Analyse et simulation du routage dans un réseau ad hoc

par

Sana JGUIRIM

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE
M. Sc. A.

MONTRÉAL, LE 29 OCTOBRE 2018

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. François Gagnon, Directeur de Mémoire
Département de génie électrique à l'École de technologie supérieure

M. Naim Batani, Président du Jury
Département de génie électrique à l'École de technologie supérieure

M. Ammar Kouki, membre du jury
Département de génie électrique à l'École de technologie supérieure

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

LE 16 OCTOBRE 2018

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens, tout d'abord, à remercier mon directeur de recherche, le Professeur François Gagnon, pour ses conseils, son orientation et son appui. Je remercie également les professeurs Naim Batani et Ammar Kouki d'avoir accepté d'évaluer mon mémoire.

J'aimerai remercier aussi l'assistant de recherche Mohammed Ahmed pour sa disponibilité et ses conseils. Mes remerciements vont aussi à mes très chers amis au Canada pour leur présence et leurs soutien moral. Je pense surtout à Dorra, Ismail, Nesrine, Fatma, Amina et Madiha. Vous étiez toujours une famille pour moi.

Enfin, en témoignage de mon amour et de ma reconnaissance pour tous les sacrifices auxquels je dois ma réussite, je dédie ce mémoire à mon cher père Sadok, ma chère mère Amna, mon frère Zied et le reste de ma famille. Malgré la longue distance, votre soutien moral sans relâche, votre patience et votre amour m'ont accompagné tout au long de cette expérience. Je pense également à Mr kais Bouzouita, sa femme Boutheyna et madame Kewther Bouzouita pour m'avoir aidé dans mon installation au Canada.

ANALYSE ET SIMULATION DU ROUTAGE DANS UN RÉSEAU AD HOC

Sana JGUIRIM

RÉSUMÉ

La technologie ad hoc est parmi les technologies les plus actives dans le domaine de télécommunication. Son déploiement a commencé depuis les années 1970 et continue jusqu'à aujourd'hui en raison de la flexibilité et la robustesse offertes. Cette technologie est le sujet principal de plusieurs travaux de recherche qui essayent de proposer des approches visant à améliorer le déploiement du réseau ad hoc. L'absence d'une administration centralisée et son remplacement par l'approche distributive diminuent énormément le coût de déploiement dans le plan réel. Chaque nœud est responsable et autonome dans ses décisions. Le niveau de routage fait partie des niveaux critiques durant l'exploitation de cette technologie. Vu le dynamisme caractérisant la technologie ad hoc et causé par les mouvements aléatoires des noeuds, un protocole de routage adéquat doit posséder certaines caractéristiques. La possession des mécanismes de contrôle et de maintenance capables d'assurer une configuration valide et adéquate aux circonstances du réseau est primordiale pour un protocole de routage dans le réseau ad hoc. La rapidité dans la recherche du trajet optimal et la disponibilité des informations caractérisent l'efficacité du protocole de routage. Plusieurs approches de routage ont été proposées afin de garantir un routage fiable dans un environnement dynamique. Dans ce contexte, le choix de protocole de routage influence les performances résultantes dans le réseau entier. Durant ce projet, nous avons approfondi le concept de routage dans le réseau ad hoc en détaillant les mécanismes les plus répandus. Une comparaison qualitative a été élaborée et des résultats qualitatifs sont présents afin de garantir le choix convenable du protocole de routage. Une description précise des mécanismes et des algorithmes utilisés par le protocole choisi vise à éclaircir les futurs usagers sur les différentes étapes de la réalisation. Enfin, une évaluation du protocole choisi a été effectuée par l'analyse des performances données dans le réseau entier.

Mots-clés: réseau ad hoc, protocole de routage, auto-organisation

ANALYSIS AND SIMULATION OF ROUTING IN AD HOC NETWORK

Sana JGUIRIM

ABSTRACT

Ad hoc Technology is one of the most efficient technology in the wireless communication, because of its robustness and its flexibility. Its deployment has been ongoing for nearly 40 years. Many researchs try to propose new approaches to improve the real deployment. Every node in ad hoc network can be autonomous and responsible to act as a router. Without a fixed infrastructure and the arbitrary movement of nodes, the task of routing becomes a critical one. A good routing protocol must have control and maintenance mechanisms to be able to guarantee the valid configuration. The ability to find out the optimal path rapidly and the necessary informations availability are requested for a proper ad hoc routing protocol. So the choice of routing protocol affects the performance in the network. In this project, we specify the most popular routing mechanisms for the ad hoc network. To find out the proper choice, a qualitative comparison and results are introduced. A description of mechanisms and algorithms of the chosen protocol is presented. Finally, the routing protocol is integrated in a WIFI simulator and the results are discussed.

