ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE
CONCENTRATION RÉSEAUX DE TÉLÉCOMMUNICATIONS
M.Ing.

PAR
ABDELLAOUI, Rachid

SU-OLSR UNE NOUVELLE SOLUTION POUR LA SÉCURITÉ DU PROTOCOLE OLSR.

MONTRÉAL, LE 05 MAI 2009

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE

M. Jean-Marc Robert, directeur de mémoire
Département génie logiciel et T.I à l'École de technologie supérieure

M. Michel Kadoch, président du jury
Département génie électrique à l'École de technologie supérieure

Mme Nadjia Kara, membre du jury
Département génie logiciel et T.I à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 21 AVRIL

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier en premier mon directeur de mémoire, le professeur Jean-Marc Robert, pour l'aide qu'il m'a apportée, ses précieux conseils et ainsi pour sa sympathie durant tout le déroulement de ce travail. Je remercie aussi les membres du comité d'avoir accepté d'y participer.

Je remercie également ma famille ainsi que mes amis pour leur soutien durant toute ma maîtrise et la période de rédaction de ce mémoire.

Ce travail de recherche a été partiellement financé par le CRSNG subvention à la découverte ainsi qu'une subvention de démarrage de l'ETS.

SU-OLSR, UNE NOUVELLE SOLUTION POUR LA SÉCURITÉ DU PROTOCOLE OLSR

ABDELLAOUI, Rachid

RÉSUMÉ

Un réseau Ad-Hoc mobile (*Mobile Ad-Hoc NETwork : MANET*) est une collection de nœuds sans fil formant un réseau dynamique sans infrastructure préexistante ou une architecture centralisée. Chaque nœud dans ce type de réseau fonctionne comme un routeur et utilise un protocole de routage pour acheminer les messages.

OLSR (*Optimized Link State Routing*) est un protocole de routage proactif pour les réseaux Ad-Hoc présenté par le groupe MANET de l'IETF. Ce protocole utilise des nœuds appelés relais multipoints (*MultiPoint Relay : MPR*) pour optimiser la diffusion dans le réseau. Chaque MPR doit diffuser les informations sur la topologie et réacheminer les messages aux nœuds destinations. Si l'un de ces MPR est malicieux, il présentera un danger pour la sécurité de tout le réseau.

Dans ce travail de recherche, nous avons présenté un nouveau protocole dérivé du protocole OLSR et qui utilise le concept de confiance entre les nœuds. L'objectif est de faire face aux attaques par mystification de lien où un nœud malicieux cherche à forcer ses voisins à le choisir comme MPR. Le nouveau protocole SU-OLSR (*SUspicious OLSR*) empêche tout nœud qui présente des comportements suspects d'être choisi comme MPR.

Grâce à diverses simulations et l'implémentation de SU-OLSR sous *ns-2*, nous avons montré que SU-OLSR fournit les mêmes performances que le protocole OLSR classique malgré qu'il soit plus sélectif lors du choix des MPR.

Mots-clés : Sécurité, OLSR, Réseaux Ad-Hoc, Protocoles de routage, Vulnérabilité

SU-OLSR: A NEW SOLUTION TO THWART ATTACKS AGAINST THE OLSR PROTOCOL

ABDELLAOUI, Rachid

ABSTRACT

Mobile Ad-Hoc network (MANET) is a collection of wireless nodes forming a dynamic network without any centralized or pre-existing infrastructure. Each node in such a network has to act as a router, using a routing protocol to achieve this task.

The OLSR (*Optimized Link State Routing*) protocol is a proactive routing protocol presented by the IETF MANET working group for ad-hoc networks. One of the key aspects of OLSR protocol is the use of special nodes called Multipoint Relay (MPR) nodes. These nodes have to broadcast the topology information through the network and to forward packets towards their destinations. If one of those nodes is malicious, it would represent a major threat against the security of the overall network.

A new approach to Ad-Hoc routing protocol using the concept of trustworthiness for the neighbour nodes is presented in this thesis. We propose a new protocol to prevent a malicious node to force its neighbours to select it as a MPR node. With our SU-OLSR protocol (*Suspicious OLSR*), a node should not choose a neighbour as a relay node if it behaves suspiciously and demonstrates strong characteristics which would influence the MPR selection algorithm.

We have ported the SU-OLSR protocol to the *ns-2* simulator. This implementation and simulation results are discussed along this thesis. We have demonstrated that our solution prevents the link spoofing attack against the OLSR protocol. The performances of the SU-OLSR related to the path lengths, data packet average end-to-end delivery time and data Packet Delivery Ratio (PDR) are quite comparable to the classical OLSR, if the network density assures some redundancy.

Keywords: Security, OLSR, Ad-Hoc Network, Routing protocols, Vulnerability

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ARP	Address Resolution Protocol
AODV	Ad-Hoc On demand Distance Vector
BASH	Bourne-Again Shell
CBR	Constant Bit Rat
CMU	Carnegie Mellon University
CSMA/CA	Carrier Sensing Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sensing Multiple Access with Collision Avoidance
DARPA	Defense Advanced Research Projects Agency
DSDV	Destination Sequenced Distance Vector routing protocol
DSR	Dynamic Source Routing
DSSS	Direct-Sequence Spread-Spectrum
HNA	Host and Network Association
HOLSR	Hierarchical Optimized Link State Routing Protocol
IEEE	The Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
INRIA	Institut National de Recherche en Informatique et Automatique
LAN	Local Area Network
MAC	Medium Access Control
MANET	Mobile Ad-Hoc NETwork
MID	Multiple Interface Declaration
MPR	Multi-Point Relays
NAM	Network Animator

NS-2	Network Simulator 2
OLSR	Optimized Link State Routing Protocol
PDA	Personal Digital Assistant
PDR	Packet Delivery Ratio
PHY	Physical Layer
PKI	Public Key Infrastructure
QOLSR	Quality of Service extension introduced to the OLSR protocol
QoS	Quality of Service
RAM	Random Access Memory
RERR	Route ERRor
RREP	Route REPly
RREQ	Route REQuest
RTS	Request To Sent
SU-OLSR	SUspicious Optimized Link State Routing Protocol
TC	Topology Control
TCL	Tool Command Language
TTL	Time To Live
UDP	User Datagram Protocol
VINT	Virtual InterNetwork Testbed
WRP	Wireless Routing Protocol

INTRODUCTION

1 - Point de départ

La prolifération des ordinateurs portables et des appareils de télécommunication sans fil (téléphone cellulaire, ordinateur et PDA…) change d'une manière révolutionnaire nos architectures de télécommunication. En effet, nous passons actuellement d'un monde d'ordinateurs personnels à un monde où l'informatique est omniprésente (*Ubiquitous Computing*). Chaque personne pourra partager des informations et accéder à plusieurs ressources sur différentes plateformes électroniques quand et où elle veut.

Les récents développements dans le domaine des réseaux sans fil vont accélérer sans doute la migration vers ce type de réseau. On peut noter les deux technologies concurrentes : la technologie 4G LTE (*Long-term Evolution*) avec un débit théorique de 173 Mbits/s et la technologie WiMax mobile avec un débit de 70 Mbits/s (3GPP, 2008). L'informatique omniprésente a besoin donc de solutions d'interconnexion entre ces appareils mobiles afin de faciliter la mise en place d'infrastructures flexibles. Le réseau Ad-Hoc est un excellent candidat pour répondre à ces besoins.

Les réseaux Ad-Hoc sont une collection d'appareils mobiles qui peuvent dynamiquement échanger des informations entre eux sans utiliser une infrastructure réseau préexistante et fixe ou une administration centralisée. Chaque appareil de ce type de réseau communique directement avec les autres appareils qui se trouvent dans son rayon de communication (portée radio). La communication avec les appareils distants se fait par le routage des communications à travers des appareils intermédiaires. Au début, les réseaux Ad-Hoc ont été développés pour les milieux militaires. Mais, grâce à leurs natures d'auto-organisation et de facilité de déploiement, ces réseaux ont eu d'autres vocations et d'applications civiles comme dans les situations d'urgence et les opérations de secours lors d'un désastre (80% de la ville de La Nouvelle-Orléans est inondée en 2005, saturation des infrastructures de télécommunication à cause d'un volume très élevé d' appels lors des attentats à New York en

2001) ou l'échange d'informations entre les PDA et les ordinateurs portables lors de réunion…

La communication avec les nœuds hors portée radio se fait par l'intermédiaire des autres nœuds qui acheminent les messages à destination. Ce processus se fait grâce au protocole de routage. À cet effet, plusieurs protocoles de routage ont été proposés et standardisés par le groupe MANET (*Mobile Ad-Hoc NETwork*).

La nature des réseaux Ad-Hoc, où les nœuds sont mobiles et peuvent se joindre ou quitter le réseau à tout moment, présente des grands défis pour la sécurité, la qualité de service et le routage. Or la sécurité est le point névralgique des réseaux Ad-Hoc. Il est donc primordial de préserver la confidentialité, l'intégrité, la non répudiation et la disponibilité dans ce type d'environnement. Pour se faire, il est important de prévenir les attaques en particulier contre les protocoles de routage. Or, ces derniers ne proposent pas de modèles de sécurité ou des mécanismes pour faire face à certains types d'attaques.

2 - Objectifs de recherche

Dans ce travail, nous nous intéressons au protocole OLSR (*Optimized Link State Routing Protocol*) et en particulier aux vulnérabilités et aux attaques contre ce protocole. En effet, OLSR est un protocole qui utilise une diffusion optimisée des messages grâce à des nœuds appelés MPR (Relais Multipoints). Les MPR permettent d'acheminer les messages de la source à la destination. Ils ont un rôle très important dans le réseau et tout comportement malveillant aurait un impact direct sur le bon fonctionnement du réseau. Or, un nœud malicieux qui a été sélectionné comme MPR aurait une position privilégiée dans le réseau et pourrait altérer, modifier ou rejeter tout message qui transite par lui. Afin d'avoir cette position privilégiée dans le réseau, un nœud malicieux peut utiliser les techniques d'attaques par mystification de lien. Ainsi, un nœud malicieux peut forcer ses voisins à le choisir comme MPR en déclarant qu'il couvre un nœud isolé (nœud inexistant dans le réseau ou distant).

Notre objective est, donc, de proposer une nouvelle solution pour lutter contre les attaques par mystification de lien dans le cas du protocole OLSR et ainsi empêcher tout nœud qui présente des comportements suspects et malveillants d'être sélectionné comme MPR.

3 - Organisation du mémoire

Ce mémoire est organisé de la manière suivante. Le chapitre 1 introduit les réseaux Ad-Hoc et les différentes familles de protocoles de routage en détaillant le fonctionnement de certains de ces protocoles. Pour sa part, le protocole OLSR sera complètement détaillé à la fin de ce chapitre suivi d'un sommaire comparatif de certains protocoles de routage représentant les trois familles : réactif, proactif et hybride.

Nous présenterons une revue de littérature dans le chapitre 2 afin de mettre la lumière sur les défis et les attaques sur les réseaux Ad-Hoc en général et le protocole OLSR en particulier. Ce chapitre classifiera aussi les différentes solutions proposées à ce sujet.

Le chapitre 3 quant à lui présente notre approche et le nouveau protocole SU-OLSR précédé d'une revue de littérature des différentes heuristiques de sélection des MPR pour le protocole OLSR. Ce chapitre donne aussi le modèle d'attaques relatives à SU-OLSR.

Par des simulations dans le chapitre 4, nous avons cherché à évaluer et à valider expérimentalement le nouveau protocole SU-OLSR dans différents environnements de mobilité et à le comparer aux performances du protocole OLSR. Dans une première étape, nous avons développé un programme en C pour évaluer SU-OLSR et OLSR dans un environnement statique. Dans une seconde phase, nous avons implémenté le protocole SU-OLSR sous *ns-2* afin de le comparer avec OLSR sous différents scénarios de mobilité.

En conclusion, le dernier chapitre propose un récapitulatif des principaux travaux réalisés et les résultats obtenus. Pour finir, nous donnons plusieurs perspectives à l'ensemble des contributions réalisées au cours de ce mémoire.

CHAPITRE 1

LES RÉSEAUX SANS FIL AD-HOC

1.1 Historique des réseaux Ad-Hoc

Au début, le développement des réseaux Ad-Hoc a été le résultat de la demande du milieu militaire pour le déploiement rapide d'infrastructures de télécommunication pouvant survivre aux pannes et aux attaques. Un réseau centralisé autour de stations de base n'est pas une bonne option dans ce milieu car elles doivent être déployées en premier lieu (presque impossible dans un terrain hostile) et le réseau est vulnérable dans le cas où une ou plusieurs de ces stations de base sont détruites.

Face à ces limites, en 1972 le département de la défense américaine, en particulier DARPA (*The Defense Advanced Research Projects Agency*), a sponsorisé le programme de recherche PRNet (Jubin et Tornow, 1987) (*Packets Radio Network*). Ce projet traitait en particulier la problématique de routage et l'accès au média dans un réseau de communication multi-sauts par onde radio.

En 1983, ce projet a évolué vers le programme SURAN (*SUrvivable RAdio Networks*) qui traitait en particulier la problématique de la sécurité, la gestion d'énergie et la capacité de traitement (Freebersyser et Leiner, 2001). Les objectifs étaient d'augmenter le nombre de nœuds supportés par PRNet dans une zone géographique étendue et réduire la consommation d'énergie en développant des nouveaux algorithmes de routage. Le LPR (*Low-cost Packet Radio*) a été le fruit de ces recherches en 1987 (Fifer et Bruno, 1987). La technologie LPR offrait la commutation de paquets et des améliorations aux niveaux de la sécurité et la gestion de la consommation de l'énergie par les nœuds.

Dès 1990, les ordinateurs portables ont été équipés de cartes sans fil et de ports infrarouge qui permettaient la communication directe et sans intermédiaire entre les ordinateurs portables. Ainsi, la technologie de PRNet était devenue accessible au grand public avec de

réelles applications civiles. L'IEEE (*Institute of Electrical and Electronics Engineers*) adoptait alors le terme 'réseaux *Ad-Hoc*' pour le standard IEEE 802.11 des réseaux locaux sans fil.

Avec l'importance que prenaient les réseaux sans-fil, en 1994, le DARPA sponsorisait les programmes GloMo (*Global Mobile Information Systems*) et NTDR (*Near-term Digital Radio*). Ces programmes avaient pour but le développement des réseaux Ad-Hoc sans fil qui offraient un environnement de communication multimédia n'importe quand et n'importe où (Leiner, Ruther et Sastry, 1996). Le NTDR est encore utilisé actuellement par l'armée américaine.

Un certain nombre de standards ont suivi ce développement des réseaux Ad-Hoc. C'est ainsi que le groupe de travail MANET (*Mobile Ad-Hoc Networks*) a été fondé au sein de l'IETF (*The Internet Engineering Task Force*). Ce groupe avait pour but d'essayer de standardiser les protocoles de routage dans les réseaux Ad-Hoc (Corson et Macker, 1999).

Plusieurs applications militaires et civiles ont suivi, par la suite, cette émergence des réseaux Ad-Hoc.

1.2 Applications des réseaux Ad-Hoc

Les réseaux Ad-Hoc ont été développés en premier lieu pour les communications dans le domaine militaire. Plusieurs applications de ces réseaux ont été mises en place surtout par l'armée américaine :

- En 1997, cette dernière a procédé au test de l'Internet Tactique (IT) (Freebersyser et Leiner, 2001) et la numérisation du champ de bataille en grandeur nature. La grande révolution de l'IT a été l'introduction du *Force XXI Battlefield Command Brigade and Below* (FBCB2) qui est une plateforme de communication basée sur MANET et qui permet aux soldats de traquer en temps réel les forces alliées et adverses dans le champ de bataille (FBCB2, 2008).

- Une autre application au sein de l'armée américaine est l'ELB ACTD (*Extending the Littoral Battle-space Advanced Concept Technology Demonstration*). Cette expérimentation avait pour but de démontrer la possibilité d'offrir une architecture de communication entre les navires militaires en mer et les soldats sur terre par l'intermédiaire d'un lien aérien.

- D'autres applications et projets de MANET sont actuellement en cours de développement au sein de DARPA. On pourra noter en particulier le projet ITMANET (*Information Theory for Mobile Ad-Hoc Networks*) qui a pour but le développement et l'exploitation de puissantes théories concernant MANET. Ce projet s'étendra sur cinq ans et prendra fin en 2011 (*ITMANET*, 2008).

D'un autre coté, avec l'émergence des technologies sans fil et la prolifération des terminaux équipés de carte 802.11, des applications civiles sont apparues. On pourra en distinguer plusieurs :

- Les réseaux mobiles sans fil Ad-Hoc pourraient être utilisés dans les cas d'urgence et les opérations de recherche et secours lors d'un désastre (feux, inondation, séisme). Ces opérations de secours peuvent parfois avoir lieu là où les infrastructures de télécommunication sont inexistantes, endommagées ou lors d'un besoin de déploiement rapide. C'est dans ce contexte que le NCS (*National Communications System*) a mis en place le projet *NS/EP Priority Telecommunications (Suraci, Ephrath et Wullert, 2007).* Cette infrastructure de télécommunication contient le service GETS (*Government Emergency Telecommunications Service*), le service TSP (*Telecommunications Service Priority*) et finalement le service WPS (*Wireless Priority Service*).

- Les réseaux mobiles sans fil Ad-Hoc pourraient être utilisés pour simplifier l'intercommunication et le partage des applications entre plusieurs équipements mobiles (PDA, ordinateur portable, téléphone cellulaire ou autres) dans une zone limitée qu'on appelle *Personal Area Network* (PAN). Ce principe a offert des nouvelles perspectives pour les jeux en réseau. En effet, Sony a offert le premier modèle de console de jeu

portable *PSP* permettant le jeu en réseau (jusqu'à 16 consoles) grâce à la technologie Ad-Hoc (Sony, 2008). Ainsi, avec l'émergence que prennent les appareils mobiles, PAN est très prometteur pour MANET.

- Les réseaux Mesh sans fil constituent une technologie qui permettra aux réseaux Ad-Hoc d'être au cœur des infrastructures de télécommunication de demain (Bruno, Conti et Gregori, 2005). En effet, la flexibilité et la facilité qu'offrent les réseaux Ad-Hoc permettent d'étendre les réseaux Mesh et offrir de nouveaux services. La Figure 1.1 donne un exemple de possible interaction entre les réseaux Mesh et les réseaux Ad-Hoc sans fil.

- D'autres applications sont actuellement en test comme le projet *Vehicular Ad-Hoc Networks* (VANET) (Yi et Moayeri, 2008). Ce type de réseau permet la communication entre les véhicules ainsi qu'avec les infrastructures de télécommunication. Ainsi, un conducteur sur la route pourrait avoir un accès fiable et rapide aux informations pratiques (par exemple, l'état du trafic, la gestion de distance de sécurité entre les voitures, etc). La norme 802.11p a été associée par IEEE à ce projet pour gérer les communications entre les entités à très forte mobilité jusqu'à 200km/h (Palmen et al., 2006).

- Les réseaux Ad-Hoc offrent une solution à faible coût pour étendre la couverture des points d'accès Internet et des réseaux sans fil (UMTS, WiMax). Par exemple, les passagers d'un train auraient l'accès à Internet grâce à des points d'accès installés dans chaque wagon. Ces points d'accès seraient reliés entre eux et connectés à Internet par la suite via une connexion Ad-Hoc à des passerelles dans les gares. Ce genre de solution est offert par une société anglaise *NOW Wireless* (*NowWireless*, 2008). Elle commercialise d'autres applications très intéressantes de la technologie Ad-Hoc : Feux de signalisation, services de pompier et police, transport intelligent, solutions pour le réseautage domestique, télématiques et solutions de réseaux de capteurs. Ces solutions sont déployées dans plusieurs villes (Coventry, Glasgow).

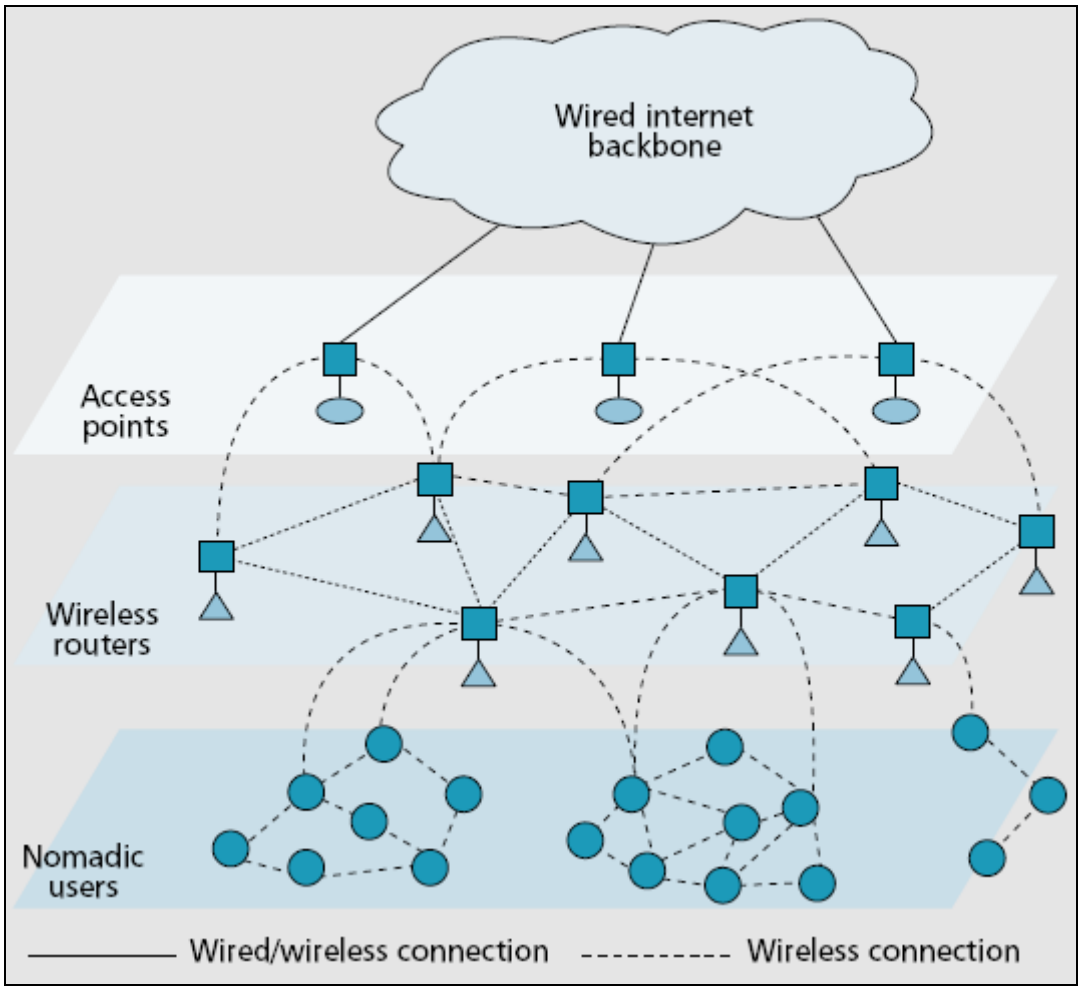

Figure 1.1 Extension des réseaux Mesh grâce aux réseaux Ad-Hoc.
Tirée de Bruno, Conti et Gregori (2005, p. 126)

- Les réseaux Ad-Hoc ont trouvé leur place dans les réseaux de capteurs (*Wireless Sensor Networks*). Ce type de réseau a plusieurs applications dans les domaines commercial et environnemental (réseaux de capteurs pour observer les tsunamis, les feux de forêts, la pollution, le climat, les activités séismiques, etc) (Xu, 2002). Un exemple dans le domaine commercial est la possibilité de recueillir des données fournies par des étiquettes intelligentes RFID par l'intermédiaire de communications Ad-Hoc (utiliser par les transporteurs pour garantir la traçabilité des marchandises).

Dans le domaine routier, le *Vehicular Sensor Network* (VSN) est une couche de plus dans les réseaux VANET (*Voir* Figure 1.2) et qui pourrait être utilisé par exemple pour la surveillance du trafic routier (Chenxi et al., 2008).

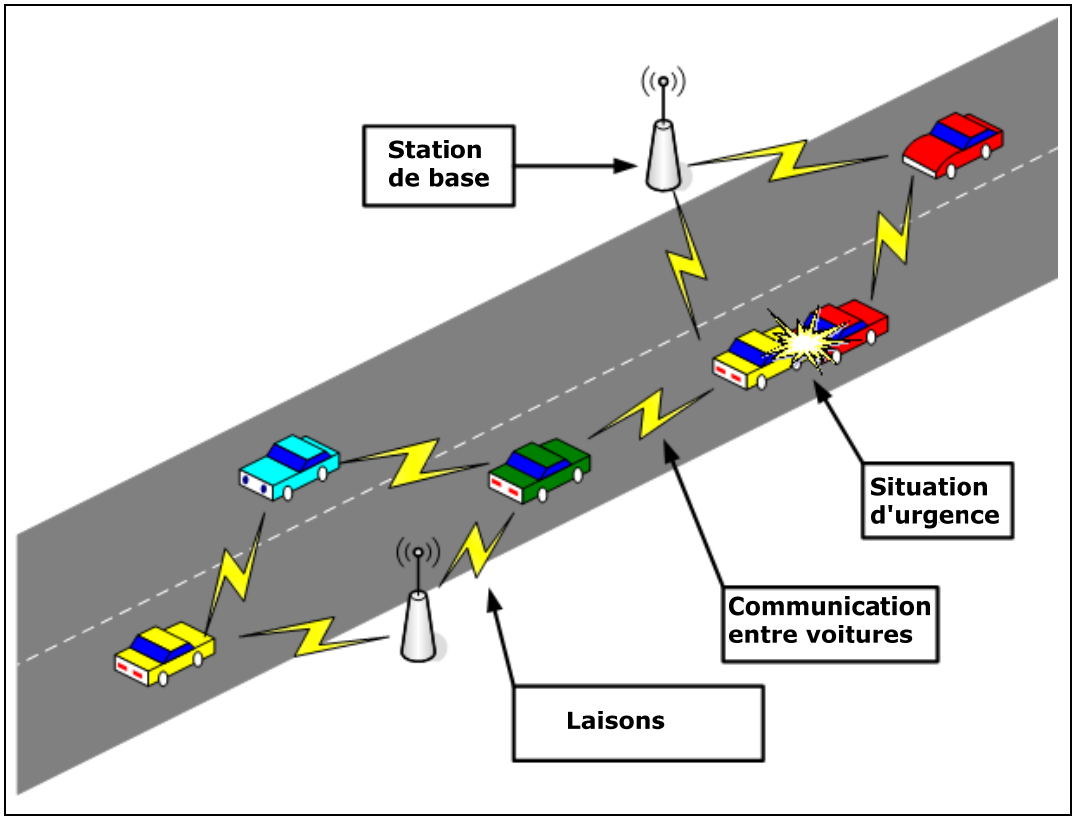

Figure 1.2 VANET et VSN.

1.3 Défis dans les réseaux Ad-Hoc

Les réseaux Ad-Hoc ont hérité des problèmes traditionnels des communications et des réseaux sans fil. S'ajoutent à cela, de nouveaux problèmes et défis liés spécifiquement à la nature des réseaux Ad-Hoc. Ainsi, les principaux problèmes de ces réseaux sont :

- **Interférences radio :** Avec la croissance de l'utilisation des appareils sans fil, des interférences radio peuvent avoir lieu si des transmissions se font sur une même

fréquence ou des fréquences proches l'une de l'autre. Ces interférences peuvent perturber les communications radio et nuire à leur qualité.

- **Erreurs de transmission :** Les problèmes et la particularité des transmissions radio engendrent plus d'erreurs de transmission comparativement aux transmissions sur câble.

- **Débit :** Le débit reste un grand challenge pour le développement de certains services dans MANET (IPTV, vidéo…).

- **Signal :** La puissance du signal engendre la portée radio d'un terminal sans fil. Or dans MANET, ce facteur est très important pour garantir une certaine densité et ainsi offrir de meilleures performances à ces réseaux.

- **Collisions :** Dans un environnement sans fil, il est impossible à un terminal de détecter les collisions lors de ses transmissions. En effet, pour détecter les collisions, le terminal doit faire la transmission et l'écoute en même temps, ce qui est impossible.

- **Énergie :** Dans un monde de mobilité, chaque nœud a la responsabilité d'acheminer les paquets qui arrivent d'un nœud et de les transmettre vers un nœud voisin. Or l'acheminement des paquets vers d'autres nœuds consomme de manière significative l'énergie d'un terminal. D'un autre côté, la batterie de chaque terminal mobile a des ressources limitées et risque de causer des problèmes si un terminal ne dispose pas d'assez d'énergie pour garantir le routage des paquets vers les autres nœuds.

- **Routage :** Chaque nœud dans un réseau Ad-Hoc agit comme un terminal ou un routeur. Ainsi, le développement des protocoles Ad-Hoc doit prendre en compte plusieurs facteurs dont la mobilité, changement brusque de la topologie, maintenir dynamiquement les routes et la gestion de la consommation de l'énergie lors du routage des paquets.

- **Découverte des services :** Les réseaux Ad-Hoc sont des réseaux dynamiques. Chaque nœud peut se joindre ou quitter le réseau à tout moment. Il est important d'offrir à chaque nœud qui se joint à un réseau, un mécanisme rapide et efficace pour découvrir les services offerts par les nœuds existants.

- **Sécurité :** Les réseaux Ad-Hoc soulèvent de nombreux problèmes de sécurité. Ces problèmes sont dus essentiellement aux protocoles de routage, l'environnement sans fil et à la nature de ces réseaux.

- **Qualité de Service (QoS) :** La mobilité des nœuds dans les réseaux Ad-Hoc rend très complexe la tâche d'offrir une bonne qualité de service.

- **Mobilité :** La mobilité a un impact très important sur les protocoles de routage, la topologie et les performances du réseau, les services et la QoS.

1.4 Protocoles de routage pour les réseaux Ad-Hoc

Le routage dans les réseaux Ad-Hoc présente des défis plus complexes en comparaison avec le routage dans les réseaux filaires traditionnels. En effet, une stratégie intelligente de routage est nécessaire pour supporter la nature et les paramètres du réseau (la mobilité, le nombre de nœuds, la densité du trafic, la qualité du service et la superficie du réseau).

Dans ce contexte, le groupe de travail MANET a été fondé au sein de l'IETF pour définir les spécifications des protocoles de routage pour les réseaux sans fil Ad-Hoc. Ce groupe a défini trois types de protocoles : les protocoles proactifs, les protocoles réactifs et les protocoles hybrides.

1.4.1 Les types de protocoles

Protocoles de routage proactifs

Les protocoles proactifs entretiennent en permanence les routes vers chaque nœud du réseau et maintiennent à jour les informations et les tables de routage. Les routes sont donc établies à l'avance et disponibles immédiatement lorsqu'elles sont sollicitées. Ces protocoles reposent sur les principes du routage basé sur l'état des liens (*Link-State*) (McQuillan, Richer et Rosen, 1979) ou basé sur les vecteurs de distance (*Distance Vector*) déjà utilisés dans les réseaux filaires (Perlman, 2000). Mais les ressources très limitées dans les réseaux Ad-Hoc empêchent l'utilisation des protocoles traditionnels déjà utilisés dans les réseaux filaires. En effet, la bande passante est très sollicitée lors des échanges des messages entre les nœuds pour maintenir les chemins et les tables de routage dans le cas des protocoles utilisant l'état des liens ou les vecteurs de distance. Ainsi, de nouveaux protocoles ont été proposés pour pallier les problèmes des protocoles traditionnels et surtout le fort taux de trafic de contrôle.

Dans l'approche basée sur l'état des liens, chaque nœud envoie périodiquement des messages de contrôle ou des messages d'état des liens à ses voisins et par la suite au reste des nœuds du réseau. Une fois qu'un nœud reçoit ces messages, il maintient à jour une vue d'un sous ensemble de la topologie du réseau et applique un algorithme de plus court chemin (par exemple, l'algorithme de Dijkstra (Dijkstra, 1959)) pour déterminer les chemins vers les autres nœuds (Cormen et al., 2001) . L'avantage des protocoles basés sur l'état des liens est leur possibilité de trouver immédiatement des routes alternatives dans le cas où un lien est perdu. Plusieurs routes peuvent être utilisées simultanément vers une même destination pour répartir la charge du trafic ou pour garantir une meilleure qualité de service. Un exemple de protocole basé sur l'état des liens est OLSR (*Optimized Link State Routing*) (Clausen et Jacquet, 2003). Cet algorithme est le sujet principal de cette recherche et il est présenté plus en détail à la section 1.5 de ce chapitre.

Dans l'approche basée sur les vecteurs de distance, chaque nœud transmet à tous ses voisins des copies périodiques de sa table de routage. Ces mises à jour permettent à chaque nœud de

connaître les modifications apportées à la topologie. Le nom de vecteur de distance provient du fait que les routes sont données comme un vecteur (*distance, direction*); La distance représente une métrique, alors que la direction définit le prochain saut. Les protocoles basés sur les vecteurs de distance utilisent l'algorithme de Bellman-Ford (Bellman, 1957; Ford et Fulkerson, 1962). Le protocole DSDV (*Destination-Sequenced Distance-Vector*) est un protocole proactif basé sur les vecteurs de distance (Perkins et Bhagwat, 1994).

Protocoles de routage réactifs

Le principe des protocoles réactifs est de créer et maintenir les routes selon les besoins. Ainsi, aucune route ou information de routage ne sera calculée tant qu'un nœud n'a pas initié une communication pour demander une route vers le nœud destinataire. Lorsqu'un nœud a besoin d'une route pour communiquer avec le destinataire, une procédure de découverte de route par inondation est lancée dans tout le réseau. Grâce à cette méthode, les nœuds du réseau ne génèrent aucun trafic de contrôle sans qu'il soit nécessaire. Ceci permet de réduire la charge du trafic dans le réseau. Par contre, au moment de l'inondation pour la création d'une route, le mécanisme est très coûteux au niveau de la bande passante car tous les nœuds participent au mécanisme. S'ajoute à cela, durant cette phase de recherche de route, les paquets de données à envoyer seront mis en attente en attendant la disponibilité d'une route. Ceci entrainera une grande temporisation des paquets et un délai d'attente. Le protocole AODV (*Ad-Hoc On demand Distance Vector*) est un exemple de protocole de routage réactif que nous allons détailler dans le paragraphe 1.4.2 (Perkins, Royer et Das, 2002).

La Figure 1.3 donne une nomenclature et une classification des principaux protocoles proactifs, réactifs et hybrides développés ces dernières années (Karmakar et Dooley, 2008). Dans la suite, on explicitera plus en détail certains protocoles présentés dans la Figure 1.3.

