Communications dans les réseaux fortement dynamiques

Florent Kaisser

N° D'ORDRE

THÈSE DE DOCTORAT

SPECIALITE : PHYSIQUE

Ecole Doctorale « Sciences et Technologies de l'Information des Télécommunications et des Systèmes »

Présentée par :

FLORENT KAISSER

Sujet :

Communications dans les réseaux fortement dynamiques

Soutenue le 21 juin 2010 devant les membres du jury :

- Mme Véronique Vèque (Directrice de thèse)
- Mme Colette Johnen (Directrice de thèse)
- Mr Bertrand Ducourthial (Rapporteur)
- Mme Houda Labiod (Rapporteur)
- Mme Monique Becker (Examinatrice)
- Mr Mohamed Naïmi (Examinateur)

À mes parents

Remerciements

Je remercie mes directrices Pr Véronique Vèque et Pr Colette Johnen pour leurs précieux conseils, leur confiance et surtout pour la liberté qu'elles m'ont données dans la réalisation des mes travaux durant ces quatre dernières années. Je les remercie également pour leur patience et leurs disponibilités durant nos nombreuses entrevues.

Je remercie les rapporteurs, Dr Houda Labiod et Dr Bertrand Ducourthial, pour leurs disponibilités le jour de la soutenance et leurs nombreuse remarques pour améliorer ce manuscrit. Je remercie également les examinateurs Pr Mohamed Naïmi et Pr Monique Becker.

Je remercie la région Île-de-France pour le financement des trois premières années de recherches. Je remercie l'Université de Cergy-Pontoise où j'ai non seulement enseigné pendant ces quatre dernières années, mais où j'ai également reçu un enseignement de qualité durant les quatre premières années de ma formation supérieure.

Je remercie mes collègues et amis pour la bonne ambiance de travail : Alvaro, Ana, Amel, Amina, David, Fatma, Féh, Husnain, José, Lynda, Minh, Muriel, Saioa, Stephane.

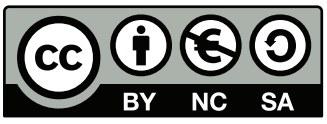
Un grand merci à mon ami François Destelle, qui m'a accompagné durant toute ma thèse, pour ses précieux conseils, relectures et surtout, son soutien inconditionnel. Merci à un autre ami, Piotr, pour son aide technique et pour m'avoir fait découvrir un grand nombre d'outils informatiques qui m'ont facilité le travail durant mes recherches. Je le remercie également pour son exigence typographique.

Je remercie ma compagne, Fatima, pour son soutien moral et les formidables séjours en Algérie durant ma thèse.

Enfin, je remercie infiniment mes parents pour leur éducation, leur soutien, leurs encouragements et leur inspiration tout au long de ma vie. Je remercie mon père pour m'avoir fait découvrir l'informatique quand j'étais tout petit.

Notes de l'auteur

Ce document et l'ensemble des travaux associés ont été réalisés exclusivement à l'aide de logiciels libres ou open source. Les figures au format vectoriel ont été réalisées à l'aide d'Inkscape. Kile à été utilisé comme environnement d'édition des fichiers LaTeX. Pour les simulations, les langages utilisés sont Java et Python. Eclipse à été utilisé comme environnement de développement Java. Ce document et le code source des simulations sont versionnés à l'aide du très pratique gestionnaire de version Mercurial. Tout cela fonctionnant avec la distribution Ubuntu Linux. Merci aux contributeurs de ces logiciels.



Le végétarisme, par son action purement physique sur la nature humaine, influerait de façon très bénéfique sur la destinée de l'humanité.

Albert Einstein

Table des matières

Table des figures

Introduction

L'explosion d'Internet ces dix dernières années, précédée par l'engouement pour la téléphonie mobile, ont mis au centre de notre vie les réseaux de télécommunications. À ce jour, les nouveaux services proposés par les opérateurs de télécommunications pour les terminaux mobiles nécessitent un accès à l'Internet disponible partout et tout le temps. Les techniques nécessaires à la mise en œuvre d'un tel réseau nécessite une gestion de la mobilité et une connectivité à l'Internet. L'approche actuelle des opérateurs est un réseau centralisé avec une gestion de mobilité au niveau de l'infrastructure. La gestion décentralisée de la mobilité est une autre approche développée dans de nombreux travaux de recherche, cependant peu déployée à ce jour. Dans un tel réseau, l'ensemble des nœuds peuvent être terminaux comme routeur, mobile comme fixe (Figure 1). Nous nous intéressons à ce type de gestion de la mobilité décentralisée, plus particulièrement dans un contexte fortement dynamique où la topologie du réseau change fréquemment. Nous avons choisi un cas d'étude particulier ayant certaines propriétés simplifiant son étude : les réseaux de véhicules, ou VANET, selon le terme le plus utilisé par la littérature anglophone.

Dans le domaine des systèmes de transports intelligents (ITS), les communications sans fil entre véhicules (V2V) apparaissent comme une solution à la prévention des accidents en offrant une vision plus étendue que les traditionnels capteurs. En reliant les véhicules à un réseau de télécommunications (V2I), de nouvelles perspectives sont offertes tant aux passagers qu'au conducteur avec des applications de communication classiques telles que l'accès Internet, les jeux ou le tchatche. Les réseaux de véhicules sont une technologie émergente intégrant les dernières techniques de communication. Un réseau de véhicules fournit (1) une connectivité au monde extérieur par l'intermédiaire de passerelles vers d'autres réseaux, et (2) une *communication inter-véhiculaire* pour les *véhicules intelligents*. Sans infrastructure, le réseau est décentralisé, un protocole de routage ad hoc doit être utilisé pour assurer les communications inter-véhiculaires.

Nous nous sommes essentiellement intéressés à trois problèmes : la gestion de la forte dynamicité, le passage à l'échelle et l'augmentation de la connectivité. Ces trois problématiques sont intrinsèques aux réseaux de véhicules sur autoroute. Le déplacement des véhicules à grande vitesse implique une forte dynamique du réseau. L'étendue d'un réseau autoroutier impose une bonne résistance au passage à l'échelle. Selon l'environnement (au-

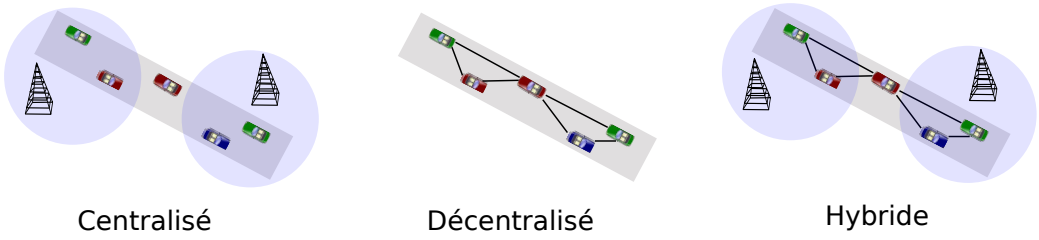


Figure 1 – Trois types de réseaux mobiles

toroutier, urbain ou rural) la densité et la vitesse des véhicules n'ont pas le même impact sur la topologie et sa dynamicité. Si l'on considère deux véhicules roulant en sens opposé à 25 m/s (90 km/h), avec une couverture radio de 250 m, alors la durée de la communication directe entre les véhicules est seulement de 10 secondes. Par conséquent, tout au long de nos travaux, nous avons pris les réseaux de véhicules sur autoroute comme cas d'étude de ces problématiques permettant d'une part d'aborder les problèmes des réseaux mobiles en général, et d'autre part de faciliter leur étude en profitant de leurs propriétés.

Les travaux existants révèlent un désaccord sur la comparaison de deux grandes classes de protocoles dans un contexte autoroutier : basée sur la topologie et basée sur la position géographique. La seconde classe prenant en compte l'information géographique obtenue à l'aide d'un système de positionnement comme GPS, sachant que la première classe ayant aucune de ces informations. Trancher la question est difficile car les travaux comparant ces classes de protocoles utilisent des approches de simulation différentes et aboutissent à des conclusions divergentes. Nous avons établi un modèle quantitatif de comparaison du passage à l'échelle de ces deux classes de protocoles de routage. Il prend en compte les dernières optimisations développées pour les protocoles basés sur la position géographique.

Nous avons étudié l'amélioration de la connectivité du réseau en y ajoutant une infrastructure (composée de points d'accès fixes). Les points d'accès disposés le long des voies ne couvrent pas toute la route mais ce réseau est étendu par des communications inter-véhiculaires à plusieurs sauts. Le réseau de véhicules devient alors hybride (Figure 1). Le protocole permettant la communication inter-véhiculaire doit être adapté et optimisé pour gérer les fréquents changements de topologie. Pour cela nous avons amélioré un simulateur de réseaux (JIST/SWANS) permettant d'évaluer les performances des protocoles existants et trouver de nouvelles techniques afin de les améliorer.

La dernière partie de ce manuscrit porte sur le développement d'un protocole permettant la gestion de convois de véhicules. Le but est de grouper plusieurs véhicules ayant des caractéristiques de stabilité proches en s'inspirant des algorithmes de formation de grappes de nœuds. La communication à l'intérieur de ces convois est réalisée grâce à un protocole ad hoc basé sur la position géographique. Ce groupe composé de plusieurs véhicules se connecte alors à l'infrastructure directement. Une telle méthode permet de mieux gérer la mobilité dans un réseau ad hoc hybride de véhicules sur autoroute. Notre protocole est également évalué à l'aide du simulateur JIST/SWANS.

Pour aborder l'ensemble des travaux traités dans ce manuscrit, nous avons besoin de notions allant des réseaux sans fil jusqu'aux réseaux de véhicules sur autoroute en passant

par la gestion de la mobilité dans les réseaux IP et les réseaux ad hoc. L'ensemble de ces notions sont abordées dans les deux premiers chapitres concernant les réseaux mobiles et les réseaux ad hoc. Un troisième chapitre traite de notre apport au principal outil d'évaluation de performances que nous utilisons : le simulateur JIST/SWANS. Enfin, les trois derniers chapitres concernent nos travaux décrits précédemment.

Chapitre 1

Les réseaux mobiles

Les réseaux de véhicules sont une technologie émergente intégrant les dernières techniques de communications. Ils permettent, d'une part, une communication directe entre les véhicules et d'autre part, une communication entre un véhicule et un correspondant extérieur. De nombreuses applications de réseaux de véhicules sont possibles : sécurité, confort des passagers, services d'information de proximité, information de trafic en temps réel, etc. Les défis pour la mise en place d'un tel réseau sont nombreux. Le déplacement très rapide des véhicules (130 km/h sur les autoroutes françaises) impose des transmissions sans fil à la fois fiables et adaptées, mais également la gestion de la mobilité, c'est-à-dire le changement de position des véhicules par rapport aux autres véhicules et par rapport à une infrastructure communicante mise en place aux abords de la route.

Dans ce chapitre premier d'état de l'art, nous introduisons les réseaux sans fil en décrivant les technologies associées existantes à ce jour. La majorité des réseaux se basant sur IP, nous aborderons la mobilité dans les réseaux IP pour ensuite introduire cette mobilité dans les réseaux ad hoc au chapitre suivant.

1.1 Généralités sur les réseaux sans fil

Depuis longtemps, les communications sans fil ont existé grâce à des signaux optiques comme le feu, la fumée, des signes, mais aussi grâce au son, comme le langage sifflé. Le premier réseau de télécommunications sans fil de grande envergure est apparu en 1794 avec le Télégraphe Chappe utilisant un moyen de communication optique par sémaphore, étendant le réseau sur des centaines de kilomètres. Cependant en 1895, avec les expériences de Guglielmo Marconi, la transmission sans fil (TSF) à l'aide d'ondes électromagnétiques fait son apparition et va devenir le meilleur moyen de transmettre de l'information via le télégraphe et le langage Morse. Rapidement, les communications sans fil se sont développées dans de nombreux domaines comme l'aéronautique, les voitures (CB), etc. Entre temps le réseau téléphonique filaire s'est répandu dans le monde et, depuis les années 1980, l'idée de pouvoir téléphoner partout à n'importe qui a donné naissance au réseau téléphonique mobile (ou *radiotéléphone*).

Plus tard, avec l'augmentation de l'utilisation des ordinateurs personnels mobiles, les réseaux informatiques sans fil ont fait leur apparition, leur but étant de relier les ordinateurs dans un espace réduit (pièce, bâtiment). Rapidement, le besoin d'être connecté à Internet partout est devenu une problématique importante des réseaux informatiques sans-fil.

Avant de décrire plus en détails les réseaux sans fil, nous allons présenter les réseaux informatiques.

1.1.1 Les réseaux informatiques

Les réseaux informatiques interconnectent plusieurs équipements pour échanger des informations entre eux. Le réseau est composé d'un ensemble de *nœuds*. Chaque nœud communique avec d'autres nœuds à l'aide d'une *interface réseau* qui peut être *filaire* ou *sans fil*. Un nœud peut être un terminal comme un téléphone portable, un équipement informatique comme un PC, un capteur pour acquérir des données sur l'environnement (température, pression, etc.), un routeur dédié, un point d'accès sans fil... Un nœud possède un *identifiant* (ou ID) unique. Cet identifiant peut être son adresse IP, son adresse MAC, son nom d'hôte, etc. On appelle *paquet* ou *message* la partie élémentaire des transmissions de données dans le réseau. On appelle *message de signalisation* un paquet ne contenant pas de données utilisateur, mais des données utiles au fonctionnement des protocoles de routages. Un paquet contenant des données destinées aux applications utilisateurs peut contenir également des données utiles au routage. Les données utilisées pour le routage sont appelées l'*en-tête* d'un paquet. L'ensemble des messages de signalisation et celui des en-têtes de paquets constituent l'*overhead* ou surplus de gestion du protocole.

Lorsque deux nœuds s'échangent des données alors le *nœud source* est celui qui envoie un paquet, et le *nœud destination* ou *nœud correspondant* est celui qui reçoit un paquet. Par abus de langage, on utilise simplement *source* ou *destination* au lieu de nœud source ou nœud destination.

Les réseaux informatiques sont classés suivant leur portée :
- le réseau personnel (PAN) relie des appareils électroniques personnels,
- le réseau local (LAN) relie les ordinateurs et/ou les postes téléphoniques situés dans la même pièce ou dans le même bâtiment,
- le réseau métropolitain (MAN) est un réseau à l'échelle d'une ville,
- le réseau étendu (WAN) est un réseau à grande échelle qui relie plusieurs sites ou des ordinateurs du monde entier.

1.1.1.1 Méthodes de transmission

On dit qu'un nœud *transmet* un message s'il envoie le message reçu à un de ses voisins.

Dans un réseau, un message peut être transmis suivant trois façons :

- *Unicast* ou Point-à-point : la source envoie un message à un nœud particulier.
- *Broadcast* ou diffusion : la source envoie un message à tous les nœuds du réseau.
- *Multicast* : la nœud envoie un message à un ensemble de nœuds du réseau.

– *Géocast* : transmission multicast où le message est reçu par chaque nœud d'une zone géographique déterminée.

On dit qu'un message est *inondé* ou *diffusé* sur le réseau, s'il est transmis en mode broadcast.

1.1.2 Les réseaux sans fil

La transmission sans fil utilise les ondes radio pour transférer les informations entre stations. Leur principal avantage est de permettre la mobilité. Il existe alors dans le réseau sans fil des nœuds appelés *nœuds mobiles* ou *stations mobiles* communiquant avec d'autres nœuds mobiles ou des nœuds fixes d'une *infrastructure*. La disposition de ces nœuds par rapport aux autres constitue la *topologie* du réseau. Une *interface radio* est utilisée pour la communication sans fil avec d'autres nœuds. On dit que deux nœuds sont *voisins* s'ils peuvent communiquer entre eux. Le *voisinage* d'un nœud est alors l'ensemble de ses voisins.

Deux nœuds sont accessibles si leur interface radio est réglée sur la même *fréquence*, si la puissance du signal reçu est au delà d'un seuil et s'il permet d'être décodé. La puissance du signal décroît avec la distance entre deux nœuds : plus les nœuds sont éloignés, moins le signal est fort. Un signal suffisamment fort ne suffit pas pour pouvoir décoder son contenu, car des interférences se produisent lors de la propagation et réduisent la qualité de réception du signal. Ces interférences peuvent provenir d'un signal d'une fréquence proche (si les fréquences ne sont pas orthogonales), de parasites (objets émettant des ondes électromagnétiques), ou de phénomènes de multi-chemins résultant du rebond de l'onde sur une surface.

Nous allons détailler deux types de réseaux sans fil : le réseau de radio téléphone cellulaire et le réseau local sans fil (WLAN).

1.1.3 Réseau de radio téléphone cellulaire

La technique des réseaux de radiotéléphones cellulaires a permis la rapide expansion de la téléphonie mobile dans le monde à partir de sa seconde génération (2G) . Il existe plusieurs normes de réseaux cellulaires de seconde génération. La norme GSM (Global System for Mobile Communication) a été adoptée par l'Europe et dans de nombreux pays dans le monde, la norme IS-95 (Interim Standard 95) au Etat-Unis, PDC (Personal Digital Cellular) au Japon.

L'architecture et la terminologie présentées par la suite est celle de la norme GSM. Le concept de base d'un réseau de radiotéléphone cellulaire est, d'une part, la division du territoire en un ensemble de zones appelées *cellules* et, d'autre part, le partage des canaux radio entre les cellules. La taille d'une cellule est comprise entre 200 m et 35 km selon le relief du territoire, la localisation (urbaine, rurale, suburbaine), ou bien la densité d'abonnés [120]. Dans chaque cellule d'un réseau cellulaire, la *station de base* (BS, pour "base station") est un émetteur-récepteur. Les stations de base sont interconnectées entre elles par une infrastructure filaire comprenant des commutateurs (MSC, pour Mobile Switching Centre) afin de relier le réseau mobile au réseau téléphonique fixe. A chaque cellule est associé un groupe de fréquences radio (aussi appelé canaux de communications). Deux cellules adjacentes n'ont

pas de canaux de communications communs. Pour éviter les interférences, une distance minimale de deux cellules est exigée pour séparer deux cellules utilisant les mêmes canaux de communications. La cellule est l'unité géographique d'un réseau. Un motif de sept cellules, chacunes utilisant des canaux différents, constitue un cluster. Ce motif peut être répété à l'infini pour couvrir un vaste territoire.

La division du réseau en cellules exige une gestion de la mobilité en et hors communication. En effet, lorsque l'utilisateur se déplace, il arrive fréquemment qu'il change de cellule. Pour garantir une qualité de service, la communication ne doit pas être coupée pendant un changement de cellule. Pour gérer la mobilité dans un réseau cellulaire, l'exploitant du réseau doit répondre aux prérequis suivants [120] :

- identifier chacun des utilisateurs,
- localiser chaque utilisateur,
- estimer la direction de déplacement des utilisateurs dans le réseau,
- maintenir les communications pendant un changement de cellule d'un utilisateur.

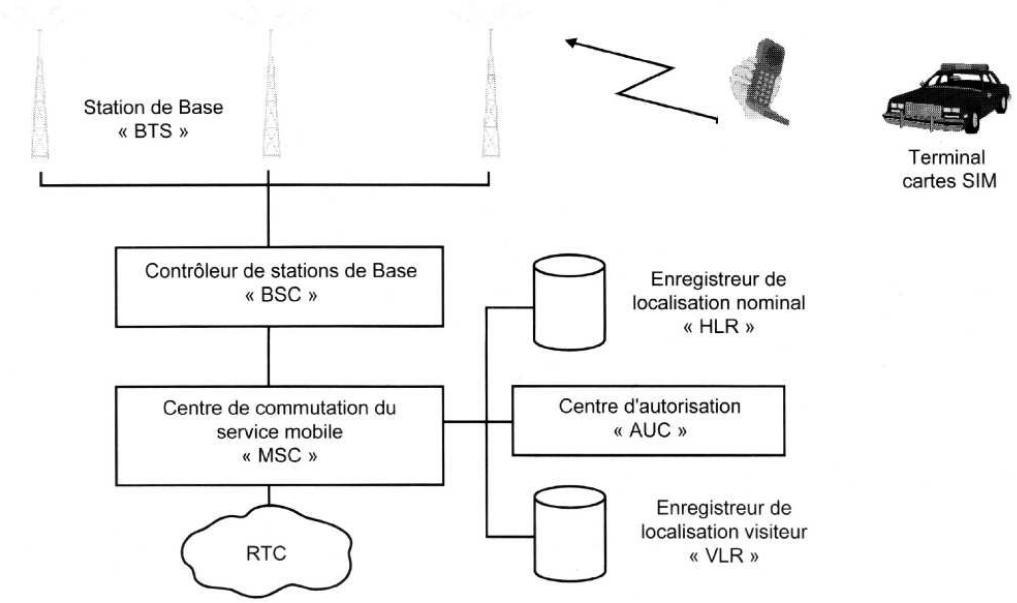


Figure 1.1 – Architecture d'un réseau GSM (*Issu de* [120])

Pour maintenir la communication entre deux cellules, le réseau, dans une première étape, synchronise deux stations de base sur le terminal de l'utilisateur : la station de base de la cellule qu'il quitte (l'ancienne station de base), et la nouvelle station de base de la cellule où il entre. Après un délai pour vérifier que le déplacement prévu est bien le déplacement effectué par l'utilisateur, un transfert intercellulaire, appelé *handover* ou *handoff*, est effectué. Le handover est effectué de sorte qu'il n'y ait aucune coupure de communication perceptible par l'utilisateur.

La mobilité entre différents réseaux (opérateur) est appelée *roaming*. Ce type de mobilité permet à l'utilisateur de se connecter dans un réseau différent (le réseau visité, *visited network*) de son réseau mère (*home network*). Si l'utilisateur n'est pas enregistré dans la base de

donnée HLR (Home Location Register) du réseau d'accueil, alors le réseau visité doit demander au réseau mère d'authentifier l'utilisateur et de récupérer les informations le concernant.

La Figure 1.1 représente l'architecture d'un réseau GSM permettant de répondre aux exigences de la mobilité des utilisateurs. Une base de données enregistre la localisation des visiteurs (VLR pour Visitor Location Register), et contient, entre autres, le *Location Area Code* correspondant à une zone contenant plusieurs BS. Lorsque la station mobile change de Location Area (ou tout simplement lorsqu'elle vient d'être allumé), elle émet un message de type Location Update pour indiquer dans quelle zone et dans quelle cellule elle se trouve.

1.1.4 Réseau local sans fil

Le réseau local sans fil, Wireless Local Area Network (WLAN) permet de relier plusieurs ordinateurs sans connexion filaire. La norme la plus utilisée à ce jour est le WiFi ou IEEE 802.11. Par rapport au modèle OSI, elle concerne la couche physique (1) et la couche liaison de données (2).

La portée radio est assez limitée, contraignant l'utilisation de ce type de réseau dans un bâtiment pour la grande majorité des cas. Néanmoins, elle fonctionne également dans des espaces extérieurs et, selon l'antenne utilisée, la portée peut atteindre plusieurs kilomètres.

A la différence du réseau de radio téléphone, la fréquence utilisée par le WiFi est libre, c'est-à-dire qu'aucune licence d'exploitation n'est nécessaire à son utilisation. Néanmoins la puissance d'émission est limitée dans certains pays. En France, la puissance d'émission du WiFi est limitée à 100 mW. Le WiFi peut fonctionner en deux modes : *infrastructure* ou *mode ad hoc*.

1.1.4.1 Mode infrastructure

En mode infrastructure, un nœud mobile communique uniquement avec un nœud fixe, appelé *point d'accès*. Le point d'accès peut être relié à un réseau fixe et donc à Internet, mais il peut jouer aussi le rôle de concentrateur, permettant à tous les nœuds mobiles connectés à ce point d'accès de communiquer entre-eux. L'avantage de ce mode est de pouvoir contrôler l'accès au réseau, car les nœuds mobiles sont obligés de passer par le point d'accès. En revanche, le seul moyen d'étendre le réseau est d'ajouter d'autres points d'accès.

Au niveau de la couche de liaison de données, le mécanisme centralisé d'accès au médium utilisé est PCF pour *Polling Coordination Function*. Ce mécanisme garantit à chaque station un accès minimum au médium ; cette garantie est assurée par le point d'accès qui distribue les temps d'accès à chaque nœud.

Dans les versions de WiFi utilisés actuellement (802.11b, g et n), il n'existe pas de mécanisme gérant la mobilité. Néanmoins des solutions propriétaires existent. La mobilité peut être également gérée au niveau application par le système en choisissant un point d'accès accessible en cas de coupure avec le point d'accès courant mais l'association avec un nouveau point d'accès peut durer plusieurs secondes. Pour palier à ce problème, un amendement récent (IEEE 802.11r-2008) permet la mobilité entre points d'accès (IEEE 802.11r-2008) avec

handover sans coupure. Cette fonctionnalité est essentiellement utilisée pour les communications VoIP imposant une coupure de communication inférieure à 50 ms.

1.1.4.2 Mode ad hoc

En mode ad hoc, deux nœuds peuvent communiquer entre eux directement sans passer par un point d'accès. Ce mode permet de connecter plusieurs ordinateurs rapidement sans la mise en place de points d'accès. Il permet aussi de créer un réseau autonome, à l'aide de protocoles de routage dits ad hoc, permettant à un nœud de communiquer avec un autre nœud qui n'est pas dans son voisinage. Le protocole de routage permet de découvrir des chemins dans un réseau, dont la topologie est inconnue ; la complexité du routage est accentuée si les stations sont mobiles. Nous décrirons en détail les réseaux ad hoc dans la section 2.1.

Au niveau de la couche de liaison de données, le mécanisme centralisé d'accès au médium utilisé est DCF, pour *Distributed Coordination Function*. Cet algorithme est basé sur CSMA/CA. Ce mode consiste à écouter le médium avant de transmettre pour éviter les collisions. Lorsque le médium est libre l'émission est possible, mais le nœud attend pendant un délai aléatoire avant d'émettre. Ce mécanisme permet de désynchroniser les émissions des différentes stations afin de limiter les collisions.

Mais ce mécanisme n'est pas suffisant dans le cas des stations cachées. En effet comme le montre la Figure 1.2, la station B peut atteindre A et C, mais A (ou C) ne reçoit pas le signal de C (ou A). Donc si A envoie des trames vers B, C ne peut pas les entendre, et peut émettre vers B, les trames émises par C rentrant alors en collision avec celles de A. Afin d'éviter ce problème, deux trames spécifiques sont ajoutées : RTS, pour *Ready To Send* et CTS pour *Clear To Send*. Avant d'émettre les trames de données, une station émet une trame RTS, par exemple A vers B ; B répond alors par une trame CTS indiquant qu'elle accepte l'échange de données. Dans ces deux trames, la durée de l'échange est indiquée, donc toute station recevant une trame RTS ou CTS dont elle n'est pas destinataire, comme C, ne doit pas émettre pendant la durée indiquée dans la trame. Ceci permet de limiter le risque de collision en plus de l'algorithme CSMA/CA.

1.1.5 Problème de passage à l'échelle

D'après [111], la passage à l'échelle est défini comme *la capacité d'un réseau à supporter l'augmentation de ses paramètres limitants*. Le paramètre limitant du réseau pouvant être sa taille, sa mobilité, le débit du trafic de données, etc.

Dans un réseau sans fil la taille du réseau (en nombre de nœuds) augmente, soit en fonction de la densité ou soit en fonction de l'étendue du réseau. La *densité* du réseau est définie comme étant le nombre de nœuds par unité géographique, soit le nombre de nœuds par mètre ou par mètre carré.

Le problème de passage à l'échelle est un problème fondamental des réseaux sans fil comme filaire. La hiérarchisation des réseaux, c'est-à-dire créer un ensemble de sous-réseaux reliés entre-eux, permet le passage à l'échelle des réseaux étendus comme Internet. Un réseau

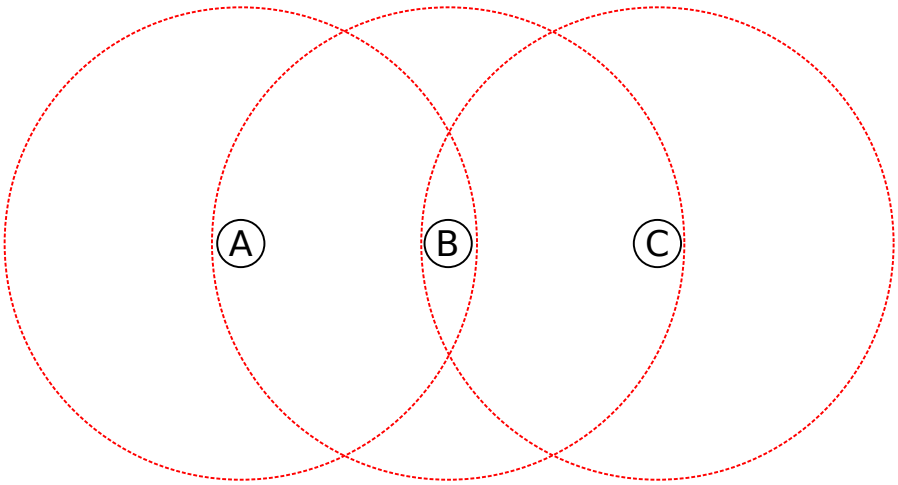


Figure 1.2 – Problème des stations cachées

cellulaire est également organisé comme un réseau hiérarchique permettant de couvrir un large territoire tout en conservant une bonne qualité de service. Nos travaux s'intéressent au problème de passage à l'échelle, dans un contexte fortement dynamique, comme le sont les réseaux de véhicules.

Avant d'introduire les problèmes de forte dynamacité des réseaux de véhicules, nous nous intéressons à la mobilité dans les réseaux.

1.2 Mobilité dans les réseaux IP

Nous l'avons vu, la mobilité dans et entre les réseaux est associée à divers mécanismes liés à la gestion des coupures mais aussi à l'authentification, l'identification et l'échange d'informations entre réseaux et points d'accès. Dans un réseau Mobile IP, la mobilité peut être gérée dans différentes couches. Comme, nous l'avons vu, WiFi 802.11r (couche accès au réseau) gère le handover. La couche application peut également gérer la mobilité. Par exemple dans une bibliothèque, une conférence, un hôtel, chez un client, des amis, on se connecte au réseau local disponible. Le système se configure automatiquement grâce au protocole d'auto-configuration DHCP. Une fois la connectivité à Internet établie, un client d'un système de téléphonie IP sur Internet (comme Skype ou systèmes basés sur SIP), peut être joint depuis n'importe quel correspondant sur Internet. Mais cette possibilité passe par un enregistrement du client à un serveur central accessible sur Internet par une adresse IP connue et publique. La localisation est donc effectuée au niveau de la couche application. Ce scénario est également identique pour les systèmes de messagerie instantanée. C'est actuellement ce type de mobilité qui est utilisé dans la quasi totalité des situations d'itinérance entre et dans les réseaux IP.

Mobile IP est une solution pour gérer la mobilité sur Internet, mais au niveau de la couche réseau (IP). Contrairement à la mobilité au niveau application, la mobilité IP, proposée par

Mobile IP, permet à leurs utilisateurs de rester connecté en permanence à Internet. Ceci implique que tout correspondant sur Internet puisse les joindre de manière permanente par l'intermédiaire d'une adresse IP fixe et qu'il puisse joindre leur correspondant sur Internet. Ces deux conditions ne sont pas réunies lorsqu'un nœud mobile se configure avec une adresse IP privée non accessible depuis Internet. Mobile IP est une solution de roaming, mais, nous le verrons plus tard, elle peut également être adaptée pour la gestion des handovers.

Dans un premier temps, nous allons décrire Mobile IP dans sa version initiale développée pour IPv4, puis ensuite nous décrirons les améliorations apportées à Mobile IP avec IPv6.

1.2.1 Mobile IP

Deux versions de Mobile IP co-existent, une pour les réseaux IP en version 4 et une autre pour les réseaux IP version 6.

1.2.1.1 Mobile IPv4

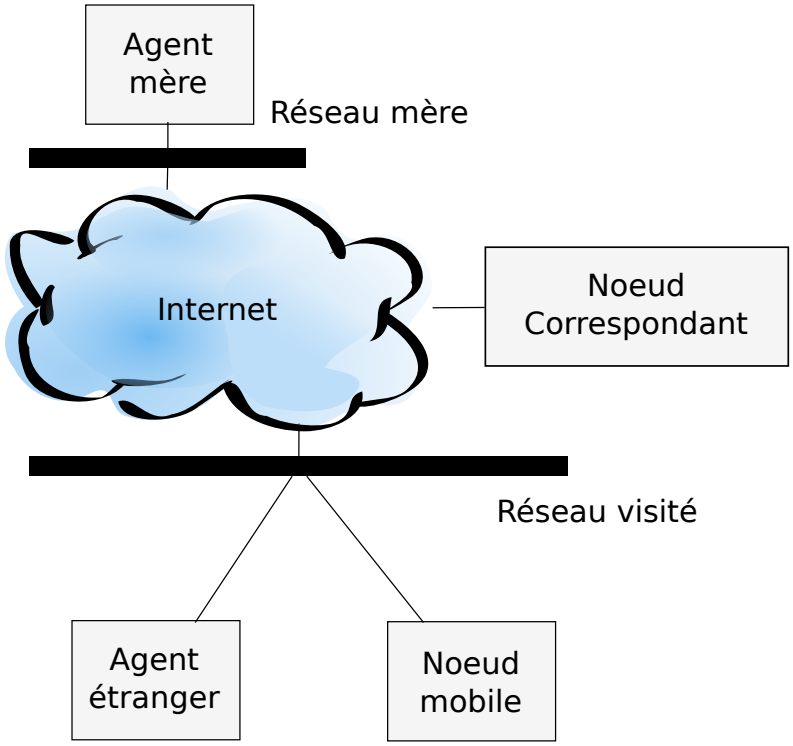


Figure 1.3 – Architecture de Mobile IPv4

Mobile IPv4 [100] définit trois entités :

Le nœud mobile, ou Mobile Node (MN), change de réseau au cours de son déplacement,

L'agent mère, ou Home Agent (HA), est situé dans le *réseau mère*. Sa tâche est de maintenir la localisation courante du nœud mobile, et de servir d'intermédiaire entre le nœud correspondant et l'agent étranger.

L'agent étranger, ou Foreign Agent (FA), est situé dans le *réseau visité* par le nœud mobile. Il sert d'intermédiaire entre le nœud mobile et l'agent mère.

Dans Mobile IP, chaque nœud mobile est associé avec une *adresse IP mère* connue par son réseau mère. Cette adresse est statique : elle ne change pas quand le nœud se déplace d'un réseau à un autre. Quand le nœud est dans son réseau mère, il utilise le routage IP classique. Lorsque le nœud mobile change de réseau et se connecte à un réseau visité, il contacte un agent étranger (Figure 1.3).

Lorsqu'un correspondant souhaite communiquer avec le nœud mobile, il envoie les paquets vers l'adresse IP qu'il connaît, l'adresse mère, à l'aide d'un routage classique. L'agent mère, sachant que le nœud mobile n'est pas connecté à son réseau, encapsule le paquet dans un autre paquet IP pour l'envoyer à l'agent étranger. Cette encapsulation, appelée *tunneling*, permet de transmettre des paquets dans un réseau visité sans modifier le paquet IP original. On dit qu'on crée un tunnel entre les deux nœuds. Le paquet encapsulé arrive au bout du tunnel à l'agent étranger qui le désencapsule et le transmet au nœud mobile. Le nœud mobile reçoit le paquet comme s'il était reçu dans son réseau mère, c'est-à-dire l'adresse IP de la destination est l'adresse IP mère.

Lorsque le nœud mobile souhaite répondre au correspondant, il peut envoyer le paquet directement en utilisant le routage classique. Nous remarquons que dans un contexte de mobilité, le sens Correspondant → Noeud Mobile (communication *down*) est toujours plus compliqué que le sens Nœud Mobile → Correspondant (comunication *up*).

Deux processus restent à décrire : la découverte d'un agent étranger et l'enregistrement. Les agents étrangers s'annoncent dans le réseau en envoyant un *Agent Advertisement* ; étant en fait une extension ajoutée au *ICMP Router Advertisement*. Un nœud mobile peut également solliciter un agent étranger à l'aide d'un *Agent Solicitation*. Nous remarquons que la première méthode est proactive, alors que la seconde est réactive.

Lorsque le nœud mobile reçoit un *Agent Advertisement*, il détecte s'il a changé de réseau, et s'il est dans son réseau mère ou dans un réseau visité. De plus, il obtient l'adresse de l'agent étranger.

Quand le nœud mobile change de réseau, il initie une procédure d'enregistrement pour garder son agent mère informé de sa position courante. Pour cela il envoie une *requête d'enregistrement* contenant l'adresse de l'agent étranger obtenu par l'*Agent Advertisement*, soit à son agent mère, soit directement ou soit par l'intermédiaire de l'agent étranger. L'agent mère confirme l'enregistrement en répondant au nœud mobile. Une fois l'enregistrement terminé avec succès, l'agent mère peut intercepter les paquets à destination du nœud mobile, puis l'envoyer à l'agent étranger par l'intermédiaire du tunnel, comme expliqué précédemment.

1.2.1.2 Mobile IPV6

La mobilité sous IPV6, appelé Mobile IPV6 [68] , est optimisée en profitant des nouvelles fonctionnalités d'IPV6. En effet, le problème de Mobile IPV4 est le routage triangulaire (Fig-

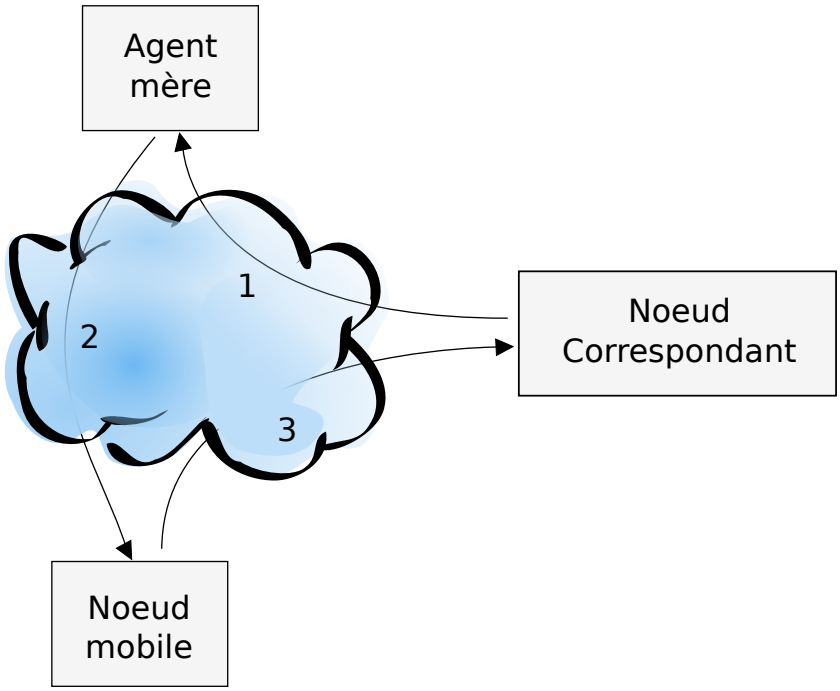


Figure 1.4 – Route non optimale dans IPV4

ure 1.4 et Figure 1.6). Nous voyons que les paquets doivent transiter par l'intermédiaire d'un *proxy*, qui est dans notre cas l'agent mère. Ce déroutement augmente la longueur du chemin entre le correspondant et le nœud mobile, augmentant le délai d'acheminement des paquets. Pour réduire la longueur du chemin, Mobile IPV6 introduit la notion d'*association*. Une association permet de relier une *adresse* IP *temporaire*, celle donnée par le réseau visité, et l'adresse IP mère

IPV6 introduit *les extensions* qui sont des données ajoutées à la fin de l'en-tête d'IPV6. Parmi ces extensions, l'extension *destination* est conçue principalement pour la mobilité et comporte quatre options :

Mise à jour de l'association Cette option est utilisée par un nœud mobile pour avertir, soit un correspondant, soit un agent mère de son association courante. Un message contenant cette option est envoyé par le nœud mobile à son agent mère et s'appelle *l'enregistrement principal* qui remplace la *requête d'enregistrement* de Mobile IPV4.

Acquittement de l'association Cette option est utilisée pour acquitter la réception d'un paquet contenant l'option de mise à jour de l'association.

Demande de mise à jour de l'association Cette option est utilisée par un correspondant pour demander à un nœud mobile de lui envoyer un message *Mise à jour de l'association*. Cette option est utilisée par un correspondant pour rafraîchir son association avec le nœud mobile.

Adresse principale Cette option est utilisée par un nœud mobile pour informer le corres-

pondant de l'adresse mère de celui-ci. Les correspondants sont alors capables de substituer l'adresse temporaire par l'adresse mère du nœud mobile.

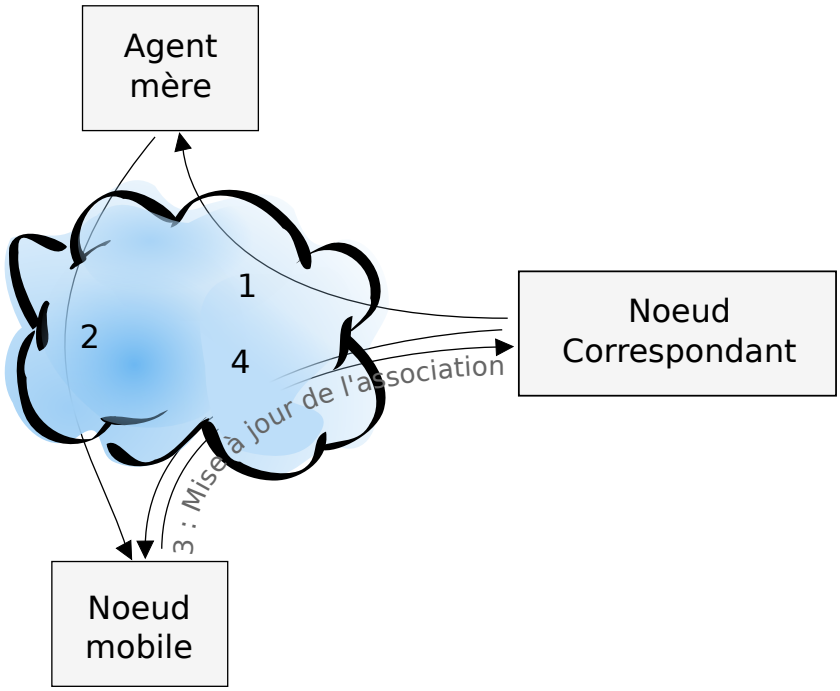


Figure 1.5 – Transmisions des données dans IPV6

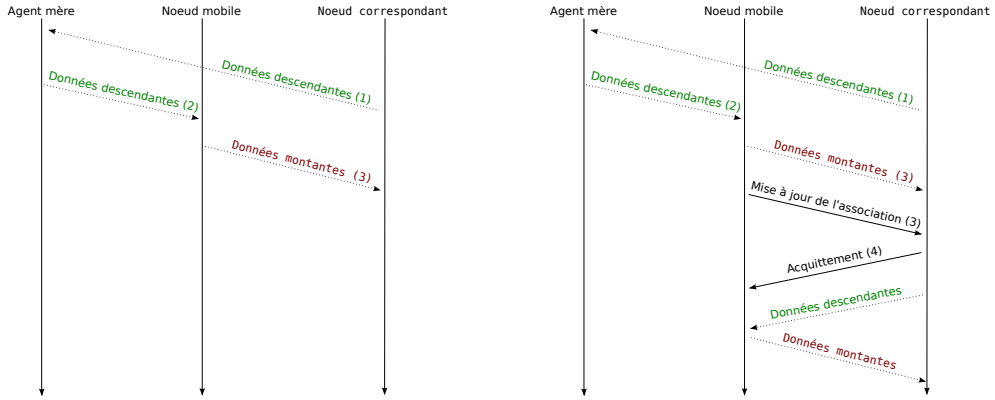
En plus des extensions, chaque nœud possède une table d'association permettant de stocker l'ensemble des associations en cours. Décrivons en détail le fonctionnement de Mobile IPV6. Lorsque l'association est établie, le correspondant peut alors envoyer les paquets directement au nœud mobile avec un routage classique. Les paquets ne passent plus par l'agent mère, ni l'agent étranger qui n'existe plus dans Mobile IPV6. Néanmoins, pour que l'association soit effective, les paquets sont d'abord envoyés à l'agent mère (message 1 sur la Figure 1.5) et sont transmis au nœud mobile (message 2), avec le même processus décrit dans Mobile IPV4.

Lorsque le nœud mobile reçoit un paquet encapsulé, il envoie un message *Mise à jour de l'association* au correspondant par le routage classique (message 3), pour qu'il puisse mettre à jour son association et donc lui envoyer directement les paquets sans passer par l'agent mère (message 4).

Lorsque le nœud mobile se déplace, il envoie à la fois à son agent mère et au correspondant, dont l'association reste valide, un message de *Mise à jour de l'association* pour les informer du changement d'adresse temporaire.

Nous remarquons que l'architecture IPV6 ne contient pas d'agent étranger. La détection de mouvement, ou de changement de réseau, peut se faire sans l'agent étranger à l'aide de n'importe quel mécanisme qu'offre IPV6 (découverte de routeur par exemple) ou par des

protocoles de couches inférieures.



(a) Routage triangulaire de Mobile IPv4 (b) Route optimale de Mobile IPv6

Figure 1.6 – Mobile IPv6 optimise la route par rapport à Mobile IPv4

Nous avons vu que Mobile IP, dans ses deux déclinaisons pour IPv4 et IPv6, permet de gérer la mobilité. Néanmoins, deux problèmes sont soulevés : le delai de transmission de l'échange de messages avec un réseau mère éloigné peut être important, entrainant des coupures de communication lors du changement de réseau, d'autant plus gênantes si le déplacement est rapide. Nous traiterons ce problème avec l'introduction des protocoles de micro-mobilité.

Le second problème est le déplacement groupé de nœuds mobiles. Concrètement, admettons que les passagers d'un train utilisent tous Mobile IPv6 pour gérer leur mobilité. Les nœuds mobiles se connectent au réseau en utilisant des points d'accès sans fil disposés à l'extérieur du train sur le terrain. Lorsque le train s'éloigne trop d'un point d'accès, les nœuds mobiles doivent changer de point d'accès et donc de réseau. L'opération de mise à jour de l'association se fait alors quasiment simultanément par tous les nœuds mobiles du train. Le nombre de messages de mise à jour est fonction du nombre de nœuds mobiles dans le train, alors qu'une seule entité (le train) se déplace. Ce problème se ramène au problème de mobilité de réseau. En effet les nœuds du train constituent un réseau. Il est alors plus judicieux de considérer le réseau comme mobile, au lieu de considérer tous les nœuds mobiles. Nous allons détailler le protocole de mobilité de réseau NEMO (NEtwork MObility).

1.2.2 Mobilité de réseau : NEMO

Le standard NEMO [45], pour NEtwork MObility, de l'IETF est dérivé de Mobile IP pour gérer la mobilité des réseaux IP. Un sous-réseau comporte des nœuds mobiles avec le même préfixe d'adresse IP (MNP). Ce sous-réseau est associé à un réseau mère et peut changer de réseau en changeant de point d'ancrage, c'est-à-dire de point d'accès. Un des objectifs est de ne pas imposer des modifications aux nœuds mobiles.

Deux nouvelles entités sont introduites (Figure 1.7) : le nœud du réseau mobile (MNN)

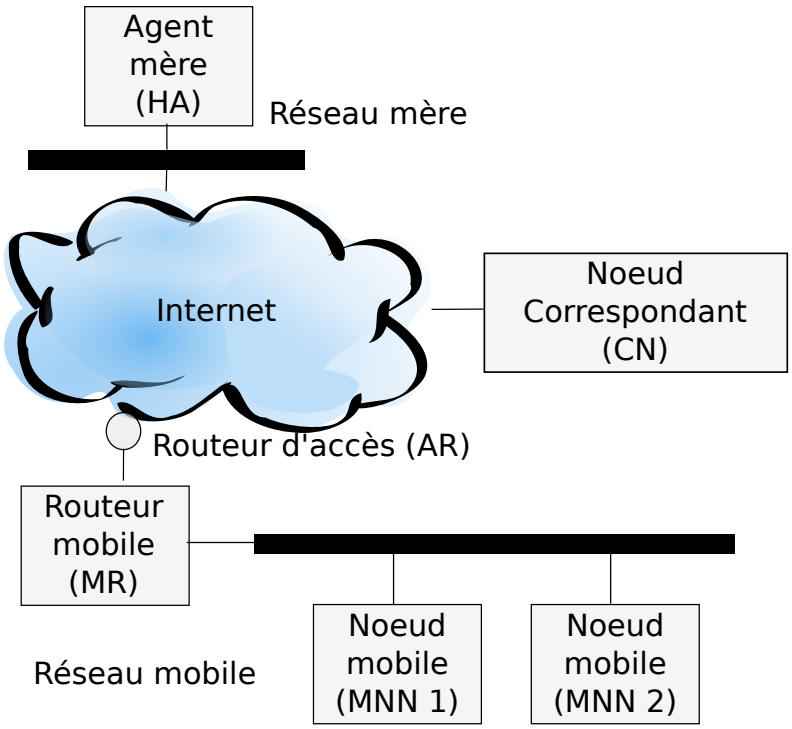


Figure 1.7 – Architecture de NEMO

et le routeur mobile (MR). Le MNN est un nœud mobile qui fait partie du réseau mobile. Le routeur mobile est l'entité la plus importante de NEMO. Le changement de point d'accès ne provoque pas de changement d'adresse IP du MNN. La gestion de la mobilité est déléguée au MR. Le MR, comme le nœud mobile de Mobile IP, possède deux adresses. Le première, l'adresse mère, est permanente et identifie le MR dans le réseau mère. Son préfixe est le même que les MNN. Du fait qu'elle ne change pas, tout correspondant sur Internet peut atteindre le MR. La seconde adresse, l'adresse temporaire, est obtenue dans le réseau visité, où se trouve le point d'accès.

Le protocole NEMO établit ainsi une association entre le préfixe MNP du sous-réseau mobile et l'adresse temporaire. Lorsque le réseau change de point d'ancrage (Figure 1.8), le MR envoie son adresse temporaire à l'agent mère (HA) du réseau mère. L'agent mère actualise l'association entre le préfixe du sous-réseau et l'adresse temporaire. Un tunnel est ensuite établi entre le MR et le HA pour transmettre les paquets provenant d'Internet vers le MR. Le HA encapsule le paquet pour le transmettre au MR, puis le MR désencapsule le paquet. Le MR utilise ensuite le protocole de routage du sous-réseau pour transmettre les paquets vers le MNN.

Pour le correspondant, la mobilité est alors transparente, puisqu'il envoie ses données à un routeur du réseau mère. Les paquets à destination d'un correspondant sur Internet sont envoyés au MR, avec le protocole de routage utilisé par le sous-réseau mobile, puis encapsulés pour être envoyés au HA. Le HA désencapsule le paquet, puis transmet le paquet au

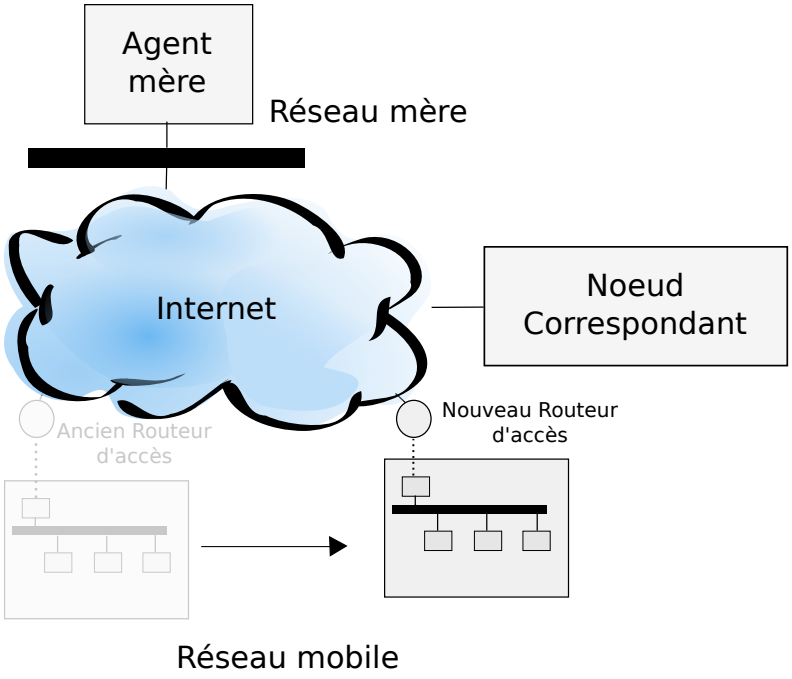


Figure 1.8 – Changement de routeur d'accès dans NEMO

correspondant. Tout le trafic en provenance et à destination du sous-réseau mobile passe alors par le HA, ce qui rend le routage sous optimal. Nous avons vu dans IPV4 le phénomène de routage triangulaire résolu dans IPV6. Ici, le problème du routage sous optimal est similaire, mais dans les deux sens de communication : nœud mobile vers nœud correspondant et vice versa. Les travaux en cours tentent, entre autres, de résoudre ce problème.

1.2.3 Protocoles de micro-mobilité

Dans un contexte d'utilisation de Mobile IP avec des nœuds mobiles changeant de point d'accès fréquemment, l'overhead, le délai et les paquets augmentent significativement. Ce phénomène est dû à la mise en place d'un tunnel entre l'agent mère et l'agent étranger à chaque changement de point d'accès (handoff). Ce délai est intrinsèque à l'envoi de la requête d'enregistrement envoyé par l'agent étranger a l'agent mère, et de l'attente de la confirmation d'enregistrement.

Le but des protocoles de micro-mobilité est de réduire le délai des handoffs avec un minimum de paquets perdus tout en minimisant les messages de signalisation à travers l'utilisation de techniques de localisation afin de gérer localement la mobilité sans interaction avec des entités de Mobile IP situées sur Internet (l'agent mère par exemple). Cette fonctionnalité est indispensable à Internet pour supporter un grand nombre de terminaux mobiles.

1.2.3.1 Protocoles de micro-mobilités basés sur le routage

L'équipe du COMET Group du Columbia University à New-York, USA, a introduit Cellular IP avec Ericson et Nokia et a recensé l'ensemble des protocoles de micro-mobilité [34]. Pour simplifier, nous retenons de ce recensement deux grandes catégories de protocoles de micromobilité : basée sur le routage et basée sur des tunnels.

Cette méthode utilise le routage pour transmettre les paquets au point d'accès où est attaché le nœud mobile. L'architecture est composée d'une *passerelle* (*gateway* ou GW), de *routeurs*, et de *points d'accès* (AP). La gateway fait la jonction avec Mobile IP et elle est l'extrémité du tunnel avec le HA. Un routeur transmet les paquets, soit à un autre routeur, soit à la gateway ou à un point d'accès. L'AP permet à un nœud mobile d'accéder au réseau d'accès contenant les routeurs. Dans Cellular IP, le routeur peut également être le point d'accès. Dans la suite, nous allons détailler deux protocoles de micro-mobilité basés sur le routage : Cellular IP et HAWAII.

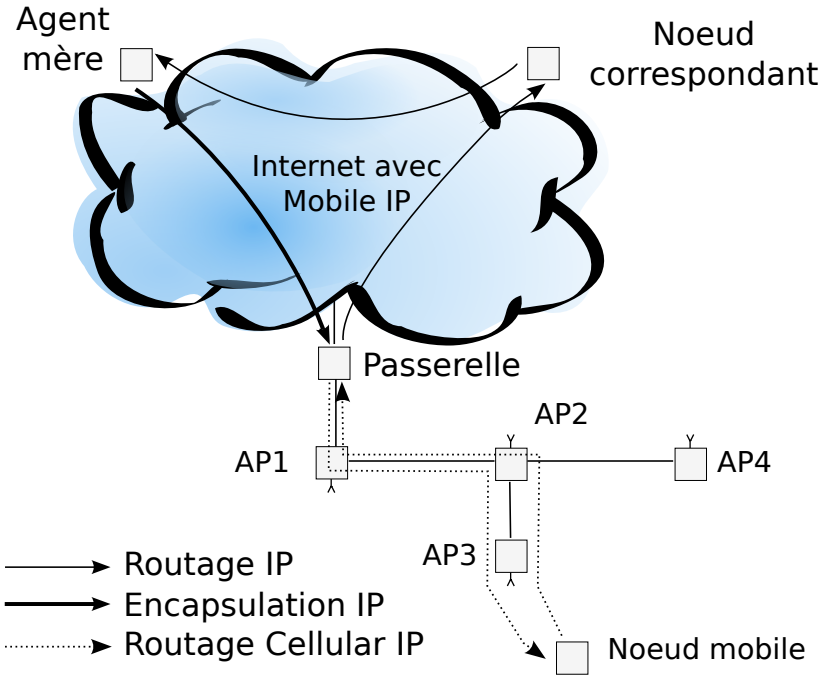


Figure 1.9 – Routage dans cellular IP

La proposition Cellular IP (CIP) [35, 33] supporte le handoff rapide et les techniques de paging. Pour limiter les messages de signalisation, les paquets de données transmis par le nœud mobile vers Internet sont utilisés pour actualiser les informations de localisation du nœud mobile. CIP utilise les paquets provenant du nœud mobile pour maintenir le chemin de la gateway au point d'accès (Figure 1.9). Néanmoins, le nœud mobile peut aussi envoyer un message *route-update packet* au point d'accès pour actualiser explicitement sa position.

Cellular IP supporte deux types de handoff : *hard handoff* et *semisoft*. Cellular IP hard handoff est basé sur le compromis entre diminuer le nombre de paquets perdus pendant le

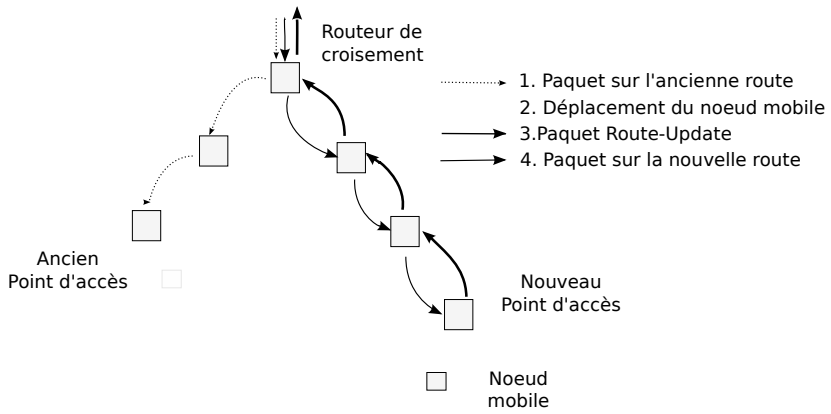


Figure 1.10 – Handoff dans cellular IP

handoff et diminuer les messages de signalisation pendant le handover.

Pour initier un handoff, le nœud mobile s'associe avec le nouveau point d'accès (au niveau de la couche 2) et envoie ses paquets de données ou un message *route-update packet* pour actualiser les tables de routage des routeurs. La durée du handoff est alors égale au délai d'acheminement entre l'initiation du handoff et le moment où le paquet transmis au nouveau point d'accès est reçu par un *routeur de croisement*, voir Figure 1.10. Un routeur de croisement est un routeur à la fois sur le chemin vers le nouveau point d'accès et l'ancien point d'accès. Durant cet intervalle, les paquets à destination du nœud mobile sont perdus. Cependant, il existe différentes méthodes pour éviter la perte de paquets.

Une des approches est de créer une interaction entre l'ancien et le nouveau point d'accès. Dans ce cas, le nouveau point d'accès prévient l'ancien point d'accès du handover imminent. Les paquets arrivant à l'ancien point d'accès sont alors transmis au nouveau point d'accès, puis transmis au nœud mobile. Cellular IP *semisoft handoff* exploite la possibilité à un nœud mobile de recevoir des paquets provenant à la fois de l'ancien point d'accès et du nouveau point d'accès. Avant qu'un nœud mobile change définitivement de point d'accès, le nœud mobile envoie un *semisoft packet* au nouveau point d'accès et revient à écouter l'ancien point d'accès immédiatement. Le but du paquet *semisoft* est d'établir une nouvelle route entre le routeur de croisement et le nouveau point d'accès. Durant la phase d'établissement de cette nouvelle route, le nœud mobile reste connecté à l'ancien point d'accès. Après un délai prédéfini, appelé *semisoft delay*, le nœud mobile initie un handoff classique. Ce délai doit être proportionnel au délai d'acheminement des paquets entre le nœud de croisement et le nœud mobile.

Cellular IP supporte le paging. Un nœud mobile inactif, c'est-à-dire n'envoyant pas de paquet de données, envoie à intervalles réguliers un message *paging-update* pour qu'il reste atteignable. Un nœud mobile envoie ce message pendant son déplacement au point d'accès avec le meilleur signal.

Le protocole HAWAII [104] de Lucent Technology propose un protocole de routage spécifique pour gérer la mobilité. Les nœuds du réseau dans HAWAII utilisent un protocole de routage IP classique. Les informations de localisation sont créées, actualisées et supprimées

par un message de signalisation spécifique, appelé *path setup*.

HAWAII définit quatre procédés d'actualisation des chemins vers le nœud mobile pour gérer le handoff lorsque le nœud mobile change de point d'accès. Le choix de la méthode dépend de la qualité de service demandée : éliminer les paquets perdus, minimiser la durée des handoffs ou maintenir l'ordre d'arrivée des paquets. Les quatre procédés sont répartis en deux catégories : *forwarding* ou *non-forwarding*.

Dans les procédés forwarding, les paquets sont d'abord transmis de l'ancien point d'accès au nouveau point d'accès avant d'être dérouté par le routeur de croisement une fois sa table de routage mis a jour.