Keywords: ad hoc network, routing protocol, self-organize

TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 RÉSEAU SANS FIL AD HOC 3
1.1 Le réseau ad hoc 3
1.1.1 Définition du réseau ad hoc 3
1.1.2 Historique du réseau ad hoc 3
1.1.3 Le concept d'auto-organisation 4
1.1.4 Les classifications du réseau ad hoc 5
1.1.5 Les applications du réseau ad hoc 8
1.1.6 Les contraintes du réseau ad hoc 10
1.2 Le standard IEEE 802.11 12
1.3 Les protocoles de routage 14
1.3.1 Les protocoles proactifs 15
1.3.2 Les protocoles réactifs 18
1.3.3 Les protocoles hybrides 21

CHAPITRE 2 LE CHOIX DE PROTOCOLE DE ROUTAGE 25
2.1 Étude comparative des protocoles réactifs, proactifs et hybrides 25
2.1.1 Taux de paquets livrés 25
2.1.2 Délai de bout en bout 28
2.1.3 Volume du trafic de contrôle 30
2.1.4 Longueur de la route 31
2.1.5 Débit 32
2.1.6 Taux de perte de paquets 35
2.1.7 Conclusion 36
2.2 Le protocole OLSR 37
2.2.1 OLSRV2 41
2.2.1.1 Le message de contrôle de lien Hello 42
2.2.1.2 le message de contrôle de topologie 44
2.3 Conclusion 46

CHAPITRE 3 L'IMPLÉMENTATION DE NHDP ET D'OLSRV2 49
3.1 Implémentation de protocole NHDP 49
3.1.1 Implémentation de message de contrôle de lien Hello 49
3.1.2 La base de données de NHDP 50
3.1.3 L'organigramme de protocole NHDP 53
3.1.4 La réalisation sous forme de code de NHDP 56
3.2 L'implémentation de protocole OLSRV2 58
3.2.1 L'implémentation de message de contrôle de topologie 58
3.2.2 La base de données de topologie 59

3.2.3 La base de données des messages reçus 61
3.2.4 L'organigramme de la structure topologique 62
3.2.5 La réalisation sous forme de code de la structure topologique 64
3.2.5.1 Le graphe de topologie 65
3.2.6 Le tableau de routage 68
3.2.6.1 L'algorithme de routage 69

CHAPITRE 4 ÉVALUATION DE PROTOCOLE OLSRV2 75
4.1 Environnement de simulation 75
4.1.1 Les environnements de simulation existants 75
4.1.2 Le simulateur fourni 77
4.2 Les paramètres d'évaluation 78
4.3 Les paramètres de simulation 79
4.4 Les résultats de simulation 79

CONCLUSION ET RECOMMANDATIONS 89

BIBLIOGRAPHIE 92

LISTE DES TABLEAUX

Page

Tableau 2.1 Scénario1 : Nombre de noeuds augmente 36

Tableau 2.2 Scénario2 : Nombre de flux de trafic augmente 37

Tableau 2.3 Scénario3 : Trafic de contrôle augmente 37

Tableau 3.1 Uplet d'interface locale *(Local Interface Tuple)* 50

Tableau 3.2 Uplet des adresses éliminées d'interface (*Removed Interface Address Tuple*). 51

Tableau 3.3 Uplet de lien (*Link tuple*) 52

Tableau 3.4 Uplet de voisin à 2 sauts (*2 Hop Tuple*) 52

Tableau 3.5 Uplet de voisin *(Neighbor Tuple)* 53

Tableau 3.6 Uplet du voisin perdu (*Lost Neighbor Tuple*) 53

Tableau 3.7 Uplet de routeur distant annoncé *(Advertising Remote Router Tuple)* 59

Tableau 3.8 Uplet Topologique de routeur *(Router Topology Tuple)* 60

Tableau 3.9 Uplet topologique d'adresse routable *(Routable Address Topology Tuple)* 60

Tableau 3.10 Uplet du message reçu *(received tuple)* 61

Tableau 3.11 Uplet du message transmis *(Forwarded tuple)* 62

Tableau 3.12 Uplet de routage *(Routing Tuple)* 68

Tableau 4.1 Paramètres de simulation 79

Tableau 4.2 Variation du temps d'exécution en fonction du nombre de nœuds 80

Tableau 4.3 Variation du temps d'exécution en fonction du nombre de paquets 80

Tableau 4.4 Taux moyen d'utilisation de mémoire physique en fonction du nombre de nœuds 81

Tableau 4.5 Taux moyen d'utilisation de mémoire physique en fonction du nombre de paquets 81

LISTE DES FIGURES

Page

Figure 1.1 La communication multi-sauts et la communication à un saut. Tirée de (Frodigh *et al.*, 2000) 6

Figure 1.2 Une vue à hiérarchie du réseau ad hoc. Tirée de (Loo *et al.*, 2016) 7

Figure 1.3 La configuration hétérogène du réseau ad hoc. Tirée de (Loo *et al.*, 2016) 8

Figure 1.4 L'application militaire de la technologie ad hoc. Tirée de (Bisnik, 2007) 9