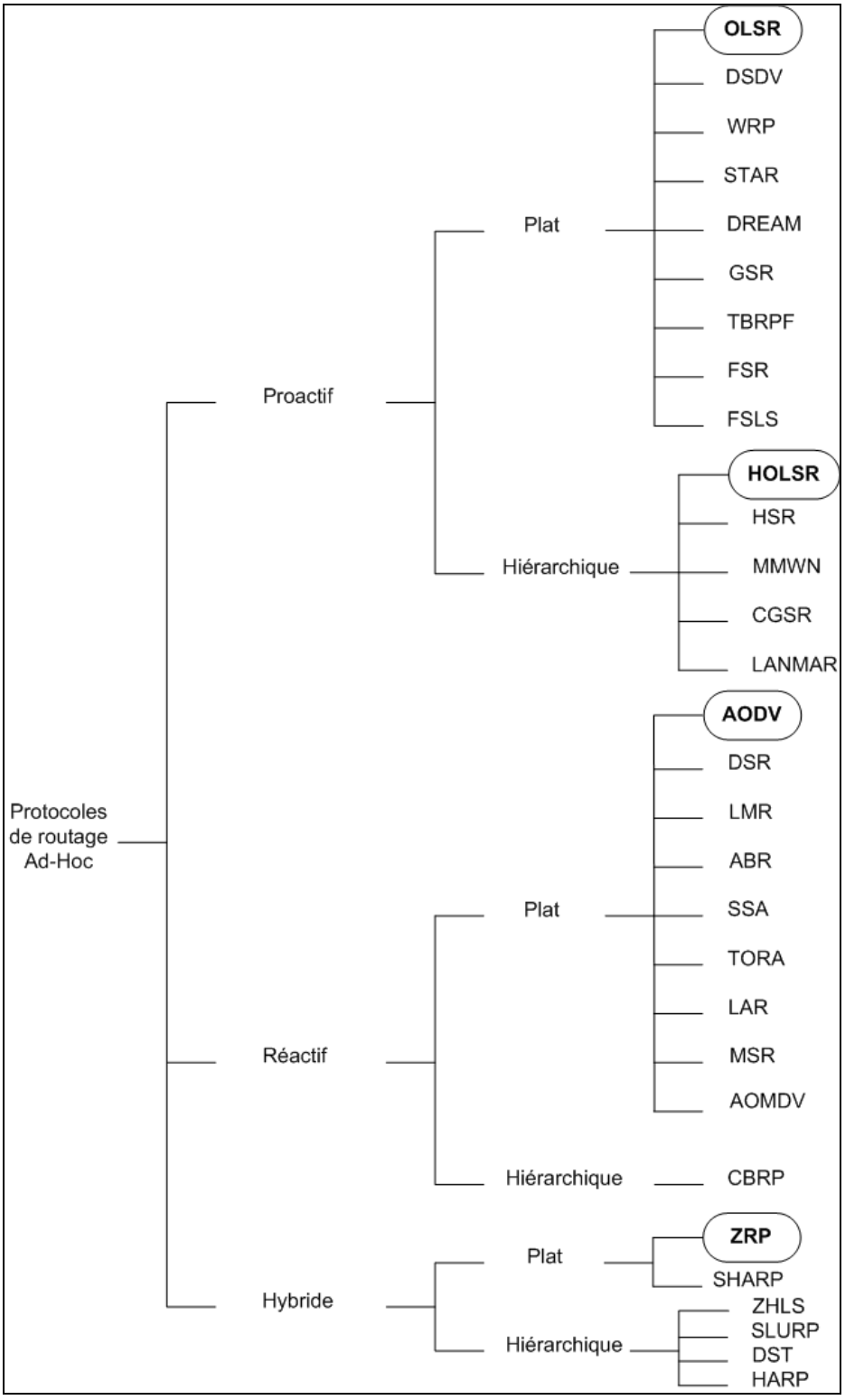

Figure 1.3 Classification des protocoles de routage Ad-Hoc.

Protocoles de routage hybride

Ce type de protocole combine les mécanismes des protocoles proactifs et réactifs. Dans cette approche, les protocoles hybrides utilisent les méthodes proactives (messages périodiques de contrôle) pour découvrir les routes dans un voisinage prédéfini. Les techniques d'inondation des protocoles réactifs sont utilisées pour obtenir les routes vers les nœuds lointains.

1.4.2 Descriptions de certains protocoles de routage pour réseaux Ad-Hoc

Le protocole AODV

AODV (*Ad-Hoc On demand Distance Vector*) est un protocole de routage réactif basé sur les vecteurs de distance (Perkins et Royer, 1999; Perkins, Royer et Das, 2002). Il est une combinaison d'une part de DSDV (*Destination-Sequenced Distance Vector Protocol*) avec son routage de saut par saut (Perkins et Bhagwat, 1994), et de DSR (*Dynamic Source Routing Protocol*) avec ses mécanismes de découverte et maintenance des routes (Broch, Johnson et Maltz, 2002).

Avec ses mécanismes de découverte et de calcul de routes sur demande, AODV réduit de façon significative la consommation de ressources dans le réseau. On ajoute à cela le fait qu'il utilise les numéros de séquence dans ses messages de contrôle pour pallier les problèmes de boucle et comptage à l'infini de l'algorithme de Bellman-Ford utilisé dans les protocoles basés sur les vecteurs de distance (Bellman, 1957; Ford et Fulkerson, 1962; Perkins, Royer et Das, 2002). Mais lorsqu'il s'agit d'un réseau dense, le protocole devient très coûteux lors du lancement des procédures de découverte de routes. Le délai d'attente moyen devient aussi grand lorsqu'il s'agit d'un réseau où les nœuds ont une grande mobilité. Dans ce cas, la topologie du réseau est très instable et de nombreuses routes doivent être recalculées.

Découverte des routes : Avec le protocole AODV, chaque nœud doit maintenir une liste de ses voisins actifs. Cette liste est obtenue par un échange périodique des messages HELLO de chaque nœud avec ses voisins immédiats. Quand un nœud source S veut envoyer des données

à un destinataire D et qu'aucune route vers cette destination n'est stockée dans la table de routage de la source, le nœud S initialise une procédure de découverte de routes. La source S envoie à ses voisins une demande de route RREQ (*Route REQest*) qui contient l'adresse de S, l'identifiant de la requête, un compteur de séquence, l'adresse de D et le compteur de nombre de sauts avec une valeur initiale zéro. Chaque nœud qui reçoit le message recherche dans sa table de routage locale s'il existe une route vers le nœud D sinon le nœud qui traite la requête RREQ incrémente le nombre de sauts et la diffuse à nouveau. Lorsque la requête atteint la destination D ou un nœud qui connaît une route vers la destination, une réponse RREP (*Route REPly*) est diffusée sur la même route de réception du RREQ (chemin inverse). La réponse RREP contient l'adresse source, l'adresse de destination, le nombre de sauts, un numéro de séquence de destination et la durée de vie du paquet. La réponse RREP passe par la route inverse vers le nœud source S. Ainsi chaque nœud, sur cette route, enregistre une entrée dans sa table de routage local vers le nœud destination avant de renvoyer le paquet. Une fois la source S reçoit le message, elle commence à envoyer les données vers D (*Voir* Figure 1.4).

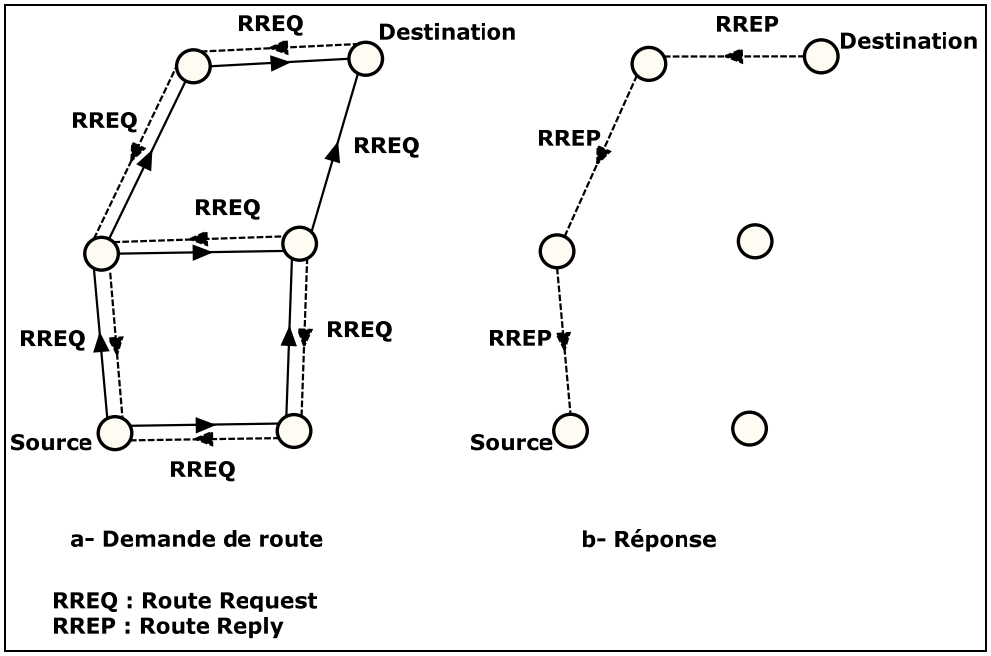

Figure 1.4 Procédure de demande de recherche de route.

Entretien des routes : L'échange des messages HELLO entre les voisins immédiats permet de mettre à jour la liste des voisins de chaque nœud. Lorsqu'un nœud N détecte qu'un autre nœud Q n'est plus accessible (Q a quitté le réseau ou est hors porté radio), N procède à une mise à jour des liens dans sa table de routage. En effet, il recherche dans sa table de routage toutes les routes qui passent par le nœud Q et les détruit avant d'annoncer à ses voisins actifs que la route passant par le nœud Q n'est plus valide. Un message RERR (*Route ERRor*) est envoyé alors au nœud source. Ainsi, la mise à jour est diffusée à travers le réseau saut-par-saut et le nœud source initie une nouvelle procédure de recherche de route vers la destination.

Le protocole OLSR

Optimized Link State Routing (Clausen et Jacquet, 2003) est un protocole de routage proactif basé sur l'état des liens pour les réseaux sans fil Ad-Hoc. Ce protocole a été choisi par le groupe de travail MANET de l'*Internet Engineering Task Force* (IETF) comme l'un des principaux protocoles de routage pour les réseaux Ad-Hoc.

L'avantage d'OLSR est qu'il utilise une technique optimisée basée sur des nœuds appelés relais multipoints MPR (*MultiPoint Relays*) pour une diffusion optimisée des messages de contrôle. Ceci réduit considérablement la charge du trafic dans le réseau. Les relais multipoints sont choisis après la phase de découverte des voisins de tous les nœuds du réseau en utilisant les messages HELLO. Ces messages permettent à chaque nœud d'avoir une vision de ses voisins immédiats et les voisins à 2-sauts. Le choix des MPR se fait alors en se basant sur les informations échangées avec les messages HELLO. Un autre type de message de contrôle de topologie TC (*Topology Control*) permet pour chaque nœud de diffuser à travers le réseau, la liste des nœuds qu'ils l'ont choisi comme MPR. Grâce aux messages TC, tous les nœuds calculent leur table de routage pour chaque destination dans le réseau. Ce protocole sera détaillé dans la section 1.5 de ce chapitre.

Le protocole HOLSR

Hierarchical Optimized Link State Routing (Villasenor-Gonzalez, Ying et Lament, 2005) est un protocole basé sur les spécifications du protocole OLSR. HOLSR se classifie comme étant

un protocole proactif hiérarchique adapté pour les réseaux sans fil hétérogènes à grande échelle. Les réseaux hétérogènes sont des réseaux multifournisseurs où plusieurs générations de réseaux sont interconnectées (par exemple GSM, UMTS, WI-FI, WIMAX, satellite, etc). Ces types de réseaux sont caractérisés par des nœuds mobiles ayant des capacités de communication distinctes, une large topologie et un défi d'interopérabilité (par exemple, les réseaux militaires). Or les protocoles de routage déjà utilisés dans les réseaux Ad-hoc ne s'adaptent pas aux réseaux hétérogènes. En effet, dans le cas d'OLSR, le volume des messages de contrôle augmente avec la densité et l'échelle du réseau ce qui diminue les performances du protocole. S'ajoute à cela, l'impossibilité de reconnaître les capacités de transmission des différents nœuds et d'en tirer profit.

Face à ces limites, HOLSR a été développé pour améliorer les performances, l'extensibilité et réduire le volume des messages de contrôle du protocole OLSR dans les réseaux hétérogènes à grande échelle en utilisant au mieux les liens à grand débit.

Le protocole HOLSR divise la topologie du réseau en plusieurs niveaux logiques. Les nœuds, avec des capacités de transmission limitées, sont classés dans le niveau 1. Les nœuds de ce niveau sont regroupés en plusieurs clusters du niveau 1 (*Voir* Figure 1.6). Ces nœuds peuvent être par exemple des soldats avec des appareils de communication à portée limitée et des liens à 1 Kbps (*Voir* Figure 1.5).

Le niveau 2 est constitué des équipements avec une ou deux interfaces sans fil qui sont capables de communiquer avec les nœuds du niveau 1 et en même temps relayer les messages au niveau 2. Les équipements du niveau 2 sont organisés en plusieurs clusters du niveau 2 et ils utilisent des moyens de communication différents de ceux du niveau 1 comme par exemple une bande radio. Dans la Figure 1.5, le niveau 2 est représenté par les véhicules de combat avec des liens radio de 16 kbps.

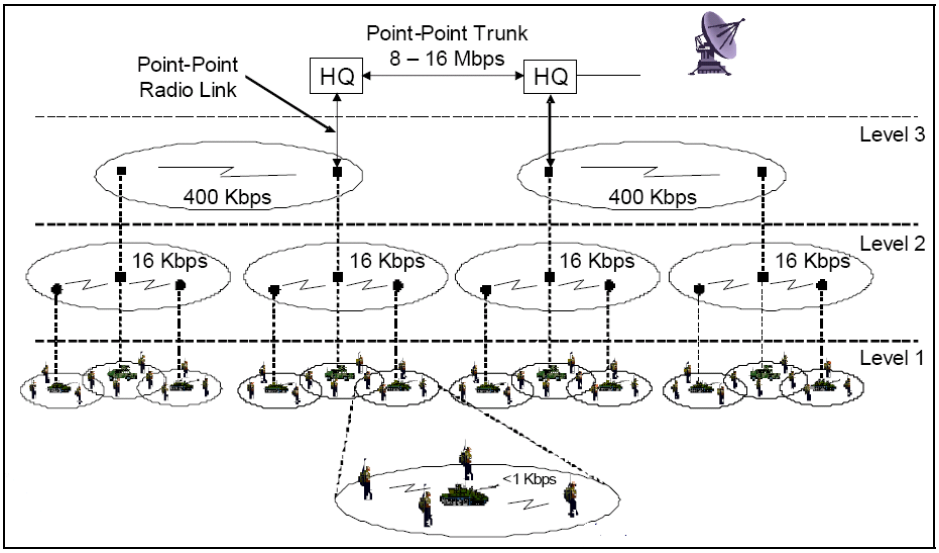

Figure 1.5 Exemple de réseau hétérogène à grande échelle.
Tirée de Defence R&D Canada (2004)

De son côté, le niveau 3 représente les nœuds à haute capacité de transmission et possédant jusqu'à trois interfaces sans fil. Ces types de nœuds peuvent communiquer avec les nœuds des niveaux 1 et 2 ainsi qu'avec les nœuds de niveau 3 (*Voir* Figure 1.6). Dans la Figure 1.5, le niveau 3 est représenté par les véhicules à très grande capacité de transmission avec un lien de 400 Kbps.

Durant la phase d'initialisation, le protocole HOLSR configure des nœuds comme des clusters. Chaque nœud déclaré comme cluster, envoie des messages périodiques appelés CIA (*Cluster ID Announcement*) pour inviter ses voisins proches à joindre son cluster. Ces messages sont envoyés avec les messages HELLO utilisés par OLSR afin de limiter le nombre de paquets envoyés sur le réseau. Ceci permet de former les clusters de chaque niveau. Une fois la hiérarchie du réseau est construite, le routage des messages entre les clusters se fait en respectant cette hiérarchie.

Les simulations avec OPNET (Villasenor-Gonzalez, Ying et Lament, 2005) ont démontré que le protocole HOLSR améliore le rendement du mécanisme de routage d'OLSR en

réduisant considérablement les messages de contrôle TC. Ainsi, HOLSR améliore et adapte au mieux l'extensibilité du protocole OLSR dans les vastes réseaux hétérogènes.

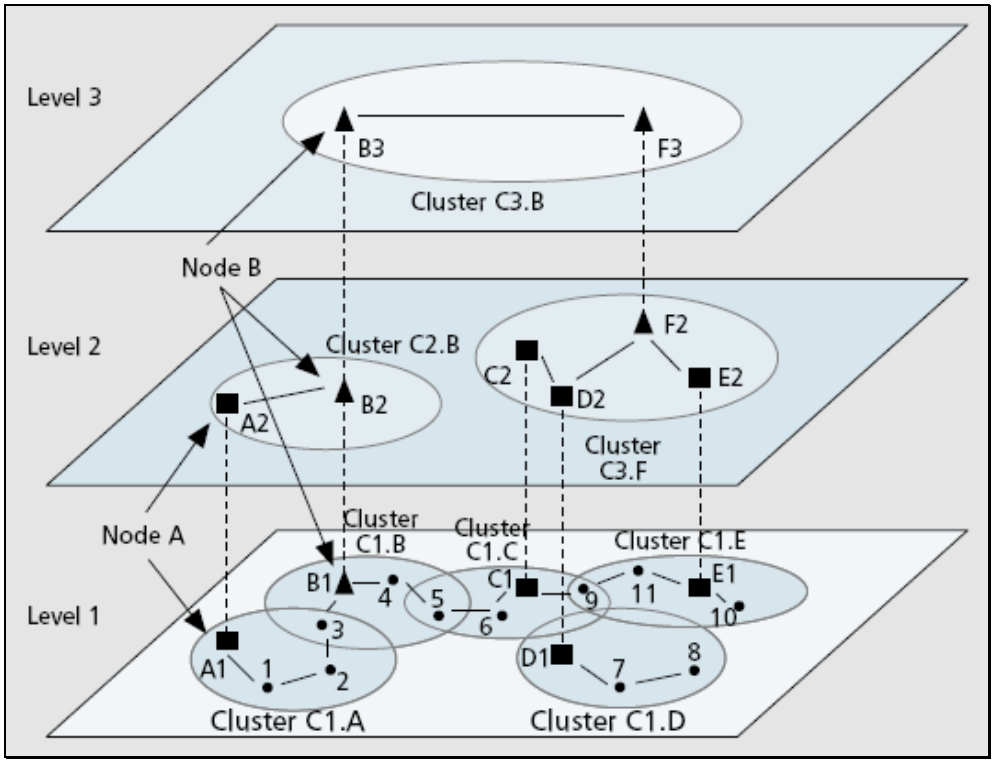

Figure 1.6 Modèle hiérarchique d'HOLSR.
Tirée de Villasenor-Gonzalez, Ying et Lament (2005, p.121)

Le protocole ZRP

ZRP (*Zone Routing Protocol*) (Haas et Pearlman, 1998; Hass, Pearlman et Samar, 2002) est un exemple de protocole hybride qui combine les approches proactive et réactive afin d'en tirer des avantages.

Le protocole ZRP divise le réseau en différentes zones qui peuvent être de différentes tailles. En effet, il définit pour chaque nœud S une zone de routage exprimée en nombre de sauts maximal σ. Ainsi, la zone de routage de S inclut tous les nœuds qui sont à une distance au maximum de σ sauts par rapport à S. Les nœuds qui sont exactement à σ sauts de S sont appelés nœuds périphériques. À l'intérieur de cette zone, ZRP utilise son protocole proactif mais à l'extérieur de sa zone de routage il utilise son protocole réactif.

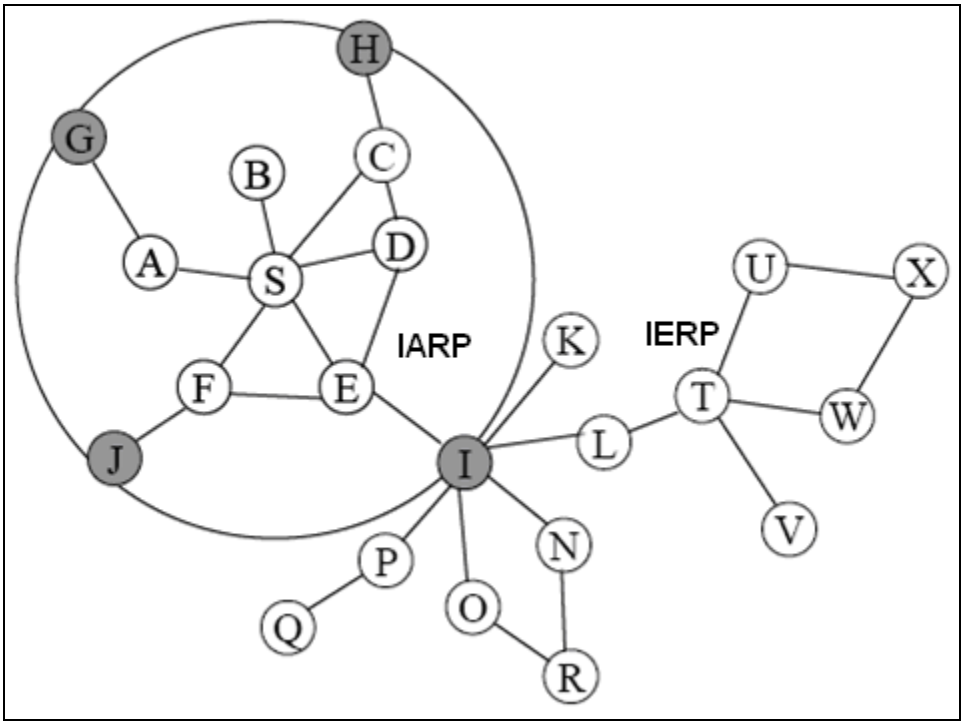

Figure 1.7 Zone de routage du nœud S (σ = 2 sauts).

Les mécanismes de routage de ZRP sont donc basés sur deux protocoles, IARP (*IntrAzone Routing Protocol*) (Haas, Pearlman et Samar, 2002c) et IERP (*IntErzone Routing Protocol*) (Haas, Pearlman et Samar, 2002b). Mais avant de passer à la phase de routage, chaque nœud doit connaître ses voisins. Dans ce but, ZRP utilise le protocole de contrôle d'accès au support (MAC) pour connaître les voisins immédiats ou le protocole NDP (*Neighbour Discovery Protocol*) pour la transmission et la gestion des échanges de messages HELLO (Hass, Pearlman et Samar, 2002). Pour un nœud S donné, ZRP utilise par la suite le protocole IARP pour découvrir les routes vers tous les autres nœuds qui se trouvent dans la zone de routage de S. Par contre, le protocole IERP est utilisé à la demande pour chercher les routes entre S et une destination D qui se trouvent à l'extérieur de la zone de routage de S (*Voir* Figure 1.7).

Un troisième protocole BRP (*Bordercast Resolution Protocol*) (Haas, Pearlman et Samar, 2002a) est inclus avec IERP pour fournir des services de *bordercasting* et définir les

frontières des zones c.-à.-d. les nœuds périphériques de chaque nœud du réseau. La Figure 1.8 donne l'architecture globale de ZRP.

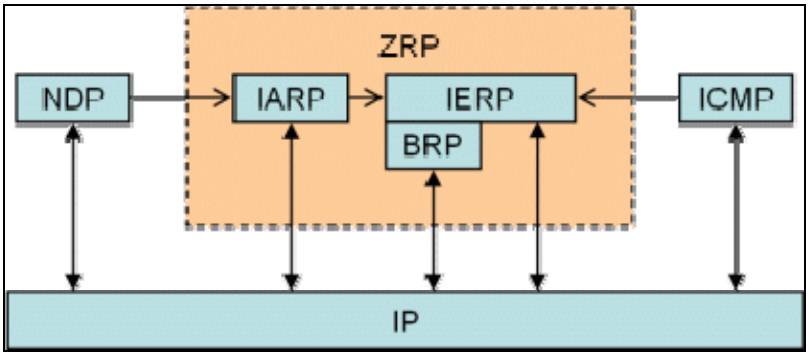

Figure 1.8 L'architecture globale de ZRP.

1.5 Le protocole OLSR

1.5.1 Description du protocole OLSR

Le protocole OLSR (*Optimized Link State Routing*) appartient à la famille des protocoles proactifs et il a été développé pour les réseaux Ad-Hoc (Clausen et al., 2001; Clausen et Jacquet, 2003; Jacquet et al., 2001). Il a été développé dans le cadre du projet *HIPERCOM* à l'*INRIA*. OLSR a été retenue par le groupe MANET de l'IETF en vue d'une standardisation. La version 1 d'OLSR a été standardisée dès 2003 et elle est spécifiée dans le *RFC3626* (Clausen et Jacquet, 2003). La version suivante, OLSR v2 est en cours de développement et de standardisation. Cette version apporte certaines optimisations et des petites modifications, mais en générale elle ressemble à OLSR v1.

Le protocole OLSR utilise des échanges périodiques de messages HELLO pour permettre à chaque nœud de connaître ces voisins à 1-saut et à 2-sauts. Par la suite, OLSR utilise une diffusion optimisée des messages de contrôle grâce à l'utilisation des relais multipoints MPR. Ceci permet à chaque nœud d'avoir une vue de la topologie et ainsi utiliser un algorithme de plus court chemin pour calculer sa table de routage vers toutes les destinations.

L'inondation par relais multipoints réduit considérablement la charge du trafic dans le réseau car juste une partie des nœuds participent au processus d'inondation. Ainsi, le protocole OLSR est très souhaitable pour les réseaux très denses.

Le protocole OLSR a été mis en application dans de grands projets. La DARPA a choisi OLSR comme protocole de référence pour les réseaux tactiques (Brown et al., 2003). D'autres applications dans le domaine civil ont été réalisées. Avec ce grand développement d'OLSR, de nombreuses extensions de ce protocole sont en chantier, on pourra noter par exemple :

- **QOLSR :** OLSR avec qualité de service (QoS)
- **OLSR v6 :** OLSR avec auto-configuration
- **MOLSR :** Multicast OLSR
- **SOLSR :** OLSR avec sécurité
- **PS-OLSR :** OLSR avec power saving

1.5.2 Détection de voisinage

Les réseaux Ad-Hoc sont caractérisés par une topologie dynamique et changeante. Afin de détecter tout changement dans le réseau et générer les informations sur la topologie, le protocole OLSR se base essentiellement sur la détection et la mise à jour de la liste des voisins de chaque nœud. Dans tout ce qui suit, on considère que tous les nœuds ont une seule interface sans fil.

On peut classifier les liens entre deux nœuds en trois catégories :

- **Asymétrique :** Un lien est dit asymétrique si le premier nœud reçoit les messages de l'autre nœud mais il n'a pas reçu la confirmation que l'autre nœud l'entend.
- **Symétrique :** Un lien est dit symétrique si chaque nœud entend l'autre.
- **Perdu :** Un lien est dit perdu si ce lien a été déclaré précédemment étant symétrique ou asymétrique mais à ce moment aucun message n'est reçu du nœud déclaré perdu.

Dans le but de découvrir les nœuds voisins, chaque nœud envoie périodiquement à tous ses voisins des messages HELLO. Ces messages contiennent les informations concernant les nœuds voisins, les nœuds qui sont choisis comme MPR (c.-à-d. *MPRSelector set*) et la liste des nœuds qui sont déclarés par ce nœud comme asymétriques.

La Figure 1.9 décrit le processus de découverte des voisins entre deux nœuds A et B. En premier, le nœud envoie à B un message HELLO qui ne contient aucune information. Une fois B reçoit ce message, il enregistre A comme voisin asymétrique car B ne trouve pas son adresse dans le message. Le nœud B envoie par la suite un message HELLO déclarant qu'il entend A. Ce dernier trouve son adresse dans le message et enregistre B comme voisin symétrique. À son tour, B trouve son adresse dans le message HELLO de A et déclare ce dernier comme voisin symétrique.

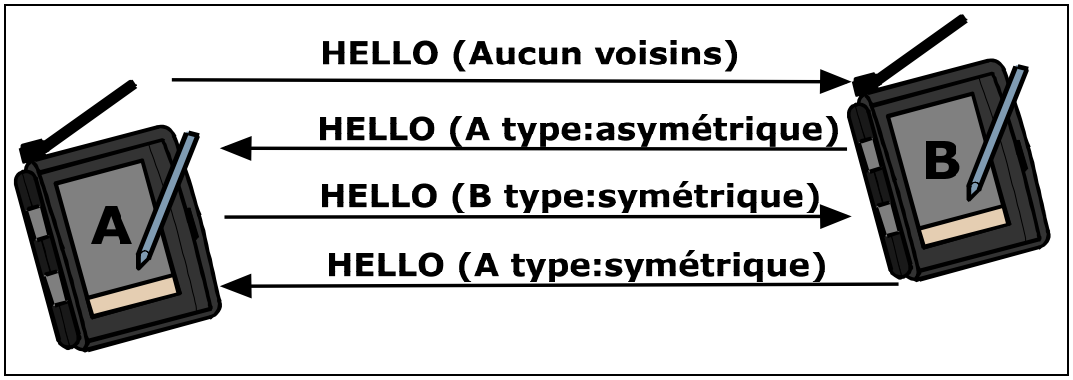

Figure 1.9 Échange des messages HELLO.

C'est ainsi que chaque nœud du réseau génère périodiquement des messages HELLO avec une duré de vie *TTL=1*. Ces messages sont reçus par les voisins à 1-saut et ne sont pas relayés par ceux-ci.

La Figure 1.10 présente le format des messages HELLO. Chaque message se compose en plusieurs sections qui correspondent à différents états de liens. La liste des adresses des interfaces voisins qui possèdent un lien symétrique sont listés dans les champs *Neighbor Interface Address*.

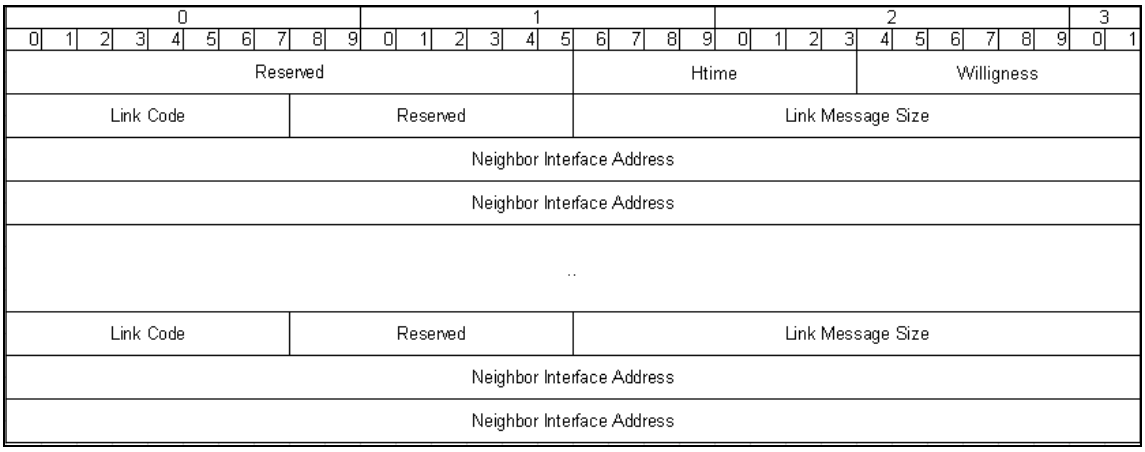

Figure 1.10 Format du message HELLO.
Tirée de Clausen et Jacquet (2003, p. 27)

Le champ *Link Code* de taille 8 bits, contient à la fois les informations concernant les liens vers les nœuds voisins et le type de ces derniers. Le Tableau 1.1 et le Tableau 1.2 présentent la liste des valeurs possibles pour les deux champs de type de lien et type de voisin selon les spécifications du *RFC3626* (Clausen et Jacquet, 2003).

Tableau 1.1 Champs *Link Code*

0	1	2	3	4	5	6	7
				Type de voisin		Types de liens	

Tableau 1.2 Valeurs possible pour le champ Link Code

Types de lien	
UNSPEC_LINK	Pas d'informations
ASYM_LINK	Lien asymétrique
SYM_LINK	Lien symétrique
LOST_LINK	Lien est perdu
Types de voisin	
SYM_NEIGH	Voisin symétrique
MPR_NEIGH	Voisin a été sélectionné comme MPR
NOT_NEIGH	Pas de voisins / Pas encore symétrique

L'ensemble des voisins immédiats (ou à 1-saut) d'un nœud s et qui possèdent un lien symétrique avec ce dernier est noté $N_1(s)$. Les voisins à 2-sauts d'un nœud s sont définis comme étant l'ensemble suivant : $N_2(s) = \{y \mid y \neq s \wedge y \notin N_1(s) \wedge (\exists x \in N_1(s))[y \in N_1(s)]\}$. Ces deux ensembles $N_1(s)$ et $N_2(s)$ de chaque nœud s sont construits grâce aux échanges périodiques des messages HELLO. Ceci permet à tous les nœuds d'avoir une vision à 1-saut et à 2-sauts de la topologie du réseau et ainsi avoir toutes les informations nécessaires pour construire les chemins entre une source et une destination dans la zone à 1-saut et à 2-sauts.

1.5.3 Sélection des Relais Multipoints

La technique d'inondation est utilisée dans plusieurs algorithmes de routage pour la diffusion des messages à tous les nœuds dans un réseau. Avec cette technique, chaque nœud renvoie une copie du message qu'il reçoit pour la première fois à tous ces voisins immédiats. Ce mécanisme a un impact sur les ressources du réseau en termes de bande passante. Or, les réseaux Ad-Hoc ont des ressources limitées et les enjeux de performance sont capitaux.

Dans ce contexte, le protocole OLSR utilise une technique appelée inondation par relais multipoint pour optimiser la diffusion à travers le réseau et ainsi réduire la charge du trafic. Ainsi, chaque nœud s sélectionne un sous ensemble de points appelés MPR (*Multipoint Relay*) parmi ses voisins de $N_1(s)$ et qui lui permettent d'être rejoint par tous les nœuds dans $N_2(s)$ (Clausen et Jacquet, 2003). Or, la connaissance de $N_1(s)$ et $N_2(s)$ de chaque nœud s permet la diffusion des messages dans tout le réseau (Jacquet et al., 1997). L'échange périodique des messages HELLO permet à chaque nœud dans le réseau de mettre à jour ses ensembles $N_1(s)$ et $N_2(s)$. C'est ainsi que l'ensemble des relais multipoints est recalculé à chaque changement dans la topologie du réseau.