Dans les procédés non-forwarding, les données sont reroutées par le routeur de croisement vers le nouveau point d'accès sans transmission des paquets par l'ancien point d'accès. La catégorie *non-forwarding* se décline en deux procédés Unicast Non-Forwarding (UNF) et Multicast Non-Forwarding (MNF). Le procédé UNF est adapté pour les réseaux où le nœud mobile est capable d'écouter et transmettre des paquets à deux points d'accès différents simultanément.

1.2.3.2 Protocoles de micro-mobilité basés sur des tunnels

L'avantage de l'utilisation d'un tunnel est de pouvoir déployer un protocole de micro-mobilité sur un réseau déjà existant sans appliquer de modifications aux routeurs de ce réseau. C'est pour cela qu'on dit que le protocole fonctionne au niveau de la couche OSI "3.5", contrairement aux protocoles basés sur le routage qui fonctionnent à la couche 3. Le désavantage vient de l'overhead généré par l'encapsulation de paquets nécessaires au fonctionnement des tunnels. Ici nous allons décrire les deux protocoles Hierarchical Mobile IP (HMIP) et Fast-handover Mobile IP (FMIP).

L'approche tunnel est privilégiée par l'IETF. En effet seul HMIP et FMIP sont des RFC, alors que Cellular IP et HAWAII sont restés à l'état de draft, expiré à ce jour. HMIP et FMIP ont fait chacun l'objet de deux RFC : un pour IPV4 et un pour IPV6, issus respectivement du groupe de travail *mip4* [7] et *mipshop* [6].

La version pour IPV4 de HMIP est connu sous le nom de Mobile IPV4 Regional Registration [52]. Nous décrirons ici la version pour IPV6, Mobile IPV6 Fast Handovers (HMIPV6) [116].

Dans HMIPV6 un nouveau type de nœud est ajouté à Mobile IP, appelé Mobility Anchor Point (MAP). Un MAP est essentiellement un HA local situé dans un réseau visité et pouvant être situé à n'importe quelle hiérarchie de routeur dans le réseau. MAP constitue un niveau de hiérarchie supplémentaire à Mobile IP. Idéalement, chaque domaine contient un MAP. Le MAP associé est dit alors MAP du domaine. La Figure 1.11 montre l'architecture de HMIPV6 avec le MAP. Un routeur d'accès (AR) est le routeur par défaut du nœud mobile. Dans un réseau sans fil, le AR est typiquement le point d'accès.

Deux types d'adresse temporaire sont délivrés pour le MN : Regional Care-Of-Adresse (RCOA) et On-link Care-of-Adresse (LCOA). La RCOA est l'adresse utilisée par le MAP pour recevoir les messages du HA. La LCOA est l'adresse utilisée par le MN pour communiquer avec le MAP. Lorsqu'un nœud change de routeur d'accès, il envoie un Binding Update au MAP pour changer l'association entre le MAP et le routeur d'accès.

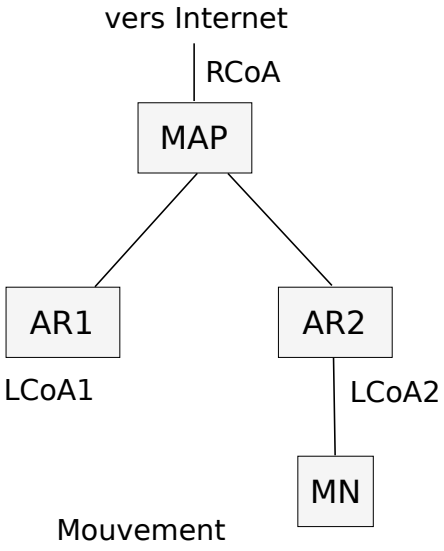


Figure 1.11 – Mobile IPv6 Fast Handovers architecture : MAP est ajouté l'architecture de Mobile IP

Le Binding Update (BU) correspond à la requête d'enregistrement de Mobile IP, mais le BU est envoyé en local au MAP au lieu d'être envoyé au HA. L'envoi à un nœud local d'un seul BU (au lieu de l'envoyer à l'agent mère et à chaque nœud correspondant) pour chaque changement de routeur d'accès permet de réduire le délai d'un handover, ce qui est le but des protocoles de micro-mobilité.

Lorsqu'un MN change de routeur d'accès dans un même domaine il reste associé avec le même MAP. L'association entre l'adresse du HA et la RCOA ne change donc pas : la mobilité est transparente pour le HA et le CN. Par contre, quand le MN change de domaine, une nouvelle association entre le HA et la nouvelle RCOA (délivrée par le nouveau MAP). Pour cela le processus d'enregistrement classique de Mobile IP est utilisé. Deux tunnels sont utilisés : un pour communiquer entre le MN et le MAP, et un autre entre le HA et le MAP ou entre le CN et le MAP.

FMIP est l'autre protocole basé sur les tunnels définis par l'IETF. Comme pour HMIP, FMIP est décliné en deux versions : Mobile IPv4 Fast Handovers [75] et Mobile IPv6 Fast Handovers [74]. FMIP est complémentaire à HMIP, son but n'est pas de réduire le délai d'un handover (comme dans HMIP) mais d'accompagner la mobilité pour assurer la continuité de la communication lors d'un changement de routeur d'accès en évitant la perte de paquets. On note respectivement PAR et NAR, l'ancien routeur d'accès et le nouveau routeur d'accès. FMIP comporte les messages suivants :

Binding Update (BU) met à jour l'association entre le HA et le NAR.

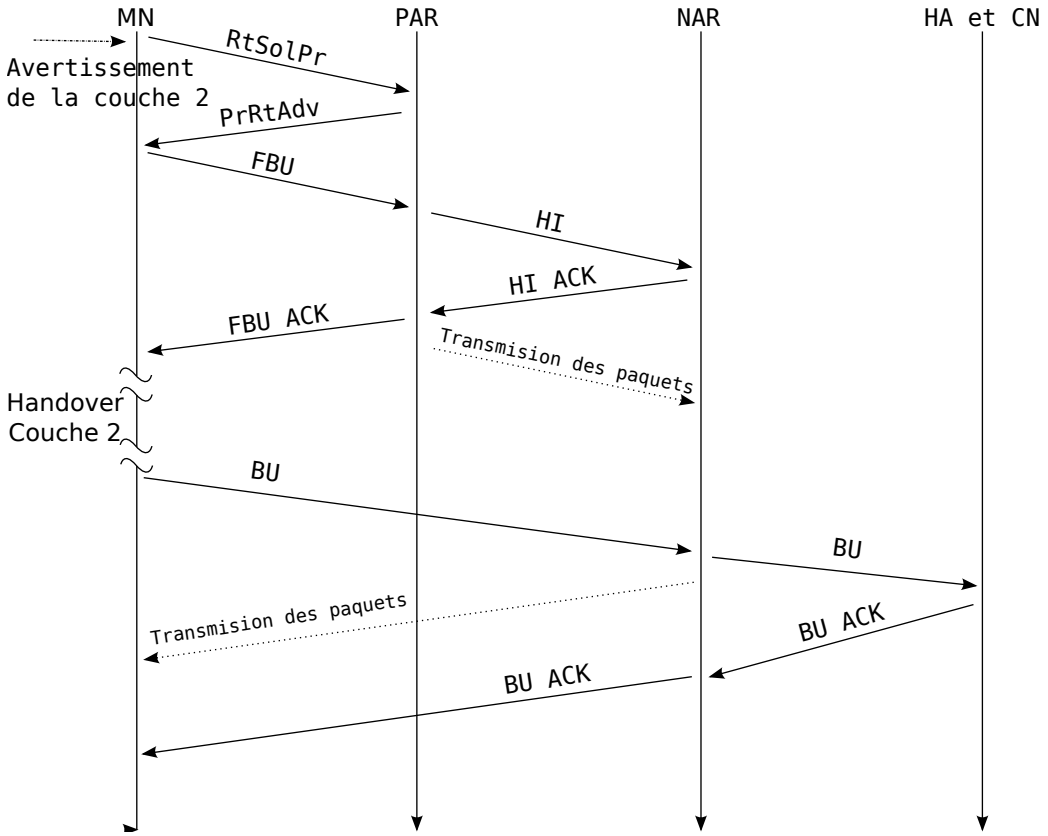


Figure 1.12 – Fast Handover prédictif initié par le noeud mobile

Proxy Router Advertisement (PrRtAdv) Envoyé par le PAR au MN, pour l'informer d'un nouveau routeur d'accès disponible dans le voisinage. Un nœud mobile peut solliciter le PAR pour lui demander d'envoyer un PrRtAdv, avec un message Router Solicitation for Proxy Advertisement (RtSolPr).

Handover Initiate (HI) Envoyé par le PAR au NAR pour initier le handover. Le NAR acquitte la réception de ce message par un Handover Acknowledge (HAck).

Fast Binding Update (FBU) Permet d'avertir le PAR de l'initiation d'un handover avec un NAR.

La décision d'initier un handover peut être prise, soit par le nœud mobile, à partir d'informations de la couche 2, soit par le réseau. Si le handover est initié par le nœud mobile, alors il envoie un RtSolPr, pour indiquer au PAR qu'il souhaite changer de routeur d'accès. Le PAR lui répond donc avec un PrRtAdv contenant les informations du nouveau routeur d'accès. Si la décision d'initier un handover est prise par le réseau, alors le PAR envoie au nœud mobile un PrRtAdv, lui indiquant qu'il faut initier un handover.

A partir du moment où un handover est initié deux scénarios sont possibles : Predictive Fast Handover ou Reactive Fast Handover. Le Predictive Fast Handover est décrit en Figure 1.12. Il est utilisé quand le nœud mobile est capable d'envoyer un FBU au PAR. Donc, le FBU est

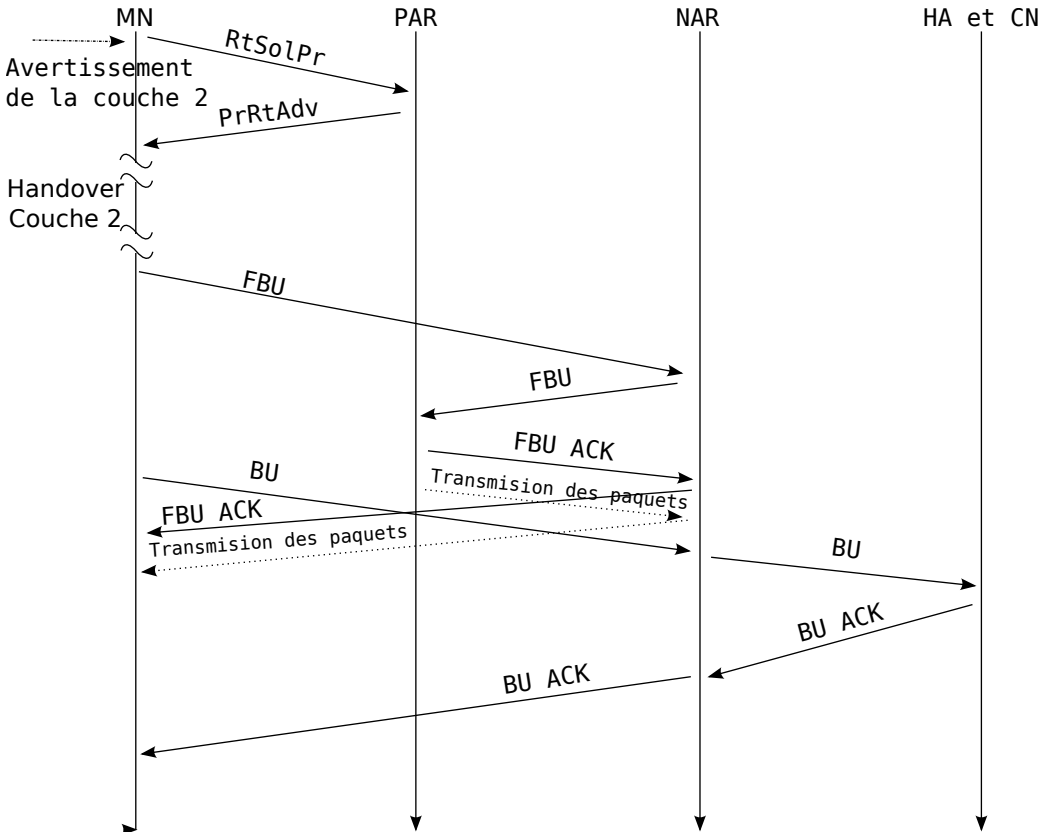


Figure 1.13 – Fast Handover réactif initié par le noeud mobile

envoyé au PAR qui envoie un HI au NAR. Une fois que le PAR a reçu l'acquittement du HA, il transmet alors tous les paquets de données à destination du MN au NAR par l'intermédiaire d'un tunnel. Le NAR stocke ces paquets. Une fois que le MN est associé avec le NAR, il lui envoie un BU, qui le transmet au HA. Le BU sert à l'actualisation de l'association entre le HA et le NAR, il est équivalent à la requête d'enregistrement de Mobile IP. Une fois que le NAR a reçu le BU, il délivre les paquets stockés pendant le handover de niveau 2 au MN (durée pendant lequel le MN se désassocie du PAR, s'associe au NAR, et acquiert une nouvelle adresse auprès du NAR). Le scénario Reactive Fast Handover est différent, comme cela est montré en Figure 1.13. Le MN change de routeur d'accès avant d'envoyer un FBU au PAR. Il envoie alors le FBU au NAR une fois qu'il s'y est associé. Le PAR stocke les messages à destination du MN pendant le handover de niveau 2, en prévision de les transmettre plus tard. Une fois que le NAR a reçu le FBU, il envoie lui même un FBU au PAR. A la réception du FBU par le PAR, il transmet alors tous les paquets stockés dans le NAR, qui va lui même les transmettre au MN. Pendant ce temps, le MN envoie un BU au NAR, qui le transmet au HA pour actualiser l'association.

Nous avons vu que quelque soit le scénario adopté, FMIP permet la continuité de la communication en évitant la perte de paquets. Par contre HMIP permet de réduire le délai d'association avec le nouveau routeur d'accès en ajoutant un niveau hiérarchique supplémentaire à

l'aide d'un MAP. Un protocole appelé S-MIP [65] propose d'associer ces deux méthodes pour à la fois réduire le délai d'un handover et éviter la perte de paquets.

Conclusion

Nous avons vu différentes techniques permettant la mobilité dans les réseaux sans fil, notamment dans les réseaux basés sur IP à l'aide de Mobile IP et ses extensions. Ces techniques sont mises en œuvre dans des réseaux centralisés possédant une infrastructure. En revanche, un réseau décentralisé est mieux adapté au réseau dont la topologie est fortement dynamique. Il permet également un déploiement rapide sans infrastructure pré-existante. Dans le chapitre suivant nous allons présenter les réseaux décentralisés, dit ad hoc, afin d'aborder ensuite les réseaux de véhicules.

Chapitre 2

Les réseaux ad hoc : des principes de base aux réseaux ad hoc de véhicules hybrides

Le précédent chapitre nous a permis de comprendre la terminologie et les principes de base que nous allons utiliser par la suite. Dans ce second chapitre d'état de l'art, nous abordons les réseaux ad hoc. Le réseau ad hoc est un réseau décentralisé et particulierement adapté au réseau fortement dynamique. Après avoir introduit les réseaux ad hoc, nous détaillerons un cas particulier des réseaux ad hoc : les réseaux ad hoc hybrides ; nécessaires à l'interconnexion des réseaux ad hoc mobiles avec une infrastructure fixe existante. Ensuite, nous introduirons le cas d'études des réseaux de véhicules associé à l'ensemble des technologies précédemment citées.

2.1 Réseaux ad hoc mobiles

Un réseau informatique statique est composé de terminaux et routeurs. Un routeur permet de router les paquets, soit vers la destination finale, le terminal, soit vers un autre routeur. A l'opposé, un réseau ad hoc ne contient pas de routeurs dédiés : tout nœud peut être terminal et routeur. Par conséquent deux nœuds d'un réseau ad hoc peuvent communiquer sans la mise en place d'infrastructure comme dans un réseau classique. De plus, un réseau ad hoc est principalement utilisé dans un contexte de mobilité, alors que les protocoles de routage utilisés dans un réseau classique ne sont pas adaptés à un changement de topologie fréquent.

Dans un réseau ad hoc, deux nœuds d'un réseau s'échangent des données sous la forme de *paquets* à l'aide de nœuds appelés *nœuds relais* (ou nœuds intermédiaire). Pour cela, le nœud initiateur de l'échange, appelé nœud *source*, transmet les paquets à ses voisins. Si le nœud n'est pas la *destination* du paquet, un nœud relais transmet un paquet à ses voisins, et les voisins transmettent à leur voisins, jusqu'à ce que le paquet ait atteint la destination. Lorsqu'un nœud reçoit un paquet, un *protocole de routage* prend la décision de, soit le transmettre à un nœud voisin, soit l'envoyer à la couche application, soit l'ignorer. Si le nœud est le destinataire de paquet, le paquet est envoyé à la couche application. S'il est un nœud

relais pour la destination du paquet, alors il transmet le paquet. Sinon, le paquet est ignoré.

Dans un réseau ad hoc mobile, les nœuds se déplacent et donc la topologie change fréquemment. Un protocole adapté doit être utilisé et doit supporter la mobilité des nœuds. Chaque nœud exécute ce protocole pour calculer le prochain nœud relais. Suivant ce protocole, les nœuds échangent des messages de *signalisation* entre eux pour avoir connaissance de la topologie en créant un graphe du réseau ou pour créer un *chemin* vers la destination.

Un réseau ad hoc est utilisé là où il n'y a pas ou il n'y a plus d'infrastructure pré-existante. Nous recensons trois grandes catégories d'applications : champ de bataille, catastrophe naturelle, applications orientées utilisateur.

Sur un champ de bataille l'infrastructure du réseau peut être détruite ou non utilisable [56]. Les réseaux ad hoc permettent d'être indépendant d'une infrastructure existante, et donc d'assurer le fonctionnement du réseau dans des conditions extrêmes. Un réseau ad hoc est également rapidement opérationnel, un réseau peut alors être rapidement déployé.

Lors d'une catastrophe naturelle, comme un tremblement de terre, une inondation ou une éruption volcanique, il est fort probable que l'infrastructure pour communiquer ne soit plus opérationnelle. Un réseau ad hoc est alors utile à l'organisation des secours. Il est également possible d'utiliser une partie de l'infrastructure opérationnelle avec des nœuds mobiles pour renforcer le réseau ad hoc [32].

Il existe des nombreuses applications orientées utilisateur basées sur un réseau ad hoc lors d'un rassemblement ponctuel de personnes : conférence, salon. Un réseau ad hoc peut-être complémentaire d'un réseau déjà existant ponctuellement saturé [72]. Les réseaux citoyens, ou Wireless community network [51], sont également une application directe des réseaux ad hoc. Un réseau citoyen permet de relier en réseau les foyers d'un quartier voire d'une ville à l'aide d'un équipement WiFi standard. L'intérêt est de passer outre les limitations des fournisseurs d'accès, soit en terme de débit, de surveillance du réseau ou d'un éventuel filtrage sur les contenus échangés. Enfin, on peut citer les réseaux de véhicules. Une infrastructure dédiée au réseau de véhicules et couvrant l'ensemble du réseau routier étant inexistant, un protocole ad hoc est utilisé pour la communication inter-véhiculaire. Le réseau de véhicules est notre contexte d'étude. Nous le détaillerons d'avantage dans la section 2.3.

Pour acheminer un paquet du noeud source au noeud destinataire, un réseau ad hoc a besoin d'un protocole de routage. Nous allons décrire les différents protocoles de routage ad hoc existants.

2.1.1 Protocoles de routage des réseaux ad hoc

Le protocole de routage est indispensable pour connaître le chemin permettant d'atteindre la destination. Nous distinguons deux grandes classes de protocoles ad hoc : les protocoles topologiques et les protocoles géographiques. La première classe regroupe les protocoles ad hoc basés sur la topologie du réseau où les nœuds n'ont aucune connaissance de leur position géographique ni de celle des autres nœuds. Par échanges de messages entre voisins, ces protocoles doivent découvrir et maintenir la topologie du réseau pour router les paquets, soit de manière proactive soit de manière réactive.

Dans la deuxième catégorie de protocoles, on trouve les protocoles géographiques pour lesquels chaque nœud doit connaître sa position géographique qui peut être obtenue à l'aide d'un système de géolocalisation tel que le GPS. Aucune connaissance globale du réseau (telle que la topologie) n'est nécessaire au routage des paquets : un nœud transmet un paquet au voisin le plus proche de la destination. De fait, ils sont bien adaptés aux changements fréquents de topologie du réseau. Par contre, la connaissance de la position géographique du nœud destinataire est indispensable et est fournie par un service de localisation associé au protocole ad hoc géographique

Dans la suite, nous allons détailler ces deux classes de protocoles : basées sur la topologie et basées sur la position géographique.

2.1.2 Protocoles de routage basés sur la topologie

Les protocoles basés sur l'information topologique sont classés en trois grandes catégories : proactif, réactif ou hybride.

2.1.2.1 Protocoles de routage proactif

Ces protocoles sont inspirés des protocoles de routage des réseaux filaires. Ils maintiennent la topologie globale du réseau, soit à l'aide d'une table de routage contenant le prochain saut pour chaque destination, soit à l'aide d'un graphe contenant l'état de tous les liens du réseau.

Largement inspiré du protocole Routing Information Protocol (RIP) [88] *Destination Sequenced Distance-Vector Routing Protocol*, (DSDV) [103] est un des premiers protocoles proposés pour les réseaux ad hoc sans fil. Ce protocole est basé sur une version distribuée de l'algorithme de Bellman-Ford [26] qui maintient une table contenant la longueur et le premier nœud du chemin vers chaque nœud du réseau. La table de routage est envoyée à tous les voisins à intervalle régulier afin de la maintenir à jour. Pour maintenir à jour la table de routage d'un réseau dynamique, l'échange de table complet à chaque changement de topologie implique la transmission d'un grand nombre de messages de signalisation. La taille de ces messages et leur fréquence d'envoi sont proportionnellles au diamètre du graphe représentant le réseau. En effet, plus le réseau est grand, plus la taille de la table de routage est grande, et plus il existe de liens dans le réseau, plus le nombre de changements d'état d'un lien du réseau est grand. Ce problème de *passage à l'échelle*, ou *scalability* en anglais, est fondamentale dans les réseaux en général, et plus particulièrement dans les réseaux sans fil où la ressource radio est limitée.

Optimized Link State Routing Protocol (OLSR) [42], tente de résoudre ce problème de passage à l'échelle. OLSR est un protocole à état de lien. Comme le protocole OSPF pour les réseaux filaires, le protocole maintient l'état de tous les liens du réseau pour construire le graphe du réseau et calculer le plus court chemin à l'aide d'un algorithme comme Dijkstra. OLSR utilise deux messages : *Hello* et *Topology Control* (TC). Le message Hello permet de découvrir les voisins et détecter le changement d'état d'un lien. Le message TC permet de propager une information de changement d'état d'un lien à l'ensemble des nœuds du réseau. Pour permettre le passage à l'échelle du protocole, chaque nœud sélectionne un ensemble de

nœuds appelé *multipoint relays* (MPR). Seuls ces nœuds relient les messages TC. La sélection des nœuds MPR se fait de manière à ce que tous les voisins à deux sauts ont un lien bi-directionnel avec au moins un nœud MPR.

2.1.2.2 Protocoles de routage réactifs

Lorsqu'un nœud source souhaite communiquer avec un nœud destinataire, il lui envoie une requête de recherche de chemin, requête nommée RREQ (Route request). Quand un nœud reçoit cette requête, il la transmet à ses voisins sauf s'il l'a déjà reçue ou qu'il est le destinataire du message. A la réception de la requête, le destinataire répond par un message nommé RREP (Route Reply). La réponse RREP suivra le chemin inverse parcouru par la requête RREQ. Lors de la transmission d'une requête RREQ, selon le type de protocole utilisé, le traitement diffère. Dans le cas d'un protocole à table de routage, tel que AODV [102] ou DYMO [36], le nœud actualise sa table de routage en ajoutant une entrée qui indique le prochain saut pour atteindre le nœud source indiqué dans le message RREQ. Cette entrée permettra le routage du message RREP vers la source. Lorsque qu'un nœud reçoit un message RREP, il exécute un processus similaire pour ajouter une entrée dans sa table de routage pour atteindre la destination recherchée.

Dans le cas d'un protocole de routage à la source *source routing protocol*, tel que DSR [69], à réception d'un message RREQ, le nœud intermédiaire ajoute son identifiant au chemin parcouru depuis la source. Le chemin est stocké dans sa totalité dans le message RREP pour le router vers la source. Une fois que la source a reçu le message RREP, il copie le chemin vers la destination dans sa cache.

Si un nœud intermédiaire connaît le chemin vers la destination alors il peut répondre directement au nœud source à l'aide d'un message appelé *Gratuitious Route Reply*, qui est similaire en tout point à un message RREP, à l'exception qu'il n'est pas envoyé par la destination, mais par un nœud intermédiaire.

Le protocole doit également maintenir les routes en détectant si un lien est rompu. Pendant l'échange de données, un nœud intermédiaire w vérifie que le paquet est bien reçu par le nœud suivant v, sur le chemin. Si le nœud v est le destinataire final du paquet, il envoie un accusé de réception à w ; sinon la transmission par v du paquet au prochain nœud sur le chemin est interprété par w comme un accusé de réception. Si un nœud ne reçoit pas d'accusé de réception à l'expiration d'un délai prédéterminé t, il suppose que le nœud suivant n'est plus accessible. Donc, ce délai t est le temps nécessaire pour considérer un lien comme rompu. Le choix de la valeur de t influe à la fois sur l'overhead généré par le protocole, mais aussi sur le délai moyen d'acheminement d'un paquet. Il est donc important d'adapter sa valeur selon la mobilité du réseau.

2.1.2.3 Protocoles de routage hybride

Zone routing protocole (ZRP) [62] est un protocole de routage ad hoc hybride qui combine les avantages d'un protocole réactif avec les avantages d'un protocole proactif. L'idée est d'utiliser un routage proactif à l'intérieur d'une zone limité à *r-hop* de tout nœud et d'utiliser un routage réactif au delà de cette zone. Un *intra-zone routing protocole* (IARP) est utilisé dans

la zone où le routage est proactif et un *inter-zone routing protocole* (IERP) est utilisé au delà. IARP peut être n'importe quel protocole ad hoc proactif et IERP n'importe quel protocole ad hoc réactif.

2.1.3 Protocoles de routage basés sur la position géographique

Un protocole basé sur la position, appelé aussi protocole géographique, route les paquets à l'aide uniquement de l'information géographique des nœuds. L'utilisation d'un protocole géographique impose donc que chaque nœud connaît sa position géographique à l'aide d'un système tel que le GPS (Global Positioning System). Avant de présenter les protocoles basés sur la position géographique, nous allons décrire les systèmes de géolocalisation existants.

2.1.3.1 Systèmes de géolocalisation

Le positionnement par satellite est la méthode la plus couramment employée pour des raisons de simplicité et d'efficacité. Un système de géolocalisation est composé d'une multitude de satellites en orbite autour de la terre. Un satellite communique avec un équipement de géolocalisation à l'aide d'une transmission sans fil unidirectionnelle (satellite vers nœud).

Pour connaître sa position géographique, un équipement va tout d'abord synchroniser son horloge interne avec celle d'un satellite. Ensuite le satellite fournit sa position absolue au nœud et l'heure à laquelle l'information est envoyée. En fonction du temps de propagation du message (heure de réception du message moins heure du message lors de l'envoi), la distance au satellite est calculée. Une droite passant par la position de l'équipement et celle du satellite est définie. En exécutant le même processus pour chacun des satellites dont il reçoit leurs informations de manière similaire, il définit un ensemble de droites. L'intersection de ces droites est alors la position géographique de l'équipement sur terre (méthode de trilatération).

Le système américain GPS (Global Positioning System) fonctionne sur ce principe à l'aide de la constellation de satellites associés. Un système européen, appelé Galiléo [2], est également en cours de mise en œuvre. Les limites du systèmes GPS est la mauvaise ou la non réception des satellites dans un bâtiment ou bien une forêt dense. L'équipement associé est également trop important en terme d'espace et de ressource pour être associé à certains nœuds, comme les capteurs.

Des systèmes de géolocalisation sans satellite existent également, comme RADAR [17]. Ils se basent sur des nœuds fixes dont on connaît leur position géographique au préalable. A l'aide de plusieurs de ces nœuds, un nœud mobile peut calculer sa position en calculant la distance à ses nœuds à l'aide du temps de propagation du message. Par une méthode de triangulation, le nœud mobile connaît sa position. Cette méthode peut être une alternative au positionnement par satellites pour des raisons d'environnement d'utilisation (bâtiment), d'indépendance par rapport à un système centralisé ; ou bien être associé à un système de géolocalisation par satellite pour améliorer sa précision.

2.1.3.2 Protocole géographique glouton

Les protocoles géographiques appelés *glouton* (*Greedy* en anglais), comme *Most Forward within Radius* (MFR) [118] sont basés sur le concept simple d'envoi des paquets dans la direction de la position géographique du destinataire, c'est le voisin le plus proche de la destination qui transmet le message. Néanmoins un tel protocole a plusieurs inconvénients. Le premier est le surplus de messages générés pour la découverte du voisinage. La fréquence d'envoi de ces messages doit être fonction du changement de topologie du réseau, sa dynamicité, pour maintenir à jour la table de voisinage. Le second problème est causé par l'existence de zones desertées (ou *zone vide*) : l'algorithme ne calcule pas le chemin menant à la destination malgré son existance. En effet, le prochain saut peut temporairement ne pas améliorer la distance à la destination, mais quand même y mener. Le troisième problème est la connaissance nécessaire de la position géographique de la destination. Un service de localisation permet d'obtenir cette information, mais le protocole associé à ce service ne doit pas affecter les performances globales du réseau. Nous allons à présent décrire les protocoles qui tentent de résoudre ces problèmes.

2.1.3.3 Protocole géographique GPSR

Greedy Perimeter Stateless Routing (GPSR) [71], résoud le problème du choix du voisin qui transmet le message vers la destination, sans forcement améliorer la distance géométrique à la destination. Nous allons décrire comment GPSR résoud ce problème. Tout d'abord, le protocole définit précisément une zone où se trouve l'ensemble des nœuds qui améliore le déplacement vers la destination. S'il n'existe aucun nœud dans cette zone, alors aucun voisin n'améliore le déplacement vers la destination ; cette zone est alors nommée *zone vide*. GPSR applique par défaut le protocole Greedy, mais s'il existe une zone vide, alors il applique un routage *périmétrique*. Le routage périmétrique a pour but de contourner la zone vide pour atteindre la destination, à l'aide de nœuds intermédiaires situés à ses abords. Les auteurs choisissent de contourner le périmètre de cette zone à l'aide de *la règle de la main droite*, qui consiste à conserver toujours à sa droite la zone vide. Pour cela, la règle de base et de choisir le premier voisin en les parcourant dans le sens inverse de l'aiguille d'une montre. Malheureusement, avec cette seule règle on ne choisit pas le chemin optimal qui contourne la zone vide. Les auteurs introduisent alors une heuristique appelé *no-crossing* basée sur la théorie des graphes planaires permettant de choisir un chemin proche du chemin optimal.

2.1.3.4 Optimisation avec CBF

Nous avons vu que le routage géographique permet de router un paquet vers la destination en connaissant seulement les positions d'un nœud et de ses voisins. Néanmoins, connaître la position de ses voisins implique d'envoyer, soit une balise appelée *Hello* à intervalles réguliers (mécanisme proactif), soit à la demande (mécanisme réactif) pour donner sa position à ses voisins. La méthode réactive permet d'envoyer une requête de voisinage seulement quand un nœud a besoin de la position de ses voisins. Sur réception d'une telle requête, les voisins répondent par un message contenant leur position géographique.

Deux optimisations sont utilisées. La première consiste à enregistrer la position des

voisins dans un cache. Dans la seconde optimisation, la balise peut être associée à un autre message (piggyback) pour limiter l'overhead. Malgré ces optimisations, l'overhead généré par la découverte des voisins n'est pas négligeable [53]. Dans [53], une méthode appelée CBF (Contention-Based Forwarding) permet au nœud de décider s'il doit transmettre ou non un paquet sans connaître la position de ses voisins. A la base, tout voisin qui améliore le déplacement du paquet vers la destination est candidat à la transmission de ce paquet. Le but de CBF est qu'un seul candidat puisse s'auto-proclamer le gagnant, c'est-à-dire le nouveau transmetteur. Pour cela, CBF inclut trois étapes : premièrement, le nœud transmetteur envoie le paquet à tous ses voisins. Deuxièmement, les voisins entrent en compétition entre-eux. C'est pendant cette *période de compétition* ("contention period" en anglais) qu'est élu le nouveau transmetteur, c'est-à-dire le nœud suivant sur le chemin vers la destination. Troisièmement, le voisin qui a gagné la compétition, s'auto-proclame comme le nouveau transmetteur, et *supprime* tous ses voisins encore en compétition. Deux méthodes restent à décrire : le déroulement de la compétition et la stratégie de suppression.

La compétition se ramène au problème du choix d'un nœud de manière distribuée. La méthode la plus répandue utilise un *timer*, et c'est celle choisie dans CBF pour élire le transmetteur. Chaque voisin en compétition attend un certain délai. A l'expiration de ce délai d'attente, le voisin s'auto-proclame comme le nouveau transmetteur. Le délai d'attente t est défini par l'équation 2.1. T le délai maximum de transmission et P est le taux de progression défini par l'équation 2.2.

$$t(P) = T(1 - P) \tag{2.1}$$

P est fonction de : f la position du transmetteur, z la position de la destination, n la position du voisin, et R le rayon de couverture radio d'un nœud. Ainsi, le délai d'attente t d'un nœud est proportionnel à la distance qui le sépare de la destination et le voisin le plus proche de la destination attend le plus court délai. C'est bien le voisin le plus proche de la destination qui est le nouveau transmetteur. Malheureusement cette méthode peut mener à choisir plusieurs voisins si le délai d'attente est très proche entre deux voisins, c'est-à-dire les voisins ont une distance quasiment identique à la destination. Ce problème peut entrainer une duplication de paquets car dans ce cas plusieurs transmetteurs sont élus.

$$P(f, z, n) = \max\left\{0, \frac{dist(f, z) - dist(n, z)}{R}\right\} \tag{2.2}$$

Lorsqu'un nœud s'auto-proclame transmetteur, il doit supprimer tout les candidats existants. Pour cela CBR propose trois méthodes : basique, zone, active. La méthode *basique* consiste à *écouter* les paquets envoyés par ses voisins pendant le délai de compétition. Si ce paquet est le paquet à transmettre alors cela implique l'élection du nouveau transmetteur, le voisin annule alors la compétition, il ne sera pas le nouveau transmetteur (le voisin est supprimé). Cette solution comporte un désavantage : seuls les voisins à portée radio du nouveau transmetteur reçoivent les paquets transmis. Tous les autres voisins restent en compétition ; entrainant une duplication de paquets.

La méthode à *zones* se base sur trois zones présentée en Figure 2.1. La première zone, appelée *triangle de Reuleaux* [59], est celle contenant les voisins recevant le paquet transmis par le voisin élu. Dans un premier temps, le nœud envoie le message à transmettre à ses

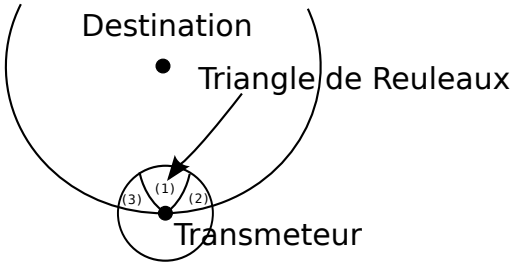


Figure 2.1 – Trois zones de transmission pour éviter la duplication de messages

voisins. Les voisins situés dans la première zone exécutent le processus d'élection et de suppression identique à celui décrit précédemment. Si un nœud existe bien dans cette zone, alors les nœuds qui ne sont pas élus seront bien supprimés par le processus de suppression décrit précédemment, puisque la réception par tous les nœuds en compétition du paquet transmis est garantie. Le problème de la méthode basique est ainsi résolu. Si aucun nœud ne se trouve dans cette première zone, alors le transmetteur initie une nouvelle élection dans la seconde zone puis dans une troisième où se trouve potentiellement des nœuds améliorant la transmission du paquet vers la destination. Malgré tout, cette méthode n'élimine pas la duplication de paquets car plusieurs nœuds peuvent être élus s'ils sont à une distance similaire de la destination.

La dernière méthode, appelée *active*, permet d'éliminer toute duplication de paquets. Elle s'inspire des messages Clear To Send, et Request To Send (CTS/RTS) de la méthode utilisé par IEEE 802.11 (§ 1.1.4.2). Au lieu d'envoyer immédiatement le paquet à transmettre, le transmetteur envoie à ses voisins un message Request To Forward (RTF). Les nœuds voisins exécutent alors la même méthode de compétition/suppression décrit précédemment, mais à la fin de la période de compétition, au lieu de transmettre le paquet, il envoie au transmetteur un message CTS. Le transmetteur choisit le nouveau transmetteur parmi les voisins qui lui ont envoyé un CTS. Il envoie alors le paquet à ce nouveau transmetteur. Un seul paquet est bien transmis.

2.1.3.5 Services de localisation

Le service de localisation est utile pour connaître la position géographique de la destination. Une taxonomie des services de localisation est proposée dans [43] représentée par la Figure 2.2. Tout d'abord les services de localisations sont séparés en deux approches : *flooding-based* (gloutone, basée sur l'inondation de messages sur le réseau) et *rendez-vous-based*. Les protocoles flooding-based sont divisés eux-même en approches proactive et réactive. Dans l'approche flooding-based proactive, chaque nœud (potentiellement destinataire) envoie périodiquement sa position à tous les autres nœuds du réseau qui maintiennent à jour une table de positions. L'intervalle d'envoi de la position est déterminé en fonction de la mobilité du réseau. Un compromis doit être trouvé entre maintenir la table de position des nœuds suffisamment à jour et limiter la signalisation dûe à l'inondation du message d'actualisation des positions. DREAM [23] est un bon exemple de protocole flooding-based proactive. Dans le cas d'un protocole flooding-based réactif (appelé aussi Reactive Location

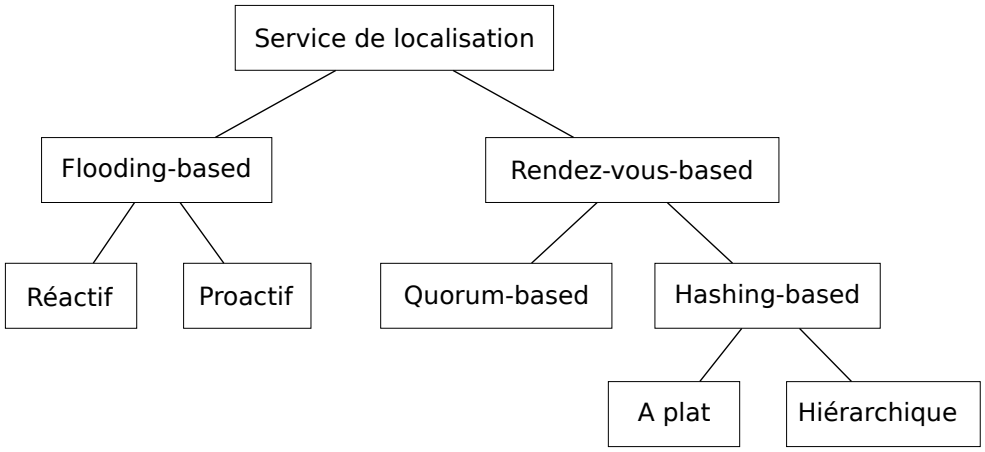


Figure 2.2 – Taxonomie des services de localisation

Service, RLS) , si un nœud ne possède pas la position géographique de la destination, alors il envoie une requête nommée LREQ (Location Request) qui est diffusée dans le réseau. Lorsque le nœud destinataire reçoit la requête LREQ, il envoie une réponse nommée LREP (Location Reply) vers la source à l'aide d'un protocole géographique (la position géographique de la source étant stockée dans la requête LREQ). Ce processus de localisation est donc similaire au processus de découverte de chemin d'un protocole topologique réactif.

Pour limiter le nombre de messages de signalisation générés par un protocole de service de localisation, une approche rendez-vous-based est proposée. Cette approche consiste à associer tout identifiant unique d'un nœud du réseau avec un ou plusieurs autres nœuds du réseau. Le nœud associé est alors appelé *serveur de localisation*. Comme le montre la Figure 2.3, chaque nœud envoie sa position au(x) nœud(s) serveurs, avec lequel ou lesquels il est associé à l'aide d'un message appelé *location update*. Lorsqu'un nœud souhaite connaître la position d'un autre nœud, dont il connaît l'identifiant, alors il envoie une requête appelée *location request* à un des serveurs de localisation associés à l'identifiant. Le serveur envoie en unicast la position du nœud demandée à la source à l'aide d'un message *location reply*. Deux approches différentes pour l'association sont proposées : *quorum-based* et *hashing-based*. Dans l'approche quorum-based, le message *location update* est envoyé à un sous ensemble (update quorum) de nœuds disponibles, et le message location request est envoyé à un sous ensemble différent de nœud (query quorum). Les deux sous-ensembles sont déterminés tel qu'il existe une intersection non vide et donc pour satisfaite la requête par au moins un nœud de l'ensemble query quorum. Plusieurs méthodes pour générer le système de quorum sont proposées dans [61].

Dans les protocoles hashing-based, l'association entre l'identifiant et le serveur de localisation est réalisée à l'aide d'une fonction de hachage, soit dans l'espace de l'identifiant d'un nœud, ou soit dans l'espace de la localisation d'un nœud. Les protocoles hashing-based sont dit *hiérarchiques* si une hiérarchie de sous-zones est utilisée, sinon ils sont dit *à plat*. Dans un protocole hashing-based hiérarchique [126, 81], la zone dans laquelle se trouve le serveur de

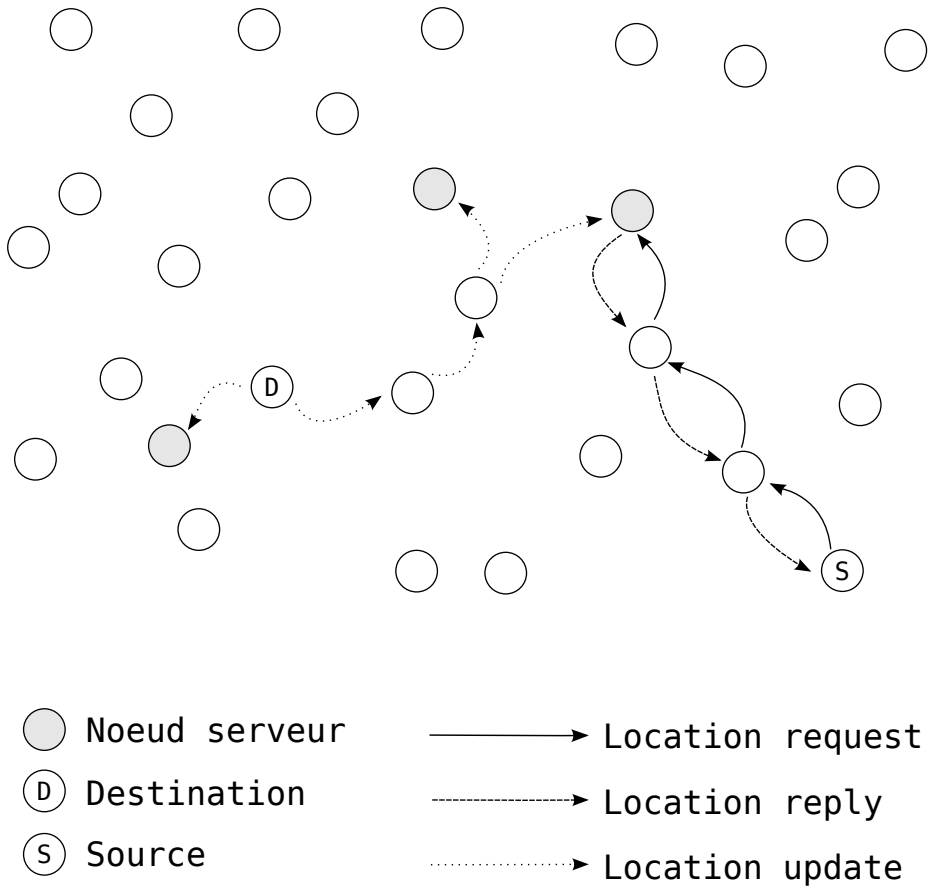


Figure 2.3 – Approche "rendez-vous" pour la localisation géographique d'un noeud.

localisation est récursivement divisée en une hiérarchie de grilles de plus en plus petites. Pour chaque nœud, un ou plusieurs serveurs de localisation sont choisis pour chaque grille à chaque niveau de la hiérarchie. Les messages location update et location request traversent la hiérarchie de haut en bas. Un exemple de protocole hashing-based hiérarchique est Grid Location Service (GLS) [81]. GLS effectue le hachage dans l'espace des identifiants d'un nœud. La zone géographique dans laquelle se trouvent les nœuds est divisée en une hiérarchie de carrés de plus en plus petits (Figure 2.4).

Un nœud *B* doit recruter un serveur de localisation dans chaque hiérarchie. La stratégie du choix du nœud dans chaque hiérarchie se base sur le nœud ayant l'*identifiant le plus proche* de l'identifiant du nœud *B*. GLS calcule cet identifiant en appliquant une fonction de hachage au *nom unique* du nœud. Ce nom unique peut être le nom de la machine, l'adresse IP, ou l'adresse MAC.

Dans la Figure 2.4, les serveurs de localisation d'ordre 1 pour le nœud B (17) sont 2,

23 et 63. Ceux d'ordre 2 sont 43, 31 et 26. Enfin ceux d'ordre 3 : 37, 19 et 20. Nous allons décrire comment une requête de demande de localisation est effectuée. A chaque étape, la requête est transmise au nœud de niveau supérieur qui a l'identifiant le plus proche. Par exemple, toujours sur la Figure 2.4, si le nœud A (76) souhaite localiser le nœud B (17), il identifie le nœud le plus proche d'ordre 1, qui est lui-même. Ensuite, comme ce nœud n'est pas un nœud serveur du nœud B, il envoie la requête au nœud 21, qui est le meilleur nœud du carré adjacent d'ordre 1 (celui possédant l'identifiant le plus proche de B). Le nœud 21 trouve le nœud 20 qui est le meilleur nœud dans le carré adjacent d'ordre 2, et le nœud 20 est le serveur de localisation pour B, il peut alors répondre à la requête. L'actualisation de la localisation (par un message location update) est transmise de la même manière à intervalles réguliers. Cet intervalle doit être déterminé pour trouver un compromis entre la mobilité du nœud et le maintien d'une position valide dans les serveurs de localisation.

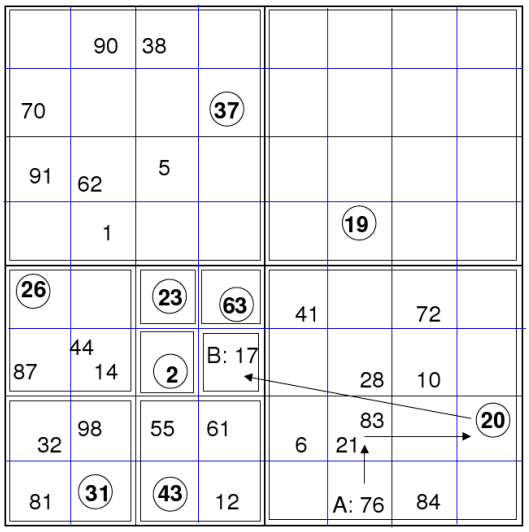


Figure 2.4 – Localisation avec GLS. Les nœuds en gras entourés sont les serveurs de localisation de B. A est un nœud souhaitant connaître la position de B. (*Issu de* [81])

Les protocoles hasing-based à plat décrits, entre autres, dans [58, 117, 123] utilisent une fonction de hachage pour associer l'identifiant d'un nœud à une *home zone*. Tous les nœuds dans la home zone maintienent l'information de localisation pour le nœud associé, et sont donc des serveurs de localisation pour ce nœud. La Figure 2.5 montre les processus d'actualisation et de requête d'actualisation. Le nœud A souhaite connaître la localisation du nœud B. Il calcule, à l'aide d'une fonction de hachage, la *home zone*. Ensuite il envoie, à l'aide du protocole géographique, la requête de localisation vers les *home zones*. La requête est alors transmise à tous les nœuds de la home zone, et le nœud qui possède la position du nœud B répond à cette requête. L'actualisation de la localisation se fait de manière similaire en envoyant périodiquement aux nœuds de la home zone un message location update. Dans [43] est introduit Geographic Hashing Location Service (GHLS) qui pousse le principe des protocoles hasing-based à plat à son extrême : GHLS considère la home zone comme étant un seul nœud : le nœud le plus proche de la position obtenue par une fonction de hachage. Les

processus d'envoi des messages vers le serveur est similaire à celui décrit précédemment. Le nœud A calcule la position associée à l'identifiant de B à l'aide de la fonction de hachage. Ensuite il envoie la requête vers cette position, et le nœud le plus proche de cette position, qui est le serveur de localisation, répond à cette requête. GHLS prévoit que le serveur de localisation peut transmettre la position d'un nœud auquel il est associé à un autre nœud voisin s'il s'éloigne trop de la position associée à l'identifiant du nœud.

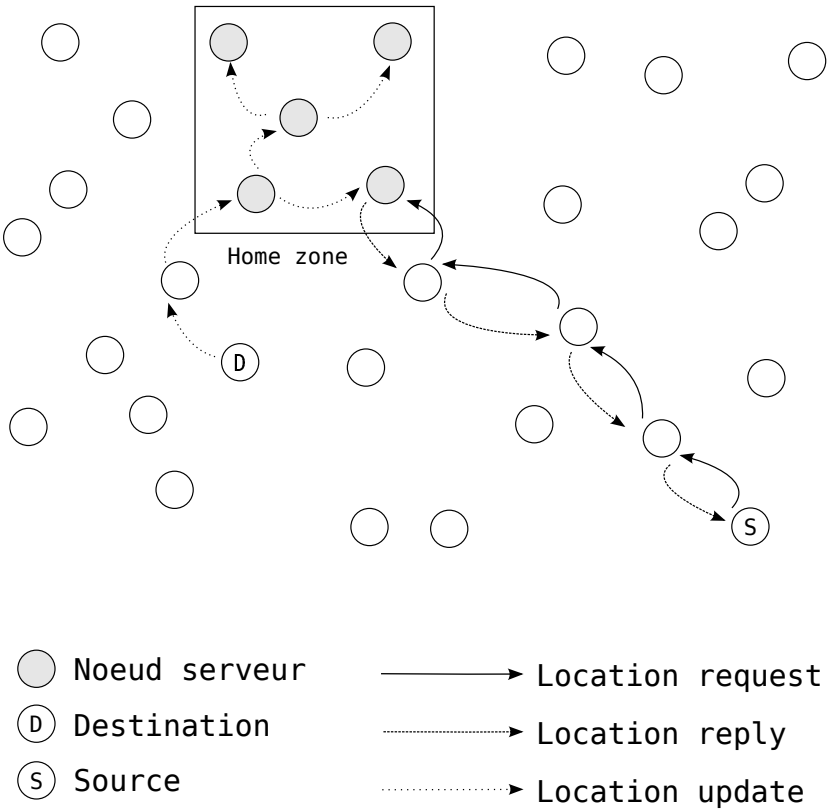


Figure 2.5 – Localisation à "rendez-vous" avec hachage pour lier la destination à une *home zone*

Dans [43] les différents protocoles de localisation sont comparés en fonction du nombre de messages de signalisation envoyés dans le réseau. Pour cela, un modèle théorique est proposé pour le cas statique, et des simulations sont effectuées pour le cas dynamique.

2.1.4 Normalisation des protocoles ad hoc par l'IETF

La normalisation des protocoles de routage est primordiale pour que tout équipement réseau puisse communiquer entre eux. L'Internet Engineering Task Force(IETF) possède des groupes de travail qui ont pour objectif de standardiser les protocoles de l'internet. Ce

groupe de travail appelé Mobile Ad-hoc Networks (MANET) [5] traite des protocoles de routage IP existants pour les réseaux sans fil dont la topologie est dynamique ou statique. Le cahier des charges de ces protocoles est la simplicité, la disponible pour un grand nombre de matériel, l'intégration dans des infrastructures IP existantes. Reprenant les travaux aboutis sur les protocoles de routage proactif ou réactif, le groupe de travail a développé deux classes de protocoles ad hoc :

– Reactive MANET Routing Protocol (RMRP)
– Proactive MANET Routing Protocol (PMRP)

Le groupe de travail précise également que IPv6 et IPv4 doivent être supportés. De plus, la sécurité de ces réseaux doit être étudiée.

Le groupe de travail a déjà publié plusieurs RFC concernant les protocoles de routage ad hoc AODV [101], OLSR [41], DSR [67], mais également des travaux plus généralistes concernant tous protocoles de routage ad hoc confondu, comme le format des messages [40], la définition d'un TLV (type-length-value structure) pour la représentation des valeurs de temps (intervalle ou durée) [38], ou bien des recommandations sur la variation du délai [39].

Des *drafts* sont également régulièrement publiés par le groupe de travail MANET, permettant d'avoir un aperçu des recherches actuelles et des protocoles qui aboutiront peut-être à une RFC.

2.2 Les réseaux ad hoc hybrides

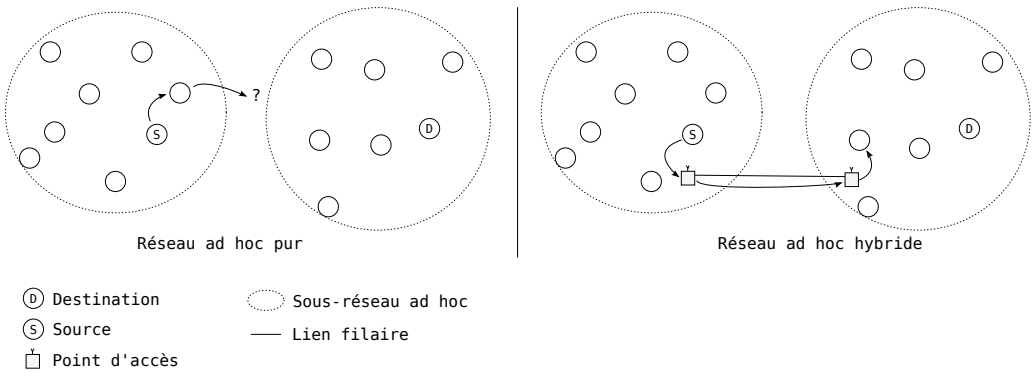


Figure 2.6 – Amélioration de la connexité du réseau ad hoc à l'aide de points d'accès.

Un réseau ad hoc n'a pas besoin d'infrastructure fixe pour fonctionner. Cependant un tel réseau présente deux limites : la connexité du réseau et la connectivité à l'internet. Un réseau est dit *connexe* si tout nœud du réseau peut atteindre n'importe quel autre nœud du réseau. Si le réseau ad hoc n'est pas suffisamment dense, il arrive qu'il ne soit plus connexe et coupe le réseau ad hoc en plusieurs sous réseaux distincts (Figure 2.6). La connectivité du réseau à Internet permet à tout nœud du réseau d'accéder à des correspondants sur Internet. L'ajout de points d'accès sans fil dans le réseau permet alors de résoudre ces deux problèmes. Ce type d'architecture est nommé *réseau ad hoc hybride*. Nous allons présenter

les differentes architectures hybrides existantes pour ensuite détailler des processus liés, tels que la découverte de points d'accès et la méthode de transmision à un point d'accès. Des protocoles pour réseaux ad hoc hybrides utilisant ces processus seront ensuite présentés.

2.2.1 Les différentes architectures des réseaux ad hoc hybrides

Un réseau ad hoc hybride comporte, en plus des nœuds mobiles, appelés aussi *mobile node* (MN), des *points d'accès sans fil*. Ces points d'accès sont parfois appelés, *passerelle* ou *station de base*. Ces termes existent dans la littérature anglophone sous le nom d'*access point* (AP), *gateway* (GW) ou *base station* (BS). Le terme point d'accès est plutôt utilisé dans le contexte des WLAN, typiquement des équipements WiFi. Un point d'accès peut être vu aussi comme une passerelle, c'est-à-dire un dispositif reliant deux réseaux : le réseau sans fil et Internet. Enfin, le terme station de base est emprunté au réseau cellulaire. Par la suite, par souci d'homogénéité, nous gardons le terme *access point* (AP) pour désigner un nœud du réseau ad hoc possédant au moins deux interfaces réseau : une interface sans fil reliée au réseau ad hoc, et une autre, soit reliée à d'autres points d'accès, soit à un réseau d'accès, ou soit directement à Internet. Cette seconde interface peut être sans fil ou filaire. Ces trois méthodes de liaison à un réseau extérieur du réseau ad hoc, nous conduisent à considérer trois architectures de réseau ad hoc hybride différentes (Figure 2.7) : MESH (a), réseau d'accès (b), passerelle (c).

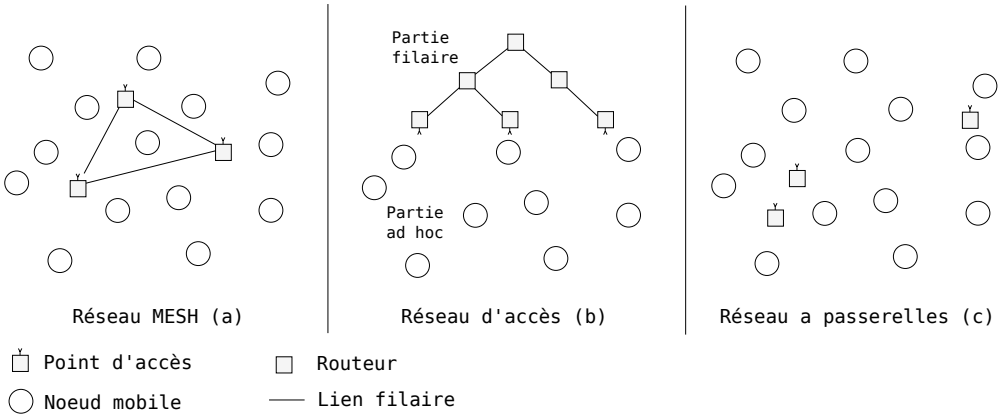


Figure 2.7 – Trois architectures de réseau ad hoc hybride. Dans (c) les points d'accès sont reliés à Internet.

Un réseau MESH est une terminologie souvent employée pour désigner un réseau ad hoc. Mais un réseau MESH ne couvre pas tous les domaines des réseaux ad hoc. Ici nous considérons un réseau MESH comme étant un réseau ad hoc dont les nœuds sont fixes ou très peu mobiles. Les points d'accès d'un réseau ad hoc hybride MESH sont reliés entre eux par un autre réseau ad hoc par maillage. Les points d'accès ont donc deux interfaces sans fil.

Dans une architecture à réseau d'accès, comme dans un réseau cellulaire, les points d'accès font partie d'un réseau d'accès. Ce réseau d'accès est parfois hiérarchisé, c'est-à-dire qu'il comporte des routeurs où sont reliés plusieurs point d'accès, et ces routeurs sont eux-mêmes reliés à d'autres routeurs de niveau supérieur. Certains routeurs peuvent être re-

liés par l'intermédiaire d'une passerelle au réseau Internet. Cette architecture, inspirée des réseaux cellulaires peut aussi être appelée réseau cellulaire multi-hop. Les terminaux des réseaux cellulaires multi-hop peuvent transmettre des paquets à d'autres terminaux, tout comme les nœuds d'un réseau ad hoc. Ainsi on parle *d'extension de couverture* de réseaux cellulaires. On peut voir un réseau ad hoc hybride comme un réseau cellulaire multi-hop et un réseau cellulaire multi-hop comme un réseau ad hoc hybride. Ce sont deux visions de la même architecture.

Enfin, la dernière architecture, la plus simple, comporte des passerelles (fixes ou mobiles) réparties dans le réseau ad hoc et directement reliées à Internet.

Quelque soit l'architecture ou le protocole utilisés, les points d'accès doivent être connus des nœuds mobiles du réseau ad hoc. Nous allons décrire ce processus de *découverte de points d'accès*.

2.2.2 Découverte de points d'accès

Dans un réseau sans fil à un saut, le point d'accès s'annonce à l'aide d'une *balise* ("beacon" en anglais) envoyée périodiquement. C'est le cas des points d'accès WiFi dans les WLAN, les réseaux cellulaires au niveau de la couche 2 (MAC), et dans la mobilité des réseaux IP (Mobile IP) au niveau de la couche 3 (Réseau). Une balise contient des informations comme l'identifiant du point d'accès, le nom du réseau, les services fournis par le réseau, etc. Un nœud mobile recevant cette balise est informé de la présence d'un point d'accès dans son voisinage. Un nœud mobile peut recevoir plusieurs balises de plusieurs points d'accès différents. Dans ce cas, le nœud mobile enregistre dans une table les points d'accès accessibles et choisit celui de son choix en fonction des paramètres qu'il a à sa disposition comme la qualité du signal. Nous remarquons que la découverte de point d'accès est proche de la découverte de service. En effet, l'accès à Internet peut être vu comme un service accessible dans le réseau ad hoc.