Figure 1.5 Un exemple de déploiement de WSN. Tirée de (Rashid & Husain, 2016) 10

Figure 1.6 Le problème du terminal caché et du terminal exposé. Tirée de (Murthy & Manoj, 2004) 11

Figure 1.7 Transmission des fragments avec SIFS. Tirée de (IEEE 802.11, 2012) 14

Figure 1.8 Transmission avec DIFS. Tirée de (Murthy & Manoj, 2004) 14

Figure 1.9 Diffusion à l'aide de MPR. Tirée de (Voorhaen & Blondia, 2006) 18

Figure 1.10 La procédure de demande de route pour DSR. Tirée de (Johnson *et al.*, 2001) 19

Figure 1.11 La procédure de recherche de route par AODV. Tirée de (Royer & Toh, 1999) 21

Figure 1.12 IERP et IARP selon le protocole ZRP Tirée de (Beijar, 2002) 22

Figure 2.1 Le format proposé du message Hello dans la RFC 3626. Tirée de RFC 3626 39

Figure 2.2 Le format proposé du message MID dans la RFC 3626. Tirée de RFC 3626 39

Figure 2.3 Le format proposé du message TC dans la RFC 3626. Tirée de RFC 3626 40

Figure 2.4 Le format du message Hello dans la RFC 6130. Tirée de RFC 6130 44

Figure 2.5 Le format de message de contrôle de topologie dans la RFC 7181. Tirée de RFC 7181 46

Figure 3.1 La fonction générant le message Hello 49

Figure 3.2 La sous-fonction générant les blocs TLV à adresse dans le message Hello 50

Figure 3.3 L'organigramme de protocole NHDP 54

Figure 3.4 L'organigramme de mise à jour de l'ensemble de liens 55

Figure 3.5 Les fonctions de NHDP 56

Figure 3.6 La transmission du message Hello du nœud 1 vers nœud 2 57

Figure 3.7 La mise à jour de l'ensemble de voisins du nœud récepteur 57

Figure 3.8 La mise à jour de l'ensemble de liens du nœud récepteur 57

Figure 3.9 La fonction générant le message de contrôle de topologie 58

Figure 3.10 La sous-fonction générant les blocs TLV à adresse dans le message de contrôle de topologie 58

Figure 3.11 L'organigramme de la structure topologique 63

Figure 3.12 La machine d'état de la mise à jour d'un uplet de la structure topologique 64

Figure 3.13 Les fonctions de la structure topologique 65

Figure 3.14 Le premier type d'arête dans le graphe de topologie 66

Figure 3.15 Le deuxième type d'arête dans le graphe de topologie 66

Figure 3.16 Le troisième type d'arête dans le graphe de topologie 66

Figure 3.17 Le graphe de topologie de nœud1 67

Figure 3.18 Un exemple d'uplet sauvegardé dan l'ensemble *router topology set* du nœud1. 68

Figure 3.19 La fonction générant le tableau de routage 69

Figure 3.20 La première étape de l'algorithme de Dijkstra 70

Figure 3.21 La sous-fonction présentant la première étape de l'algorithme de Dijkstra 70

Figure 3.22 L'organigramme de deuxième étape d'algorithme de Dijkstra 71

Figure 3.23 La sous-fonction présentant la deuxième étape de l'algorithme de Dijkstra 72

Figure 3.24 La topologie du scénario traité 72

Figure 3.25 Un exemple du trajet le plus court 73

Figure 4.1 L'architecture du simulateur Matlab. Tirée de (Groleau, 2012) 78

Figure 4.2 Le débit en fonction du nombre de nœuds 82

Figure 4.3 Le délai de bout en bout en fonction du nombre de nœuds 83

Figure 4.4 Taux de paquets livrés en fonction du nombre de nœuds 84

Figure 4.5 Débit en fonction du nombre de paquets 85

Figure 4.6 Délai de bout en bout en fonction du nombre de paquets 86

Figure 4.7 Taux de paquets livrés en fonction du nombre de paquets 87

Figure 4.8 Résumé des performances 88

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AODV	Ad hoc On Demand Distance Vector
AOMDV	Ad Hoc Multipath Distance Vector
ALOHA	Areal Locations of Hazardous Atmospheres
ANSN	Advertised Neighbor Sequence Number
BSS	Basic Service Set
CBR	Constant Bit Rate
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CCA	Clear Channel Assessment
CW	Contention Window
DARPA	Defense Advanced Research Projects Agency
DS	Distribution System
DCF	Distributed Coordination Function
DIFS	Distributed Inter frame Space
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing Protocol
ESS	Extended Service Set
GloMo	Global Mobile Information Systems
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	Internet Protocol Television
IBSS	Independent Basic Service Set
IARP	Intrazone Routing Protocol