L'ensemble $MPR(s)$ des relais multipoints d'un nœud s forme un arbre recouvrant et il est défini de la manière suivante :

a) $MPR(s) \subseteq N_1(s)$

b) $(\forall y \in N_2(s))(\exists x \in MPR(s))[y \in N_1(s)]$

La Figure 1.11 montre la différence entre l'inondation par relais multipoints et l'inondation classique. On remarque que dans le cas classique, il faut 24 retransmissions pour atteindre les nœuds à 3-sauts du nœud s. Alors que dans le cas où on utilise les relais multipoints, seulement 11 retransmissions sont nécessaires pour avoir les mêmes résultats.

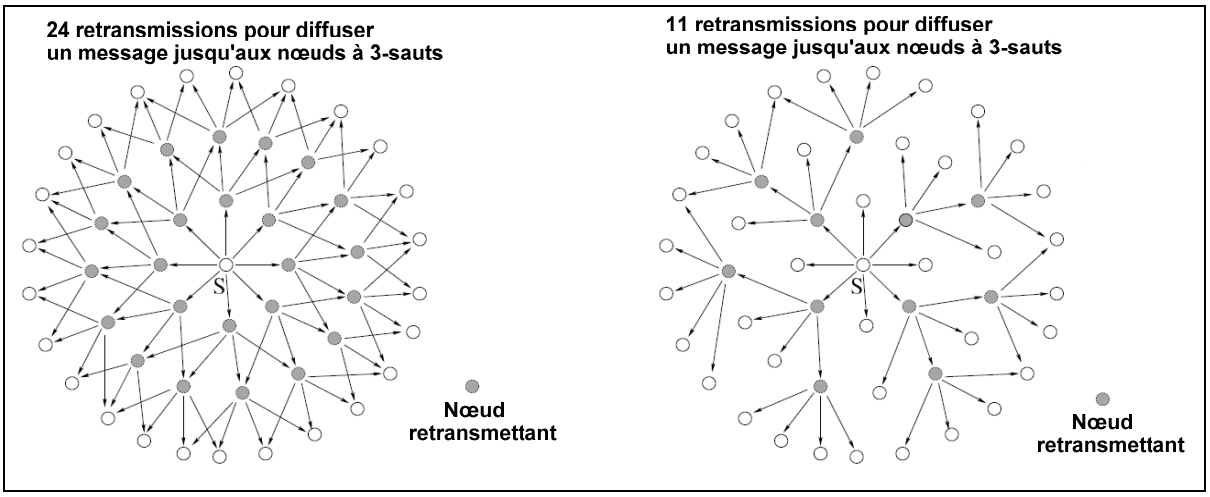

Figure 1.11 Diffusion par inondation classique vs inondation par relais multipoints.

Dans un cas général, pour atteindre le voisinage $N_2(s)$ d'un nœud *s*, il faut $|MPR(s)|+1$ émissions. Donc pour minimiser le nombre d'émissions possibles dans le réseau et par la suite augmenter les performances, il faut trouver un nombre minimal de relais multipoints pour chaque nœud (c.-à-d., minimum de $|MPR(s)|$). Or ce problème du choix des MPR est un problème NP-complet car ceci est équivalent à calculer un sous ensemble dominant dans un graphe (Qayyum, Viennot et Laouiti, 2002). Plusieurs heuristiques ont été proposées à cet effet dans plusieurs articles (Clausen et Jacquet, 2003; Mans et Shrestha, 2004; Qayyum, Viennot et Laouiti, 2002) .

Données : Tout nœud s avec ses voisins $N_1(s)$ et $N_2(s)$.
Résultat : L'ensemble $MPR(s)$.

début
  $MPR(s) \leftarrow \emptyset$;
  Trouver les nœuds isolés dans $N_2(s)$ qui sont couverts par un seul nœud dans $N_1(s)$;
  pour *tout nœud* y *dans* $N_2(s)$ *isolé* **faire**
    Soit $x \in N_1(s)$ le seul voisin de ce nœud y;
    Ajouter x à $MPR(s)$;
    Éliminer tous les nœuds dans $N_2(s)$ couverts par x;
  fin
  tant que $N_2(s) \neq \emptyset$ **faire**
    Trouver $x \in N_1(s)$ tq
    - x couvre le maximum des nœuds dans $N_2(s)$;
    - x a le maximum des voisins ;

    Ajouter x à $MPR(s)$;
    Éliminer tous les nœuds dans $N_2(s)$ couverts par x;
  fin
fin

Algorithme 1.1 Sélection des MPR par OLSR (*RFC3626*).

Pour la suite de ce mémoire, seul l'algorithme vorace présenté dans le *RFC3626* sera considéré (Clausen et Jacquet, 2003). Cette heuristique de sélection des MPR est présentée dans l'Algorithme 1.1. Ce algorithme représente une solution optimale à log(*n*) près (Laouiti, Qayyum et Viennot, 2000; Viennot, 1998).

La Figure 1.12 présente un exemple d'application de l'algorithme décrit précédemment pour la sélection de l'ensemble des relais multipoints $MPR(s)$ d'un nœud s.

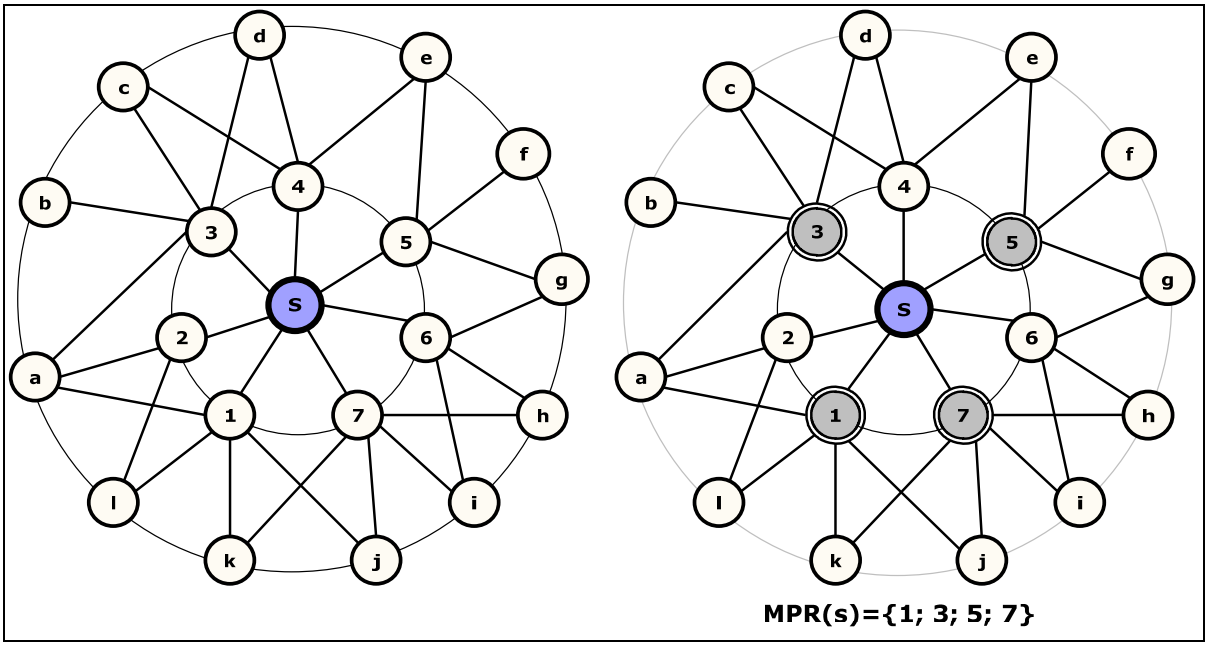

Figure 1.12 Exemple de sélection des relais multipoints.

1.5.4 Déclaration des relais multipoints

L'Algorithme 1.1 est utilisé par chaque nœud dans le réseau pour construire l'ensemble des relais multipoints. Afin de fournir les informations sur la topologie nécessaire pour construire les routes et ainsi garantir le routage des paquets, chaque nœud qui a été sélectionné comme étant MPR, diffuse périodiquement des messages de contrôle de la topologie TC (*Topology Control*) (Clausen et Jacquet, 2003). Ces messages sont reçus par tous les nœuds mais transmis juste par les MPR.

Chaque nœud x qui a été sélectionné comme MPR maintient une liste des voisins qui l'ont sélectionné comme relais multipoint. On définie cette ensemble par :

$$MPRSel(x) = \left\{ y \in N_1(x) \mid x \in MPR(y) \right\}$$

Chaque message *TC* (*Voir* Figure 1.13), envoyé par un nœud x, contient la liste $MPRSel(x)$ ainsi qu'un numéro de séquence (*ANSN*) associé au message. Ces messages permettent à chaque nœud de maintenir à jour sa table d'information sur la topologie et ainsi faciliter le calcul de sa table de routage.

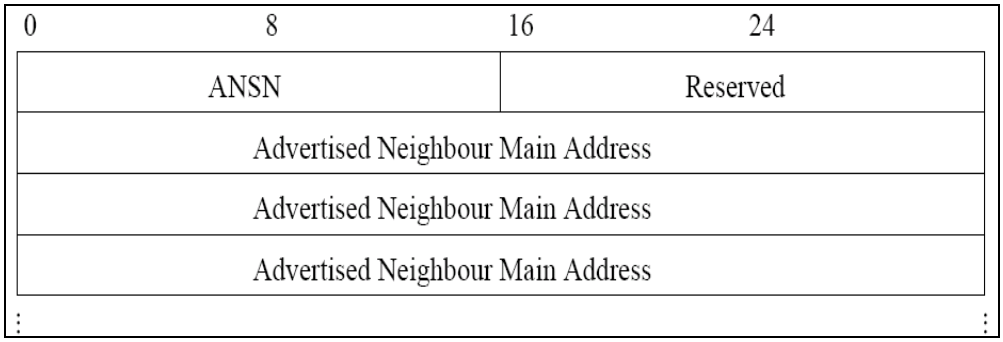

Figure 1.13 Format du message TC.
Tirée de Clausen et Jacquet (2003, p. 42)

1.5.5 Calcule des routes

Chaque nœud maintient une table de routage qui lui permet d'acheminer les paquets vers un destinataire. Ces tables de routage sont calculées grâce à l'algorithme de plus courte chemin de Dijkstra (Dijkstra, 1959) en se basant sur les informations conservées par les nœuds et aussi les informations fournies par les messages de contrôle TC. Ces tables de routage sont recalculées à chaque changement survenu dans la topologie et ainsi permettre de mettre à jour les routes vers toutes les destinations dans le réseau.

1.6 Complexité et comparaison des protocoles de routage

Les tableaux suivants donnent une brève comparaison entre plusieurs protocoles de routage pour les réseaux Ad-Hoc selon leur catégorie proactif, réactif ou hybride (Abolhasan, Wysocki et Dutkiewicz, 2004; Karmakar et Dooley, 2008; Zou, Ramamurthy et Magliveras, 2002).

Tableau 1.3 Sommaire des protocoles proactifs

	WCC	*WTC*	*SR*	*Fréquence des mises à jour*	*Nœuds Critiques*	*MH*	*Advantages*	*Inconvénients*
OLSR	O*(N)*	O*(D)*	P	Périodique	MPR	Oui	Diffusion optimisée des messages de contrôle par rapport aux autres protocoles basés su l'état des liens.	Les informations sur les voisins à 1-saut et 2-sauts sont nécessaires.
HOLSR	O*(N)*	O*(D)*	H	Périodique	Racines des clusters	Oui	Idéal pour les réseaux MANET hétérogènes à grande échelle.	Les informations sur les voisins à 1-saut et 2-sauts sont nécessaires; Ajout de nouvelles structures pour former et maintenir les clusters.
DSDV	O*(N)*	O*(D)*	P	Périodique et sur-demande	Non	Oui	Simple, absence de problème de boucle de routage et de compteur à l'infinie.	Importante activité sur le réseau lors des demandes de mise à jour; Convergence lente; Tendance à créer des boucles de routage dans les réseaux très grands.
CGSR	O*(N)*	O*(D)*	H	Périodique	Racines des clusters	Non	Surdébit de routage est petit par rapport à DSDV; Schéma d'adressage simple.	La complexité en temps *WTC* est très grande par rapport à DSDV et WRP en cas des ruptures des liens entre les racines des clusters.
DREAM	O*(N)*	O*(D)*	P	Sur-demande	Non	Non	Petit surdébit de routage.	Nécessite le GPS.
STAR	O*(N)*	O*(D)*	P	Sur-demande	Non	Non	Réduit le nombre de paquets de mise à jour dans le réseau.	Choix non optimal des routes vers les destinations; Consomme beaucoup de mémoire et surcharge le réseau dans le cas des réseaux MANET très grands.
HSR	O*(n*l)*	O*(D)*	H	Périodique	Racines des clusters	Non	Petit surdébit de routage et consomme peu de mémoire par rapport à tous les protocoles proactifs plats.	Ajout de nouvelles structures pour former et maintenir les clusters.
TBRPF	O*(N)*	O*(D)*	P	Périodique et sur-demande	Nœud parent	Oui	Petit WCC.	La charge dans le réseau augmente avec la mobilité des nœuds et la taille du réseau.
FSR	O*(N)*	O*(D)*	P	Périodique	Non	Non	Réduit le nombre de paquets de mise à jour dans les réseaux très grands.	Choix non optimal des routes vers les destinations.
LANMAR	O*(N)*	O*(D)*	H	Périodique	Non	Non	Offre l'extensibilité dans les réseaux MANET très grands.	Choix non optimal des routes vers les destinations.

WCC: *Worst Case Communication Complexity*, c.-à.-d. nombre de messages nécessaires pour effectuer une opération de mise à jour;
WTC: *Worst Case Time complexity*, c.-à.-d. complexité en temps (ou le nombre d'étapes) pour effectuer une mise à jour;
SR : Structure du routage; *P*: Plat; *H*: Hiérarchique; *MH*: Messages Hello; *N*: Nombre de nœuds dans le réseau; *D*: Diamètre du réseau;
h: Hauteur de l'arbre de routage; *n*: Nombre moyen de nœuds dans un cluster; *l*: nombre de niveaux hiérarchiques.

Tableau 1.4 Sommaire des protocoles réactifs

	WCC [DR]	*WCC [MR]*	*WTC [DR]*	*WTC [MR]*	*SR*	*RM*	*Advantages*	*Inconvénients*
AODV	O*(2N)*	O*(2N)*	O*(2D)*	O*(2D)*	P	Non	Adapté aux topologies très	Grand délai; Nécessite les

							dynamiques; Possibilité de routage Multicast.	messages *HELLO*; Ne supporte pas des routes multiples.
DSR	O*(2N)*	O*(2N)*	O*(2D)*	O*(2D)*	P	Oui	Routes multiples; Les nœuds intermédiaires ne stockent pas les informations sur les routes.	Grand délai.
LMR	O*(2N)*	O*(2A)*	O*(2D)*	O*(2D)*	P	Oui	Routes multiples.	Problèmes de boucle de routage.
CBRP	O*(2X)*	O*(2A)*	O*(2D)*	O*(2B)*	H	Non	Réduit la charge de communication; Juste les clusters échangent les informations de routage.	Problèmes de boucle de routage. Ajout de nouvelles structures pour former et maintenir les clusters.
AOMDV	O*(2N)*	O*(2N)*	O*(2D)*	O*(2D)*	P	Oui	Multiples routes disjoints.	Nécessite des messages HELLO périodiques.

WCC: *Worst Case Communication Complexity*, c.-à.-d. nombre de messages nécessaires pour effectuer une opération de mise à jour;
WTC: Worst Case Time complexity, c.-à.-d. complexité en temps (ou le nombre d'étapes) pour effectuer une mise à jour;
DR: Découverte de route; *MR* : Maintenance de route; *SR* : Structure du routage; *P* : Plat; *H*: Hiérarchique; *RM*: Routes multiples;
N: Nombre de nœuds dans le réseau; *D*: Diamètre du réseau; *A*: Nombre de nœuds affectés; *B*: Diamètre de la région affecté; *X*: Nombre de clusters dans CBRP.

Tableau 1.5 Sommaire des protocoles hybrides

	WCC [I]	***WCC [In,DR]***	***WCC [In,MR]***	***WTC [I]***	***WTC [In,DR]***	***WTC [In,MR]***	***SR***	***Advantages***	***Inconvénients***
ZRP	O*(n)*	O*(N+r)*	O*(N+r)*	O*(d)*	O*(2D)*	O*(2D)*	P	Réduit la charge de communication en comparaison avec les protocoles proactifs; Découverte rapide des routes en comparaison avec les protocoles réactifs.	Problème de chevauchement des zones entre les nœuds.
ZHLS	O*(N/M)*	O*(N+r)*	O*(N+r)*	O*(d)*	O*(2D)*	O*(2D)*	H	Absence de chevauchement des zones.	Nécessite le GPS.

WCC: *Worst Case Communication Complexity*, c.-à.-d. nombre de messages nécessaires pour effectuer une opération de mise à jour;
WTC: *Worst Case Time complexity*, c.-à.-d. complexité en temps ou (le nombre d'étapes) pour effectuer une mise à jour;
DR: Découverte de route; *MR* : Maintenance de route; *SR* : Structure du routage; *P* : Plat; *H*: Hiérarchique;
I: Intra zone; *In*: Inter zone; *N*: Nombre de nœuds dans le réseau; *D*: Diamètre du réseau ; *d*: Diamètre d'une zone locale;
n: Nombre de nœuds dans une zone; *r*: Nombre de nœuds dans le chemin *Reply-Path; M* : Nombre de zones.

CHAPITRE 2

SÈCURITÉ ET VULNÉRABILITÉS DES RÉSEAUX AD-HOC

Ce chapitre fait la revue de littérature des différentes vulnérabilités dans les réseaux Ad-Hoc. L'objectif est de donner les grandes catégories d'attaques dans les réseaux Ad-Hoc et détailler les vulnérabilités du protocole OLSR. Des solutions pour pallier les problèmes de sécurité d'OLSR seront données dans la dernière partie de ce chapitre.

2.1 Objectifs de la sécurité

Les principaux objectifs de sécurité pour les réseaux Ad-Hoc sont regroupés sous trois catégories importantes : la confidentialité, l'intégrité et la disponibilité. Cette approche classique est décrite dans (Bishop, 2005).

2.1.1 Confidentialité

L'objectif de la confidentialité est de garantir l'accès aux informations seulement pour les utilisateurs ou les systèmes autorisés. Dans le contexte des réseaux Ad-Hoc, la confidentialité consiste à refuser l'accès aux informations échangées entre deux nœuds dans le réseau par tout nœud malveillant ou non désiré. Or, les réseaux Ad-Hoc sont caractérisés par la diffusion générale des informations, ce qui constitue un vrai challenge pour la confidentialité.

2.1.2 Intégrité

Le rôle de l'intégrité est de garantir la protection des messages ou des informations échangées dans le réseau contre toute modification ou altération par une personne non autorisée. Dans les réseaux Ad-Hoc, le bon fonctionnement du réseau repose essentiellement sur l'échange des messages de contrôle fournissant les informations pour le routage. Dans ce contexte, il est important, en premier lieu, de garantir l'intégrité des fonctionnalités de routage et des messages de contrôle contre toutes les modifications non autorisées.

En plus de l'intégrité, il faut s'assurer de l'imputabilité ou de la non-répudiation des informations. Cette fonction donne l'assurance de l'identité du nœud originaire d'un message. La non-répudiation est utile dans la détection et l'isolation des nœuds malicieux. Par exemple, si un nœud *A* reçoit un faux message de la part d'un nœud *B*, la non-répudiation permet au nœud *A* de dénoncer *B* et informer les autres nœuds que *B* est compromis.

2.1.3 Disponibilité

La disponibilité garantit le fonctionnement en permanence des services et garantit l'accès des usagers à ces services. La disponibilité est difficilement applicable dans les réseaux Ad-Hoc. En effet, à cause de la mobilité des nœuds, un protocole de routage ne pourrait pas maintenir toutes les routes vers tous les nœuds et surtout les nœuds qui quittent le réseau. Mais certains types d'attaques pourraient avoir effet sur la disponibilité de nœuds participant au réseau ou la disponibilité de certains services (attaques qui épuisent les batteries des nœuds, attaques par déni de service, etc).

2.2 Vulnérabilités et types d'attaques dans les réseaux Ad-Hoc

2.2.1 Classification d'attaques dans les réseaux Ad-Hoc

Les mécanismes de sécurité dans les environnements des réseaux Ad-Hoc présentent de grands défis (Anjum et Mouchtaris, 2007). Ce type de réseaux a hérité à la fois des problèmes de sécurité des réseaux câblés et aussi ceux des réseaux sans fil. S'ajoute à cela, la nature des réseaux Ad-Hoc qui se caractérise par une architecture *peer-to-peer* ouverte, une topologie dynamique et extensible, des ressources limitées et un canal radio accessible par tout le monde (Anjum et Mouchtaris, 2007; Mishra, 2008).

Les attaques sur les réseaux Ad-Hoc sont généralement divisées en deux catégories :

- Attaques passives : Principalement des attaques d'écoute de données.
- Attaques actives : Des attaques pour lesquelles un attaquant doit modifier, altérer ou générer des messages.

Figure 2.1 donne une classification des attaques par rapport aux couches OSI. Pour plus de détails, voir Tableau 2.1.

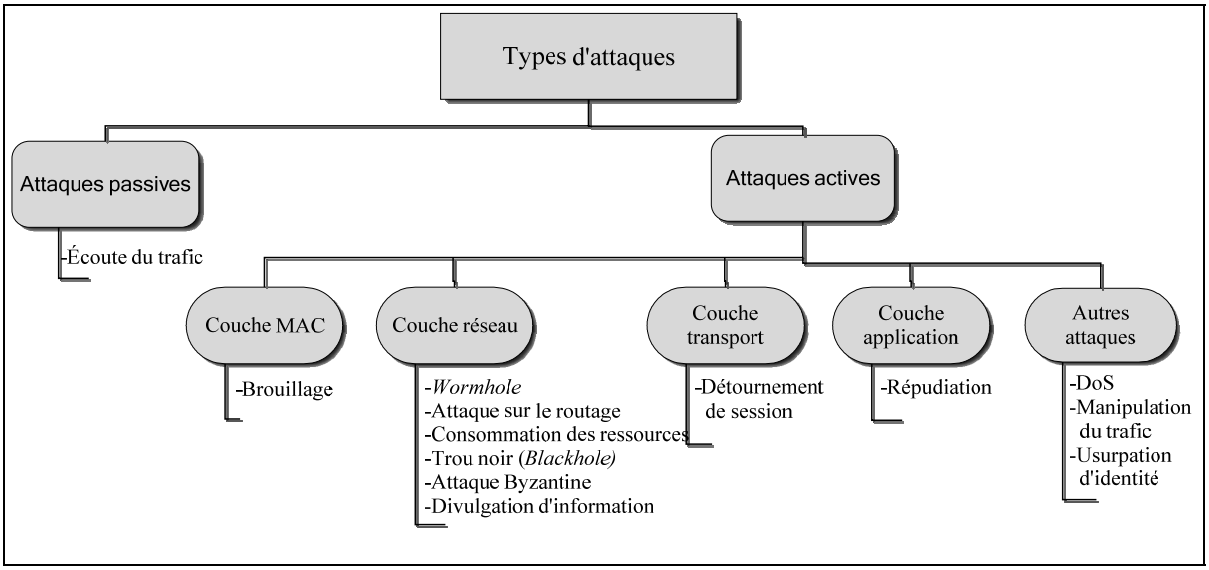

Figure 2.1 Classifications des attaques dans les réseaux Ad-Hoc.

2.2.2 Défense contre les attaques dans les réseaux Ad-Hoc

Plusieurs solutions ont été proposées pour pallier les problèmes de sécurité dans les réseaux Ad-Hoc. Le Tableau 2.1 présente des exemples de solutions proposées pour la sécurité des réseaux Ad-Hoc.

Tableau 2.1 Solutions proposées pour la sécurité dans les réseaux Ad-Hoc

Attaques	Définition	Solutions proposées
Wormhole	Un attaquant pourrait rediriger le trafic entre deux zones géographiquement éloignées pour créer un vertex dans la topologie et ainsi avoir une bonne position géographique pour contrôler le trafic qui passe par lui.	*Packet Leashes (Hu, Perrig et Johnson, 2003).*
Attaque de routage	Un nœud malicieux pourrait perturber le fonctionnement d'un protocole de routage en modifiant les informations de routage, fabriquer les fausses informations de routage ou usurper l'identité d'un autre nœud.	SEAD (Perkins et Bhagwat, 1994), ARAN (Sanzgiri et al., 2002), ARIADNE (Hu, Perrig et Johnson, 2002), SAODV (Zapata, 2002).
Brouillage *(Jamming)*	C'est une attaque classique sur la disponibilité du canal de communication grâce à la génération massive d'une grande quantité d'interférence radio.	FHSS, DSSS (Wang et al., 2006).
Attaque trou noir (*Backhole attack*)	Le but de cette attaque est la falsification des informations de routage ou le détournement du trafic.	(Ramaswamy et al., 2003).
Attaque sur les ressources	Les réseaux MANET sont caractérisés par des ressources limitées (batterie et bande passante). Une attaque sur les ressources pourrait avoir des conséquences sur la disponibilité.	SEAD (Perkins et Bhagwat, 1994).
Attaque Byzantine	Grâce à cette attaque, un nœud malicieux altère les messages et pourrait créer des problèmes de boucle de routage, routage de paquets vers des chemins non optimaux, sélectionner les paquets à rejeter… Ce type d'attaque est difficile à détecter car le réseau semble fonctionner correctement.	OSRP (Awerbuch et al., 2002), (Awerbuch et al., 2004).
DoS	Ce type d'attaque consiste à envoyer délibérément des messages pour causer une saturation de la bande passante et paralyser le réseau.	SEAD (Perkins et Bhagwat, 1994), ARIADNE (Hu, Perrig et Johnson, 2002), SAODV (Zapata, 2002).
Divulgation d'information	L'échange des informations confidentielles doit être protégées contre l'écoute ou l'accès non autorisé.	SMT (Papadimitratos et Haas, 2003), SRP (Papadimitratos et Haas, 2002).
Répudiation	Ce type d'attaque a une conséquence sur l'intégrité des communications entre les nœuds dans le réseau.	ARAN (Sanzgiri et al., 2002).
Usurpation d'identité	L'usurpation d'identité a pour but la falsification des informations relatives aux identités. Ce qui pourrait conduire à l'isolement de nœuds, l'échange de fausses informations de routage et l'atteinte à la confidentialité et l'intégrité.	ARAN (Sanzgiri et al., 2002), SAODV (Zapata, 2002)..

SEAD: *Secure Efficient Ad hoc Distance vector routing protocol*; SAODV: *Secure Ad-Hoc On-demand Distance Vector routing;* ARAN: *Authenticated Routing for Ad-Hoc Networks*; ARIADNE: *A Secure On-Demand Routing Protocol for Ad-Hoc Networks*; FHSS: *Frequency-Hopping Spread Spectrum*; OSRP*: On-demand Secure Routing Protocol;* SMT: *Secure Message Transmission Protocol;* SRP: *Secure Routing Protocol for Mobile Ad-Hoc Network;* DSSS: *Direct-Sequence Spread Spectrum.*

2.3 Vulnérabilités et types d'attaques spécifiques au protocole OLSR

Dans un réseau utilisant le protocole de routage OLSR, chaque nœud doit correctement générer et renvoyer les messages HELLO et TC selon les spécifications du protocole (Clausen et Jacquet, 2003). Une faille dans ce processus aurait un effet sur le bon fonctionnement du protocole de routage et ainsi sur le réseau. Or, le protocole OLSR ne fournit aucune spécification de sécurité à prendre en compte ce qui rend ce protocole vulnérable à plusieurs types d'attaques. Dans cette section, on fournira les différentes vulnérabilités du protocole OLSR en se basant sur différents articles et en particulier sur les travaux de (Adjih et al., 2005) et de (Clausen et Baccelli, 2005).

2.3.1 Classifications des vulnérabilités et des attaques

a) Génération incorrecte du trafic

On peut distinguer deux façons de générer du trafic de contrôle incorrect :

1. Mystification d'identité (*Identity spoofing*) : Un nœud malveillant peut générer un trafic de contrôle prétendant qu'il est un autre nœud.
2. Mystification de lien (*Link spoofing*) : Un nœud malveillant peut signaler une relation de voisinage avec des nœuds inexistants ou qui ne font pas partie de ses voisins. Il peut aussi signaler un ensemble incomplet de voisins.

Génération incorrecte du message HELLO

Un nœud malveillant peut usurper l'identité d'un autre nœud en utilisant les messages HELLO (*Identity spoofing*). Dans la Figure 2.2, le nœud malveillant *m* peut usurper l'identité du nœud *a* en envoyant des messages HELLO prétendant qu'il est le nœud *a*. Dans ce cas, les nœuds *i* et *g* vont annoncer à ses voisins que le nœud *a* est accessible à travers le nœud *m*. Ceci peut causer des conflits de routes vers le nœud *a* dans tout le réseau.

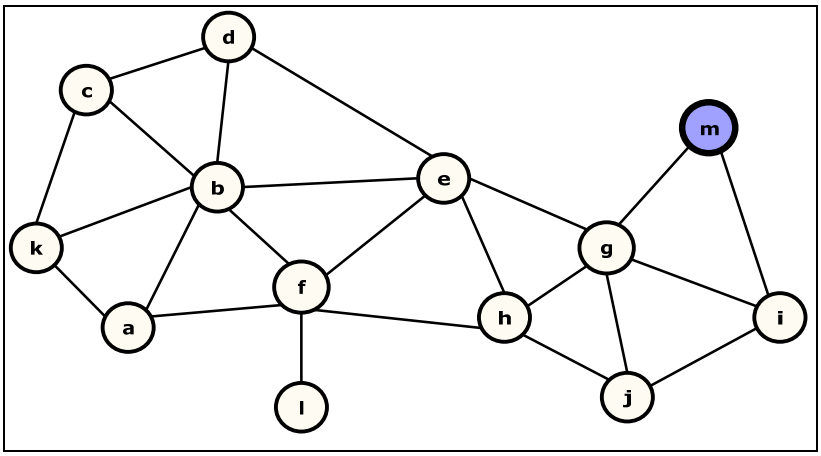

Figure 2.2 Usurpation d'identité du nœud *a* par *m* (HELLO).

Attaque sur la sélection des MPR par mystification de lien

Le cas qui nous intéresse dans ce mémoire est lorsqu'un nœud malicieux oblige ses voisins à le choisir comme relais multipoint. On appelle cette vulnérabilité *attaque sur la sélection des MPR par mystification de lien.* Ceci peut survenir lorsqu'un nœud malveillant signale dans ses messages HELLO une fausse relation de voisinage avec des nœuds inexistants ou des nœuds éloignés qui ne font pas partie de ses voisins (*Link spoofing*). Comme il est le seul à avoir un lien avec ces nœuds, la spécification du protocole OLSR, et plus précisément l'Algorithme 1.1, lui permet d'être choisi comme relais multipoint par ses voisins. Ceci entraine une fausse sélection des MPR par tous les voisins de ce nœud malicieux.

Dans la Figure 2.3, le nœud malicieux *m* annonce dans ses messages HELLO un lien avec un nœud inexistant *i*. Dans ce contexte, le nœud *a* le choisit comme MPR. Par la suite, tous les messages entre les nœuds *a* et c passent par *m* et ce dernier peut les modifier ou les rejeter. Or, dans l'absence de ce lien inexistant entre *m* et *i,* le nœud *a* peut choisir soit *b* soit *m* comme MPR.

Dans le même contexte, un nœud malveillant *m* peut créer des liens virtuels avec tous les nœuds à 2 sauts d'un nœud *a*. Ainsi, le nœud *m* est l'unique MPR de *a*. Le nœud malicieux

peut par la suite l'ignorer dans ses messages TC et le nœud *a* se retrouve alors couper du réseau. Ce type d'attaque s'appelle *Node Isolation Attack* (Kannhavong et al., 2006).

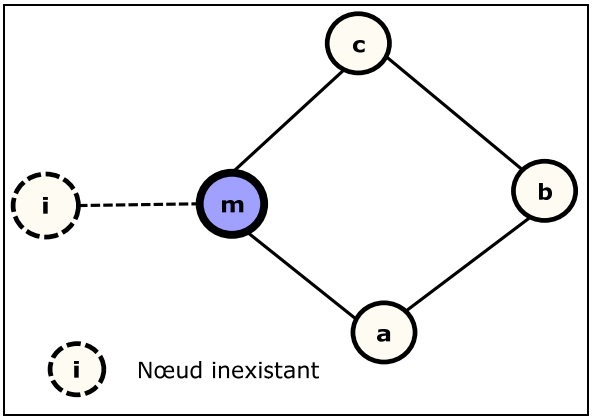

Figure 2.3 Attaque sur la sélection des MPR.

Un nœud malveillant peut aussi déclarer dans ses messages HELLO un ensemble incomplet de ses voisins. Les nœuds ignorés peuvent être coupés du reste du réseau si le nœud malveillant est leur seul lien.

Génération incorrecte du message TC

Un nœud malicieux peut envoyer des messages TC et usurper l'identité d'un autre nœud dans le réseau. Dans Figure 2.4, le nœud *m* génère des messages TC ayant pour origine le nœud *v* et déclarant les nœuds *e* et *g* comme voisins. Lorsqu'un nœud dans le réseau reçoit les messages TC (par exemple le nœud *a*), il conclut que les nœuds *e*, *g* et *v* sont voisins. Ceci entraine des conflits des routes dans le réseau et une mauvaise vision de la topologie par les nœuds.

Lorsque le nœud *a* reçoit un message TC de la part du nœud *m* et *v* (pour lui c'est de la même origine), il va rejeter le message avec le plus petit numéro de séquence ANSN (*Advertised Neighbor Sequence Number).* Le nœud malveillant doit alors générer des messages TC avec des ANSN plus grands que ceux envoyés par le nœud *v*. Ce mécanisme peut être considéré en lui-même comme un type d'attaque à part entière. En effet, il suffit pour un nœud

d'écouter les messages TC des autres nœuds légitimes et envoyer par la suite des messages TC avec usurpation d'identité des nœuds avec un numéro de séquence plus grand que celui des messages des victimes.

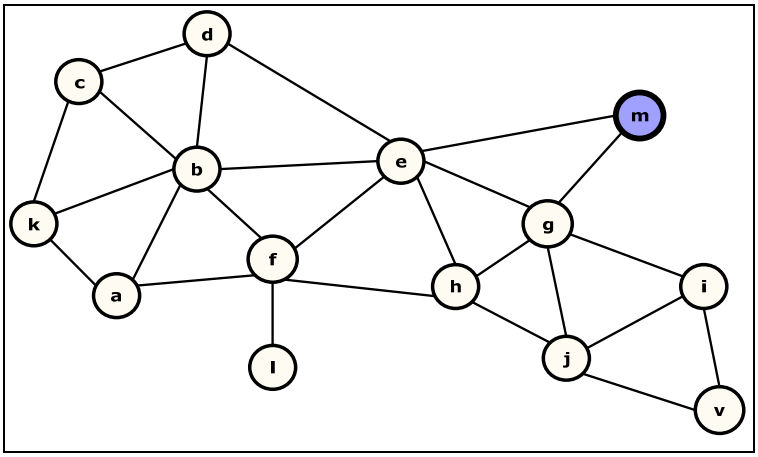

Figure 2.4 Usurpation d'identité du nœud *v* par *m* (TC).