Un nœud mobile d'un réseau ad hoc hybride peut accéder à un point d'accès directement (en un saut) ou alors en plusieurs sauts. En effet, l'utilisation d'un réseau ad hoc implique qu'il existe des communications multi-saut vers le point d'accès. Si un point d'accès envoie une balise à son voisinage direct, un nœud mobile situé à plus d'un saut du point d'accès ne sera pas informé de la présence d'un point d'accès accessible. Il est alors nécessaire de développer une stratégie appelée *découverte de point d'accès*. La littérature parle souvent de découverte de passerelle (*gateway discovery* en anglais). Comme le routage dans les réseaux ad hoc, cette découverte de point d'accès fonctionne à la couche 3. Il existe trois méthodes de découverte de points d'accès : réactive, proactive, hybride :

Proactive Chaque point d'accès diffuse une balise dans le réseau ad hoc. Un nœud mobile recevant cette balise la transmet à tous ses voisins.

Réactive Un nœud mobile diffuse une requête de découverte de point d'accès dans le réseau ad hoc. Un nœud mobile recevant cette requête la transmet à ses voisins. Un point d'accès recevant cette requête répond directement au nœud mobile par un message contenant les mêmes informations qu'une balise.

Hybride Les points d'accès diffusent une balise dans le réseau ad hoc, mais seulement aux nœuds mobiles situés à proximité. Un nœud mobile situé au-delà peut diffuser une

requête de découverte de points d'accès. Les nœuds mobiles recevant cette requête peuvent répondre directement au nœud mobile comme le fait un point d'accès.

La diffusion de balises, comme celle d'une requête est toujours limitée par un Time To Live (TTL), définissant le nombre de sauts maximal parcourus par un paquet. Ce paramètre permet à un message de ne pas être diffusé dans tout le réseau, et donc de ne pas saturer l'ensemble du réseau. L'avantage de la méthode proactive est qu'à partir d'une seule diffusion, tous les nœuds mobiles du réseau à k-sauts, suivant la valeur du TTL, sont informés de la présence du point d'accès. Cependant une diffusion génère un trafic non négligeable et inutile si peu de nœuds souhaitent accéder au point d'accès. Dans ce cas, la méthode réactive est recommandée. En effet, seuls les nœuds mobiles souhaitant communiquer diffusent une requête de demande de chemin dans le réseau. Par souci de compromis entre les deux méthodes, une troisième méthode, dite hybride, est développée dans [105]. Les point d'accès diffusent la balise seulement aux nœuds se trouvant à proximité en limitant le TTL. C'est la méthode proactive. Mais les nœuds les plus éloignés peuvent diffuser des requêtes de découverte de points d'accès (méthode réactive) et un nœud mobile peut répondre à cette requête s'il a reçu la balise d'un point d'accès. L'étude [57] compare les performances de ces trois méthodes de découverte de point d'accès. Basés sur des simulations avec plusieurs scénarios de mobilités, les résultats montrent une méthode proactive plus performante en termes de délai et d'overhead. La méthode hybride montre des performances situées entre la méthode réactive et proactive.

L'utilisation d'un chemin multi-saut vers un point d'accès a des conséquences sur la détection de mouvement et donc le déclenchement des handovers. En effet, une communication sans fil directe à un point d'accès dans un réseau cellulaire ou WLAN permet d'obtenir, via la couche 2, les points d'accès accessibles et la qualité du lien vers ces points d'accès en terme de puissance du signal, interférences, ou d'erreurs de transmissions. Ces informations permettent à la fois de pouvoir choisir le meilleur point d'accès pour garantir la qualité de la communication mais aussi de prédire une coupure et d'initier un changement de point d'accès avant qu'il ne soit plus accessible ou que la qualité du lien se dégrade trop. En revanche, une communication multi-sauts implique plusieurs liens de qualités hétérogènes. La prévision de coupure et le choix du meilleur chemin vers un point d'accès n'est pas fiable avec les seules informations des couches plus basses.

Dans [20], les auteurs proposent une solution, appelée VANETII (VANET Internet Integration) permettant à la fois de limiter l'overhead généré par la diffusion périodique de messages de découverte de points d'accès et de choisir un chemin le plus stable possible vers le point d'accès. Comme son nom l'indique, VANETII est adapté aux réseaux de véhicules (que nous détaillerons en 2.3), mais la méthode de découverte de points d'accès peut être appliquée dans n'importe quel réseau mobile multi-sauts. Dans ce protocole, le nœud a besoin de connaître sa position géographique. Il se base sur le même principe que CBF (§ 2.1.3) : un nœud ne transmet pas immédiatement le message d'avertissement de point d'accès et attend un délai proportionnel à une valeur F. Cette valeur dépend d'un paramètre de stabilité S des liens vers le point d'accès et d'un paramètre qui exprime la progression P du message en terme de distance euclidienne au point d'accès ; ce qui limite la duplication de messages et donc l'overhead. Ces deux paramètres sont pondérés par une valeur α. L'équation 2.3

exprime la valeur F en fonction de S et P.

$$F = \alpha S + (1 - \alpha)P \tag{2.3}$$

2.2.3 Méthodes de transmission à un point d'accès

La transmission des paquets vers le point d'accès ne pose pas de problème si le nœud mobile est directement connecté au point d'accès. Dans un environnement multi-saut, l'accès au point d'accès se fait en plusieurs sauts. Les paquets transmis vers un réseau extérieur au réseau ad hoc doivent être routés correctement. De manière générale, les protocoles de routage ad hoc, contrairement aux réseaux classiques, n'incluent pas la fonctionnalité de routage vers une destination extérieure au réseau. L'utilisation d'un protocole ad hoc proactif résoud le problème implicitement car chaque nœud peut maintenir le prochain saut vers une destination extérieure. De même, le routage par la source d'un protocole réactif (comme DSR), permet d'acheminer les paquets en suivant le chemin vers la passerelle. Néanmoins, un protocole réactif comme AODV n'utilise pas de chemin pour atteindre une destination et ne possède pas une table de routage représentant la topologie entière du réseau, le problème de transmission à un point d'accès se pose.

Le routage dans un réseau ad hoc est dit *à plat*, contrairement à un réseau IP, les adresses des nœuds n'ont pas de préfixe qui permet de déterminer dans quel réseau se trouve un nœud. Il n'est alors pas possible de savoir si une destination est dans le réseau ad hoc ou dans un réseau extérieur, comme Internet. Les protocoles proactifs considèrent un nœud extérieur au réseau ad hoc si et seulement si la destination n'est pas présente dans sa table de routage. Contrairement aux protocoles proactifs, les protocoles réactifs découvrent les nœuds du réseau ad hoc seulement au moment où il souhaite communiquer avec un nœud. La méthode la plus courante pour déterminer si le nœud est dans le réseau ad hoc est d'envoyer un requête de demande de chemin. Si le nœud source reçoit une réponse vers la destination alors il sait que le nœud appartient au réseau ad hoc sinon, après une durée d'attente prédéterminée (timeout), la source considère que la destination n'est pas accessible et donc que la destination est située dans un réseau extérieur.

Si le nœud destinataire est considéré comme se situant dans un réseau extérieur, alors il existe deux méthodes pour transmettre le paquet au point d'accès [95] : la route par défaut et le tunnel.

La méthode de *la route par défaut* se base sur l'affectation de la route par défaut, dans la table de routage d'un nœud, à l'adresse du point d'accès ou l'adresse du nœud suivant sur le chemin vers le point d'accès. Les deux méthodes sont décrites dans [95] et la seconde approche est conseillée. Quelque soit la méthode utilisée, la route par défaut a un désavantage : lorsqu'une route vers un point d'accès est actualisée, un nœud intermédiaire n'étant pas dans le chemin avant l'actualisation, doit modifier les informations pour le routage dans le sens de communication nœud mobile vers le réseau extérieur (up). Autrement, il ne sera pas capable de transmettre les paquets dans le sens montant. La méthode à tunnel permet de résoudre ce problème. Cette méthode, présentée dans [70], encapsule les paquets à destination du point d'accès. Le paquet encapsulé est envoyé au point d'accès avec son adresse IP. Le routage de ce paquet est assuré par le protocole ad hoc. Arrivé au point d'accès, le

paquet est désencapsulé et envoyé sur Internet. Dans la direction inverse, du point d'accès au nœud mobile, le paquet n'est pas encapsulé, l'adresse destination est juste remplacée par celle du nœud mobile sur le réseau. La transmission par encapsulation seulement dans un sens est appelé demi-tunnel.

Les performances de ces différentes méthodes sont évaluées dans [70] en terme d'overhead, de débit utile, taux d'acheminement. Les auteurs conseillent l'utilisation de tunnel au vu des performances évaluées et des raisons citées ci-dessus.

2.2.4 Routage dans les réseaux ad hoc hybrides

Un grand nombre de protocoles ont été proposés pour les réseaux ad hoc hybride, principalement pour la connectivité à Internet, ou *global conectivity*. Mais d'autres permettent d'associer deux protocoles existants : un protocole ad hoc et un protocole pour la mobilité. Il est intéressant de noter qu'un protocole spécifique n'est pas nécessaire pour assurer les communications dans un réseau ad hoc hybride. En effet, tout protocole pour réseau ad hoc peut, en théorie, assurer le routage dans un réseau ad hoc hybride, les points d'accès étant vus comme n'importe quel nœud du réseau ad hoc, à la seule différence qu'ils possèdent plusieurs interfaces radios. Mais le routage n'est alors pas optimal du point de vue des performances du réseau si le protocole n'est pas adapté. C'est pour cela que les auteurs de DSR [69] ont rapidement proposé une extension de leur protocole [30] pour prendre en considération les nœuds possédant plusieurs interfaces hétérogènes. Nous allons détailler cette extension.

2.2.4.1 Extensions de DSR

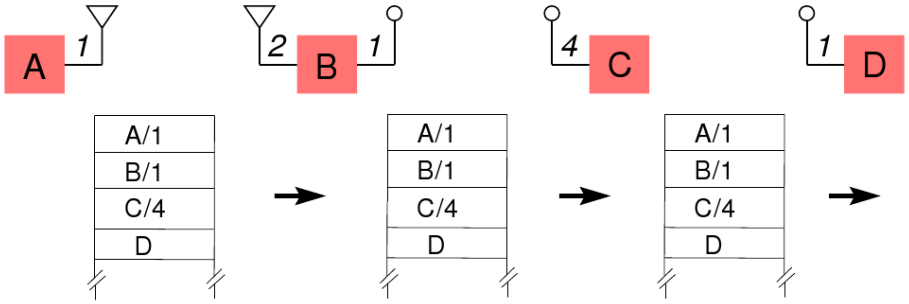


Figure 2.8 – Chemin créé avec DSR entre la source et la destination avec des nœuds à plusieurs interfaces. (*Issu de* [30])

Cette extension est basée sur l'ajout d'un identifiant de l'interface dans le chemin entre la source et la destination. Le chemin contient donc une liste de couples (identifiant du nœud, interface). Les paquets suivent ce chemin en utilisant l'interface appropriée (Figure 2.8). En plus de cette extension, les auteurs proposent une intégration à Internet. L'intégration à Internet n'est ni plus ni moins qu'une architecture de type *passerelle* pour la connectivité

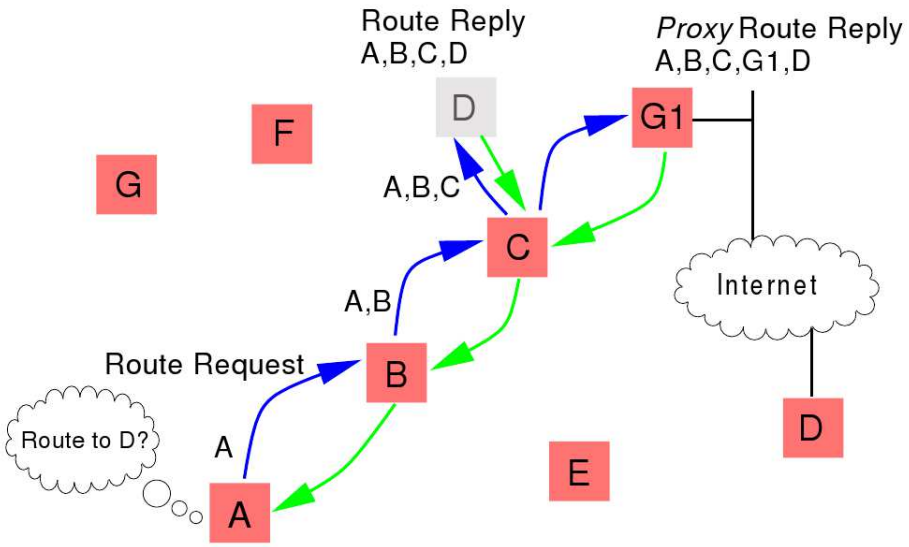


Figure 2.9 – Connectivité à Internet d'un réseau ad hoc utilisant routage DSR à l'aide d'une passerelle. (*Issu de* [30])

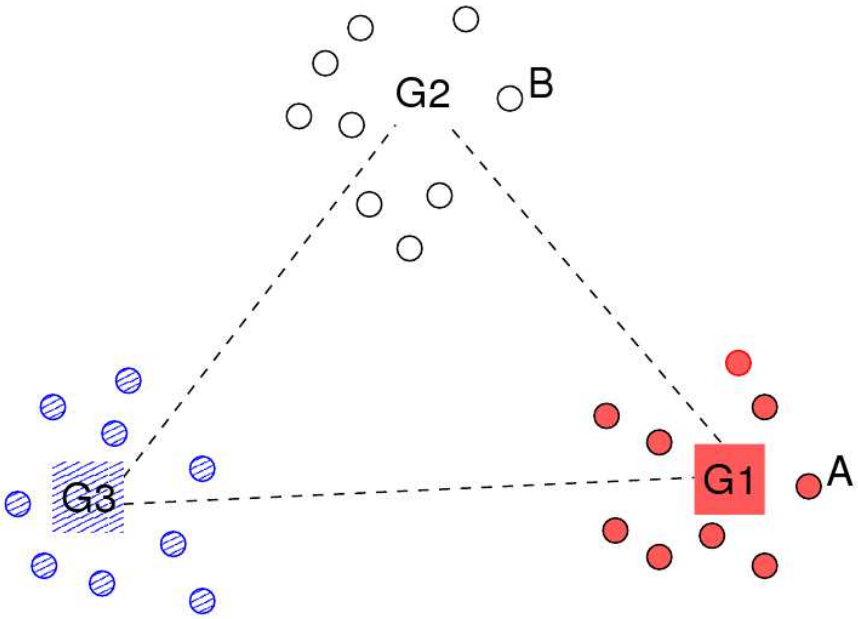


Figure 2.10 – Trois réseaux ad hoc sont reliés à l'aide de passerelles. (*Issu de* [30])

à Internet (Figure 2.9). Les nœuds possédant une interface vers le réseau ad hoc hybride et une interface vers le réseau Internet, sont appelés *gateway*. Une gateway peut envoyer un *proxy route reply* à la source en réponse d'une route request. Ce proxy route reply est un route reply non pas envoyé par la destination, mais par une gateway. Le proxy route reply contient donc le chemin entre la source et la gateway. Ces messages proxy route reply sont similaires au *gratuitious route reply* envoyés par un nœud intermédiaire. De telles gateways peuvent également être utilisées pour relier plusieurs réseaux ad hoc (Figure 2.10), ajoutant une hiérarchie au routage. L'intégration à Mobile IP est réalisée en permettant à un nœud mobile de pouvoir atteindre un agent étranger. La méthode pour découvrir un agent étranger est réactive : la sollicitation de l'agent (*Agent Solicitation*) est adossée à un message *Route Request*. Quand l'agent étranger reçoit la sollicitation il répond en envoyant sur le réseau ad hoc un *Agent Advertisement* directement au nœud source ou nœud mobile. Le nœud mobile possède alors le chemin pour atteindre l'agent étranger.

2.2.4.2 Intégration de protocoles ad hoc à Mobile IP

Peu après, dans [70] il est proposé une solution pour connecter un réseau ad hoc à Internet à l'aide de Mobile IP. Le protocole proposé, appelé MIPMANET, associe un protocole ad hoc réactif, AODV dans leur étude, avec Mobile IP pour IPV4. La différence avec le protocole pour DSR décrit précédemment est qu'un nœud mobile souhaitant communiquer vers un correspondant situé sur Internet utilise son *adresse mère*. Il encapsule alors le paquet pour le transmettre à l'agent étranger (typiquement situé dans la passerelle), qui le transmet, après désencapsulation, au correspondant. Dans le sens opposé, du correspondant au nœud mobile du réseau ad hoc, l'agent étranger délivre le paquet au nœud mobile à l'aide du protocole de routage ad hoc. Contrairement à l'extension de DSR décrite précédemment, l'annonce des agents étrangers ne se fait pas de manière réactive, mais de manière proactive : les agents étrangers diffusent périodiquement leur annonce dans le réseau ad hoc . Les auteurs ont fait leurs simulations avec une période de 5 secondes (alors qu'elle est d'une seconde maximum pour Mobile IP), et le temps de vie d'un agent étranger est fixé à trois fois cette période, ce qui implique qu'au bout de 15 secondes sans recevoir de message de l'agent étranger il est considéré comme inaccessible.

Il existe également une intégration des protocoles ad hoc proactifs à Mobile IP. Dans [27], les auteurs proposent une intégration d'OLSR [42, 41] et Mobile IP dans les futurs réseaux tout IP. L'évaluation de cette intégration est détaillée dans [28]. L'architecture (Figure 2.11) proposée est composée d'un réseau d'accès appelé OLSR-IP qui est connecté à Internet par l'intermédiaire de Mobile IP. OLSR-IP est composé de plusieurs types de nœuds :

OLSR-GW un routeur qui permet à OLSR-IP de se connecter à Internet et intègre un agent étranger pour gérer la mobilité des nœuds mobiles.

OLSR-BS un point d'accès qui possède deux interfaces : sans fil et filaire.

OLSR-N un nœud mobile du réseau ad hoc implémentant Mobile IP.

OLSR-W un nœud filaire qui supporte un protocole de micro-mobilité.

La découverte d'un agent étranger est possible par la méthode proactive ou réactive. L'adressage utilisé par un nœud mobile n'est pas explicité clairement. Il semble que les

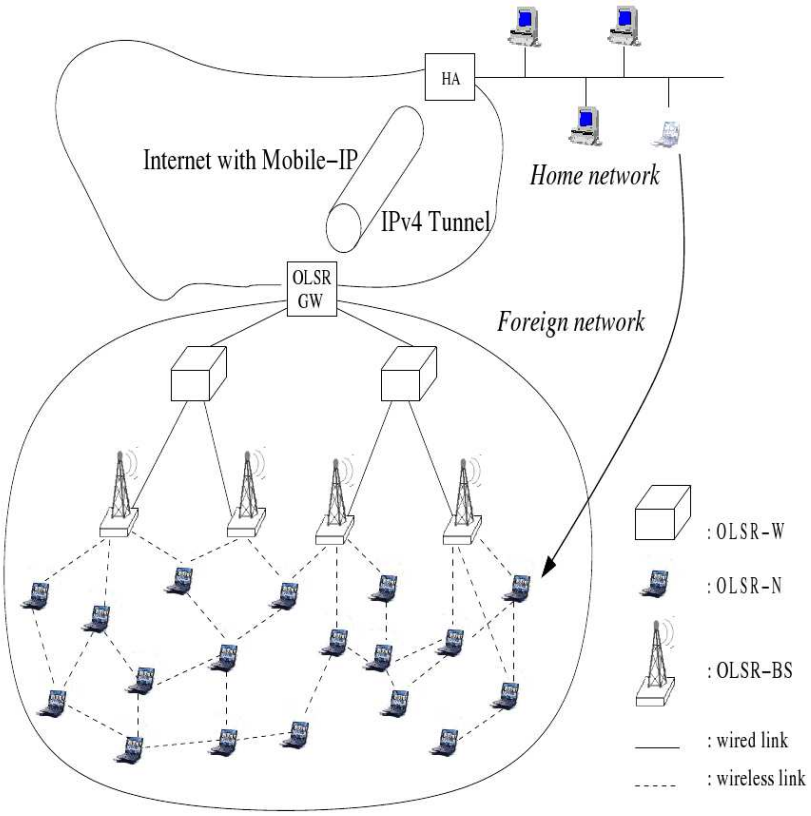


Figure 2.11 – Intégration d'OLSR à Mobile IP. (*Issu de* [28])

nœuds diffusent à leurs voisins leur adresse mère, et donc le routage ad hoc se ferait à l'aide de cette adresse. L'adresse de la route par défaut d'un nœud mobile est le OLSR-GW avec lequel il est associé. Aucun tunnel n'est donc utilisé entre le nœud mobile et l'agent étranger situé dans OLSR-GW.

En parallèle, un groupe de travail de l'IETF, dirigé par Ryuji Wakikawa, a publié à partir de 2001 plusieurs versions de drafts sur la *Global Connectivity for IPv6 Mobile Ad Hoc Networks* [110]. Le document décrit comment fournir une connectivité à Internet avec un réseau ad hoc mobile. En particulier comment un nœud mobile obtient une adresse de la passerelle et communique avec le point d'accès. Trois pré-requis sont nécessaires : le réseau ad hoc doit être capable de router des adresses IPv6, il existe un point d'accès connecté à Internet quelque part dans le réseau ad hoc, et tout nœud du réseau a ou a acquis une adresse IPv6. Il est possible pour un nœud d'utiliser son adresse IP mère de Mobile IP. La découverte de point d'accès se fait de manière proactive ou réactive. Deux propositions d'implémentation sont présentées : modification du standard Neighbor Discovery Protocol (NDP) [90], ou bien modification des messages d'un protocole ad hoc existant. Ces modifications peuvent être effectuées sur n'importe quel protocole ad hoc. Quelque soit le choix adopté, le document introduit trois messages :

IGWADV diffusé sur le réseau périodiquement par le point d'accès pour s'annoncer. Il contient, entre autres, un nombre de sauts limite (équivalent au TTL), un flag indiquant si le point d'accès exige un acquittement, et la durée de vie de la route vers le point d'accès.

IGWSOL utilisé pour la découverte réactive. Le nœud mobile diffuse sur le réseau ce message pour demander une route vers Internet. Le point d'accès répond par un message IGWADV.

IGWCON envoyé par le nœud mobile seulement si le point d'accès exige un acquittement. Ce message est utilisé pour gérer les nœuds associés à un point d'accès. C'est similaire à un enregistrement d'un nœud au point d'accès.

Sur réception d'un IGWADV, le nœud mobile ajoute le point d'accès à sa liste de points d'accès. La méthode de transmission utilisée est celle de la route par défaut. Le document décrit la modification des messages d'AODV (puis DYMO dans la dernière version du document), et d'OLSR. La dernière version du document a expiré en 2006 et aucun RFC n'a découlé directement de ce brouillon. Cependant, le brouillon de DYMO [36] (encore actif à ce jour) inclut une méthode simple *d'attachement à Internet* par l'intermédiaire d'un routeur spécifique appelé IDR.

2.2.4.3 Extension de protocoles de micro-mobilité

Au lieu d'intégrer un protocole ad hoc à Mobile IP, d'autres travaux [124, 122] ont plutôt pour objectif d'étendre un protocole de micro-mobilité de Mobile IP à des communications multi-sauts. Étendre un protocole de micro-mobilité permet d'une part la connectivité à Internet, et d'autre part permet de gérer les handovers rapides entre les points d'accès et d'atteindre un point d'accès en plusieurs sauts. Le principe est à la fois d'adapter les messages de signalisation des protocoles de micro-mobilité à une communication multi-hop, mais également d'utiliser les principes des protocoles ad hoc pour découvrir les points d'accès à plusieurs sauts.

Dans sa thèse [122], Ville Typpo propose d'associer un protocole de micro-mobilité comme HAWAII ou CIP, à un protocole ad hoc réactif comme AODV ou DSR. L'auteur choisit arbitrairement d'utiliser les versions IPV6 de CIP et AODV. La Figure 2.12 montre le processus pour créer une route entre le nœud mobile et le nœud correspondant. Le message RREQ de AODV est étendu pour découvrir les points d'accès accessibles, il est nommé RREQ ALL-BS. C'est donc une découverte de point d'accès réactive. Sur réception de ce message, par un point d'accès, il répond au nœud mobile par un RREP contenant le message beacon de CIP. Pour s'enregistrer auprès du point d'accès choisi, le nœud mobile lui envoie un GRAT RREP (Gratuitous Route Reply). Le nœud mobile doit ensuite actualiser sa route auprès des routeurs CIP en envoyant un Route Update au réseau d'accès. Le message route update est relayé par d'autres nœuds mobiles pour atteindre le réseau d'accès et donc les routeurs de CIP.

Dans [124, 125], les auteurs présentent le protocole Multi-hop Cellular IP (MCIP) qui a une approche cellulaire. Il n'utilise pas de protocoles ad hoc particuliers existants, mais adapte CIP pour que les nœuds mobiles puissent atteindre un point d'accès en plusieurs sauts. MCIP utilise deux caches : *multi-hop routing cache* pour maintenir la localisation du nœud mobile et *multi-hop paging cache* pour garder la route vers le nœud mobile. Un point d'accès maintient

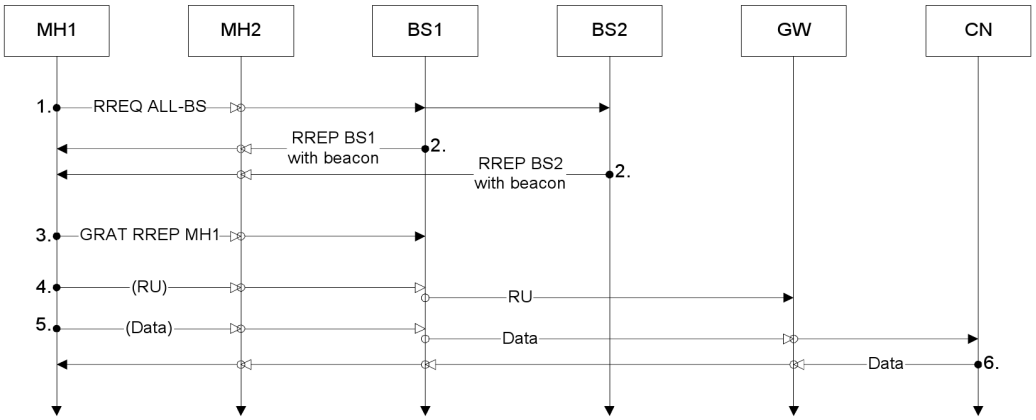


Figure 2.12 – Aquisition d'une route entre le nœud mobile et le nœud correspondant. (*Issu de* [122])

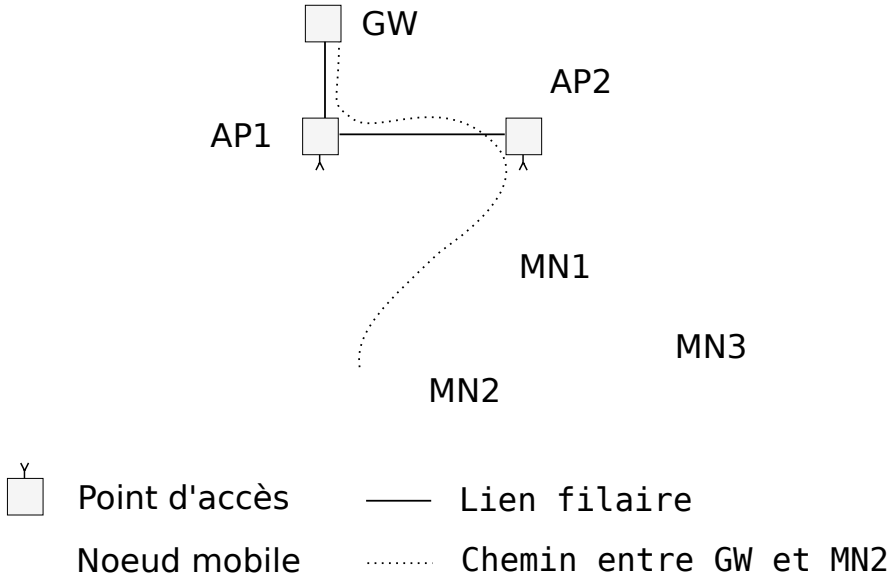


Figure 2.13 – Communication multi-hop dans MCIP

un paging cache pour chaque nœud mobile auquel il est associé et maintient un routing cache pour chaque nœud mobile actif, c'est-à-dire qui transmet des données. Comme le montre la Figure 2.13, quand MN2 est inactif le multi-hop paging cache correspondant dans la passerelle (GW) a comme chemin GW-AP1-AP2-MN2. Par contre quand MN2 passe en mode actif, MN2 ou AP2 initie une découverte de chemin pour construire la route entre MN2 et AP2. Une fois le processus de découverte de chemin terminé, la route GW-AP1-AP2-MN1-MN2 est ajoutée au multi-hop routing cache. La découverte de chemin correspond à une découverte de point d'accès réactive. MCIP permet à un point d'accès d'initier une recherche d'un nœud mobile par un processus similaire.

2.2.4.4 Réseau cellulaire multi-hop

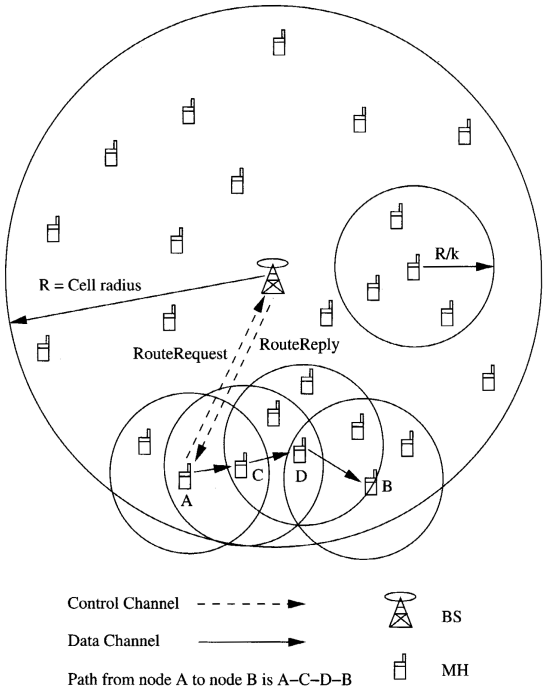


Figure 2.14 – Chemin en plusieurs sauts pour atteindre le point d'accès. (*Issu de* [15])

Il existe des protocoles pour réseau ad hoc hybride qui n'ont pas pour objectif une intégration à Mobile IP. Multi-hop cellular networks (MCN) [15] est une architecture cellulaire où une connexion entre la source et la destination est établie à travers un chemin à plusieurs sauts (Figure 2.14). La couverture radio du point d'accès est plus grande d'un facteur k que la couverture radio d'un nœud mobile. Les auteurs proposent un $k = 2$.

Lorsqu'un nœud mobile souhaite connaître le chemin multi-saut vers un autre nœud, il envoie directement un RouteRequest au point d'accès. Le point d'accès calcule le plus court chemin avec l'algorithme Dijkstra, et l'envoie au nœud mobile avec RouteReply. L'inconvénient de cette architecture est l'hypothèse qu'un nœud mobile puisse communiquer avec le point d'accès en un seul saut. Cela suppose l'utilisation de deux interfaces radio : une

pour transmettre les données entre les nœuds mobiles et une pour communiquer avec le point d'accès.

Pour résoudre ce problème, une architecture appelée Single-interface Multi-hop Cellular Network (SMCN) [114] est proposée. Le protocole associé, SMCN Routing Protocole (SMRP) permet à un nœud mobile de découvrir les points d'accès les plus proches et s'y enregistrer. Les messages RouteRequet et RouteReply sont également transmis vers le point d'accès en plusieurs sauts. En plus des messages RouteRequet et RouteReply, SMRP utilise les messages Registration Request, Registration Acknowledgement, Beacon et Neighbor Update. Le message Beacon, est une balise diffusée sur le réseau ad hoc envoyée par les points d'accès pour s'annoncer. La diffusion de ce message est limitée par un TTL. Quand un nœud reçoit un Beacon, il enregistre le point d'accès dans la table de points d'accès accessibles. Le message Registration Request permet à un nœud mobile de s'enregistrer auprès du point d'accès. Le point d'accès confirme l'enregistrement en envoyant un message Registration Acknowledgement avec l'inverse du chemin suivi par le message Registration Request. Le message Neighbor Update est envoyé au point d'accès par un nœud mobile lorsque son voisinage a changé. La découverte du voisinage se fait lorsqu'un nœud mobile reçoit un message Beacon de ses voisins.

Le réseau ad hoc hybride est avant tout un compromis entre le coût d'une couverture totale (comme un réseau cellulaire classique) et les performances en terme de capacité et connexité (problèmes d'un réseau ad hoc classique) du réseau. La mise en place d'un réseau ad hoc hybride doit prendre en considération ces deux aspects : quelles sont les performances apportées par l'ajout de points d'accès par rapport au coût que cela entraine ? Le choix de l'architecture et du protocole ad hoc hybride doit aussi prendre en compte la dynamicité, la topologie, et les applications du réseau ; la mise en œuvre de l'ensemble des mécanismes décrits précédemment, indispensables à un réseau ad hoc hybride, impactant sur les performances globales.

Nous allons donc présenter les réseaux ad hoc de véhicules, notre cas d'étude d'un réseau fortement dynamique, avant de discuter sur l'intégration des réseaux ad hoc hybrides dans un contexte de réseaux de véhicules sur autoroute.

2.3 Les réseaux de véhicules

Les réseaux de véhicules, ou VANET pour Vehicular Ad Hoc Networks en anglais, sont une technologie émergente intégrant les dernières techniques de communication. Chaque nœud du réseau est un véhicule équipé d'une ou plusieurs interfaces radio sans fil, les véhicules communiquent être-eux grâce à cet équipement. Un réseau de véhicules fournit [80] (1) le long de la route, une connectivité au monde extérieur par l'intermédiaire de passerelles vers d'autres réseaux, et (2) une *communication inter-véhiculaire* pour les *véhicules intelligents*, ou ITS pour *Inteligeant Transportation Systems* en anglais. Sans infrastructure, le réseau est un réseau ad hoc, un protocole doit donc être utilisé pour assurer les communications inter-véhiculaires. Les réseaux de véhicules sont aussi appelés IVC, pour *Inter-Vehicule Communication* ou IVCS pour *Inter-Vehicular Communication Systems*.

La recherche sur les réseaux de véhicules ou les communications inter-véhiculaires a commencé au Japon au début des années 1980 par la JSK (Association of Electronic Technology for Automobile Traffic and Driving). Plus tard, California PATH [63] et Chauffeur [55] ont présenté des techniques permettant de relier deux véhicules ou plus, pour former un convoi. Récemment, le projet Européen CarTalk 2000 [107] tente de résoudre des problèmes liés à la sécurité ou au confort des passagers par l'intermédiaire de communications inter-véhiculaires. Depuis 2002, avec le développement rapide de technologies sans fil, le nombre de publications dans le domaine des réseaux de véhicules a rapidement augmenté. Dans cette dynamique, divers workshops ont été créés, comme *ACM International Workshop on Vehicular Ad Hoc Network* en 2004 ou *International Workshop on Inteligent Transportation* en 2003.

Les véhicules intelligents sont la principale application des réseaux de véhicules, notamment pour augmenter la sécurité ou le confort des passagers. Les fonctionnalités sont [80] la surveillance du trafic, le contrôle du trafic, la visibilité augmentée des carrefours dangereux, la détection de collisions, les services d'information de proximité, le calcul de trajet en temps réel selon le trafic. D'autres applications, en dehors des transports intelligents, permet de fournir la connectivité à Internet, ou toute communication entre véhicules, comme le jeux, tchatche, ou échange de fichiers.

2.3.1 Architectures et caractéristiques des réseaux de véhicules

2.3.1.1 Architectures du réseau

Comme pour les réseaux sans fil en général, il existe différentes architectures pour les réseaux de véhicules. Comme le montre la Figure 2.15, il existe trois catégories d'architectures différentes pour les réseaux de véhicules [80] : cellulaire/WLAN , ad hoc et hybride.

Les réseaux de véhicules cellulaires ou WLAN peuvent utiliser des points d'accès fixes couvrant la totalité d'un tronçon de route ou certains points stratégiques comme les intersections pour se connecter à Internet ou récupérer des informations sur le trafic. Le réseau de véhicules peut combiner à la fois l'utilisation de points d'accès WLAN et un réseau cellulaire existant, comme un réseau cellulaire (Figure 2.15 (a)) de troisième génération, dit 3G.

L'installation de points d'accès le long d'une route permet la connectivité des véhicules,

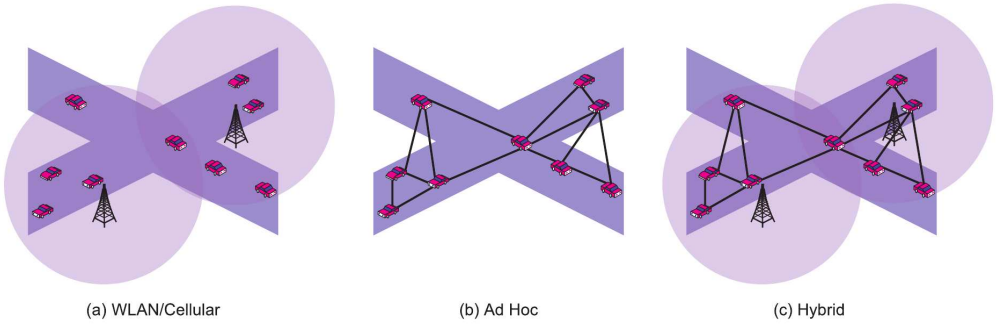


Figure 2.15 – Trois catégories d'architectures pour les réseaux de véhicules. (*Issu de* [80])

mais elle est parfois non faisable pour des raisons de coûts. L'utilisation d'un réseau ad hoc avec des communications inter-véhiculaires est alors nécessaire (Figure 2.15 (b)).

Enfin, la troisième architecture est hybride combinant à la fois points d'accès et réseau ad hoc (Figure 2.15 (c)). Cette architecture est similaire à celle que nous avons présentée en section 2.2.

2.3.1.2 Caractéristiques des réseaux de véhicules

Un réseau ad hoc de véhicules possède des caractéristiques particulières par rapport à un réseau ad hoc classique. En plus du fait que les nœuds sont des routeurs pour les autres nœuds, que le réseau soit auto-organisé, les réseaux ad hoc de véhicules ont les propriétés suivantes [80] :

Environnement varié Les réseaux ad hoc de véhicules peuvent fonctionner dans trois types d'environnement : autoroutier, urbain, rural. Les caractéristiques de ces environnements seront détaillées par la suite.

Forte dynamique de la topologie La vitesse élevée du déplacement des véhicules entrainent des changements de topologie fréquents du réseau ad hoc. En effet, si on considère deux véhicules roulant en sens opposé à 25 m/s (90 km/h), avec une couverture radio de 250 m, alors la durée de la communication directe entre les véhicules est seulement de 10 secondes.

Faible connectivité du réseau La densité d'un réseau de véhicules est très variable. Une forte densité de véhicules permet au réseau d'être connexe, et donc il existe toujours un chemin entre deux nœuds qui souhaitent communiquer. A l'inverse, une faible densité de véhicules faible a pour conséquence un taux élevé de ruptures de communications, un délai d'acheminement plus long si le véhicule conserve le paquet, voire une impossibilité pour deux véhicules de communiquer.

Énergie et stockage suffisant Un véhicule produit sa propre énergie électrique en roulant et possède une batterie de grande capacité comparativement à l'énergie nécessaire à un système électronique de communication. Cette batterie permet de stocker cette énergie quand le véhicule est à l'arrêt.

Mobilité prévisible Un véhicule est contraint de suivre la trajectoire de la route. Son déplacement est alors prévisible, d'autant plus si le système possède un plan du réseau routier. Cette mobilité prévisible permet de prévoir la position d'un véhicule dans le futur.

Information de localisation disponible Actuellement, la majorité des véhicules neufs possède un système de géolocalisation que nous avons présenté dans le sous-paragraphe 2.1.3.1. Pour un coût quasiment nul, le système peut obtenir la position géographique du véhicule, ainsi que sa direction, s'il est en mouvement. Cette information est utilisée par les protocoles ad hoc géographiques (§ 2.1.3) mais également pour tous les services disponibles aux passagers où la localisation est indispensable.

2.3.1.3 Environnements routiers

Comme on l'a mentionné précédemment, nous recensons trois types d'environnement où fonctionnent les réseaux ad hoc de véhicules : autoroutier, urbain, rural.

Le contexte urbain a pour particularité une densité de véhicules importante. Mais la présence de nombreux bâtiments ou d'arbres perturbent la transmission radio et donc il n'existe pas toujours une communication en ligne directe entre deux véhicules séparés par une distance inférieure à la couverture radio. Dans les villes américaines, on remarque que le réseau routier est bien souvent très proche d'une grille.

Dans un contexte rural, le réseau routier est peu dense, et n'a pas de caractéristique géométrique particulière. Les obstacles sont moins présents qu'en environnement urbain. En France, la majorité du réseau routier est en contexte rural. Malheureusement, c'est le contexte le plus difficile : faible densité de véhicules, donc peu de connectivité, et couverture totale du réseau routier non envisageable.

Le contexte autoroutier est particulier. Le réseau autoroutier est par nature adapté à la circulation de véhicules à grande vitesse. Les conséquences sont une route proche d'une ligne, au moins deux voies de circulation par sens de circulation, terre-plein central, infrastructure liée à la sécurité renforcée. Les véhicules se déplacent à grande vitesse dans deux directions opposées. Le réseau de véhicules est alors un réseau fortement dynamique mais est considéré à une seule dimension, il s'apparente alors à une ligne.

2.3.2 Protocoles utilisés dans les réseaux de véhicules

Le choix d'un protocole adapté est primordial car les performances du réseau en dépendent directement. Le choix d'un protocole dépend également de l'application au niveau utilisateur. En effet, si l'application est d'avertir la survenue d'un accident, on privilégiera le délai. Par contre, si l'application concerne le transfert de fichiers, la garantie d'un débit suffisant pour l'utilisateur est privilégiée. Le choix du protocole doit être également adapté au type de communication. Deux types de communications existent dans les réseaux de véhicules : les communications inter-véhiculaires (véhicules à véhicules ou V2V) ou véhicules-infrastructure (V2I). A celles-ci peuvent s'ajouter les communications intra-véhicules (directement dans le véhicule) [19]. Pour les communications inter-véhicules, un protocole ad hoc classique est adapté, bien que des optimisations soient toujours possibles. Par contre les com-

munications véhicules-infrastructure implique l'utilisation de protocoles ad hoc hybrides [24, 25].

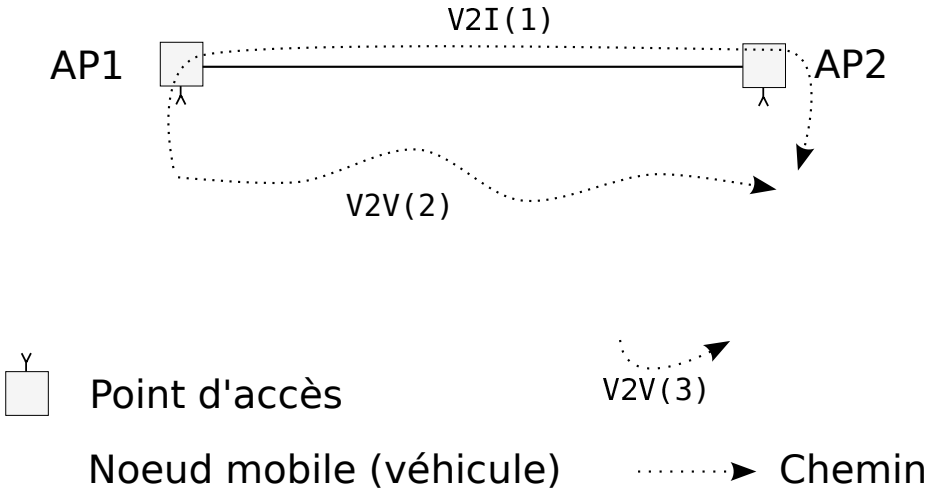


Figure 2.16 – Choix du type de communication pour optimiser la longueur de la route.

Le choix du type de communication permet d'optimiser le chemin vers le nœud destinataire. Sur la Figure 2.16, la longueur du chemin (1) en passant uniquement pas des nœuds mobiles est plus longue que la longueur du chemin en passant par des points d'accès (2). En revanche, si le nœud destinataire est proche de la source le chemin sera plus court en passant par un nœud intermédiaire (3). Dans [19], une solution est proposée pour choisir le mode de communication optimal.

Les réseaux de véhicules peuvent fonctionner à un saut. Dans ce cas, soit le véhicule avec qui il communique doit être dans son voisinage (communication inter-véhicule), soit le véhicule est connecté à un point d'accès pour communiquer avec un nœud distant (communication véhicule-infrastructure). Les communications sans fil à un saut de type véhicule-infrastructure sont déjà utilisées avec un équipement d'aide à la navigation pour obtenir les informations sur le trafic en temps réel. La communication véhicule-infrastructure est basée sur le réseau cellulaire existant.

La caractéristique des réseaux ad hoc (communications à plusieurs sauts) est adaptée pour les réseaux de véhicules, d'une part, le routage est robuste au changement de topologie fréquent, d'autre part, cette robustesse est réalisée à faible coût (pas d'infrastructure nécessaire). Pour ces raisons, à ce jour, la majorité des recherches sur les réseaux de véhicules utilisent un protocole ad hoc. Les protocoles proactifs ont montré leurs bonnes propriétés pour les réseaux statiques ou quasi-statiques [44]. Mais les réseaux de véhicules étant fortement dynamiques, les protocoles réactifs ou géographiques sont mieux adaptés aux réseaux dynamiques donc aux réseaux de véhicules. Les protocoles topologiques réactifs ont donc d'abord été retenus. Cependant, depuis quelques années, les protocoles géographiques (basés sur la position géographique) semblent être prometteurs pour les réseaux de véhicules ; des études montrant que les protocoles géographiques sont plus performants en tout point que les protocoles réactifs [89]. Il est d'autant plus intéressant d'utiliser un

protocole géographique pour les communications dans les réseaux de véhicules, qu'une part importante non négligeable de véhicules possèdent un système de géo-localisation et qu'il est possible pour un coût supplémentaire négligeable d'en installer dans chaque véhicule "communiquant'. Néanmoins, il existe des études récentes qui soutiennent qu'un protocole géographique n'est pas plus performant qu'un protocole topologique dans un contexte de réseau de véhicules sur autoroute [86, 85].

Plusieurs auteurs ont adapté les protocoles ad hoc géographiques (Greedy) afin de prendre en compte les spécificités des réseaux de véhicules. Des dérivés des protocoles qui ont pour but d'améliorer la diffusion des paquets et des messages dans un réseau de véhicules existent également. GPCR (Greedy Perimeter Coordinator Routing) [83] et CAR (Connectivity-Aware Routing) [91] sont des exemples de ces protocoles. D'autres protocoles tentent d'améliorer les performances des protocoles géographiques en utilisant une carte numérique des routes d'une ville. Ainsi, la carte permet d'obtenir des informations sur la topologie des rues. Cette fonctionnalité permet au nœud source de calculer la liste des jonctions de chemins possibles pour atteindre la destination. Afin d'atteindre chaque jonction, le protocole géographique classique est utilisé le long de la rue. GSR (Geographic Source Routing) [82], A-Star (Anchorbased Street and Traffic Aware Routing) [113] ou GyTar (Greedy Traffic Aware Routing) [66] sont des exemples de protocoles basés sur la position géographique utilisant une carte. En plus d'utiliser une carte, A-Star utilise les informations sur les véhicules parcourant un même trajet régulièrement, comme les transports publics d'une ville.

L'ensemble de ces protocoles fonctionnent seulement s'il existe un chemin de bout en bout de la source à la destination au moment d'envoyer le paquet. Dans le cas contraire, les paquets ne peuvent pas être acheminés. Ainsi, pour résoudre ce problème, une méthode appelée *enregistrer-transporter-transmettre* (store-carry-forward) est utilisée. Cette méthode permet à un véhicule de conserver le paquet s'il y a aucun voisin qui améliore son déplacement vers la destination. Quand un voisin adéquat est trouvé, alors le paquet lui est transmis. Ce principe est utilisé également dans les réseaux appelés *Delay Tolerant Network* (DTN). SAR (Spatially Aware Routing) [119] et VADD (Vehicle-Assisted Data Delivery) [129] utilise la méthode enregistrer-transporter-transmettre. D'autres protocoles se basent sur la trajectoire des véhicules voisins pour décider de transmettre le message. Le véhicule transmet le paquet aux voisins qui ont une meilleure trajectoire que celui qui porte le paquet. Les critères sur le choix de la meilleure trajectoire dépendent des considérations prisent par les protocoles. Ainsi, les protocoles comme GeOpps [79] essayent d'atteindre la destination le plus rapidement possible, alors que Move [78] prend en compte le véhicule qui avance le mieux vers la destination tout en considérant le vecteur vitesses des véhicules.

Une autre approche du routage est introduite dans [46] pour les réseaux de véhicules. Cette approche se base sur la transmission conditionnelle. Un nœud transmet un message si et seulement si les conditions de transmission définies dans le message sont satisfaites. Les conditions peuvent être des informations géographiques (distance de la source, position géographique, aire ...), information liée au temps (délai depuis l'émission, date, durée du message ...), information sur la trajectoire, identité du message (source, destination, relai). Les avantages de cette approche est la résistance à la forte dynamicité du réseau.

Malgré l'utilisation de protocoles ad hoc généralistes, géographiques ou réactifs, soit possible dans un réseau ad hoc de véhicules, l'utilisation de protocoles spécifiques aux réseaux

de véhicules semblent privilégiée. Les protocoles doivent être avant tout choisis par rapport à l'application fonctionnant sur le réseau. Par exemple, un protocole DTN ne permet pas une communication de type voix sur IP ou de tchatche, alors qu'il est parfaitement adapté à des applications pour l'envoi de signaux d'alertes (accidents, bouchons, etc.).

2.3.3 Standards des réseaux ad hoc de véhicules

Depuis la forte progression de la recherche dans le domaine de réseaux de véhicules, les grands organismes de standardisation et les constructeurs automobiles ont entrepris la conception de standards pour les réseaux de véhicules. La standardisation des protocoles ou d'architecture des réseaux de véhicules est d'autant plus importante qu'un réseau ad hoc est autonome.

2.3.3.1 IEEE WAVE (802.11p)

La norme IEEE 802.11, dite WiFi, s'est imposée pour les premières expérimentations et les simulations des réseaux de véhicules. Beaucoup d'études utilisent les versions b ou g, car très répandues et fabriquées à bas coût. Néanmoins, une version *p* spécialisée pour les réseaux de véhicules est en cours de standardisation [13]. Un canal de communication sans fil, appelé DSRC pour *Dedicated short-range communications*, fonctionnant dans la bande des 5.9 GHz aux Etat-Unis ou les 5.8 GHz en Europe et au Japon, a été spécifiquement conçu pour les communications véhiculaires. Par exemple, le système de télépéage sur autoroute utilise la norme DSRC.

La norme IEEE 802.11p est une norme pour la *couche d'accès au médium*, couche MAC, se basant principalement sur la norme 802.11a et fonctionnant dans la bande de fréquence DSRC.

2.3.3.2 IETF : Mobile IP

Les standards de l'IETF *Mobile* IP [100, 68] sont également un domaine de recherche appliqué aux réseaux de véhicules. En effet, Mobile IP une solution pour gérer la mobilité sur l'internet par l'intermédiaire d'un réseau filaire mais aussi par l'intermédiaire d'un réseau sans fil. Les réseaux de véhicules peuvent être vus comme un réseau mobile, on peut donc utiliser NEMO [18]. En effet, on peut considérer un bus, un train, un avion ou autre moyen de transport collectif, comme un réseau où il existe plusieurs terminaux, chacun possédant sa propre adresse IP publique [99]. On peut également considérer un réseau de capteurs déployés dans un véhicule [48].

2.3.3.3 ISO : Calm

Calm, pour *Communications, Air-interface, Long and Medium creaks* [1] , spécifie une architecture pour les réseaux de véhicules. Il est à la fois basé sur les protocoles de l'IETF pour la communication IP comme Mobile IP et NEMO, mais également sur les standards IEEE. Cette architecture permet principalement de rendre transparent l'utilisation de ces standards pour les applications et services des réseaux de véhicules.

2.3.3.4 Car-to-car communication consortium

Plusieurs fabricants de voitures se sont réunis [80], Audi, BMW, Daimler Chrysler, Fiat, Renault, Volkswagen pour former le Car-to-car communication consortium, C2CCC, afin d'améliorer l'efficacité et la sécurité du trafic routier par l'utilisation des réseaux de véhicules.

2.3.4 Projets existants

Plusieurs projets de recherches au niveau national ou international ont été lancés : aux USA avec VII , en Europe avec CVIS, au Japon avec SmartWay.

2.3.4.1 USA : Vehicle-Infrastructure Integration (VII)

Le but du projet Vehicle-Infrastructure Integration (VII) du département États-Unien des Transports est de fournir des équipements de communication interopérables entre les véhicules sur une route, équipés d'un équipement embarqué spécifique ; et une infrastructure le long de la route pour le développement de systèmes permettant d'améliorer la sécurité, l'efficacité et le confort des réseaux de transport. Il est basé sur l'utilisation de la norme IEEE 802.11p, développée conjointement avec le projet. VII relie différents acteurs concernés : industries, autorité des transports, et organisations professionnelles. Le projet a trois priorités : évaluation du modèle économique, validation des technologies utilisées et développement de structures légales et politiques permettant un succès sur le long terme. Le projet prévoit d'équiper 15 millions de nouveaux véhicules par an et de couvrir 70 % de toutes les intersections à feux tricolores dans 454 zones urbaines dans un rayon de 2 minutes de temps de trajet du centre ville. Plusieurs déploiements ont déjà été réalisés en Floride, Michigan et Californie, permettant d'évaluer les technologies utilisées.

2.3.4.2 Europe : European Commission's Cooperative Vehicle-Infrastructure System (CVIS)

Les objectifs de CVIS est :

- d'unifier les solutions techniques permettant aux véhicules et les infrastructures de communiquer entre-eux de manière continue et transparente en utilisant différents média de communications disponibles,
- définir et valider une architecture ouverte, un concept pour des systèmes coopératifs, développer des composants communs supportant le modèle de coopération dans les applications de la vie courante,
- d'étudier l'intégration avec les utilisateurs, la sécurité de données, la protection de la vie privées, l'ouverture et l'interopérabilité des systèmes, les risques et responsabilités, les politique publiques nécessaires, le modèle économique, le plan de déploiement.

Calm fait partie intégrante du projet CVIS. SafeSpot est un également un projet européen orienté sur la conception de système coopératif pour la sécurité sur route. Plusieurs projets Allemands existent également : Now, Car2Car, FleetNet. Lara (La Route Automatisée) est

un projet Français lancé récemment, plutôt orienté pour l'assistance à la conduite, voire le pilotage totalement automatisé du véhicule.

2.3.4.3 Japon : SmartWay

Le principal apport de SmartWay, basé au Tokyo Metropolitan Expressway est la réalisation de démonstrations utilisant un réseau de véhicules comme l'assistance à la conduite en temps réel, la délivrance de messages à l'intérieur du véhicule ou la communication à deux sens pour le paiement électronique (péage et autres). Au Japon, tous les nouveaux véhicules sont équipés d'un système de navigation intégrés et d'un composant appelé *Vehicle Information and Communication System* VICS. Les informations agrégés par le Japan Road Traffic Information Center sont transmis à cet équipement par liaison optique (IR) ou par radio sur la fréquence des 2.4 Ghz. SmartWay a également mis en avant que les 75 % de véhicules au Japon utilisant les autoroutes, sont équipés d'un Electronic Toll Collection (ETC) dont le fonctionnement est basé sur la bande de fréquence DSRC 5.8 Ghz. Cette équipement sert au télépéage.

Conclusion

Les réseaux ad hoc ont permis de répondre à des problématiques des réseaux sans fil quand la mobilité est trop importante ou lorsqu'une infrastructure fixe fait défaut. Malgré tout, des solutions hybrides sont développées en combinant réseau ad hoc et infrastructure pour la connectivité à Internet, augmenter la connexité du réseaux ad hoc ou étendre la portée d'un point d'accès.

Le réseau de véhicules est une des multiples applications des réseaux ad hoc, et permet aux véhicules de communiquer entre-eux afin d'augmenter la sécurité et le confort des passagers. Les réseaux de véhicules sont un domaine d'étude à part entière à cause des caractéristiques particulières des réseaux ad hoc qu'ils forment, dont la forte dynamique et la géométrie de la topologie. L'étude de la gestion de la mobilité des protocoles ad hoc dans un contexte de réseau véhiculaire est alors indispensable afin d'en améliorer les performances. Nous allons par la suite développer une étude sur passage à l'échelle d'un réseau de véhicules sur autoroute. Mais avant étudions les différentes méthodes d'évaluation des réseaux ad hoc.

Chapitre 3

Mesures de performances des réseaux ad hoc mobiles par simulation

L'évaluation des protocoles développés pour les réseaux est indispensable avant de pouvoir les déployer dans les routeurs et terminaux d'un réseau réel. Cette évaluation est nécessaire à la fois pour vérifier un fonctionnement correct d'un protocole dans divers scénarios, mais surtout pour mesurer ses performances et les comparer à d'autres protocoles existants.

Les performances d'un protocole sont très sensibles aux changements rapides de la topologie du réseau, comme dans un réseau fortement dynamique. Ils le sont également à la taille du réseau. L'évaluation d'un protocole dans un contexte de forte dynamicité et de grand réseau, tel qu'un réseau de véhicules, est primordiale pour choisir le protocole le mieux adapté et en améliorer ses performances.

La simulation est couramment utilisée pour mesurer les performances d'un réseau informatique, et plus particulièrement les performances d'un protocole. Elle est nécessaire quand l'expérimentation est trop coûteuse et l'étude théorique trop complexe ou quand on souhaite valider des hypothèses. Notre cas d'étude de réseau fortement dynamique sont les réseaux de véhicules sur autoroute. Ce type de réseau est très étendu et leur dynamique complexe, il est donc typiquement évalué à l'aide de simulateur de réseau, bien qu'une étude sur un point particulier peut être l'objet d'une étude théorique.

Dans ce chapitre, nous introduirons les principes globaux de la simulation et les différents simulateurs existants pour l'évaluation de protocoles ad hoc, ainsi que les modèles de mobilité associés. Le simulateur utilisé dans nos travaux est ensuite détaillé avec les modifications et extensions que nous y avons apportées. Enfin nous donnerons les métriques utilisées pour comparer et évaluer les performances des protocoles ad hoc que nous évaluerons dans les chapitres suivants.

3.1 Les simulateurs de réseaux

La simulation est une technique de modélisation du monde réel, et elle permet de représenter le fonctionnement d'un système que l'on veut observer. La modélisation de ce système consiste à répertorier plusieurs grandeurs intéressantes, que nous appelons *variable*. On définit alors l'*état* d'un système comme l'ensemble des valeurs que prennent ces variables à un instant donné.

3.1.1 Systèmes de simulation

On divise les systèmes de simulation en deux catégories [47] : les systèmes discrets et les systèmes continus.

Un *système discret* est un système dans lequel les états ne changent de valeur qu'en nombre fini de points sur l'axe du temps (événements). La modélisation des attentes aux caisses d'un magasin est un système discret, car le nombre de clients ne se modifie qu'à l'arrivée ou au départ d'un client.

Un *système continue* est un système dans lequel le temps s'écoule de façon continue et où les variables peuvent changer de valeur à tout instant. La simulation du vol d'un avion est un système continue car les coordonnées et la vitesse de l'avion sont des fonctions qui prennent des valeurs en tout point de l'axe du temps.

Dans un modèle continu, souvent décrit par un système d'équations différentielles, le temps est discrétisé selon un pas donné et, à chaque fois que le temps avance, les valeurs des variables du systèmes sont mises à jour.

Dans un réseau informatique, un nœud change typiquement d'état quand il reçoit un message ou quant un de ses timers arrive à expiration. On peut aisément caractériser à quel moment ces événements doivent se produire, c'est pourquoi nous utilisons un simulateur à état discret pour la simulation de réseaux.

3.1.2 Simulateur par événement discret

Un simulateur à événement discret est une suite d'événements programmés à un instant de la simulation. Il possède quatre composants principaux [31] :

- Une horloge correspondant à l'*heure de simulation* et indépendante de l'*heure réelle* du système,
- Une liste d'événements. Un événement est défini comme le moment où il doit se produire un processus à exécuter.
- Une condition d'arrêt, par exemple un événement particulier ou à un temps de simulation donné.
- Statistiques. Le simulateur produit des statistiques sur l'exécution de la simulation. Ces statistiques permettent dévaluer l'exécution de la simulation, notamment en terme de performances. Typiquement, dans le domaine des réseaux, les statistiques sont les messages reçus et envoyés.

La liste des événements et leurs paramètres sont générés par un programme modélisant le *scénario* de simulation. Le cœur du simulateur inclut généralement un moteur de simulation à événements discrets. Ce type de moteur permet d'obtenir un ordonnancement dynamique des événements. Lorsqu'un événement de la liste est généré, l'échéancier exécute le processus correspondant.

L'exécution d'un processus est susceptible de générer à leur tour des événements. La simulation continue tant que des événements sont générés. Le système est dit stable lorsqu'aucun événement n'est en attente au temps courant, et le simulateur peut continuer en incrémentant l'heure de simulation. Un échéancier dynamique produit un ordonnancement au fur et à mesure de la simulation.

En revanche, un échéancier est dit *statique* s'il ne permet pas d'ajouter des événements en cours de simulation. Il permet cependant de calculer un ordre optimal d'activation des événements en début de simulation.