IERP	Interzone Routing Protocol
LPR	Low Cost Packet Radio
LAN	Local Area Network
LAR	Location Aided Routing Protocol
LSR	Link State Routing
MANET	Mobile Ad hoc Network
MAC	Media Access Control
MRL	Message Retransmission List
MID	Multiple Interface Declaration
MPR	Multi-Point Relays
NTDR	Near Term Digital Radio
NHDP	Mobile Ad Hoc Neighborhood Discovery Protocol
OLSR	Optimized Link State Routing Protocol
OLSRV2	Optimized Link State Routing Protocol Version 2
PRNET	Packet Radio Network
PMD	Physical Medium Dependent Sublayer
PLCP	Physical Layer Convergence Protocol
RSSI	Received Signal Strength Indicator
RDMAR	Relative Distance Micro-Discovery Ad Hoc Routing
RREQ	Route Request
RREP	Route Reply
SURAN	Survivable Radio Networks
SIG	Special Interest Group
SIFS	Short Inter Frame Space
STAR	Source Tree Adaptive Routing Protocol

TC	Topology Control
TTL	Time To Live
TCP	Transmission Control Protocol
TORA	Temporary Ordered Routing Algorithm Protocol
TLV	Type Length Value
WSN	Wireless Sensor Network
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

INTRODUCTION

L'auto-organisation et l'autonomie des nœuds caractérisant la technologie ad hoc créent la flexibilité de la technologie ad hoc. Une capacité de s'adapter aux exigences des diverses applications complexes et critiques est de plus en plus en actualité en raison des améliorations proposées dans les différents niveaux de la technologie. La valeur de cette technologie augmente dans un monde où l'utilisation des équipements de communication mobile augmente de plus en plus grâce à la disponibilité d'internet à travers des réseaux WIFI ou WIMAX. D'après les dernières statistiques, 88% des individus de plus de 16 ans utilisent l'internet en 2017, au Québec. Les domaines militaires et industriels critiques continuent aussi de déployer la technologie et visent à poursuivre sa progression. Plusieurs travaux de recherche traitent la technologie sous différents aspects. Un grand nombre de ces travaux se focalisent sur le niveau routage vu son importance primordiale. Un protocole de routage convenable au réseau ad hoc est parmi les premiers facteurs qui garantissent une livraison des paquets et un débit élevés dans le réseau. Vu l'absence de contrôle centralisé et le déploiement des mécanismes distribués par chaque nœud dans le réseau, plusieurs problèmes peuvent exister comme le comptage à l'infini, la non-convergence des données exploitées et la non-capacité du protocole d'épouser l'état instantané du réseau. Ainsi, un choix d'un protocole ayant les outils et les mécanismes nécessaires pour s'adapter aux différentes caractéristiques du réseau ad hoc est un des facteurs importants dans la réussite du déploiement de la technologie ad hoc.

C'est dans ce contexte que se situe le présent travail de recherche. Son but est le choix, la réalisation sous forme de code et l'intégration dans un simulateur fourni d'un protocole de routage adéquat au réseau ad hoc.

Le présent mémoire est organisé comme suit :

- Le chapitre 1 décrit la technologie ad hoc. Une description historique détaillant la progression de cette technologie au cours du temps est établie dans un premier temps. Dans

un second temps, nous analysons l'aspect d'auto-organisation qui est le caractère principal du réseau ad hoc. Ensuite, nous citons les différentes classifications du réseau ad hoc, en soulignant les facteurs influençant le déploiement de ce type du réseau. Les applications importantes de la technologie ainsi que les contraintes rencontrées lors de son déploiement sont soulignées. Puis, nous décrivons les mécanismes basiques de standard IEEE 802.11. Enfin, nous établissons une description détaillée du mécanisme de routage dans le réseau ad hoc ainsi que des descriptions précises pour certains protocoles de routage conçus pour le réseau ad hoc.

- Dans le chapitre 2, une comparaison qualitative des protocoles de routage les plus déployés et cités dans les travaux de recherche est établie. L'accent est mis sur les performances résultantes dans le réseau ad hoc. Cette comparaison a pour but de choisir le protocole de routage à coder. Puis, nous soulignons en détail les mécanismes du protocole choisi et nous comparons les versions standardisées.
- Le chapitre 3 englobe les différentes étapes de la réalisation sous forme du code du protocole choisi. Nous détaillons les différents organigrammes et les bases de données exploités, ainsi que les différents blocs de fonctions utilisées.
- Le chapitre 4 présente les différents résultats obtenus après l'intégration du protocole dans le simulateur fourni, après une description des stratégies de certains simulateurs réseau parmi les plus connus. Une analyse des performances résultantes est établie en soulignant l'effet de deux facteurs importants : le nombre de nœuds et le nombre de trafics.

CHAPITRE 1

RÉSEAU SANS FIL AD HOC

1.1 Le réseau ad hoc

1.1.1 Définition du réseau ad hoc

Un réseau ad hoc est un réseau sans fil composé de deux nœuds ou plus, capables de communiquer entre eux sans aucune administration centralisée contrôlée par des points d'accès. Chaque nœud dans le réseau fonctionne, à la fois, comme routeur et hôte (Loo *et al.*, 2016).
Ainsi, une absence d'infrastructure fixe laisse la place à une auto-organisation arbitraire des nœuds. En ajoutant le facteur de mobilité, on nomme le réseau MANET.