Finalement, un nœud malveillant peut être choisi comme MPR légitime par d'autres nœuds. Par la suite, il peut refuser de générer les messages TC ou déclarer dans ses TC un ensemble incomplet de voisins qui l'ont choisi comme MPR. Ceci peut entrainer la déconnexion du réseau de l'ensemble des nœuds non déclarés dans les messages TC du MPR malveillant.

b) Relayage incorrect du trafic

Les opérations de routage et de communication dans les réseaux Ad-Hoc se basent essentiellement sur le relayage correct du trafic de routage et de données. Un relayage incorrect a des conséquences sur le bon fonctionnement du réseau. On peut distinguer différents types d'attaques dans cette catégorie.

Relayage incorrect du trafic de contrôle

Un nœud malveillant peut être choisi comme MPR légitime par d'autres nœuds mais il refuse de relayer les messages TC des autres MPR. Dans le cas où il n'existe pas de route qui ne

passe pas par le nœud malicieux, le refus de ce dernier de relayer les messages TC peut avoir comme conséquence une perte de connectivité de certains nœuds dans le réseau.

Attaque par retransmission des messages de contrôle

Un nœud malicieux peut renvoyer à d'autres nœuds des messages de contrôle (TC ou HELLO) déjà envoyés dans le passé par d'autres nœuds qu'il a pu écouter à travers le réseau. Il faut que les messages renvoyés par l'attaquant aient des numéros de séquence plus élevés sinon ces messages seront rejetés par les nœuds qui ont reçu une copie originale de ces messages. Ce type d'attaque cause un échange de fausses informations et un conflit dans le calcul de la topologie qui peut entrainer des problèmes de routage.

Attaque *wormhole*

Ce type d'attaque redirige le trafic entre deux zones géographiquement éloignées pour ainsi avoir une bonne position géographique pour contrôler le trafic qui passe par lui.

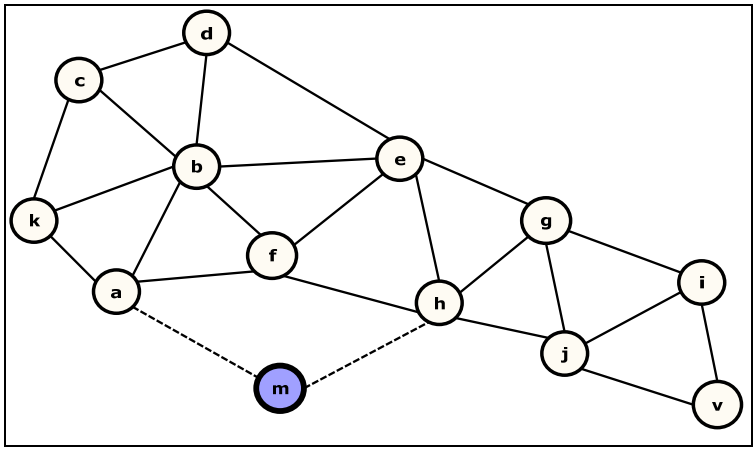

Figure 2.5 Attaque *wormhole* créée par le nœud *m*.

Dans la Figure 2.5, le nœud malicieux *m* crée un lien virtuel entre les nœuds *a* et *h* sans être visible par les deux nœuds. Le but est de leur faire croire qu'ils sont des nœuds voisins. En effet, le nœud *m* renvoie les messages HELLO du nœud *h* vers *a* et inversement. Ainsi, chacun de ces nœuds va déclarer par la suite qu'il a un lien symétrique entre eux. La route

entre *a* et *h* devient alors une route préférée par les autres nœuds car c'est le plus court chemin. Ce chemin est totalement contrôlé par le nœud malveillant *m* ce qui présente un danger pour l'intégrité et la confidentialité des messages. Deux nœuds malicieux *m1* et *m2* peuvent aussi collaborer pour créer une attaque *wormhole* entre deux zones très éloignées (*Voir* Figure 2.6).

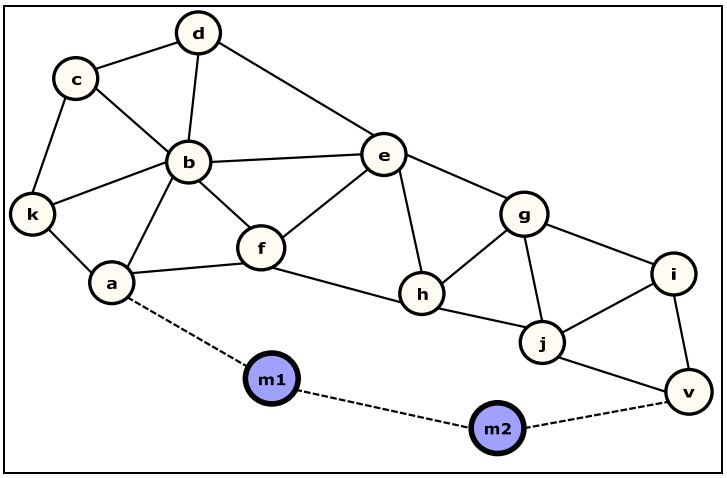

Figure 2.6 Collaboration pour créer un *wormhole.*

Attaque trou-noir

Un nœud malicieux qui a été choisi par ses voisins comme MPR peut rejeter tous les paquets de données reçus de ses voisins (*Blackhole Attack*). Ce type d'attaque entraine une perte de connectivité et la dégradation de la communication.

2.3.2 Mécanismes de sécurité proposés pour OLSR

Ces dernières années, de nombreuses contributions ont été proposées pour la sécurité du protocole OLSR (Adjih et al., 2003; Adjih et al., 2005; Adjih, Mühlethaler et Raffo, 2006; Adjih, Raffo et Mühlethaler, 2005; Dhillon et al., 2004; Raffo et al., 2004). Dans cette section, on présentera une revue des principales solutions proposées pour sécuriser le protocole OLSR.

Architecture de sécurité pour OLSR

Ce système a été proposé et étudié dans (Adjih et al., 2003; Adjih et al., 2005; Adjih, Mühlethaler et Raffo, 2006). L'objectif est d'isoler les nœuds malicieux en utilisant un système de signature et d'authentifier les messages OLSR de bout-en-bout. Ce crypto-système repose sur l'ajout d'une signature aux messages de contrôle d'OLSR. En effet, chaque nœud génère la signature lors de la création de chaque message de contrôle (HELLO/TC/HNA/MID).

La signature est envoyée par la suite avec le message de contrôle dans le même paquet. Or, il est impossible de signer le TTL (*Time To Live*) et l'indice de nombre de sauts (*Hop_Count*) présents dans les messages de contrôle. La solution pour pallier le problème et ainsi faire face aux attaques par retransmission des messages de contrôle (*Replay Attacks*), est d'utiliser un *Timestamp* dans les messages au lieu du TTL (Adjih, Mühlethaler et Raffo, 2006). Le *Timestamp* est inséré lors de la création du message en même temps que la signature. Ainsi, lorsqu'un nœud reçoit un message de contrôle, il contrôle le *Timestamp* et vérifier la signature du message. Si le *Timestamp* et la signature sont corrects, le nœud traite le message; sinon ce dernier le rejette et alors le nœud malicieux, originaire de ce message, se trouve isolé du reste du réseau (*Voir* Figure 2.7).

L'architecture PKI (*Public Key Infrastructure*) est basée sur une autorité centralisée qui est soit proactive ou réactive. La version proactive envoie périodiquement les certificats à tout le réseau. Par contre, la version réactive répond sur demande aux requêtes d'obtention de certificat par les nœuds.

Mécanisme de signature avancé pour OLSR (*ADVISG*)

Les mécanismes de signature et de *Timestamp* ne sont pas suffisants pour faire face aux différents types d'attaques. En effet, cette solution n'est pas efficace dans le cas où un nœud légitime est compromis car ce nœud malicieux peut alors générer des messages signés correctement avec son identité et ainsi envoyer de faux messages de contrôle à travers le réseau. Dans ce contexte, un mécanisme additionnel ADVSIG (*ADVanced SIGnature*) a été

proposé (Raffo et al., 2004). Le but du mécanisme ADVSIG est de garantir l'authenticité de bout-en-bout de l'ensemble des informations concernant l'état des liens à travers le réseau, ç.-à.-d., assurer l'intégrité du réseau et des messages échangés dans ce dernier. Cette approche nécessite les mécanismes de PKI et de *Timestamp* définis dans (Adjih et al., 2003; Adjih et al., 2005).

Lors de la création des informations de la topologie et durant l'échange des messages OLSR, chaque nœud attache la signature ADVSIG et doit certifier l'authenticité des informations qu'il fournit à ses voisins. Il doit aussi générer une preuve qui sera utilisée par ses voisins pour prouver l'authenticité de ce lien avec le nœud originaire du message. Ce mécanisme ne fournit pas une solution pour les attaques de type DoS ou *wormhole* et il ajoute une charge de trafic importante à cause des signatures échangées entre les nœuds.

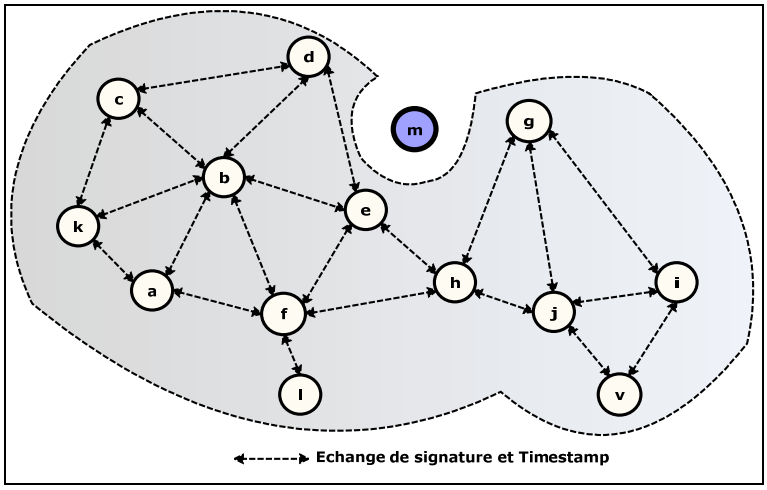

Figure 2.7 Isolation du nœud malicieux *m*.

Mécanisme basé sur la position géographique

Une modification basée sur la géo-localisation a été apportée aux mécanismes précédents (Adjih et al., 2005). Cette solution SIGLOC (*SIGnature and LOCalization*) présente une approche pour faire face aux attaques de type *wormhole* et aux attaques par mystification des liens. En effet, les trois mécanismes regroupés ensemble (PKI, *Timestamp* et la localisation

géographique) permettent de détecter tout relayage incorrect du trafic d'un point du réseau vers un autre éloigné.

Autorité de certification distribuée

Un autre mécanisme basé sur une autorité de certification entièrement distribué DCA (*Distributed Certificate Authority*) a été proposé par (Dhillon et al., 2004). Le DCA est distribué de telle manière que chaque nœud peut demander un certificat de la part de n'importe quelle coalition de k nœuds dans le réseau (*Shareholders*). Lorsqu'un nœud demande à une coalition un certificat, il doit attendre la réception de k certificats partiels avant de générer un certificat valide et de l'utiliser dans le réseau. Ce mécanisme nécessite une intervention externe pour initialiser au début et mettre en place une autorité de certificat. Ceci rend cette solution difficile à utiliser dans les réseaux Ad-Hoc auto-organisé.

Modèle de confiance implicite pour le routage dans OLSR

Une approche basée sur un modèle de confiance implicite pour le routage dans le protocole OLSR a été proposée dans (Adnane, Bidan et de Sousa Jr, 2008) et (Adnane et al., 2008). Le but est de fournir une extension au protocole OLSR basée sur la confiance entre les nœuds du réseau afin de renforcer le mécanisme de sélection des MPR et réduire les attaques contre ce protocole. Pour ce faire, ce mécanisme procède à une validation de la table de routage et des informations de routage en utilisant un modèle de confiance entre les nœuds. Le mécanisme suit les quatre étapes d'initialisation du protocole OLSR : découverte des voisins, sélection des MPR, signalisation des MPR et comptage de table de routage.

La réception des messages HELLO permet à chaque nœud x du réseau de commencer à construire sa liste de voisins. Avant qu'il fasse confiance à un de ses voisins, il doit s'assurer que ce dernier opère selon les spécifications du protocole OLSR. Le nœud x diffuse alors un message HELLO à son voisin et devient un nœud de confiance pour tous ses voisins (Marsh, 1994). Si ce voisin opère selon les spécifications du protocole, il va émettre un message HELLO pour déclarer qu'il a un lien avec le nœud x. Par la suite, le nœud x l'ajoute comme voisin symétrique avec qui une relation de confiance est établie. Le calcul des MPR se base

sur les informations de voisinage fournit par les nœuds, or ces informations sont basées sur un modèle de confiance. Le choix des MPR se fait alors parmi les nœuds avec qui une relation de confiance a été établie. Cette même relation de confiance détermine le choix de route pour acheminer les messages car chaque MPR suggère aux nœuds qui l'ont choisi comme MPR le plus court chemin vers la destination qui passent par les nœuds à qui il fait confiance.

Mécanisme de réputation basé sur la contre-réaction

Une approche de réputation basée sur la contre-réaction pour faire face aux attaques de type mystification de lien a été proposée dans (Vilela et Barros, 2007). L'objectif de ce mécanisme est de s'assurer que chaque nœud génère correctement les messages de contrôle. Pour ce faire, deux opérations sont utilisées : le message de contre-réaction et la table d'évaluation. Le message de contre-réaction est utilisé pour indiquer le chemin parcouru par le message de contrôle TC. Quand un MPR reçoit un message TC, il envoie, au nœud originaire du message TC, un message de contre-réaction indiquant le chemin traversé par le message de contrôle. D'autre part, une table d'évaluation est maintenue par chaque nœud dans le réseau afin d'évaluer et classifier les autres nœuds selon leur réputation de transmission correcte des messages de contrôle. Pour chaque nœud dans le réseau, deux valeurs sont stockées dans la table de réputation et qui correspondent à la bonne et mauvaise réputation. La réputation est générée grâce à un mécanisme de surveillance (*watchdog*). Lorsqu'un nœud est déclaré comme nœud qui génère des fausses informations de routage, sa réputation va être diminuée par un indice PV (*Punishment Value*). Dans le cas contraire, sa valeur de bonne réputation augmente.

La détection de la génération incorrecte des messages HELLO par un nœud se base sur la corrélation de l'information sur le chemin traversé par le message (obtenu grâce au message de contre-réaction) et les informations obtenues à partir des échanges locaux des messages HELLO. D'autre part, la détection de la génération incorrecte des messages TC est obtenue grâce à la corrélation de l'information sur le chemin obtenu grâce au message de contre-mesure et l'information locale obtenue grâce aux messages TC. Ce mécanisme de réputation

nécessite la mise en place tout au début de solutions cryptographique basées sur les signatures PKI et les *Timestamp* données dans (Adjih, Raffo et Mühlethaler, 2005). De plus, il n'est pas souhaitable pour les réseaux Ad-Hoc à grande mobilité.

Détection de mystification de lien qui cause l'isolation d'un nœud du réseau

Certains types d'attaques peuvent avoir pour conséquence l'isolation d'un nœud du reste du réseau. Une approche a été proposée pour détecter la mystification de lien par un nœud malicieux en ajoutant la liste des voisins à 2-sauts aux messages HELLO (Kannhavong, Nakayama et Jamalipour, 2006; Kannhavong et al., 2006). L'idée de ce mécanisme est de détecter le faux lien en comparant les informations contenues dans les messages HELLO de chaque voisin. Le principal inconvénient de cette approche est qu'elle augmente la taille des messages HELLO. Or ces messages sont périodiques et essentiels à la mise à jour de la topologie. Ceci surcharge la taille du trafic encore plus.

Schéma de détection d'attaque *wormhole* dans le protocole OLSR

Une nouvelle approche a été proposée pour détecter les attaques *wormhole* dans les réseaux Ad-Hoc utilisant le protocole de routage OLSR (Nait-Abdesselam, Bensaou et Yoo, 2007). Cette approche se base sur deux nouveaux messages de contrôle $HELLO_{req}$ et $HELLO_{rep}$. Ces messages sont ajoutés dans le but de détecter les liens suspects en comparant le délai entre le temps d'envoi du message $HELLO_{req}$ et celui de la réception du message $HELLO_{rep}$. Mais ce mécanisme ne présente pas une solution complète car il est vulnérable aux attaques par mystification de lien.

Modèle de confiance

TARP (*Trust-Aware Routing Protocol*) est un protocole de sécurité pour les réseaux Ad-Hoc basé sur la confiance (*Trustworthiness*) pour établir des chemins optimaux entre les nœuds (Abusalah, Khokhar et Guizani, 2006). Ce protocole est basé sur trois nouveaux concepts. Le premier concept utilise un nouveau mécanisme de routage sécuritaire pour calculer les routes. Le deuxième concept utilise six paramètres (énergie de la batterie, configuration logiciel, configuration matérielle, historique d'événements, l'organisation hiérarchique et

l'exposition) pour définir le niveau de confiance d'un nœud. Finalement, le troisième mécanisme est basé sur la réputation et la métrique de confiance. Chaque nœud doit évaluer uniquement le niveau de confiance de ses voisins en se basant sur ces paramètres.

En conclusion, les extensions de sécurité proposées pour OLSR couvrent un grand nombre de problèmes distincts. Une grande partie se base sur des mécanismes pour garantir l'intégrité et authentifier le nœud originaire du trafic de contrôle. L'introduction de *Timestamps* a permis la limitation et la détection des attaques qui utilisent les anciens messages pour les envoyer sur le réseau. De nombreux travaux de recherche s'ajoutent à ces solutions. Mais toutes ces solutions restent des mécanismes partiels et jusqu'à aujourd'hui, il n'existe pas de solution complète pour faire face à toutes les vulnérabilités du protocole OLSR.

CHAPITRE 3

LE PROTOCOLE SU-OLSR

Dans ce chapitre, on introduira un nouveau protocole basé sur OLSR pour faire face aux attaques par mystification de lien et ainsi réduire l'impact des nœuds malicieux qui forcent leurs voisins à les choisir comme MPR. On présentera le fonctionnement global de notre solution et le nouvel algorithme de sélection des MPR.

3.1 Motivations et objectifs

Les MPR jouent un rôle important dans la diffusion optimisée des messages dans le cas du protocole OLSR. Or, ces types de nœuds sont des points névralgiques du réseau et une défiance de ces nœuds aurait un impact très important sur le fonctionnement global du réseau. En effet, un nœud malicieux qui a été sélectionné comme MPR par ses voisins, aurait un contrôle complet sur tous les messages qui passent par lui. Il pourrait ainsi les modifier, les altérer ou rejeter. Un nœud malicieux pourrait même forcer ses voisins à le choisir comme MPR, par exemple, en utilisant des techniques de mystification de lien (*Voir* la section 2.3.1).

Dans ce contexte, le but de notre approche est de proposer :

- Une nouvelle solution pour faire face aux attaques de type mystification de lien où un nœud malicieux force ses voisins à le choisir comme MPR.

On introduit ainsi le protocole SU-OLSR (SUspicious-OLSR) qui propose une solution préventive et réactive en même temps au lieu des solutions réactives proposées dans la littérature. SU-OLSR utilise une nouvelle heuristique de sélection des MPR basée sur la confiance envers les nœuds voisins. Or, les heuristiques de sélection des MPR présentées dans la littérature et qu'on présentera dans la section 3.2, ont pour but d'améliorer l'inondation des messages de contrôle ou fournir une bonne qualité de service. Notre heuristique de sélection pourrait être classée dans une toute nouvelle catégorie, la sélection des MPR pour garantir la sécurité et la prévention des attaques.

3.2 Revue de littérature des heuristiques de sélection des MPR

Les schémas de sélection des MPR présentés dans la littérature peuvent être classifiés en trois catégories.

3.2.1 Schémas classiques

Les heuristiques proposées dans ce groupe sont des schémas basés sur l'heuristique originale proposée dans (Clausen et Jacquet, 2003) . Le but est d'améliorer certaines performances relatives à l'impact des MPR sur le réseau. Par exemple, l'optimisation des choix des MPR, la réduction du nombre de collisions dues au nombre élevé des MPR choisis ou la réduction de l'énergie utilisée par les nœuds.

Quatre approches ont été proposées dans (Mans et Shrestha, 2004; Shrestha, 2003) soit pour réduire le nombre de MPR et limiter la collision dans le réseau ou soit pour réduire la complexité de l'algorithme de sélection des MPR.

1) ***In degree Greedy set cover***

L'objectif de cette approche est de maximiser le nombre de MPR choisis afin de réduire la complexité de calcul des MPR et d'augmenter la rapidité de l'algorithme de sélection. Cet algorithme utilise la même approche que l'heuristique originale pour couvrir les nœuds isolés. Dans la deuxième phase, tant qu'il reste des nœuds non couverts dans $N_2(S)$, l'algorithme prend au hasard un de ces nœuds (appelons ce nœud *y*). Il cherche alors un nœud *x* dans $N_1(S)$ tel que $y \in N_1(x)$ et *x* est adjacent au minimum de nœuds non-couverts de $N_2(S)$.

2) Sélection avec minimum de chevauchement

L'heuristique originale ne prend pas en compte les problèmes de collisions lors du calcul des MPR. Le but de cette approche est de distribuer les MPR sélectionnés à l'entour du nœud source pour limiter le nombre de collision dans le réseau. La première phase de sélection des

MPR couvrant les nœuds isolés est la même que le schéma original. Lorsqu'il reste des nœuds non couverts à 2-sauts d'un nœud S, on sélectionne comme MPR un nœud x dans $N_1(S)$ qui couvre le minimum des nœuds dans $N_2(S)$ qui ne sont pas encore couverts par un MPR.

3) Sélection avec priorité secondaire

Le but de cet algorithme de sélection est de réduire le degré de chevauchement des MPR sans diminuer leur nombre et ainsi limiter le nombre de collision dans le réseau. La première phase de sélection des MPR de S couvrant les nœuds isolés est la même que le schéma original. Dans la deuxième phase, lorsqu'il reste plusieurs nœuds dans $N_1(S)$ qui couvrent le même nombre de nœuds dans $N_2(S)$, l'algorithme choisi comme MPR le nœud dans $N_1(S)$ qui a le minimum de voisins dans $N_2(S)$.

4) Sélection aléatoire

Une variante propose de choisir de manière aléatoire un MPR parmi les nœuds qui couvrent le même nombre de nœuds dans $N_2(S)$.

Pour terminer, une approche différente a été proposée par (Lipman et al., 2003; Lipman, Boustead et Judge, 2002). Elle consiste à chercher à minimiser le cheminement des messages et de prendre en compte certaines caractéristiques des nœuds (énergie, utilité par rapport aux voisins) lors du calcul des MPR.

3.2.2 Schémas basés sur des ensembles dominants connectés

Une autre approche est de réduire le nombre de nœuds qui participent au cheminement des messages en construisant des ensembles dominants connexes de MPR, voir (Matousek, Nesetril et Hachez, 2004). Deux heuristiques utilisent cette approche.

Une première heuristique a été proposée pour calculer des ensembles dominants connectés de MPR (Adjih, Jacquet et Viennot, 2002). L'ensemble dominant est calculé à partir des MPR

obtenus grâce à l'algorithme original de sélection des MPR. En d'autres termes, il faut d'abord calculer l'ensemble des MPR en utilisant l'algorithme classique avant de procéder au calcul des ensembles dominants connectés des MPR.

Une amélioration a été apportée à l'heuristique précédente pour générer des petits ensembles dominants connectés de MPR (Wu, Wei et Dai, 2006). Cette amélioration se base sur l'algorithme original de sélection des MPR mais utilise les informations à 3-sauts d'un nœud pour calculer les MPR qui couvrent le plus de nœuds à 2-sauts.

3.2.3 Schémas basés sur la qualité de service

Des heuristiques ont été proposées pour prendre en compte la qualité de service (QoS) ou en d'autres termes, sélectionner des MPR qui garantissent une certaine QoS. Ce genre d'approche sera utile dans les réseaux où la qualité de service est primordiale (VoIP, vidéo, applications qui nécessitent des petits délais, etc).

Deux approches ont été proposées pour faire face à la limitation de l'heuristique originale en terme de QoS dans (Badis et al., 2004). Le but de ces approches est de sélectionner les MPR en se basant sur des métriques de QoS (délai, bande passante). Les nœuds qui offrent une large bande passante ou un minimum de délai seront privilégiés pour être choisis comme MPR lors du mécanisme de sélection des MPR.

Finalement, une approche a été proposée dans (Ge, Kunz et Lamont, 2003) pour garantir que tous les nœuds à deux-sauts d'un nœud source ont un chemin optimal en termes de bande passante vers le nœud source. Pour tout nœud à 2-sauts d'un nœud S, le nœud dans $N_1(S)$ et qui offre la plus large bande passante vers S est choisi comme MPR. De cette manière on couvre tous les nœuds dans $N_2(S)$ par des MPR qui offrent une large bande passante.

3.3 Le nouveau protocole SU-OLSR

Comme on l'a précisé précédemment, le but de notre approche est de proposer une solution pour faire face aux attaques sur la sélection des MPR par mystification de lien (*Voir* la section 2.3.1).

Pour se faire, on introduit le concept de confiance entre les nœuds voisins. En effet, chaque nœud ne doit pas faire confiance à un nœud voisin X qui présente des caractéristiques malicieuses et qui pourraient influencer le choix des MPR. On a appelé notre protocole SU-OLSR (SU pour *SUspicious nodes* ou nœuds suspects).

Il est important de définir certaines terminologies qu'on utilisera tout au long de ce mémoire. Soit S un nœud donné du réseau. On dit qu'un nœud $X \in N_1(S)$ satisfait le critère I ou le critère II si :

- ***Critère I*** **:** Le nœud X couvre un ou plusieurs nœud isolés β dans $N_2(S)$.
- ***Critère II*** **:** Le nœud X couvre plus qu'une fraction fixé de nœuds dans $N_2(S)$.

Un nœud est dit suspect s'il présente des comportements suspects selon l'un de ces critères. Pour chaque nœud S, on définit aussi l'ensemble des nœuds X qui présentent un comportement suspect par rapport à S de la manière suivante :

$$Suspects(S) = \{X \in N_1(S) | X \text{ est suspect}\}.$$

3.3.1 Le nouveau algorithme de sélection des MPR

Le protocole SU-OLSR utilise un nouvel algorithme de sélection des MPR basé en partie sur l'algorithme original (Clausen et Jacquet, 2003) décrit dans 1.5.3 et sur les critères I et II. L'objectif de cet algorithme est de déterminer à la fois l'ensemble des nœuds suspects selon nos critères I et II et aussi l'ensemble des MPR de confiance.

L'Algorithme 3.1 décrit l'heuristique de sélection des MPR pour le protocole SU-OLSR. Pour tout nœud *S* donné dans le réseau, on commence d'abord par trouver tous ses voisins à 1- saut et à 2-sauts ($N_1(S)$ et $N_2(S)$). On recherche par la suite tous les nœuds *X* dans $N_1(S)$ qui démontrent un comportement suspect selon nos critères I ou II. Si c'est le cas, on ajoute les nœuds trouvés à l'ensemble $Suspects(S)$. L'étape suivante de l'algorithme est de redéfinir l'ensemble des voisins à 1-saut de S $N^*_1(S) = \{y \in N_1(S) \setminus Suspects(S)\}$ ainsi que l'ensemble des voisins à 2-sauts basés sur l'ensemble $N^*_1(S)$ et que l'on définit par :

$$N^*_2(S) = \left\{ z \mid z \neq S \wedge z \notin N^*_1(S) \wedge (\exists y \in N^*_1(S)) \left[z \in N_1(y) \right] \right\}$$

À partir des ensembles $N^*_1(S)$ et $N^*_2(S)$, on ajoute à l'ensemble des MPR chaque nœud dans $N^*_1(S)$ qui couvre un nœud isolé dans $N^*_2(S)$ et on élimine par la suite ses nœuds isolés de $N^*_2(S)$ ainsi que les nœuds couverts par l'un des MPR choisi dans cette étape. Tant que tous les nœuds dans $N^*_2(S)$ ne sont pas tous couverts, on ajoute à l'ensemble des MPR un nœud de $N^*_1(S)$ qui couvre le maximum de nœuds dans $N^*_2(S)$. De cette manière, l'ensemble des MPR de confiance du nœud *S* ainsi que l'ensemble des nœuds suspects sont calculés selon l'algorithme relatif à SU-OLSR. On définit aussi l'ensemble des voisins qui ont déclaré un nœud *z* digne de confiance par $Sel_{MPR}(z) = \{y \in N_1(z) | z \in MPR(y)\}$. De même, l'ensemble des voisins d'un nœud *z* qui l'ont déclaré non sécuritaire est défini par :

$$Sel_{Suspects}(z) = \{y \in N_1(z) \mid z \in Suspects(y)\}.$$

Données : Tout nœud s avec ses voisins $N_1(s)$ et $N_2(s)$.
Résultat : Les ensembles $MPR(s)$ et $Suspects(s)$.

début
  $Suspects(s) \leftarrow \emptyset$;
  pour *tout nœud* x *dans* $N_1(s)$ **faire**
    si x *démontre le critère choisi* **alors**
      Ajouter x à $Suspects(s)$;
    fin
  fin
  $N_1^*(s) \leftarrow N_1(s) \backslash Suspects(s)$;
  $N_2^*(s) \leftarrow$ voisins à $2-$sauts basés sur $N_1^*(s)$;
  $MPR(s) \leftarrow \emptyset$;
  pour *tout nœud* y *dans* $N_2^*(s)$ *isolé* **faire**
    Soit $x \in N_1^*(s)$ le seul voisin de ce nœud y;
    Ajouter x à $MPR(s)$;
    Éliminer tous les nœuds dans $N_2^*(s)$ couverts par x;
  fin
  tant que $N_2^*(s) \neq \emptyset$ **faire**
    Trouver $x \in N_1^*(s)$ tq.
    - x couvre le maximum des nœuds dans $N_2^*(s)$;
    - x a le maximum des voisins ;

    Ajouter x à $MPR(s)$;
    Éliminer tous les nœuds dans $N_2^*(s)$ couverts par x;
  fin
fin

Algorithme 3.1 Sélection des MPR par SU-OLSR.

La Figure 3.1 donne une comparaison entre l'algorithme de sélection des MPR relatif à SU-OLSR et l'heuristique classique définie dans (Clausen et Jacquet, 2003). Les nœuds 1, 3 et 7 sont déclarés non sécuritaires par le protocole SU-OLSR.

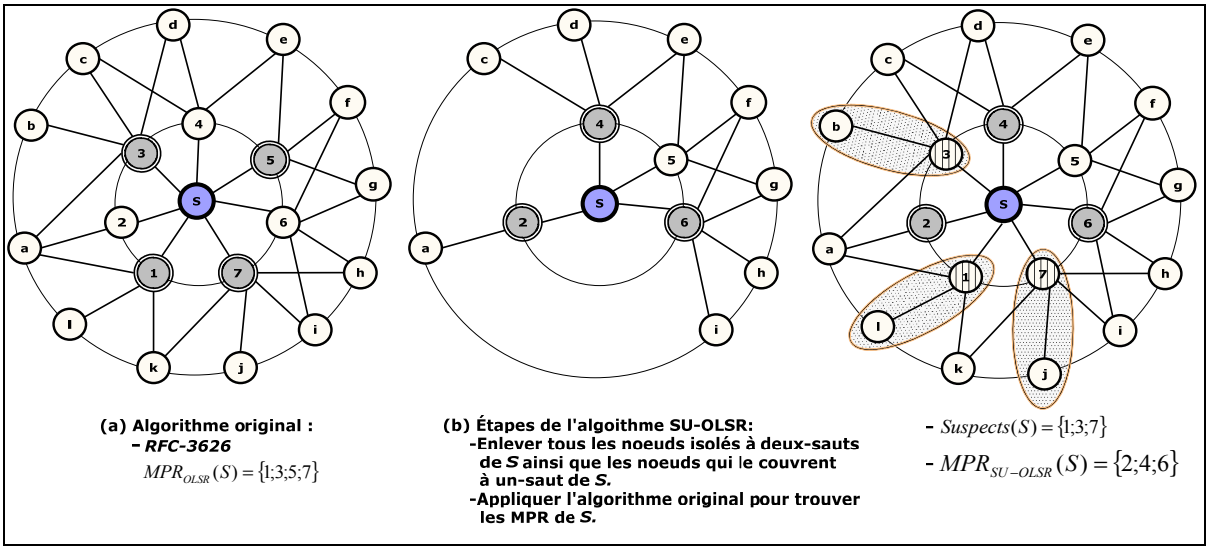

Figure 3.1 Application de l'algorithme de sélection des MPR de SU-OLSR.

L'algorithme de sélection des MPR de SU-OLSR se base sur le choix des nœuds de confiance. Certains nœuds légitimes pourraient présenter des caractéristiques du critère I (exemple d'un nœud légitime qui couvre un nœud isolé légitime). Ils pourraient être ainsi déclarés comme non sécuritaires. D'autre part, certains nœuds dans $N_2(S)$ ne pourraient plus être rejoints. Dans la Figure 3.1, le nœud *k* représente un dommage collatéral de l'heuristique de sélection SU-OLSR car les seuls nœuds dans $N_1(S)$ qui le couvrent sont déclarés suspects par le nœud *S*.

Tout ceci influencera les chemins entre les nœuds légitimes dans le réseau. Mais avec un réseau qui présente une certaine densité, le graphe de la topologie sera connexe. Ce point sera détaillé dans la section qui suit.

3.3.2 Messages de contrôle et algorithme d'inondation dans SU-OLSR

Une fois que l'ensemble des MPR de confiance a été calculé ainsi que l'ensemble des suspects à travers le réseau, il faut définir un mécanisme pour diffuser les informations de la topologie à travers le réseau.

Le protocole SU-OLSR utilise les mêmes mécanismes d'inondation que le protocole classique OLSR et qui sont donnés dans (Clausen et Jacquet, 2003) . Les seuls changements concernent les formats et le contenu des messages HELLO et TC. On doit modifier les messages de contrôle pour qu'ils prennent en considération les informations concernant le mécanisme de confiance. En effet dans SU-OLSR, il faut que chaque nœud *S* dans le réseau donne dans ses messages HELLO les MPR de confiance qu'il a choisis ainsi que les nœuds suspects dans ses voisins. De la même manière, il faut que chaque nœud émettant des messages TC déclare dans ses messages les nœuds qui l'ont choisi comme MPR de confiance et aussi les nœuds qui l'ont déclaré comme suspect. Ceci revient à diffuser les ensembles $Sel_{MPR}(S)$ et $Sel_{Suspects}(S)$ dans les messages TC d'un nœud *S*. Ce changement dans la forme des messages de contrôle n'a aucun impact sur la charge du trafic dans le réseau en comparaison avec le protocole classique OLSR.

