] IEEE Standards for Local and Metropolitan Area Networks. Port-Based Network Access Control. IEEE Std. 802.1X-2004.

[99] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). RFC 2865 (Draft Standard), Juin 2000. Mis à jour par RFCs 2868, 3575.

[100] J. Malinen. Pilote Linux HostAP pour les chipsets Intersil Prism. http://hostap.epitest.fi/.

[101] J. Tourrilhes. Extensions sans-fil (Wireless Extensions) pour Linux. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.Extensions.html.

[102] J. Tourrilhes. Outils sans-fil (Wireless Tools) pour Linux. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html.

[103] M. Balazinska and P. Castro. Characterizing Mobility and Network États-Unisge in a Corporate Wireless Local-area Network. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys'03)*, pages 303–316, San Francisco, CA, États-Unis, Mai 2003.

[104] D. Kotz and K. Essien. Characterizing États-Unisge of a Campus-Wide Wireless Network. Technical Report TR2002-423, Dartmouth College, Mars 2002.

[105] A. Balachandran, G. Voelker, P. Bahl, and V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Proceedings of the 2002 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'02)*, pages 195–205, Marina Del Rey, CA, États-Unis, Juin 2002.

[106] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proceedings of the 22nd IEEE Conference on Computer Communications (INFOCOM '03)*, volume 2, pages 836–843, San Francisco, CA, États-Unis, Avril 2003.

[107] J. N. Chiappa. Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture. Draft Available at http://users.exis.net/~jnc/tech/endpoints.txt.

[108] E. Kohler, M. Handley, and S. Floyd. Datagram Congestion Control Protocol (DCCP). IETF Internet Draft draft-ietf-dccp-spec-11, March 2005. (Travaux en cours).

[109] R. Hinden and S. Deering. Internet Protocol Version 6 (IPv6) Addressing Architecture. RFC 3513 (Proposed Standard), April 2003.

Gestion de la mobilité dans les réseaux ambiants

Résumé

Cette thèse a été motivée par le nouveau contexte ubiquitaire des réseaux ambiants sans-fil. Les protocoles Internet ont été conçus il y a 30 ans sans prendre en considération l'usage nomade des réseaux et ne répondent pas aux nouvelles contraintes de la mobilité. Notre but a été de concevoir des mécanismes pour rendre la mobilité transparente aux applications et utilisateurs. Une partie de notre travail a été axé sur l'amélioration du délai du handoff au niveau physique 802.11 à quelques dizaines de millisecondes. Le déplacement des hôtes et le changement de réseau demandent souvent une reconfiguration des plusieurs paramètres au niveau IP. Dans le cas d'une mobilité locale, l'approche que nous avons choisi est de permettre aux hôtes de garder inchangées leurs adresses et de mettre en place des routes d'hôte dans le domaine local. Au contraire, pour la mobilité globale, le routage optimal à travers l'Internet nous oblige de concevoir une solution dans laquelle les hôtes mobiles changent leurs adresses IP, en conformité avec le réseau ou ils s'y trouvent. Notre solution est basée sur le principe général « de bout en bout » dans laquelle les deux hôtes impliquées dans une connexion sont les seuls à assurer le transfert de cette connexion aux nouveaux points d'attachement. Elle utilise l'interception des appels de la bibliothèque socket et des requêtes DNS ainsi que la translation locale d'adresses pour virtualiser les adresses IP réelles dans des identificateurs d'hôte stables présentés aux niveaux supérieurs.

Mots clés : mobilité, handoff, réseaux sans-fil, 802.11, micro-mobilité, routes d'hôte, couche 3.5, identificateurs d'hôte, sockets, NAT.

Mobility Management in Ambient Networks

Abstract

This thesis was motivated by the new ubiquitous context of ambient wireless networks. The Internet protocols were designed 30 years ago without taking into consideration the nomadic use of networks and do not respond to the new constraints of mobility. We aimed to conceive mechanisms to make the mobility transparent to the applications and users. A part of our work was focused on the improvement of handoff delay of physical layer 802.11 to about 20 ms. Host mobility and network changes often require the reconfiguration of several parameters at IP layer. For the local mobility we chosen to allow the hosts to keep theirs IP address unchanged and to propagate host routes in the local domain. On the contrary, for global mobility, the optimal routing in Internet force us to conceive a solution where the mobile hosts change their IP addresses according to the subnet there are connected to. Our solution is based on "end-to-end" paradigm where the two hosts implied in a connection are the only ones to ensure the connection transfer to the new attachment points. It use interception of calls to the socket library and of DNS requests as well as local address translation to virtualise IP addresses into stable host identifiers that are presented to upper layers.
Keywords : mobility, handoff, wireless networks, 802.11, micro-mobility, host routes, layer 3.5, host identifiers, sockets, NAT.

Discipline : Informatique, Systèmes et Communications

Laboratoire : LSR-IMAG – Équipe Drakkar
BP 72, 38402 Saint Martin d'Hères Cedex, France