1.1.2 Historique du réseau ad hoc

Le réseau traditionnel, avec son infrastructure fixe, présente différents défis dans son déploiement. On peut citer, par exemple, les pannes imprévisibles des stations de base pouvant engendrer une interruption de service dans le réseau. Dans le milieu militaire, un tel cas rendrait le réseau vulnérable et non robuste.
En 1972, après un premier projet par *Defense Advanced Research Projects Agency* (DARPA), la première génération du réseau ad hoc est apparue et a été nommée *Packet Radio Network* (PRNET). PRNET a été associé avec un autre projet *Areal Locations of Hazardous Atmospheres* (ALOHA). Ces deux projets ont abouti à des approches d'accès au médium et de routage de type vecteur de distance, testées dans un environnement militaire .

La deuxième génération du réseau ad hoc est apparue dans les années 1980. En 1983, le projet *Survivable Radio Networks* (SURAN) a été mis en place. Il visait à fournir une connectivité ad hoc en se servant des équipements avec un coût réduit et une faible énergie consommée. Également, le nombre de nœuds dans le réseau a été augmenté afin de vérifier et d'améliorer le

facteur de mise à l'échelle. Ainsi, la technologie nommée *Low Cost Packet Radio* (LPR) a été mise en place, en 1987 (Fifer & Bruno, 1987).

Dans les années 1990, la croissance du nombre d'ordinateurs et la disponibilité des équipements sans fil ont poussé les communautés scientifiques et industrielles à créer le concept de la commercialisation de la technologie ad hoc. L'accès du public à cette technologie est ainsi devenu une réalité. Les grandes sociétés informatiques et de télécommunication (Ericsson, IBM, Intel,...) ont formé le groupe *Special Interest Group* (SIG), ayant pour but de fournir des solutions pour le déploiement de la connectivité ad hoc entre des équipements avec des caractéristiques hétérogènes. S'ajoute à cela la poursuite de l'effort de DARPA qui a mis en place : *Global Mobile Information Systems* (GloMo) et *Near Term Digital Radio* (NTDR). Ces deux projets visaient à offrir un milieu de connexions multimédia entre des équipements portables (Leiner *et al.*, 1996) ; (Redi, 2002).

Le comité d'IEEE 802.11 a adopté le terme ad hoc et les chercheurs ont commencé à voir la possibilité de déploiement de réseaux ad hoc avec des nouvelles applications. En plus en 1998, le groupe de travail *Mobile ad hoc Network* (MANET) a été créé au sein d'*Internet Engineering Task Force* (IETF) afin de fournir des standards valides de protocoles de routage basés sur la technologie IP dans le réseau ad hoc.

1.1.3 Le concept d'auto-organisation

La grande évolution dans les réseaux sans fil, des réseaux traditionnels vers les réseaux ad hoc, est basée principalement sur le mécanisme distribué. Le caractère arbitraire oblige chaque nœud dans le réseau ad hoc d'avoir recours à un mécanisme fiable d'auto-organisation.
Qu'on le considère comme un but ou comme un moyen, le concept d'auto-organisation est primordial afin d'aboutir à une persistance dans le réseau ad hoc. Ainsi, une compréhension profonde de concept est nécessaire.
Dressler (2008) présente, de manière abstraite, l'auto-organisation comme une émergence de la configuration globale du système à partir des seules collaborations et interactions de divers

dispositifs élémentaires du système.

Cette définition implique plusieurs autres concepts. En fait, on ne peut pas aboutir à une auto-organisation fiable sans avoir recours à l'auto configuration, qui est présentée par les méthodes de reproduction des configurations adéquates aux circonstances environnementales. Dans le cas du réseau ad hoc, ces circonstances peuvent être l'état et la qualité de la connexion. En cohérence avec l'auto-configuration, le concept de l'auto-gestion impose aux différents dispositifs d'être toujours en lien avec les paramètres propres du système. Ensuite, l'auto-optimisation prend place afin de définir les choix optimaux des méthodes en se basant sur le comportement du système. Le dernier concept est l'adaptation aux conditions du milieu. Dans le réseau ad hoc, on peut citer comme exemple le nombre des nœuds voisins pour chaque noeud.

En conclusion, l'auto-organisation est centrée sur les interactions locales des éléments sans aucune référence à un modèle global. C'est pourquoi chaque élément développe certaines propriétés comme l'autonomie dans la prise de décision et l'adaptation à un environnement dynamique. En ce qui concerne, le niveau routage dans le réseau ad hoc, les protocoles conçus respectent les exigences de l'auto-organisation. Ces protocoles de routage exploitent des mécanismes de contrôle et des algorithmes de routage visant à garantir une convergence et consistance des données et à choisir les trajets optimaux (Biskupski *et al.*, 2007). On peut citer l'algorithme de Dijkstra (Dijkstra, 1986).