Du moment où les informations de la topologie sont échangées entre les nœuds selon les nouvelles spécifications des messages de contrôle de SU-OLSR, chaque nœud devrait procéder au calcul des plus courts chemins vers une destination donnée en utilisant l'algorithme de Dijkstra. Deux options s'offrent à chaque nœud pour calculer ses chemins, on les définit de la manière suivante :

- ***Option I* :** Utiliser pour calculer ses routes justes les MPR déclarés sécuritaires par tous les nœuds dans le réseau.
- ***Option II* :** Utiliser pour calculer ses routes les MPR déclarés sécuritaires par certains nœuds dans le réseau.

En d'autres termes, on pourrait classifier les MPR utilisés par les nœuds dans SU-OLSR pour calculer les chemins et acheminer les messages en deux catégories : Les MPR avec confiance totale et les MPR avec confiance partielle définie par :

$$MPR_{Total} = \left\{x \mid Sel_{MPR}(x) \neq \varnothing \wedge Sel_{Suspects}(x) = \varnothing\right\} \text{ et } MPR_{Partiel} = \left\{x \mid Sel_{MPR}(x) \neq \varnothing\right\}$$

Le calcul des chemins, en prenant en compte soit l'*Option I* soit l'*Option II*, aura un effet sur la longueur des routes et sur la nature du graphe des MPR. Ce point sera regardé attentivement dans la partie expérimentale du protocole SU-OLSR (*Voir* la section 4.3).

3.4 Analyse du modèle d'attaque

3.4.1 Hypothèses et limitations

La mise en place du protocole SU-OLSR nécessite trois hypothèses pour garantir l'efficacité de notre solution :

H-1. Chaque nœud légitime pourrait signer ses messages de contrôle avec sa clé cryptographique prédéfinie (Adjih et al., 2003).

Cette hypothèse a pour but de limiter les attaques Sybil.

H-2. Il n'y a pas de collaboration entre les nœuds malicieux.

Dans le cas où deux nœuds malicieux collaborent dans le réseau, ils peuvent déclarer qu'ils couvrent le même nœud inexistant dans le réseau pour être certain que l'un d'entre eux soit choisi comme MPR.

H-3. Tous les nœuds dans le réseau sont équipés d'une seule interface réseau sans fil.

Dans le cas où un nœud malicieux possède deux interfaces réseau sans fil, il peut déclarer par chacune de ses interfaces qu'il couvre un nœud inexistant et les deux interfaces utilisées par ce nœud sont vues par les autres nœuds comme deux nœuds distincts. Ainsi il est certain qu'il va être choisi comme MPR par l'un de ses voisins.

3.4.2 Modèle d'attaque

Grâce au protocole SU-OLSR, un nœud malicieux qui déclare qu'il couvre un nœud isolé ou lointain non connu par les autres voisins, ne sera jamais choisi comme MPR. Les seules possibilités qui lui restent sont de mentir sur son réel statut dans le réseau. Afin d'évaluer

SU-OLSR à la présence de nœud malicieux, on suppose que les nœuds peuvent mentir sur leur statut dans le réseau.

Comme premier cas, on suppose qu'un nœud malicieux *m* déclare qu'il a été choisi par un de ses voisins *x* comme nœud non sécuritaire (ç.-à.-d, $x \in Sel_{Suspectes}(m)$). Dans ce cas, le nœud malicieux ne bénéficie d'aucune position avantageuse dans le réseau car il est déclaré par les autres nœuds comme non sécuritaire pour relayer leurs messages.

Dans un deuxième cas, on suppose qu'un nœud malicieux *m* déclare qu'il a été choisi comme MPR de confiance par un nœud *x* (ç.-à.-d, $x \in Sel_{MPR}(m)$). Or, dans le cas où *x* est l'un des voisins de *m* ou s'il est dans la même partie connexe du graphe du réseau, le nœud *x* devrait recevoir les messages TC générés par le nœud malicieux et dans lesquels ce dernier déclare que *x* l'a choisi comme MPR de confiance. À la réception de ces messages TC, le nœud *x* pourrait initier un mécanisme de contre-mesure pour dénoncer le nœud malicieux auprès des autres nœuds. Le mécanisme de contre-mesure présenté dans (Vilela et Barros, 2007) pourrait être modifié et utilisé dans ce contexte. La mise en place de ce mécanisme ne sera pas traiter dans ce travail et il sera laisser comme un futur axe recherche.

Par contre, il est impossible de détecter cette attaque si le nœud *x* et le nœud malicieux ne se trouvent pas dans la même partie connexe (ou *x* est inexistant dans le réseau). En effet, comme *x* ne se trouve pas dans le graphe connexe de *m*, il ne va jamais recevoir des messages de contrôle TC de *m*. Ainsi, il sera impossible de mettre en place un mécanisme de contre-mesure pour dénoncer le nœud malicieux. Ceci représente une limitation à notre solution dans le cas où le graphe du réseau est non connexe. Pour remédier à ce point, nous allons supposer que notre réseau est suffisamment dense pour que le graphe du réseau soit connexe. Dans ce cas, l'efficacité d'un mécanisme de contre-mesure est garantie contre l'attaque précédente. D'une manière générale, si le graphe du réseau est $k+1$ connexe et à la présence de k nœuds malicieux, on peut toujours trouver un chemin qui relie deux nœuds légitimes et qui ne passent pas par l'un des nœuds malicieux. Cette proposition découle d'un théorème présenté dans (Penrose, 2003) : *Si la densité des nœuds dans le réseau est*

suffisamment grande pour avoir un graphe $(k+1)-$*connexe*, *alors si on enlève* k *nœuds du réseau le graphe reste toujours connexe.*

CHAPITRE 4

ÉVALUATION EXPÉRIMENTALE DE SU-OLSR

4.1 Problématique et objectifs

Dans SU-OLSR, tout nœud ayant un comportement suspect selon nos deux critères ne peut pas être sélectionné comme MPR par l'algorithme du SU-OLSR. Or, certains nœuds légitimes risquent de ne pas être sélectionnés comme MPR par le protocole. Ceci aurait un impact sur la connectivité du réseau.

Nous allons analyser et comparer les performances du SU-OLSR et OLSR dans le cas d'un réseau avec nœuds fixes grâce à un programme développé en *C* et, finalement dans le cas d'un réseau dynamique grâce à *ns-2*.

4.2 Paramètres d'évaluation

Délai de bout-en-bout :
Il est donné par le délai de transmission plus le délai de propagation. En d'autres termes, c'est le temps qu'un paquet met entre la source et la destination. Un protocole a de meilleurs performances s'il garantit un petit délai de bout-en-bout.

Pourcentage de paquet délivré :
Le pourcentage de paquet délivré PDR (*Packet Delivery Ratio*) est une métrique qui permet de calculer le nombre total de paquets délivrés à une destination par rapport au nombre de paquets envoyés.

Nombre de sauts :
Le nombre de sauts qu'un paquet prend pour aller d'un nœud source à la destination. Ce paramètre permet de déterminer les performances des protocoles par rapport aux chemins optimaux trouvés.

Effet de la densité :

Pour chaque simulation, on génère un ensemble N de 100 nœuds distribués aléatoirement dans un carré $1000m\,x\,1000m$. Pour un rayon de communication R donné, cet ensemble de points définit un graphe non orienté $G(R) = \langle N, E_R \rangle$, avec :

$$E_R = \{(a,b) \mid a,b \in N \wedge d_2(a,b) \leq R\}$$

Où d_2 est la distance Euclidienne entre deux points dans le plan.

Les propriétés de ce type de graphe ont été étudiées dans la théorie des graphes géométriques aléatoires (Penrose, 2003). Le résultat le plus intéressant dans cette théorie est qu'un graphe composé de n nœuds est connexe avec une probabilité maximale si le rayon de communication de ses nœuds est au moins égale à :

$\sqrt{\frac{\ln n + O(1)}{\pi n}}$ (J. Diaz, Mitsche et Pérez-Giménez, 2008; Penrose, 1999).

Ainsi, un graphe est connexe avec une probabilité $1 - \frac{1}{s}$ si le rayon de communication des nœuds est supérieur à :

$\sqrt{\frac{\ln n + \ln s}{\pi n}}$ (M. Barbeau et Kranakis, 2007).

Donc dans notre cas, le graphe $G(r)$ est connexe avec une probabilité supérieure à 99% si le rayon de communication des nœuds *r* est supérieur à 171 *m*. Ainsi, durant nos simulations, nous avons remarqué que si le rayon de communication des nœuds est supérieur ou égal à 190 *m*, le graphe est généralement connexe ou possède très peu de nœuds isolés. C'est pour cette raison que dans toutes nos simulations on utilisera des rayons supérieurs ou égaux à 190 *m* afin d'assurer une bonne connectivité entre les nœuds.

4.3 Modèle sans mobilité

4.3.1 Objectifs et implémentation du SU-OLSR et OLSR

Notre but dans cette partie est l'évaluation de notre protocole sous un environnement simple où tous les nœuds sont fixes. Notre évaluation passe par le calcul du nombre de MPR pour les deux protocoles, le nombre de messages TC générés par les nœuds et enfin le calcul du plus court chemin entre les nœuds dans le réseau.

Pour se faire, on simule les deux protocoles SU-OLSR et OLSR grâce à un programme développé en langage C. Le but est d'avoir une plateforme de simulation pour le corps des deux protocoles (algorithmes de sélection et l'algorithme du plus court chemin).

Notre programme permet de simuler les protocoles SU-OLSR et OLSR classique dans une superficie de $1\,km^2$ où tous les nœuds sont fixes. Les performances des deux protocoles sont étudiées sous différents rayons de communication. Les résultats de chaque expérimentation est le résultat d'une moyenne de 50 simulations indépendantes. Ceci permet d'obtenir un intervalle de confiance plus élevé. Dans chaque simulation, les positions des nœuds dans la topologie sont générées d'une manière aléatoirement et indépendante.

Le Tableau 4.1 donne un sommaire des paramètres des simulations statiques.

Tableau 4.1 Simulations statiques

Scénario	Topologie	Rayons de communication	Répétition
Statique	$1000\,m\ x\ 1000\,m$	190, 250, 290, 330, 390 m	50 simulations par rayon
Nombre de simulations par protocole			$50 * 5 = 250$ simulations

4.3.2 Résultats de simulation

Nombre de MPR

Le nouveau protocole SU-OLSR et son algorithme de sélection ont un impact sur le nombre de MPR dans le réseau. Il est important alors de comparer le nombre de MPR sélectionnés par les deux protocoles SU-OLSR et OLSR classique.

Durant nos simulations en mode statique du protocole OLSR, nous avons remarqué qu'un grand nombre de MPR sont sélectionnés dans la première phase de l'algorithme de sélection. Le Tableau 4.2 donne (1) le nombre total des nœuds qui ont choisi un MPR qui couvre un nœud isolé, (2) le nombre total des MPR qui couvrent des nœuds isolés, (3) le nombre moyen par nœud des MPR qui courent des nœuds isolés. Ces résultats montent que plus de 75% des MPR sont sélectionnés dans la première phase de l'algorithme. Ceci confirme les résultats présentés dans (Busson, Mitton et Fleury, 2005). Or, cette phase est critique pour le nouvel algorithme de sélection de MPR du protocole SU-OLSR. En effet, certains nœuds légitimes risquent de ne pas être choisis comme MPR dans le cas où ils couvrent un ou plusieurs nœuds isolés (*Critère I*).

Tableau 4.2 Nombre de MPR couvrant des nœuds isolés

Rayons de communication	(1)	(2)	(3)
190 *m*	96.0	48.8	2.41
290 *m*	97.5	44.1	2.48
390 *m*	90.7	33.3	1.85
490 *m*	56.8	17.6	0.89
590 *m*	14.0	4.7	0.15
690 *m*	0.8	0.3	-

La Figure 4.1 donne, en fonction du rayon de communication, le nombre moyen de MPR sélectionnés par le protocole SU-OLSR avec le *Critère I* (*Option I* et *Option II*) et le

protocole OLSR classique. Cette figure montre que dans le cas de l'*Option I*, le nombre de MPR sélectionnés par SU-OLSR est très inférieur au nombre de MPR sélectionnés par le protocole OLSR classique. Ceci n'est pas surprenant puisque l'algorithme de SU-OLSR ne peut choisir que les MPR déclarés sécuritaires par tous les nœuds.

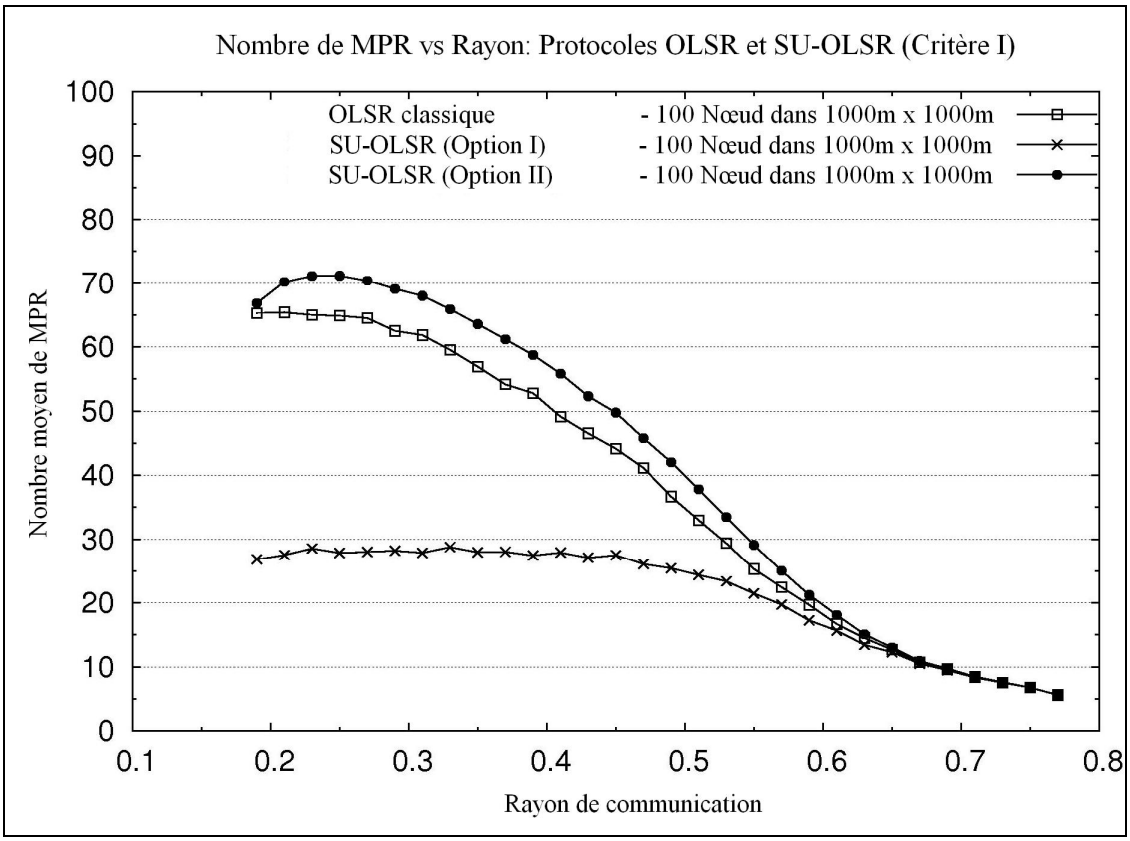

Figure 4.1 Comparaison du nombre de MPR dans le cas du *Critère I.*

Dans le cas de l'*Option II*, le nombre de MPR sélectionnés par SU-OLSR est supérieur au nombre de MPR choisis par OLSR. Dans ce cas, le nombre élevé de MPR n'implique pas forcement que le graphe qui représente la topologie est plus connexe que celui obtenu dans le cas du protocole OLSR. Il peut y avoir des pairs de nœuds qui ne sont plus connectés lors de l'utilisation de SU-OLSR. En effet, si un nœud X a été choisi comme un MPR sécuritaire par un nœud a et un MPR non sécuritaire par un autre nœud b, alors seulement l'arc orienté $X \rightarrow a$ est ajouté au graphe qui est utilisé par l'algorithme Dijkstra pour le calcul du plus

court chemin. Alors que dans le cas du protocole OLSR, les deux arcs orientés $X \to a$ et $X \to b$ sont ajoutés.

Dans le cas du *Critère II*, il faut varier les paramètres de densité et le coefficient qui représente la fraction maximale qu'un nœud peut couvrir. Or, dans une topologie où les nœuds sont distribués aléatoirement, un nœud *y* dans $N_1(x)$ couvre dans le cas idéal un ratio de 20% des voisins à deux sauts de *x*. Ce pourcentage peut atteindre 40% dans certains cas.

En effet, si on considère des nœuds avec un rayon de communication *R*, la surface de la région contenant les voisins à deux sauts de *x* est donné par $Surface_{N_2} = \pi(2R)^2 - \pi R^2 = 3\pi R^2$. Or, dans le cas où *y* est à la distance *R* de *x* (*Voir* Figure 4.2-a), la surface de la région $Surface_{N_1}$ contenant $N_1(y)$ est maximale. On a alors l'équation suivante : $Surface_{N_1} \le \pi R^2 - (2R^2 \frac{\pi}{3} - \frac{\sqrt{3}}{2} R^2) = (\frac{\pi}{3} + \frac{\sqrt{3}}{2})R^2$.

Donc, dans le cas idéal (Figure 4.2-a), le ratio de ces deux surfaces est donné par :

$$Ratio_{Idéal} \le \frac{(\frac{\pi}{3} + \frac{\sqrt{3}}{2})R^2}{3\pi R^2} = \frac{2\pi + 3\sqrt{3}}{18\pi} \le 0.203$$

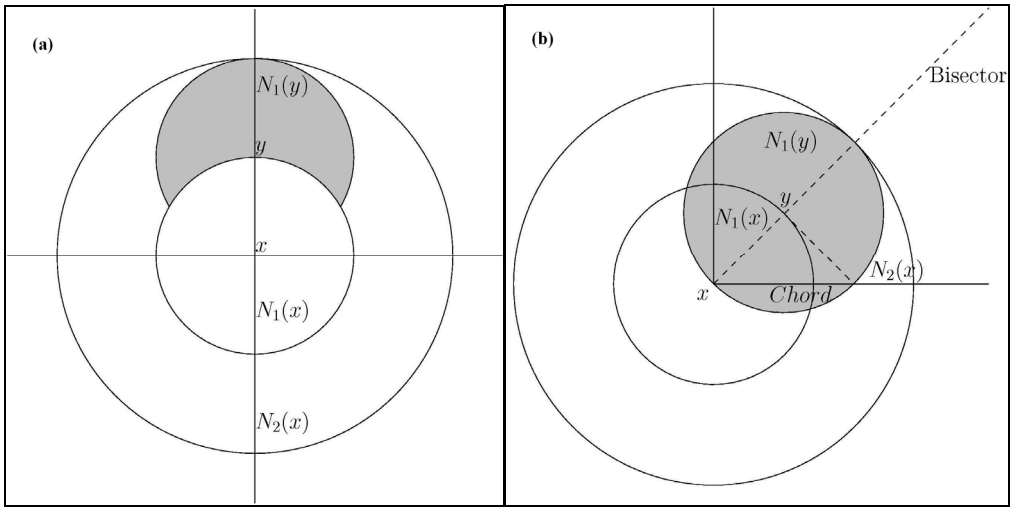

Figure 4.2 Région de couverture à 2-sauts.

Si le nœud *x* est sur la frontière de la région simulée, le ratio est alors donné par :

$$Ratio_{Frontière} \leq \frac{2(\frac{\pi}{3}+\frac{\sqrt{3}}{2})R^2}{3\pi R^2} = \frac{2\pi+3\sqrt{3}}{9\pi} \leq 0.406 \text{ (}Voir\text{ Figure 4.2-b)}$$

Nous avons alors fixé notre coefficient à 25% dans nos simulations. En d'autres termes, un nœud pourrait être au maximum connecté à un quart du nombre de nœuds à 2-sauts d'un nœud donné. Figure 4.3 montre que les nombres de MPR sélectionnés par SU-OLSR et par OLSR sont les mêmes quand le rayon de communication est supérieur à 300 *m*. En d'autres termes, SU-OLSR avec le *Critère II* (*Option I* et *Option II*) ne rejette aucun nœud légitime. Pour les rayons de communication inférieure à 300 *m*, on remarque que le nombre de MPR complètement sécuritaires sélectionnés par SU-OLSR (*Option I*) est inférieur au nombre de MPR choisis par le protocole OLSR classique.

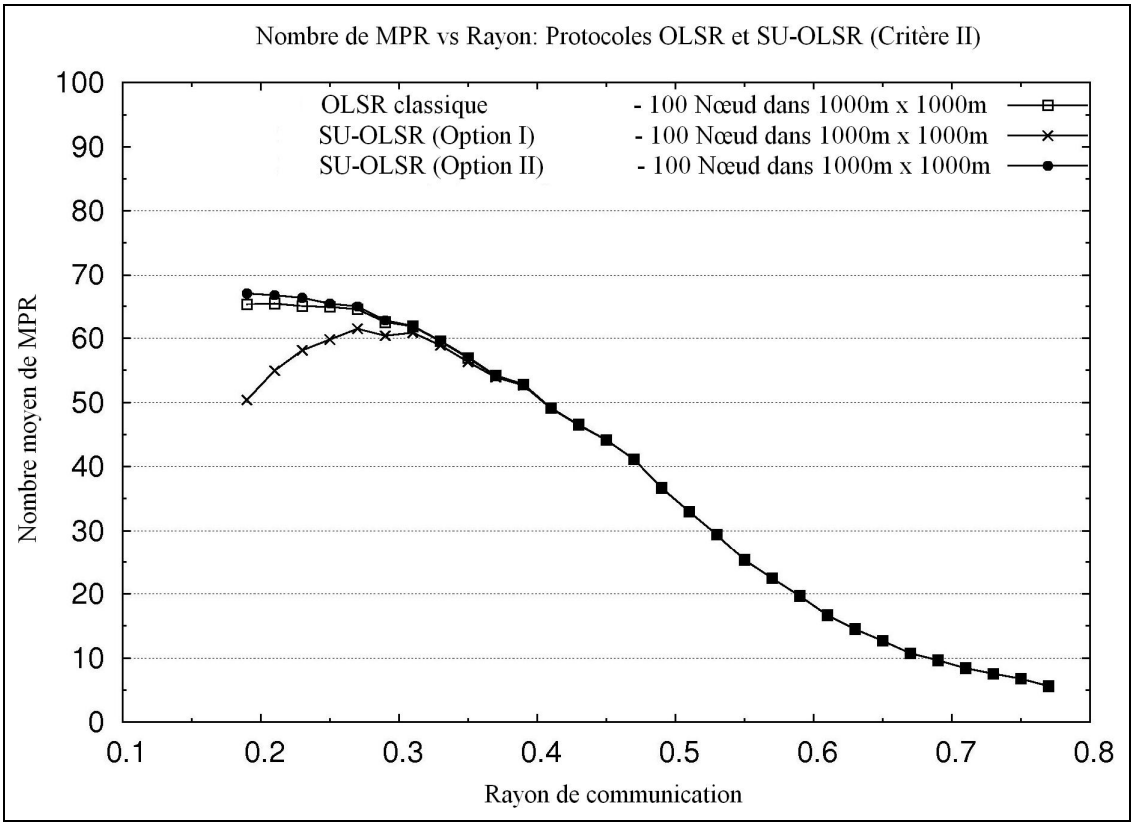

Figure 4.3 Nombre de MPR dans le cas du *Critère II* avec le coefficient 0.25.

Nombre de messages TC

Le nombre de MPR sélectionnés a un effet direct sur le nombre de messages TC diffusés dans le réseau. En effet, chaque MPR doit envoyer périodiquement les messages TC pour déclarer les nœuds qui l'ont choisi comme relais multipoint sécuritaire ou non sécuritaire. Ce type de message est diffusé par les autres MPR. Or pour limiter la charge du trafic dans le réseau, chaque message TC généré par un MPR est diffusé une seule fois dans un intervalle de temps par les autres MPR.

Le nombre de messages TC correspond au carré du nombre de MPR dans le réseau. La Figure 4.4 donne une comparaison entre le nombre de messages TC générés dans le cas des deux protocoles SU-OLSR (*Critère I*) et OLSR. On peut remarquer que dans le cas du SU-OLSR utilisant le *Critère I* et l'*Option I*, le nombre de messages TC diminue d'environ 33% par rapport au nombre de messages générés dans le cas d'utilisation du protocole OLSR.

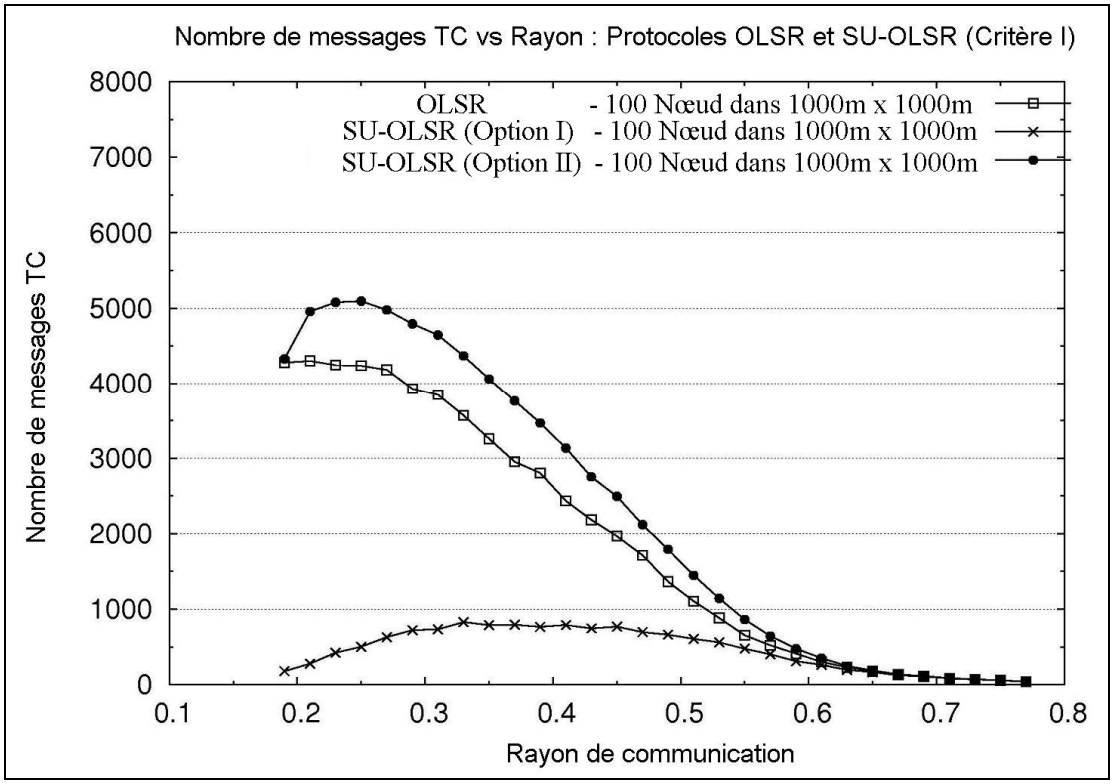

Figure 4.4 Comparaison du nombre de messages TC générés.

Le plus court chemin entre les nœuds

Les performances d'un protocole dépendent en particulier de la longueur des plus courts chemins entre les nœuds dans le réseau. Cette métrique a un effet direct sur le délai de bout-en-bout, la congestion dans le réseau et pourcentage des paquets perdus. Il est alors important d'analyser et comparer le choix des plus courts chemins par SU-OLSR en comparaison avec le protocole OLSR classique.

Une fois que l'étape de sélection des MPR a été faite, chaque nœud calcule ces chemins vers tous les nœuds du réseau. Ainsi, chaque nœud construit un graphe orienté qui représente sa vison de l'ensemble de la topologie du réseau. En d'autres termes, un nœud *s* calcule le graphe définit par $G(s) = \left\langle N(s), E_{N_1}(s) \cup E_{N_2}(s) \cup E_{MPRSel} \right\rangle$ avec :

- $N(s)$ est un sous ensemble de nœuds connu par *s*.
- $E_{N_1}(s) = \{(s,x) \mid x \in N_1(s)\}$
- $E_{N_2}(s) = \{(x,y) \mid x \in N_1(s) \wedge (y \in N_2(s) \wedge y \in N_1(x)\}$
- $E_{MPRSel} = \{(a,b) \mid b \in Sel_{MPR}(a)\}$

Remarquons que l'ensemble E_{MPRSel} ne dépend pas de *s* mais uniquement de l'ensemble de MPR (MPR_{Totale} et $MPR_{Partiel}$). L'algorithme de Dijkstra permet par la suite de calculer le plus court chemin entre un nœud *s* donné et tous les nœuds destinations dans le réseau.

a) Cas de l'*Option I* :

Débutant avec le *Critère I* (Pas de nœud isolé). On considère juste les MPR déclarés sécuritaires par tous les nœuds. Ainsi, et comme on l'a vu précédemment (*Voir* Figure 4.1), le protocole SU-OLSR sélectionne moins de MPR que le protocole OLSR classique. Ceci a un effet sur la connexité du graphe et plusieurs paires de nœuds ne sont pas connectées dans le cas d'un réseau à faible densité utilisant SU-OLSR. Dans ce cas, le plus court chemin entre deux nœuds sera très grand en comparaison avec le protocole OLSR classique.

Le Tableau 4.3 donne le pourcentage de paires de nœuds que voient leur plus court chemin augmenter d'un pas Δ. La dernière ligne ($\Delta = \infty$) représente le pourcentage de paires de nœuds pour lesquelles le protocole SU-OLSR n'a pas trouvé de chemin contrairement à OLSR. Les résultats de ce tableau confirment la nature non connexe du graphe résultant de SU-OLSR. En effet, pour avoir un graphe connexe, il faut un rayon de communication d'au moins 390 *m*. Pour ce rayon, SU-OLSR ne trouve pas de chemin valide pour 4.8% de ses nœuds (*Voir* Tableau 4.3). Alors que dans les mêmes conditions, OLSR a besoin juste d'un rayon de communication de 190 *m*.

Tableau 4.3 Longueur de chemins
SU-OLSR (Critère I + Option I) vs OLSR

Δ	190 *m*	250 *m*	330 *m*	390 *m*
0	0.3128	0.5357	0.8007	0.9198
1	0.0210	0.0486	0.0560	0.0276
2	0.0053	0.0188	0.0198	0.0069
3	0.0016	0.0073	0.0071	0.0020
4	0.0005	0.0031	0.0026	0.0006
≥ 5	0.0002	0.0025	0.0017	0.0007
∞	0.6586	0.3843	0.1120	0.0480

Ces résultats montrent que SU-OLSR avec le *Critère I* et l'*Option I* est très couteux et aura sûrement un impact sur la consommation d'énergie des équipements ainsi que sur leur coût.

Passons maintenant au *Critère II* (Pourcentage de couverture). L'utilisation de SU-OLSR est similaire dans ce cas au protocole OLSR (*Voir* Figure 4.3). Ceci se reflète dans les résultats sur la différence de la longueur de chemin entre les deux protocoles (*Voir* Tableau 4.4). Avec un rayon de communication de 250 *m* et un coefficient de 25%, SU-OLSR ne trouve pas de chemin pour seulement 0.49% de paires de nœuds en comparaison avec OLSR. Alors qu'avec un coefficient de 20% et le même rayon, les résultats sont toujours excellents avec juste 1.33% de paires de nœuds sans chemin.

Tableau 4.4 Longueur de chemins
SU-OLSR (Critère II + Option I) vs OLSR

Δ	190 *m*	230 *m*	250 *m*	250 *m*
Coefficient	0.25	0.25	0.25	0.20
0	0.5937	0.8767	0.9312	0.8808
1	0.1012	0.0705	0.0493	0.0765
2	0.0430	0.0175	0.0104	0.0196
3	0.0226	0.0059	0.0027	0.0060
4	0.0115	0.0025	0.0001	0.0022
≥ 5	0.0161	0.0018	0.0005	0.0015
∞	0.2120	0.0256	0.0049	0.0133

b) Cas de l'Option II :

Afin de voir si le protocole peut avoir de meilleures performances dans le cas du *Critère I,* nous allons réduire les contraintes. Ainsi, le protocole SU-OLSR peut utiliser les MPR déclarés sécuritaires que par certains nœuds pour calculer ses chemins entre les paires de nœuds.

On propose les trois alternatives suivantes :
***Option II-a* :** Les MPR déclarés sécuritaires par tous les nœuds ou par certains nœuds sont utilisés sans distinction.
***Option II-b* :** Les MPR déclarés sécuritaires par tous les nœuds sont préférés par rapport à ceux déclarés sécuritaires par certains nœuds.
***Option II-c* :** Un lien déclaré non sécuritaire par certains nœuds ne peut être utilisé que pour le dernier saut pour compléter le chemin vers la destination.

Les deux options II-a et II-b pourraient être implémentées en associant différents poids au arrêtes de E_{MPRSel} . On obtient donc un graphe *valué*. La Figure 1.1 montre les poids associés

pour ces deux options. Dans cette figure, les nœuds *A* et *B* considèrent le nœud *M* comme MPR sécuritaire alors que le nœud *C* le déclare non sécuritaire.

Dans le cas de l'*Option II-a,* on associe le poids $w = 1$ aux arrêtes dont le sommet est un MPR déclaré sécuritaire par tous ou certains nœuds (*Voir* Figure 4.5-(a)). Alors que dans le cas de l'*Option II-b*, les poids $w = 1$ et $w = n$ sont associés respectivement aux arrêtes dont le sommet est un MPR déclaré sécuritaire par tous les nœuds et les MPR déclarés sécuritaires par certains nœuds (*Voir* Figure 4.5-(b)). Le paramètre *n* représente ici le nombre de nœuds dans le réseau. Dans les deux cas, il est donc possible d'utiliser un MPR déclaré sécuritaire par un nœud pour atteindre ce nœud peu importe l'évaluation que les autres nœuds font de ce MPR.