3.1.3 Simulateur de réseau sans fil

Nous nous intéressons aux réseaux ad hoc, cas particulier des réseaux sans fil. La simulation de réseau ad hoc correspond à la simulation d'un réseau sans fil avec les protocoles nécessaires à sa mise en œuvre. Ces protocoles sont au niveau de la couche MAC (mode ad hoc du WiFi) et de la couche routage (protocoles de routage). [64] présente un état de l'art exhaustif sur les simulateurs existants (en 2006) pour les réseaux ad hoc.

Contrairement aux réseaux filaires, les nœuds d'un réseau sans fil sont mobiles. Il existe alors, en plus d'une dimension temporelle, une dimension spatiale à simuler comprenant deux composantes : un modèle de *mobilité* des nœuds et un modèle de propagation du signal. Nous détaillerons par la suite les différents modèles de mobilité. Le modèle de propagation du signal permet de déterminer si deux nœuds peuvent s'échanger des messages. Plusieurs types de modèles plus ou moins fins existent.

Un modèle de propagation simple est basé sur un rayon de couverture d'un nœud R, donné soit en entrée de simulation ou soit calculé à l'aide de paramètres physiques : fréquences, puissance d'émission, seuil de réception, etc. Dans ce modèle, deux nœuds peuvent s'échanger des messages si et seulement si la distance les séparant est inférieure à R.

D'autres modèles plus fins, prennent en compte la qualité du signal en fonction de sa distance parcourue. Dans ce modèle, deux nœuds peuvent s'échanger des messages avec une certaine probabilité. Plus les nœuds sont éloignés l'un de l'autre, plus la probabilité de pouvoir s'échanger un message est faible.

Enfin, des modèles peuvent également prendre en compte des interférences produites par des objets physiques situés dans l'environnement des nœuds.

3.1.4 Critères de choix d'un simulateur

Il existe une multitude de simulateurs de réseaux, certains plus spécialisés, d'autres généralistes. Le choix du simulateur est basé sur plusieurs critères [64] :

Bibliothèque de modèles Typiquement les protocoles implémentés dans le simulateur. Si l'on souhaite utiliser un protocole déjà inclus dans la bibliothèque, il est alors inutile de l'implémenter.

Fiabilité du simulateur et des protocoles simulés La fiabilité des protocoles inclus dans le simulateur est primordiale pour rendre la mesure de performances d'un protocole la plus fidèle à la réalité.

Performances brutes Se mesurent en temps d'exécution et en utilisation de la mémoire. Si nous souhaitons simuler un réseau comportant un grand nombre de nœuds, le temps d'exécution doit rester raisonnable et la mémoire utilisée adaptée à la machine exécutant le simulateur.

Facilité d'extension La facilité d'ajout de nouveaux modèles au simulateur est primordiale pour en évaluer les performances.

Mesure de performances Certains simulateurs incluent la génération automatique de statistiques en fonction de différentes métriques.

Type de réseau Architecture (filaire ou ad hoc) ou ses applications.

Licence de distribution Définit les droits d'utilisation du logiciel, les droits de diffusion (duplication) et les droits de modification.

Tout simulateur n'est pas entièrement fidèle à la réalité, il existe toujours un niveau d'imprécision. Il existe trois raisons à cela [64].

Premièrement un simulateur dépend du niveau de modélisation, c'est-à-dire de sa granularité. Par exemple, le niveau de granularité est plus fin si on modélise la propagation d'une onde à l'aide d'un modèle physique au lieu d'utiliser un modèle simple de propagation ne considérant pas l'atténuation du signal. En général, plus le niveau de granularité est fin, plus le temps de calcul est important. Le critère de granularité doit être choisi selon les applications. Si la propagation de l'onde est déterminante dans l'application à simuler, alors le temps de calcul supplémentaire est probablement nécessaire.

Deuxièmement, les performances d'un simulateur dépendent du modèle de mobilité. Selon le scénario et le type d'application du réseau, le déplacement des nœuds diffère. Un réseau comportant une majorité de piétons n'aurait pas la même dynamique qu'un réseau de véhicules. Nous décrirons en détail le problème de mobilité dans le paragraphe 3.2.2.

Troisièmement, la fiabilité du simulateur dépend de la taille du réseau. Il existe un seuil en nombre de nœuds à partir duquel la simulation du réseau est moins fiable [64].

Les performances d'un simulateur en temps de calcul dépendent de plusieurs critères. Il existe plusieurs techniques pour optimiser les performances d'un simulateur à événements discrets et réduire la durée d'une simulation.

La première technique est le parallélisme et l'exécution distribuée du simulateur. Le parallélisme est possible si tout ou une partie du code du programme est parallélisable. La parallélisation d'un programme consiste en la possibilité d'exécuter plusieurs instructions du code du simulateur simultanément. Un tel programme profite de l'architecture de machines comportant plusieurs unités de calcul fonctionnant en parallèle. L'exécution distribuée d'un simulateur consiste à répartir les données de simulation ou répartir des parties du code du simulateur (ou les deux à la fois) sur plusieurs machines (station de travail). En effet, le simulateur peut exécuter la simulation entièrement avec des paramètres différents sur chaque

machine. Une partie de la simulation peut également être effectuée sur une machine, puis une autre partie simultanément sur une autre machine. Ces deux méthodes pouvant être combinées. L'exécution distribuée est la plus simple à mettre en œuvre car le code du simulateur n'a pas besoin d'être parallélisable, et facilement déployable sur un ensemble de PC standards (cluster), contrairement au parallélisme qui exige une machine multi-processeurs à mémoire partagée (SMP).

D'autres techniques d'optimisation, appelés *mise en scène de simulation*, consistent en l'utilisation d'une mémoire cache où est stocké le résultat de fonctions. Une fonction appelée avec les mêmes paramètres doit retourner le même résultat. Ainsi, lorsque la fonction est appelée dans le futur avec les mêmes paramètres, il n'est pas nécessaire de l'exécuter, mais de juste renvoyer les résultats stockés dans la mémoire cache. Dans la même catégorie d'optimisation, il est possible de réordonner des événements indépendants pour les présenter dans un ordre propice à l'architecture de la machine pouvant ainsi les exécuter plus rapidement.

La localisation spatiale des nœuds du réseau est aussi une partie du simulateur pouvant être optimisée. Une méthode appelé *bining* consiste en la division de la surface de simulation en structure de liste (pour les simulations dans une aire à 1 dimension), d'une grille (simulation dans une aire en 2D), d'un arbre (simulation dans une aire 3D). La méthode pour une aire 2D ou 3D s'appel aussi *flat-bining*.

Enfin, la *simulation hybride* utilise à la fois un simulateur et un modèle analytique. Une partie de la simulation est donc modélisée à l'aide d'un modèle analytique dont le temps de calcul est moindre que l'exécution d'un simulateur à événement discret.

3.1.5 La licence du simulateur

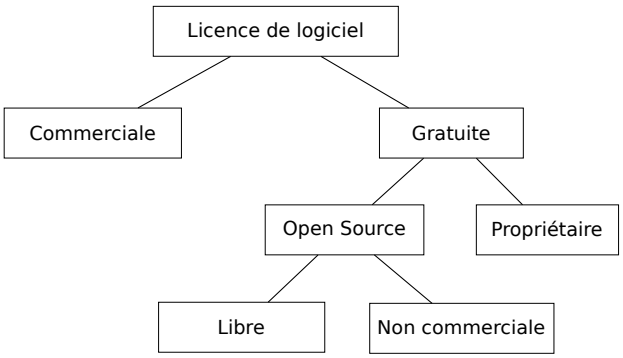


Figure 3.1 – Proposition de classement des licences de logiciels pour les simulateurs existants

La reproduction des résultats d'une simulation nécessite l'acquisition du simulateur avec lequel les résultats ont été produits et la possibilité d'améliorer le simulateur ou d'en avoir le contrôle total nécessite son code source. C'est la licence du simulateur qui définit les droits de diffusion et de modification du simulateur. En Figure 3.1 nous résumons les différentes licences des simulateurs existants.

Nom	Licence	Dernière version	Langage
GlomoSim	Non commerciale	Déc. 2000 (2.0)	Parsec
GTNets	Libre	Oct. 2008	C++
JiST/SWANS	Non commerciale	Mars 2005 (1.0.6)	Java
J-Sim	Non commerciale	Aout 2006 (0.6.0)	Java
ns-2	Non commerciale	Juin 2009 (2.34)	C++/OCTL
ns-3	Libre (GPLv2)	Jui. 2009 (3.5)	C++/Python
OMNet++	Non commerciale	Fev. 2009 (4.0)	C++
OPNet	Commercial	-	C
QualNet	Commercial	Mars 2008 (4.5)	Parsec

TABLE 3.1 – Liste des simulateurs existants

Certains simulateurs peuvent être utilisés sans acheter de licence, c'est le cas des logiciels gratuits. Dans le cas contraire, la licence est dite *commerciale*. Les logiciels gratuits se divisent en deux groupes : Open Source et propriétaire. Le logiciel gratuit propriétaire (Freeware) peut être diffusé sans restriction mais les sources du logiciel ne sont pas disponibles, la modification du logiciel n'est donc pas autorisée. Les logiciels Open Source peuvent être diffusés librement avec les sources. La copie et la modification sont donc autorisées. Par contre, la diffusion des modifications n'est pas toujours autorisée.

Dans le cas des simulateurs, nous avons recensé deux types de licences Open Source : libre et non commerciale. La licence libre est celle des logiciels libres (Free Software) ; elle permet : le droit d'utiliser, d'étudier, de modifier, de dupliquer, et de diffuser (donner et vendre). Les licences libres les plus connues sont GNU, BSD, Apache etc. La licence que nous appelons *non commerciale* permet dans un cadre d'utilisation non commerciale seulement, d'utiliser, modifier et dupliquer le logiciel.

3.1.6 Les simulateurs existants

Les simulateurs les plus utilisés dans le domaine des réseaux ad hoc sont récapitulés dans le Tableau 3.1. Les dernières versions correspondent aux versions disponibles en août 2009. Nous allons présenter sept simulateurs Open Source.

3.1.6.1 GlomoSim

GlomoSim [128] a été développé à UCLA (California, USA). GloMoSim est écrit en Parsec, dérivé du C. Parsec permet au programme d'être exécuté sur une architecture multiprocesseur à mémoire partagée. En conséquence le simulateur permet le parallélisme. GloMoSim respecte la sépération des couches du standard OSI. GlomoSim a une approche basée sur les messages pour la simulation à événement discrets. Chaque couche est représentée par un objet appelé *entité*. Un événement est représenté par un message daté géré par entité. Par contre, un nœud du réseau n'est pas une entité pour des raisons de performances. Par ses qualités (par rapport à ns-2), GlomoSim a été choisi comme base du simulateur commercial QualNet.

3.1.6.2 GTNets

Georgia Tech Network Simulator (GTNetS) [108] est basé sur la philosophie "l'environnement de simulation est structuré comme sont structurés les réseaux actuels". Concrètement, chaque couche protocolaire du réseau est clairement et distinctement séparée. GTNetS est organisé comme une liste de "Protocol Data Units" (PDU), qui peuvent être ajoutés ou supprimés et montés ou descendus dans la pile de protocoles. Un nœud peut avoir une ou plusieurs interfaces, chaque interface étant associée à une adresse IP et un lien du réseau. L'interface entre la couche transport et la couche application est réalisée par l'appel de fonctions POSIX standards pour la gestion des sockets comme send, connect, listen, etc. IEEE 802.11 et Bluetooth sont implémentés dans GTNetS. Une interface graphique pour les utilisateurs est également intégrée, permettant de visualiser la topologie du réseau, d'activer ou désactiver certains liens.

3.1.6.3 J-Sim

J-Sim [115] a été développé par l'Université de l'Ohio et de l'Illinois au USA. Il est inspiré de l'environnement de simulation développé avec le langage Simula. En effet, Simula est considéré comme le premier langage orienté objet et il intègre une bibliothèque de classes permettant la simulation à événements discrets. Il intègre également une bibliothèque de classes pour la génération de nombres aléatoires d'après une distribution uniforme, gaussienne, exponentielle ou booléenne. D'abord conçu pour les réseaux filaires, J-Sim a été étendu pour les réseaux sans fil avec le support de IEEE 802.11 [3].

3.1.6.4 ns-2

ns a commencé à être développé en 1989 comme une variante du simulateur de réseau nommé REAL. Son successeur, ns-2 [9], est de loin le simulateur le plus utilisé dans le domaine des réseaux. D'abord conçu pour les réseaux filaires, il a été le premier simulateur supportant les réseaux ad-hoc. Du fait de sa grande utilisation, de nombreuses extensions et protocoles ont été développés pour ns-2. Cela concerne aussi bien, les modèles de propagation, de mobilité, que les protocoles de routage. Néanmoins, de conception ancienne, il n'utilise pas les langages de programmation modernes : l'utilisation du langage OCTL en plus du C++, dégrade les performances dramatiquement, et il ne supporte pas un grand nombre de nœuds. De plus, la conception souffre d'un manque de modularité et d'une complexité inhérente. En effet, l'ajout et la modification d'un module existant est loin d'être simple.

3.1.6.5 ns-3

ns-3 [8] est un projet récent démarré en juillet 2006 par la Georgia Institute of Technology (auteur de GTNetS), l'ICSI Center for Internet Research et le groupe de travail "Planète" de l'INRIA Sophia-Antipolis. Toujours en phase de développement, ns-3 est destiné au remplacement éventuel de ns-2, afin de combler ses lacunes en termes de performances et de modularité. ns-3 est développé en C++ et peut être relié au langage de script Python. ns-3 privilégie l'utilisation de logiciels libres, comme Waf (outils de compilation et distributions

ou, Doxygene (génération automatique de documentation) et, tout comme GTNets, le standard POSIX pour les fonctions liées à la gestion des sockets. Comme J-Sim, il intègre la génération de nombres aléatoires suivant diverses distributions. ns-3 supporte le passage à l'échelle grâce à des paquets "virtual zero byte" ou des fonctionnalités optionnelles pour les nœuds. Il supporte également les fonctionnalités de "cross layer". L'intégration des outils extérieurs sont multiples par : la génération de fichiers de traces PCAP (utilisé par Wireshark), l'ordonnancement temps réel (intégration dans des plateformes d'expérimentation temps réel), intégration possible de modules du noyau Linux de la couche TCP/IP aux simulations (permettant ainsi l'émulation), enfin l'utilisation, sans modification, d'applications respectant la norme POSIX.

3.1.6.6 OMNet++

OMNet++ [10] est un simulateur de conception rigoureuse développé en C++ à l'Université technique de Budapest, département des télécommunications (BME-HIT). OMNet++ est un simulateur universel capable de simuler tout système composé de matériels qui communiquent entre-eux. Son approche "framework" lui confère une grande modularité : au lieu de créer un simulateur supportant explicitement les réseaux informatiques filaires, ou autre domaine de simulation, OMNet++ fournit une infrastructure permettant cela. Des frameworks (pour la plupart Open Source) gérant différents domaines de simulation des réseaux informatiques lui sont ensuite associés. Il existe les frameworks *INET* pour la simulation filaire et sans fil de réseaux TCP/IP, *Mobility* ou *MiXim* pour la simulation des réseaux mobiles sans fil, et *Castelia* pour les réseaux de capteurs.

3.1.6.7 JIST/SWANS

JiST (Java in Simulation Time) [4] est un simulateur à événements discrets écrit en Java et développé par l'université de Cornell (USA). Pour la gestion des réseaux sans fil, il est associé à SWANS (Scalable Wireless Ad hoc Network Simulator), permettant, comme son nom l'indique, le passage à l'échelle pour simuler de larges réseaux ad hoc. Nous étudions en détail ce simulateur dans la section suivante.

3.2 Le simulateur JIST/SWANS

Les auteurs de JiST [21] proposent de classer les simulateurs en trois catégories d'un point de vue de l'architecture (noyaux ou bibliothèque) et de l'outil de modélisation (langages).

L'architecture *noyaux* crée une abstraction de l'horloge de simulation, la rendant transparente. Pour celà, il gère les processus en contrôlant leur ordonnancement, les communications inter-processus et l'horloge système pour ses applications qu'il exécute. Un tel système supporte l'exécution distribuée et concurrente des applications simulées. De plus, l'interface entre le système et les applications sont des appels standards des systèmes d'exploitation donc les applications déjà conçues n'ont pas besoin d'être modifiées. Malheureusement, le contrôle d'exécution de processus des simulations à noyaux n'est pas efficace, rendant le simulateur moins performant.

Les *bibliothèques* de simulation abandonnent l'idée de transparence en faveur d'une plus grande efficacité. Le but étant de réunir le noyau du simulateur et ses applications dans un seul processus monolithique contenant à la fois le modèle de simulation et le moteur d'exécution. Malheureusement, cette méthode rend le programme complexe par l'appel de fonctions de la bibliothèques, limitant les possibilités d'optimisation de son exécution.

Les *langages* spécifiques à la simulation, comme Simula ou Parsec, sont conçus pour simplifier le développement de simulations. Les contraintes sur l'état de la simulation et les causalités des événements peuvent être statistiquement forcés par le simulateur permettant ainsi d'importantes optimisations statiques comme dynamiques. Malheureusement, l'utilisation d'un langage spécifique impose la réécriture des programmes et peut paraître rébarbatif pour l'utilisateur du simulateur.

La motivation des auteurs pour créer JiST est de développer un système de simulation réunissant les avantages des trois catégories de simulateur décrits ci-dessus. L'approche originale de JiST est d'adapter la machine virtuelle Java en une plateforme de simulation ; permettant de réunir toutes ces caractéristiques dans un même simulateur.

3.2.1 Présentation de JIST/SWANS

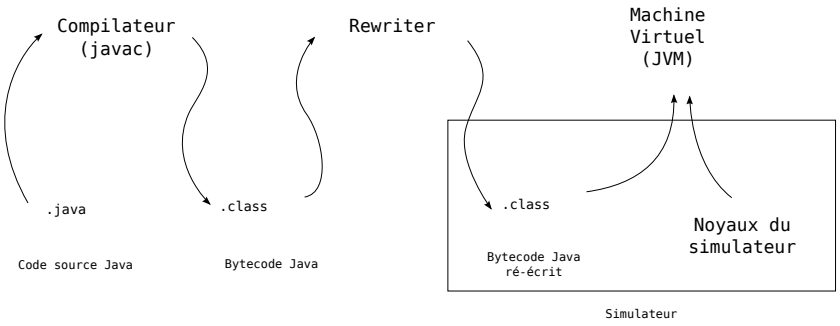


Figure 3.2 – L'architecture de JiST est composée de quatre composants

Le fonctionnement détaillé de JiST est issu de la thèse de R. Barr [21]. L'architecture de JiST est composée de quatre composants distincts : un compilateur, un *rewriter* de *bytecode*, le noyaux du simulateur et une machine virtuelle (Figure 3.2). Une fois la classe compilée, elle est modifiée par le rewriter de bytecode. Le rewriter ré-écrit le code de la classe au niveau du langage compris par la machine virtuelle (le byte code). Cette ré-écriture est nécessaire pour que la classe ne s'exécute plus suivant l'horloge du système, mais suivant l'horloge de simulation.

Le développeur de la classe à simuler, ajoute dans le code une instruction *sleep* qui permet d'avancer l'heure de simulation. Une simulation simple appelée "Hello word", présentée en Figure 3.3, est proposée par le concepteur de JiST. C'est un programme Java valide, et on peut le compiler avec un compilateur Java standard. Mais pour démarrer le programme

avec la sémantique de simulation, il doit être démarré avec JiST dans une machine virtuelle standard.

Le code de cette simulation met en avant plusieurs points. Premier point, la classe "hello" est une entité car elle implémente l'interface "Entity" (ligne 3). L'entité est créée (ligne 8) et ses méthodes sont appelées (ligne 9 et 15) comme un objet Java normal. La classe JistAPI de JiST, utilisée aux lignes 3, 14, et 17, représente l'interface avec le noyau de simulation. L'appel d'une méthode est effectué par rapport à l'heure de simulation. Ce type d'appel apparaît en ligne 15, comme un appel récursif de la méthode. Si le programme est exécuté sans JiST dans la machine virtuelle,le programme s'interrompt avec un dépassement de pile. Heureusement, avec JiST, la sémantique est de programmer l'événement "appel de la méthode myEvent" à l'aide de l'heure de simulation et donc l'appel devient *non bloquant*. En effet, l'appel de la méthode "sleep" (ligne 14) avance l'horloge d'une étape, et l'appel de la méthode "myEvent" est ordonnancé comme un nouvel événement dans le futur. Finalement, le programme affiche un message avec l'heure de simulation (ligne 17). Au lieu d'un dépassement de pile, le programme affiche alors :

```
> simulation start
> hello world, t=1
> hello world, t=2
> hello world, t=3
> etc.
```

Le re-writer rend donc possible le ré-ordonnancement de la classe par le noyau du simulateur, tout en rendant toujours possible l'exécution du bytecode par une machine virtuelle non modifiée. Ainsi, au moment de l'exécution de la simulation, JiST exécute les classes non plus selon le seul ordre établi par la sémantique du langage Java, mais en prenant en considération l'horloge de la simulation.

JiST est juste le simulateur à événements discrets. On lui associe un module, appelé SWANS, permettant la simulation de réseau ad hoc. L'ensemble des classes de SWANS sont regroupées dans le paquet *jist.swans*. SWANS est composé de plusieurs composants correspondant à des paquets java (mentionné entre parenthèses) :

Physical (*jist.swans.field* et *jist.swans.radio*) modélise la propagation du signal (entité Field) venant des différents équipements radios (Radio), ainsi que la mobilité des nœuds (Mobility).

Link (*jist.swans.mac*) met en œuvre les protocoles d'accès au média (MAC). Trois protocoles MAC sont implémentés : 802.11b, dump (transmet seulement quand le transmetteur est libre), loop.

Network (*jist.swans.net*) met en œuvre IP. Envoie également le paquet au protocole de routage utilisé pour connaître le prochain saut.

Routing (*jist.swans.route*) met en œuvre les protocoles de routage. Les protocoles disponibles sont : AODV, DSR et ZRP. Le protocole OLSR est également disponible via notre implémentation à l'IEF de l'université Paris-Sud.

Transport (*jist.swans.trans*) met en œuvre les protocoles de transport, ici, TCP et UDP.

```
1  import jist.runtime.JistAPI;
2
3  class hello implements JistAPI.Entity
4  {
5    public static void main(String[] args)
6    {
7      System.out.print("start␣simulation");
8      hello h = new hello();
9      h.myEvent();
10   }
11
12   public void myEvent()
13   {
14     JistAPI.sleep(1);
15     myEvent();
16     System.out.print("hello␣world,␣t="
17       + JistAPI.getTime());
18   }
19 }
```

Figure 3.3 – Exemple d'une simulation simpliste avec JiST. L'entité affiche un message à chaque étape de simulation. Reproduit à partir de [21]

Application (*jist.swans.app*) en haut de la pile de protocoles. SWANS permet l'exécution d'une application réseau développée en Java sans qu'elle soit modifiée. Une application "-heartbeat" est également disponible pour détecter les ruptures de lien avec un protocole pro-actif.

Comme le montre la Figure 3.4, chacun de ces composants peuvent être composés ensemble, pour créer un simulateur de réseau ad hoc opérationnel.

3.2.2 Les modèles de mobilité de JIST/SWANS

Pour la simulation d'un réseau mobile, le modèle de mobilité est indispensable pour calculer le déplacement des nœuds. Le réalisme de ce modèle permet d'obtenir des simulations plus proche de la réalité. Le modèle de mobilité le plus couramment employé est le déplacement aléatoire. Plusieurs variantes existent.

La première, nommée *random way point*, consiste à choisir une position aléatoirement dans le plan, cette position étant la cible à atteindre. Le nœud se déplace alors vers cette cible avec une vitesse aléatoire. Quand il l'a atteinte, il choisit une autre cible.

Dans la seconde variante, nommée *random walk*, ce n'est pas une cible qui est choisie, mais une direction aléatoire fixée pour une certaine distance de déplacement. Le nœud se déplace dans cette direction avec une vitesse aléatoire et choisit une autre direction quand la distance définie au préalable est parcourue.

Enfin une troisième variante, nommée *teleport*, moins réaliste consiste à choisir une cible et à s'y déplacer instantanément, attendre un délai aléatoire, et recommencer.

Le modèle de mobilité aléatoire peut être réaliste dans le cas du déplacement d'un piéton dans un espace libre sans obstacle. Cela diffère d'une mobilité réelle où un grand nombre

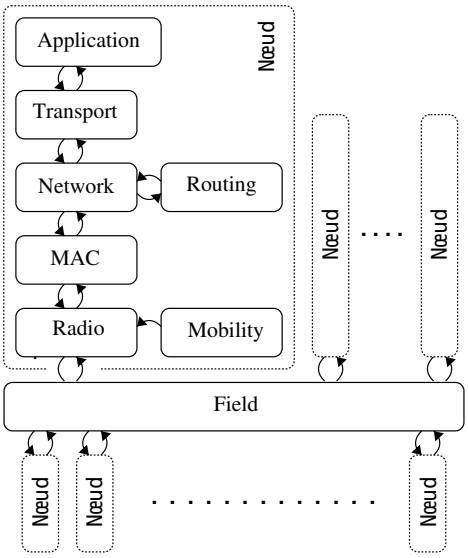


Figure 3.4 – Exemple d'assemblage des composants de SWANS pour créer un réseau ad-hoc

de facteurs rentrent en jeu. Par exemple, un piéton ou un véhicule suit de préférence un chemin ou une route, influençant la mobilité du nœud. Un nœud ne pouvant pas traverser un obstacle fixe comme mobile, un obstacle mobile pouvant être un autre nœud. Cette contrainte influe sur la vitesse du nœud, il doit ralentir si celui-ci ne peut contourner un obstacle mobile ou s'arrêter si l'obstacle est fixe. Nous voyons qu'une telle mobilité prend en compte à la fois la topologie du terrain et la vitesse des autres nœuds, c'est-à-dire son *environnement*.

3.3 Simulateur de trafic routier sur autoroute

Dans un réseau de véhicules, l'environnement du nœud doit être pris en compte pour obtenir une mobilité la plus proche de la réalité. Pour cela, un *simulateur de trafic routier* est utilisé. Ce simulateur interagit avec le simulateur de réseau, soit par l'intermédiaire de traces (fichier texte décrivant la mobilité des nœuds) qui peuvent être lues par le simulateur de réseau, soit en l'intégrant comme modèle de mobilité au simulateur de réseau. Il existe deux types de simulateurs de trafic : microscopique ou macroscopique. Un simulateur microscopique modélise le déplacement d'un véhicule par rapport aux autres. Il simule alors le déplacement de chaque véhicule sur la route, en estimant régulièrement son accélération, sa vitesse, et sa position. Un simulateur macroscopique est moins fin ; il voit les véhicules sur la route comme un flux. Par analogie avec la physique, un simulateur microscopique modélise le déplacement de chaque atome dans un gaz, alors qu'un simulateur de macroscopique modélise le gaz comme ayant certaines propriétés : volume, pression, température (base de la thermo-dynamique), sans se préoccuper de la position et du déplacement des molécules

constituant le gaz.

Nous nous intéressons uniquement aux réseaux de véhicules sur autoroute. Nous avons donc besoin d'un simulateur de trafic adapté à ce type de configuration. Les simulateurs Open Source disponibles au moment où nous avons débuté nos travaux étaient SUMo [11] et STRAW [12]. Seulement ces deux simulateurs sont orientés pour les réseaux routiers urbains, et ne sont pas assez fins pour la simulation du déplacement de véhicules la plus réaliste possible sur autoroute. Entre autre, STRAW ne gérait pas à l'époque le changement de voie. Nous avons donc choisi d'utiliser un simulateur issu de la thèse de M. Mabiala [84] et développé en interne par l'IEF de l'université Paris-Sud. Nous allons décrire le fonctionnement de ce simulateur en détail.

Le déplacement d'un véhicule est basé sur : (1) les caractéristiques du véhicule et (2) le modèle de mouvement du véhicule.

Les caractéristiques du véhicule comprennent [127] : les performances du véhicule et le comportement du conducteur. Les performances du véhicule dépendent de l'accélération maximale, de la décélération maximale (freinage en cas d'urgence), de la décélération normale (freinage ou ralentissement en cas de non-urgence). Le comportement du conducteur est essentiellement sa vitesse désirée, c'est-à-dire la vitesse à laquelle il souhaite se déplacer, si les conditions lui permettent. La vitesse désirée est limitée par la vitesse maximum d'une voie. Cette vitesse corrigée est appelée *vitesse cible*.

Le modèle de mouvement d'un véhicule comprend : le modèle de poursuite [127] et le modèle de changement de voie [14]. Le modèle de poursuite détermine l'accélération (ou la décélération) courante en fonction de l'environnement. L'environnement étant essentiellement le véhicule qui est devant lui (s'il existe) ; ce véhicule étant appelé *leader*. Nous définissons plusieurs régimes en fonction de la position et de la vitesse du leader :

- Free flowing : Le véhicule peut se déplacer librement sur la voie, le leader étant suffisamment éloigné. Dans ce régime le véhicule accélère jusqu'à atteindre sa vitesse cible.
- Emergency : Le leader est situé à une distance très proche. Le véhicule utilise une décélération maximale pour éviter de le percuter.
- Car-following : Le leader est suffisamment proche rendant le déplacement libre impossible, mais suffisamment loin pour ne pas appliquer un freinage d'urgence. Dans ce régime le véhicule adapte son accélération en fonction de la vitesse du leader.

Grâce à la description de ces différents régimes, nous en déduisons l'accélération du véhicule à tout instant. L'autre modèle utilisé, est un modèle de changement de voie probabiliste. Un véhicule change de voie si deux conditions sont remplies : le véhicule est dans des conditions non favorables (le leader à une vitesse trop faible) et un changement de voie est possible à droite ou à gauche (les véhicules sur la voie adjacente ne le gêne pas). Si ces deux conditions sont remplies, le véhicule change de voie avec une certaine probabilité.

L'implémentation de ce simulateur a été intégré dans SWANS. Néanmoins, il n'a pas été possible d'intégrer ce simulateur comme modèle de mobilité de SWANS, car l'interface proposée ne permet pas de prendre en compte l'interaction des nœuds entre eux, comme le fait notre modèle de mobilité véhiculaire. Nous avons alors développé un simulateur de trafic autoroutier, respectant le modèle décrit précédemment, en java, mais totalement indépendant de JIST/SWANS (classe *highway*). Une autre classe *jist.swans.highway.HighwayProxy* est

implémentée comme entité de Jist et permet l'interfaçage entre SWANS et le simulateur de trafic routier.

Nous avons décrit les différents simulateurs existants, ainsi qu'un modèle de mobilité réaliste pour réseaux de véhicules sur autoroute. A partir de ces deux outils nous pouvons simuler un réseau ad hoc de véhicules, afin d'en évaluer les performances d'un protocole. Néanmoins, plusieurs extensions doivent être apportées à SWANS pour qu'il puisse gérer les réseaux ad hoc hybrides.

3.4 Réseau ad hoc hybride pour JIST/SWANS

Pour effectuer les simulations nécessaires à l'évaluation d'un protocole dans un contexte hybride, nous avons apporté plusieurs améliorations et modifications à SWANS. Bien que SWANS, contrairement à NS-2, soit adapté à un réseau contenant des nœuds possédant plusieurs interfaces réseaux, il ne fournit pas une interface radio filaire, de type Ethernet. Étant donné que les interfaces filaires actuelles utilisent Ethernet avec un switch, nous n'avons pas implémenté le protocole Ethernet, mais juste une interface MAC ayant un débit de 100 Mbit/s ; la gestion des collisions n'étant pas nécessaire. Une entité Switch est également implémentée pour relier plusieurs interfaces MAC Ethernet entre elles : chaque paquet reçu est envoyé à l'interface MAC correspondant à l'adresse MAC de destination. Avec ces modifications, nous pouvons alors créer un nœud possédant des interfaces radios hétérogènes. Cette extension est située dans le paquet *jist.swans.bus*.

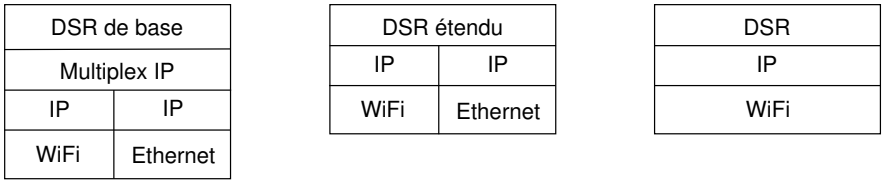
Nous avons étendu l'implémentation de DSR. Cette implémentation permet de définir une AP comme un nœud possédant une interface radio et deux interfaces filaires : une reliant l'AP suivante et une l'AP précédente. Elle permet également, entre autre, l'annonce des points d'accès sur le réseau, l'enregistrement des nœuds mobiles auprès des AP et la recherche d'un nœud dans un réseau ad hoc hybride.

Nous avons renommé la classe *RouteDsr*, implémentant DSR dans swans, en *RouteDsrBase*. Notre extension est implémentée comme une classe nommée *RouteDhg* héritant de *RouteDsrBase*. Pour garder la compatibilité avec la version originale de SWANS, une nouvelle classe nommée *RouteDsr* hérite de *RouteDsrBase* mais contenant aucune implémentation. Cette classe se comporte alors comme la classe *RouteDsr* originelle.

Le DSR de base ne permet pas d'utiliser plusieurs interfaces radio ; les paquets étant envoyés et reçus à une seule interface IP. Pour utiliser le DSR de base sans le modifier, nous avons créé une entité IP (nommé *jist.swans.net.NetMultiInterface*) regroupant plusieurs entités IP. Tout paquet reçu par cette entité est envoyé à l'ensemble des entités IP. L'architecture logique du réseau ad hoc hybride utilisant DSR de base et DSR étendu est décrit en Figure 3.5.

3.4.1 Routage géographique pour JIST/SWANS

Nous avons implémenté un protocole de routage géographique simple adapté au réseau de véhicules. Ce protocole de routage est implémenté dans la classe *jist.swans.route.RoutePos*. Ce protocole possède trois fonctionnalités :



(a) Point d'accès avec DSR de base.

(b) Point d'accès avec DSR étendu pour réseau ad hoc hybride.

(c) Noeud mobile. La version de DSR peut être celle de base ou celle étendue.

Figure 3.5 – Architecture réseau au niveau de la couche 2 et 3 pour les points d'accès (3.5a et 3.5b) ou les nœuds mobiles (3.5c).

- la diffusion d'un paquet dans la direction de déplacement des véhicules,
- la diffusion d'un paquet dans la direction opposée au déplacement du véhicules,
- le routage d'un paquet vers une position géographique.

La diffusion d'un paquet dans la direction (respectivement dans la direction opposée) du déplacement du véhicule permet de délivrer un message à tous les véhicules en amont (respectivement en aval) du véhicule initiateur du message. Ce routage est un cas particulier de la transmission conditionnelle que nous avons décrit dans le paragraphe 2.3.2, lors de la description des protocoles de routage adaptés au réseau ad hoc de véhicules.

Un réseau de véhicules sur autoroute pouvant être dense, nous adoptons également l'optimisation CBF (§ 2.1.3.4) permettant de réduire l'overhead. De plus, le réseau étant linéaire et la couverture radio très grande devant la largeur de la route, nous pouvons utiliser la méthode basique de suppression des candidats. En effet, il n'existe pas de nœuds à l'extérieur du triangle de Reuleaux [59] (Figure 3.6) donc le problème de duplication de paquets n'existe pas.

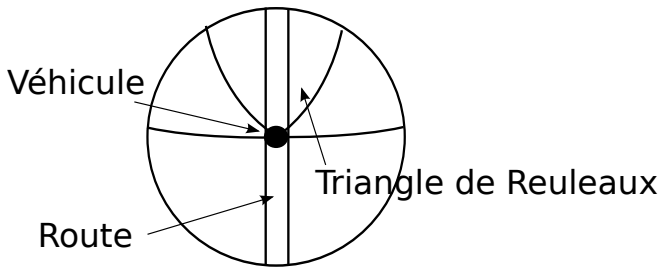


Figure 3.6 – Triangle de Reuleaux et une route : il n'existe pas de nœuds sur la route à l'extérieur du triangle.

Nous avons également implémenté la fonctionnalité de positionnement géographique dans la classe *jist.swans.field.Gps*. Cette entité permet d'obtenir la position courante du nœud, ainsi que sa vitesse courante.

3.4.2 Bugs corrigés dans SWANS

Nous avons corrigé deux bugs dans l'implémentation de DSR dans SWANS. Le premier rend non fonctionnel DSR dans le cadre d'un réseau non-statique. En effet, il concerne le traitement du message ROUTEERR. La réception de ce message par la source est mal interprétée, donc toute modification de topologie cassant une route entre la source et la destination, n'est pas prise en compte par la source pour renvoyer une demande de chemin. L'origine du bug est la lecture au mauvaise endroit dans le message ROUTEERR de l'adresse du nœud qui n'a pas pu être atteint.

Le second bug empêche le fonctionnement de l'optimisation *Gratuitous Route Reply*. Cette optimisation permet à un nœud intermédiaire connaissant le chemin vers la destination de répondre par un ROUTEREPLY à la place de la destination. Le nœud crée alors le chemin en combinant le chemin parcouru par le message ROUTEREQUET et le chemin contenu dans son cache pour atteindre la destination. Le bug concerne la construction de ce chemin par combinaison de deux chemins qui est mal réalisée.

L'ensemble des modifications apportées à JIST/SWANS nous permet d'évaluer les protocoles que nous avons développés dans un contexte de réseau de véhicules sur autoroute. Ces protocoles seront décrits dans les chapitres suivants. Pour l'évaluation de performances nous avons besoin de métriques décrites dans la section suivante.

3.5 Métriques utilisées pour la mesure de performances

Nous avons besoin de critères pour comparer les performances d'un protocole ad hoc. Nous utilisons des métriques différentes évaluées par expérimentation, simulation ou à l'aide de modèles théoriques afin de déduire les points forts et les points faibles des protocoles.

Nous utilisons cinq métriques différentes pour évaluer les performances d'un protocole ad hoc : le délai, le débit, le taux de paquets reçus (PDR pour Packet Delivery Ratio), le surplus de messages de gestion (overhead), la longueur du chemin et la capacité. Nous allons détailler ces métriques.

3.5.1 Délai d'acheminement des paquets

Le délai d'acheminement d'un paquet (ou simplement *délai*) est la durée qui sépare le moment où le paquet est envoyé par la source et le moment où il est reçu par la destination. L'unité de mesure et la seconde. Le délai dépend de la durée de stockage du paquet dans les files d'attente de chaque nœud intermédiaire (aussi bien au niveau de la couche réseau ou MAC) et de la durée de transmission physique du paquet (propagation du signal), mais cette durée est négligeable par rapport au temps de stockage du paquet dans les nœuds. Le délai d'acheminement d'un paquet est alors fonction du nombre de relais est donc de la distance entre la source et la destination.

Dans un réseau ad hoc, la mise en file d'attente d'un paquet est causée par de multiples facteurs comme le temps de calcul pour trouver le prochain saut, la recherche d'un chemin

par la source (protocoles topologiques réactif), le délai nécessaire pour considérer un nœud intermédiaire comme inaccessible, l'attente d'accès au médium imposé par le protocole de la couche MAC ou le débit de transmission.

3.5.2 Taux de réception

Le taux de réception permet d'évaluer la part des paquets arrivés à destination. C'est le nombre de paquets reçus divisé par le nombre de paquets envoyés. C'est une proportion et donc typiquement exprimée en pourcentage. Les causes de la perte d'un paquet, et donc de la diminution du taux de réception, est la suppression du paquet de la file d'attente d'un nœud sans qu'il soit transmis au nœud suivant. Ceci est provoqué si, après un délai déterminé, le nœud n'a pu transmettre le paquet car le prochain nœud est inaccessible ou le canal de transmission est saturé dû à une erreur de transmission. C'est aussi le cas si le file d'attente est remplie (congestion).

3.5.3 Débit de réception

Le débit de réception, appelé aussi *throughput* en anglais, est la quantité de données reçues par la destination (donc transmis avec succès) par unité de temps. L'unité est le nombre de bits par seconde (bit/s, kb/s, Mb/s, etc). Le débit est directement lié au taux de réception. En effet, s'il n'y a aucune perte, le débit de données reçues par la source et égal au débit de données transmis par la source. Plus il y a de pertes, plus le débit de réception diminue.

3.5.4 Surcharge de trafic (overhead)

La surcharge de trafic, ou *overhead* en anglais, est provoquée par les données transmises sur le réseau pour assurer le fonctionnement des protocoles de routage : établir et maintenir les routes. Ces données sont soient des messages de signalisation, c'est-à-dire un paquet ne contenant pas de données utilisateur, ou soient des données en tête de paquets contenant des données utilisateur. L'overhead est nécessaire au fonctionnement des protocoles de routage, il ne peut être nul. Diminuer l'overhead permet d'augmenter le débit disponible par l'utilisateur. Soit P la taille totale des messages de signalisation transmis sur le réseau par le ou les protocoles, U la taille totale des données *utiles* transmises sur le réseau, et T la taille totale des données transmises sur le réseau. Alors, l'équation 3.1 exprime l'overhead. L'overhead est donc une proportion typiquement exprimée en pourcentage.

$$\frac{P}{T} = \frac{T - U}{T} \tag{3.1}$$

3.5.5 Longueur de chemin

La longueur du chemin est le nombre de nœuds intermédiaires plus un sur le chemin emprunté par un paquet pour atteindre la destination. C'est également le nombre de sauts

pour atteindre la destination. Dans un protocole ad hoc, on cherche à diminuer la longueur du chemin moyen pour diminuer le délai d'acheminement moyen. En effet, moins il y a de nœuds intermédiaires, moins il est sujet aux pertes.

3.5.6 Intervalle de confiance

La mesure de métriques par simulation, en utilisant un modèle de mobilité probabiliste ou par expérimentation, fournit des résultats différents dûs aux conditions initiales différentes ou à l'environnement d'expérimentation changeant. Pour obtenir un résultat convaincant, nous avons besoin d'effectuer plusieurs fois la mesure de la métrique. L'ensemble de ces mesures (noté X), génère un *échantillon* de k résultats. On note x_i i-ème mesure d'un échantillon. Une moyenne des mesures (noté $\bar{x}$) de l'échantillons permet alors de résumer le résultat. Cependant, nous avons besoin de connaître la confiance que nous accordons à cette estimation et donc savoir si nous devons effectuer d'avantage de simulations pour affiner cette moyenne. Le degré de confiance d'une moyenne est évaluée à l'aide d'un *intervalle de confiance*. Le degré de confiance est en principe exprimé sous la forme d'une probabilité. Par exemple, un intervalle de confiance à 95 % (ou au seuil de risque de 5 %) a une probabilité égale à 0,95 de contenir la valeur du paramètre que l'on cherche à estimer. Plus l'intervalle de confiance est de taille petite, plus l'incertitude sur la valeur estimée est petite. L'équation 3.2 permet de calculer cet intervalle de confiance à 95 % et est basé sur la méthode Monte-Carlo.

$$\frac{1,96\sqrt{Var(X)}}{\sqrt{k-1}} \tag{3.2}$$

3.5.7 Capacité

La capacité est le nombre de bits qu'un réseau peut transmettre par seconde [60]. C'est équivalent au taux de réception maximal de tous les nœuds du réseau. Dans un réseau sans fil, la capacité dépend du taux de transmission de la technologie radio utilisé, de l'overhead et de la réutilisation spatiale, c'est-à-dire utiliser l'ensemble de l'espace disponible sur le réseau pour éviter les collisions provoquer par la transmission de données sur un même canal radio. Dans le cas d'une autoroute, la réutilisation spatiale est limiter car le réseau à une dimension. L'évaluation de la capacité est alors important.

Conclusion

Nous avons présenté plusieurs simulateurs existants avec leurs caractéristiques et leurs limites. Nous avons choisi d'utiliser le simulateur JIST/SWANS. Nous avons étendu la simulateur pour le support de nœuds à plusieurs interfaces réseaux, ajouté un protocole ad hoc minimaliste basé sur la position géographique, et corrigé des bugs liés à l'implémentation de DSR. Nous avons également implémenté un simulateur de trafic routier pour ajouter la mobilité de véhicules sur autoroute à SWANS.

Ces modifications nous permettent d'évaluer les protocoles que nous avons développé dans un contexte de réseau de véhicules sur autoroute à l'aide des métriques décrites précédemment.

La simulation n'est pas le seul moyen de mesurer une performance d'un réseau. Avant de présenter les protocoles que nous avons développés, nous proposons une méthode alternative à la simulation seule pour évaluer le passage à l'échelle d'un réseau de véhicules. Cette méthode est basée sur un modèle d'évaluation *hybride* de l'overhead, utilisant à la fois la simulation, mais utilisant également une étude théorique des protocoles.

Chapitre 4

Modélisation de protocoles de routage pour réseau ad hoc de véhicules

Dans le chapitre 2, nous avons discuté de l'importance des protocoles de routage dans les réseaux de véhicules. En effet, ils réalisent la connectivité du réseau même en présence de mouvements fréquents ou de vitesses élevées de véhicules. Pour les réseaux de véhicules, deux catégories de protocoles [80] ont été identifiées : topologique et géographique.

Dans ce chapitre, notre objectif est d'évaluer le passage à l'échelle d'un réseau ad hoc de véhicules sur autoroute. Nous proposons un modèle d'évaluation de protocole à l'aide d'une méthode hybride utilisant simulation pour évaluer un paramètre caractérisant la mobilité et un modèle quantitatif pour évaluer le nombre de messages de contrôle envoyés par un protocole et donc le surplus de messages générés (overhead).

Dans une première partie, nous allons décrire un modèle d'évaluation de l'overhead pour une communication. Ce modèle nécessitant la connaissance d'un paramètre caractérisant la mobilité, nous allons l'évaluer à l'aide d'une simulation basée sur la simulateur de trafic routier. Enfin, nous présenterons les résultats de notre évaluation concernant le passages à l'échelle de réseau ad hoc véhicules pour des protocoles topologiques et géographiques.

4.1 Comparaisons des approches de routage

La comparaison de protocoles à l'aide de simulations, dans le cadre d'un réseau de véhicules, a fait l'objet de nombreux travaux [80, 54, 85, 121]. Ceux-ci comparent les protocoles réactifs ou géographiques en mesurant leurs performances en termes de délais, d'overhead et de taux de réception des paquets.

Les problèmes de la comparaison de protocoles proviennent des simulations et/ou du simulateur. La complexité des simulateurs, leur hétérogénéité et la différence d'implémentations des modèles, la non justification de toutes les hypothèses (explicites ou implicites) utilisées rendent les résultats discutables et non reproductibles. En effet, il n'est pas rare d'être confronté à la difficulté de reproduction d'une simulation, car on ne possède pas

l'ensemble des paramètres de simulation, le code source de la simulation, ou la bonne version du simulateur. L'implémentation en elle-même pose souvent problème, l'utilisation de la programmation impérative (avec les langage utilisés par la plupart des simulateurs : C, C++, java, ect.) permet difficilement la vérification du code et les bugs sont plus courants, contrairement à une programmation fonctionnelle.

4.2 Évaluation du passage à l'échelle à l'aide d'un modèle quantitatif

Notre cas d'étude est le passage à l'échelle d'un réseau de véhicules sur autoroute, en particulier l'évolution de l'overhead en fonction de la taille du réseau. Dans ce contexte, la taille du réseau, en nombre de nœuds, dépend de la densité du trafic routier et de la longueur de l'autoroute. Le passage à l'échelle est un des critères de comparaison de performances d'un réseau ad hoc, il est donc important d'avoir un outil fiable permettant de l'évaluer.

L'évaluation du passage à l'échelle du protocole topologique réactif DSR est présenté en Figure 4.1. Elle est effectuée à l'aide du simulateur JIST/SWANS. Cette courbe exprime le nombre de messages de contrôle (RREQ, RREP, et RERR) envoyés dans le réseau en fonction de la densité et de la taille de l'autoroute. Ces deux paramètres influent sur l'overhead, rendant difficile le passage à l'échelle d'un tel réseau utilisant ce protocole.

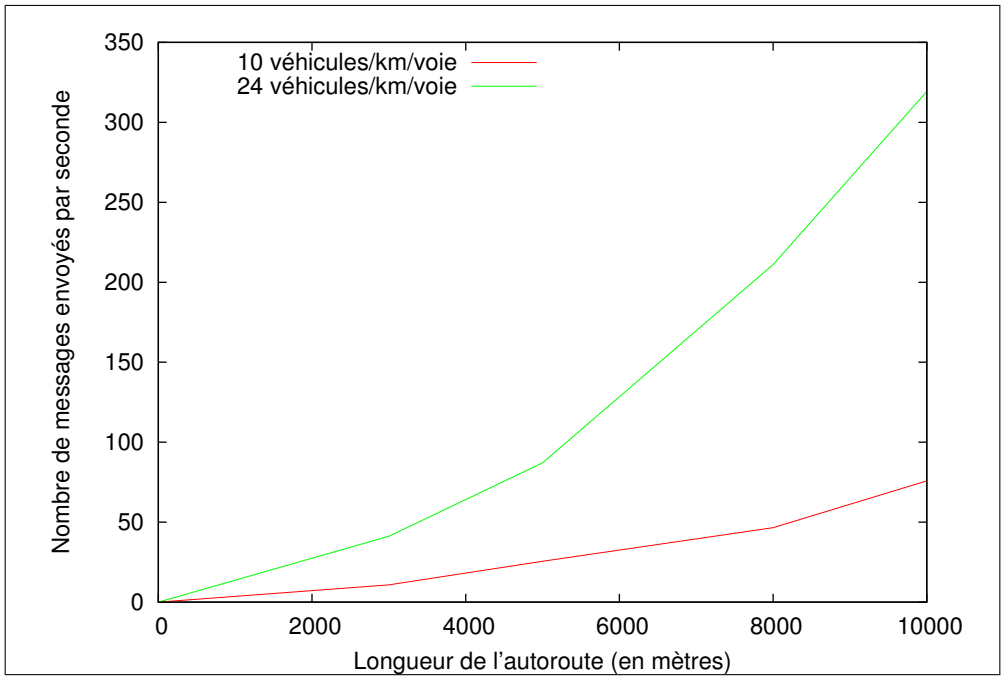


Figure 4.1 – Évolution du nombre de messages de contrôle envoyés en fonction de la longueur de l'autoroute et de la densité de véhicules.

Pour évaluer l'overhead d'un protocole, nous adoptons un modèle quantitatif, consistant à compter le nombre de messages de contrôle envoyé dans le réseau par un protocole

pendant une période donnée. Cette approche est adoptée dans [43] pour évaluer des protocoles de localisation, mais il ne modélise qu'un réseau statique, le réseau dynamique étant simulé. [112] propose une modélisation pour évaluer le passage à l'échelle des protocoles ad hoc topologiques (proactifs, réactifs et hybrides) et définit une métrique appelée *total overhead* qui correspond à l'overhead généré par un protocole pour le bon fonctionnement du routage. Dans cette étude, pour DSR (Dynamic Source Routing), seule la borne inférieure de l'overhead total est calculée lorsqu'il n'y a pas de mobilité (i.e. pas de reconstruction de chemin).

Pour prendre en compte la mobilité, les auteurs de [77] s'intéressent à la prédiction de la durée de vie des liens dans un réseau de véhicules sur autoroute pour le choix optimal des nœuds intermédiaires dans un chemin entre la source et la destination. La durée de vie d'un lien est définie comme la durée pendant laquelle les deux nœuds peuvent communiquer entre-eux et la durée de vie d'un chemin contenant M liens étant alors le minimum des durées de vie des liens du chemin. Leur problématique est de trouver la vitesse optimale affectée à chaque nœud intermédiaire en maximisant la durée de vie d'un chemin en fonction de la distance géométrique entre les nœuds intermédiaires. A partir de l'affectation de ces vitesses, un protocole peut choisir le chemin optimal permettant de maximiser sa durée de vie. Néanmoins, leur étude ne permet pas de déterminer la mobilité des nœuds, c'est-à-dire la distribution des vitesses et des positions relatives des nœuds du réseau. Nous verrons par la suite que ce sont ces valeurs qui nous intéressent pour quantifier la mobilité du réseau et donc évaluer un protocole ad hoc dans un contexte fortement dynamique.

Dans la section suivante nous introduisons notre modèle quantitatif permettant d'évaluer le passage à l'échelle dans un réseau ad hoc de véhicules sur autoroute. Ce modèle nous permettra ensuite de comparer l'évolution de l'overhead des protocoles topologiques et géographiques en fonction de la taille du réseau.

4.3 Modèle d'évaluation de l'overhead pour une communication

Dans un réseau ad hoc, diminuer le coût de la signalisation des protocoles augmente globalement le débit utile du réseau. La taille des messages de signalisation étant petite en comparaison de la taille des messages de données, nous nous attachons à limiter le nombre de messages de signalisation générés, plutôt qu'à réduire la quantité d'information transmise. Dans un premier temps, nous allons présenter globalement notre modèle d'évaluation de l'overhead d'un protocole ad hoc. Ensuite, nous allons décrire les paramètres permettant d'appliquer le modèle aux protocoles évalués.

Nous déterminons l'overhead dans un réseau produit par un protocole donné pour une communication entre deux nœuds. En sans fil, un paquet émis une seule fois par un nœud peut être reçu par tous les nœuds *voisins* c'est-à-dire à portée radio de l'émetteur. La portée radio est notée R : c'est la distance maximale entre deux nœuds pour qu'ils puissent communiquer à un saut. Comme dans la plupart des simulateurs, nous considérons que deux nœuds peuvent communiquer si et seulement si la distance les séparant est inférieure à R. Nous ne considérons donc pas les interférences du canal radio ou les erreurs de transmissions.

Les réseaux de véhicules sont un cas particulier des réseaux sans fil. En effet, la topologie

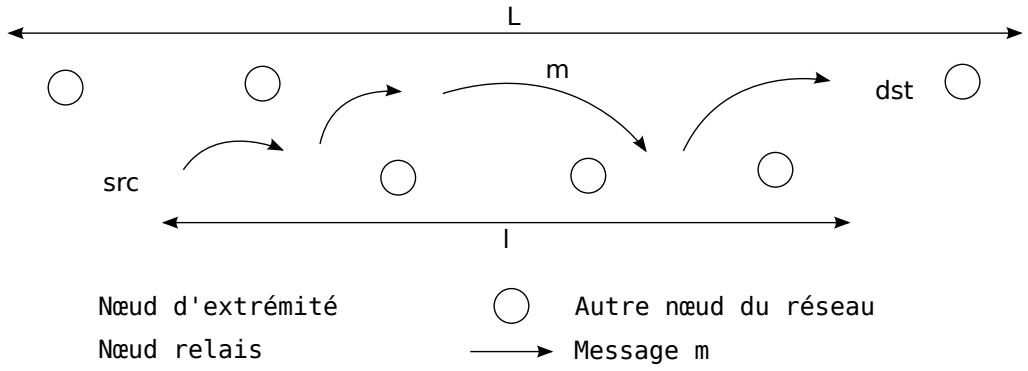


Figure 4.2 – Modèle proposé. Ici, le nombre de relais $n_r(m) = 3$

d'un réseau de véhicules sur autoroute étant proche d'une ligne et R grand devant la largeur d'une autoroute, nous considérons que le réseau n'a qu'une seule dimension. Un nœud i a une seule coordonnée x_i, impliquant qu'un nœud i peut communiquer avec un nœud j en un seul saut si et seulement si $|x_i - x_j| < R$

La communication, à plusieurs sauts, est réalisée entre le nœud source *src* et le nœud destination *dst*. Nous retrouvons l'ensemble des notations utilisées en Figure 4.2. Nous restons dans le cas général, où la position des nœuds source et destination ne dépend pas de l'application utilisée. Un message envoyé par la source (ou la destination) est relayé par plusieurs nœuds. Ces nœuds sont sur le chemin allant à la destination (ou source) ; on les appelle les nœuds *relais*. Pour un message de signalisation envoyé par la source ou le destinataire, on calcule le nombre de nœuds relais car ce nombre détermine le nombre de retransmission d'un message de signalisation.

On note L la longueur totale du réseau autoroutier en mètre et l la distance entre le nœud source et destination. On note $d(m)$ la densité des relais d'un message de type m en nombre de relais par mètre. Cette densité est nécessaire car nous devons différencier les différents modes de transmission d'un message suivant le protocole utilisé. Soit $n_r(m)$ le nombre de nœuds relais pour un message de type m dans le cadre d'une communication. Alors, $n_r(m) = d(m).l$. On note $f(m)$ la fréquence d'envoi (nombre d'envois par seconde) d'un message de type m par la source ou par la destination pour la communication entre *src* et *dst*.

On note $N(m)$ le nombre de messages de type m transmis par seconde. Alors,

$$N(m) = f(m).n_r(m) = f(m).d(m).l$$

Soit $M(P)$ l'ensemble des messages de signalisation d'un protocole. Par exemple pour DSR, $M(DSR) = \{RREQ, RREP, RERR\}$. L'équation 4.1 permet de déterminer le nombre total de messages transmis par seconde N_P pour une communication de *src* à *dst* dans le

cadre du protocole P.

$$N_P = \sum_{m \in M(P)} N(m) \tag{4.1}$$

Nous devons déterminer N_P pour chacun des protocoles à comparer. Pour cela, nous déterminons les valeurs des trois paramètres $d(m)$, l et $f(m)$ pour chaque type de messages m nécessaires au calcul de $N(m)$. Tout d'abord, nous caractérisons le mode de transmission d'un message de signalisation.

4.3.1 Densité des relais $d(m)$

Nous définissons trois modes de transmission :

- Point-à-point : la source et les nœuds relais transmettent le message à un seul de leurs voisins.
- Diffusion : les nœuds relais et la source transmettent le message à tous leurs voisins qui eux mêmes le relaieront à tous leurs voisins.
- Point-à-point avec connaissance du voisinage : la source et les nœuds relais transmettent le message à un seul de leurs voisins, en connaissant la position de leurs voisins.

Suivant le mode de transmission la densité de relais n'est pas la même. En effet dans une communication point-à-point, seuls les nœuds sur le chemin entre la source et la destination sont relais. Contrairement à la diffusion où tous les nœuds sont relais dans la zone de diffusion.

4.3.1.1 Densité des relais pour le point-à-point

Si un message se propage en *point-à-point*, la distance entre deux nœuds intermédiaires est au maximum R, d'où

$$d(m) \geqslant \frac{1}{R}$$

Le choix du plus court chemin, en nombre de nœuds relais, implique une distance entre chaque relais proche de R. Par simplification, nous considérons

$$d(m) = \frac{1}{R}$$

4.3.1.2 Densité des relais pour la diffusion

Soit d_t la densité du trafic routier (le nombre de véhicules par mètre), si un message est *diffusé* alors chaque nœud intermédiaire envoie un message, d'où

$$d(m) = d_t$$

4.3.1.3 Densité des relais pour le point-à-point avec connaissance du voisinage

Pour la découverte réactive du voisinage, chaque nœud intermédiaire doit connaître la position de ses voisins. Pour cela, le nœud intermédiaire envoie un message NREQ (Neighbour Request) à tous les voisins, et l'ensemble des voisins envoient leur position par un message NREP (Neighbour Reply). Tous les nœuds dans le voisinage ont envoyé un message, donc la densité de nœuds qui envoie un message est égale à la densité de véhicules, soit d_t.

Pour la découverte proactive du voisinage, la position étant stockée dans une table, le nœud relais n'a donc pas besoin d'envoyer de messages pour connaître la position de ses voisins. Par contre, les messages Hello envoyés régulièrement pour actualiser la table seront pris en compte dans le modèle.

4.3.2 Discussion sur la distance l entre la source et la destination

x_{src} est la position du nœud src et respectivement x_{dst} est celle du nœud dst. Nous considérons que les véhicules sont répartis uniformément sur l'autoroute. Dans ce cas, la distance moyenne l entre le nœud src et le nœud dst est l'espérance $L \times E(|x_{src} - x_{dst}|)$, où L est la longueur de l'autoroute. L'équation 4.2 calcule cette espérance.

$$\begin{aligned} E(|x_{src} - x_{dst}|) &= \int_0^1 \int_0^1 |x - y| dx dy \\ &= \int_0^1 \int_0^x (x - y) dy dx + \int_0^1 \int_0^y (y - x) dx dy \\ &= \int_0^1 x^2 - \frac{1}{2}x^2 dx + \int_0^1 \frac{1}{2}y^2 - y^2 dy \\ &= \frac{1}{2}\int_0^1 x^2 dx + \frac{1}{2}\int_0^1 y^2 dy \\ &= \frac{1}{2} \times \frac{1}{3} + \frac{1}{2} \times \frac{1}{3} \\ &= \frac{1}{3} \end{aligned} \tag{4.2}$$

Nous obtenons donc $l = \frac{1}{3}L$

Nous avons conscience que l'hypothèse de la répartition uniforme des véhicules sur l'autoroute est forte et est irréaliste au regard du trafic d'une autoroute. En revanche, cette distance l n'est pas déterminante pour évaluer l'évolution de l'overhead en fonction de la taille du réseau, si nous supposons la valeur l comme une fonction linéaire de L. Nous avons donc simplifié le modèle en considérant $l = \frac{1}{3}L$.

Notons que le message RERR a la particularité d'être envoyé par n'importe quel nœud sur le chemin. Si on considère que la probabilité de rupture d'un lien est homogène sur le chemin alors la distance entre le nœud émetteur du message et la source est égale à $\frac{1}{2}l = \frac{1}{6}L$

4.3.3 Nombre de relais n_r entre la source et la destination

Il existe trois cas de calcul de $n_r(m)$ suivant le mode de transmission de m :

- Si m est transmis en point-à-point, alors $d(m) = \frac{1}{R}$ d'où $n_r(m) = \frac{1}{R}.l = \frac{1}{3R}.L$
- Si m est diffusé, alors $d(m) = d_t$ d'où $n_r(m) = d_t.L$
- Si m est transmis en point-à-point avec découverte réactive du voisinage, alors $d(m) = d_t.l$ d'où $n_r(m) = d_t.l = \frac{1}{3}d_t.L$

4.3.4 Fréquence de rupture pour un protocole topologique réactif f_b

Pour un protocole ad hoc réactif, la fréquence d'envoi d'un message de signalisation dépend directement de la fréquence de rupture de chemin. Lorsqu'un chemin est rompu, le protocole envoie une nouvelle requête de recherche de chemin sur le réseau.