1.1.4 Les classifications du réseau ad hoc

D'un réseau ad hoc à un autre, plusieurs facteurs diffèrent, dépendant du but et des caractéristiques voulues du déploiement. On peut avoir une diversité dans les procédures de communication (un saut ou plusieurs sauts), une variété des architectures, des configurations homogènes ou hétérogènes des nœuds et des zones de couvertures distinctes. Ainsi, une classification du réseau ad hoc, en se basant sur ces facteurs, est possible (Loo *et al.*, 2016).

La première classification est basée sur la procédure de communication. En effet, le réseau où tous les nœuds sont dans leurs portées mutuelles est appelé réseau ad hoc à un saut. Chaque nœud peut communiquer directement avec n'importe quel autre nœud dans le réseau sans au-

cun intermédiaire. Ce type de procédure de communication nécessite généralement une énergie de transmission élevée. Dans le cas où le trafic doit passer par un nœud intermédiaire, afin d'arriver à la destination désignée, le réseau est appelé réseau ad hoc multi sauts. C'est le cas le plus répandu. Ajoutant le critère de mobilité dans le réseau, la sélection du trajet pour l'envoi de trafic de données serait de plus en plus complexe, d'où la nécessité de déploiement des protocoles de routages capables de s'adapter, essentiellement, aux changements rapides de la topologie.

La Figure 1.1 illustre la différence entre les deux procédures de communication. La communication multi-sauts réutilise le domaine spatiale et temporel, tandis que la communication à un saut se concentre uniquement sur la disponibilité temporelle de médium (Frodigh *et al.*, 2000).

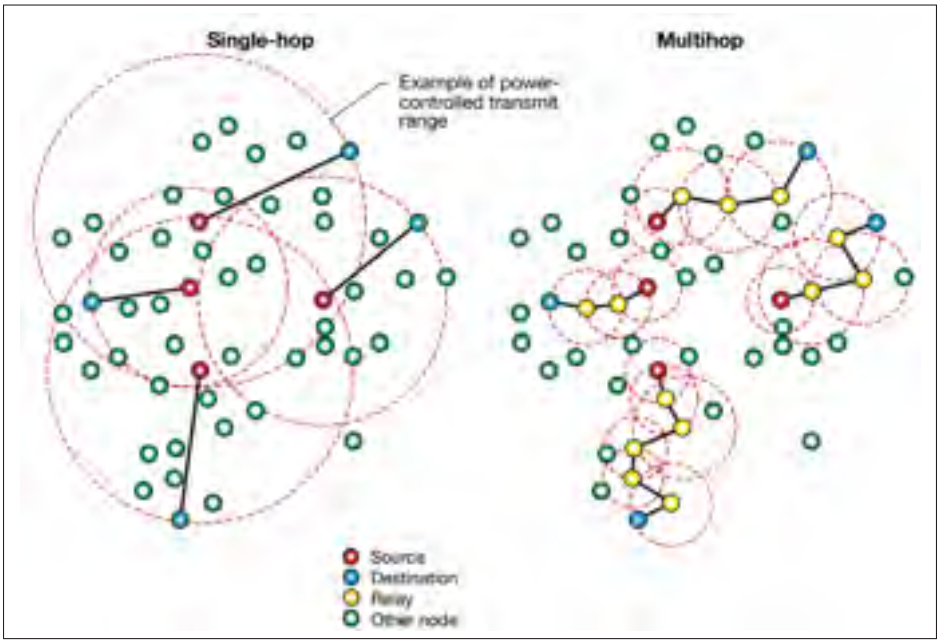

Figure 1.1 La communication multi-sauts et la communication à un saut.
Tirée de (Frodigh *et al.*, 2000)

Aussi, les réseaux ad hoc peuvent avoir plusieurs architectures en se basant sur la topologie. En fait, cette classification se base sur l'existence ou non d'une hiérarchie entre les nœuds. On aboutit, en premier lieu, à une vue à plat du réseau où tous les nœuds fonctionnent de manière identique. La conception de ce type de réseau est relativement simple, mais une croissance

intensive du nombre de nœuds peut affecter le facteur de mise à l'échelle. Une deuxième vue du réseau ad hoc choisit de créer une hiérarchie (Walikar & Biradar, 2017). Tout d'abord, la procédure commence par la division du réseau en clusters liés entre eux. Chaque cluster confie à un noeud le rôle du chef (*clusterhead*) responsable de contrôle et de gestion des connexions entre les différents nœuds. Le routage serait simple puisqu'il est établit de passerelle à passerelle jusqu'à celle directement liée à la destination. L'apport de l'approche hiérarchique est essentiellement la facilité de maintien des informations topologiques vu que la grande part de gestion de connexions est supportée par le nœud chef. Ensuite, le déploiement des passerelles améliore le facteur de mise à l'échelle . La Figure 1.2 éclaircit cette vue hiérarchique du réseau ad hoc.