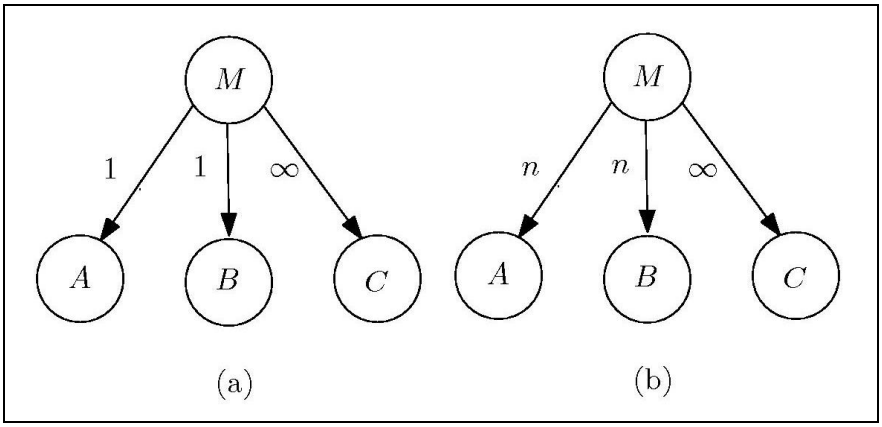

Figure 4.5 Poids associés dans le cas de l'*Option II-a* et *II-b*.

Par construction, la longueur des plus courts chemins est inférieure à *n* dans le cas de l'*Option II-a*. Dans l'autre cas, la longueur des plus courts chemins est sous la forme $a * n + b$, avec *a* est le nombre de sauts passant par des arrêtes dont les sommets sont des MPR déclarés non sécuritaires par certains nœuds; *b* est le nombre de sauts passant par des arrêtes dont les sommets sont des MPR déclarés sécuritaires par tous les nœuds dans le réseau.

Le Tableau 4.5 compare le protocole OLSR avec l'*Option II-a* du protocole SU-OLSR où on utilise sans distinction des MPR déclarés sécuritaires par tous les nœuds ou par certains nœuds. On remarque que SU-OLSR fournit des bonnes performances par rapport au protocole OLSR classique. Par exemple, dans le cas de SU-OLSR avec un rayon de communication 250 *m*, plus de 79.87% des paires de nœuds ne voient pas la longueur de leur plus court chemin être modifiée. On remarque aussi que dans ce cas, 17.84% des nœuds voient la longueur de leur plus court chemin augmenté d'un seul pas et 1.47% de deux pas. Alors que juste 0.46% des paires de nœuds ne trouvent pas de chemin valide (nœuds déconnectés du réseau). Les performances dans le cas du rayon 290 *m* sont parfaites. Le plus court chemin de seulement 7.9% des paires de nœuds a été augmenté d'un pas.

Tableau 4.5 Longueur de chemins
SU-OLSR (Critère I + Option II-a) vs OLSR

Δ	190 *m*	230 *m*	250 *m*	290 *m*
0	0.4456	0.7065	0.7987	0.9192
1	0.1970	0.2275	0.1784	0.0795
2	0.1009	0.0425	0.0147	0.0007
3	0.0490	0.0096	0.0023	0.00005
4	0.0278	0.0038	0.0008	–
5	0.0161	0.0016	0.0003	–
6	0.0096	0.0005	0.00004	–
7	0.0063	0.0002	0.00001	–
≥ 8	0.0131	0.00006	–	–
∞	0.1346	0.0077	0.0046	0.0005

Dans le cas de l'*Option II-b,* on préfère les MPR déclarés sécuritaires par tous les nœuds par rapport à ceux déclarés sécuritaires que par certains nœuds afin de construire le plus court chemin entre les nœuds. Avec cette option, au lieu de construire des plus courts chemins avec des MPR mixtes, le protocole SU-OLSR préfère des chemins qui ne passent pas par des MPR

déclarées non sécuritaires même si ses chemins sont plus longs. Ainsi, SU-OLSR choisit le plus court chemin en minimisant le nombre de MPR non sécuritaires utilisés.

Tableau 4.6 Longueur de chemins
SU-OLSR (Critère I + Option II-b) vs OLSR

Ω	Δ	190 *m*	230 *m*	250 *m*	290 *m*
0	0	0.3128	0.4673	0.5357	0.6773
0	1	0.0210	0.0413	0.0486	0.0582
0	2	0.0053	0.0166	0.0188	0.0257
0	3	0.0016	0.0074	0.0074	0.0104
0	4	0.0009	0.0034	0.0031	0.0041
0	≥ 5	0.0002	0.0031	0.0021	0.0027
1	0	0.0891	0.1424	0.1545	0.1377
1	1	0.0554	0.0796	0.0769	0.0414
1	2	0.0155	0.0230	0.0237	0.0109
1	≥ 3	0.0072	0.0161	0.0168	0.0056
2	0	0.0232	0.0342	0.0296	0.0128
2	1	0.0658	0.0630	0.0411	0.0099
2	2	0.0283	0.0201	0.0114	0.0011
2	≥ 3	0.0191	0.0152	0.0079	0.0005
≥ 3	–	0.2205	0.0595	0.0178	0.0011
–	∞	0.1346	0.0077	0.0046	0.0005

Le Tableau 4.6 donne la différence de longueur de chemins entre OLSR et SU-OLSR utilisant cette option. La première colonne Ω représente le nombre de MPR non sécuritaires utilisés pour construire le plus court chemin. On remarque que lorsqu'on utilise que les MPR sécuritaires (Ω=0), les résultats obtenus sont les mêmes que dans le cas de l'*Option I* (*Voir* Tableau 4.3 page 70). On peut noter aussi que la dernière ligne ($\Delta = \infty$) des Tableau 4.5 et Tableau 4.6 sont identiques. En effet, la connectivité du graphe et l'existence des chemins

entre les nœuds sont indépendantes du choix de l'*Option II-a* ou de l'*Option II-b*. La seule chose qui change entre ses deux options est la longueur des chemins entre les nœuds.

Les performances du protocole SU-OLSR sont très bonnes. Pour construire le plus courts chemins pour chaque nœud avec un rayon de communication de $250\,m$, 61.57% des paires de nœuds utilisent juste les MPR sécuritaires pour, 27.19% des nœuds utilisent un seul MPR non sécuritaire, 9% des paires de nœuds utilisent deux MPR non sécuritaires, 1.78% des paires de nœuds utilisent plus que trois MPR non sécuritaires et finalement juste 0.46% des paires de nœuds ne sont pas connectés.

Pour sa part, l'*Option II-c* ne sera pas traiter dans ce travail et sera laisser comme un axe de recherche pour de futurs travaux.

4.4 Environnement de simulation dynamique

4.4.1 Simulateur *ns-2*

Motivations pour l'utilisation de *ns-2*

ns-2 (Network Simulator version 2) est un simulateur orienté objet (DARPA/NSF, 2008). Il a été développé comme partie du projet VINT (*Virtual InterNetwork Testbed*) à ISI (*Information Sciences Institute*) et supporté par DARPA. La première version *ns-1* date de 1989. L'apparition de *ns-2* sous le projet VINT avait pour but d'unifier les efforts de la communauté des chercheurs pour fournir une plateforme de simulation puissante, reconnue et permettant des simulations complexes et proches des conditions réelles. Le résultat, *ns-2* est devenu l'un des simulateurs réseau le plus utilisé par la communauté des chercheurs pour expérimenter de nouvelles idées. Il permet entres autres une simulation complète de toutes les couches réseau et la pile TCP/IP avec une variété de protocoles et de paramètres de simulation (mobilité, rayon de transmission et densité, etc).

Ce simulateur a fait l'objet de plusieurs études et comparaisons avec d'autres simulateurs existants. Figure 4.6-a donne les statistiques concernant la popularité de simulateurs les plus connus et on remarque que *ns-2* devance tous les simulateurs et en particulier OPNET avec 88.8% (Hogie, Bouvry et Guinand, 2006). Figure 4.6-b donne les résultats d'une étude statistique sur l'utilisation de différents simulateurs dans 151 articles de publication acceptés dans la conférence ACM MobiHoc entre 2000 et 2005 (Kurkowski, Camp et Colagrosso, 2005). On y remarque aussi que *ns-2* est le simulateur de référence pour les simulations dans les réseaux sans fil Ad-Hoc avec un pourcentage de 43.8%. Une autre approche basée sur un système de points a été également proposée dans (Lang, 2008) pour comparer *ns-2,* GloMoSim et QualNet.

Lorsqu'il s'agit de mettre en place une nouvelle implémentation d'un protocole ou une solution, il est souvent nécessaire de faire des modifications dans le code source du simulateur ou dans le code des protocoles qu'il utilise. Or, *ns-2* est un logiciel

multiplateformes, gratuit sous licence GNU/GPL et son code source est accessible contrairement aux logiciels commerciaux ce qui le rend encore plus attractif.

Toutes ses raisons ainsi que l'existence d'une implémentation d'OLSR sous *ns-2* nous ont encouragés à l'adopter comme simulateur de référence dans notre implémentation expérimentale.

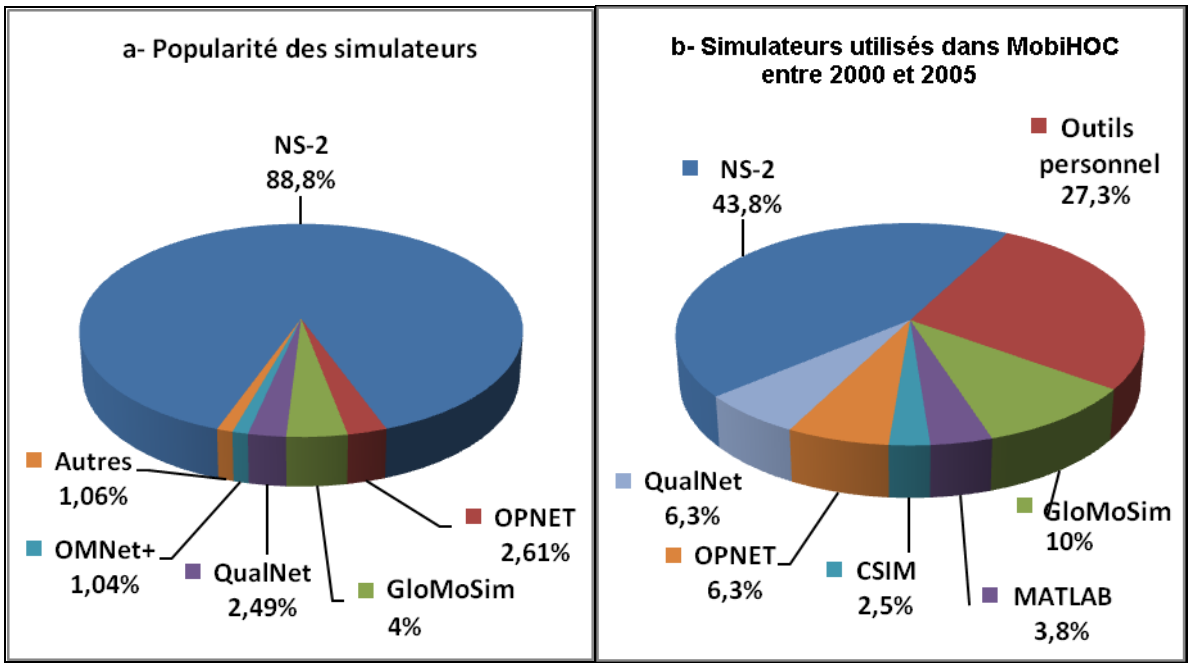

Figure 4.6 Taux d'utilisation des simulateurs réseau.

Présentation de *ns-2*

Le simulateur *ns-2* permet de simuler différents types de réseau, incluant les réseaux filaires et sans fil. Il est écrit en deux langages *C++* et OTcl. Les composantes en *C++* sont utilisées pour faire fonctionner le corps du simulateur. OTcl est utilisé comme interface et interprète pour les scripts de simulation, la configuration des nœuds et pour faire la liaison avec les classes objets de *C++* de *ns-2*. OTcl est une extension orientée objet de Tcl (*Tool command language*) et permet une facile intégration avec d'autres langages. Le lien entre les librairies

des deux langages est réalisé grâce à la librairie TclCL ce qui permet d'interfacer le code *C++* et Tcl.

L'idée de ce simulateur est d'utiliser *C++* dans toutes les opérations qui nécessitent un calcul intense et ainsi profiter de la vitesse d'exécution des objets compilés en *C++*, alors qu'OTcl est utilisé dans la configuration des scénarios de simulation. En effet, aucune compilation n'est nécessaire lorsqu'on change les paramètres de simulation dans l'interprète OTcl ce qui représente un gain de temps considérable lors des simulations. Pour chaque objet OTcl utilisé dans l'interprète, le simulateur a un objet miroir en *C++*. Ainsi *ns-2* établit une correspondance (*mapping*) entre les objets OTcl et les objets *C++*. Ainsi, à chaque objet OTcl est associé un objet *C++* et vice versa.

Le simulateur *ns-2* fournit un environnement de simulation sans fil complet appelé *CMU-Wireless Model*. Ce modèle a été développé dans le cadre du projet Monarch (CMU Monarch Project, 1999) et il a été intégré dans *ns-2* dès 1998-1999.

Figure 4.7 donne le schéma interne des nœuds sans fil dans *ns-2* ainsi que le mécanisme de communication et interaction entre les composantes de deux nœuds α et β échangeant un trafic CBR (*Constant Bite Rate*). Les composants internes des nœuds sont sous forme de classes d'objet et sont appelés agents. Lors des simulations, *ns-2* fait des liens entre ses agents (appelé aussi *Plumbing*) pour construire l'ensemble du réseau simulé. Par la suite, on donnera une brève description des composantes les plus importantes des nœuds utilisés dans les simulations des réseaux sans fil Ad-Hoc (Fall et Varadhan, 2008).

Couche liaison (LL) : La couche liaison a plusieurs fonctionnalités dans les réseaux LAN (retransmission, gère la les files d'attentes des paquets, etc). Une liaison entre le module LL et le module ARP (*Adress Resolution Protocol*) a été ajoutée dans le cas du modèle des réseaux sans fil dans *ns-2* (*Voir* Figure 4.7).

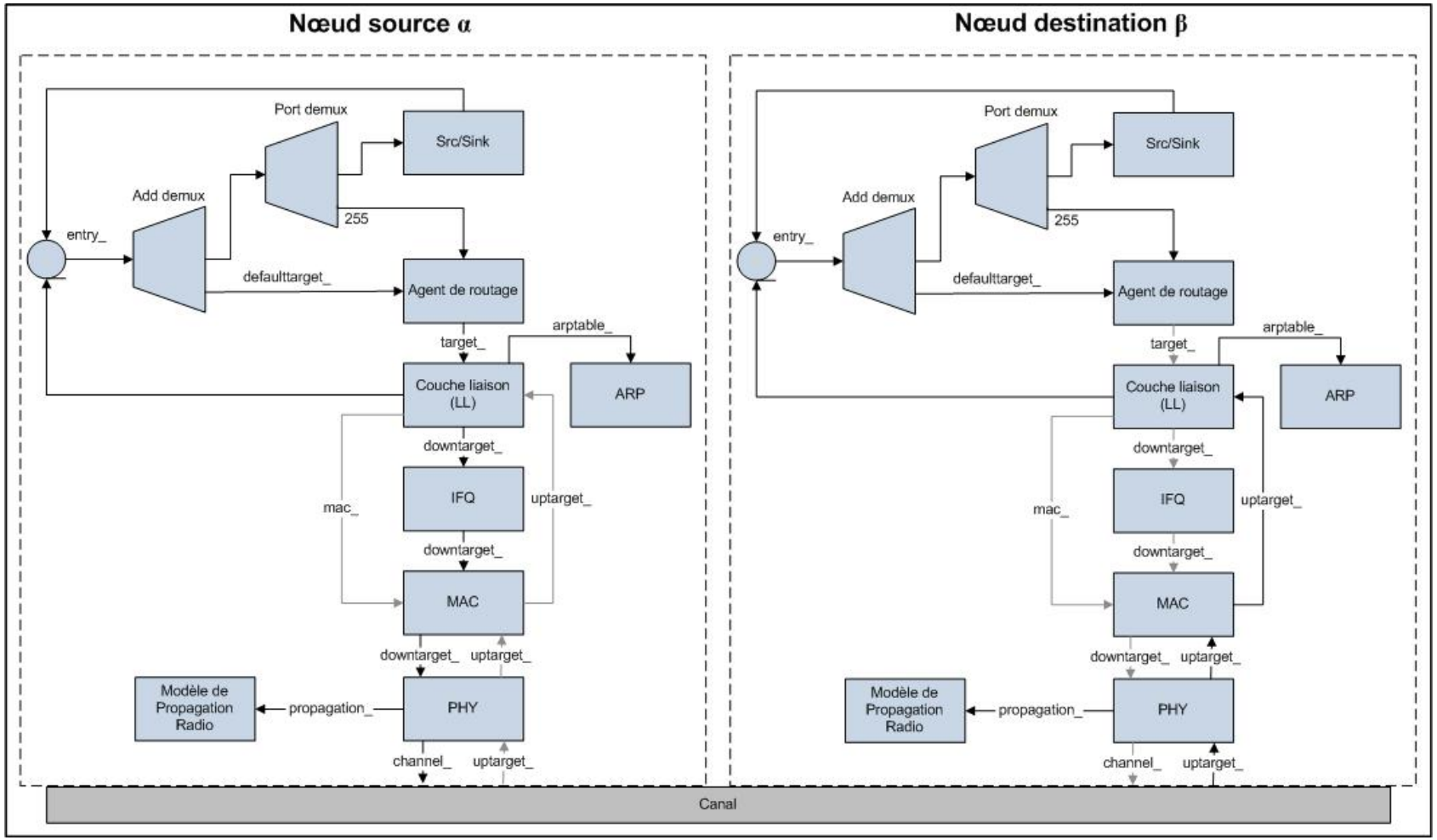

Figure 4.7 Structure interne et communications (*Plumbing*) entre les composantes de deux nœuds dans *ns-2*.

Couche MAC : Le standard IEEE 802.11 avec la technique DCF (*Distributed Coordinated Functio*n) sont implémentés sous ns-2 (Bianchi, 2000; IEEE 802.11, 2007). Cette technique utilise le schéma CSMA/CA (*Carrier Sense Multiple Access / Collision Avoidance access*). Le mécanisme CA ainsi que son schéma RTS/CTS (*Request To Send / Clear To Send*) sont utilisés car il est impossible de détecter les collisions dans les réseaux radio.

Interface réseau (PHY) : Dans *ns-2* l'interface réseau est utilisée par les nœuds sans fil pour accéder au canal. Ce média donne une émulation de la carte Lucent Wavelan DSSS (*Direct-Sequence Spread-Spectrum*) opérant à 914 Mhz et avec une bande passante de 2 Mbps. Ce qui respecte les normes du standard IEEE 802.11.

Modèle de propagation radio : Le simulateur utilise deux modèles de propagation radio *Friss* et *Two Ray Ground* que l'on verra dans la section 4.5.1.

Antenne : Les antennes utilisées par les nœuds dans *ns-2* sont omnidirectionnelles et isotopiques ç.-à.-d les antennes rayonnent de la même manière dans toutes les directions.

Chaque simulation dans *ns*-2 est exécutée à partir d'un script écrit en Tcl. Dans ce script, on spécifie les paramètres de simulation (temps de simulation, modèle de propagation radio, type de canal sans fil, modèle de mobilité, nombre de nœud dans le réseau, transmission radio, type de protocole, bande passante, type de trafic, etc). Les résultats des simulations sont fournis dans *ns-2* sous forme de fichiers traces. Le simulateur inscrit chaque événement survenu durant la simulation dans une ligne du fichier texte.

L'interface NAM (*Network Animator*) permet d'avoir une visualisation graphique de la simulation. Cette extension de *ns-2* est idéale pour avoir une idée visuelle sur le type de la topologie, la densité, la mobilité et les données échangées. La Figure 4.8 donne une version simplifiée du processus de simulation dans *ns-2*.

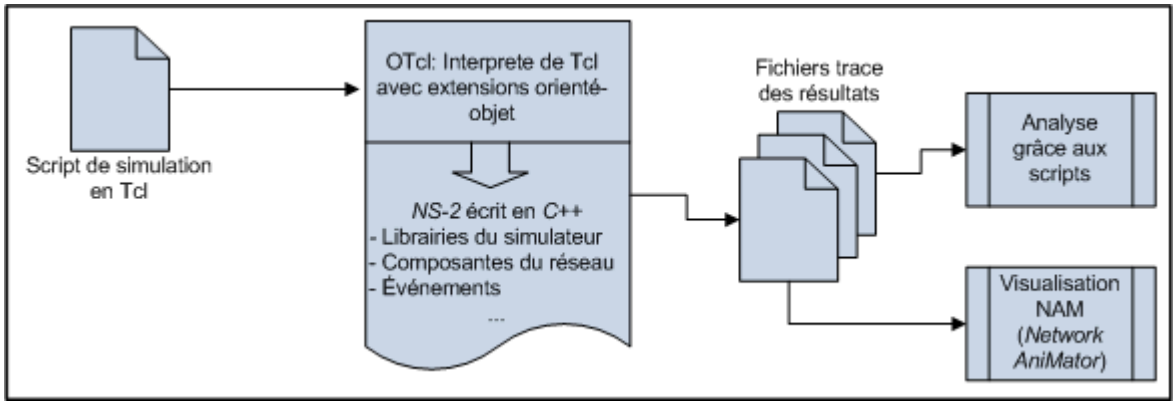

Figure 4.8 Processus de nos simulations dans *ns-2*.

4.4.2 Implémentation de OLSR et SU-OLSR sous *ns-2*

Le simulateur *ns-2* fournit une plateforme de simulation pour plusieurs protocoles Ad-Hoc. Par défaut, ce simulateur intègre DSR et AODV mais malheureusement pas le protocole OLSR. Mais il existe des extensions pour *ns-2* afin d'implémenter le protocole OLSR. Notre choix a été UM-OLSR (Roy, 2006). Ce package permet d'intégrer OLSR (RFC-3626) dans le simulateur *ns-2*, il est sous licence GNU et il a été utilisé dans plusieurs articles afin d'expérimenter le protocole OLSR. Or, ce package est compatible avec la version 2.29 de *ns-2*. Ceci explique notre choix de la version de *ns-2* à utiliser comme plateforme de simulation.

Le développement de la version du protocole SU-OLSR (*Critère I* avec l'*Option II-a*) pour *ns-2* a été basé sur le package UM-OLSR. En effet, nous avons remplacé la partie de l'algorithme de sélection de MPR dans UM-OLSR par notre nouvel algorithme de sélection. Le programme a été développé en *C++*. Nous avons par la suite intégré le nouveau package UM-SU-OLSR dans *ns-2* sous une machine Linux (Ubuntu). Et finalement, la validation de notre implémentation a été faite grâce à des modèles de topologie bien déterminés.

Tableau 4.7 Plateforme de simulation

Protocole	Simulateur	Package
SU-OLSR (*Critère I* avec l'*Option II-a*)	*ns-2 v2.29*	UM-SU-OLSR
OLSR	*ns-2 v2.29*	UM-OLSR

4.5 Simulation dynamique

Dans cette section, on comparera les performances des protocoles SU-OLSR et OLSR dans un environnement dynamique avec mobilité variable pour avoir des scénarios proches de ceux réels. Le but est de pouvoir évaluer notre protocole dans un environnement dynamique et valider les résultats obtenus dans la partie statique.

4.5.1 Paramètres de simulations

Modèle de mobilité et mouvement des nœuds dans *ns-2*

Les nœuds dans les réseaux Ad-Hoc peuvent être mobiles. L'évaluation des performances des protocoles dans les réseaux Ad-Hoc nécessite donc des tests dans des conditions réelles. Or, dans *ns-2* il est possible d'intégrer des modèles de la mobilité des nœuds pour rendre les simulations proches de la réalité. Un modèle de mobilité permet de générer un scénario de mouvement des nœuds.

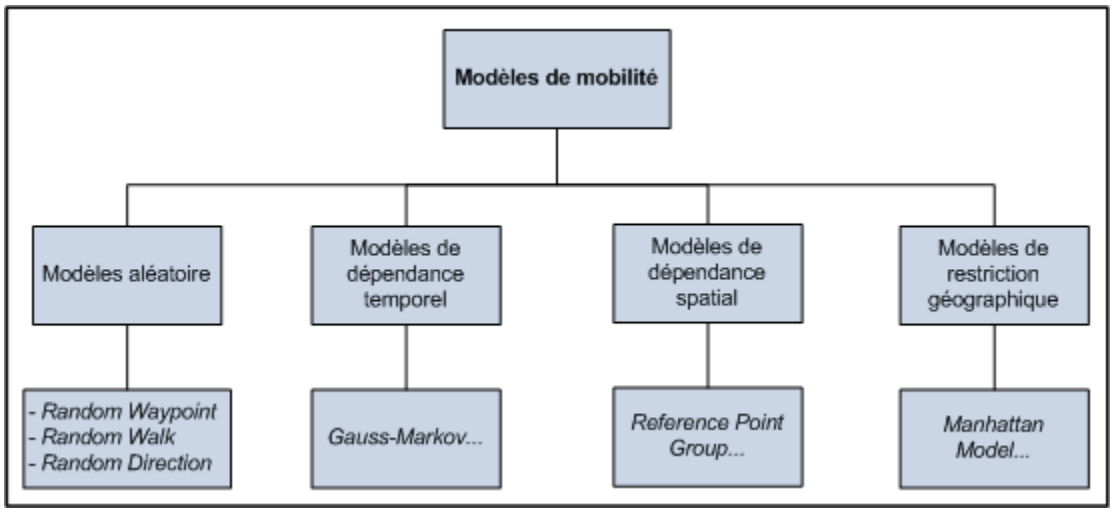

Figure 4.9 Classification des modèles de mobilité.

La Figure 4.9 donne une classification des modèles de mobilité les plus connus (Sarkar, Besavaraju et Puttamadappa, 2008). On peut y remarquer quatre catégories : modèles de mobilité aléatoire, modèles de mobilité de dépendance temporelle, modèles de mobilité de

dépendance spatiale et modèles de mobilité de restriction géographique. Une étude de ces modèles a été donnée dans (Camp, Boleng et Davies, 2002). Or, le modèle aléatoire et en particulier *Random Waypoint Mobility Model* (Bettstetter, Resta et Santi, 2003; Le Boudec et Vojnovic, 2005) a été choisi et souvent utilisé dans plusieurs travaux pour les simulations et l'analyse des performances des protocoles de routage dans les réseaux Ad-Hoc. D. Lang donne les raisons de ce choix pour la simulation et la comparaison des différents protocoles (Lang, 2008).

Dans ce modèle de mobilité, les nœuds sont placés d'une manière aléatoire dans une zone carrée. Le mouvement de chaque nœud est indépendant. Chaque nœud choisit un point destination de manière aléatoire et il se déplace vers ce point avec une vitesse uniforme choisie entre $0\,\text{m/s}$ et V_{Max}. Une fois que le nœud arrive à la destination, il marque une pause de temps donné et se déplace de nouveau vers une autre destination choisie aléatoirement avec d'autres paramètres de vitesse (*Voir* Figure 4.10).

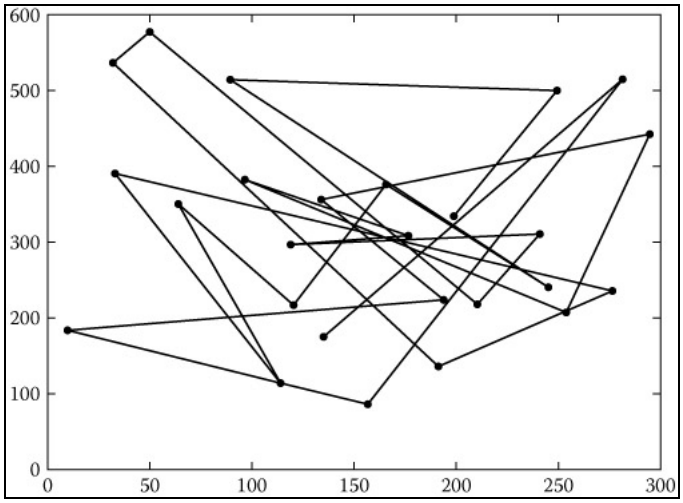

Figure 4.10 Mouvement d'un nœud selon le modèle *Random Waypoint.*
Tirée de Sarkar, Besavaraju et Puttamadappa (2008, p. 264)

Dans nos simulations, nous avons utilisé ce modèle de mobilité. Nos scénarios de mobilité ont été générés grâce à l'implémentation sous *ns-2* du code source d'une contribution qui

donne les scénarios de mobilité *Random Waypoint* sous format TCL de *ns-2* (Palchaudhuri, Le Boudec et Vojnovic, 2005). La Figure 4.11 donne un exemple de la topologie utilisée.

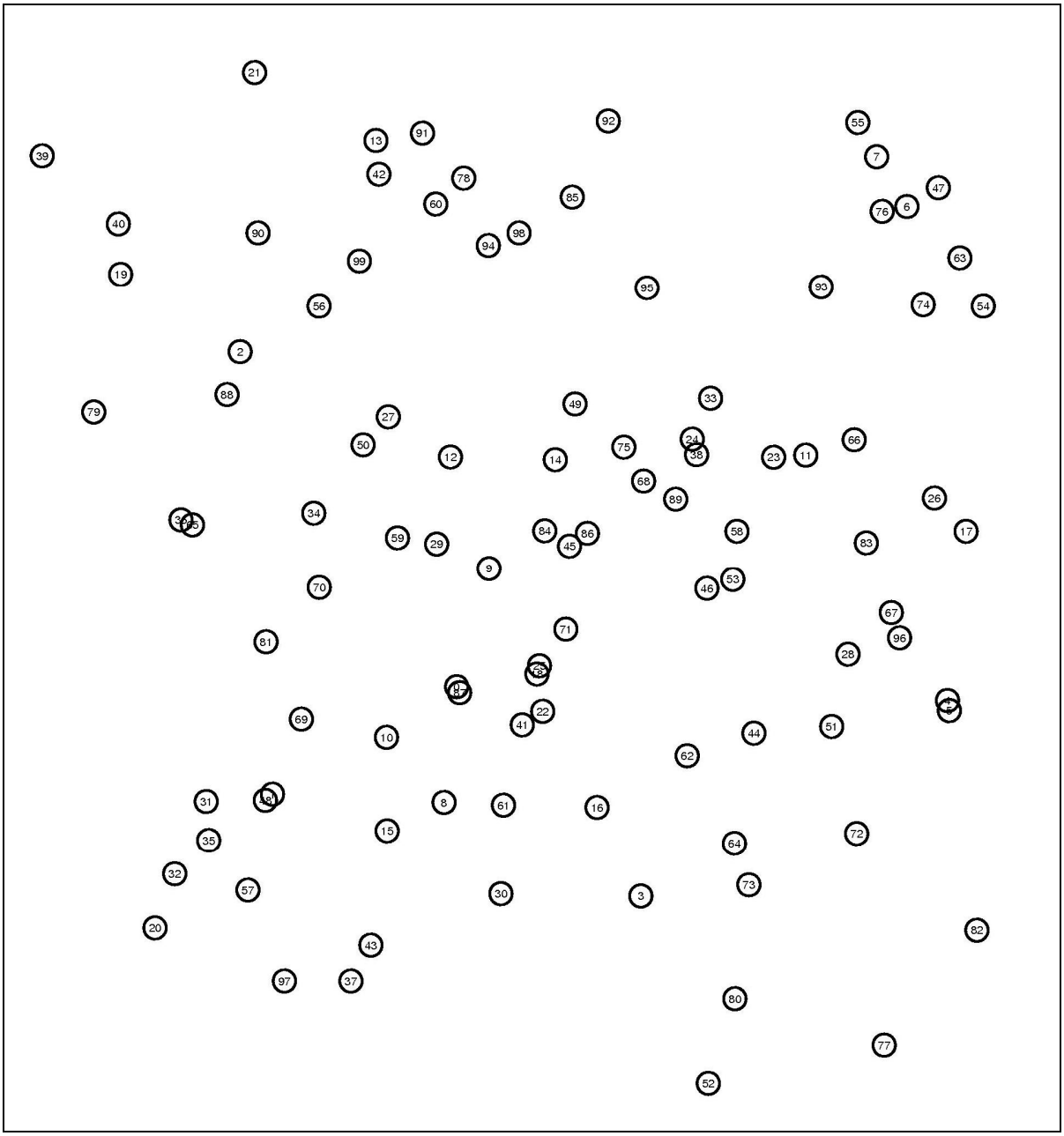

Figure 4.11 Exemple de 100 nœuds dans $1000\,m \times 1000\,m$ **.**

Scénarios des trafics de donnés

Dans un réseau Ad-Hoc, différents types de communication pourraient être établis entre les nœuds. Pour émuler ce type de communication dans *ns-2*, on utilise des applications de trafic à débit constant CBR (*Constant Bit Rate*) et qui modélisent la couche application sur des agents de transport UDP (*User Datagrammes Protocol*). On génère les scénarios de trafic aléatoires grâce à la modification d'un script *cbrgen.tcl* en TCL fourni sous *ns-2*. Nos scénarios de trafic modélisent des communications entre 25 paires disjointes de nœuds choisies à chaque fois d'une manière aléatoire. Chaque nœud source émet, à un instant donné, choisi aléatoirement entre 10 seconds et 300 seconds, des paquets de taille 512 octets avec un débit de 4 paquets par seconde. Chaque communication est établie jusqu'à la fin de la simulation.

Modèle de propagation radio et rayon de transmission

Le simulateur *ns-2* utilise deux modèles de propagation radio. En effet, la propagation radio dans une communication LOS (*Line-Of-Sight*) est caractérisée soit par le modèle *Friis free space* (Friis, 1946) ou soit par le modèle *Two-Ray Ground* en fonction de la distance *d* de laquelle l'antenne reçoit le signal.

Dans le cas où *d* est petit, le calcul de l'énergie reçue est donné par l'équation : $P_r(\mathrm{d}) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$ où P_t est l'énergie transmise, G_t et G_r sont le gain de l'antenne émettrice et l'antenne réceptrice respectivement, *L* est la perte du système, λ est la longueur d'onde du signal radio (Fall et Varadhan, 2008). Dans *ns-2*, $G_t = G_R = 1$ et $L = 1$.

Dans le cas où *d* est grand, le modèle *Two-Ray Ground* est utilisé avec l'équation d'énergie reçue : $P_r(\mathrm{d}) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}$ où h_t et h_r sont respectivement la hauteur d'antenne émettrice et l'antenne réceptrice et elles sont fixées à 1.5 dans *ns-2*. Le simulateur permet un calcul simple de l'énergie reçue[1].

La valeur de λ est fixée à 0.32822757 dans *ns-2* ce qui correspond à la fréquence 914Mhz de la carte Lucent Wavelan DSSS (Fall et Varadhan, 2008). Ce modèle de propagation sera utilisés au lieu du modèle *Friis free space* si : $d \geq d_c = \frac{4\pi h_t h_r}{\lambda}$.