On remarque que la fréquence d'envoi des messages de signalisation dépend de la mobilité du réseau. Dans le cadre d'un réseau statique avec un protocole réactif, on n'a pas besoin d'envoyer de message pour reconstruire un chemin ou actualiser les tables de routage, la fréquence d'envoi est alors nulle.

A l'opposé, plus le réseau est dynamique, plus la fréquence d'envoi est élevée. Dans la suite, nous déterminons la fréquence de rupture d'un chemin en fonction de la probabilité de rupture d'un lien entre deux nœuds.

Un lien est une communication directe entre deux nœuds mobiles. Un lien est rompu lorsque les nœuds reliés ne peuvent plus communiquer entre eux directement. Un chemin est rompu si un des liens du chemin est rompu.

4.3.4.1 Probabilité de rupture d'un lien

On appelle *durée de vie d'un lien*, la durée qui sépare l'initialisation de la rupture du lien. La probabilité de rupture d'un lien p_t est la probabilité qu'un lien soit rompu pendant l'intervalle de temps $[t, t + \Delta t]$.

On souhaite connaître la probabilité de rupture d'un lien. Calculer p_t de manière théorique reste irréalisable pour un réseau de véhicules. L'évaluation de p_t est donc la seule donnée calculée expérimentalement.

4.3.4.2 Probabilité de rupture d'un chemin

Connaissant la probabilité de rupture d'un lien p_t, on souhaite calculer la probabilité de rupture d'un chemin q_t. La probabilité de rupture d'un chemin dépend de sa longueur. La

probabilité qu'un lien ne soit pas coupé est $1 - p_t$. Si les ruptures sont indépendantes, la probabilité qu'un chemin ne se rompe pas est $(1 - p_t)^n$, où n est le nombre de liens sur le chemin.

La probabilité qu'un chemin soit rompu pendant un intervalle t est alors

$$q_t(n) = 1 - (1 - p_t)^n$$

L'hypothèse forte sur l'indépendance des ruptures de liens est confirmée par simulation dans la section 4.4.

4.3.4.3 Fréquence de rupture d'un chemin

On appelle la *fréquence de rupture de chemin* le nombre de ruptures par seconde, notée $f_b(n)$, où n est la longueur du chemin. La fréquence de rupture en fonction de la probabilité de rupture d'un chemin est

$$f_b(n) = \frac{q_t(n)}{t}$$

Soient n_{avg} le nombre moyen de nœuds relais de la source à la destination, R le rayon de couverture d'un nœud, alors

$$n_{avg} = \frac{L}{3R}$$

si on considère que la distance moyenne entre la source et la destination est $\frac{1}{3}L$, où L est la longueur de l'autoroute.

Soit m_r un message envoyé lors de la rupture d'un chemin, nous avons

$$f(m_r) = f_b(n_{avg}) = \frac{q_t(n_{avg})}{t}$$

Alors, la fréquence d'envoi des messages de signalisation, RREQ, RREP et RERR, d'un protocole réactif est

$$f(RREQ) = f(RREP) = f(RERR) = f_b(n_{avg})$$

4.3.5 Fréquence f_r de relocalisation pour un protocole géographique

Un protocole géographique est également sensible à la dynamicité du réseau. En effet, lorsque le paquet atteint le nœud le plus proche de la position du destinataire (position initialisée par la source), alors le nœud vérifie qu'il est le destinataire (grâce à son identifiant). S'il n'est pas le destinataire, alors cela signifie que la position de la destination a changé ; le

paquet n'a pas atteint la destination. La source doit alors envoyer une nouvelle requête de localisation LREQ, ce qui induit de l'overhead.

Nous devons connaître la fréquence d'envoi de cette requête. Les véhicules sur autoroute ont une vitesse moyenne que nous pouvons facilement déterminer. En cas de fluidité du trafic, cette valeur est égale à la vitesse cible moyenne des véhicules et est proche de la limitation de vitesse sur l'autoroute. Nous notons cette vitesse moyenne v_{avg}.

L'écart maximal entre la position du destinataire indiquée par la source et la position réelle de la destination doit être inférieur au rayon de couverture radio d'un nœud pour que le paquet puisse atteindre sa destination. Alors la *fréquence de relocalisation* est

$$f_r = \frac{v_{avg}}{R}$$

Ainsi, la fréquence d'envoi d'un message LREQ et LREP d'un protocole de localisation est

$$f(LREQ) = f(LREP) = f_r = \frac{v_{avg}}{R}$$

4.3.6 Résultats sur le nombre de messages de signalisation générés pour chaques protocoles

Le tableau 4.1 précise les valeurs des principaux paramètres nécessaires pour évaluer l'overhead (1) d'un protocole topologique réactif, (2) d'un protocole de service de localisation glouton et (3) d'un protocole de service de localisation géographique GHLS (basé sur un protocole *rendez-vous*). Pour chaque protocole, nous donnons : le type de transmission, le nombre de messages par seconde $f(m)$. Nous considérons deux paramètres constants des protocoles : C_{to} et U_L. C_{to} est la durée de validité de la position géographique d'un voisin. U_L est la fréquence d'envoi de la position géographique vers le nœud serveur intermédiaire dans un protocole de localisation de type "rendez-vous".

Lorsque nous avons abordé le protocole GHLS dans le sous-paragraphe 2.1.3.5, nous avons mentionné que le serveur de localisation peut transmettre la position d'un nœud auquel il est associé à un autre nœud voisin s'il s'éloigne trop de la position associée à l'identifiant du nœud. Dans un réseau de véhicules, nous n'avons pas besoin de cette fonctionnalité car la vitesse des véhicules est quasi constante et élevée. Dans ce cas, seul un message LUpdate envoyé suffisamment régulièrement est nécessaire. Cependant, dans le cas d'un réseau dont les nœuds ont des vitesses variables et faibles, un message LUpdate est envoyé plus rarement.

Nous remarquons que le protocole géographique réactif n'est pas présent. En mode réactif, l'envoi de la position d'un nœud à ses voisins n'est pas effectué à intervalles réguliers, mais seulement quand un nœud intermédiaire souhaite transmettre un message. Donc, si la transmission est en Point-à-point voisinage et le protocole géographique est réactif, nous prenons alors en considération le fait que chaque voisin d'un nœud relais envoie un message contenant sa position.

Nous pouvons ainsi, à partir du modèle qui vient d'être décrit, calculer pour chaque protocole le nombre de messages de signalisation en fonction de la taille de l'autoroute.

Topologique Réactif		
Type de Message m	**Mode de transmission**	$f(m)$
RREQ	Diffusion	$f_b(n_{avg})$
RREP	Point-à-point	$f_b(n_{avg})$
RERR	Point-à-point	$f_b(n_{avg})$

Géographique (proactif)		
Type de Message m	**Mode de transmission**	$f(m)$
Hello	Diffusion	$\frac{1}{C_{to}}$

Service de localisation glouton		
Type de Message m	**Mode de transmission**	$f(m)$
Avec CBF		
LREQ	Diffusion	f_r
LREP	Point-à-point	f_r
Sans CBF		
LREQ	Diffusion	f_r
LREP	Point-à-point voisinage	f_r

Location service avec protocole à rendez-vous		
Type de Message m	**Mode de transmission**	$f(m)$
Avec CBF		
LREQ	Point-à-point	f_r
LREP	Point-à-point	f_r
LUpdate	Point-à-point	$\frac{1}{U_L}$
Sans CBF		
LREQ	Point-à-point voisinage	f_r
LREP	Point-à-point voisinage	f_r
LUpdate	Point-à-point voisinage	$\frac{1}{U_L}$

TABLE 4.1 – Transmission et fréquence des messages générés par les protocoles topologiques et géographiques.

Par exemple, pour évaluer le nombre de messages générés pas seconde de type RREQ :

$$\begin{aligned} N(RREQ) &= u(RREQ).d(RREQ).l.f(RREQ) \\ &= d_t l \frac{q_t(n_{avg})}{t} \end{aligned}$$

Nous déduisons le nombre de messages envoyés par seconde pour chaque protocole (Tableau 4.2).

Le seul paramètre que nous ne pouvons pas déterminer est la fréquence de rupture de chemin qui dépend directement de p_t. Dans la suite, nous allons évaluer p_t à partir de simulations, puis vérifier si le modèle pour déterminer la fréquence de rupture de chemin à partir de p_t est valide.

Topologique Réactif	$N_{Reactif} = (d_t + \frac{1}{2R})l\frac{1-(1-p_t)^{navg}}{t}$
Géographique (proactif)	$N_{Geo} = d_t l \frac{1}{C_{to}}$
Service de localisation glouton	$N_{Glouton} = \frac{4}{3} d_t l f_r$
Service de localisation glouton avec CBF	$N_{GloutonCBF} = (d_t + \frac{1}{3R})l f_r$
Service de localisation avec protocole à rendez-vous	$N_{Rdv} = (\frac{2}{3} f_r + \frac{1}{3U_L})l d_t$
Service de localisation avec protocole à rendez-vous et CBF	$N_{RdvCBF} = (\frac{2}{3} f_r + \frac{1}{3U_L})l \frac{1}{R}$

TABLE 4.2 – Nombre de messages générés pour chaque protocole

4.4 Évaluation de la fréquence de rupture de chemin

Pour calculer le nombre de messages d'overhead générés par un protocole, nous avons besoin de la fréquence d'envoi d'un message de signalisation par la source (ou la destination). Pour certains types de message, cette fréquence dépend de la mobilité des nœuds.

Pour un message de type LREQ ou LREP du protocole géographique, nous avons pu déterminer cette fréquence. Elle correspond à la fréquence de relocalisation. Malheureusement, pour les messages de type RREQ, RREP ou RERR d'un protocole réactif, nous devons évaluer la probabilité de rupture de lien par simulation.

Nous avons proposé un modèle pour déterminer la fréquence de rupture de chemin à partir de la probabilité de rupture d'un lien. Nous allons d'abord déterminer cette probabilité de rupture en fonction de la densité de véhicules, puis vérifier l'hypothèse que les ruptures de lien d'un même chemin sont indépendantes en confrontant le modèle aux valeurs obtenues par simulation. Nous commençons par décrire nos outils de simulation.

4.4.1 Méthode d'évaluation par simulation

Pour évaluer la probabilité et la fréquence de ruptures, nous utilisons un simulateur qui comporte deux parties : la simulation du déplacement des véhicules et la simulation du réseau. Le simulateur de trafic routier est celui expliqué dans le paragraphe 3.2.2 concernant les modèles de mobilité.

4.4.1.1 Simulation du réseau

La simulation du réseau est modélisé comme un graphe. L'ensemble des nœuds du graphe sont les véhicules. Un lien existe entre deux nœuds si et seulement si la distance entre les deux nœuds est inférieure à R. En d'autres termes, il existe une arête entre le nœud i et j ssi $\|x_i - x_j\| < R$. Nous obtenons ainsi le graphe du réseau. Le simulateur actualise les liens du graphe toutes les t secondes en fonction de la mobilité des nœuds, ajoutant et supprimant ainsi de nouveaux liens entre les nœuds à intervalles réguliers. Ce *graphe dynamique* nous permet d'estimer la durée de vie des liens et des chemins. Dans le simulateur, la durée de vie d'un lien est calculée à partir de l'inter-distance et de la vitesse relative entre

les deux nœuds. Soit v_i et v_j la vitesse respective des nœuds i et j aux deux extrémités des liens. Soit x_i et x_j la position respective des nœuds sur l'autoroute. On considère l'autoroute comme une ligne, donc sur une seule dimension. Soit $d_{i,j} = x_j - x_i$, la distance entre deux nœuds. Soit $t_{i,j}$ la durée de vie du lien entre le nœud i et j. En supposant les vitesse v_i et v_j constantes, et $x_i < x_j$, nous pouvons déterminer la durée de vie d'un lien :

Si $v_i < v_j$:

$$t_{i,j} = \frac{R - d_{i,j}}{v_j - v_i}$$

Si $v_i > v_j$:

$$t_{i,j} = \frac{R + d_{i,j}}{v_i - v_j}$$

Sinon :

$$\lim_{v_i \to v_j} (t_{i,j}) = \infty$$

La durée de vie d'un chemin est calculée comme étant la plus petite durée de vie des liens du chemin. La probabilité de rupture d'un lien p_t est la proportion des liens ayant une durée de vie inférieure à t. De même, pour la probabilité de rupture d'un chemin q_t est la proportion des chemins ayant une durée de vie inférieure à t.

4.4.2 Évaluation de la probabilité de rupture p_t d'un lien

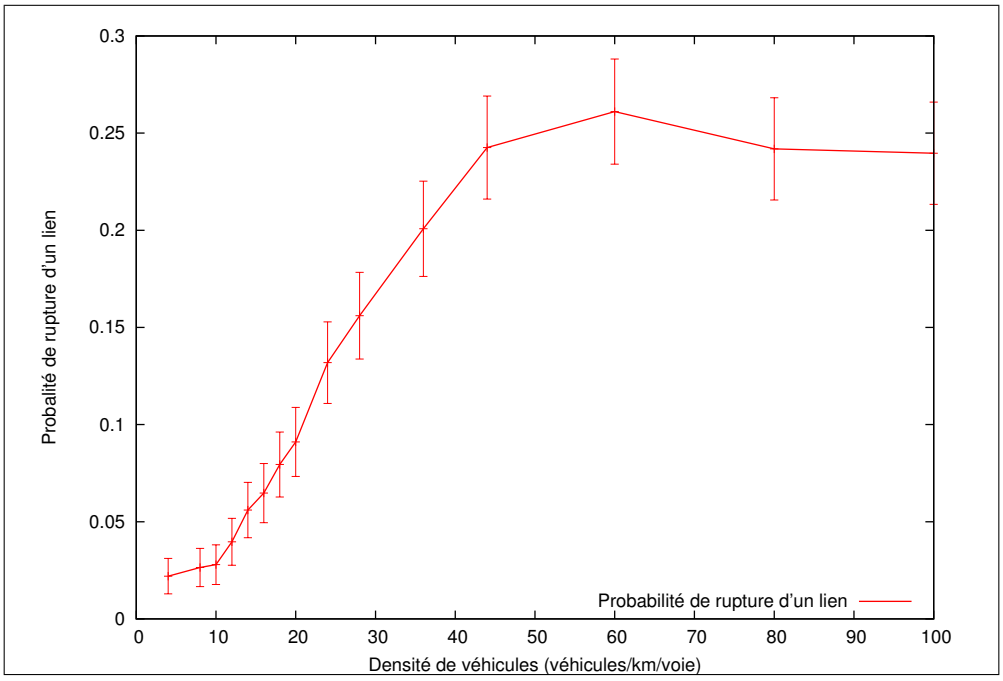


Figure 4.3 – Probabilité de rupture d'un lien $p_{1.5}$ en fonction de densité de véhicules

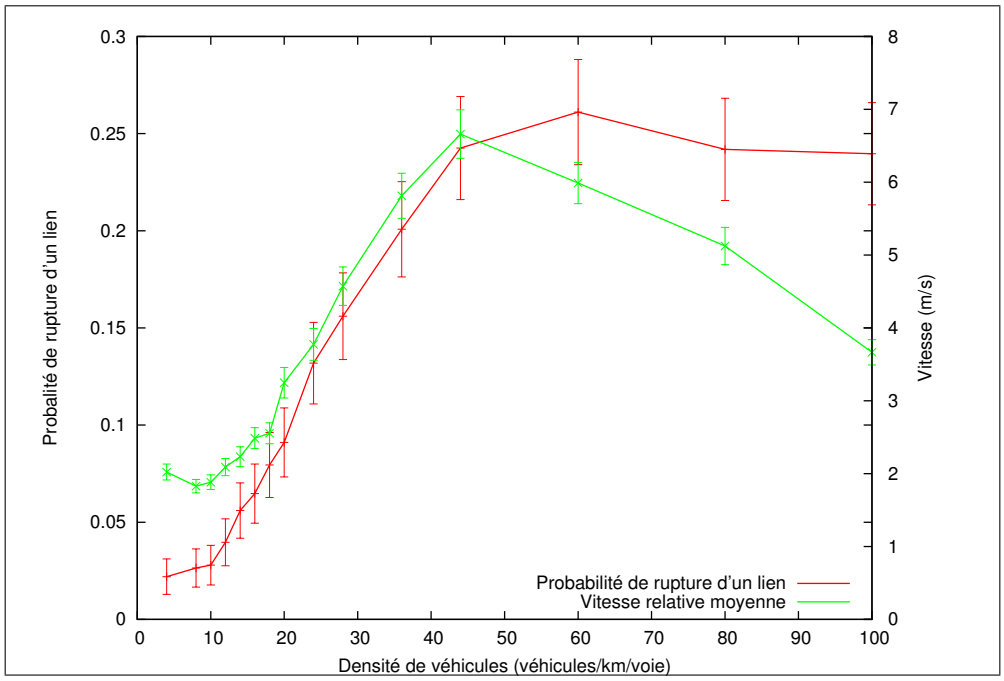


Figure 4.4 – Comparaison entre la variation de la probabilité et la vitesse relative moyenne

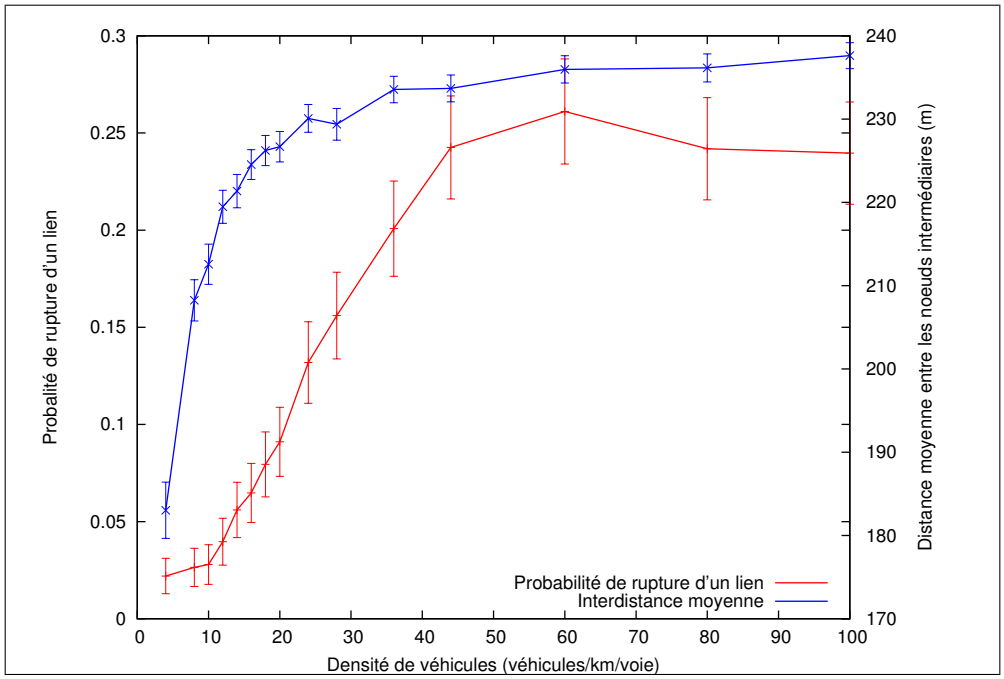


Figure 4.5 – Comparaison entre la variation de la probabilité et la distance moyenne entre les nœuds intermédiaires

Pour compléter le modèle, nous avons besoin d'évaluer la probabilité de rupture d'un lien qui est représentée sur la Figure 4.3. Le paramètre $t = 1,5s$ est défini comme étant le temps d'attente d'un accusé de réception avant de considérer que le lien est rompu. Nous voyons que plus la densité est importante plus la probabilité de rupture d'un lien est grande.

Ce résultat non intuitif est la conséquence du choix du chemin le plus court en nombre de relais. En effet, plus le chemin est court en nombre de relais, plus la distance D entre deux nœuds intermédiaires tend vers R (250 m dans nos simulations), car la distance euclidienne entre la source et la destination est la même quelque soit le nombre de relais ou la valeur de R. Donc, si on considère que la vitesse relative entre deux nœuds est constante pendant la durée de vie d'un lien, alors la durée de vie tend vers zéro quand D tend vers R.

Nous déduisons donc que plus le chemin est court en nombre de relais, plus la durée de vie d'un lien du chemin est petite. Or, si la densité augmente, le nombre de véhicules augmente et donc le nombre de chemins possibles augmente. La probabilité de trouver un chemin court (et donc une distance moyenne entre les nœuds plus proches de R) est plus grande, et donc la durée de vie moyenne diminue. De plus, la probabilité de rupture d'un lien est proportionnelle à la durée de vie moyenne, donc nous concluons que plus la densité augmente, plus la probabilité de rupture est grande. La Figure 4.5 confirme ce résultat.

La Figure 4.4 met en corrélation la probabilité de rupture de chemin et la vitesse relative moyenne des véhicules. La vitesse relative moyenne étant la moyenne des différences de vitesses entre chaque paire de nœuds voisins sur le chemin. Au dessus d'une densité de 40 véhicules par km et par voie nous remarquons une baisse significative de cette vitesse. Ceci est la conséquence de la saturation des voies, le nombre de véhicules ayant atteint la capacité optimale de l'autoroute. Cependant, malgré la baisse de la vitesse relative moyenne, la probabilité de rupture de chemin reste constante, la distance moyenne entre les nœuds influant d'avantage sur cette probabilité par rapport à la vitesse relative moyenne.

4.4.3 Confrontation du modèle de probabilité de ruptures de chemin

Nous allons vérifier l'hypothèse de l'indépendance de la rupture des liens d'un chemin. Pour cela, nous évaluons la probabilité de rupture d'un chemin $q_t(n)$ par simulation. La Figure 4.6 montre les courbes réalisées à partir du modèle pour différentes valeurs de la densité du trafic routier (d_t) en fonction de la longueur des chemins. Ces courbes sont comparées aux valeurs calculées par simulation. Nous remarquons que les points suivent la même tendance que la courbe théorique évaluée à partir du modèle. Le modèle est d'autant plus fiable, que la densité est forte.

De même, la Figure 4.7 évalue le modèle de fréquence de rupture d'un chemin $F_{rupture(n_{avg})}$. Nous remarquons que le modèle surestime la fréquence de rupture de chemin par rapport aux résultats de simulation, même si la tendance correspond.

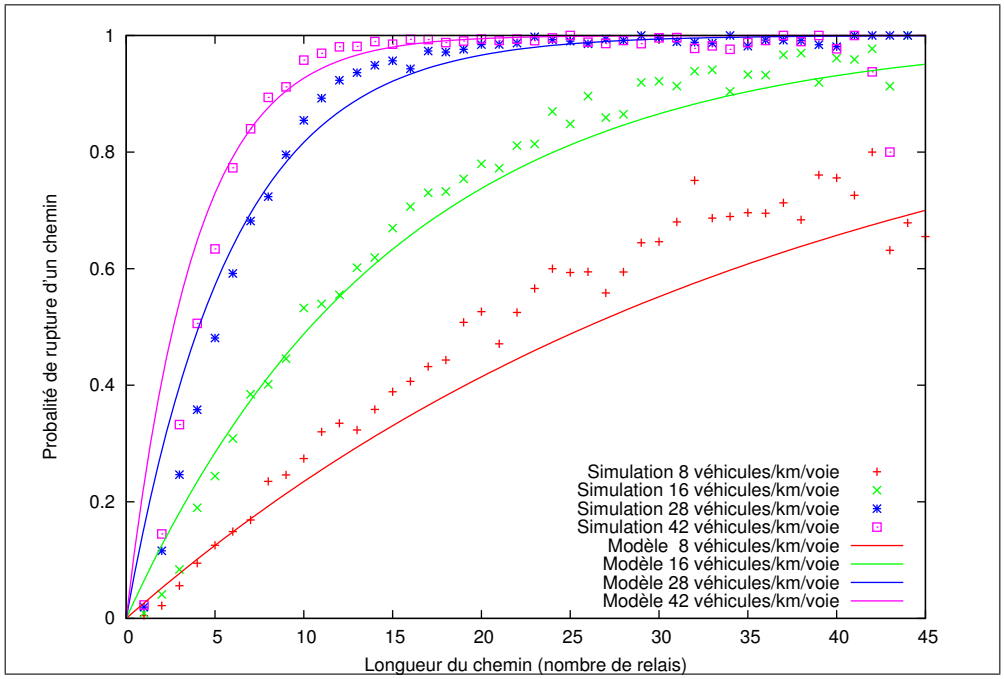


Figure 4.6 – Probabilité de rupture d'un chemin en fonction de sa longueur et de la densité de véhicules.

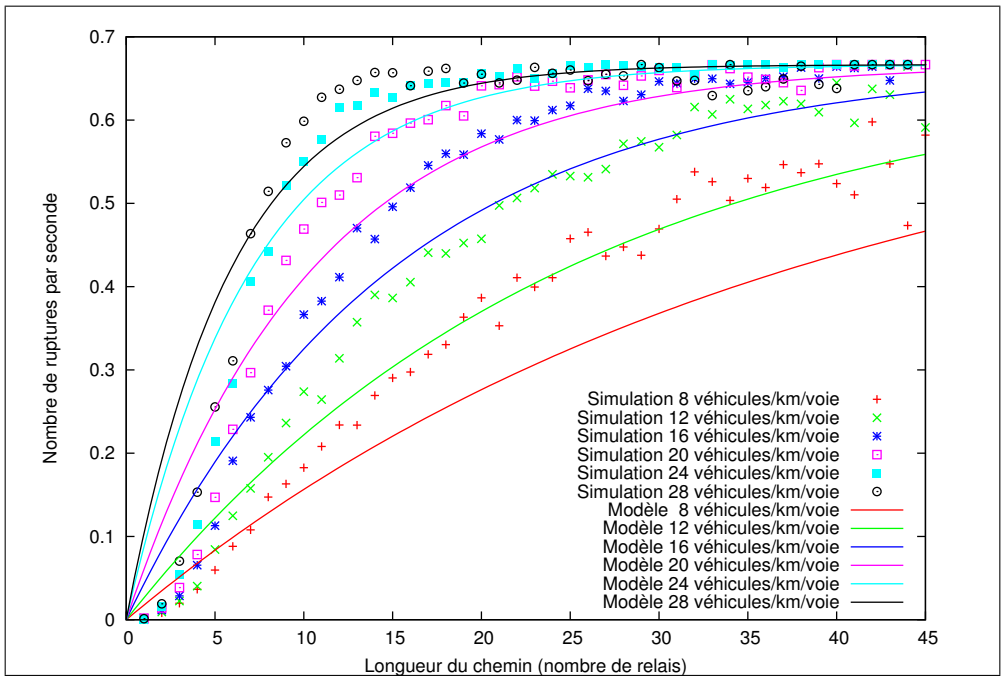
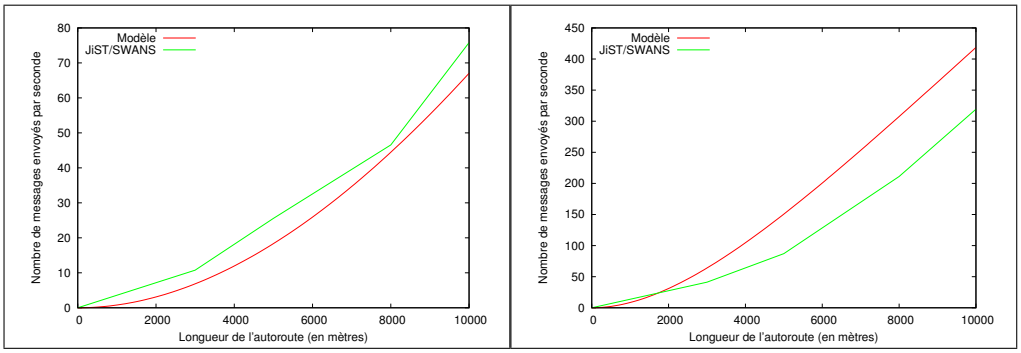


Figure 4.7 – Fréquence de rupture d'un chemin en fonction de sa longueur selon plusieurs densités de véhicules.

4.4.4 Validation du modèle quantitatif pour DSR

Nous avons confronté les résultats de notre modèle quantitatif aux résultats par simulation à l'aide du simulateur JIST/SWANS décrit en section 3.2. Nous avons utilisé les paramètres par défaut de DSR, avec l'implémentation initiale de DSR de JIST/SWANS corrigé de ses bugs. Le temps de simulation est fixé à 30 s et le débit de données à 512 kbit/s pour une seule communication. Pour une densité de 10 véhicules par voie et par kilomètre, le nombre de messages de contrôle (RREQ, RREP, RERR) envoyés par seconde (overhead), correspond à la prédiction de notre modèle, bien que légèrement sous-évaluée (Figure 4.8a). En revanche le modèle sur-évalue l'overhead pour une densité de 24 véhicules par voie et par kilomètre (Figure 4.8b), malgré une tendance similaire par rapport à la simulation.

Ce résultat montre que notre modèle est crédible car les valeurs calculées correspondent à celles provenant d'un simulateur de réseau tel que JIST/SWANS.



(a) Densité de 10 (b) Densité de 24

Figure 4.8 – Comparaison entre le nombre de messages de contrôle évalué à l'aide du modèle et de JIST/SWANS

4.5 Résultats quantitatifs

Paramètre	Symbole	Valeur
Vitesse moyenne d'un véhicule	$v_{avg.}$	33 $m.s^{-1}$
Durée de validité de la cache de positions	C_{to}	30 s
Temps avant de considérer un lien comme rompu	t	1.5 s
Intervalle entre deux actualisations de la position	U_L	30 s
Rayon de couverture d'un nœud	R	250 m

TABLE 4.3 – Paramètres fixés

Dans cette section, nous appliquons le modèle que nous avons décrit en section 4.3. En fonction des formules et des résultats sur l'évaluation de la probabilité de rupture de lien (§ 4.3.4), nous calculons le nombre de messages de signalisation pour les protocoles étudiés afin de les comparer. Il est facile de faire varier les paramètres à l'infini, le coût du calcul

étant très faible par rapport à une simulation classique. Plusieurs résultats sont exhibés en faisant varier la densité et la longueur de l'autoroute. Le Tableau 4.3 indique la valeur des paramètres fixés.

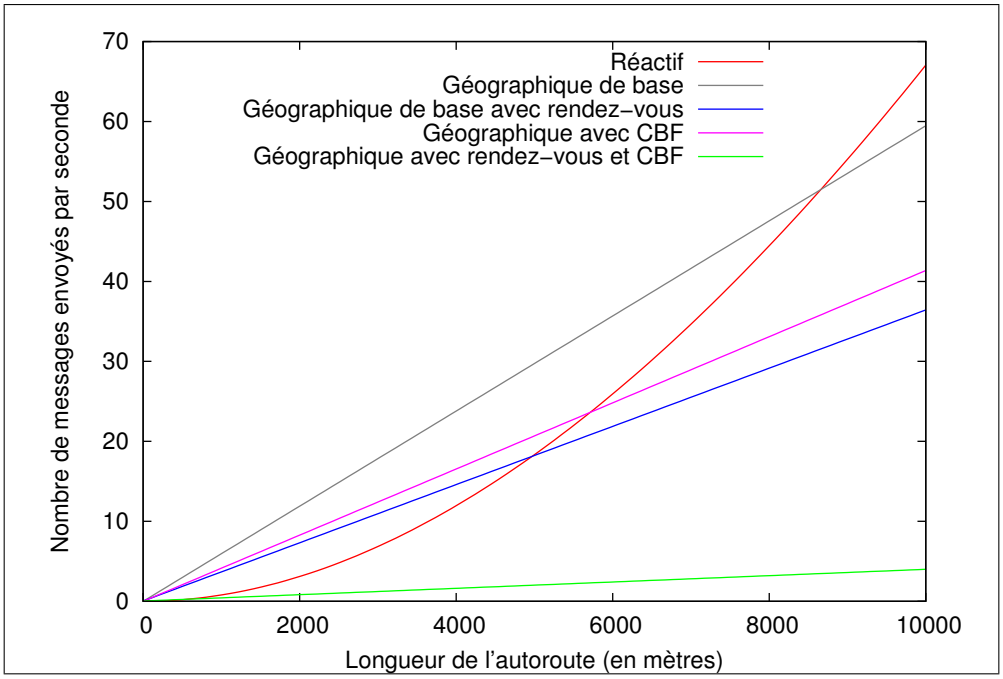


Figure 4.9 – Nombre de messages de signalisation générés en fonction de la taille de l'autoroute pour une densité de 10 véhicules/km/voie.

La Figure 4.9 et 4.10 compare le nombre de messages de signalisation générés pour chacun des protocoles étudiés. La densité est de 10 et 24 véhicules par km par voie. Nous faisons varier la taille de l'autoroute. Pour une densité de 24 véhicules par km par voie, le graphique montre qu'un protocole réactif génère d'avantage de messages que les protocoles géographiques, et qu'il est plus sensible à la longueur de l'autoroute. Ceci s'explique par une reconstruction fréquente de chemin qui oblige à transmettre des messages en mode diffusion (RREQ). L'écart est moins significatif pour une densité plus faible, ici 10 véhicules par km par voie, surtout en dessous d'une longueur d'autoroute de 5 km.

Quelque soit la densité ou la longueur de l'autoroute, nous voyons également le bénéfice de l'utilisation de CBF avec un protocole à rendez-vous. Cette combinaison permet dans le cadre d'un protocole géographique de n'avoir aucun message transmis en mode diffusion et de ne pas avoir à transmettre de message de localisation des voisins. Par conséquent, tous les messages sont transmis en mode point-à-point, rendant le protocole peu sensible au nombre de véhicules, et donc à la longueur de l'autoroute et à la densité de véhicules.

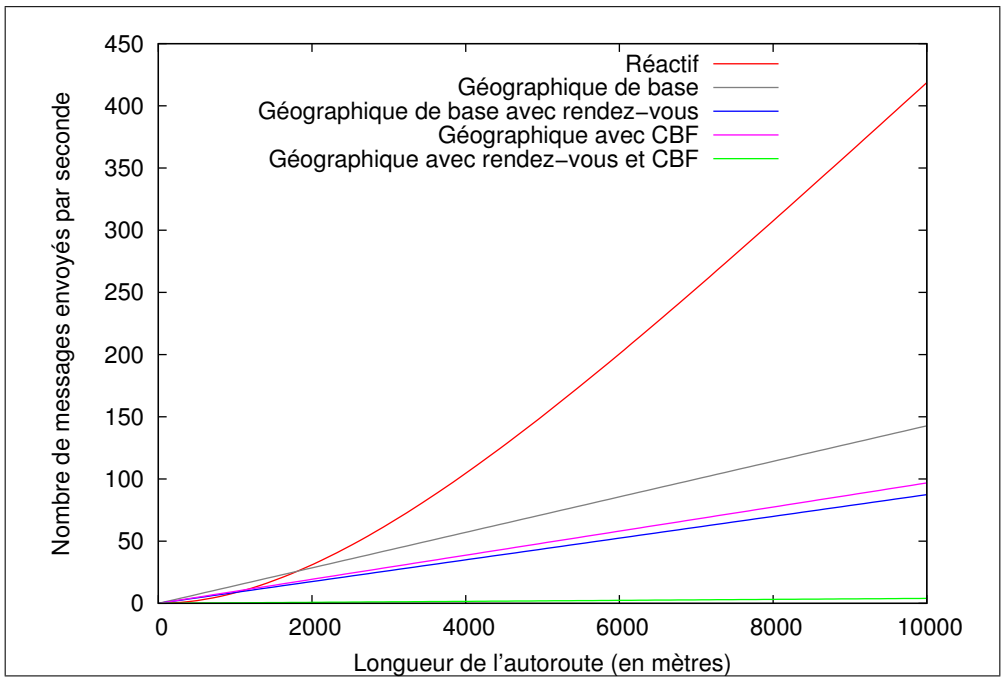


Figure 4.10 – Nombre de messages de signalisation générés en fonction de la taille de l'autoroute pour une densité de 24 véhicules/km/voie.

Conclusion

Notre objectif était de comparer les protocoles de routage adaptés aux réseaux de véhicules de manière indépendante de tout scénario de trafic ou de toute mise en œuvre d'un protocole dans un simulateur. Pour cela, à l'aide d'une modélisation fine du processus de génération des messages de signalisation, nous avons caractérisé l'overhead et le nombre de messages de signalisation générés par un protocole de routage réactif ainsi que par un protocole de routage géographique dont l'overhead dépend essentiellement du service de localisation. D'une part, ces métriques dépendent fortement de la dynamique du réseau (mobilité et densité) entraînant des changements de topologie ou de position et donc des reroutages, c'est pourquoi nous avons eu besoin d'évaluer la fréquence de rupture de chemin. A l'aide du modèle et de ces résultats, nous avons comparé deux grandes familles de protocoles ad hoc : géographique et réactif. Nous concluons que l'utilisation d'un protocole géographique avec CBF (Contention-Based Forwarding) et un protocole à rendez-vous pour la localisation permet un meilleur passage à l'échelle d'un réseau de véhicules sur autoroute.

Chapitre 5

Protocole de routage pour un réseau ad hoc hybride

Un réseau ad hoc hybride permet au réseau de véhicules non seulement d'assurer la connectivité du réseau mais aussi d'améliorer son passage à l'échelle. Dans le premier cas, les fragments du réseau seront ainsi reliés les uns aux autres et de plus, l'infrastructure réseau permet aussi d'accéder à Internet. Pour le passage à l'échelle, l'infrastructure fixe dispose d'une plus grande bande passante que la partie ad hoc, permettant à la fois un réseau plus long mais aussi avec plus de nœuds.

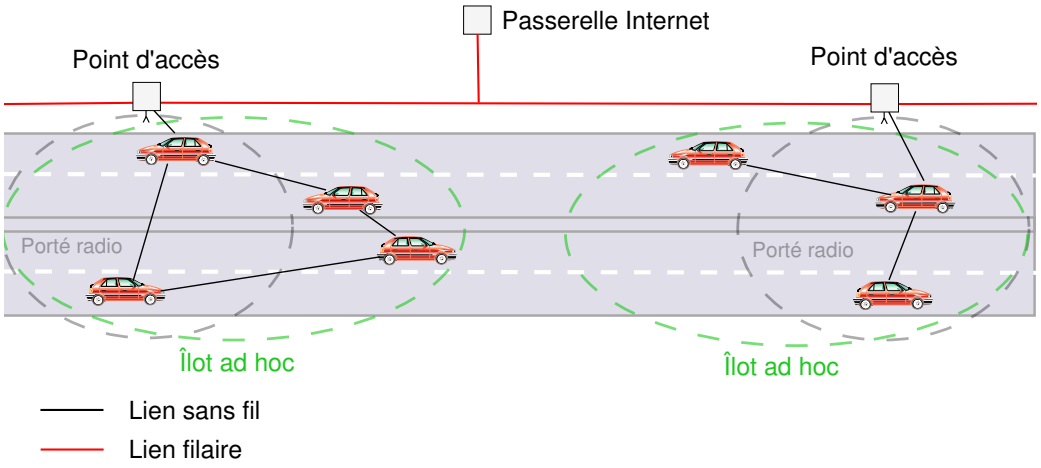


Figure 5.1 – Architecture d'un réseau ad hoc hybride dans un contexte autoroutier

La Figure 5.1 illustre une architecture d'un réseau ad hoc hybride dans un contexte autoroutier. Les points d'accès sont répartis le long de la route avec une certaine densité. Chaque point d'accès est relié à deux points d'accès voisins : le précédent et le suivant. Un nœud accède à un point d'accès en un ou plusieurs sauts. Dans notre étude nous nous inter-

essons aux communications entre un nœud source mobile et un nœud destinataire également mobile, les deux nœuds étant des véhicules sur l'autoroute. Les paquets transitent alors par plusieurs points d'accès si la destination est éloignée. Nous appelons *îlot ad hoc* d'un AP *a*, le réseau ad hoc formé par les nœuds dont le plus petit chemin vers *a* est inférieur au TTL.

Un protocole réactif, tel que DSR, n'est pas adapté aux réseaux ad hoc hybrides. Un DSR classique peut fonctionner dans un réseau ad hoc hybride, mais les points d'accès étant vus comme un nœud quelconque par DSR, les RREQ sont diffusés sur tout le réseau, saturant rapidement le réseau lorsque le nombre de nœuds augmente. Une proposition [30] consiste à considérer un point d'accès comme un *proxy route reply*. Le point d'accès répond à une requête RREQ à la place de la destination, en donnant le chemin entre la source et le point d'accès. C'est la solution que nous avons adoptée. Par contre, une telle solution ne gère pas la mobilité : lorsqu'un nœud change de point d'accès, l'ensemble des points d'accès doivent actualiser leur table de routage. C'est le rôle du processus d'enregistrement d'un nœud. De plus, un protocole de découverte de points d'accès doit être également incorporé aux nœuds pour qu'ils découvrent les points d'accès accessibles à un ou plusieurs sauts. Nous proposons une extension de DSR permettant la communication V2V dans un réseau ad hoc hybride. Cette extension est basée sur les techniques liées aux réseaux ad hoc hybrides décrites en section 2.2.

Dans ce chapitre nous décrivons dans un premier temps l'intégration de ces techniques à DSR, puis nous évaluons le protocole DSR étendu à l'aide de simulations.

5.1 Mécanismes du protocole

Adapter DSR à un réseau ad hoc hybride nécessite deux mécanismes supplémentaires : la découverte de point d'accès et l'enregistrement auprès du point d'accès. Le premier a pour rôle de connaître l'ensemble des points d'accès accessibles en un ou plusieurs sauts, le second permet à un nœud mobile d'être localisé sur le réseau par un point d'accès.

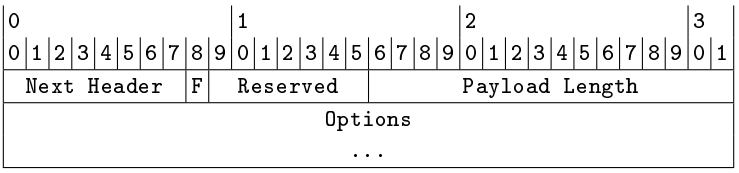


Figure 5.2 – Format d'en-tête des messages de DSR

D'après la RFC de DSR [67], l'en-tête DSR est composé d'une liste d'options spécifiques à DSR (Figure 5.2). Elle doit suivre immédiatement l'en-tête IP. Cette en-tête des options de DSR est composée des champs suivant :

- Next Header : identifie le type d'en-tête suivant immédiatement l'en-tête des options DSR.
- Flow State Header (F) : non utilisé
- Payload Length : la taille de l'en-tête des options sans les 4 premiers octets.
- Options : contient une ou plusieurs options.

Chaque option commence par deux champs de 1 octet : le type de l'option (*Option Type*), la taille de l'option (*Opt Data Len*). Le type de l'option est un identifiant unique affecté à chaque option de DSR. Par exemple, l'option RREQ a l'identifiant 1. La taille de l'option est la taille des informations contenues dans l'option. La Figure 5.3 représente l'option RREQ. Le champ *Target Address* contient la destination recherchée et les champs *Address[i]*, les adresses des nœuds du chemin vers la source.

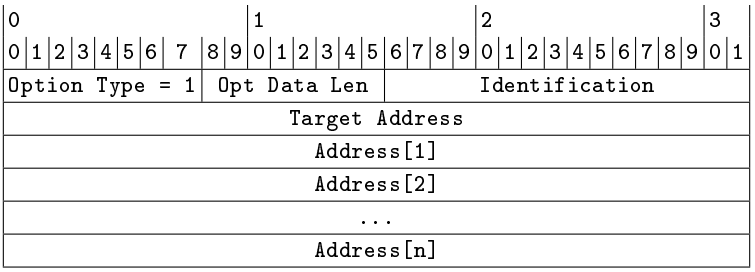


Figure 5.3 – Format de l'option RREQ

5.1.1 Découverte de points d'accès

Nous avons décrit plusieurs méthodes de découverte de points d'accès dans le paragraphe 2.2.2 : proactive, réactive et hybride. Nous avons intégré dans notre protocole les méthodes proactive et réactive.

La méthode proactive a l'avantage de ne pas nécessiter l'envoi de requête de découverte de point d'accès par chaque nœud mobile du réseau. La méthode réactive a l'avantage d'éviter la diffusion de messages dans le réseau lorsque peu de nœuds communiquent ou en cas de faible mobilité ; la fréquence de diffusion d'une requête de découverte de points d'accès étant proportionnelle à la mobilité du réseau. Un nœud communiquant est un nœud envoyant et recevant des données, par conséquent il doit trouver et maintenir une route vers un point d'accès. Le choix d'une méthode réactive ou proactive dépend à la fois de la mobilité du réseau, de la proportion de nœuds communiquants et de la densité de nœuds mobiles.

Avec un réseau de véhicules sur autoroute couvert par suffisamment de points d'accès, la vitesse d'un nœud mobile est importante : un véhicule roulant à 120 km/h sur une autoroute change de point d'accès toutes les 15 s si celui-ci a une couverture radio de 250 m. La densité de nœuds peut être également importante. Nous pensons donc qu'une méthode proactive de découverte de points d'accès est mieux adaptée dans un réseau de véhicules sur autoroute. Nous vérifierons cette hypothèse par simulation, c'est pourquoi nous avons eu besoin d'implémenter à la fois la méthode proactive et réactive.

5.1.1.1 Méthode proactive

Notre implémentation de la découverte proactive nécessite l'ajout d'un message APADV (Access Point Advertisement). Ce message est diffusé à intervalles réguliers par chaque point

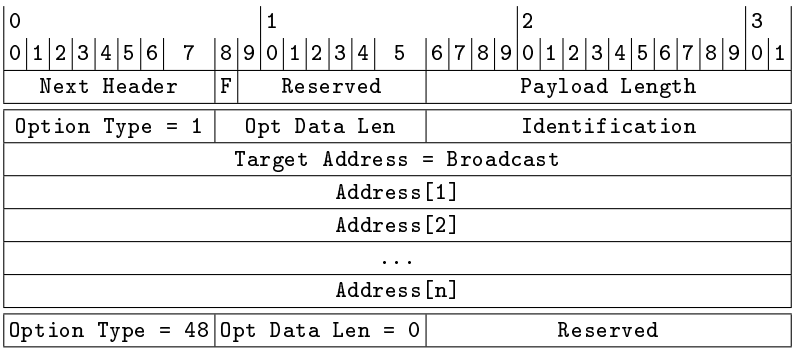


Figure 5.4 – En-tête DSR d'un message APADV avec la découverte de point d'accès proactive

d'accès.

Un message APADV est construit avec l'option RREQ suivie d'une autre option APADV (type 48) indiquant l'annonce d'un point d'accès. Cette option ne contenant aucune autre information, la taille de l'option est zéro. Ainsi, un message APADV peut être diffusé par l'implémentation de DSR classique ignorant l'option ajoutée. Un message APADV contient donc l'adresse IP du point d'accès et le TTL via les champs source correspondant de l'en-tête IP ; et le chemin parcouru par le message via l'en-tête RREQ. La cible dans l'en-tête RREQ est une adresse IP de broadcast. L'en-tête du protocole DSR pour un message APADV avec la découverte de point d'accès proactive est représentée en Figure 5.4.

Notons qu'il est possible de faire varier cet intervalle dynamiquement en fonction de la mobilité globale du réseau : plus la vitesse moyenne des véhicules est faible, plus l'intervalle d'envoi du message APADV peut être grand, sans influencer sur le choix du meilleur point d'accès par le nœud mobile. Cette méthode peut être facilement mise en œuvre ; les sociétés d'autoroute ayant accès à la vitesse moyenne des véhicules sur un tronçon donné grâce à des capteurs situés sur la route ou grâce aux données fournies par les opérateurs de téléphonie mobile sur la mobilité des téléphones mobiles localisés sur un tronçon d'autoroute.

La valeur initiale du TTL d'un message IP permet de limiter, en nombre de sauts, la diffusion d'un message. Le message APADV étant diffusé dans le réseau comme l'est le message RREQ, la valeur initiale du TTL de ce message est fonction de la densité des points d'accès, c'est-à-dire de la distance entre chaque point d'accès. Par exemple, si la couverture radio est de 250 m et qu'il existe un point d'accès tous les 2000 m, alors 4 sauts sont nécessaires pour atteindre le point d'accès le plus proche à partir du nœud mobile. Comme le chemin vers un point d'accès n'est pas forcément toujours le plus court en nombre de sauts, nous définissons le TTL initial comme étant le nombre de sauts nécessaires pour atteindre un point d'accès auquel on y ajoute deux sauts. Dans notre cas, le TTL initial est donc fixé à 6 sauts (voir résultats de simulation en section 5.2).

5.1.1.2 Méthode réactive

L'implémentation de la découverte réactive nécessite l'ajout d'un message APSEARCH. Ce message est envoyé de la même manière qu'un message APADV, sauf qu'il est envoyé par

le nœud mobile. Un message APSEARCH est construit avec l'option RREQ suivi d'une autre option APSEARCH (type 50). À la réception d'un message APSEARCH, un point d'accès répond par APADV, avec l'option RREP. Ainsi, le message APADV sera envoyé avec le chemin vers le nœud demandeur.

5.1.1.3 Table de points d'accès

Que ce soit dans les cas proactif ou réactif, un nœud mobile recevant un message APADV enregistre le point d'accès dans sa *table de points d'accès*. Chaque entrée de cette table est composée de : l'adresse IP du point d'accès, de la route vers le point d'accès, du nombre de sauts pour atteindre le point d'accès et de l'heure d'ajout de l'entrée. Une entrée de la table est supprimée si elle est considérée comme obsolète, c'est-à-dire si l'entrée n'est pas actualisée depuis un certain délai. Nous fixons ce délai égal à l'intervalle entre chaque APADV auquel nous ajoutons 50 % pour prendre en compte la variabilité du délai d'acheminement du message. Dans le cas réactif, ce délai est choisi arbitrairement. Une entrée est également supprimée s'il existe un lien rompu dans le chemin vers le point d'accès. Cette information de rupture d'un lien est obtenue par la réception d'un message RERR.

Lorsque la table des points d'accès est modifiée, le nœud mobile choisit le meilleur chemin disponible vers un point d'accès. Par défaut, le meilleur chemin est le plus court en nombre de sauts. Néanmoins, si le nœud est déjà enregistré à une AP par un chemin de longueur n, alors il change pour un meilleur point d'accès, ssi le chemin vers le nouveau point d'accès est supérieur ou égal à $n + 2$; cette restriction décrite dans [70] évite l'effet *ping-pong*.

Lors d'un changement de chemin vers le point d'accès courant, le nœud doit se ré-enregistrer à un nouveau point d'accès. C'est cette procédure que nous allons décrire dans le suite.

5.1.2 Enregistrement auprès des points d'accès

Le but de notre protocole est de réduire l'overhead en limitant la diffusion d'un message RREQ à une diffusion plus locale, tout en permettant à un nœud mobile d'atteindre au moins un point d'accès. Un point d'accès du réseau doit alors connaître le chemin pour accéder au nœud mobile afin de répondre à la place du nœud destinataire par un RREP. Le chemin contenu dans le RREP est alors la combinaison du chemin enregistré dans le point d'accès pour atteindre le nœud destinataire et le chemin inverse parcouru par le RREQ jusqu'au point d'accès. L'enregistrement est alors nécessaire pour la communication descendante : du point d'accès vers le nœud mobile. Dans l'autre sens, le processus de découverte de point d'accès suffit pour atteindre n'importe quel point d'accès.

Nous appelons AP *courante* d'un nœud mobile le chemin vers le point d'accès où est enregistré le nœud. Contrairement au réseau mobile classique où un nœud accède en un saut au point d'accès et donc l'AP courante est l'identifiant de cette AP, dans un réseau ad hoc hybride l'AP courante est un chemin vers un point d'accès. En effet, le changement de chemin n'entraîne pas forcément un changement de point d'accès où le nœud s'enregistre.

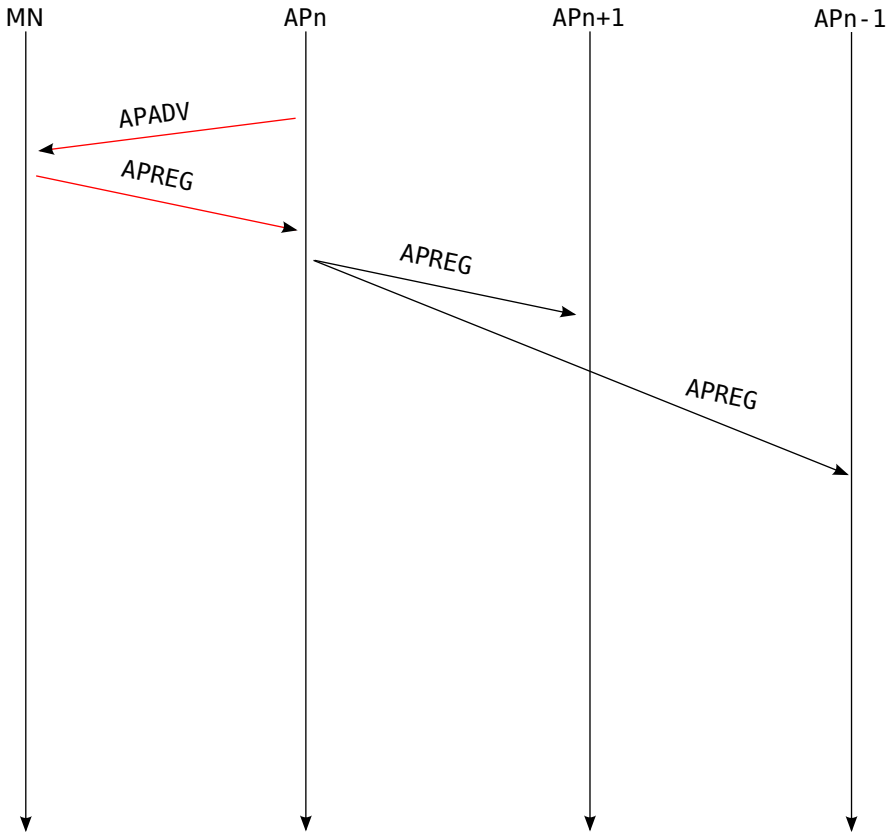


Figure 5.5 – Enregistrement de MN auprès de l'AP n. L'AP transmet l'enregistrement à ses voisins via une liaison filaire. .

L'enregistrement consiste à envoyer un message APREG par le chemin contenu dans la table de points d'accès. Le message APREG correspond à une option (type 49) associé à l'option RREP. Pour que le message soit envoyé directement au point d'accès, le chemin vers le point d'accès est contenu dans l'option RREP. Lorsque le point d'accès reçoit le message, il le traite comme un message RREP du protocole DSR original en ajoutant la route pour atteindre le nœud à son cache des routes (nommée *Route Cache* dans le protocole DSR originel [69]). De même, le processus de *gratuitous route reply* permet à l'AP de répondre à un message RREQ à la place de la destination recherchée. Le Figure 5.5 présente l'enregistrement d'un nœud mobile vers les points d'accès lors de la réception d'un message APADV avec une découverte proactive. La Figure 5.6 montre l'enregistrement lors de la réception d'un message APADV avec une méthode de découverte réactive où le nœud mobile diffuse d'abord un message APSEARCH.

Le processus décrit précédemment ne permet pas à un autre point d'accès de localiser un nœud grâce à son cache. Pour résoudre le problème nous introduisons une *synchronisation de cache*. Quand un point d'accès reçoit le message APREG, il le transmet aux points d'accès voisins, reliés directement par une liaison filaire. Les voisins font de même, diffusant l'enregistrement du nœud à tous les points d'accès. Par contre, ce message ne sera jamais

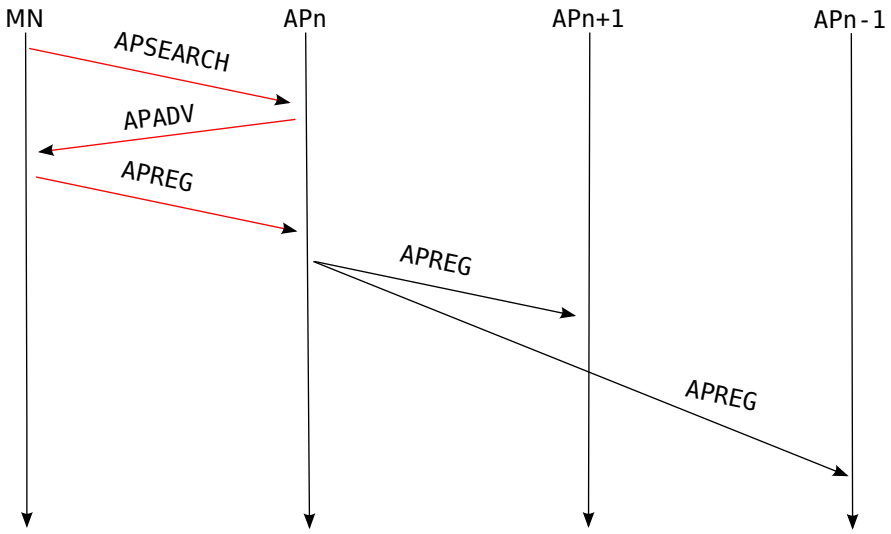


Figure 5.6 – Découverte d'AP réactive : MN envoie un message APSEARCH pour trouver les AP accessibles. Une fois l'APADV reçu, la procédure d'enregistrement est identique avec la méthode de découverte proactive. .

transmis par un point d'accès vers le réseau sans fil. Sur les Figures 5.5 et 5.6, l'AP n transmet le message APREG, permettant aux AP voisines (AP $n-1$ et $n+1$) d'ajouter à leur cache le chemin vers le nœud MN.

Lorsqu'un nœud souhaite changer de point d'accès, par exemple lorsqu'il trouve un chemin plus court vers un autre point d'accès, il doit supprimer son enregistrement. Pour cela il envoie un message RERR à l'ancien point d'accès ayant un rôle de "désenregistrement". En effet, le point d'accès en recevant le message RERR, supprime de son cache l'entrée correspondant au nœud initiateur du message.

5.1.3 Maintenance des routes

La maintenance des routes dans DSR se fait à l'aide du message RERR : quand un paquet ne parvient pas au nœud suivant (il ne reçoit pas d'accusé de réception implicite ou explicite), alors le nœud considère le lien comme mort. Il envoie alors un message RERR à la source. La source et les nœuds intermédiaires actualisent leur cache en recevant ce message. La source doit alors découvrir une autre route en envoyant un message RREQ. Dans notre protocole, la maintenance d'une route par un message RERR permet également de supprimer l'enregistrement d'un nœud auprès d'un point d'accès. Lorsqu'un nœud mobile reçoit un message RERR, il supprime de sa table les entrées dont le chemin contient le lien rompu. Si le nœud est enregistré au point d'accès via le chemin en question, il choisit alors une autre AP courante et s'enregistre. Pour la connectivité vers un point d'accès, le nœud mobile envoie à intervalles réguliers un message HELLO avec le chemin vers l'AP courante. Si l'envoi de ce message est un échec à cause d'un lien coupé sur le chemin, alors le nœud mobile reçoit un message RERR, entraînant le changement de chemin vers le point d'accès.

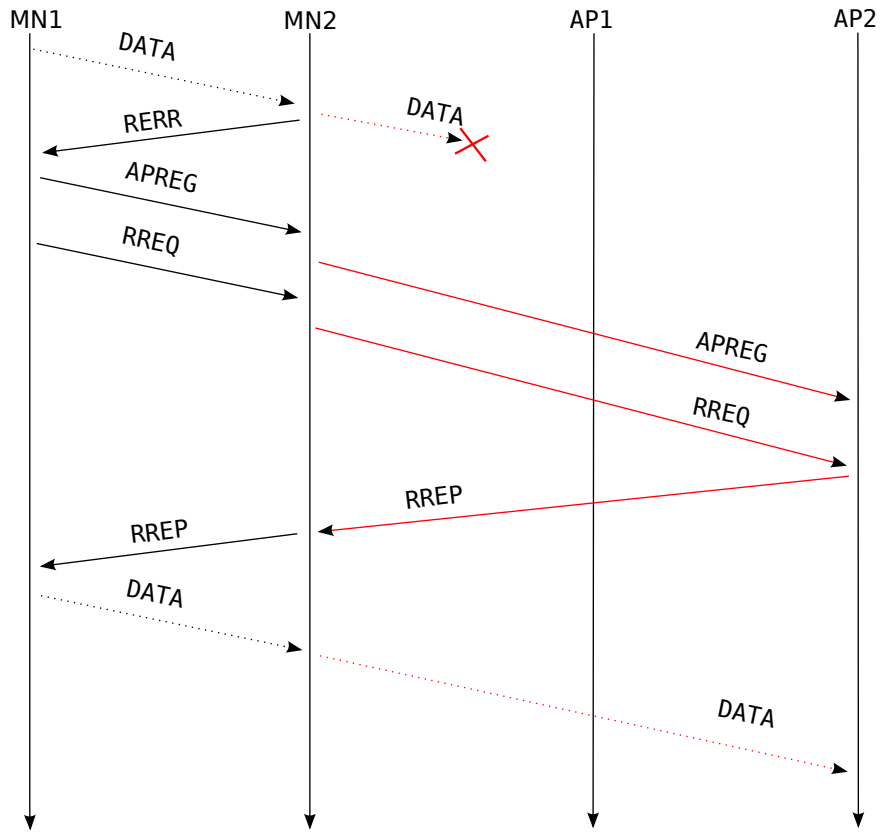


Figure 5.7 – Le chemin entre la source et l'AP est rompu. Le chemin initial vers une AP est MN1-MN2-AP1. Lorsque le lien MN2-AP1 est rompu, un nouveau chemin est trouvé : MN1-MN2-AP2. Ici, la découverte d'une AP est proactive.