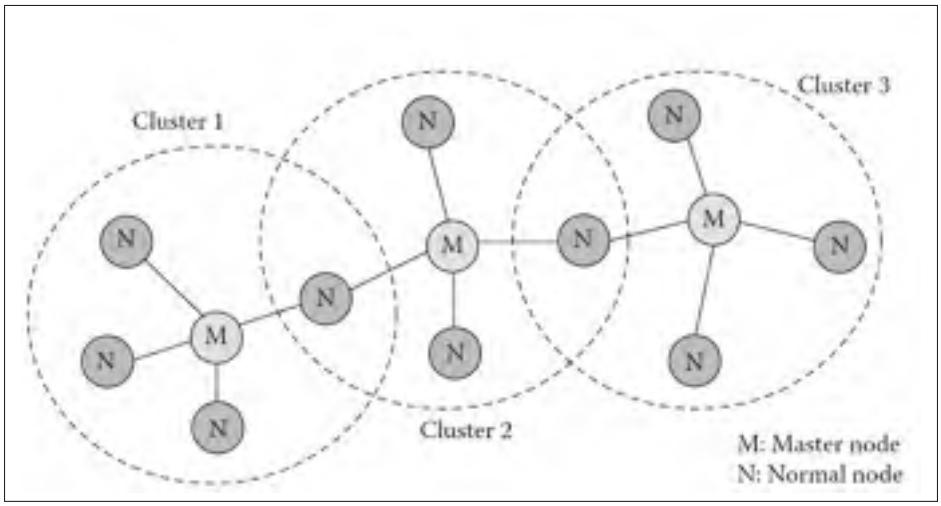

Figure 1.2 Une vue à hiérarchie du réseau ad hoc.
Tirée de (Loo *et al.*, 2016)

Enfin, une architecture agrégée du réseau ad hoc présente le troisième type de la classification basée sur la topologie (Walikar & Biradar, 2017). En fait, le réseau est décomposé en zones. Chaque nœud dans chaque zone possède deux niveaux d'identification. Le premier niveau est son identification propre. Le deuxième niveau présente l'identification de la zone. L'apport de cette approche est la facilité de la maintenance de liens aisément localisés grâce à l'indicateur de zone.

La troisième classification se base sur la configuration matérielle de chaque nœud dans le réseau (Loo *et al.*, 2016). On a deux types de configurations. La configuration homogène impose à tous les nœuds d'avoir les mêmes caractéristiques matérielles (le processeur, la mémoire...). La deuxième configuration est la configuration hétérogène. La différence entre les caractéristiques matérielles de chaque nœud entraine un fonctionnement différent. Ainsi, chaque nœud a ses ressources et ses politiques. Cette hétérogénéité peut concerner des groupes de nœuds au lieu d'un seul nœud. Ces groupes appartiennent à des réseaux déployant des technologies différentes (Bluetooth, WIFI, etc). La Figure 1.3 illustre clairement cette hétérogénéité de configuration.

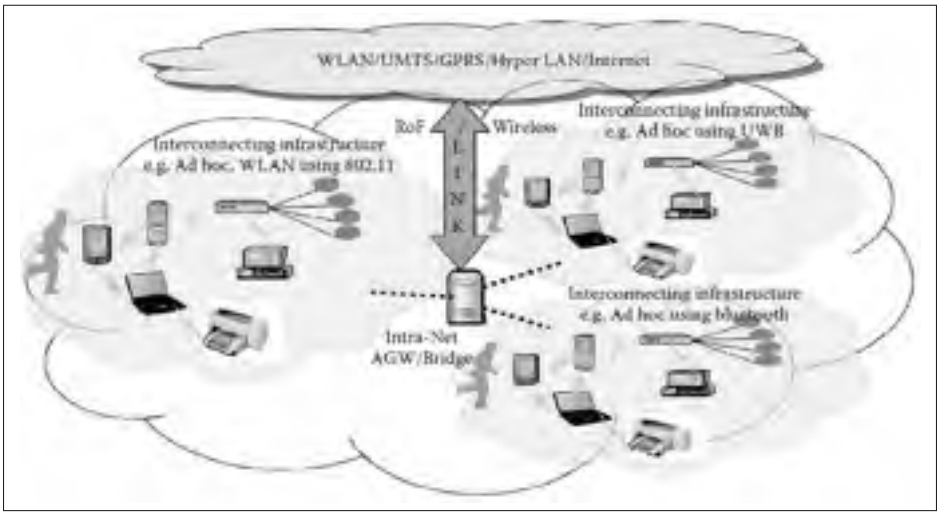

Figure 1.3 La configuration hétérogène du réseau ad hoc.
Tirée de (Loo *et al.*, 2016)

Cette diversité de classifications rend la technologie ad hoc plus flexible et applicable dans différentes applications. La section 1.1.5 souligne les applications les plus importantes.