Dans nos simulations des nœuds mobiles, on s'intéresse aux rayons de communication (*Communication range*) 190, 230, 250 et 290 mètres. Ce qui nécessite de prendre le *Two-Ray Ground* comme modèle de propagation radio. D'autre part, pour changer le rayon de communication des nœuds dans *ns-2*, il faut définir dans les paramètres de simulation la valeur exacte du *RXThresh* (*Receiving Threshold*) sous forme : Phy/WirelessPhy set RXThresh_ <valeur>. Pour nos quatre rayons de communication, on calcule la valeur correspondante de *RXThresh* grâce à un programme[2] en C fournie avec *ns-2.*

Sommaire des paramètres de simulation

Les tableaux suivants présentent le sommaire de nos simulations en mode dynamique. Il faut noter qu'on a effectué 200 simulations par protocole. Ses paramètres de simulation sont utilisés dans le script de simulation de *ns-2* écrit en TCL (*Voir* Annexe I). Nous avons également développé un script en Bash afin d'automatiser toutes nos simulations.

[1] Disponible dans ~ns2/indep-utils/mobile/tworayground.cc

[2] Disponible dans ~ns2/indep-utils/propagation/threshold.cc

Tableau 4.8 Paramètres de simulation pour SU-OLSR et OLSR

Paramètres	Valeur
Type de canal	Sans fil
Modèle de propagation radio	*Two Ray Ground*
Type d'interface réseau	IEEE 802.11
Type d'interface de fil d'attente	*DropTail / Priority Queue*
Longueur de fil d'attente	50 paquets
Type de lien de liaison	*Link Layer (LL)*
Type d'antenne	*Omni-directional*
Modèle de mobilité	*Random Waypoint*
Type de trafic	CBR
Trafic	4 paquets de 512 octets par second
Intervalle des messages HELLO	2 seconds
Intervalle des messages TC	5 seconds
Temps de simulation	300 seconds
Fréquence d'échantillonnage	À chaque 1sec (entre 2 sec et 300 sec)

Tableau 4.9 Scénarios de simulation pour SU-OLSR et OLSR

Scénarios	Vitesse max	Topologie (m^2)	Rayon de communication (m)	Scénarios de mobilité	Trafic CBR pairs disjoints
Fixe	0 m/s	1000 x 1000	190,230,250,290	10 réplications	25
Soldat	1.4 m/s	1000 x 1000	190,230,250,290	10 réplications	25
Bateau	5 m/s	1000 x 1000	190,230,250,290	10 réplications	25
Char	10 m/s	1000 x 1000	190,230,250,290	10 réplications	25
Voiture	20 m/s	1000 x 1000	190,230,250,290	10 réplications	25
Nombre de simulations par protocole				200 simulations	

4.5.2 Outils pour analyser les traces de simulation

Format de trace

Le simulateur *ns-2* fournit les résultats de simulation sous forme de fichier trace en mode texte où chaque ligne décrit un événement qui s'est produit lors de la simulation (opérations entre les couches). Le format de fichier trace a un effet direct sur la rapidité des simulations et en particulier sur l'utilisation de CPU et la mémoire de la machine de simulation. En effet, dans le cas de réseaux denses et pour des simulations de 80 nœuds et plus, les fichiers traces de chaque simulation peut dépasser un gigabit de taille. Il est possible de définir dans le script TCL de simulation quels événements doivent être conservés.

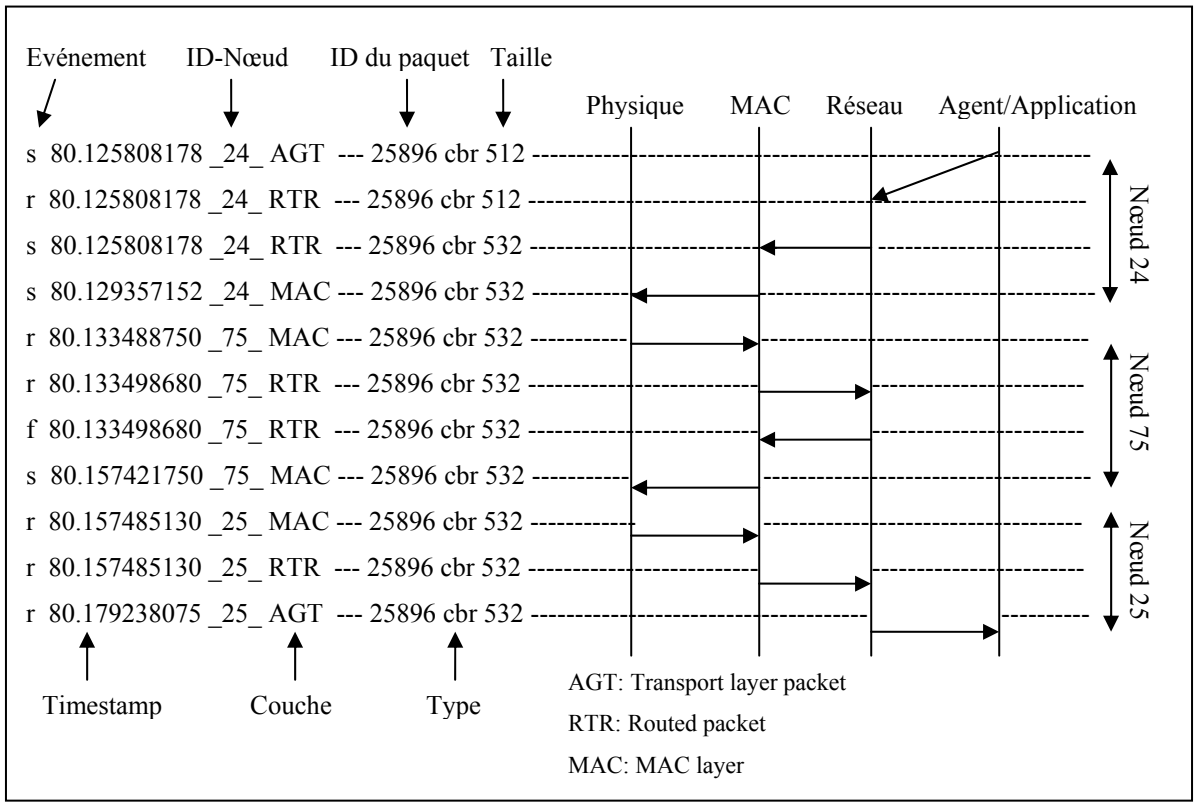

Figure 4.12 Extrait et détails importants dans les traces des simulations sous *ns-2*.

Les événements les plus importants dans les fichiers de traces sont la transmission (s), la réception (r), renvoi (f) et le rejet d'un paquet (*Voir* Figure 4.12). Chaque paquet généré dans

ns-2 durant une simulation, a un identifiant unique et permet de suivre son état dans le réseau et de calculer par la suite les performances de chaque protocole. Il faut noter que le simulateur *ns-2* ne fournit aucun outil pour analyser et exploiter les fichiers de traces. Il est ainsi nécessaire de développer des outils propres à ce qu'on cherche à examiner et extraire des simulations. Ceci présente un point négatif par rapport à certains logiciels de simulation comme OPNET.

Script MPR.py

Dans nos scripts de simulation nous avons choisi d'écrire toutes les secondes la table des MPR dans les fichiers de traces. Ainsi, à toutes les secondes entre 2 et 300, chaque nœud imprime l'ensemble des nœuds qui l'ont choisi comme MPR (*Voir* Figure 4.13).

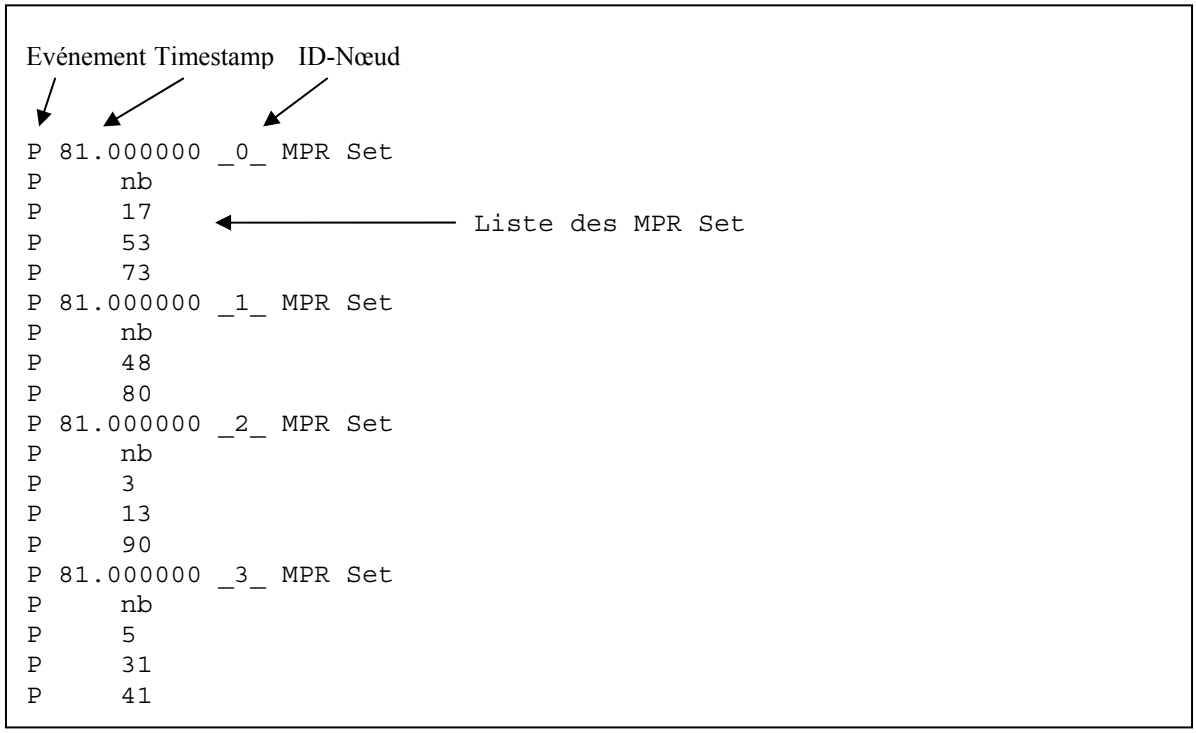

Figure 4.13 Liste des MPR dans le fichier trace de simulations sous *ns-2.*

Afin d'exploiter les résultats des MPR dans nos fichiers traces, nous avons développé un programme en Python (MPR.py). Ce programme permet d'analyser les fichiers de traces et

d'obtenir l'ensemble des MPR et les nœuds qui les ont choisis comme MPR pour chaque intervalle d'une seconde.

Script Performances.awk

Afin d'évaluer les performances de deux protocoles OLSR et SU-OLSR, nous avons utilisé un trafic CBR entre 25 paires disjointes. Chaque paquet est identifié d'une manière unique et grâce au fichier de traces on peut suivre son évolution entre la source et la destination (*Voir* Figure 4.12).

Afin d'exploiter ces informations, nous avons développé un programme en AWK (Performances.awk) permettant de tracer l'évolution de chaque paquet envoyé dans le réseau. Le délai moyen de bout-en-bout est calculé de la manière suivante :

$$\overline{Délai-de-bout-en-bout} = \frac{\sum_{1}^{n}\left(CBR_{Temps\ d'envoi} - CBR_{Temps\ réçu}\right)}{\sum_{1}^{n} CBR_{Réçu}}$$

Le pourcentage de paquets délivrés est donné par :

$$Pkt\_délivré\ \% = \frac{\sum_{1}^{n} CBR_{Réçu}}{\sum_{1}^{n} CBR_{Envoyé}} \times 100$$

Le programme Performances.awk fournit les résultats exploitables graphiquement en format DAT.

Script Graphique.dat

Finalement, on utilise Gnuplot (Janert, 2008) afin d'exploiter les résultats obtenus grâce aux programmes MPR.py et Performances.awk. Gnuplot permet la représentation graphique de données provenant d'un fichier de données texte (.dat ou .csv).

Le Tableau 4.10 donne un sommaire des outils développés pour la gestion et l'exploitation des résultats de nos simulations.

Tableau 4.10 Liste des outils développés pour nos simulations

Script	Langage	Entrer	Sortie
Sim-SU-OLSR.tcl	TCL	Paramètres	Traces
Sim-SU-OLSR.sh	Bash	Rayon; Scenarios	Traces *ns-2* SU-OLSR
Sim-OLSR.sh	Bash	Rayon; Scenarios	Traces *ns-2* OLSR
MPR.py	Python	Traces *ns-2*	Nombre de MPR "CSV"
Performances.awk	AWK	Traces *ns-2*	PDR; Paquet perdu
Graphique.dat	Gnuplot	MPR; Performances	Graphes

4.5.3 Résultats de simulation

Durant nos simulations nous avons étudié le comportement des protocoles SU-OLSR et OLSR dans un environnement dynamique. L'objectif n'est pas de simuler les attaques sur la sélection des MPR car en aucun cas, un nœud malicieux ne pourra être choisi comme MPR s'il présente l'un des comportements décris dans SU-OLSR. Le but est d'étudier l'impact du changement qu'on a effectué dans SU-OLSR car ce nouveau protocole peut choisir d'ignorer de couvrir certain nœuds contrairement à OLSR.

Nombre de MPR

La mobilité et la densité du réseau ont un impact direct sur les algorithmes de sélection des MPR. Les tableaux 4.11 et 4.12 présentent l'impact de ces paramètres sur le nombre de MPR sélectionnés dans le cas des scénarios de mobilité à 1.4 m/s et à 10 m/s. Le calcul des MPR est effectué à chaque seconde de simulation. Ces résultats sont la moyenne de toutes les simulations effectuées. Les paramètres μ et σ^2 représentent la moyenne et la variance du nombre de MPR pour chaque rayon de communication.

Dans les deux tableaux, le nombre de MPR est presque similaire pour les deux protocoles avec un léger avantage pour OLSR qui sélectionne moins de MPR. Par exemple, pour le rayon de communication 230 m, la moyenne des MPR sélectionnés dans le cas des scenarios de mobilité avec une vitesse de 1.4 m/s est de 81.55 MPR pour le protocole SU-OLSR. Pour

les mêmes scenarios, la moyenne est de 76.27 MPR pour le protocole OLSR (*Voir* Tableau 4.11). Ces donnés confirment les résultats obtenus dans le cas statique (*Voir* Figure 4.1 page 65).

Tableau 4.11 Nombre de MPR dans le cas d'une mobilité maximal 1.4 m/s

Rayon de communication		190 *m*	230 *m*	250 *m*	290 *m*
SU-OLSR	μ	80.67	81.55	80.78	77.30
	σ^2	2.65	2.45	0.94	2.39
OLSR	μ	78.97	76.27	77.04	73.70
	σ^2	1.38	5.33	0.79	2.99

Lorsqu'il s'agit des scénarios à grande mobilité, les deux protocoles choisissent plus de MPR. Si on prend les scénarios avec 10m/s comme vitesse maximale, le protocole SU-OLSR choisit en moyenne 87.91 MPR pour le rayon de communication 230 m. Par contre, OLSR ne choisit en moyenne que 85.79 MPR. Ce nombre élevé de MPR est expliqué par l'instabilité du réseau à cette vitesse.

Tableau 4.12 Nombre de MPR dans le cas d'une mobilité maximal 10 m/s

Rayon de communication		190 *m*	230 *m*	250 *m*	290 *m*
SU-OLSR	μ	87.27	87.91	87.29	85.44
	σ^2	1.46	0.53	1.14	1.62
OLSR	μ	84.46	85.79	84.84	83.84
	σ^2	2.65	1.67	1.32	2.23

Pour bien examiner le comportement des deux algorithmes de sélection de MPR, nous avons calculé à chaque seconde la différence entre les nombres de MPR choisis par les deux protocoles. Ces calculs sont effectués avec le même scénario de mobilité. La Figure 4.14

montre cette différence pour un scénario de mobilité à 1.4 m/s. Nous constatons que la différence entre le nombre de MPR pour les deux protocoles change constamment.

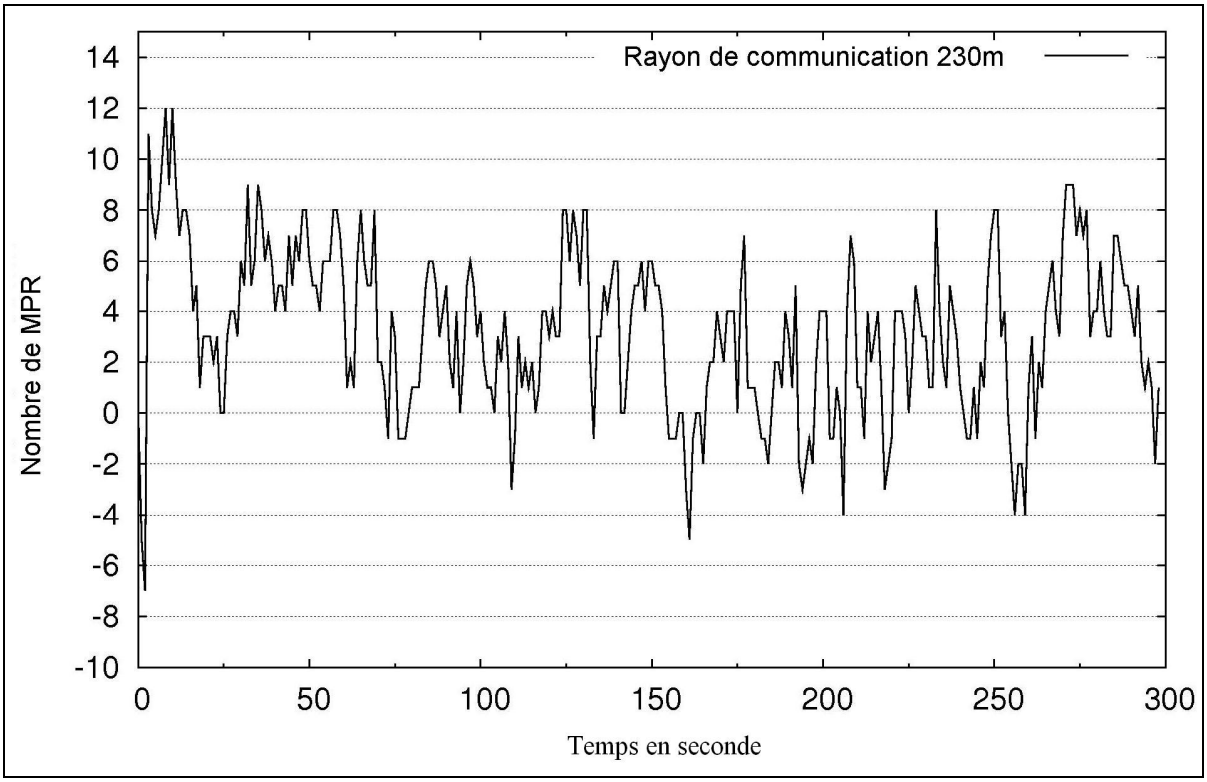

Figure 4.14 Différence de MPR choisis entre les deux protocoles (vitesse de 1.4 m/s max).

La même remarque s'applique dans le cas d'un scenario de mobilité à 10 m/s (*Voir* Figure 4.15). Il faut noter qu'à cette vitesse, le réseau est instable, ce qui explique la grand fluctuation dans le graphe en comparaison avec le graphe du Figure 4.14.

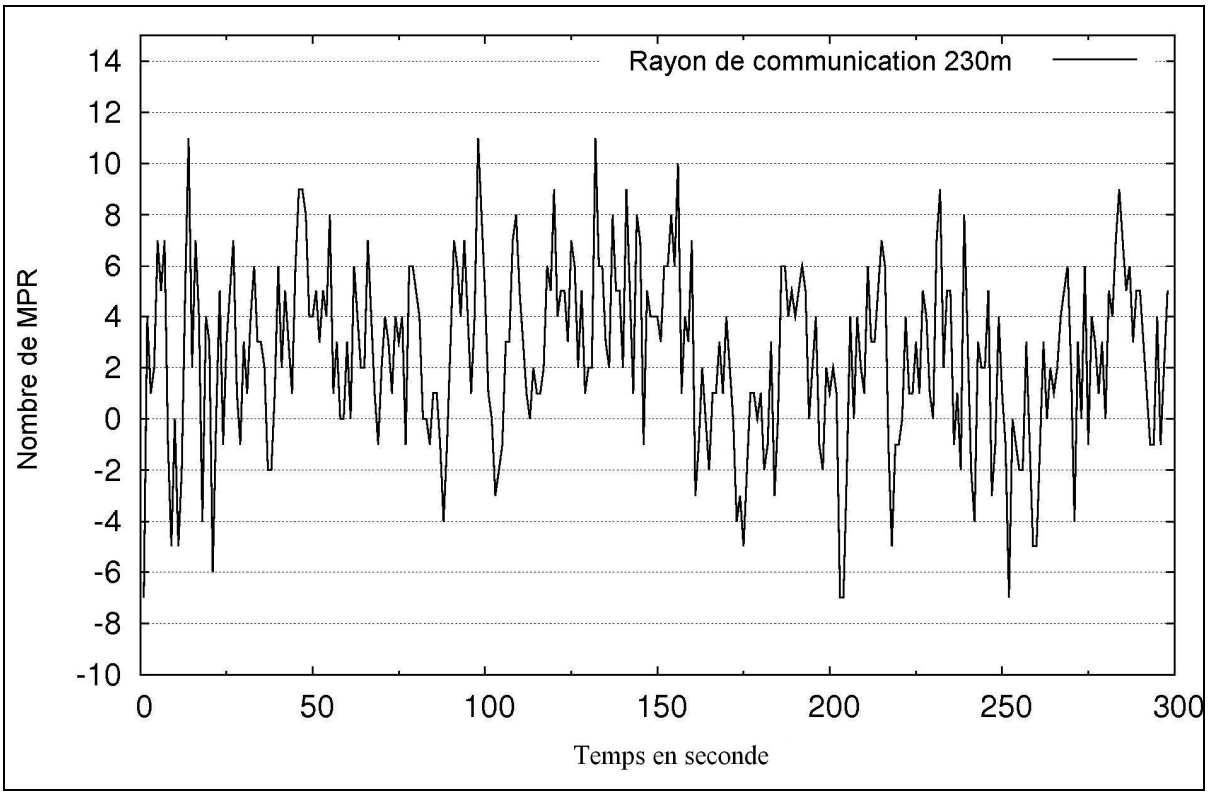

Figure 4.15 Différence de MPR choisis entre les deux protocoles (vitesse de 10 m/s max).

Pourcentage des paquets délivrés

La Figure 4.16 montre le pourcentage des paquets délivrés par rapport à la vitesse moyenne des nœuds avec un rayon de communication de 230 m et utilisant le protocole OLSR et SU-OLSR. Quand les nœuds sont stationnaires (vitesse = 0 m/s), la communication entre les nœuds se fait presque sans perte de paquets pour les deux protocoles OLSR et SU-OLSR. Dans ce cas statique, le protocole OLSR a un faible avantage avec 99.88% de paquets délivrés contre 99.29% pour SU-OLSR. Ceci confirme les résultats obtenu lors de nos simulations statiques (*Voir* la section 4.3.2 et le Tableau 4.6).

Par contre, quand la vitesse des nœuds augmente, les performances des deux protocoles se dégradent. Cette dégradation des performances est due à la vitesse à laquelle les deux protocoles doivent mettre à jour leurs tables de routage et déterminer les MPR. Dans un

environnement à haut mobilité, les liens entre les nœuds changent très rapidement et les connexions entre les nœuds existent pour des périodes de temps très courtes. Par contre pour des réseaux avec faible mobilité ou mobilité nulle, les liens existent pour une période de temps très grande favorisant ainsi le trafic CBR à être transmit correctement à la destination.

Or, jusqu'à une vitesse de mobilité de 5 m/s, les deux protocoles présentent le même comportement. Par contre, à partir de 10 m/s, le protocole SU-OLSR prend l'avantage par rapport à OLSR. Le PDR du SU-OLSR est à 38.46% contre 37.41% pour OLSR pour une vitesse 20 m/s. Ces résultats obtenues pour les deux protocoles sont conforme à ceux d'OLSR déjà traités dans plusieurs articles (Clausen, Jacquet et Viennot, 2005; Voorhaen et Blondia, 2006).

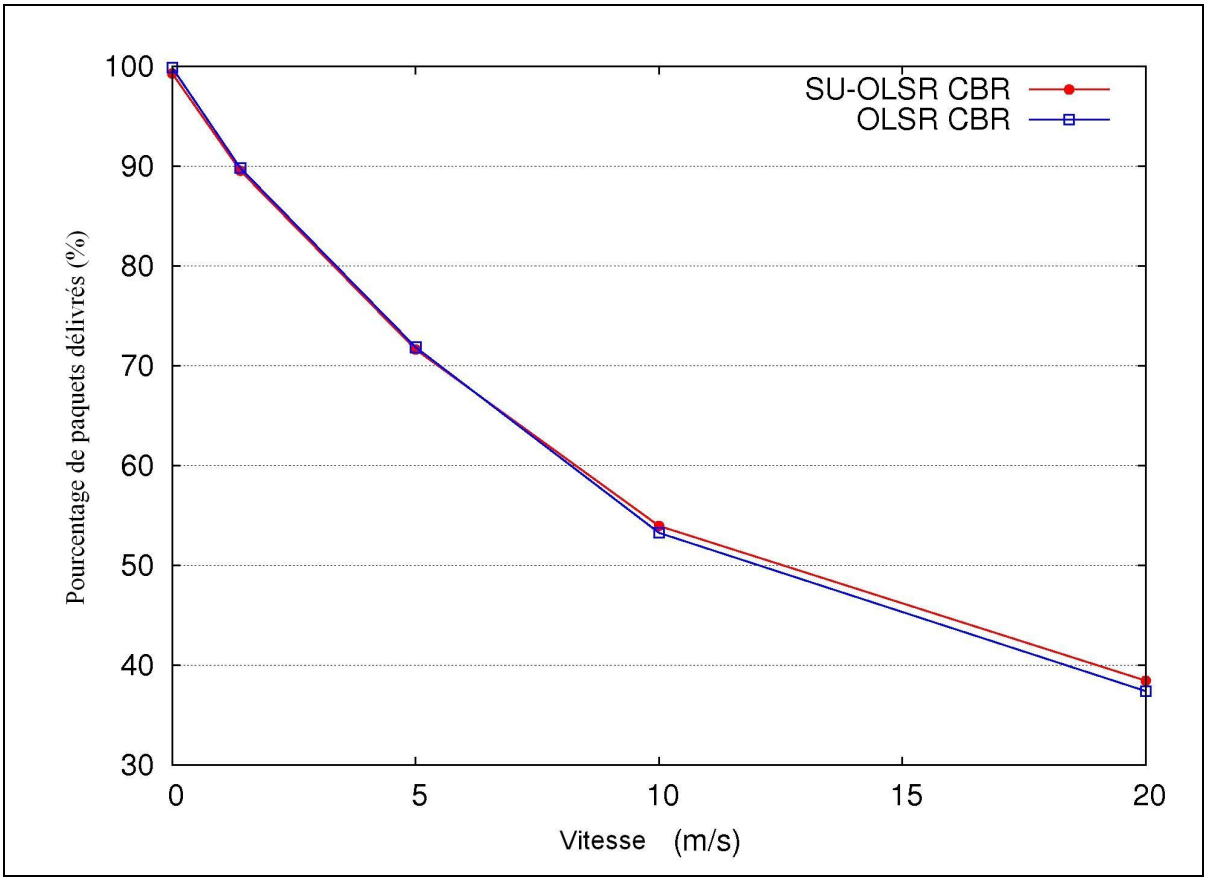

Figure 4.16 Pourcentage de paquets délivrés PDR pour les deux protocoles.

Délai de bout-en-bout

Le délai de bout-en-bout d'un paquet IP dans les réseaux est la somme des retards introduits par les nœuds intermédiaires entre la source et la destination. Il dépend du délai de traitement dans un nœud intermédiaire, du délai de la mise en file d'attente, du délai lors de l'envoi sur le support physique et le délai de la propagation selon la distance. La Figure 4.17 illustre le délai de bout-en-bout par rapport à la vitesse des nœuds dans le réseau utilisant un rayon de communication de 230 m. On constate que ce délai augmente avec la vitesse. Pour un réseau à haute mobilité (20 m/s), ce délai est maximal. Cette dégradation peut être expliquer par le fait qu'à cette vitesse, les MPR peuvent se déplacer loin et hors porté des nœuds qui les ont choisis comme MPR et cela peut se produire d'une manière très rapide. Ceci provoque des ruptures des liens entre la source du trafic CBR et la destination.

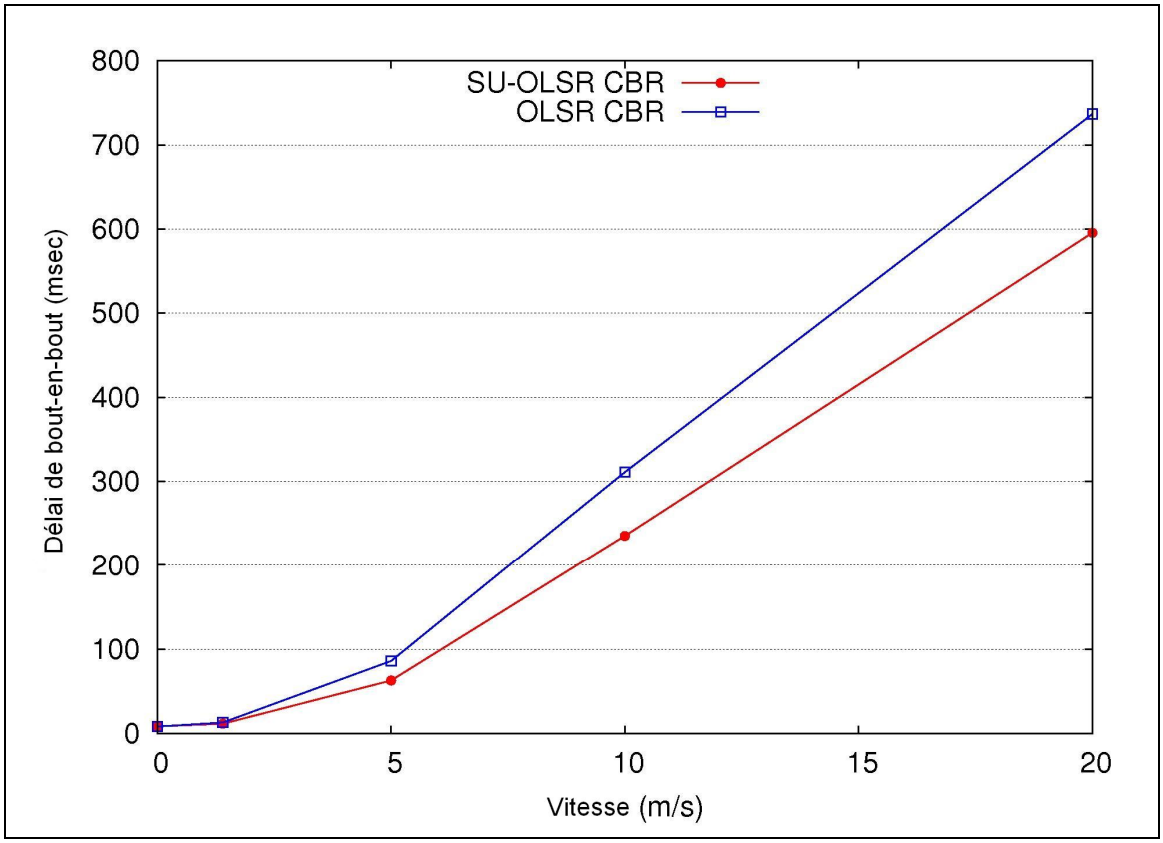

Figure 4.17 Délai de bout-en-bout pour les deux protocoles.

D'un autre coté, un délai supplémentaire est introduit à cause du processus de sélection des MPR pour remplacer les MPR déconnectés et trouver des nouveaux chemins vers la destination. Ceci affect de manière direct les performances des deux protocoles OLSR et SU-OLSR. Des résultats comparables sont obtenus dans (Clausen, Jacquet et Viennot, 2005; Voorhaen et Blondia, 2006).

Nous constatons que le protocole SU-OLSR prend de l'avantage par rapport à OLSR pour les réseaux à grands mobilités. Le nombre supérieur de MPR sélectionné par SU-OLSR à cette vitesse de mobilité par rapport à OLSR explique ce léger avantage dans le délai de bout-en-bout pour SU-OLSR. Si on regarde le cas fixe (0 m/s), le protocole OLSR a un léger avantage dans le délai avec 8.3 ms contre 8.5 ms pour SU-OLSR. La même chose s'applique pour le délai maximal (*Voir* Figure 4.18). Ceci confirme les résultats obtenues dans nos simulations statiques (*Voir* la section 4.3.2 et le Tableau 4.5). Nous remarquons que les deux protocoles obtiennent les mêmes longueurs de chemins pour 70.5% des paires de nœuds. Ceci explique cette légère différence de délai entre les deux protocoles SU-OLSR et OLSR.

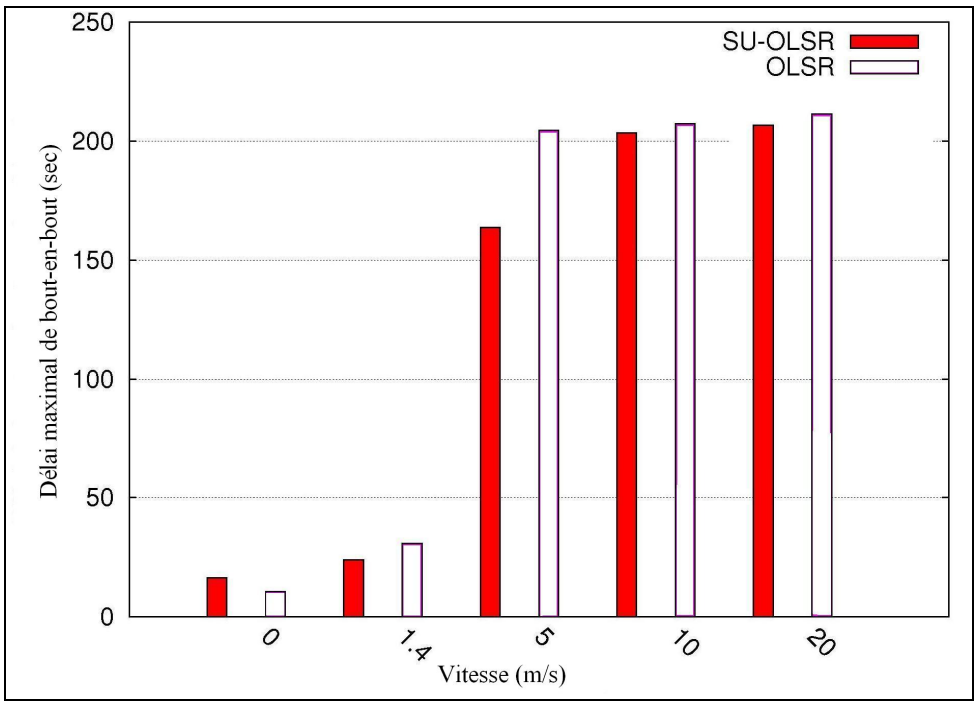

Figure 4.18 Délai maximal de bout-en-bout pour les deux protocoles.