Il existe deux scénarios possibles de rupture de connexion lors de l'échange de données entre une source et une destination : soit un lien du chemin vers l'AP courante de la *source* (MN1) est rompu (Figure 5.7) ou alors c'est un lien vers l'AP courante de la *destination* (MN3) qui est rompu (Figure 5.8). Nous prenons comme exemple le cas proactif : le message APSEARCH de recherche d'AP n'est alors pas nécessaire. Dans le premier cas, la source va se ré-enregistrer avec un chemin contenu dans la table de points d'accès (s'il existe), puis initier une nouvelle recherche de chemin ; dans DSR la réception d'un RERR entraînant une nouvelle recherche de chemin. Dans le second cas, la source reçoit un RERR, indiquant que le paquet de données n'a pas pu atteindre la destination. Mais ce n'est pas un lien dans le chemin vers le point d'accès courant de la source qui est en cause, donc la source n'initie pas un nouvel enregistrement, mais juste une recherche de chemin. Par contre, la destination, constatant un échec de l'envoi d'un message HELLO, initie un nouvel enregistrement vers un point d'accès accessible (ici l'AP3). L'enregistrement est propagé vers l'AP1 permettant au nœud source (MN1) de relocaliser la destination.

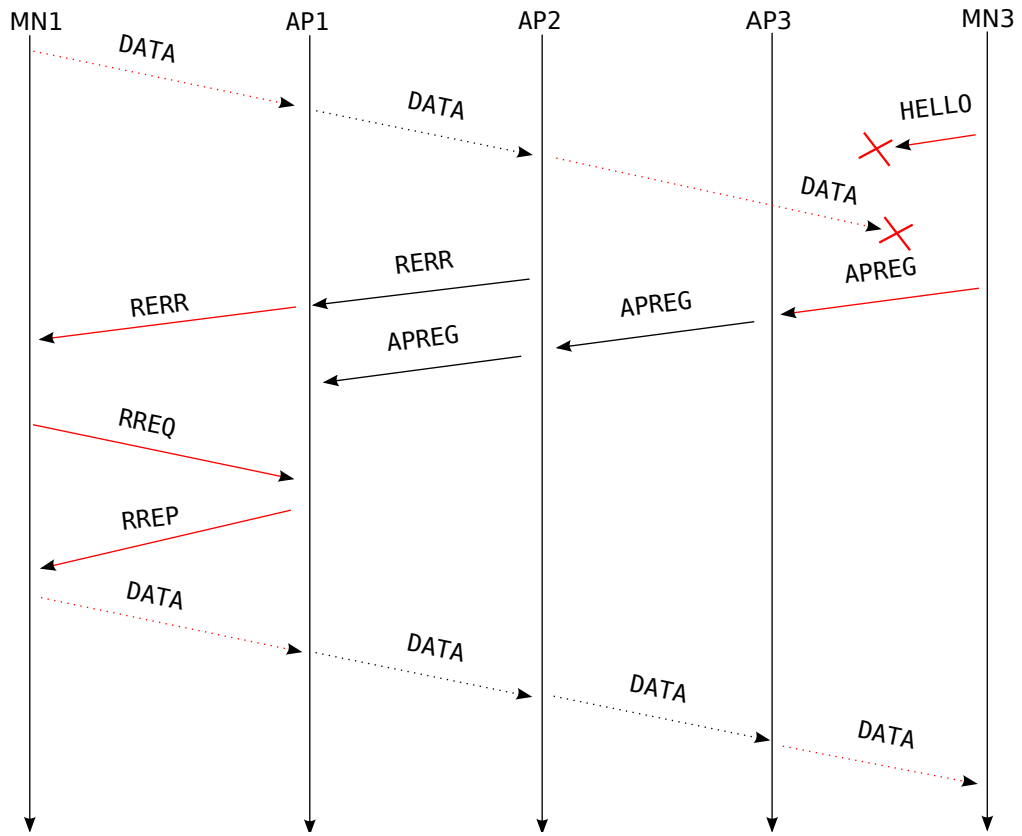


Figure 5.8 – Le chemin entre la destination et l'AP est rompu. Le chemin initial vers la destination (MN3) est MN1-AP1-AP2-MN3. Lorsque le lien AP2-MN3 est rompu, un nouveau chemin est trouvé : MN1-AP1-AP2-AP3-MN3. Ici, la découverte d'une AP est proactive.

5.1.4 Découverte de chemin

Avec DSR de base, la recherche d'un chemin se fait à l'aide du message RREQ diffusé dans l'îlot ad hoc. Dans notre protocole, la recherche de chemin se fait de manière identique. Néanmoins, l'AP ayant en cache le chemin vers le nœud recherché, il peut répondre par un RREP à la place du nœud mobile destinataire, à l'aide de la méthode du *proxy route reply* décrit dans la version initiale de DSR. Si l'AP n'a pas en cache un chemin vers le nœud destinataire, il ne peut alors pas répondre à la requête, et il ne transmet pas le message RREQ aux autres AP pour éviter la saturation du réseau. Si le nœud destinataire est dans l'îlot ad hoc, alors le nœud destinataire reçoit le RREQ et répond directement au nœud source. Les Figures 5.7 et 5.8 illustrent cette recherche de chemin.

Nous avons intégré à DSR les différentes techniques utilisées dans les réseaux ad hoc hybrides. À partir de ce DSR étendu, nous souhaitons maintenant évaluer les performances d'un tel protocole dans un contexte de réseau de véhicules.

Paramètre	Valeur par défaut
Longueur de l'autoroute	10 km
Débit des données	10 kbit/s
Taille d'un paquet de données	512 bits
Durée de simulation	180 secondes
Méthode de découverte de points d'accès	proactive
Portée radio	250 m
Débit de l'interface radio WiFi	11 Mbit/s
Densité du trafic routier	8 véhicules par voie et par km
Vitesse cible d'un véhicule	120 km/h
Nombre voies	3
Intervalle d'envoi messages APADV	10 s
Intervalle d'envoi messages HELLO	0,5 s
Distance entre les AP	1 km
Nombre de communications	1 communication/km

TABLE 5.1 – Paramètres de simulations par défaut

5.2 Paramètres optimaux de notre protocole pour un réseau ad hoc hybride

Nous avons implémenté notre extension de DSR dans le simulateur JIST/SWANS. Des améliorations de SWANS sont nécessaires pour implémenter cette extension, qui ont été décrites en section 3.4.

Notre protocole possède plusieurs paramètres permettant d'en affiner les performances :

- Intervalle d'envoi des balises de découverte d'AP
- Intervalle entre l'envoi de messages HELLO
- Distance entre les AP
- Valeur du TTL

Pour l'ensemble des simulations effectuées, nous fixons aux paramètres les valeurs par défaut résumées dans le Tableau 5.1. Ce sont ces valeurs que nous utilisons quand le paramètre est fixé dans une simulation.

Tout d'abord, nous faisons varier l'intervalle d'envoi des messages APADV envoyés par un point d'accès pour leur découverte en mode proactive. Les courbes en Figures 5.9, 5.10, 5.11 et 5.12 illustrent les résultats sur l'overhead et le débit en fonction d'un intervalle d'envoi du message APADV variant entre 3 et 20 s avec une ou dix sources. Il y a un AP par kilomètre. Les nœuds sources et destinations sont choisis aléatoirement parmi les nœuds mobiles. Avec 10 sources, l'overhead croît nettement et le débit décroît au delà d'un intervalle de 9 s (Figure 5.9 et 5.10). C'est la conséquence de l'utilisation de chemins contenant un lien rompu vers les AP accessibles, entraînant des coupures de connexions dans l'échange de données. En revanche, en Figures 5.11 et 5.12, avec une seule source, le débit décroît au delà d'un intervalle de 7 s, alors que nous observons un overhead minimum à 10 s. Par conséquent, un intervalle inférieur de 7 s n'améliore pas les performances, mais il ne faut pas dépasser les 10 s pour

éviter un taux de perte de paquets trop important, entraînant une diminution du débit.

Nous faisons varier l'intervalle d'envoi des messages HELLO, permettant la détection de rupture de lien dans le chemin vers l'AP courante, entre 0,1 et 7,5 s. Nous observons un overhead minimal à 0,8 s (Figure 5.13a), alors que le débit commence à diminuer au delà d'un intervalle de 0,5 s. Nous concluons à un intervalle d'envoi optimale entre 0,5 et 1 s.

Nous souhaitons maintenant évaluer le nombre de points d'accès nécessaires à des performances optimales, c'est-à-dire connaître la distance maximum entre les AP sans affecter les performances. La longueur de la route est fixée à 10 km, la couverture radio étant fixée à 250 m, la couverture totale de la route est atteinte avec 20 points d'accès. Nous remarquons un débit faible (5 kbit/s sur les 10 envoyés par la source), avec seulement deux AP. Ce débit augmente clairement avec 5 AP (Figure 5.14b) puis augmente jusqu'à 10 AP de manière moins significative, pour rester constant au delà. L'overhead semble atteindre un minimum au delà de 8 points d'accès (Figure 5.14a). 8 AP pour 10 km semble optimale, mais nous confirmerons ce résultat en faisant varier la densité de véhicules en Figure 5.18.

Pour estimer le TTL optimal, nous fixons l'intervalle entre les AP à 2 km. Avec une portée de 250 m, il faut au minimum 4 sauts pour atteindre un point d'accès distant au maximum d'un kilomètre. Néanmoins nous savons, en fonction de la position des véhicules entre le nœud mobile et le point d'accès, que le chemin n'est pas toujours le plus court. Nous nous demandons à combien faut-il fixer le TTL pour limiter l'overhead, tout en laissant la possibilité de trouver un chemin vers une AP. Le délai d'acheminement des paquets est clairement minimum avec TTL de 6 (Figure 5.15a) ce qui est supérieur aux 4 sauts nécessaires. Le débit atteint un maximum avec un TTL de 7 (Figure 5.15b). Un débit faible en dessous d'un TTL de 5 est la conséquence d'un manque de connectivité vers un point d'accès trop éloigné pour l'atteindre en moins de 5 sauts. En revanche une baisse du débit à partir d'un TTL de 7 est la conséquence de saturation du réseau. Ajouter 2 ou 3 sauts au nombre minimum de sauts (ici 4) pour atteindre un point d'accès, permet de fixer un TTL optimum par rapport au délai et au débit.

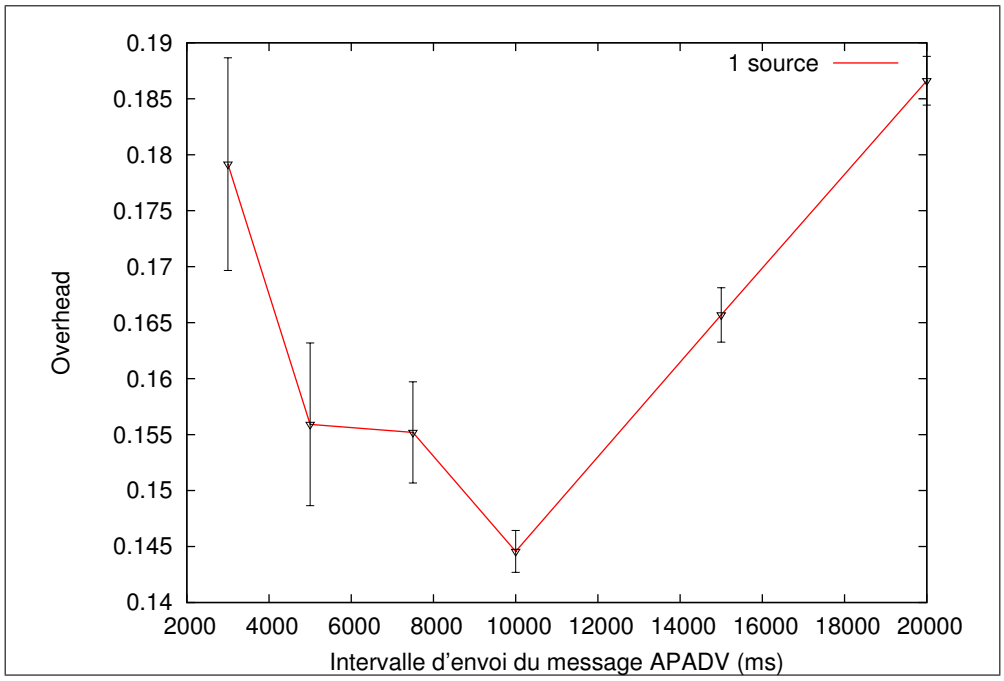


Figure 5.9 – Overhead avec 1 source en fonction de l'intervalle d'envoi des messages APADV

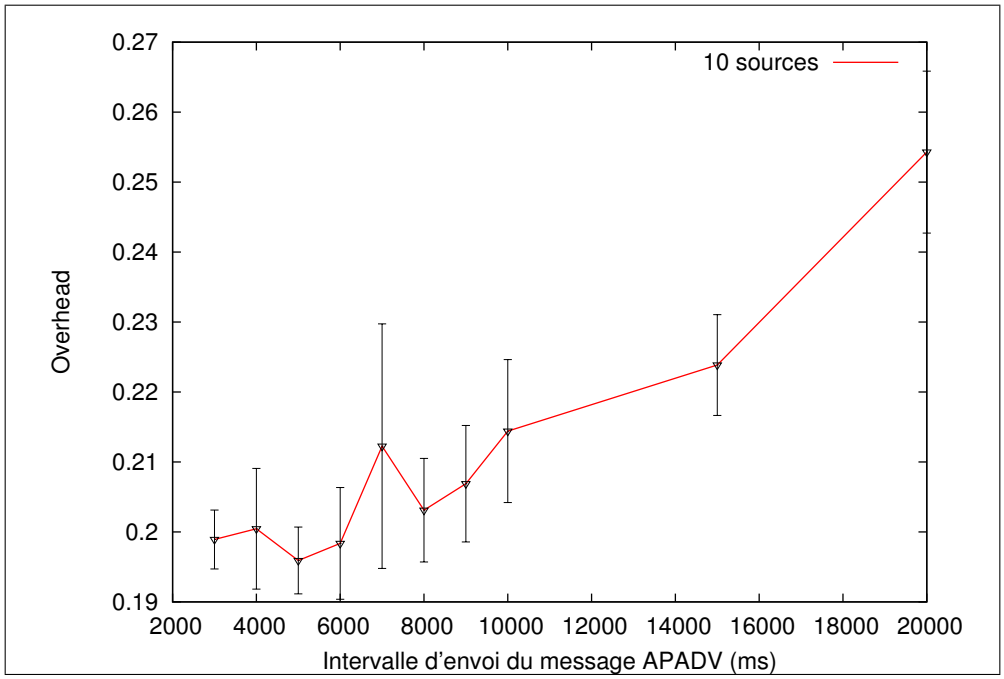


Figure 5.10 – Overhead avec 10 sources en fonction de l'intervalle d'envoi des messages APADV

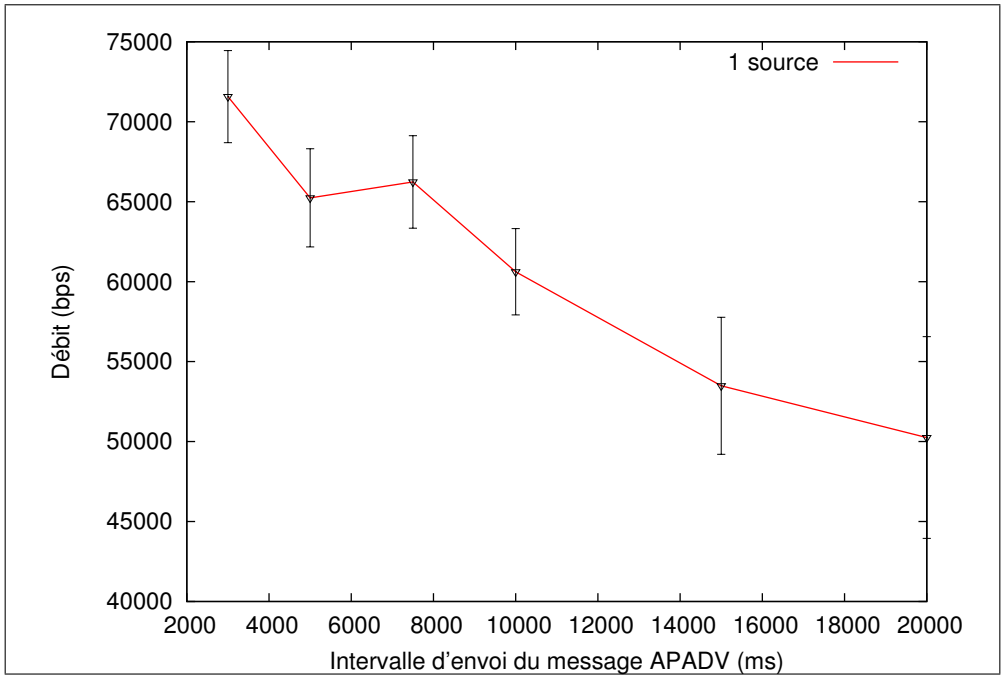


Figure 5.11 – Débit avec 1 source en fonction de l'intervalle d'envoi des messages APADV

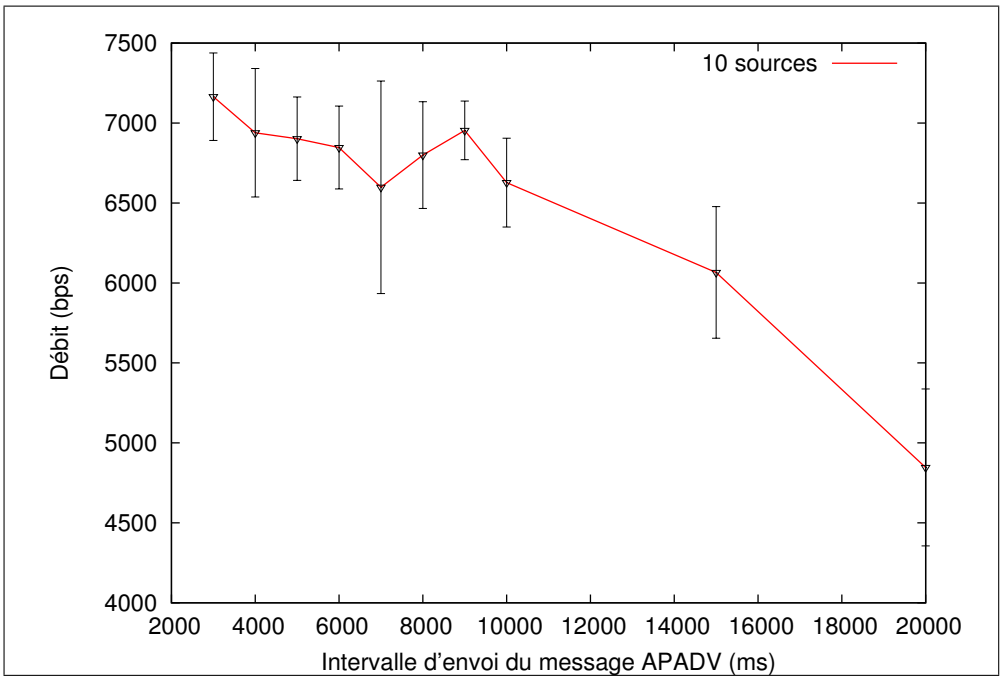
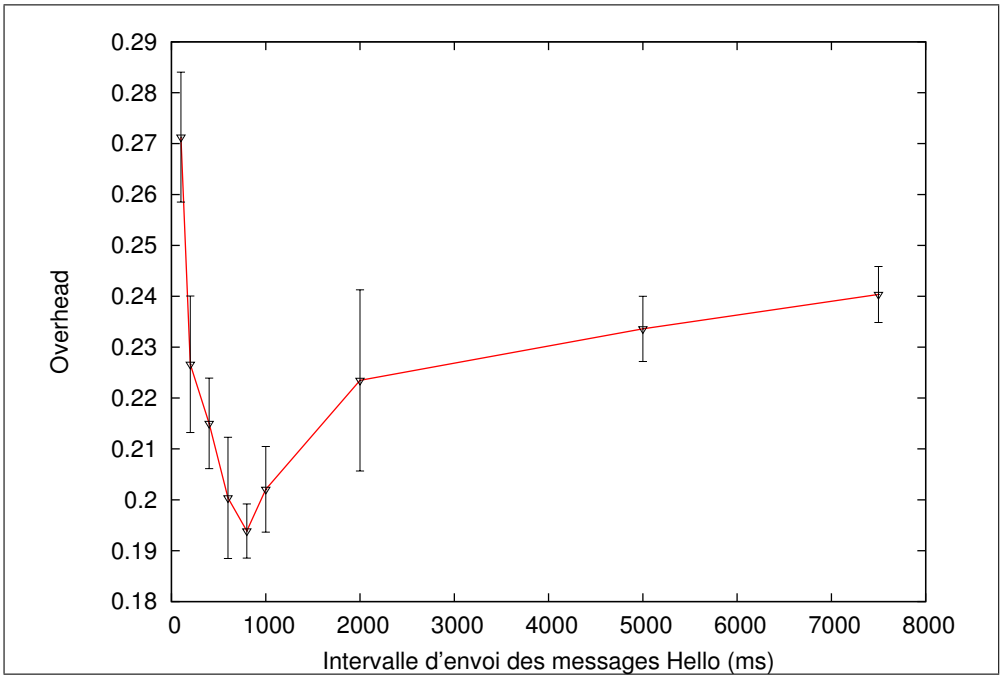
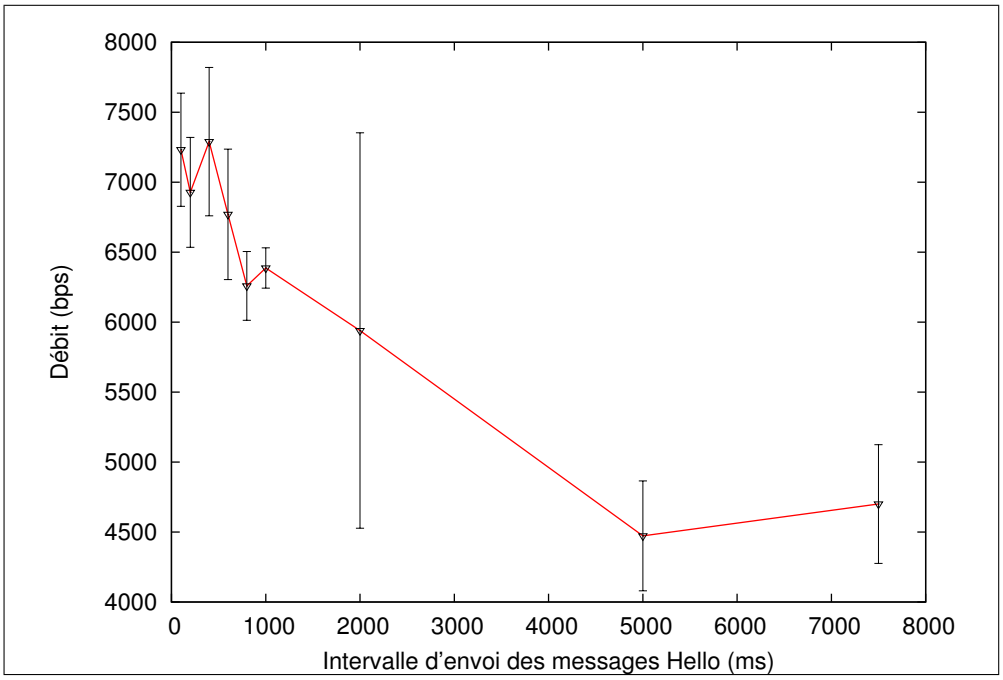


Figure 5.12 – Débit avec 10 sources en fonction de l'intervalle d'envoi des messages APADV

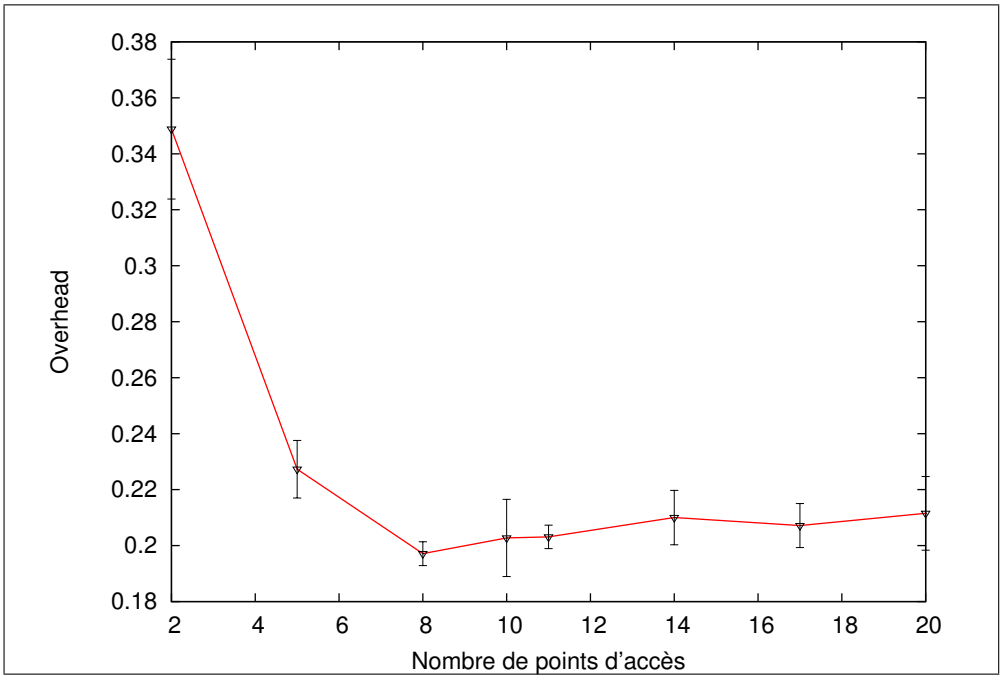


(a) Overhead

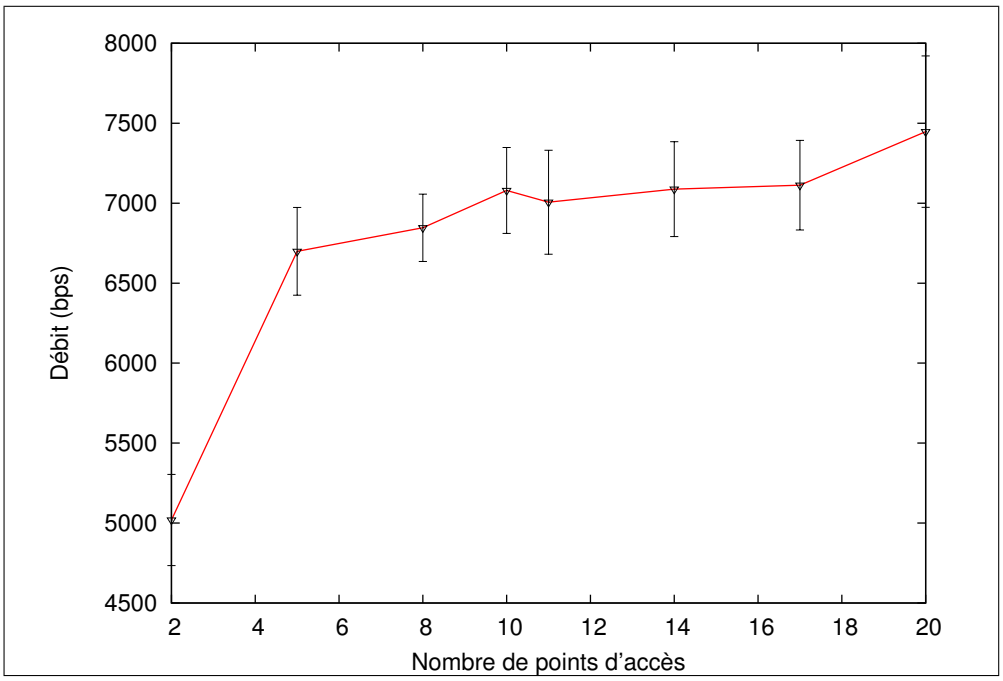


(b) Débit

Figure 5.13 – Estimation de l'intervalle d'envoi optimal du message hello vers l'AP pour détecter une rupture de chemin.

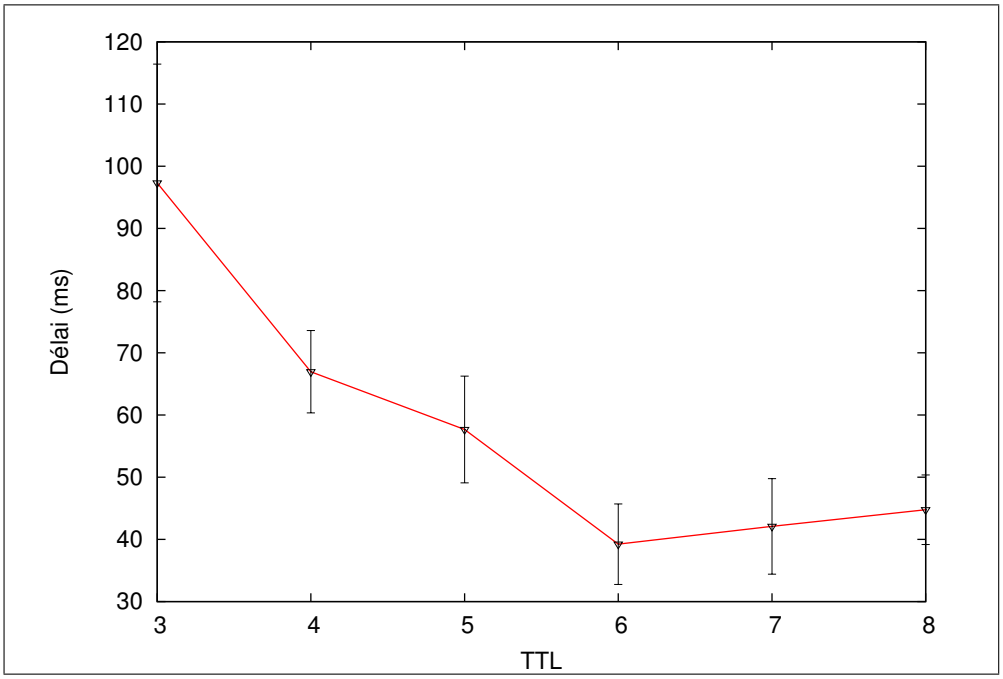


(a) Overhead

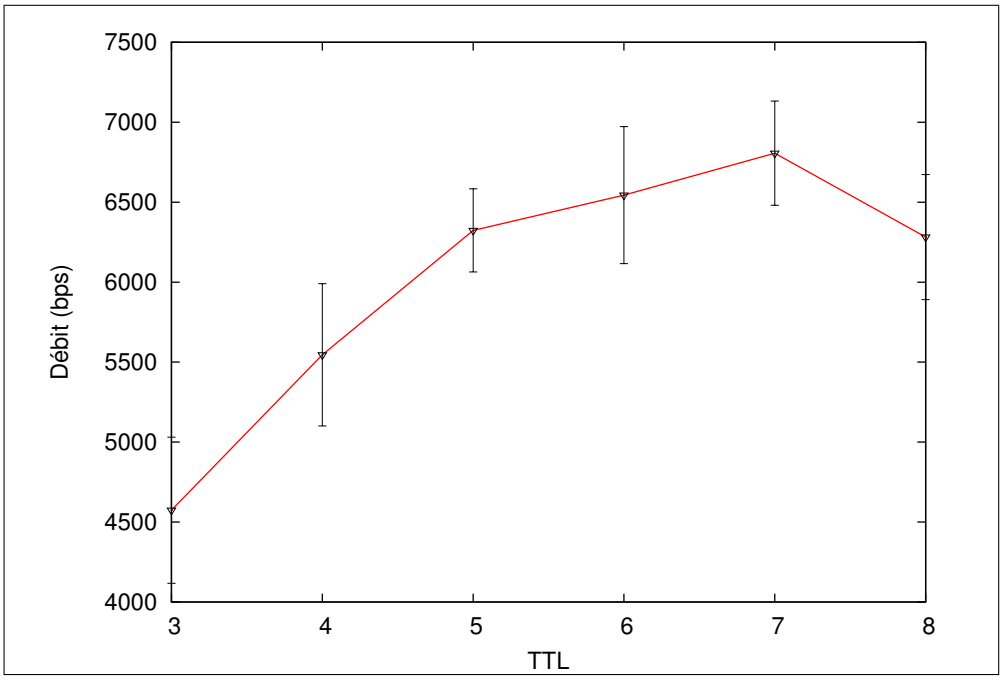


(b) Débit

Figure 5.14 – Estimation du nombre optimal d'AP. Ici la longueur de la route est 10 km, il y a une seule source et l'intervalle d'envoi des messages APADV est 10 s.



(a) Délai



(b) Débit

Figure 5.15 – Estimation du TTL optimal. Ici l'intervalle entre les AP est de 2 km avec une portée de 250 m.

5.3 Évaluation du DSR étendu

Nous avons établi plusieurs scénarios de simulation permettant de répondre à trois questions dans le contexte d'un réseau ad hoc hybride de véhicules sur autoroute :

- Notre protocole apporte-t-il une amélioration du passage à l'échelle par rapport à l'utilisation du DSR de base ?
- Quels sont les intérêts à utiliser un réseau ad hoc hybride par rapport à un réseau ad hoc sans point d'accès ?
- Dans quelle mesure la capacité est augmentée avec un réseau ad hoc hybride ?

Ces trois question sont traitées dans les trois paragraphes suivant.

5.3.1 Passage à l'échelle

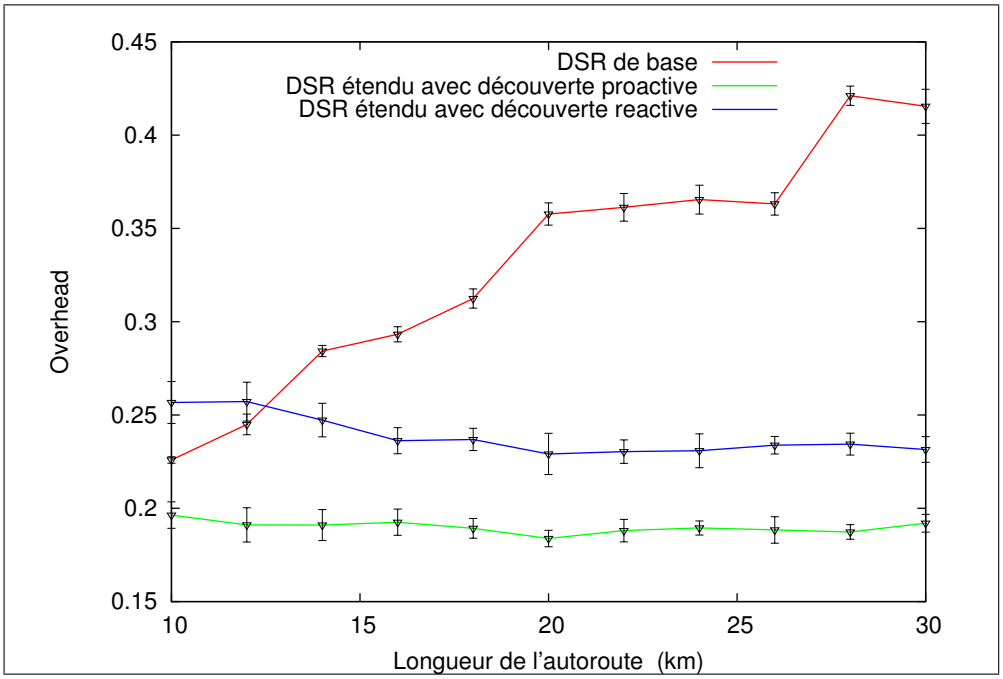
Nous souhaitons comparer le protocole DSR étendu pour réseau ad hoc hybride avec le DSR de base, toujours dans un contexte de réseau ad hoc hybride. Nous faisons varier la longueur de l'autoroute, augmentant la taille du réseau et donc le nombre de nœuds. Donc nous évaluons alors le passage à l'échelle des protocoles étudiés. La distance entre les points d'accès est fixée à 1 km. Nous prenons une source par km. La position de la source et de la destination est choisie aléatoirement parmi les nœuds situés sur l'ensemble de la route.

L'overhead (Figure 5.16a) et le délai (Figure 5.16b) augmentent linéairement pour le DSR de base, alors que le DSR étendu reste stable. En effet, avec le DSR de base, les requêtes de recherche de chemin sont diffusées sur l'ensemble du réseau, alors que le DSR étendu diffuse cette requête uniquement dans l'îlot ad hoc. Nous remarquons également un overhead plus faible avec une découverte proactive. Malgré les messages de signalisation nécessaires à l'enregistrement des nœuds auprès du point d'accès, l'overhead est bien plus faible par rapport à l'overhead produit par le protocole DSR non modifié.

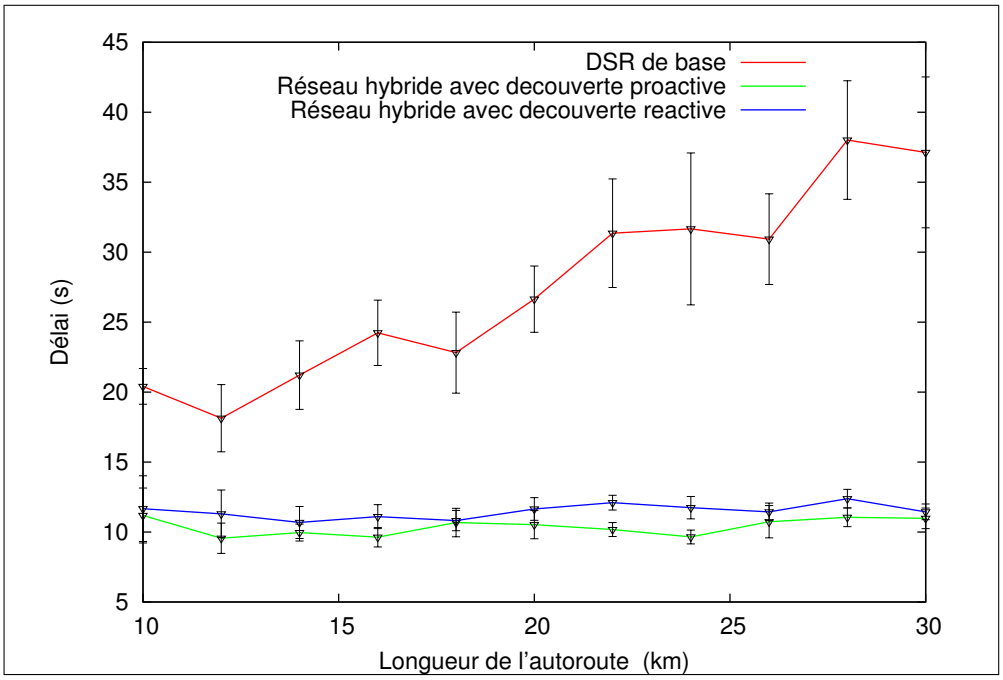
5.3.2 Intérêt d'un réseau ad hoc hybride

Nous souhaitons mesurer l'impact de la densité de véhicules sur les performances, à la fois dans un réseau ad hoc sans point d'accès, et avec plusieurs densités de points d'accès. Nous faisons varier la densité de véhicules entre 1 et 40 véhicules par km et par voie, et le nombre de points d'accès entre 0 et 17 pour 10 km, sachant que la couverture totale est atteinte avec 20 points d'accès.

Sur les Figures 5.17, 5.18 et 5.19, chaque courbe correspond à un nombre de points d'accès différents : 0, 2, 5, 10 ou 17 points d'accès. Nous identifions deux phases selon la densité de véhicules. Dans la première phase, jusqu'à 10 véhicules par kilomètre et par voie, quelque soit le nombre de points d'accès, le débit reçu croît avec la densité de points d'accès sauf pour 17 AP où le débit reçu reste constant (Figure 5.17). La connectivité est la cause de ce résultat : entre 0 et 10 véhicules par km et par voie, le réseau est peu connexe pour trouver un chemin entre une source et une destination du réseau, le taux de perte augmente alors, entraînant une diminution du débit. Sans point d'accès, ce résultat est flagrant, mais avec 5 ou 10 points d'accès, la connectivité est plus importante, augmentant le débit reçu par la



(a) Overhead



(b) Délai

Figure 5.16 – L'extension de DSR permet le passage à l'échelle dans un réseau ad hoc hybride.

destination. À 17 AP, la couverture est quasiment totale, d'où une absence d'influence de la densité du réseau sur la connectivité et donc sur le débit.

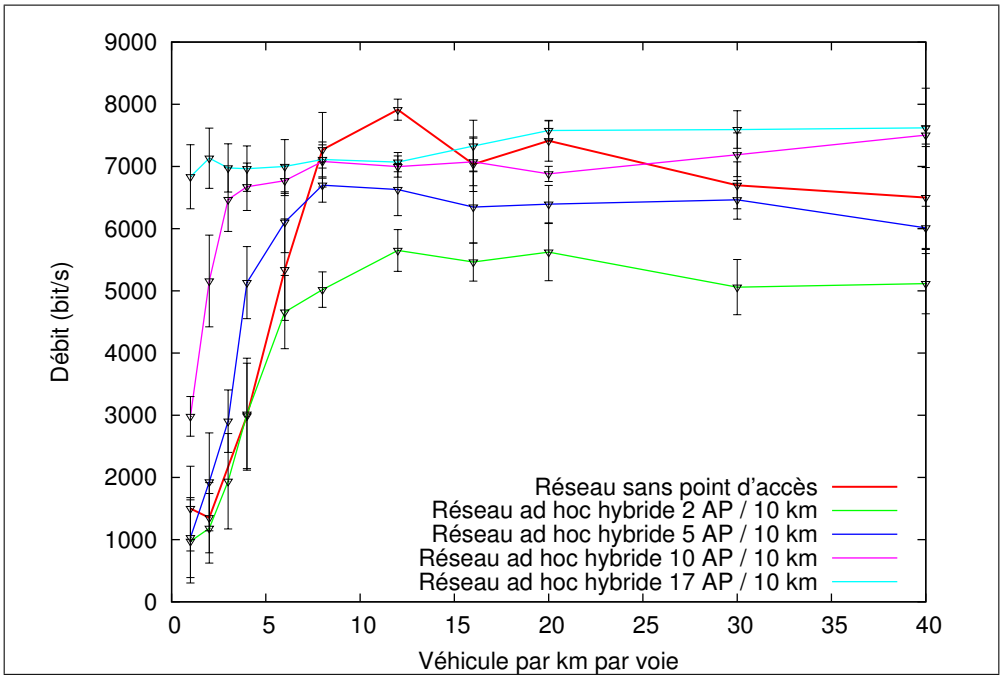


Figure 5.17 – Débit reçu selon la densité de véhicules et le nombre de points d'accès.

La seconde phase, au delà de 10 véhicules par km et par voie, le débit reste constant quelque soit le nombre de points d'accès ; la connectivité étant suffisante pour trouver un chemin entre la source et un point d'accès ou la destination. Reste l'influence de la vitesse relative expliquant un débit maximal reçu par la destination à 7,5 kbit/s sur les 10 kbit/s envoyé par la source. En effet plus la mobilité est importante, plus la rupture de chemin est fréquente. Dans un réseau ad hoc hybride, il existe deux types de mobilité : la mobilité entre nœuds mobiles et la mobilité entre un nœud et un point d'accès. La mobilité entre deux nœuds mobiles voisins est assez faible ; les véhicules circulant à vitesse semblable. Par contre elle augmente avec la longueur du chemin entre la source et la destination. La mobilité entre un nœud est un point d'accès est très importante : le point d'accès étant fixe, contrairement à un véhicule circulant à 120 km/h en moyenne. Le résultat concernant la diminution du débit reçu dans un réseau possédant un nombre de points d'accès inférieur à 10 par rapport à un réseau sans point d'accès est expliqué par la conjonction de ces deux types de mobilité.

Nous avons déjà vu sur les Figures de 5.14a, l'influence du nombre d'AP sur l'overhead. L'overhead minimum est vers 10 AP. Nous souhaitons maintenant vérifier ce résultat en faisant varier la densité du trafic routier. En dessous de 4 véhicules par km et par voie, l'overhead du réseau sans point d'accès est plus faible par rapport au réseau hybride avec points d'accès, quelque soit leur nombre (Figure 5.18). Les messages de signalisation pour l'enregistrement des nœuds et la découverte de points d'accès prennent une part important du trafic total, expliquant l'overhead plus important par rapport à celui du réseau sans point d'accès. Mais au delà de 4 véhicules par km et par voie, avec 5, 10, ou 17 AP, l'overhead

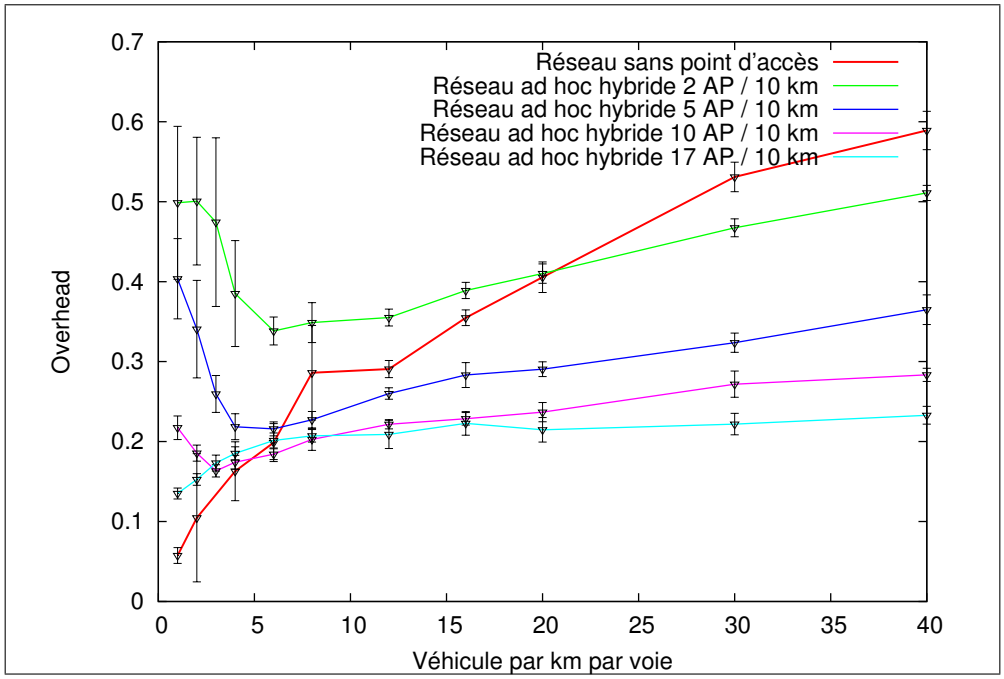


Figure 5.18 – Overhead en fonction de la densité de véhicules et le nombre de points d'accès.

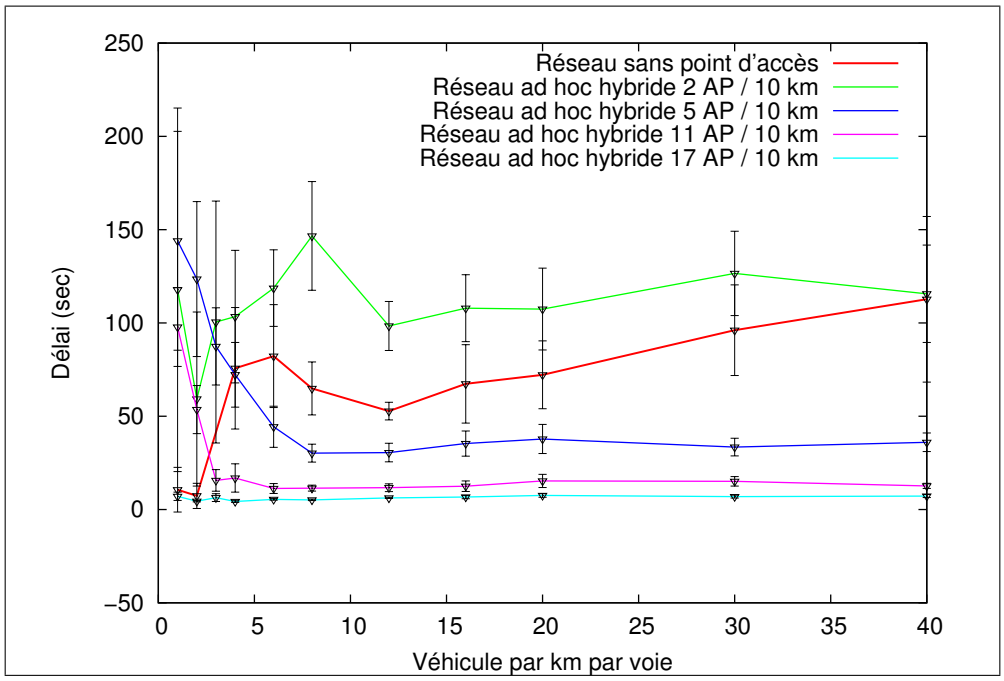


Figure 5.19 – Délai en fonction de la densité de véhicules et le nombre de points d'accès.

est inférieur à celui de réseau sans point d'accès. Nous remarquons également un overhead stable aux alentour de 20 % avec 10 et 17 points d'accès, contrairement à l'overhead avec 5 points d'accès qui varie entre 20 et 40 %, le manque de connectivité à faible densité entraînant un part important de pertes de données (voir Figure 5.17 concernant le débit). Sans point d'accès, nous remarquons également une augmentation linéaire de l'overhead avec la densité de véhicules montrant que le réseau ne supporte pas le passage à l'échelle.

Le résultat sur le délai est similaire, nous remarquons bien avec 10 ou 17 points d'accès (Figure 5.19) un délai plus faible et constant par rapport aux autres réseaux.

À densité faible, inférieure à 4 véhicules par voie et par kilomètre, l'intérêt d'un réseau ad hoc hybride est faible, la connectivité n'étant pas assez importante pour obtenir un débit significativement supérieur au réseau sans point d'accès, et les messages de signalisation produits par le protocole DSR étendu prenant une part importante du trafic. En revanche, au delà de 10 véhicules par voie et par kilomètre, avec un nombre de points d'accès supérieur ou égal à 10 (1 AP/km), l'overhead reste stable à 20 % et le débit reçu est quasiment identique au débit sans point d'accès, voire supérieur pour une forte densité de véhicules (au dessus de 30 %).

5.3.3 Résultats sur la capacité

Jusqu'à présent, le débit envoyé par la source était de 10 kbit/s. Nous souhaitons maintenant évaluer les performances du réseau quand on augmente ce débit.

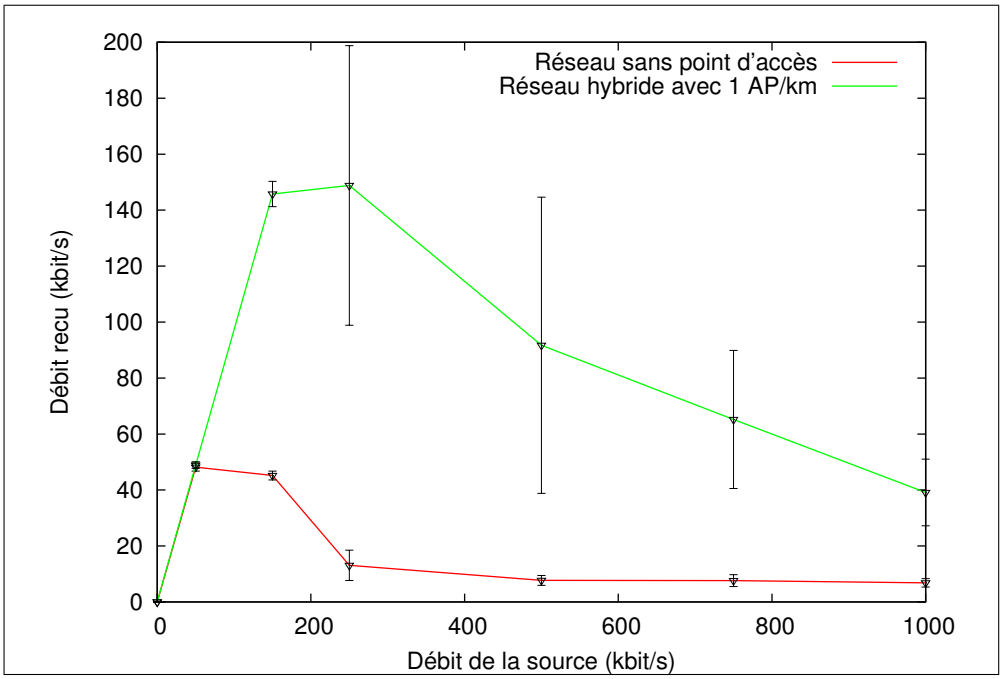


Figure 5.20 – Estimation de la capacité du réseau.

La capacité est définie comme le débit maximal reçu par la destination. Sur la Figure 5.20, nous évaluons le débit reçu par la source et donc nous pouvons en déduire la capacité du réseau. La capacité du réseau hybride à 1 point d'accès par kilomètre est clairement supérieure par rapport au réseau sans point d'accès. La capacité du réseau ad hoc hybride étant d'environ 150 kbit/s alors que la capacité sans réseau ad hoc hybride de 50 kbit/s. Le réseau de points d'accès étant à large capacité (100 Mb/s) et sans perte de paquets, nous obtenons donc une augmentation substantielle de la capacité dans un réseau ad hoc hybride. La capacité reste encore faible à cause de la capacité limitée du réseau ad hoc pour atteindre un point d'accès en plusieurs sauts.

Conclusion

Dans un réseau de véhicules sur autoroute les problèmes de connectivité du réseau empêchent la communication entre véhicules sur de grandes distances. L'aspect linéaire de ces réseaux réduit également considérablement leur capacité. L'ajout de points d'accès au réseau permet de réduire l'impact de ces problèmes tout en réduisant les coûts par rapport à un réseau totalement cellulaire où les points d'accès couvrent entièrement le réseau. Ces réseaux, appelés réseaux ad hoc hybrides, conservent l'aspect multi-sauts des communications permettant l'utilisation des réseaux ad hoc classiques existant, tel que le protocole réactif DSR. DSR étant mal adapté à un réseau ad hoc hybride, nous l'avons étendu en ajoutant des mécanismes d'enregistrement et de découverte de points d'accès. Après avoir mis en œuvre ce protocole dans un simulateur et avoir déterminé ses paramètres optimaux, nous avons montré qu'un tel protocole réduit considérablement l'overhead par rapport au DSR de base quand la longueur de la route, et donc le nombre de nœuds augmentent. Nous avons également montré qu'un réseau ad hoc hybride augmente la capacité d'un réseau de véhicules sur autoroute. Néanmoins, la forte mobilité dans un réseau ad hoc hybride causée par la vitesse relative entre un nœud mobile et un point d'accès limite le gain en terme de débit par rapport à un réseau ad hoc sans point d'accès. Nous pensons qu'une gestion de groupes de nœuds peu mobiles entre eux (comme le sont des véhicules sur autoroute) permettrait de réduire l'impacte de la mobilité sur la perte de paquets et le débit dans un réseau ad hoc hybride. Dans le chapitre suivant nous allons présenter la gestion de convois de véhicules (groupe de nœuds) dans les réseaux ad hoc et proposer un protocole inédit adapté aux réseaux de véhicules sur autoroute.

Chapitre 6

Formation de convois pour la gestion de la mobilité

Dans le chapitre précédent, nous avons recensé deux principaux problèmes des réseaux ad hoc hybrides de véhicules : d'une part, la vitesse relative entre un véhicule et un point d'accès et d'autre part, l'accès au point d'accès multi-sauts. Une grande différence de vitesse entre un véhicule et un point d'accès a pour conséquence une durée de connexion faible au point d'accès et donc une fréquence de changement de points d'accès élevées. Pendant le changement de point d'accès, les paquets en provenance de l'infrastructure ne peuvent plus atteindre le nœud mobile, provoquant leur perte.

La formation de groupes de nœuds, appelé communément clusters, permet de créer une organisation hiérarchique du réseau ad hoc afin de minimiser l'échange d'informations pour le maintien des tables de routage. Dans un réseau de véhicules les nœuds mobiles se déplacent à des vitesses similaires et dans la même direction, un cluster de nœuds est alors le plus souvent stable.

Il est intéressant d'exploiter cette stabilité dans les réseau ad hoc hybride de véhicules : ce n'est plus un nœud qui s'enregistre auprès d'un point d'accès, mais le cluster, diminuant ainsi le nombre d'entités dont on doit gérer la mobilité. De plus, il est possible que deux nœuds membres d'un cluster soient connectés à deux points d'accès différents, permettant la continuité d'accès à Internet.

La propriété unidimensionnelle d'un réseau de véhicules sur autoroute permet de créer un cluster particulier : le convoi. Dans un convoi, la tête du cluster est toute désignée, c'est le véhicule qui est devant tous les membres du convoi. Le positionnement géographique des véhicules et la connaissance de leur vitesse permet donc d'améliorer la qualité de formation d'un convoi.

L'étude de la formation de clusters étant fortement liée aux systèmes répartis, nous allons donc décrire les algorithmes de base de formation de clusters et leurs applications pour les réseaux de véhicules. Nous proposons ensuite un protocole de formation de convois adapté à un réseau de véhicules sur autoroute basé sur un algorithme de formation de clusters. Enfin, nous évaluons ce protocole à l'aide de simulations sous JIST/SWANS pour vérifier ses

propriétés dans un réseau de véhicules sur autoroute.

6.1 Formation de clusters dans un réseau

Il existe deux manières de hiérarchiser un réseau ad hoc : la hiérarchisation virtuelle ou réelle. La hiérarchisation réelle consiste à ajouter des points d'accès dans le réseau, on obtient un réseau ad hoc hybride. La hiérarchisation virtuelle consiste à affecter un rôle particulier à certains nœuds du réseau ad hoc. C'est l'objectif d'un algorithme de formation de clusters. Dans tout algorithme de formation de clusters, le rôle principal est le chef de cluster, appelé clusterhead. Un clusterhead unique est élu par cluster.

Un algorithme de formation de clusters comporte deux phases :

- Élection des clusterheads (phase d'initialisation)
- Maintenance du cluster.

Nous appelons *membre*, un nœud du cluster associé à un clusterhead. Nous appelons *passerelle* un nœud voisin avec au moins un autre cluster. Une passerelle sert à transmettre les messages d'un cluster à un autre.

Nous allons maintenant décrire divers algorithmes répartis de formation de clusters de la littérature.

6.1.1 Algorithmes de formation de clusters

Nous allons classer les algorithmes de formation de clusters selon leur manière de contrôler la taille d'un cluster. Dans un cluster k-sauts, chaque membre doit être au maximum à k-sauts du clusterhead. Dans un cluster multisauts, il n'existe pas de limite en nombre de sauts, cependant la taille du cluster (en nombre de nœuds) est bornée. Enfin, certains clusters sont organisés en arbre où les membres du cluster sont les nœuds de l'arbre. Nous allons décrire en détail ces trois modèles de construction d'un cluster.

6.1.1.1 Cluster 1-sauts

Dans [22] l'auteur introduit les algorithmes DCA (Distributed Clustering Algorithm) et DMAC (Distributed Mobility-Adaptive Clustering). DCA est une généralisation d'algorithme de formation de cluster existant. Les propriétés du cluster sont :

- Tout nœud membre a au moins un clusterhead comme voisin
- Le clusterhead de tout nœud membre est le voisin de plus gros poids.
- Deux clusterheads ne peuvent pas être voisins.

Le poids est une métrique à définir, cela peut-être le degré (nombre de liens avec les voisins), l'inverse de la vitesse moyenne, l'identifiant, etc. Ici le nœud est à un saut maximum du clusterhead, c'est donc un cluster *1-saut*. Typiquement, l'identifiant d'un cluster est l'identifiant du clusterhead. Le cluster i est alors le cluster dont le clusterhead est i.

L'algorithme DCA utilise deux messages : CH et JOIN. Le message CH est envoyé par un nœud u pour avertir ses voisins qu'il est clusterhead et le message JOIN est envoyé par un nœud v pour avertir ses voisins qu'il a rejoint un cluster. Initialement, un nœud dont le poids est le plus grand par rapport à ses voisins se déclare clusterhead, et envoie un CH. Un nœud recevant un message CH de u, rejoint le cluster u si tous ses voisins de poids plus grand que lui ont rejoint un cluster. En recevant un message JOIN de v trois cas sont possibles :

- Le nœud est un clusterhead : il ajoute à sa table des membres du cluster le nœud v.
- Le nœud n'est pas clusterhead et tous ses voisins de poids plus grand que lui ont rejoint un cluster, alors le nœud devient clusterhead.
- Le nœud n'est pas clusterhead et un des ses voisins de poids plus grand que lui est clusterhead, alors le nœud rejoint le cluster dont le poids du clusterhead associé est le plus grand.

Cet algorithme est adapté au réseau ad hoc dont la topologie est statique ou quasi-statique. Pour la formation de clusters dans un réseau dynamique, un algorithme de maintenance du cluster est nécessaire, tel que DMAC introduit dans [22]. Cet algorithme réagit aux événements de réception des messages CH et JOIN, mais aussi aux événements de topologies : nouveau lien et rupture d'un lien. Il considère que ces deux événements sont déclenchés par une couche inférieure. Avant de décrire les actions sur ces événements, définissons la procédure de sélection d'un clusterhead. Pour déterminer son rôle, le nœud teste s'il existe au moins un clusterhead voisin avec un poids plus élevé que lui. Si c'est le cas, le nœud rejoint le cluster associé (il devient alors membre), sinon il devient clusterhead.