Pour finir, la Figure 4.19 donne le nombre de messages délivrés (*Throughput*) durant les simulations en fonction des vitesses de mobilité. Nous constatons la dégradation des performances des deux protocoles dans les scénarios à grande mobilité mais avec un léger avance pour SU-OLSR.

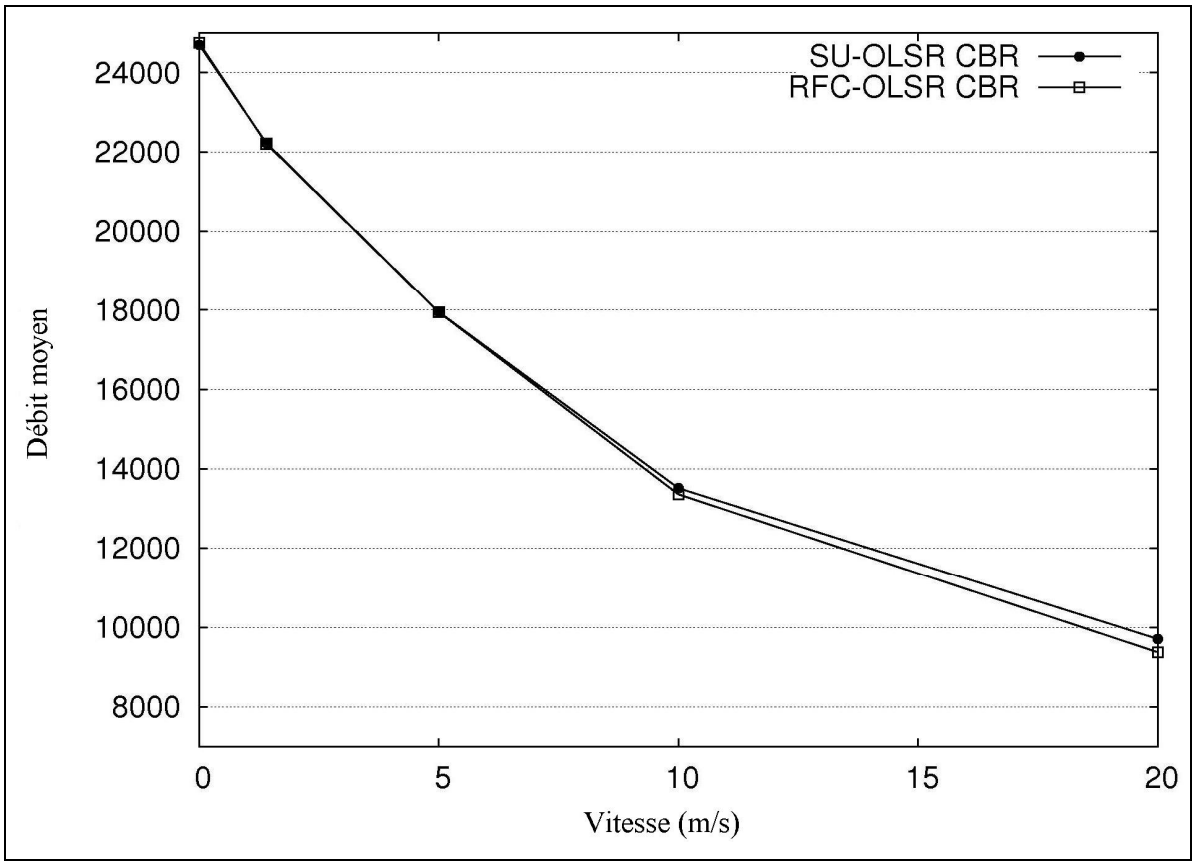

Figure 4.19 Nombre de messages délivrés.

CONCLUSION

Les réseaux Ad-Hoc sont des réseaux dynamiques et auto-configurables sans infrastructure préexistante. Ce type de réseaux n'a pas une politique claire pour séparer les nœuds légitimes de ceux non désirés ou malicieux. Un nœud légitime ou malicieux pourrait se joindre sans distinction à un réseau Ad-Hoc. Ainsi, à cause de la présence des nœuds malicieux, l'un des plus grands défis dans ce type de réseaux est de proposer des solutions de sécurité robustes pouvant protéger ces réseaux contre les différentes attaques.

Dans ce travail de recherche, nous nous sommes intéressés aux problèmes de sécurité et aux attaques contre le protocole OLSR. Plus précisément, nous avons cherché à proposer une nouvelle solution contre les attaques par mystification des liens où un nœud malicieux oblige ses voisins à le choisir comme relais multipoint (MPR). Comme résultat, nous avons proposé un nouveau protocole SU-OLSR dérivé d'OLSR qui présente moins de vulnérabilités.

Nos travaux ont débuté par l'étude des vulnérabilités et des attaques contre OLSR et en particulier ceux par mystification des liens. À l'issue de cette analyse, nous avons constaté la problématique lorsqu'un nœud malicieux force ses voisins à le choisir comme MPR. Pour se faire, il suffit au nœud malicieux de déclarer qu'il a un lien direct avec un nœud distant ou inexistant dans le réseau. Une fois choisi comme MPR, ce nœud aura un avantage dans le réseau du fait qu'il peut modifier, altérer ou rejeter le trafic qui transite par lui. Ceci présente un danger pour l'intégrité, la confidentialité et la disponibilité des communications.

Afin de surpasser cette limitation du protocole OLSR, nous avons proposé le nouveau protocole SU-OLSR. Ce nouveau protocole est plus sélectif pour les MPR. SU-OLSR empêche tout nœud malicieux qui présente certains comportements suspects d'être choisi comme MPR. Nous avons défini deux critères pour rejeter un nœud malicieux. Le premier critère est lorsqu'un nœud couvre un nœud isolé. Le deuxième est lorsqu'un nœud déclare qu'il couvre plus qu'une portion prédéfinie de ses voisins. Ainsi, un nœud qui présente l'un de ces critères, sera déclaré comme suspect et sera rejeté automatique de la phase de sélection

des MPR par SU-OLSR. Pour sa part, le calcul de chemin se fait selon quatre options. L'algorithme de Dijkstra soit utilise les MPR déclarés sécuritaires par tous les nœuds, soit utilise sans distinction les MPR déclarés sécuritaires par tous ou certains nœuds, soit utilise de préférence les MPR déclarés sécuritaires par tous les nœuds ou finalement ne pas utiliser un lien déclaré non sécuritaire par certains nœuds dans le dernier saut pour compléter le chemin vers la destination.

Avec cette approche, certains nœuds légitimes peuvent présenter malheureusement l'un de ces critères. Ceci les exclus de la phase de sélection des MPR, ce qui peut avoir un impact sur la connectivité du réseau. Nous avons donc cherché, comme deuxième phase, à étudier expérimentalement notre approche et prouver que le nouveau protocole SU-OLSR présente des performances comparables à ceux du protocole OLSR classique malgré que SU-OLSR soit plus sélectif.

Nous avons commencé notre évaluation expérimentale par le développement d'un programme en C pour simuler les deux protocoles SU-OLSR et OLSR dans un environnement sans mobilité. Ces simulations nous ont permis d'avoir une vision sur les performances de SU-OLSR par rapport à OLSR en ce qui concerne le nombre de MPR choisis et la longueur du plus court chemin entre un nœud source et un nœud destination. Ces résultats encourageants ont montré que les performances de ces deux protocoles sont très proches.

Nous avons par la suite passé à une implémentation plus proche des conditions réelles sous le simulateur *ns-2* afin de surpasser les limitations de notre implémentation statique. Après la prise en main de l'outil de simulation et de la réalisation de scénarios simples, nous avons développé une version du protocole SU-OLSR pour *ns-2*. Comme objectifs, nous avons voulu comparer les performances des deux protocoles dans un environnement avec différents scénarios de mobilité et de rayon de communication. À cause de la densité élevée du réseau lors de nos simulations et certaines limitations du simulateur, nous avons rencontré des contraintes relatives à la vitesse d'exécution des simulations et la taille très grandes de nos

fichiers trace. Nous avons réussi à contourner le problème en optimisant certains paramètres dans nos scripts de simulation. À l'aide de divers outils que nous avons développés, nous avons exploité par la suite les traces de nos simulations. Plus précisément, nous avons analysé les performances de SU-OLSR et OLSR en ce qui concerne le nombre de MPR choisis, le délai moyen de transmission de bout-en-bout et le pourcentage de paquets délivrés. Les résultats obtenus sont comparables pour les deux protocoles si le réseau présente une densité suffisamment grande. Le nouveau protocole affiche même de meilleures performances pour les scénarios à grand mobilité. Tout ceci montre que le protocole SU-OLSR donne de bonnes performances malgré qu'il soit sélectif pour les MPR.

Grâce à tout ceci, nous avons montré que notre contribution offre une bonne solution pour faire face aux attaques par mystification des liens où un nœud malicieux cherche à forcer ses voisins à le choisir comme relais multipoint. Ce nouveau protocole ne remplacera pas les solutions basées sur la cryptographie mais il offre un mécanisme complémentaire pour garantir une meilleur sécurité.

Notre approche ouvre plusieurs perspectives de recherche future. En effet, un certain nombre de nouvelles solutions ont été mises à jour dans notre travail mais leurs études n'ont pas été exhaustives. Nous avons montré et expliqué les performances du SU-OLSR avec l'*Option II-b* et *II-c* grâce à des simulations dans un environnement statique. L'évaluation expérimentale de ces deux options dans un environnement avec mobilité, nous apparaît comme un premier axe de travail dans la continuité de ce travail de recherche. En effet, dans le cas de *l'Option II-b*, les MPR déclarés sécuritaires par tous les nœuds sont préférés par rapport à ceux déclarés sécuritaires par seulement certains nœuds. Afin de mettre en place ce mécanisme sous *ns-2*, l'algorithme de calcul de plus court chemin dans le package UM-SU-OLSR devrait être modifié pour permettre la prise en compte seulement des MPR sécuritaires lors de construction des routes. Pour sa part, l'*Option II-c* nécessite aussi le changement de l'algorithme de calcul de route.

Le deuxième axe de recherche concerne la mise en place du mécanisme de contre-mesure présenté dans le modèle d'attaque de SU-OLSR. Ce mécanisme permet à un nœud de dénoncer un nœud malicieux auprès des autres nœuds. L'existence de mécanismes semblables (Vilela et Barros, 2007) devrait faciliter la mise en place de notre approche de contre-mesure.

Finalement, le dernier axe de recherche est d'examiner l'interaction entre les deux critères de SU-OLSR utilisés dans nos travaux. En d'autres termes, évaluer les performances du protocole SU-OLSR lorsqu'on sélectionne juste les MPR qui ne couvrent pas les nœuds isolés et en même temps qui ne couvrent pas plus qu'une fraction fixée de nœuds voisins.

ANNEXE I

SCRIPT DE SIMULATION

```
# ======================================================================
# SU-OLSR & OLSR simulation script for ns-2 "sim-SU-OLSR.tcl"
# Rachid Abdellaoui & Jean-Marc Robert
# École de Technologie Supérieure
# Copyright (c) 2009
# All rights reserved.
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
# 1. Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in the
# documentation and/or other materials provided with the distribution.
# 3. Neither the name of the University nor of the Laboratory may be used
# to endorse or promote products derived from this software without
# specific prior written permission.
# ======================================================================

# ======================================================================
# Initialization
# ======================================================================
if {$argc != 5} {
puts "Usage: ns sim-SU-OLSR.tcl \[protocol\] \[radius\] \[scenario nbr\]
\[traffic nbr\] \[speed nbr\]""
   exit
}
if {$argc == 5} {
    set protocol   [lindex $argv 0]
    set radius     [lindex $argv 1]
    set scenarioNb [lindex $argv 2]
    set trafficNB  [lindex $argv 3]
    set speedNode  [lindex $argv 4]
}
# ======================================================================
# Get more performance for our simulations
# ======================================================================
remove-all-packet-headers
add-packet-header ARP Common IP LL Mac TCP OLSR
#add-packet-header OLSR Diffusion LL MAC CBR IP
#add-packet-header OLSR IP Diffusion CBR
#CBR Flags IP TCP Message Diffusion LL SR

# ======================================================================
# Configurable options
# ======================================================================
set namtrace_s           y  ;# set to "n" in order to not get a nam trace
```

```
set opt(ifqlen)          50  ;# max packet in ifq
set opt(nn)             100  ;# number of mobilenodes
set opt(cp)             "/home/rachid/sim50/scen-trafic/traffic/cbr-
1000x1000-100-0-$trafficNb.tcl"     ;# connection pattern file CBR
set opt(sc)   "/home/rachid/sim50/scen-trafic/RWmobility300s10ms/RWmov-
1000-300sec-$speedNode-$scenarioNb.tcl" ;# node movement file
set opt(x)              1000         ;# x coordinate of topology
set opt(y)              1000         ;# y coordinate of topology
set opt(stop)           300.0        ;# time to stop simulation
set opt(seed) 12345
#set opt(MacRate) 2e6
#set traffic      CBR
# ======================================================================
# Define options
# ======================================================================
set opt(chan)     Channel/WirelessChannel     ;# channel type
set opt(prop)     Propagation/TwoRayGround    ;# radio-propagation model
set opt(netif)    Phy/WirelessPhy             ;# network interface type
set opt(mac)      Mac/802_11                  ;# MAC type
set opt(ifq)      Queue/DropTail/PriQueue     ;# interface queue type
set opt(ll)       LL                          ;# link layer type
set opt(ant)      Antenna/OmniAntenna         ;# antenna model
set opt(adhocRouting)   $protocol       ;# SU-OLSR or OLSR routing protocol
Mac set bandwidth_ 2Mb
Mac/802_11 set dataRate_ 11Mb
#Mac/802_11 set basicRate_ 2Mb
Phy/WirelessPhy set bandwidth_ 11Mb
# ======================================================================
# OLSR global agent configuration (default values of OLSR Protocol)
# ======================================================================
Agent/OLSR set hello_ival_ 1
Agent/OLSR set tc_ival_ 2
Agent/OLSR set use_mac_    true
Agent/OLSR set debug_      false
Agent/OLSR set debug_      false
Agent/OLSR set willingness 3
Agent/OLSR set hello_ival_ 2
Agent/OLSR set tc_ival_    5

global defaultRNG
$defaultRNG seed $opt(seed)
# ======================================================================
# Remove and create result directory
# ======================================================================
set dirNameTr "all-results/su-result/$speedNode-300s-SUM-$scenarioNb"
exec sh -c " mkdir -p $dirNameTr"
if { $namtrace_s == "y" } {
set dirNameAn "all-results/su-result/Na-$speedNode-300s-SUM-$scenarioNb"
exec sh -c " mkdir -p $dirNameAn"
# exec sh -c " chmod 755 $dirNameAn"
}
```

```
# ======================================================================
# Calcul Communication range for antenna
# ======================================================================
if {$radius ==  190} {
Phy/WirelessPhy set RXThresh_ 1.09484e-09
}
if {$radius ==  230} {
Phy/WirelessPhy set RXThresh_  5.09863e-10
}
if {$radius ==  250} {
Phy/WirelessPhy set RXThresh_  3.65262e-10
}
if {$radius ==  290} {
Phy/WirelessPhy set RXThresh_  2.01731e-10
}
if {$radius ==  330} {
Phy/WirelessPhy set RXThresh_ 1.20312e-10
}
if {$radius ==  390} {
Phy/WirelessPhy set RXThresh_  6.16746e-11
}
if {$radius ==410} {
Phy/WirelessPhy set RXThresh_ 5.04928e-11
}
# ======================================================================
# Create simulator instance
# ======================================================================
set ns_ [new Simulator]
set tracefd [open $dirNameTr/SU$radius.tr w]
if { $namtrace_s == "y" } {
      # initialize a namtrace file for logging node movements to
      # be viewed in nam (must be called after mobility is defined)
      set namtrace [open $dirNameAn/$radius.nam w]
      $ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
}
$ns_ trace-all $tracefd
$ns_ color 0 red
$ns_ color 1 blue
set tracefd [open $dirNameTr/SU$radius.tr w]

# ======================================================================
# Create topography object
# ======================================================================
set topo [new Topography]

# ======================================================================
# Define topology
# ======================================================================
$topo load_flatgrid $opt(x) $opt(y)
# ======================================================================
# Create God
# ======================================================================
      # god is used to store an array of the shortest number of
      # hops required to reach from one node to another
set god_ [create-god $opt(nn)]
```

```
# ============================================================================
# Create channel
# ============================================================================
set channel_ [new $opt(chan)]
# ============================================================================
# configure mobile nodes
# ============================================================================
$ns_ node-config \
                 -adhocRouting $opt(adhocRouting) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channel $channel_ \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF \
                 -movementTrace ON
# ============================================================================
# Create & Place nodes
# ============================================================================
for {set i 0} {$i < $opt(nn)} {incr i} {
    puts $i
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0  ;# enable random motion
}
# ============================================================================
# Define initial node position in nam
# ============================================================================
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 20
    $node_($i) color Red
}
# ============================================================================
# Source connection-pattern and node-movement scripts
# ============================================================================
if { $opt(cp) == "" } {    puts "*** NOTE: no connection pattern
specified."
    set opt(cp) "none"
} else {
    puts "Loading connection pattern..."
    source $opt(cp)
}
if { $opt(sc) == "" } {
    puts "*** NOTE: no scenario file specified."
    set opt(sc) "none"
} else {
    puts "Loading scenario file..."
    source $opt(sc)
    puts "Load complete..."
}
```

```
for {set i 0} {$i < $opt(nn) } {incr i} {
      $node_($i) start
}
# ============================================================================
# Print (in the trace file) MPR Set and other internal data
# ============================================================================
for {set i 0} {$i <= $opt(stop)} {incr i} {
   for {set j 0} {$j < $opt(nn)} {incr j} {
      $ns_ at $i "[$node_($j) agent 255] print_mprset"
#     $ns_ at $i "[$node_($j) agent 255] print_nbset"
#     $ns_ at $i "[$node_($j) agent 255] print_nb2hopset"
#     $ns_ at $i "[$node_($j) agent 255] print_rating_table"
#     $ns_ at $i "[$node_($j) agent 255] print_mprselset"
#     $ns_ at $i "[$node_($j) agent 255] print_nb2hopset"
#     $ns_ at $i "[$node_($j) agent 255] print_topologyset"
#     $ns_ at $i "[$node_($j) agent 255] print_rtable"
   }
}
# ============================================================================
# Tell all nodes when the simulation ends
# ============================================================================
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop).0 "$node_($i) reset";
}
# ============================================================================
# Finishing procedure
# ============================================================================
proc finishSimulation { } {
      global ns_ node_ null_ opt tracefd namtrace namtrace_s
      $ns_ flush-trace
      close $tracefd

      if { $namtrace_s == "y" } {
            close $namtrace
      }
      # Exit
       puts "Finished simulation."
       $ns_ halt
      exit 0
}
# ============================================================================
# Run the simulation
# ============================================================================
proc runSimulation { } {
    global ns_ finishSimulation opt
    for {set j 1.0} {$j < $opt(stop)} {set j [expr $j * 1.03 ]} {
      $ns_ at $j "puts t=$j"
    }
    $ns_ at $opt(stop) "finishSimulation"
    $ns_ run
}
puts "Starting Simulation..."
runSimulation
# ============================================================================
# ============================================================================.
```

BIBLIOGRAPHIE

3GPP. 2008. *3GPP Specification series*. <http://www.3gpp.org/ftp/Specs/html-info/36-series.htm>.

Abusalah, L., A. Khokhar et M. Guizani. 2006. « Trust Aware Routing in Mobile Ad Hoc Networks ». In *In Proceedings of the 49th IEEE Global Telecommunications Conference, GLOBECOM '06*. p. 1-5.

Adjih, C., T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler et D. Raffo. 2003. « Securing the OLSR protocol ». In *Proceedings of the IEEE Med-Hoc-Net* (Tunisia).

Adjih, C., T. Clausen, A. Laouiti, P. Mühlethaler et D. Raffo. 2005. « Securing the OLSR routing protocol with or without compromised nodes in the network ». *INRIA Research Repport RR-5494*.

Adjih, C., P. Jacquet et L. Viennot. 2002. *Computing Connected Dominated Sets with Multipoint Relays*. Coll. « INRIA Technical report RR-4597 ». INRIA

Adjih, C., P. Mühlethaler et D. Raffo. 2006. « Detailed specifications of a security architecture for OLSR ». *Rapport de Recherche no. 5593. INRIA*.

Adjih, C., D. Raffo et P. Mühlethaler. 2005. « Attacks Against OLSR: Distributed Key. Management for Security ». *In 2005 OLSR Interop and Workshop, École Polytechnique, France*.

Adnane, A., C. Bidan et R. T. de Sousa Jr. 2008. « Validation of the OLSR routing table based on trust reasoning ». *In Proceedings of the International Workshop on Trust in Mobile Environments*.

Adnane, A., R. T. de Sousa Jr, C. Bidan et .L Mé. 2008. « Autonomic trust reasoning enables misbehavior detection in OLSR ». *In Proceedings of the 23rd Annual ACM Symposium on Applied Computing (ACM SAC 2008)*, p. 2006-2013.

Anjum, .F, et .P Mouchtaris. 2007. *Security for Wireless Ad-Hoc Networks*. Wiley-Interscience.

Awerbuch, B., R. Curtmola, D. Holmer et C. Nita-Rotaru. 2004. « Mitigating Byzantine Attacks in Ad Hoc Wireless Networks ». *Technical Report v1, Archipelago project*.

Badis, H., A. Munaretto, K. Al Aghal et G. Pujolle. 2004. « Optimal path selection in a link state QoS routing protocol ». In *In Proceedings of the 59th IEEE Vehicular Technology Conference VTC 2004-Spring*. Vol. 5, p. 2570-2574.

Bellman, R. E. 1957. *Dynamic Programming*. Princeton: Princeton University Press.

Bishop, M. 2005. *Introduction to Computer Security*. Addison Wesley Professional.

Broch, J., D. B. Johnson et D. A. Maltz. 2002. « The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks ». *draft-ietf-manet-dsr-07.txt, IETF MANET, Internet Draft*.

Brown, W. W., V. Iv Marano, W. H. MacCorkell et T. Krout. 2003. « Future combat system-scalable mobile network demonstration performance and validation results ». In *IEEE Military Communications Conference, 2003 (MILCOM2003)*. . Vol. 2, p. 1286-1291.

Bruno, R., M. Conti et E. Gregori. 2005. « Mesh networks: commodity multihop ad hoc networks ». *IEEE Communications Magazine* vol. 43, n° 3, p. 123-131.

Busson, A., N. Mitton et E. Fleury. 2005. « Analysis of the Multi-Point Relays selection in OLSR and Implications ». In *Proceedings of fourth annual Med-Hoc Netorking*. p. 387-396.

Camp, T., J. Boleng et V. Davies. 2002. « A survey of mobility models for ad hoc network research ». *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*. Vol. 2, n° 5.

Chenxi, Z., L. Rongxing, L. Xiaodong, H.Pin-Han et S. Xuemin. 2008. « An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks ». In *In Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2008)*. p. 246-250.

Clausen, T., et E. Baccelli. 2005. « Securing OLSR Problem Statement ». *IETF INTERNET-DRAFT, draft-clausen-manet-solsr-ps-00.txt*.

Clausen, T., et P. Jacquet. 2003. « RFC3626 : Optimized Link State Routing Protocol (OLSR) ».

Clausen, T., P. Jacquet et L. Viennot. 2005. « Comparative Study of Routing Protocols for Mobile Ad-hoc Networks. ». In *Procceding of First Annual Mediterranean Ad Hoc Networking Workshop*. p. 10.

CMU Monarch Project. 1999. *CMU Monarch project*. Computer Science Department, Canergie Mellon University, Pittsburgh,

Cormen, T. H., R. L. Rivest, C. E. Leiserson et C. Stein. 2001. *Introduction to Algorithms*. Coll. « MIT Press and McGraw-Hill ».

Corson, S., et J. Macker. 1999. *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*. IETF RFC2501

DARPA/NSF. 2008. *The network simulator ns-2* <http://www.isi.edu/nsnam/ns>.

The Defense Advanced Research Projects Agency. <http://www.darpa.mil/>.

Dhillon, D., T. S. Randhawa, M. Wang et L. Lamont. 2004. « Implementing a fully distributed certificate authority in an OLSR MANET ». In *The IEEE Wireless Communications and Networking Conference, 2004. WCNC.* Vol. 2, p. 682-688.

Dijkstra, E. W. 1959. « A note on two problems in connecxion with graphs ». *Numerishe Mathematic*. Vol. 1, p. 269-271.

Fall, .K, et .K Varadhan. 2008. « The *ns* Manual ». The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC. <www.isi.edu/nsnam/ns/doc/>.

FBCB2. 2008. *Force XXI Battlefield Command Brigade and Below*. <http://peoc3t.monmouth.army.mil/fbcb2/fbcb2.html>.

Fifer, W. C., et F. J. Bruno. 1987. « The low-cost packet radio ». *Proceedings of the IEEE,* vol. 75, n^{o} 1, p. 33-42.

Ford, L. R., et Fulkerson. 1962. *Flows in networks*. Princeton: Princeton University Press.

Freebersyser, J. A., et B. Leiner. 2001. « A DoD perspective on mobile Ad hoc networks ». In *Ad hoc networking; C.E Perkins Ed.* p. 29-51. Addison-Wesley Longman Publishing Co., Inc.

Friis, H. T. 1946. « A Note on a Simple Transmission Formula ». *Proceedings of the IRE,* vol. 34, n^{o} 5, p. 254-256.

Ge, Y., T. Kunz et L. Lamont. 2003. « Quality of service routing in ad-hoc networks using OLSR ». In *In Proceedings of the 36th Annual Hawaii International Conference onSystem Sciences* p. 9.

Haas, Z. J., M.R. Pearlman et P. Samar. 2002a. « The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks ». *draft-ietf-manet-zone-brp-02.txt, IETF MANET, Internet Draft*.

Haas, Z. J., M.R. Pearlman et P. Samar. 2002b. « The Interzone Routing Protocol (IERP) for Ad Hoc Networks ». *draft-ietf-manet-zone-ierp-02.txt, IETF MANET, Internet Draft*.

Haas, Z. J., M.R. Pearlman et P. Samar. 2002c. « The Intrazone Routing Protocol (IARP) for Ad Hoc Networks ». *draft-ietf-manet-zone-iarp-02.txt, IETF MANET, Internet Draft*.

Hass, Z.J., M.R. Pearlman et P. Samar. 2002. « The Zone Routing Protocol (ZRP) for Ad Hoc Networks ». *draft-ietf-manet-zone-zrp-04.txt, IETF MANET, Internet Draft*

Hogie, L., P. Bouvry et F. Guinand. 2006. « An Overview of MANETs Simulation ». *Electronic Notes in Theoretical Computer Science,* vol. 150, n° 1, p. 81-101.

Hu, Y. C., A. Perrig et D. B. Johnson. 2003. « Packet leashes: a defense against wormhole attacks in wireless networks ». In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, NFOCOM 2003*. Vol. 3, p. 1976-1986.

ITMANET. 2008. <http://www.darpa.mil/ipto/programs/itmanet/itmanet.asp>.

Jacquet, P., P. Minet, P. Mühlethaler et N. Rivierre. 1997. « Increasing Reliability in Cable-Free Radio LANs Low Level Forwarding in HIPERLAN ». *Wireless Personal Communications* vol. 4, n° 1, p. 51-63.

Janert, P. 2008. *Gnuplot in Action*, EARLY Access Edition. EARLY Access Edition. <http://www.gnuplot.info/>.

Jubin, J., et J. D. Tornow. 1987. « The DARPA packet radio network protocols ». *Proceedings of the IEEE,* vol. 75, n° 1, p. 21-32.

Kannhavong, B., H. Nakayama, N. Kato, Y. Nemoto et A. Jamalipour. 2006. « Analysis of the node isolation attack against OLSR-based mobile ad hoc networks ». In *In Proceedings of the International Symposium on Computer Networks* p. 30-35.

Karmakar, G., et L. S. Dooley. 2008. *Mobile Multimedia Communications: Concepts, Applications, and Challenges*. IGI Publishing.

Kurkowski, S., T. Camp et M. Colagrosso. 2005. « MANET simulation studies: the incredibles ». *Mobile Computing and Communications Review SIGMOBILE,* vol. 9, n° 4, p. 50-61.

Lang, D. 2008. *Routing Protocols for Mobile Ad-Hoc Networks Classification, Evaluation and challenges*. VDM Verlag.

Laouiti, A., A. Qayyum et L. Viennot. 2000. « Multipoint Relaying: An eficient Technique for flooding in mobile wireless networks ». *Rapport de recherche INRIA RR-3898*.

Leiner, B. M., R. J. Ruther et A. R. Sastry. 1996. « Goals and challenges of the DARPA GloMo program [global mobile information systems] ». *IEEE Personal Communications,* vol. 3, n° 6, p. 34-43.

Lipman, J., P. Boustead, J. Chicharo et J. Judge. 2003. « Resource aware information dissemination in ad hoc networks ». In *In Proceedings of the 11th IEEE International Conference on Networks ICON2003* p. 591-596. Australia.

Lipman, J., P. Boustead et J. Judge. 2002. « Utility-based Multipoint Relay Flooding in Heterogeneous Mobile Ad hoc Networks ». In *In Proceedings of the Workshop on the Internet Telecommunications and Signal Processing (WITSP 2002)*. Australia.

M. Barbeau, M., et E. Kranakis. 2007. *Principles of Ad Hoc Networking*. Wiley.

Mans, B., et N. Shrestha. 2004. « Performance Evaluation of Approximation Algorithms for Multipoint Relay Selection ». In *In Proceedings of the 3rd Annual Mediterranean Ad-Hoc Network Workshop Med-Hoc-Net*. p. 480-491.

Marsh, S. 1994. « Formalising Trust as a Computational Concept ». University of Stirling.

Matousek, J. , J. Nesetril et D. Hachez. 2004. *Introduction Aux Mathématiques Discrètes*. Springer.

Mishra, .A. 2008. *Security and Quality of Service in Ad-Hoc Wireless Networks*. Cambridge.

Nait-Abdesselam, F., B. Bensaou et J. Yoo. 2007. « Detecting and Avoiding Wormhole Attacks in Optimized Link State Routing Protocol ». In *In Proceedings of the IEEE Wireless Communications and Networking Conference WCNC 2007*. p. 3117-3122.

NowWireless. 2008. <http://www.nowwireless.com/>.

Palchaudhuri, S., J. Y. Le Boudec et M. Vojnovic. 2005. « ns-2 Code for Random Random waypoint and Trip Mobility Model ». <http://www.cs.rice.edu/~santa/research/mobility/>.

Palmen, F. M., T. Tielert, M. D. Kamdoum, W. H. Lauppe, L. Pan et F. Seita. 2006. « Vehicular Ad-Hoc Networks : Technical Reports ». In, sous la dir. de DSN, Decentralized Systems and Network Services Research Group.

Papadimitratos, P., et Z. J. Haas. 2002. « Secure Routing for Mobile Ad Hoc Networks ». In *CS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*. San Antonio, TX.

Penrose, M. 1999. « On k-connectivity for a geometric random graph ». *Random Structures and Algorithms,* vol. 15, n° 2, p. 145-164.

Penrose, M. 2003. *Random Geometric Graphs*. Coll. « Oxford Studies in Probability ». Oxford University Press.

Perkins, C. E., et P. Bhagwat. 1994. « Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers ». *ACM SIGCOMM Computer Communication Review,* vol. 24, n° 4, p. 234-244.

Perkins, C. E., E. M. Royer et S. Das. 2002. « Ad-hoc On-demand Distance Vector (AODV) Routing ». *draft-ietf-manet-aodv-10.txt, IETF MANET, Internet Draft*.

Perlman, R. 2000. *Interconnections* ADDISON-WESLEY.

Ramaswamy, S., H. Fu, M. Sreekantaradhya, J. Dixon et K. Nygard. 2003. « Prevention of cooperative black hole attack in wireless ad hoc networks ». In *In Proceedings of 2003 International Conference on Wireless Networks (ICWN'03)*. p. 570-575.

Roy, J. F. 2006. *UM-OLSR*. University of Murcia: MASIMUM. <http://masimum.dif.um.es/?Software:UM-OLSR>.

Sanzgiri, K., B. Dahill, B. N. Levine, C. Shields et E. M. Belding-Royer. 2002. « A secure routing protocol for ad hoc networks ». In *Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP'02)*. p. 78-87.

Sarkar, S., T. Besavaraju et C. Puttamadappa. 2008. *Ad-Hoc Mobile Wireless Networks*. Auerbach.

Shrestha, N. 2003. *Performance Evaluation of Multipoint Relays: Collision and Energy efficiency Issues*. Coll. « Technical Report ». Macquarie University: Macquarie University.

Sony. 2008. « PSP Technical Specifications ». Sony Computer Entertainment.

Suraci, F. J., A. R. Ephrath et J. R. Wullert. 2007. « Global interoperability of national security and emergency preparedness (NS/EP) telecommunications services ». In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*. p. 403-410.

Viennot, L. 1998. « Complexity Results on Election of Multipoint Relays in Wireless Networks ». *Rapport de recherche INRIA RR-3584*.

Vilela, J. P., et J. Barros. 2007. « A feedback reputation mechanism to secure the optimized link state routing protocol ». In *In Proceedings of the third International Conference on Security and Privacy in Communications Networks and the Workshops, SecureComm 2007*. p. 294-303.

Villasenor-Gonzalez, L., G. Ying et L. Lament. 2005. « HOLSR: a hierarchical proactive routing mechanism for mobile ad hoc networks ». *IEEE Communications Magazine,* vol. 43, n° 7, p. 118-125.

Voorhaen, M., et C. Blondia. 2006. « Analyzing the Impact of Neighbor Sensing on the Performance of the OLSR protocol ». In *4th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*. p. 1-6.

Wang, S., J. Wang, X. Zhang et J. Wei. 2006. « Performance of anti-jamming ad hoc networks using directional beams with group mobility ». In *IFIP International Conference on Wireless and Optical Communications Networks,* . p. 4 pp.

Wu, J., Lou Wei et F. Dai. 2006. « Extended multipoint relays to determine connected dominating sets in MANETs ». *In Proceedings of the IEEE Computers* vol. 55, n° 3, p. 334-347.

Xu, N. 2002. « A Survey of Sensor Network Applications ». *IEEE Communications Magazine,* vol. 40, n° 8, p. 102–114.

Yi, Q., et N. Moayeri. 2008. « Design of Secure and Application-Oriented VANETs ». In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. p. 2794-2799.

Zapata, M. G. 2002. « Secure ad hoc on-demand distance vector routing ». *ACM Mobile Comuting and Communications Review SIGMOBILE,* vol. 6, n° 3, p. 106-107.