À l'initialisation, chaque nœud détermine son rôle. Soit v le nœud où s'exécute l'algorithme. Lors de la rupture du lien (u, v), si le nœud v est clusterhead alors il supprime le nœud u de sa table des membres, sinon, si v est associé avec le clusterhead u (v fait partie du cluster u), alors v re-calcule son rôle. Lors de la création d'un nouveau lien (u,v), si u est clusterhead avec un poids plus grand que le poids du nœud v, alors v rejoint le cluster u sinon v reste membre de son cluster.

Sur réception d'un message CH envoyé par un clusterhead u, si le poids de u est plus grand que le poids du clusterhead associé avec v, alors v rejoint le cluster u. Sur réception d'un message JOIN de u, si v est clusterhead et u souhaite rejoindre le cluster v, alors le nœud ajoute u à sa liste des membres, sinon il la supprime (u a rejoint un autre cluster). Si v n'est pas clusterhead et le cluster de v est u, alors le nœud u n'est plus clusterhead et v doit re-calculer son rôle.

6.1.1.2 Cluster k-sauts

Pour minimiser le nombre de clusters, les auteurs de [94] ont généralisés l'algorithme 1-saut à k-sauts. Pour découvrir le voisinage à k-sauts, les messages CH et JOIN sont diffusés avec un TTL égal à k. Les auteurs évaluent leur algorithme de formation de clusters avec k=1 ou k=2 et avec comme poids l'identifiant d'un nœud ou la connectivité d'un nœud. L'efficacité de l'algorithme est mesurée par le nombre moyen de clusters créés, le taux moyen de nœuds en bordure de cluster ou la taille moyenne d'un cluster.

Passer de 1-hop à k-hop en diffusant les messages à k-sauts produit une augmentation de l'overhead, particulièrement avec la diffusion du choix du rôle de chaque nœud dans un

rayon de k-sauts, l'overhead généré est d'autant plus important que le réseau est dense. En effet, plus le réseau est dense, plus il existe de nœuds membres diffusant des messages dans un rayon de k-sauts. Pour palier ce problème, un algorithme appelé Circle est présenté dans [16].

Circle est basé sur le choix d'un initiateur jouant le rôle de clusterhead initial. Cet initiateur est choisi par une phase préliminaire (non détaillée dans l'étude). Dans [87], l'initiateur est le nœud possédant le plus petit identifiant parmi les voisins. Avant de démarrer l'élection du clusterhead, l'initiateur diffuse à k-sauts un message indiquant qu'il est l'initiateur, les autres nœuds ne seront pas initiateurs. Ensuite, l'initiateur choisit la distance optimale au prochain clusterhead, c'est-à-dire le nombre de sauts optimal entre chaque clusterhead. Cette décision est basée sur la densité du réseau et d'autres considérations géométriques que nous ne détaillons pas ici. La distance choisie est appelée distance de délégation, notée g, avec $g > k$. L'initiateur diffuse un message FORMATION avec un TTL initialisé à g. Tout nœud recevant le message avec un TTL plus grand que $g - k$ est membre. Les nœuds recevant le message avec un TTL plus petit ou égal à $g - k$, deviennent candidats à devenir un clusterhead. Chaque candidat envoie un message CANDIDACY à l'initiateur. Après un délai pré-défini, l'initiateur envoie au meilleur candidat un message DELEGATION, ce dernier devient alors clusterhead et répète la même procédure que l'initiateur.

L'algorithme de formation de clusters présenté dans [87] se base sur une méthode similaire mais va plus loin : le choix du clusterhead est réparti, contrairement à Circle où le choix est effectué par l'initiateur. L'initiateur diffuse un message de formation à $2k + 1$ sauts. Le nœud situé à moins de k sauts de l'initiateur devient membre du cluster. Le nœud situé entre $k + 1$ et $2k + 1$ sauts est un candidat pour devenir clusterhead. Les candidats rentrent alors dans une période de compétition de la même manière que CBF présenté dans le paragraphe 2.1.3. Chaque candidat attend un délai inversement proportionnel à sa distance de l'initiateur et à son degré, le protocole privilégiant la connectivité des clusterheads. Une fois le délai passé, le nœud devient clusterhead et répète la même procédure que l'initiateur en diffusant un message de formation. Les autres nœuds en attente recevant ce message annulent leur candidature et deviennent membres du cluster.

6.1.1.3 Cluster multisauts

Il est également possible de limiter la taille d'un cluster, non pas par le nombre de sauts maximum entre un membre et son clusterhead, mais par le nombre de nœuds dans un cluster. Un algorithme présenté dans [96] utilise cette méthode. Si le cluster dépasse la taille limite, il est séparé en deux (en élisant un nouveau clusterhead). Si le cluster est considéré comme trop petit, un nœud en bordure d'un cluster voisin plus grand est recruté.

6.1.1.4 Cluster organisé en arbres

Dans [92] les auteurs présentent un algorithme réparti permettant de créer des arbres à partir du graphe représentant le réseau. Chaque nœud choisit un *voisin préféré* dont le degré est le plus grand, et l'identifiant le plus grand en cas d'égalité. Le voisin préféré est alors choisi comment parent du nœud. Un nœud ne possédant pas de voisin préféré (pas de nœud voisin de degré plus grand) est la racine de l'arbre. Ici l'arbre est alors un cluster, et la racine

le clusterhead. Les nœuds possédant un voisin appartenant à un autre arbre sont alors des passerelles reliant les arbres entre eux. Il n'existe pas de limite concernant la taille du cluster en nombre de sauts à la racine ou en nombre de nœuds.

6.1.2 Applications dans les réseaux ad hoc

Nous avons vu différents algorithmes de formation de clusters prenant à la fois en compte la dynamicité du réseau et la réduction de l'overhead. Associer un algorithme de formation de clusters à un réseau ad hoc permet sa hiérarchisation, limitant l'overhead produit par la diffusion de messages dans le réseau. Nous allons décrire différents algorithmes de formation de clusters pour un réseau ad hoc.

Dans [93], les auteurs proposent un protocole de routage ad hoc basé sur un algorithme de formation de clusters k-sauts. Le réseau ad hoc est hierarchisé en deux niveaux : le niveau supérieur (niveau des clusters) et le niveau inférieur (niveau des nœuds). Au niveau du cluster, le routage est proactif : chaque clusterhead maintient un graphe de l'ensemble des clusterheads du réseau et une liste des membres du cluster associé. Cependant, il ne sait pas comment les membres sont connectés entre-eux. Un clusterhead connaît donc la topologie au niveau cluster, mais ne connaît pas la topologie du réseau à l'intérieur des autres clusters. Un nœud obtient la topologie au niveau cluster en envoyant une requête CLREQ. Le clusterhead répond via un message CLREP. La source obtient un chemin vers un nœud du cluster en envoyant un message RREQ, soit pour connaître la chemin vers une destination du cluster, soit pour connaître le chemin d'une passerelle pour atteindre un autre cluster. Donc un nœud répond via un message RREP, s'il est la destination ou s'il est une passerelle, sinon il transmet le message. Si un nœud reçoit un RREQ initié par un nœud d'un autre cluster, il ne fait rien, ainsi un message RREQ est seulement diffusé à l'intérieur d'un cluster.

Une autre stratégie pour améliorer les performances d'un routage réactif est proposé dans [97]. L'algorithme de formation de clusters est celui décrit dans [96] (§ 6.1.1.2). Un clusterhead forme un arbre de recouvrement minimal à partir des informations sur la topologie de son cluster obtenu à l'aide de deux messages MEP (envoyé à intervalles réguliers) et MAP (réponse à MEP). Les requêtes de routage (RREQ) sont diffusées via les nœuds de l'arbre, et non tous les nœuds du réseau, réduisant ainsi l'overhead.

Pour améliorer le passage à l'échelle du protocole de routage proactif OLSR (§ 2.1.2), les auteurs de [109] proposent d'utiliser des clusters. Les auteurs considèrent la formation de clusters sans détailler l'algorithme utilisé. Le protocole OLSR fonctionne normalement à l'intérieur d'un cluster, à l'exception des messages TC qui ne sont pas transmis par un nœud appartenant à un cluster différent de l'initiateur du message pour limiter sa diffusion. Pour atteindre un nœud dans un autre cluster, un autre protocole appelé C-OLSR permet de créer des routes entre les clusters. Pour cela, deux messages ont été ajoutés : C-HELLO et C-TC. De la même manière qu'OLSR, des clusters MPR (C-MPR) sont choisis à l'aide des messages C-HELLO. Ces mêmes messages permettent de connaître les clusters voisins. Les messages C-TC contiennent la liste de ses voisins et sont propagés à tous les autres clusters via les C-MPR uniquement (les clusters non MPR ne transmettant pas les messages C-TC).

6.1.2.1 Applications dans les réseaux ad hoc de véhicules

Nous avons décrit dans le paragraphe 2.1.3 les protocoles ad hoc basés sur la position géographique. Le point faible de ces protocoles, d'un point de vue du passage à l'échelle, est la localisation géographique de la destination. Nous avons décrit divers services de localisation permettant d'améliorer le passage à l'échelle d'un protocole de localisation géographique. Dans [106], les auteurs proposent une solution pour améliorer le passage à l'échelle d'un protocole de localisation géographique à l'aide de clusters dans un réseau de véhicules. Un algorithme de formation de clusters à 2-sauts est décrit avec désignation de nœuds passerelles. Un protocole géographique simple de type MFR (Most Forward within Radius) [118] est utilisé, et la localisation de la destination se fait simplement par la diffusion de messages LREQ et LREP. En revanche, seul un clusterhead ou une passerelle transmet ces deux messages, limitant l'overhead d'une diffusion gloutonne. Nous regrettons néanmoins que le papier ne compare l'algorithme qu'avec AODV et DSR, alors qu'une comparaison avec l'utilisation d'un service de localisation à *rendez-vous* tel que GLS ou GHLS aurait été plus pertinente.

D'autres travaux proposent des algorithmes de formation de clusters adaptés aux réseaux de véhicules et traitent de leur évaluation. Dans [50], les auteurs prennent en compte la vitesse ou la position des véhicules voisins dans l'algorithme d'élection de clusterhead (k-sauts Lowest-ID et Highest-Degree), afin d'augmenter la stabilité du cluster. Dans [49], ces travaux sont étendus pour prendre en compte la direction de déplacement des véhicules : deux véhicules circulant dans la direction opposée ne peuvent pas être dans le même cluster. Les auteurs ajoutent également un critère pour prendre en priorité un clusterhead qui a été clusterhead le plus longtemps dans le passé.

Il est également possible de prendre en compte la dynamique particulière d'un réseau de véhicules, par exemple il existe un effet oscillatoire de la distance entre les véhicules. Dans [29] cet effet est pris en compte avec un critère de distance (géographique) limite entre deux clusterheads. Ils ajoutent également un critère basé sur le point de sortie du véhicule de la route.

6.1.2.2 Métriques d'évaluation d'un algorithme de formation de clusters

Dans l'ensemble des ces travaux, les métriques utilisées pour évaluer la qualité de la formation des clusters sont :

- la durée de vie d'un cluster,
- le nombre de changements de clusterhead,
- le nombre de clusterhead pour un membre,
- la taille moyen d'un cluster en nombre de nœuds.

La durée de vie d'un clusterhead permet d'évaluer sa stabilité. En effet, plus le cluster change de topologie (ajout/suppression d'un nœud membre ou nouveau clusterhead) plus il y a de messages générés pour sa maintenance. Le nombre de changements de clusterhead est une métrique liée à la durée de vie, car elle correspond à l'ajout/suppression de membres dans un cluster.

Le nombre de clusterhead et sa taille moyenne caractérise l'utilité d'un cluster, plus il existe de cluster ou moins ils sont gros, moins ils sont utiles. Il existe un compromis entre

stabilité / taille des cluster : plus un cluster est gros, plus il est utile, mais plus il est petits, plus il est stable.

6.2 Formation de convois

Les propriétés d'un réseau de véhicules sur autoroute permettent de créer un cluster particulier : le convoi. Dans un convoi, les nœuds se suivent, la tête du cluster est alors toute désignée : c'est le véhicule qui est devant tous les membre du convoi. Le positionnement géographique des véhicules et la connaissance de leur vitesse permet d'améliorer la qualité de formation d'un convoi.

Dans cette section nous proposons notre algorithme réparti de formation de convoi. Nous considérons que le véhicule est équipé d'une seule interface de transmission sans fil, d'un système de géolocalisation, d'un système de mesure de la vitesse actuelle et d'une énergie illimitée.

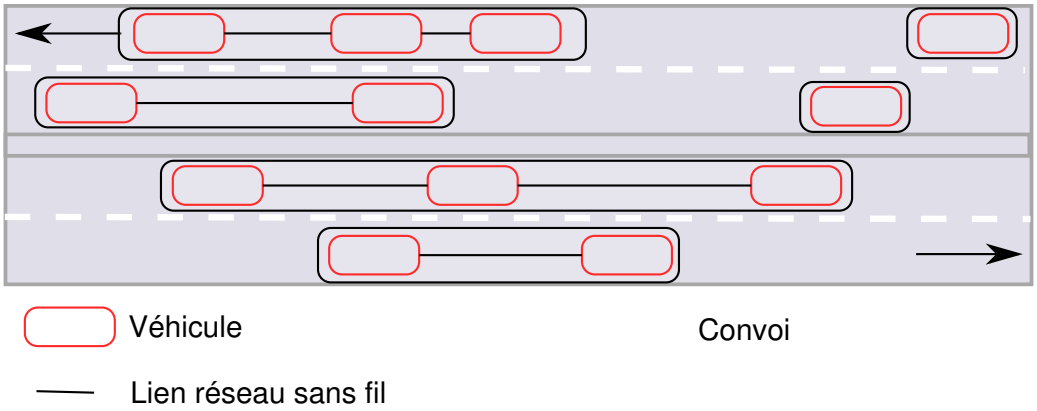


Figure 6.1 – Représentation intuitive de convois dans un réseau de véhicules sur autoroute.

Intuitivement, nous représentons un convoi comme une file de véhicules. Sur une route à plusieurs voies, comme une autoroute, nous considérons une file par voie. Cette vue n'est qu'une vue logique du réseau (Figure 6.1), mais n'est pas une vue représentant les communications possibles entres véhicules. La communication directe entre deux véhicules ne se limite pas aux véhicules situés directement devant et derrière, mais également aux véhicules situés sur une autre file ou plus loin en amont comme en aval sur la même file. Le graphe représentant le réseau ad hoc est "allongé", comme cela est représenté en Figure 6.2.

Nous souhaitons former des clusters de véhicules stables pouvant communiquer entre eux via un protocole de routage intra-cluster. À l'intérieur du cluster, le graphe représentant le réseau doit alors être connexe. Nous ne prenons pas en considération la voie où le véhicule circule, c'est-à-dire les véhicules appartenant à un même convoi ne peuvent être sur des voies différentes, la raison étant l'impossibilité d'obtenir cette information de manière précise (les systèmes de géolocalisation n'étant pas assez précis pour déterminer la voie sur laquelle circule le véhicule). Néanmoins, le regroupement de véhicules sur une même voie est favorisé

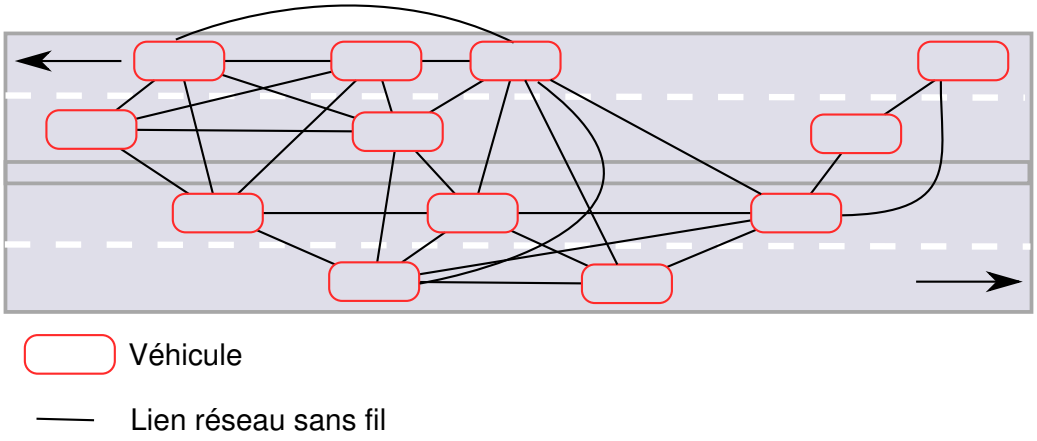


Figure 6.2 – Grahe A d'un réseau ad hoc.

par le critère d'entrée dans un convoi, que nous verrons ci-après. Sur la Figure 6.3, les véhicules qui circulent dans une direction opposée ne sont pas dans le même convoi. En revanche dans le convoi 3 et 4 les véhicules sont dans le même convoi alors qu'ils sont sur des files différentes. Les véhicules du convoi 1 par rapport à ceux du convoi 2 ont une différence de vitesses significatives, et sont alors dans deux convois différents.

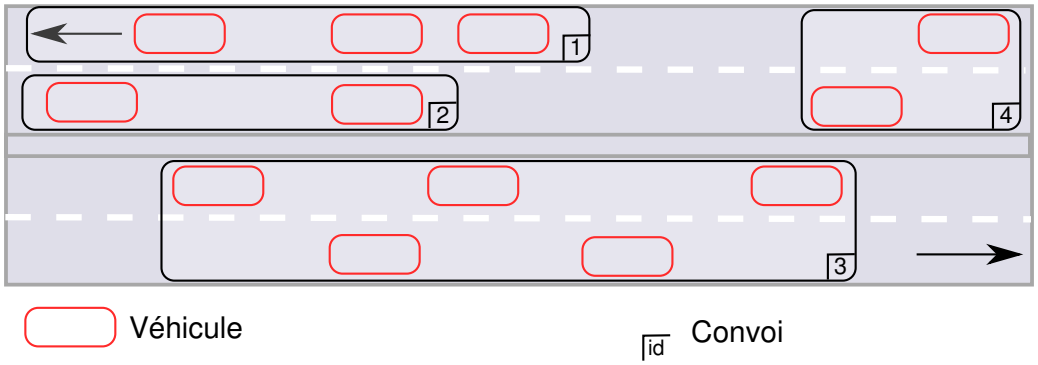


Figure 6.3 – Convois de véhicules sur le graphe A.

Le système de géolocalisation permet à la fois de connaître la vitesse, la position et la direction de déplacement du véhicule. Ces trois informations sont nécessaires à notre algorithme de formation de convois pour déterminer si un nœud peut rejoindre le convoi, afin d'en assurer la stabilité.

Nous avons vu dans le chapitre 4, qu'un protocole géographique diminue l'overhead d'un protocole de routage ad hoc. Par conséquent, nous adoptons un tel protocole de routage basé sur la position géographique comme protocole de routage intra-convoi. Avant de décrire l'algorithme de formation de convois en lui-même, nous allons présenter le protocole de

routage intra-convoi que nous avons adopté.

6.2.1 Protocole de routage intra-convoi

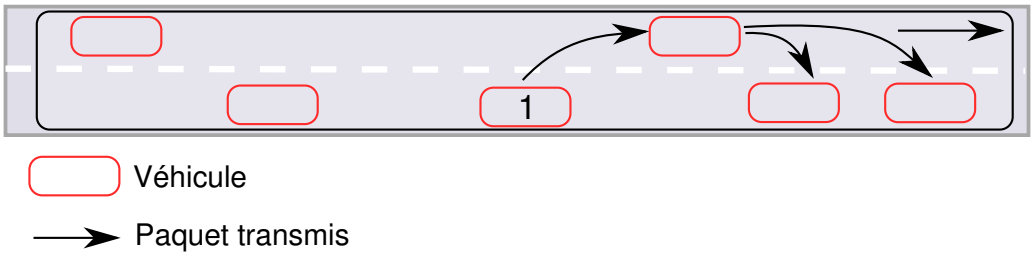


Figure 6.4 – Diffusion d'un paquet dans la direction de déplacement du convoi.

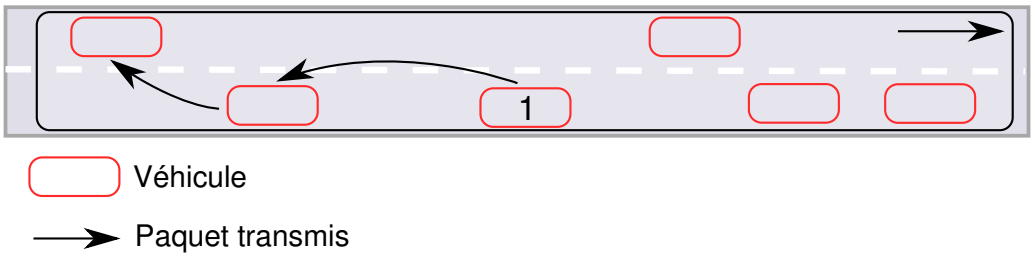


Figure 6.5 – Diffusion d'un paquet dans la direction opposée au déplacement du convoi.

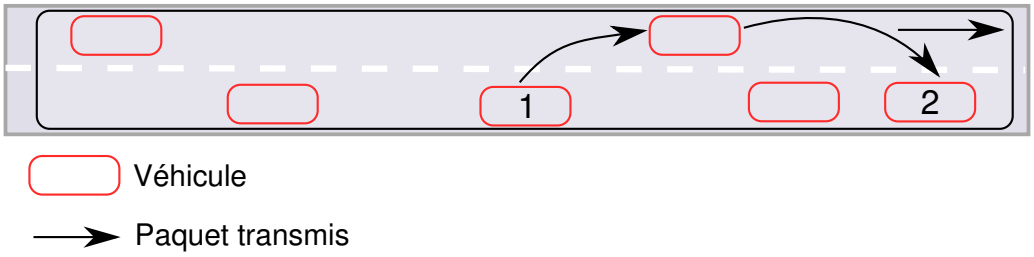


Figure 6.6 – Acheminement d'un paquet vers un nœud à une position prédéterminée.

Dans notre protocole de formation de convoi, les nœuds communiquent selon trois processus :

- diffusion d'un paquet dans la direction de déplacement du convoi (Figure 6.4),
- diffusion d'un paquet dans la direction opposée au déplacement du convoi (Figure 6.5),
- l'acheminement d'un paquet vers une position géographique relative au convoi (communication point-a-point) (Figure 6.6).

La diffusion d'un paquet dans la direction (respectivement dans la direction opposée) du déplacement du convoi permet de délivrer un message à tous les véhicules du convoi en amont (respectivement en aval) du véhicule initiateur du message.

Dans un but de simplicité, nous choisissons de séparer le protocole de formation de convoi en lui-même du protocole permettant la communication entre les nœuds. C'est pourquoi nous adoptons un protocole de routage simple basé sur la position géographique des véhicules pour les communications à l'intérieur d'un convoi. Le protocole de formation de convoi n'a donc pas la charge la transmission des messages.

6.2.2 Algorithme de formation de convois

Nous nous inspirons des algorithmes de formation de clusters, pour développer un algorithme de formation de convois. Le chef de convoi, ou clusterhead, est tout simplement le nœud en *tête de convoi*, c'est-à-dire le nœud devant tous les véhicules membres du convoi (Figure 6.7). Les algorithmes de formation de clusters que nous avons décrits limitent la taille du cluster soit par le nombre de sauts maximum au clusterhead ou soit par le nombre de nœuds membres du cluster. Notre algorithme limite la taille du cluster en limitant sa longueur, c'est-à-dire la distance maximale séparant la tête de convoi de la queue de convoi (Figure 6.7).

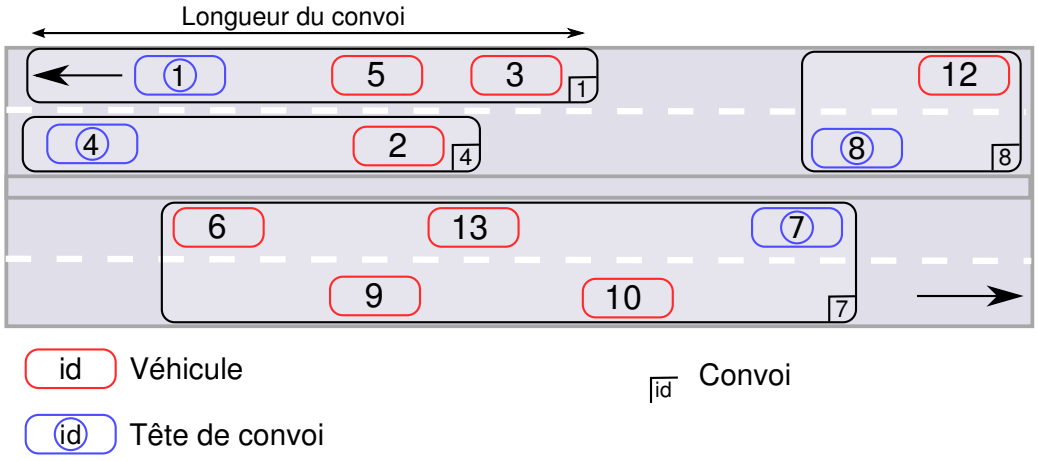


Figure 6.7 – Convois de véhicules sur une autoroute 2x2 voies. La tête de convoi est le nœud devant tous les véhicules du convoi.

Initialement tout véhicule est seul dans son propre convoi et donc tête de convoi. Alors il existe initialement autant de convois qu'il y a de véhicules dans le réseau. Un nœud déclare sa présence à ses voisins en envoyant à intervalles réguliers un message HELLO, contenant son identifiant, l'identifiant de son convoi, sa position et sa vitesse.

Un nœud peut communiquer avec la tête de convoi simplement en envoyant un message dans la direction de déplacement du convoi. La tête de convoi souhaitant diffuser un message à tous les nœuds du convoi, envoie un message dans la direction opposée au dé-

Algorithme 1 Variables, constante et initialisation d'un nœud

Constantes :
- Id ▷ Identifiant du nœud
- $LifeTimeLimit$ ▷ Durée de vie minimal d'un lien
- $ConvoyLengthLimit$ ▷ Longueur limite d'un convoi

Variables de nœud :
- $IdConvoy$ ▷ Identifiant du convoi auquel on appartient
- $ConvoyHead$ ▷ Vrai si le nœud est la tête de convoi
- $ConvoyLength$ ▷ Longueur actuelle du convoi
- $JCReqTable$ ▷ Ensemble des messages JCREQ envoyés
- $ClusterNodes$ ▷ Ensemble des nœuds appartenant au cluster

Variables extérieurs :
- $CurrentSpeed$ ▷ Vecteur vitesse courante
- $CurrentPos$ ▷ Position courante

procedure INITIALISATION
- $IdConvoy \leftarrow Id$
- $ConvoyLength \leftarrow 0$
- $ConvoyHead \leftarrow True$
- $JCReqTable \leftarrow \emptyset$
- $ClusterNodes \leftarrow \{Id\}$

end procedure

Nom du message	Sigle	Mode de diffusion
Join convoy request	JREQ	Direction du déplacement
Join convoy reply	JREP	Vers une position
Node join	NJ	Vers une position
Convoi info	CI	Direction opposée au déplacement
Acquittement d'un CI	CIACK	Direction du déplacement
Hello	HELLO	Voisinage

TABLE 6.1 – Liste des messages utilisés par l'algorithme de formation de convois.

placement du convoi. Nous avons vu que ces deux modes de diffusion sont gérés par le protocole de routage intra-cluster.

La formation et la maintenance de convoi sont régentées par deux processus distincts : la *fusion* et la *scission* de convois. La fusion permet à un convoi de s'agrandir en recrutant de nouveaux véhicules. La scission est provoquée quand le convoi n'est plus connexe : il existe un nœud ne pouvant plus communiquer avec la tête de convoi. La gestion de la scission de convoi permet la maintenance du cluster. L'ensemble des messages utilisés pour la formation et la maintenance des convois est résumé en Tableau 6.1. L'initialisation et la listes des variables et constantes d'un nœud sont décrites dans l'Algorithme 1. La constante $LifeTimeLimit$ correspond à la durée de vie minimale d'un lien pour rentrer dans un cluster.

6.2.2.1 Fusion de convois

Algorithme 2 Réception d'un message Hello. Paramètres : identifiant, position et vitesse du noeud. Identifiant du convoi.

```
1: function LIFETIME(speed, pos)
2:     d ← |pos − CurrentPos|
3:     if speed < ||CurrentSpeed|| then
4:         return (R − d) / (||CurrentSpeed|| − speed)
5:     end if
6:     if speed > ||CurrentSpeed|| then
7:         return (R + d) / (speed − ||CurrentSpeed||)
8:     end if
9:     return ∞ ;
10: end function

11: function SAMEDIRECTION(v1, v2)
12:     return v1 × v2 > 0                    ▷ Test si le produit scalaire est positif
13: end function

14: procedure HELLO(id, idConvoy, pos, speed)
15:     if ConvoyHead
16:         ∧ LIFETIME(||hello.speed||, hello.pos) < LifeTimeLimit
17:         ∧ SAMEDIRECTION(hello.speed, CurrentSpeed)
18:         ∧ IdConvoy ≠ hello.idConvoy
19:         ∧ CurrentPos < hello.pos
20:         ∧ hello.idConvoy ⊄ JCReqTable  then
21:         send JCREQ(hello.idConvoy, ConvoyLength)
22:     end if
23: end procedure
```

Dans la suite, nous allons décrire les algorithmes déclanchés lors de la réception d'un message. Chaque algorihtme contient une procédure de même nom que le message. Les paramètres de la procédure sont les paramètres du message. Par convention, nous nommons le paramètre d'un message utilisé à l'interieur de la procédure *message.paramètre*. De même, l'instruction *send* indique l'envoi d'un message et les paramètres du message sont entre parenthèse.

Pour pouvoir initier la fusion de convois C_1 et C_2, le clusterhead de C_1 (ou C_2) doit être le *voisin* (clusterhead ou non) d'un nœud contenu dans le cluster C_2 (ou C_1). À partir des informations contenues dans le message HELLO envoyé par le voisin, le clusterhead décide de se porter ou non candidat à la fusion (Algorithme 2).

Les conditions de candidature sont :

- La durée de vie du lien entre le clusterhead et le voisin est supérieure à la limite *LifeTimeLimit* fixée par le protocole, ce qui permet de garantir une durée de vie du cluster supérieure à cette limite (si nous considérons la vitesse des nœuds membres constantes)
- Les deux convois C_1 et C_2 se déplacent dans la même direction.

– C_1 et C_2 ont des identifiants différents, sinon la fusion n'a pas d'interêt.
– le nœud voisin est devant le clusterhead,
– une candidature de fusion entre C_1 et C_2 n'a pas déjà eu lieu récemment.

Après fusion, le clusterhead étant toujours devant tous les nœuds membres du cluster, le clusterhead candidat devient membre du cluster voisin.

Algorithme 3 Réception d'un message Join convoy request. Paramètres : Identifiant du convoi, taille du convoi (bornes)

```
1: procedure JCREQ(idConvoy, ConvoyLength)
2:     if ConvoyHead
3:         ∧ jcreq.ConvoyLength ∪ ConvoyLength < ConvoyLengthLimit then
4:         send JCREP(jcreq.idConvoy)
5:     end if
6: end procedure
```

Si toutes ces conditions sont vérifiées, le clusterhead candidat envoie un message *Join convoy request* (JCREQ) au nœud voisin qui le transmet à son clusterhead, que nous appelons *clusterhead voisin*. Le clusterhead voisin vérifie la taille des deux convois réunis (Algorithme 3), si elle est inférieure à la taille limite de convois, alors il envoie un message *Join convoy reply* (JCREP) au clusterhead candidat confirmant la fusion. Ce message est envoyé vers la position géographique de la destination. Si la taille dépasse la taille limite, alors le clusterhead voisin ne fait rien et le processus de fusion est avorté.

Algorithme 4 Réception d'un message Join convoy reply. Paramètres : Identifiant du convoi

```
1: procedure JCREP(idConvoy)
2:     IdConvoy ← jcrep.idConvoy
3:     ConvoyHead ← False
4:     send CI(IdConvoy)
5:     send NJ(ClusterNodes)
6: end procedure
```

Sur réception du message JCREP (Algorithme 4), le clusterhead candidat envoie un message *Node join* (NJ) au clusterhead voisin contenant la liste des nœuds de son cluster, diffuse un message *Convoi Info* (CI) dans la direction opposée au déplacement du cluster et affecte son identifiant de cluster à l'identifiant du clusterhead voisin.

Algorithme 5 Réception d'un message Node join. Paramètres Ensemble de nœuds du convoi à rejoindre

```
1: procedure NJ(nodes)
       ClusterNodes ← ClusterNodes ∪ nj.nodes
2: end procedure
```

Sur réception du message NJ (Algorithme 5), le clusterhead ajoute les nœuds contenus dans le message à sa table de nœuds du cluster. Sur réception d'un message CI (Algorithme 6), un membre du cluster affecte son identifiant de cluster à l'identifiant du clusterhead voisin. Ces deux actions permettent à tous les membres de l'ancien cluster de rejoindre le cluster voisin, le processus de fusion est donc terminé.

Algorithme 6 Réception d'un message Convoy info. Paramètres : Identifiant du convoi

```
1: procedure CI(idConvoy)
2:     IdConvoy ← ci.idConvoy
3:     ConvoyHead ← False
4:     send CIACK
5: end procedure
```

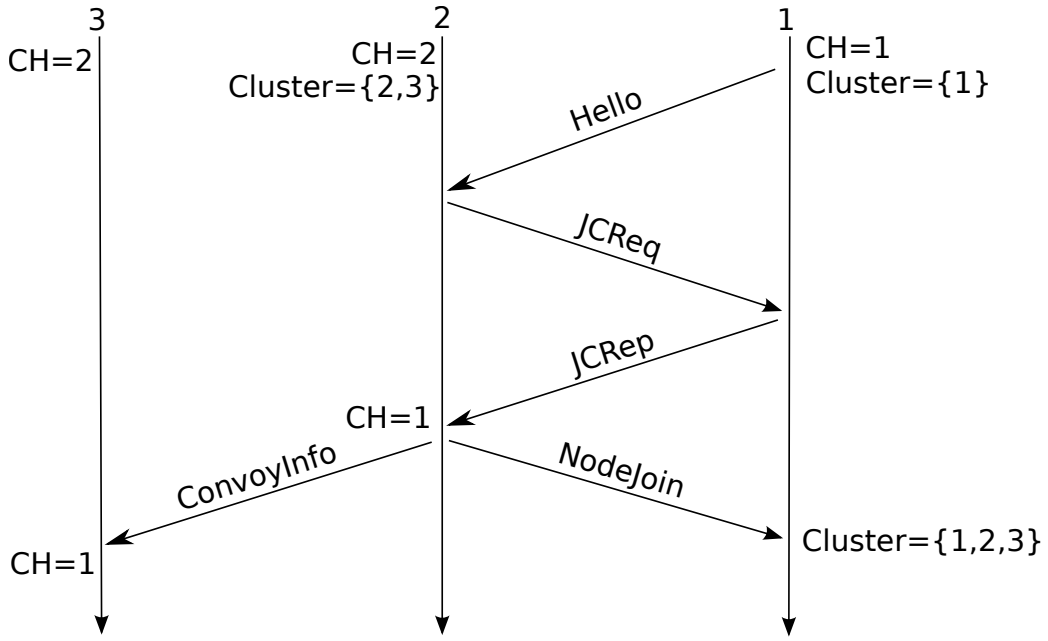


Figure 6.8 – Exemple d'un échange de messages lors d'une fusion de convois.

La Figure 6.8 est un exemple d'un échange de messages lors d'une fusion de convois entre le convoi 1 et 2. Tout d'abord, le nœud 2 reçoit un message HELLO du nœud 1. Il vérifie si les critères de fusion sont respectés et envoie un message JCREQ à 1 pour initier la fusion. Le nœud 1 vérifie la taille du futur convoi fusionné, et répond par un message JCREP afin de confirmer la fusion à 1. Sur réception de ce message, le nœud 2 affecte son identifiant de tête de convoi à 1, puis finalise la fusion en envoyant au nœud 2 la liste des nœuds appartenant à son convoi via le message NJ et enfin envoie au membre du convoi 3 un message CI pour l'avertir du changement de convoi. En recevant le message CI, le nœud 3 affecte son identifiant de tête de convoi à 1. Finalement, les nœuds 1, 2 et 3 ont bien leur identifiant de tête de convoi à 1, et le nœud 1 possède la liste de tous les membres de son convoi.

6.2.2.2 Scission de convois

La gestion de la scission de convois joue le rôle de maintenance du cluster. La détection de rupture d'un lien avec un membre voisin ne suffit pas à déduire que le nœud quitte le cluster. En effet il peut encore communiquer avec tous les membres du cluster et le cluster-

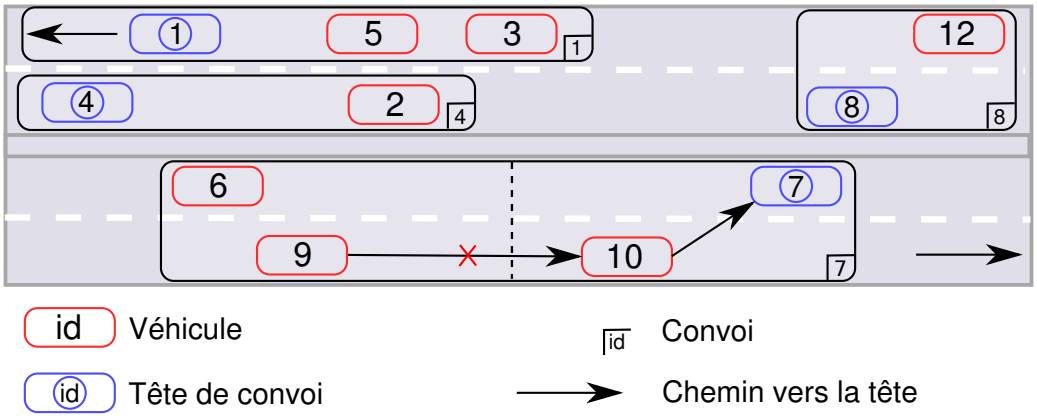


Figure 6.9 – Scission du convoi 7. Après scission, le convoi 7 subsiste et un nouveau convoi 9 s'est formé.

head par l'intermédiaire d'un autre lien. Par définition, la scission d'un convoi se produit quand un nœud membre ne peut plus communiquer avec le clusterhead, c'est-à-dire qu'il n'existe pas de chemin entre le nœud membre et le clusterhead (Figure 6.9). Le réseau étant linéaire, aucun des nœuds en aval du membre ne peut plus également communiquer avec le clusterhead. Le membre en tête devient alors clusterhead et l'ensemble des nœuds en aval appartenant à l'ancien cluster rejoignent le nouveau cluster.

Algorithme 7 Sur rupture de convois

```
1: procedure CONVOYBREAK(idConvoy)
2:     IdConvoy ← Id
3:     ConvoyHead ← True
4:     send CI(idConvoy)
5: end procedure
```

Le clusterhead diffuse à intervalles réguliers, dans la direction opposée au déplacement de cluster, un message *Convoi info* (CI) à tous les membres du cluster. Lorsqu'un membre ne reçoit pas ce message depuis un certain délai prédéterminé, il considère qu'il n'existe plus de chemin avec le clusterhead, le cluster n'est plus connexe et il y a scission du convoi. Il s'auto-proclame clusterhead en affectant son identifiant de cluster à son propre identifiant, et diffuse un message CI dans la direction opposée au déplacement du cluster, permettant aux nœuds en aval de changer de cluster (Algorithme 7).

Sur réception d'un message CI (Algorithme 6), un membre du cluster affecte son identifiant de cluster au nouvel identifiant du clusterhead et acquitte la réception du message à son nouveau clusterhead . Sur réception de l'acquittement, le nouveau clusterhead ajoute le membre du cluster à sa table. Le clusterhead en amont, ne recevant pas d'acquittement au message CI , considère que ses membres ne font plus partie de son cluster et il les enlève de sa table. Le processus de scission est alors terminé.

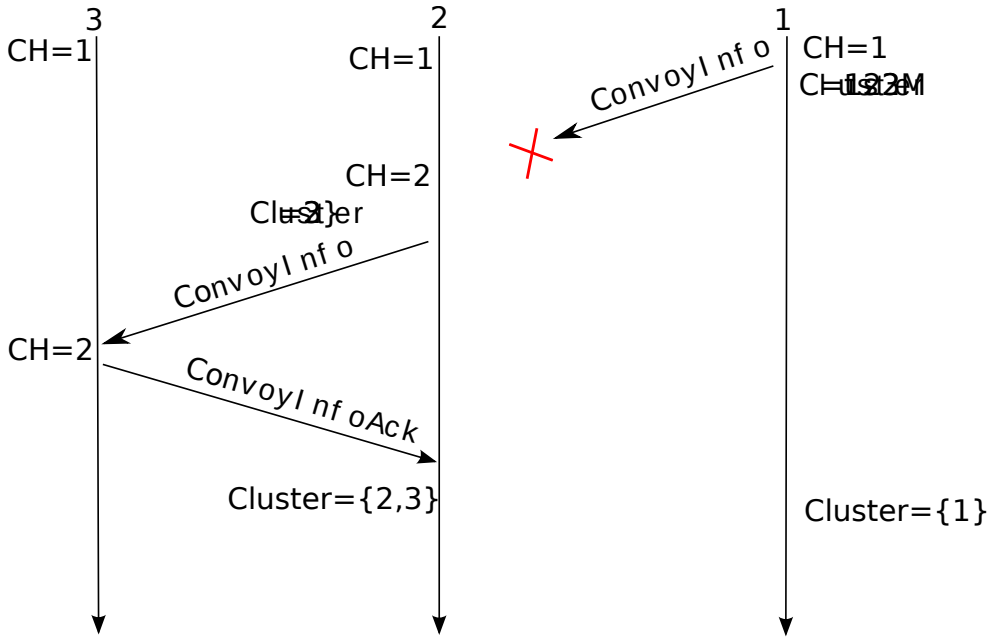


Figure 6.10 – Exemple d'un échange de messages lors d'une scission de convois.

La Figure 6.10 est un exemple d'un échange de messages lors de la scission du convoi 7. Lorsque le nœud 2 ne reçoit pas le message CI de sa tête de convoi 1, alors il change son identifiant de tête de convoi à 2 et s'ajoute à sa liste de membres du convoi. Pour avertir le changement de tête de convoi, le nœud 2 envoie au nœud 3 un message CI. Sur réception de ce message, le nœud 3 affecte sa tête de convoi à 2 et répond à 2 par un acquittement, lui permettant de l'ajouter à sa liste des membres. Le nœud 1 ne recevant plus de message CI de 2 et 3, il les supprime de sa liste des membres.

6.2.2.3 Gestion de fusion simultanée

Le cas d'une fusion simultanée se produit lorsqu'un clusterhead reçoit un message JREQ alors qu'il est dans un processus de fusion de convois, c'est-à-dire qu'il a envoyé un JREQ mais qu'il n'a pas encore reçu un message JREP. La figure 6.11 illustre un cas de fusion simultanée. Le CH2 reçoit un message JREQ de CH1 ; il répond par un message JREP, confirmant la fusion entre le cluster 1 et 2. Mais, CH2 a envoyé une requête de fusion à CH3 auparavant. Quand il reçoit le message JREP de CH3, les clusters 2 et 3 fusionnent et CH2 n'est plus alors clusterhead. Or pour CH1, CH2 est toujours clusterhead ; il lui envoie alors un message NJ. N'étant plus clusterhead, ce message ne pourra pas être traité par CH2 . Pour CH1, tous ses membres ont rejoint le cluster 2, alors que pour CH2 ils sont membres du cluster 3. Les nœuds du cluster sont dans un état incohérent.

Pour éviter ce problème, un clusterhead enregistre les requêtes de fusion lors de l'envoi du message JREQ. Quand le clusterhead envoie un message JCREP pour confirmer une fusion, il annule toutes les requêtes en cours en supprimant les enregistrements correspondants

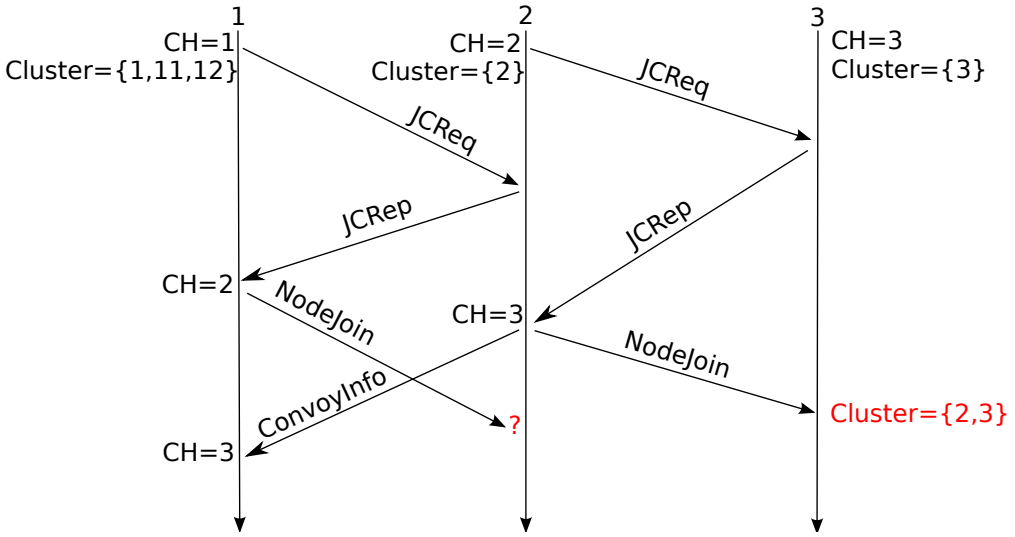


Figure 6.11 – À la fin des deux fusions entre 1 et 2 et 2 et 3, la tête de convoi 3 n'est pas le nœud membre 1 dans sa liste.

(Figure 6.12). Quand le clusterhead reçoit un message NJ pour finaliser la fusion, il annule également les requêtes en cours. Quand le clusterhead recevra la réponse à une requête, sur réception d'un message JCREP, il vérifie si l'enregistrement de la requête est toujours présente avant d'envoyer un message NJ finalisant la fusion.

Pour limiter la longueur du convoi, lorsque le clusterhead répond positivement à la fusion, il enregistre les bornes du cluster candidat contenu dans le message JREQ. Avant de recevoir le message NJ, le clusterhead peut envoyer une requête de fusion. La future longueur est calculée à l'aide des bornes des clusters candidats à la fusion, pour l'inclure dans le message JREQ (Figure 6.13).

Nous avons décrit notre protocole de formation de convois associant un routage géographique à l'intérieur d'un convoi et un algorithme réparti de formation de cluster. Dans la suite, nous avons implémenté ce protocole dans le simulateur de réseau JIST/SWANS afin d'évaluer la qualité de formation de convois dans un contexte de réseau de véhicules sur autoroute.

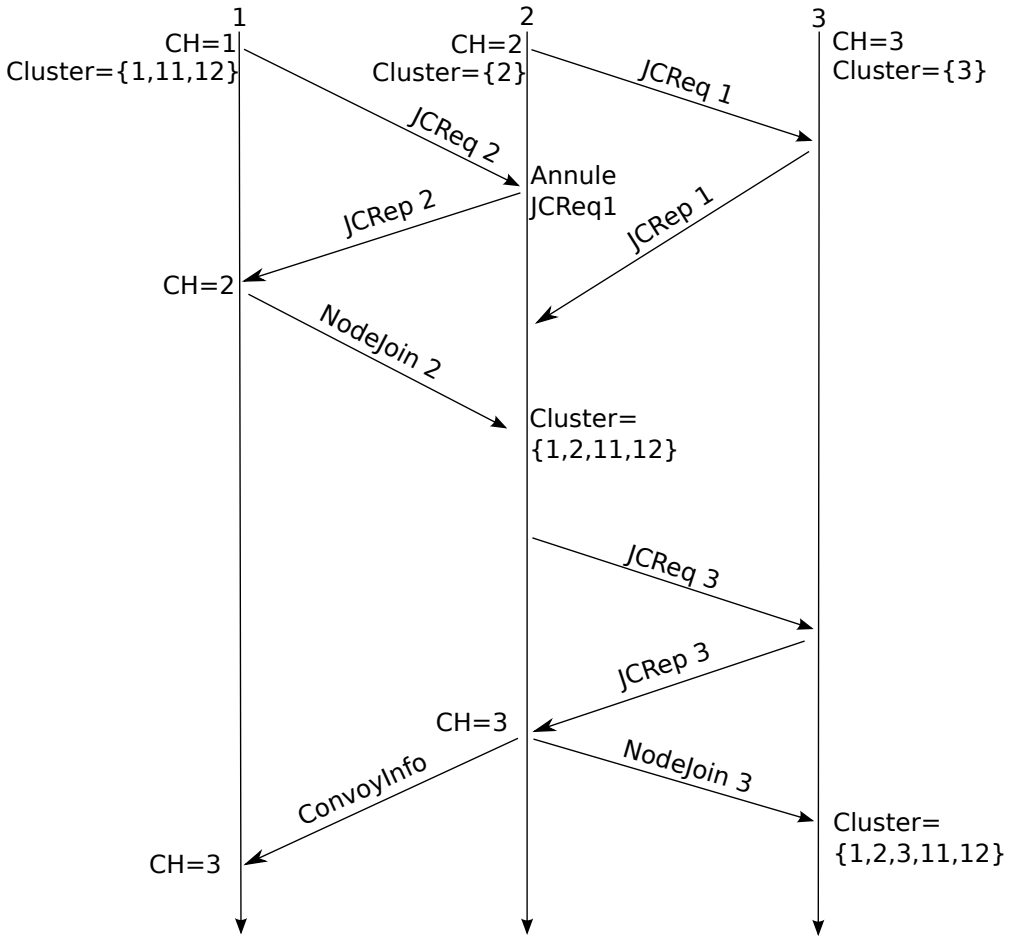


Figure 6.12 – L'envoi d'un message JCRep (confirmant une fusion) annule toute fusion en cours.

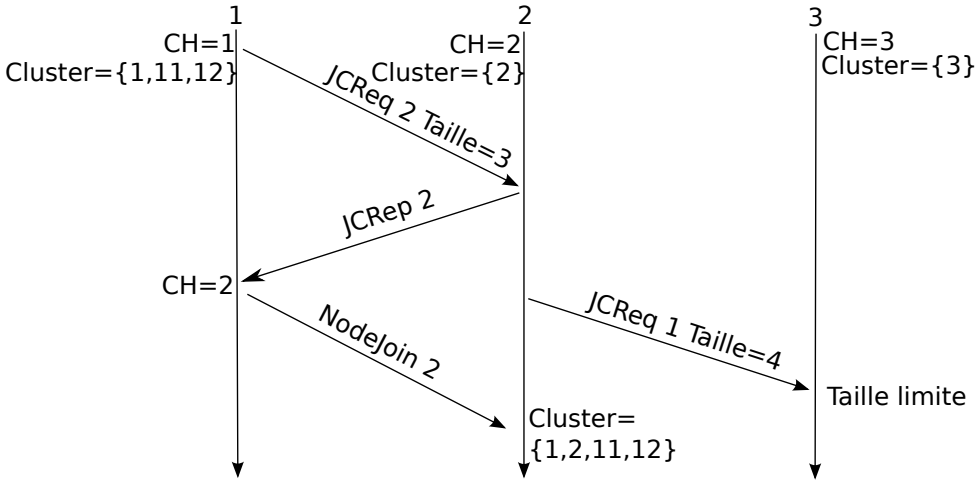


Figure 6.13 – La taille du convoi 2 inclus dans le message JCREQ1 prend en compte la future taille du convoi après fusion avec le convoi 1.

6.2.3 Évaluation du protocole de formation de convois

Nous avons implémenté l'algorithme de formation de convois dans le simulateur JIST/SWANS au niveau application (classe *jist.swans.app.AppConvoi*). En effet, le routage des messages envoyés et diffusés par le protocole de formation de convois est délégué au protocole de routage géographique décrit dans le paragraphe 3.4.1. Comme dans les simulations décrites dans les chapitres précédents, nous utilisons le simulateur de trafic routier pour modéliser la mobilité des nœuds dans un contexte autoroutier. En plus des paramètres liés au trafic routier (longueur de l'autoroute et densité de véhicules), nous utilisons quatres paramètres liés à la simulation de formation de convoi :

- Durée de simulation
- Intervalle d'envoi des messages HELLO
- Durée de vie limite d'un lien pour l'intégrer au convoi
- Longueur limite d'un convoi

Les paramètres par défaut sont donnés dans le Tableau 6.2. Dans la suite, si le paramètre ne varie pas, nous utilisons cette valeur par défaut.

Nous nous intéressons à cinq métriques :

- Nombre de ruptures de convois,
- Nombre de convoi,
- Taux de partitionnement,
- Taille d'un convoi en nombre de nœuds,
- Longueur d'un convoi en mètre,

Le *nombre de ruptures de convois* pendant une simulation permet d'évaluer la *stabilité* globale des convois formées : plus le nombre de ruptures est important, moins les convois sont stables.

Le *nombre de convois* permet d'évaluer la *qualité* de leur formation : au pire il existe autant de convois que de nœuds, chaque convoi étant alors composé d'un seul nœud, la formation de convoi n'a alors aucun intérêt. Plus il existe de convois, plus les convois sont gros, moins il sont résistants à la mobilité. Nous nous intéressons également au *taux de partitionnement* défini comme le nombre de convois divisé par le nombre de nœuds.

Nous différencions *longueur d'un convoi* et *taille d'un convoi*. La longueur d'un convoi est la distance entre la tête et la queue (véhicules derriere tous les membre du convoi) de convoi. La taille d'un convoi c'est le nombre de véhicules membres du convoi.

Temps de simulation	600 s
Intervalle des Hello	2 s
Durée de vie limite d'un lien du convoi	600 s
Longueur limite d'un convoi	2 km
Longueur de l'autoroute	10 km
Densité de véhicules	10 véhicules/voie/km

TABLE 6.2 – Paramètres de simulation par défaut

Dans un premier temps, nous cherchons les paramètres optimaux de l'algorithme de formation de convoi pour la durée de vie limite et la longueur limite. Dans un second temps, nous analysons la stabilité et la qualité des convois formés avec les paramètres optimaux trouvés.

6.2.3.1 Paramètres optimaux de l'algorithme

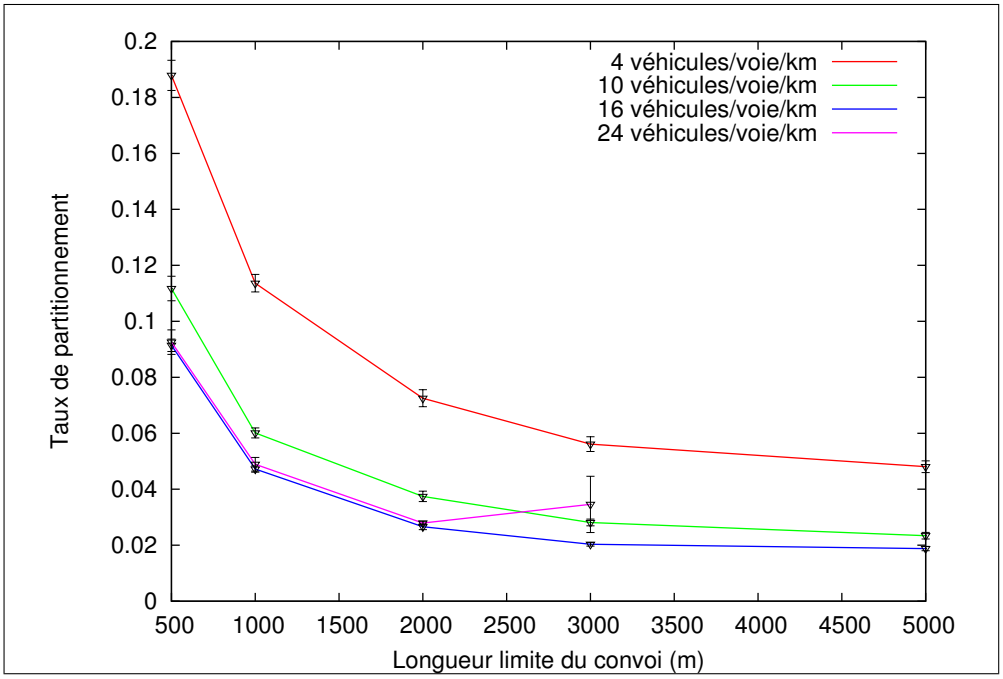


Figure 6.14 – Taux de partitionnement en fonction de la longueur limite de convoi

La connectivité d'un convoi est proportionnelle à sa densité. Plus un convoi est dense, plus la taille d'un convoi est potentiellement grand, diminuant le taux de partitionnement. Cette hypothèse est vérifiée sur les résultats en Figure 6.14 : le taux de partitionnement diminue avec la densité.

Dans cette même figure, on voit que le taux de partionnement est également proportionnel à la longueur limite du convoi (Figure 6.14). Par conséquent, le paramètre de longueur maximale des convois permet de limiter la taille des convois. La variation du taux de partitionnement est plus faible dès que la longueur limite est supérieure à 2000 m et quasi constante à partir de 3000 m.

Le nombre de ruptures de convois augmente avec la densité et la longueur limite des convois (Figure 6.15). Pour une densité de 24 véhicules/km/voie, le nombre de ruptures explose à partir de 2000 m. Au delà de 3000 m et avec une densité de 24, la puissance requise par le simulateur est trop importante pour réaliser une simulation, d'où l'absence de mesures.

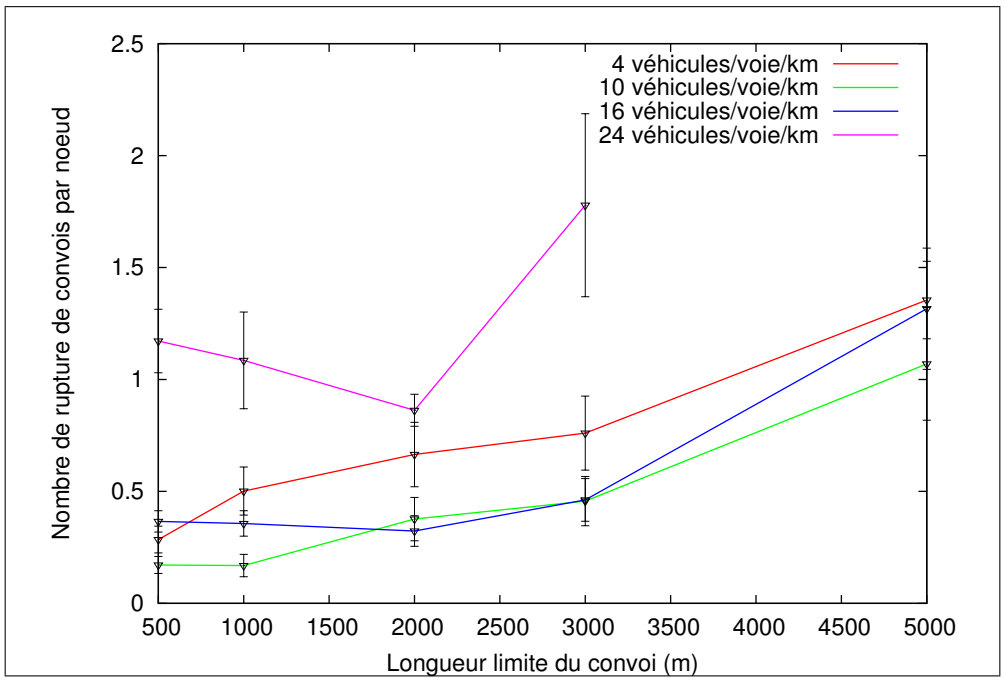


Figure 6.15 – Nombre de rupture de convois par noeud en fonction de la longueur limite de convoi

Nous concluons qu'il n'est pas utile de fixer la longueur limite d'un convois au delà de 2000 m pour satisfaire le compromis d'un taux de partitionnement proche du minimum et d'un nombre de ruptures de convoi minimal. Nous avons donc fixeé par défaut la limite de la longueur de convoi à 2000 m.

Le second paramètre de l'algorithme est la durée de vie minimale d'un lien pour l'accepter dans le convoi. Nous pensons augmenter la résistance à la mobilité en fixant ce paramètre de manière optimale. Cependant, le nombre de ruptures reste constant avec la durée de vie minimale d'un lien (Figure 6.17). Ce résultat est la conséquence d'une faible vitesse relative entre les nœuds d'un même convoi limitant la mobilité. En revanche, la durée de vie minimale d'un convoi peut également limiter leur taille moyenne (Figure 6.16), plus la durée de vie minimale d'un lien est grande, moins un convoi recrute de nœuds. Nous n'avons pas assez d'éléments pour définir une durée de vie minimale de lien optimale.

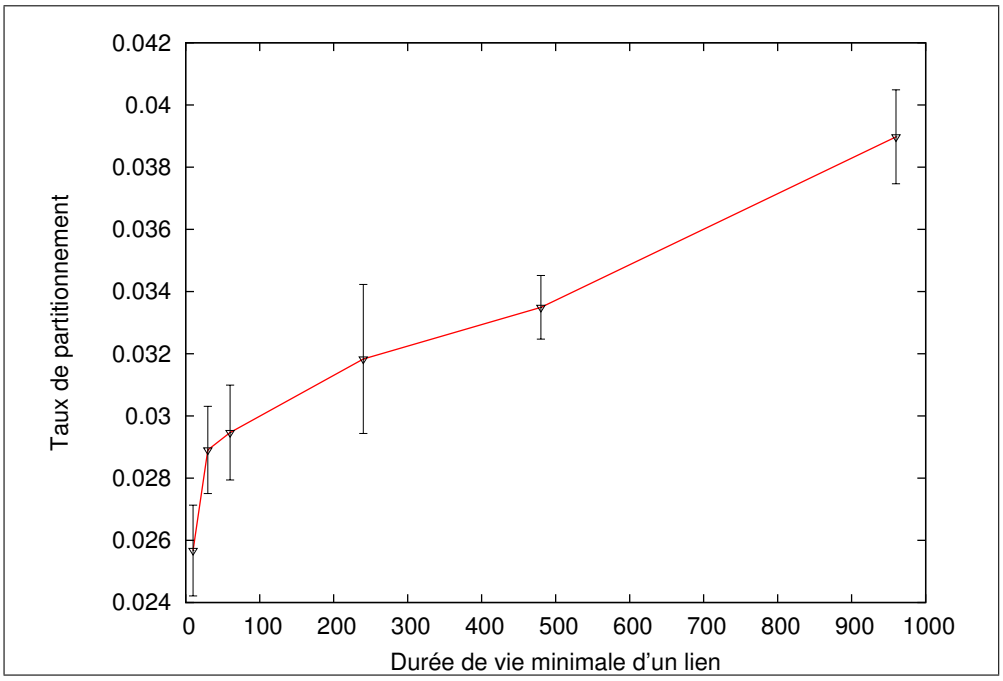


Figure 6.16 – Nombre de ruptures de convoi par noeud en fonction de la durée de vie maximale d'un lien

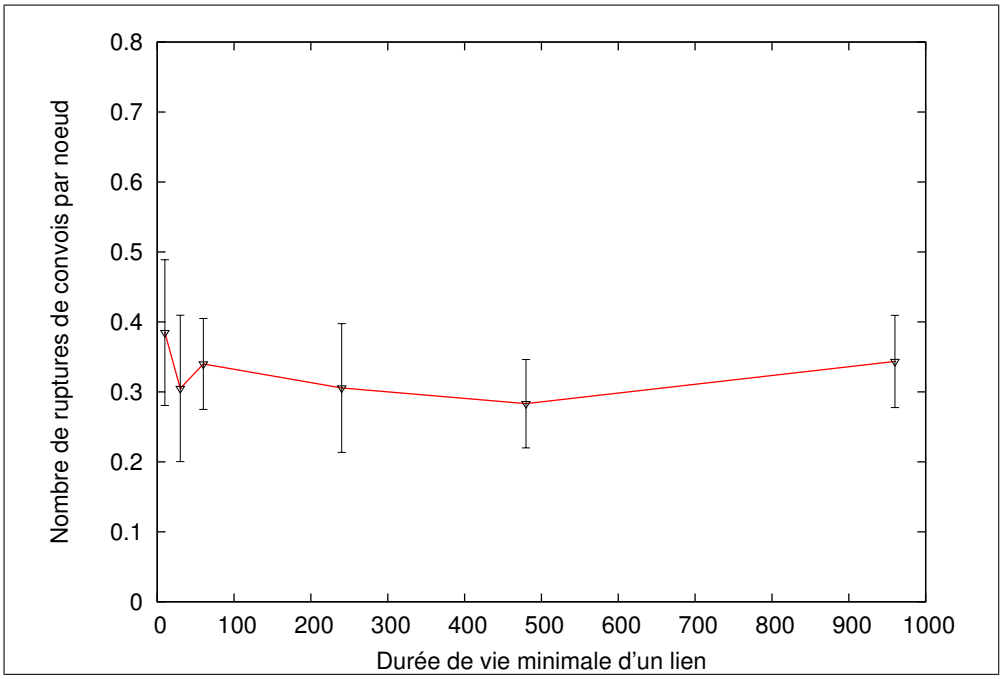


Figure 6.17 – Taux de partitionnement en fonction de la durée de vie minimale d'un lien

6.2.3.2 Coupure des convois

Nous avons discuté de la limite de longueur de convoi et nous souhaitons maintenant vérifier la distribution des longueurs de convoi. Cette distribution est calculée à la fin de la simulation. La Figure 6.18 présente cette distribution pour une limite fixée à 2000, 3000 et 5000 m. Chaque barre représente le nombre de convois compris entre une longueur l et $l + 500$ m sur un échantillons de 100 convois environ réalisé en plusieures simulations.

Nous remarquons deux anomalies apparentes : une proportion importante de convois de longueur comprise entre 0 et 500 m, et des convois de taille supérieure à la limite fixée. Quand un nœud arrive sur l'autoroute, il est tête de convoi et seul dans le convoi (aucun membre ne l'a encore rejoint). Il reste dans un temps limité seul avant de fusionner avec un autre convoi. Chaque nœud ayant été au départ dans un convoi de longueur zero, la proportion de petits convois proches de la longueur zero (un unique nœud dans le convoi) est naturellement importante, mais non significative. En effet un convoi reste peu de temps avec une taille inférieure à 10, comme nous le constatons sur la Figure 6.18.

La seconde anomalie constatée est l'existence de convoi de taille supérieure à la taille limite (2000 m). Les vitesses des véhicules étant proches entre-elle, nous fixons la limite de longueur de convoi seulement au moment de sa formation, c'est-à-dire lors de la fusion. Deux convois ne peuvent pas fusionner si la longueur totale des deux convois est supérieure à la limite. En revanche, la topologie du convoi est dynamique : la distance relative entre les véhicules varie au cours du temps. Il est possible qu'un véhicule ayant une vitesse plus faible par rapport aux autres s'éloigne des autres véhicules du convoi et augmente la longueur du

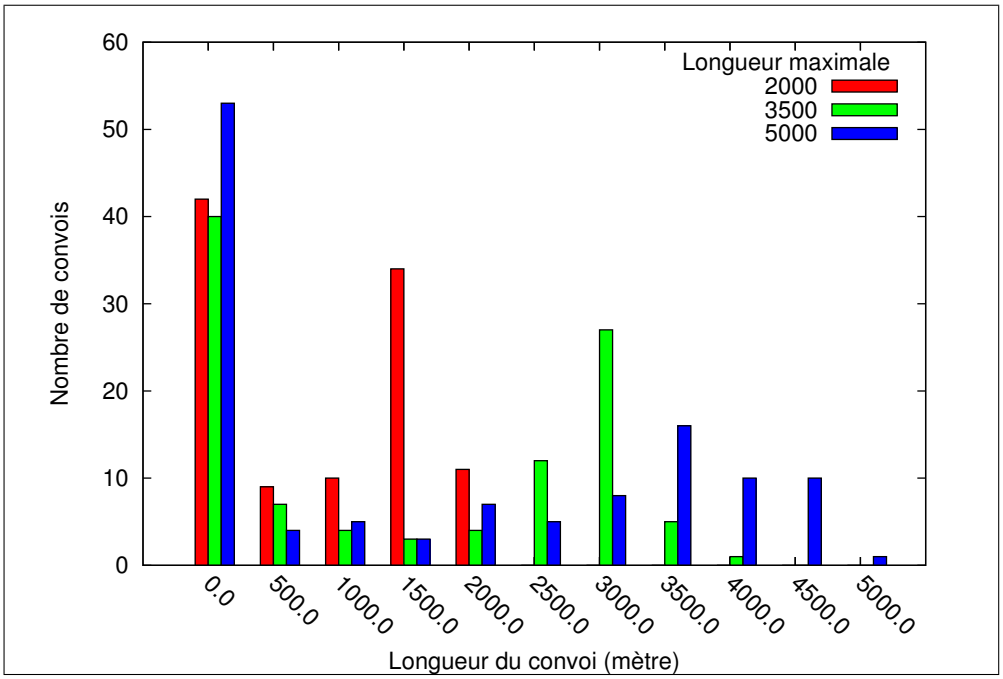


Figure 6.18 – Distribution des longueurs de convoi en metre

convoi au cours du temps. Par conséquent, bien qu'aucun nouveau véhicule n'ait rejoint le convoi, la longueur du convoi peut augmenter. Ce dépassement étant limité, nous ne considérons pas nécessaire de provoquer une rupture de convoi dans ce cas là.

La distribution est conforme aux attentes, notamment pour les convois de longueur 2000 et 3000 m, le maximum étant respectivement pour des convois de longueur compris entre 1500 et 2000 m et 3000 et 3500 m. Pour une limite de 5000 m, la distribution est plus répartie, le maintien de convoi supérieur à 3000 m étant plus difficile à cause d'une rupture de convoi plus importante (Figure 6.15).

6.2.3.3 Influence de la densité sur la formation de convoi

Nous souhaitons tout d'abord vérifier l'influence de la densité du trafic routier sur les convois. Comme nous l'avons vu dans les résultats des chapitres précédents, la mobilité augmente avec la densité du trafic. Nous retrouvons ce résultat en Figure 6.19 où le nombre de ruptures de chemin augmente avec la densité de véhicules.

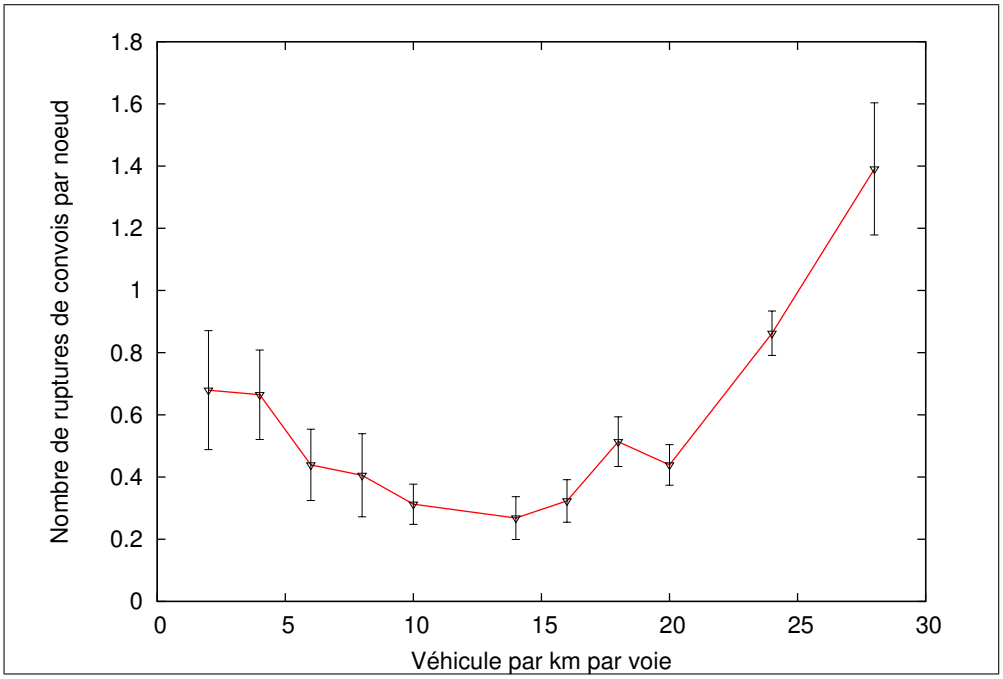


Figure 6.19 – Nombre de rupture de convois par noeud en fonction de la densité du trafic

L'objectif de la formation de convois est d'obtenir le moins de convois possibles donc des convois les plus gros possibles. En revanche, comme nous l'avons vu, le nombre de ruptures de convoi est fonction de la taille des convois. Nous avons trouvé un compromis avec un convoi limité à 2000 m. Nous nous intéressons à la durée d'un convoi ayant une taille n, avec une longueur limite de 2000 m.

Sur la Figure 6.20, nous mesurons la distribution de la durée de simulation d'un convoi en fonction de sa taille. Ici la durée de vie est exprimée en pourcentage de simulation, c'est-

à-dire la part de temps pendant laquelle le convoi a une taille comprise dans l'intervalle. La répartition de la taille de convoi dépend de la densité, en effet avec une densité de véhicules à 4 véhicules par kilomètre et par voie, la taille est répartie entre 0 et 40, pour une densité de 10 entre 0 et 80, pour une densité de 16 entre 0 et 140.

Nous remarquons également que les convois de grande taille sont ceux qui durent le plus longtemps, surtout avec des densités de 4 et 10 véhicules par kilomètre et par voie. Cette propriété est très intéressante pour obtenir des convois stables ayant une taille suffisamment grande pour justifier de la formation de convois dans un réseau de véhicules.

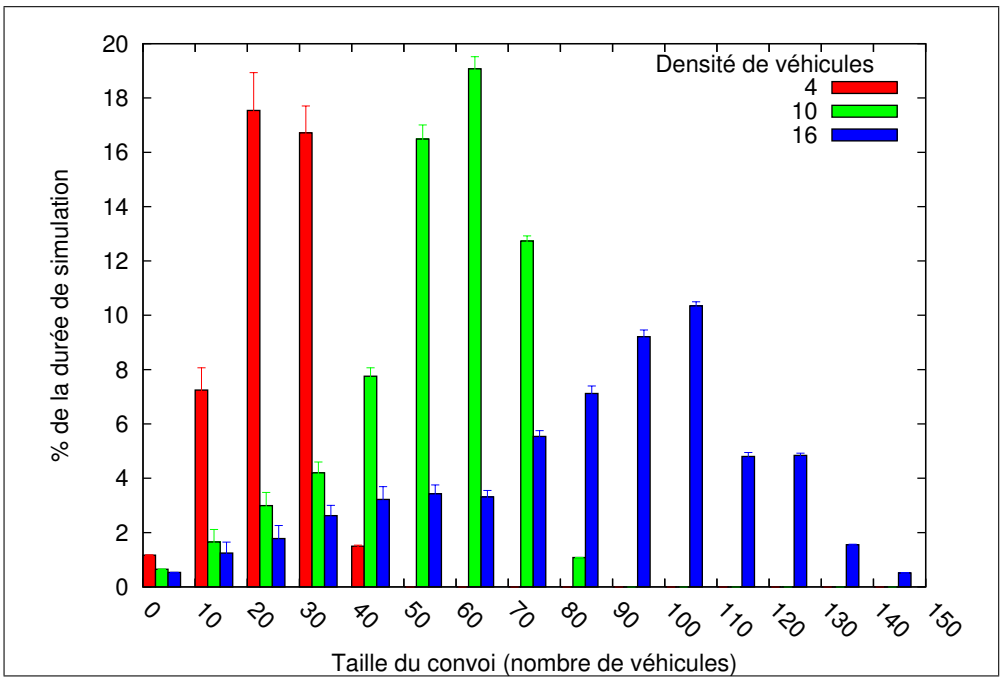


Figure 6.20 – Distribution de la durée d'un convoi en fonction de sa taille

6.3 Gestion de la mobilité dans un réseau ad hoc hybride

Dans le chapitre 5 nous avons recensé deux problèmes principaux dans un réseau ad hoc hybride de véhicules : la vitesse relative entre un véhicule et un point d'accès et l'accès au point d'accès multi-sauts. La vitesse relative élevée entre un véhicule et un point d'accès a pour conséquence une durée de connexion faible entre un nœud mobile et un point d'accès et donc une fréquence de changement de point d'accès élevé. Pendant le changement de point d'accès, les paquets en provenance de l'infrastructure ne peuvent plus atteindre le nœud mobile, provoquant leurs pertes. La prédiction de coupure de connexion avec le point d'accès permet de réduire ce délai, mais cette prédiction est peu fiable lorsque le chemin entre le nœud mobile et le point d'accès est multi-sauts. D'où le second problème.

La mobilité entre les nœuds étant faible sur une autoroute, contrairement à la mobilité entre un nœud mobile (véhicules) et un nœud fixe (AP) qui est élevée (vitesse relative égale à la vitesse du véhicule), nous séparons le problème de la mobilité dans un réseau ad hoc hybride en deux sous-problèmes : la mobilité de type véhicule-véhicule (VV) et la mobilité de type véhicule-infrastructure (VI).

Le première type de mobilité est géré par la formation de convois à l'aide d'un algorithme de formation de clusters : chaque convoi a la responsabilité de gérer la communication entre tout nœud du convoi.

Le second type de mobilité est géré à l'aide d'un protocole d'enregistrement du convoi auprès de l'infrastructure, similaire à celui d'enregistrement d'un nœud présenté dans le chapitre 5. L'enregistrement ne s'effectuant plus individuellement pour chaque nœud mais pour chaque groupe de nœuds, nous pensons pouvoir réduire l'overhead lié à l'enregistrement auprès de l'infrastructure.

Dans cette section, nous proposons une solution permettant d'une part la découverte de passerelle vers l'infrastructure, et d'autre part, le choix de la meilleure passerelle. La découverte de passerelles pouvant être considérée comme un service, nous avons intégré à notre convoi un protocole de découverte de service que nous détaillons dans la section suivant. Nous détaillerons également en quoi notre solution peut améliorer la perte de paquets pendant un handover.

6.3.1 Découverte de services intra-cluster

La découverte de services est traitée dans de nombreux travaux [37]. Le passage à l'échelle de ces protocoles dans un réseau ad hoc est un des sujets d'étude. Dans [76] les auteurs proposent l'utilisation de clusters pour améliorer le passage à l'échelle d'un protocole de découverte de services. Nous nous inspirons de ces travaux pour réaliser la découverte de services à l'intérieur d'un convoi.

Il existe deux modèles de découverte de services : *push* et *pull*. Dans le modèle push, le service s'annonce sur le réseau en diffusant un message d'annonce contenant les caractéristiques du service. Dans le modèle pull un nœud diffuse une requête de découverte de services. Quand le nœud contenant le service recherché reçoit cette requête, il répond directement au nœud en unicast.

Un annuaire peut également être ajouté pour agréger toutes les informations sur les services proposés par les nœuds du réseau. Un nœud contenant un ou plusieurs services envoie une requête d'enregistrement à l'annuaire. L'annuaire ajoute à sa base de données de services les informations concernant le nœud et le service. Les deux modèles push et pull sont également mis en œuvre pour découvrir les services contenus dans l'annuaire : le nœud diffuse une requête sur le réseau, l'annuaire lui répond, ou alors l'annuaire envoie à un intervalles réguliers un message d'annonce contenant l'ensemble des services contenus dans sa base de données.

L'annuaire est également bien plus qu'une base de données car il peut filtrer les services. Par exemple, s'il enregistre plusieurs services identiques mais localisés sur des nœuds différents, il sélectionne un seul nœud, et diffuse seulement les informations sur le service concernant ce nœud.

Il existe alors trois entité dans le réseau :
– User agent (UA) : demandeur du service
– Service agent (SA) : fournisseur de services
– Directory agent (DA) : annuaire de services

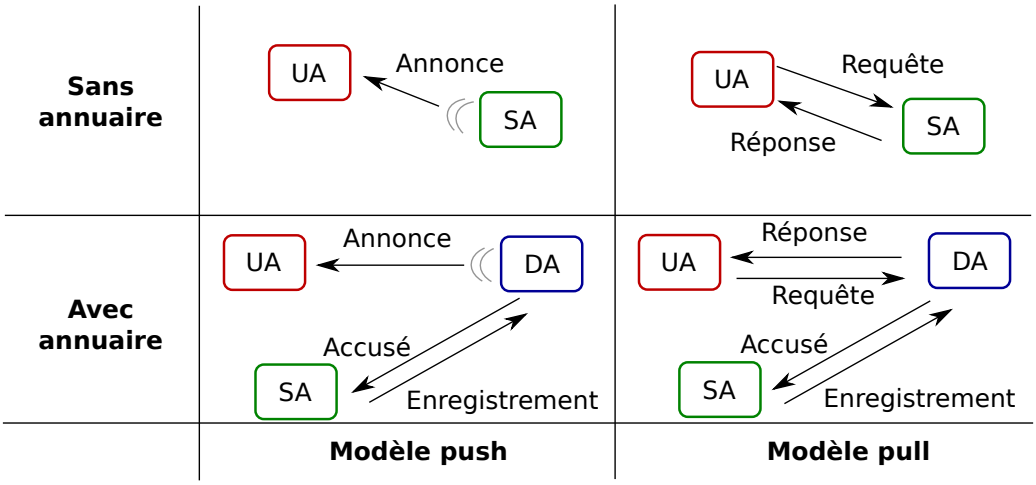


Figure 6.21 – Quatre modèles de découverte de services

Les différents modèles de découverte de services sont présentés sur la Figure 6.21. Les protocoles les plus connus utilisant ces modèles et mis en œuvre dans les réseaux actuels sont SLP, UPnPn, Jini et Salutation.

Dans notre convoi, nous adoptons une architecture à annuaire avec le modèle de découverte push. La tête de convoi étant supposé connu et accessible par tous les nœuds du convoi, nous lui affectons le rôle d'annuaire (Figure 6.22). C'est pourquoi les nœuds membres s'adresse à la tête de convoi pour enregistrer leurs services et la tête de convoi diffuse l'ensemble des services du convoi.

L'information sur les services proposés par les nœuds du convoi sont contenus dans le message CI, ce message étant déjà envoyé à intervalles réguliers, il est pas nécessaire d'ajouter un message d'annonce de services spécifiques. Un nœud possédant un service

s'enregistre auprès de l'annuaire à l'aide d'un message d'enregistrement d'un service *service register* (SVCREG), envoyé dans la direction de déplacement du convoi pour atteindre la tête de convoi. Pour actualiser les informations sur le service, nous ajoutons ces informations au message CIACK. Chaque nœud peut ainsi enregistrer ses services auprès de l'annuaire, et connaître les services fournis par les nœuds du convoi.

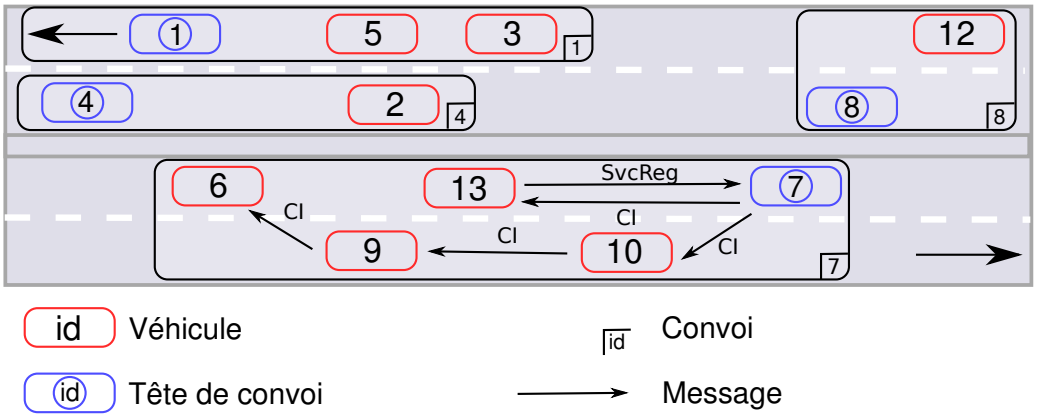


Figure 6.22 – Découverte de service dans le convoi 7. 8 enregistre son service auprès de la tête de convoi.

6.3.2 Service de localisation

La tête de convoi connaît la position de tous les nœuds du convoi à la réception du message JOIN et l'actualise à la réception du message CIACK. Ainsi, la tête de convoi fournit un service de localisation, un nœud du convoi peut connaître la position d'un autre nœud du convoi à l'aide des messages LREQ et LREP.

6.3.3 Service de passerelle

Dans un réseau ad hoc hybride, une passerelle (gateway) est un nœud mobile pouvant à la fois communiquer avec d'autres nœuds mobiles et un point d'accès. Nous considérons qu'il existe une connexion au niveau de la couche physique entre le nœud mobile et le point d'accès, et qu'il est possible d'obtenir au niveau application des informations concernant la qualité du lien, c'est-à-dire la puissance du signal, le taux d'erreur ou le bruit.

Une passerelle a pour rôle de router un paquet à destination d'Internet vers le point d'accès, et les paquets reçus par le point d'accès vers le nœud mobile. La passerelle obtient la position de la destination à l'aide du service de localisation.

6.3.4 Gestion des handovers

La gestion de la mobilité par nœud impose l'enregistrement de chaque nœud auprès de l'infrastructure, comme nous l'avons vu dans le chapitre 5. Ici, le convoi regroupe un ensemble de nœuds avec des différences de vitesse faibles. Donc l'enregistrement du convoi auprès de l'infrastructure suffit à gérer la mobilité de tous les nœuds du convoi.

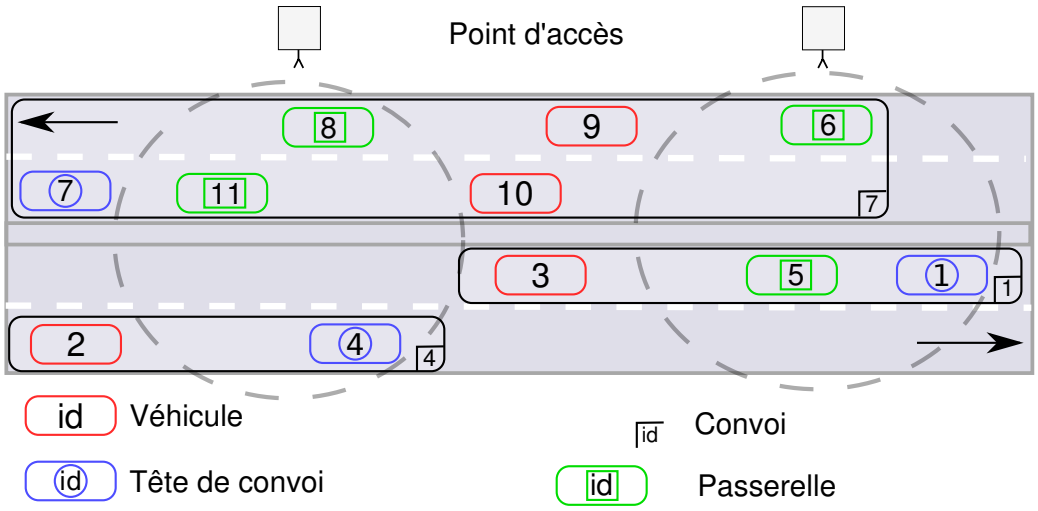


Figure 6.23 – Le convoi 7 contient trois passerelles connectées a deux points d'accès différents.

Dans un convoi, il peut exister plusieurs passerelles. Dans un convoi suffisamment grand (dont la longueur est supérieure à la distance entre points d'accès), il existe au moins deux passerelles connectées à des AP différentes (Figure 6.23). C'est cette propriété que nous exploitons pour effectuer un handover sans coupure. En effet, nous avons vu dans un protocole de micro-mobilité cellulaire tel que CIP ou HAWAII, un handover sans coupure utilise deux interfaces réseau pour pouvoir être connecté à deux AP simultanément. Nous utilisons cette méthode, non pas avec un nœud connecté à plusieurs AP simultanément, mais avec un convoi connecté à plusieurs AP simultanément, la mobilité étant gérée au niveau d'un convoi.

L'utilisation de passerelles (ou routeur) multiples avec NEMO (§ 1.2.2) est présenté dans [73, 98] : plusieurs routeurs mobiles d'un même réseau mobile peuvent être associés à des AR (*access router*, équivalent d'un point d'accès). Néanmoins, dans le cas de NEMO, un routeur mobile est un nœud spécialisé. Dans un convoi, tout nœud mobile peut être passerelle car nous conservons la propriété d'un réseau ad hoc : un nœud est à la fois terminal et routeur.

Chaque passerelle enregistre son service en fournissant une *métrique de qualité de lien* correspondant à la qualité du lien avec le point d'accès et ses voisins. Il doit également actualiser ces informations en répondant aux messages CI, reçus à intervalles réguliers, avec un CIACK.

Nous avons vu que l'annuaire (la tête de convoi) peut sélectionner le meilleur nœud fournissant des services similaires. Le choix de la meilleure passerelle, nommée *passerelle*

courante, est donc réalisé par la tête de convoi à l'aide des informations envoyées par les passerelles : son identifiant, identifiant de l'AP avec lequel elle est connectée, et métrique de qualité de lien. Ces informations sont envoyées à chaque enregistrement ou actualisation d'un service. Le choix de la meilleure passerelle peut utiliser une stratégie de *link hysteresis*, comme le fait OLSR pour le choix des MPR [41].

La tête de convoi doit aussi prédire une rupture de lien soit entre la passerelle et le convoi, et surtout entre la passerelle et le point d'accès. Si la tête de convoi considère qu'une rupture de lien est imminente (la métrique de qualité de lien passe en dessous d'un seuil critique), il choisit une meilleure passerelle courante.

Lorsque le choix de la meilleure passerelle est fixé, et qu'elle est différente de celle courante, l'identifiant de passerelle courante est modifié puis diffusé immédiatement via une annonce de service (CI) permettant aux nœuds du convoi d'actualiser les informations liées au service de passerelle et donc changer de passerelle. Sur réception de ce message, la nouvelle passerelle courante envoi une requête d'enregistrement *Binding Update* BU à l'infrastructure.

La Figure 6.24 est un scénario de changement de passerelle courante connectée à des AP différentes.

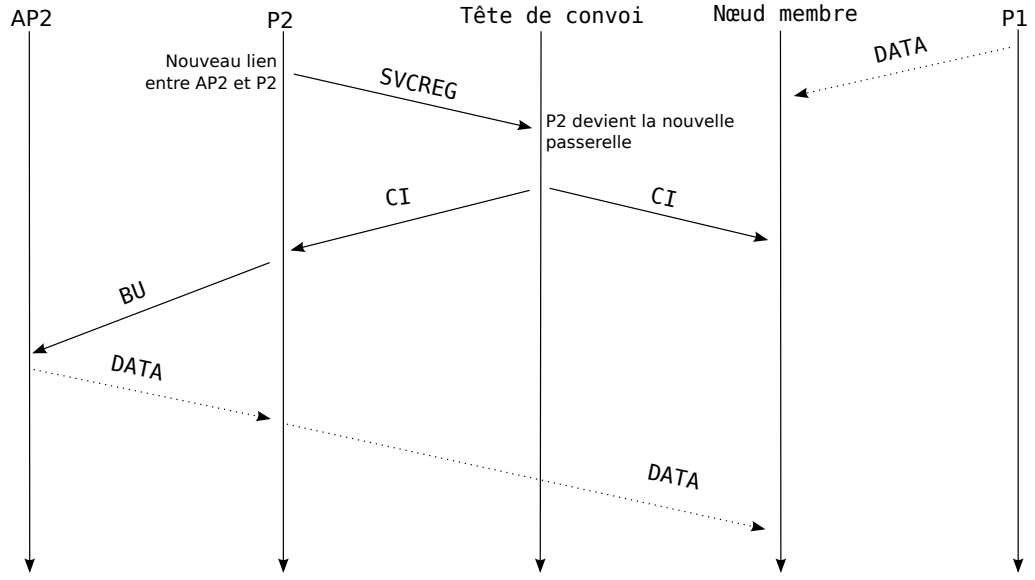


Figure 6.24 – Le convoi découvre une nouvelle passerelle P2 vers l'AP2. Après le handover, les données délivrées par l'ancienne passerelle P1 via l'AP1 sont délivrées par P2 via l'AP2.

6.3.5 Optimisations possibles

Deux optimisations sont possibles pour limiter le nombre de messages envoyés. Tout d'abord, chaque nœud peut écouter les messages CIACK à destination de la tête de convoi pour remplir un cache de position des nœuds du réseau. En effet, un CIACK contient la

position relative d'un nœud, une passerelle peut alors conserver cette position pour éviter d'envoyer une requête de localisation à la tête de convoi.

Une passerelle peut être le relai d'un message CIACK ou SVCREG contenant une information sur le service d'une autre passerelle. La passerelle relais peut regarder si la qualité du lien contenu dans le message est inférieure à celle de son propre lien avec le point d'accès. Si c'est le cas, elle peut choisir de ne pas retransmettre le message, car la tête de convoi ne choisira pas cette passerelle comme passerelle courante.

Le choix de la bonne métrique de qualité du lien est également important : plusieurs paramètres peuvent être pris en compte, comme le bruit ou la puissance du signal.

Conclusion

Nous avons présenté un algorithme de formation de clusters original adapté au réseau de véhicules, appelée formation de convois. Le convoi est formé en prenant en compte la qualité des liens entre les nœuds et sa longueur maximale afin de former des convois les plus stables possible. Les simulations réalisées à l'aide d'un simulateur de réseaux de véhicules dans un contexte autoroutier ont montré la bonne stabilité des convois (peu de rupture de convoi) et une distribution de la taille et longueur de convois satisfaisante.

Ces résultats nous permettent de penser qu'il est possible d'exploiter les propriétés d'un convoi (stabilité relative entre les nœuds d'un convoi), afin de gérer plus finement la mobilité. Nous avons présenté un service de passerelle (nœud connecté à une AP) intra-convoi pour fournir un accès à l'infrastructure, et donc à Internet, à tous les nœuds du convoi. Contrairement au réseau ad-hoc hybride classique, ce n'est plus un nœud qui s'enregistre individuellement à l'infrastructure, mais le convoi contenant des nœuds. Ceci s'apparente à la mobilité de réseau, mais ici le réseau est formé comme un réseau ad hoc : chaque nœud pouvant être routeur. Nous avons également exploité la possibilité pour le convoi d'être connecté à plusieurs AP simultanément via plusieurs passerelles. Cette propriété permet d'effectuer un handover sans coupure à condition que le convoi soit suffisamment grand par rapport à la distance séparant deux AP. Les résultat ont montré que la formation de convoi de longueur proche de 2 km et de longue durée est possible dans un réseau de véhicules sur autoroute. Avec un réseau ad hoc hybride et un intervalle de point d'accès égal à 1 km, chaque convoi peut être connecté simultanément à deux points d'accès, garantissant des handovers sans coupure à l'aide de la découverte de service de passerelle présenté précédemment.

Conclusion

Nous avons présenté nos travaux sur les réseaux de véhicules en environnement autoroutier. Ce cadre présente des particularités car les véhicules y sont très rapides ce qui provoque des coupures de connexion et des changements de topologie fréquents. La densité des véhicules peut être faible rendant le réseau peu connexe sur une grande étendue. Pour palier ces problèmes, nous proposons donc de mettre en place un réseau ad hoc hybride où les communications se font à la fois de véhicule à véhicule ou via une infrastructure de points d'accès interconnectés entre eux. Ainsi, nous traitons le problème de manque de connectivité du réseau ad hoc de véhicules. Mais cette solution nous permet aussi de traiter deux autres problèmes importants : la gestion de la forte dynamicité et le passage à l'échelle.

L'étude de performance des protocoles de routage ad hoc et notamment pour les réseaux de véhicules, est réalisée dans la majorité des cas par un simulateur de réseaux. Ce qui nous a amené à étudier un grand nombre de simulateurs existants. Parmi ceux-ci, nous avons choisi le simulateur JIST/SWANS pour l'évaluation de nos travaux car il est modulaire et efficace. Une part de nos travaux a donc porté sur des modifications du simulateur et l'enrichissement de la bibliothèque de protocoles et de modèles : DSR étendu aux réseaux hybrides, gestion de convois, mobilité d'un réseau de véhicules sur autoroutes, routage basé sur la position géographique.

Les simulations sont très dépendantes de leur implémentation et de leurs conditions expérimentales et sont difficilement reproductibles d'une étude à l'autre. Notre objectif était de comparer les protocoles de routage adaptés aux réseaux de véhicules de manière indépendante de leur mise en œuvre dans un simulateur de réseaux afin de caractériser leur aptitude à passer à l'échelle. Nous avons donc proposé une méthode alternative à la simulation seule pour évaluer le passage à l'échelle d'un réseau de véhicules. Cette méthode est basée sur un modèle quantitatif d'évaluation *hybride* de l'overhead, basé sur une modélisation fine des protocoles, notamment le nombre de messages de signalisation généré pour chaque événement et sur une simulation des paramètres non aisément quantifiables. Nous avons considéré d'une part un protocole de routage réactif et d'autre part, un protocole de routage géographique dont l'overhead dépend essentiellement du service de localisation. Nous concluons que l'utilisation d'un protocole géographique avec CBF (Contention-Based Forwarding) et un protocole à rendez-vous pour la localisation permet un meilleur passage

à l'échelle d'un réseau de véhicules sur autoroute.

Dans un réseau de véhicules sur autoroute, les problèmes de connectivité du réseau empêchent la communication entre véhicules sur de grandes distances. L'aspect linéaire de ces réseaux réduit également considérablement leur capacité. L'ajout de points d'accès au réseau permet de réduire l'impact de ces problèmes tout en réduisant les coûts par rapport à un réseau totalement cellulaire où les points d'accès couvrent entièrement le réseau. Ces réseaux, appelés réseaux ad hoc hybrides, conservent l'aspect multi-sauts des communications permettant l'utilisation des réseaux ad hoc classiques existants, tel que le protocole réactif DSR. DSR étant mal adapté à un réseau ad hoc hybride, nous l'avons étendu en ajoutant des mécanismes d'enregistrement et de découverte de points d'accès. Nous avons montré par simulation qu'un tel protocole étendu réduit l'overhead par rapport au DSR de base quand la longueur de la route, et donc le nombre de nœuds augmentent. Nous avons également montré qu'un réseau ad hoc hybride augmente la capacité d'un réseau de véhicules sur autoroute.

La principale problématique des réseaux ad hoc hybrides dans un contexte de réseau de véhicules sur autoroute est la grande vitesse des nœuds mobiles (véhicules) par rapport aux nœuds fixes (point d'accès). Nous avons étudié l'utilisation de convois de véhicules dans un tel réseau pour limiter ce problème. En effet, dans un convoi, plusieurs véhicules peuvent être connectés simultanément à des points d'accès différents, permettant un handover sans coupure. Nous avons donc présenté un algorithme de formation de clusters adapté au réseau de véhicules, appelé formation de convois. Nous avons adapté ce type d'algorithme au contexte véhiculaire, la tête de cluster étant la tête de convoi. La position relative de ses voisins pouvant être connue grâce à un système de géolocalisation, chaque véhicule peut déterminer la tête de convoi via leur position géographique.

Les simulations réalisées à l'aide d'un simulateur de réseaux de véhicules dans un contexte autoroutier ont montré la bonne stabilité des convois et des distributions de la taille et de la longueur de convois satisfaisantes. Ces résultats nous permettent de penser qu'il est possible d'exploiter les propriétés d'un convoi afin de gérer plus finement la mobilité.

La comparaison en termes d'overhead et de débit entre un réseau ad hoc hybride de base et un réseau ad hoc hybride à formation de convois reste à réaliser pour démontrer le réel intérêt de l'utilisation d'un convoi dans un contexte de réseau de véhicules sur autoroute. Cette comparaison pourrait être réalisée soit à l'aide d'une simulation, ou bien à l'aide du modèle que nous avons présenté au chapitre 4. Il serait également intéressant d'étendre les services intra-convoi à des services accessibles par tous nœuds du réseau, tout en exploitant la hiérarchisation à deux niveaux : convoi et infrastructure fixe.

Enfin, des expérimentations sur un réseau réel de véhicules nous permettraient de mieux appréhender la robustesse de notre protocole dans un environnement où les phénomènes physiques (effet Doppler, multi-chemin, interférences, etc.) entraînent des pertes de paquets. Il serait également intéressant de connaître les différences de mobilité entre notre modèle et un réseau réel de véhicules sur autoroute.

Références bibliographiques

[1] *Communications, air-interface, long and medium creaks (CALM)*, http://www.calm.hu.

[2] *Galileo on european commission web site*, http://ec.europa.eu/transport/galileo.

[3] *J-Sim Wireless Extension*, http://sites.google.com/site/jsimofficial/wireless-package.

[4] *Jist / Swans : Java in simulation time / scalable wireless ad hoc network simulator*, http://jist.ece.cornell.edu.

[5] *Mobile ad-hoc networks (manet)*, http://www.ietf.org/dyn/wg/charter/manet-charter.html.

[6] *Mobility for ip : Performance, signaling and handoff optimization (mipshop)*, http://www.ietf.org/dyn/wg/charter/mipshop-charter.html.

[7] *Mobility for ipv4 (mip4)*, http://www.ietf.org/dyn/wg/charter/mip4-charter.html.

[8] *The ns-3 network simulator*, http://www.nsnam.org.

[9] *NS2 : the network simulator*, http://www.isi.edu/nsnam/ns.

[10] *OMNeT++ community site*, http://www.omnetpp.org.

[11] *SUMO - simulation of urban mobility*, http://sumo.sourceforge.net/.

[12] *SWANS++ : Extensions to the scalable wireless ad-hoc network simulator*, http://www.aqualab.cs.northwestern.edu/projects/swans++/.

[13] *Status of project IEEE 802.11p*, 2009, http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm.

[14] Kazi I. Ahmed, *Modeling drivers'acceleration and lane changing behavior*, Doctor of science in transportation systems, Massachusetts Institute of Technology, Boston (MA), 1999.

[15] R. Ananthapadmanabha, BS Manoj, and C.S.R. Murthy, *Multi-hop cellular networks : the architecture and routing protocols*, Proc. IEEE PIMRC 2001 (2001), 78–83.

[16] G. Angione, P. Bellavista, A. Corradi, and E. Magistretti, *A k-hop clustering protocol for dense mobile ad-hoc networks*, 26th IEEE International Conference on Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006, 2006, p. 10.

[17] P. Bahl and V. Padmanabhan, *RADAR : An in-building RF-based user location and tracking system*, IEEE infocom, vol. 2, INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2000, pp. 775–784.

[18] R. Baldessari, T. Ernst, A. Festag, and M. Lenardi, *Automotive Industry Requirements for NEMO Route Optimization*, Internet-Draft draft-ietf-mext-nemo-ro-automotive-req-02, Internet Engineering Task Force, January 2009, Work in progress.

[19] R. Baldessari, A. Festag, A. Matos, J. Santos, and R. Aguiar, *Flexible connectivity management in vehicular communication networks*, Proc. of 3 rd International Workshop on Intelligent Transportation, 2006, pp. 211–216.

[20] S. Barghi, A. Benslimane, and C. Assi, *Connecting vehicular networks to the internet : A life time-based routing protocol*, 10th IEEE WoWMoM (Kos, Greece), 2009.

[21] R. Barr, *An efficient, unifying approach to simulation using virtual machines*, Ph.D. thesis, Citeseer, 2004.

[22] S. Basagni, *Distributed clustering for ad hoc networks*, Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth InternationalSymposium on (Perth/Fremantle, WA, Australia), 1999.

[23] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward, *A distance routing effect algorithm for mobility (dream)*, MobiCom '98 : Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking (New York, NY, USA), ACM, 1998, pp. 76–84.

[24] M. Bechler, W.J. Franz, and L. Wolf, *Mobile internet access in FleetNet*, 13. Fachtagung Kommunikation in verteilten Systemen, Leipzig, Germany (2003).

[25] M. Bechler, L. Wolf, O. Storz, and WJ Franz, *Efficient discovery of Internet gateways in future vehicular communication systems*, Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual **2** (2003).

[26] R. Bellman, *On a routing problem*, Quart. Appl. Math. **16** (1958), 87.

[27] M. Benzaid, P. Minet, and K. Al Agha, *A framework for integrating Mobile-IP and OLSR ad-hoc networking for future wireless mobile systems*, 1st Mediterranean Ad-Hoc Networks Wksp.(Med-Hoc-Net).

[28] M. Benzaid, P. Minet, K. Al Agha, C. Adjih, and G. Allard, *Integration of Mobile-IP and OLSR for a Universal Mobility*, Wireless Networks **10** (2004), no. 4, 377–388.

[29] J. Blum, A. Eskandarian, and L. Hoffman, *Mobility management in IVC networks*, Intelligent Vehicles Symposium, 2003. Proceedings. IEEE, June 2003, pp. 150–155.

[30] J. Broch, D.A. Maltz, and D.B. Johnson, *Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks*, Workshop on Mobile Computing at I-SPAN (1999).

[31] R. Buchmann, *Modélisation et Simulation Rapide au niveau cycle pour l'Exploration Architecturale de Systèmes Intégrés sur puce*, Ph.D. thesis, 2006.

[32] D. Camara, N. Frangiadakis, F. Filali, A. Loureiro, and N. Roussopoulos, *Virtual access points for disaster scenarios*, Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE (Budapest,), April 2009, pp. 1–6.

[33] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, and Z. Turanyi, *Design, implementation, and evaluation of Cellular IP*, 2000.

[34] A. Campbell and J. Gomez-Castellanos, *IP micro-mobility protocols*, ACM SIGMOBILE Mobile Computer and Communication Review (MC2R), 2001.

[35] AT Campbell, J. Gomez, and AG Valko, *An overview of cellular IP*, Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE (1999), 606–610.

[36] I. Chakeres and C. Perkins, *Dynamic MANET On-demand (DYMO) Routing*, Work in Progress, March 2009.

[37] C. Cho and D. Lee, *Survey of service discovery architectures for mobile ad hoc networks*, Term Paper, Computer and Information Science and Engineering Department, University of Florida. Gainesville (2005).

[38] T. Clausen and C. Dearlove, *Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)*, RFC 5497 (Proposed Standard), March 2009.

[39] T. Clausen, C. Dearlove, and B. Adamson, *Jitter Considerations in Mobile Ad Hoc Networks (MANETs)*, RFC 5148 (Informational), February 2008.

[40] T. Clausen, C. Dearlove, J. Dean, and C. Adjih, *Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format*, RFC 5444 (Proposed Standard), February 2009.

[41] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626 (Experimental), October 2003.

[42] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, *The optimized link state routing protocol, evaluation through experiments and simulation*, IEEE Symposium on Wireless Personal Mobile Communication 2001, Septembre 2001.

[43] SM Das, H. Pucha, and YC Hu, *Performance comparison of scalable location services for geographic ad hoc routing*, INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. **2** (2005).

[44] S.R. Das, R. Castañeda, and J. Yan, *Simulation-based performance evaluation of routing protocols for mobile ad hoc networks*, Mobile Networks and Applications **5** (2000), no. 3, 179–189.

[45] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*, RFC 3963 (Proposed Standard), January 2005.

[46] B. Ducourthial, Y. Khaled, and M. Shawky, *Conditional transmissions : Performance study of a new communication strategy in vanet*, Vehicular Technology, IEEE Transactions on **56** (2007), no. 6, 3348 –3357.

[47] P.J. Erard and P. Déguénon, *Simulation par événements discrets*, PPUR presses polytechniques, 1996.

[48] T. Ernst, K. Uehara, and K. Mitsuya, *Network mobility from the InternetCAR perspective*, Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on, 2003, pp. 19–25.

[49] Peng Fan, Abolfazl (Kouros) Mohammadian, Peter C. Nelson, James Haran, and John Dillenburg, *A novel direction based clustering algorithm in vehicular ad hoc networks*, 86th Annual Transportation Research Board Meeting, January 2007.

[50] Peng Fan, Peter C. Nelson, James Haran, and John Dillenburg, *Cluster-based framework in vehicular ad-hoc networks*, (2005), 32–42.

[51] R. Flickenger, *Building Wireless Community Community Networks*, O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2003.

[52] E. Fogelstroem, A. Jonsson, and C. Perkins, *Mobile IPv4 Regional Registration*, RFC 4857 (Experimental), June 2007.

[53] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, *Contention-based forwarding for mobile ad hoc networks*, Ad Hoc Networks **1** (2003), no. 4, 351–369.

[54] Holger Füßler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, and Dieter Vollmer, *A Comparison of Routing Strategies for Vehicular Ad Hoc Networks*, Tech. Report TR-02-003, Department of Computer Science, University of Mannheim, July 2002.

[55] O. Gehring and H. Fritz, *Practical results of a longitudinal control concept for truckplatooning with vehicle to vehicle communication*, Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on (Boston, MA, USA), November 1997, pp. 117–122.

[56] Mario Gerla, *From battlefields to urban grids : new research challenges in ad hoc wireless networks*, Pervasive Mob. Comput. **1** (2005), no. 1, 77–93.

[57] M. Ghassemian, P. Hofmann, C. Prehofer, V. Friderikos, and H. Aghvami, *Performance analysis of Internet gateway discovery protocols in ad hoc networks*, Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE **1** (2004).

[58] S. Giordano and M. Hamdi, *Mobility management : The virtual home region*, EPFL, Lausanne, Switzerland, Tech. Rep. SSC/1999/037 (1999).

[59] W. Gleißner and H. Zeitler, *The reuleaux triangle and its center of mass*, Resultate der Mathematik **37** (2000), no. 3-4, 335–344.

[60] P. Gupta and P. R. Kumar, *The capacity of wireless networks*, IEEE Trans. on Information Theory **46** (March 2000), no. 2, 388–400.

[61] Zygmunt J. Haas and Ben Liang, *Ad hoc mobility management with uniform quorum systems*, IEEE/ACM Trans. Netw. **7** (1999), no. 2, 228–240.

[62] Zygmunt J. Haas and Marc R. Pearlman, *ZRP : a hybrid framework for routing in Ad Hoc networks*, Ad hoc networking (Boston, MA, USA), 2001, pp. 221–253.

[63] J. K. Hedrick, M. Tomizuka, and P. Varaiya, *Control issues in automated highway systems*, IEEE Control Systems Magazine **14** (1994), no. 6, 21–32.

[64] Luc Hogie, Pascal Bouvry, and Frédéric Guinand, *An overview of manets simulation*, Electronic Notes in Theoretical Computer Science **150** (2006), no. 1, 81 – 101, Proceedings of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005).

[65] Robert Hsieh and Aruna Seneviratne, *A comparison of mechanisms for improving mobile ip handoff latency for end-to-end tcp*, MobiCom '03 : Proceedings of the 9th annual international conference on Mobile computing and networking (New York, NY, USA), ACM, 2003, pp. 29–41.

[66] Moez Jerbi, Rabah Meraihi, Sidi-Mohammed Senouci, and Yacine Ghamri-Doudane, *Gytar : improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments*, VANET '06 (New York, NY, USA), 2006, pp. 88–89.

[67] D. Johnson, Y. Hu, and D. Maltz, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*, RFC 4728 (Experimental), February 2007.

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

[69] David B. Johnson, David A. Maltz, and Josh Broch, *Dynamic source routing in ad hoc network*, Mobile Computing, Kluwer Academic Publishers, 1996, pp. 153–181.

[70] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson, and G.Q. Maguire Jr, *MIPMANET : mobile IP for mobile ad hoc networks*, Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing (2000), 75–85.

[71] Brad Karp and H. T. Kung, *GPSR : Greedy perimeter stateless routing for wireless networks*, The sixth Annual ACM/IEEE International Conference On Mobile and Networking (MobiCom 2000), Aout 2000.

[72] M. Kawai, M. Nozaki, and K. Gyoda, *A wireless ad-hoc community network with reconfigurable topology architecture*, GLOBECOM 98, vol. 4, 1998.

[73] W.T. Kim, *DynaMoNET : Dynamic Multi-homed IPv6 Mobile Networks with Multiple Mobile Routers*, Ubiquitous Computing Systems (2006), 398–413.

[74] R. Koodli, *Mobile IPv6 Fast Handovers*, RFC 5268 (Proposed Standard), June 2008.

[75] R. Koodli and C. Perkins, *Mobile IPv4 Fast Handovers*, RFC 4988 (Experimental), October 2007.

[76] Ulas C. Kozat and Leandros Tassiulas, *Service discovery in mobile ad hoc networks : an overall perspective on architectural choices and network layer support issues*, Ad Hoc Networks **2** (2004), no. 1, 23 – 44.

[77] Dinesh Kumar, Arzad A. Kherani, and Eitan Altman, *Route Lifetime based Interactive Routing in Intervehicle Mobile Ad Hoc Networks*, Research Report RR-5691, INRIA, 2006.

[78] J. LeBrun, C.N. Chuah, D. Ghosal, and M. Zhang, *Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks*, IEEE 61st Vehicular Technology Conference, vol. 4, 2005.

[79] I. Leontiadis and C. Mascolo, *GeOpps : Geographical opportunistic routing for vehicular networks*, WoWMoM, 2007, pp. 1–6.

[80] F. Li and Y. Wang, *Routing in vehicular ad hoc networks : A survey*, Vehicular Technology Magazine, IEEE **2** (2007), no. 2, 12–22.

[81] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, *A scalable location service for geographic ad hoc routing*, MobiCom '00 : Proceedings of the 6th annual international conference on Mobile computing and networking (New York, NY, USA), ACM, 2000, pp. 120–130.

[82] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, *A routing strategy for vehicular ad hoc networks in city environments*, IEEE Intelligent Vehicles Symposium, 2003. Proceedings, 2003, pp. 156–161.

[83] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, *Geographic routing in city scenarios*, ACM SIGMOBILE Mobile Computing and Communications Review **9** (2005), no. 1, 69–72.

[84] Muriel Mabiala, *Propriétés structurelles des réseaux ad hoc de véhicules*, Ph.D. thesis, Paris Sud XI, 2007.

[85] Muriel Mabiala, Anthony Busson, and Véronique Vèque, *Inside vanet : Hybrid network dimensioning and routing protocol comparison*, IEEE VTC Spring, 2007, pp. 227–232.

[86] Muriel Mabiala, Anthony Busson, and Véronique Vèque, *Analyse du trafic et du routage dans un réseau ad hoc de véhicules*, Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006, 2006.

[87] K.T. Mai and H. Choo, *Connectivity-based clustering scheme for mobile ad hoc networks*, IEEE International Conference on Research, Innovation and Vision for the Future, 2008. RIVF 2008, 2008, pp. 191–197.

[88] G. Malkin, *RIP Version 2 Carrying Additional Information*, RFC 1388 (Proposed Standard), January 1993, Obsoleted by RFC 1723.

[89] Marc Torrent-Moreno and Felix Schmidt-Eisenlohr and Holger Füßler and Hannes Hartenstein", *Packet forwarding in vanets, the complete set of results*, Tech. report, Institute of Telematics, University of Karlsruhe, Germany ; Computer Science IV, University of Mannheim, Germany, 2006.

[90] T. Narten, E. Nordmark, and W. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, RFC 2461 (Draft Standard), December 1998, Obsoleted by RFC 4861, updated by RFC 4311.

[91] V. Naumov and TR Gross, *Connectivity-aware routing (CAR) in vehicular ad-hoc networks*, IEEE INFOCOM, 2007, pp. 1919–1927.

[92] N. Nikaein, H. Labiod, and C. Bonnet, *DDR-distributed dynamic routing algorithm for mobile ad hocnetworks*, Mobile and Ad Hoc Networking and Computing, MobiHOC. (Boston, MA, USA), 2000, pp. 19–27.

[93] X. Niu, Z. Tao, G. Wu, C. Huang, and L. Cui, *Hybrid Cluster Routing : An Efficient Routing Protocol for Mobile Ad Hoc Networks*, IEEE International Conference on Communications, ICC'06, vol. 8, 2006.

[94] Fabian Garcia Nocetti, Julio Solano Gonzalez, and Ivan Stojmenovic, *Connectivity based k-hop clustering in wireless networks*, Telecommunication Systems **22** (2003), no. 1–4, 205–220.

[95] E. Nordström, P. Gunningberg, and C. Tschudin, *Comparison of forwarding strategies in internet connected MANETs*, ACM SIGMOBILE Mobile Computing and Communications Review **8** (2004), no. 4, 72–76.

[96] Tomoyuki Ohta, Shinji Inoue, and Yoshiaki Kakuda, *An adaptive multihop clustering scheme for highly mobile ad hoc networks*, Proceedings of the The Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03) (Washington, DC, USA), 2003, p. 293.

[97] Tomoyuki Ohta, Naoyoshi Murakami, and Yoshiaki Kakuda, *Performance evaluation of autonomous clustering for hierarchical routing protocols in mobile ad hoc networks*, ICDCSW '07 (Washington, DC, USA), 2007, p. 56.

[98] Eun Kyoung Paik, Ho sik Cho, Thierry Ernst, and Yanghee Choi, *Load sharing and session preservation with multiple mobile routers for large scale network mobility*, Advanced Information Networking and Applications, International Conference on **1** (2004), 393.

[99] E. Perera, V. Sivaraman, and A. Seneviratne, *Survey on network mobility support*, ACM SIGMOBILE Mobile Computing and Communications Review **8** (2004), no. 2, 7–19.

[100] C. Perkins, *IP Mobility Support for IPv4*, RFC 3344 (Proposed Standard), August 2002, Updated by RFC 4721.

[101] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, RFC 3561 (Experimental), July 2003.

[102] C. E. Perkins and E. M. Royer, *Ad Hoc On-Demand Distance Vector Routing (AODV)*, IEEE Workshop on Mobile Computing Systems and Applications 1999, Fevrier 1999, pp. 90–100.

[103] Charles E. Perkins and Pravin Bhagwat, *Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers*, SIGCOMM Comput. Commun. Rev. **24** (1994), no. 4, 234–244.

[104] Ramachandran Ramjee, Thomas F. La Porta, S. Thuel, Kannan Varadhan, and S. Y. Wang, *HAWAII : A domain-based approach for supporting mobility in wide-area wireless networks*, ICNP, 1999, pp. 283–292.

[105] P. Ratanchandani and R. Kravets, *A hybrid approach to Internet connectivity for mobile ad hoc networks*, IEEE WCNC **3** (2003).

[106] S. Raúl Aquino and B. Arthur Edwards, *A Reactive Location Routing Algorithm with Cluster-Based Flooding for Inter-Vehicle Communication*, Computación y Sistemas **9** (2006), no. 4, 297–313.

[107] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz, *CarTALK 2000 : safe and comfortable driving based upon inter-vehicle-communication*, Intelligent Vehicle Symposium IEEE, vol. 2, June 2002, pp. 545–550.

[108] G.F. Riley, *Simulation of large scale networks II : large-scale network simulations with GTNetS*, Proceedings of the 35th conference on Winter simulation : driving innovation, Winter Simulation Conference, 2003, pp. 676–684.

[109] Francisco J. Ros and Pedro M. Ruiz, *Cluster-based olsr extensions to reduce control overhead in mobile ad hoc networks*, IWCMC '07 : Proceedings of the 2007 international conference on Wireless communications and mobile computing (New York, NY, USA), ACM, 2007, pp. 202–207.

[110] Ryuji Wakikawa and Charles E. Perkins and Anders Nilsson and Antti J. Tuominen, *Global connectivity for IPv6 mobile ad hoc networks*, Work in Progress, March 2006.

[111] C. A. Santivanez, B. Mcdonald, I. Stavrakakis, and R. Ramanathan, *On the scalability of ad hoc routing protocols*, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 3, 2002, pp. 1688–1697 vol.3.

[112] César A. Santiváñez, Bruce Mcdonald, Ioannis Stavrakakis, and Ram Ramanathan, *On the scalability of ad hoc routing protocols*, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 3, 2002, pp. 1688–1697 vol.3.

[113] B.C. Seet, G. Liu, B.S. Lee, C.H. Foh, K.J. Wong, K.K. Lee, et al., *A-STAR : A mobile ad hoc routing strategy for metropolis vehicular communications*, Lecture Notes in Computer Science (2004), 989–999.

[114] V. Sekar, B. S. Manoj, and C. Siva Ram Murthy, *Routing for a single interface mcn architecture and pricing schemes for data traffic in multihop cellular networks*, IEEE ICC, May 2003.

[115] Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, and Honghai Zhang, *J-sim : A simulation environment for wireless sensor networks*, ANSS '05 : Proceedings of the 38th annual Symposium on Simulation (Washington, DC, USA), 2005, pp. 175–187.

[116] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*, RFC 5380 (Proposed Standard), October 2008.

[117] I. Stojmenovic, *Home agent based location update and destination search schemes in ad hoc wireless networks*, Advances in Information Science and Soft Computing (2002), 6–11.

[118] H. Takagi and L. Kleinrock, *Optimal transmission ranges for randomly distributed packet radio terminals*, IEEE Transactions **32** (1984), no. 3, 246–257.

[119] J. Tian, L. Han, K. Rothermel, and C. Cseh, *Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks*, IEEE ITSC, vol. 2, 2003, pp. 1546–1551.

[120] Joachim Tisal, *Le réseau gsm. l'évolution gprs : une étape vers umts*, troisième ed., Dunod, 1999.

[121] M. Torrent-Moreno, F. Schmidt-Eisenlohr, H. Füßler, and H. Hartenstein, *Packet Forwarding in VANETs, the Complete Set of Results*, Tech. report, Univ., Fak. für Informatik, Bibl., 2005.

[122] Ville Typpo, *Micro-mobility within wireless ad hoc networks : Towards hybrid wireless multihop networks*, Ph.D. thesis, Department of Electrical Engineering, University of Oulu, Finland, 2001.

[123] S.C.M. Woo and S. Singh, *Scalable routing protocol for ad hoc networks*, Wireless Networks **7** (2001), no. 5, 513–529.

[124] Bin Xie, A. Kumar, D. Cavalcanti, D. P. Agrawal, and S. Srinivasan, *Mobility and routing management for heterogeneous multi-hop wireless networks*, IEEE Mobile Adhoc and Sensor Systems Conference., November 2005.

[125] Bin Xie, Anup Kumar, Dharma P. Agrawal, and S. Srinivasan, *Secured macro/micro-mobility protocol for multi-hop cellular ip*, Pervasive and Mobile Computing **2** (2006), no. 2, 111–136.

[126] Yuan Xue, Baochun Li, and Klara Nahrstedt, *A scalable location management scheme in mobile ad-hoc networks*, LCN '01 : Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (Washington, DC, USA), 2001, p. 102.

[127] Q. Yang, *A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems*, Ph.D. thesis, Carnegie Mellon University, 1997.

[128] X. Zeng, R. Bagrodia, and M. Gerla, *GloMoSim : a library for parallel simulation of large-scale wireless networks*, ACM SIGSIM Simulation Digest **28** (1998), no. 1, 154–161.

[129] J. Zhao and G. Cao, *VADD : Vehicle-assisted data delivery in vehicular ad hoc networks*, IEEE Transactions on Vehicular Technology **57** (2008), no. 3, 1910–1922.