1.1.5 Les applications du réseau ad hoc

Comme le domaine militaire a déclenché l'étincelle de recherche pour le réseau ad hoc, le déploiement de la technologie ad hoc persiste grâce à son autonomie et sa robustesse. Afin de maintenir la communication dans un milieu de conflit, lors des déplacements rapides des

objets militaires, la connexion doit être fiable, rapide et sécurisée (Kumar Sarkar *et al.*, 2013). La Figure 1.4 illustre la coordination entre les différents soldats dans un scénario de combat à l'aide de la technologie ad hoc.

Figure 1.4 L'application militaire de la technologie ad hoc.
Tirée de (Bisnik, 2007)

La technologie ad hoc prend place dans les situations d'urgence, vu sa capacité de s'adapter à la mobilité aléatoire des nœuds et de s'auto organiser (Sanchez-Garcia *et al.*, 2016). Dans le cas des désastres naturels qui peuvent causer la destruction d'infrastructure existante, la technologie ad hoc aide à organiser les activités de sauvetage à travers l'acheminement des données par l'intermédiaire de connexions multi sauts. Aussi, vu que les communications vocales dominent dans ces cas, l'ad hoc doit supporter les applications temps réel.

Autre application de la technologie est *wireless sensor network* (WSN) (Rashid & Husain, 2016) qui est répandue dans les environnements ruraux et urbains. Plusieurs domaines ont profité de cette technologie, essentiellement où l'intervention humaine directe est impossible ou dangereuse. On peut citer le domaine industriel avec des conditions critiques comme la haute température, les réactions nucléaires et la haute pression. Afin de prévenir des désastres naturels, un déploiement de WSN peut fournir un contrôle en temps réel afin de prévenir des

catastrophes. La Figure 1.5 montre l'intégration de WSN dans un système pipeline souterrain responsable de la distribution du gaz naturel.

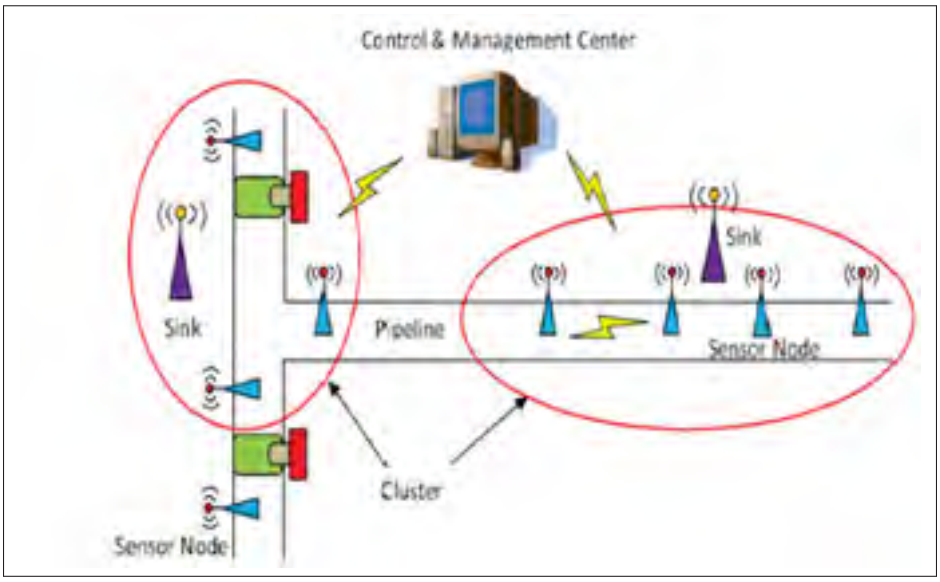

Figure 1.5 Un exemple de déploiement de WSN.
Tirée de (Rashid & Husain, 2016)

1.1.6 Les contraintes du réseau ad hoc

Malgré ses avantages, le déploiement de la technologie ad hoc présente certains problèmes liés essentiellement au caractère imprévisible des nœuds, au médium et au routage. Parmi ces problèmes, on peut citer :

- L'interférence : Si des transmissions se font sur une même fréquence, des interférences peuvent prendre place, ce qui mène à une perturbation des connexions et à une réduction dans la qualité de liens.

- Le terminal caché : Dans le réseau sans fil, la détection de collision ne peut pas être exploitée à cause de la nature du médium. Le problème de terminal caché peut avoir lieu. Par exemple, on prend le cas de deux nœuds S1 et S2 qui sont hors de portée l'un de l'autre et un nœud R1 joignable par les deux. S2 ne peut pas détecter une transmission de S1 vers R1.

Une collision peut avoir lieu si S2 émet vers R1, simultanément avec S1.

- Le terminal exposé : Ce problème est référé par l'impossibilité d'un premier nœud de transmettre, à cause de son estimation de l'occupation du canal par les transmissions d'un deuxième nœud placé dans la portée du premier.
  La Figure 1.6 illustre le problème du terminal exposé et du terminal caché.

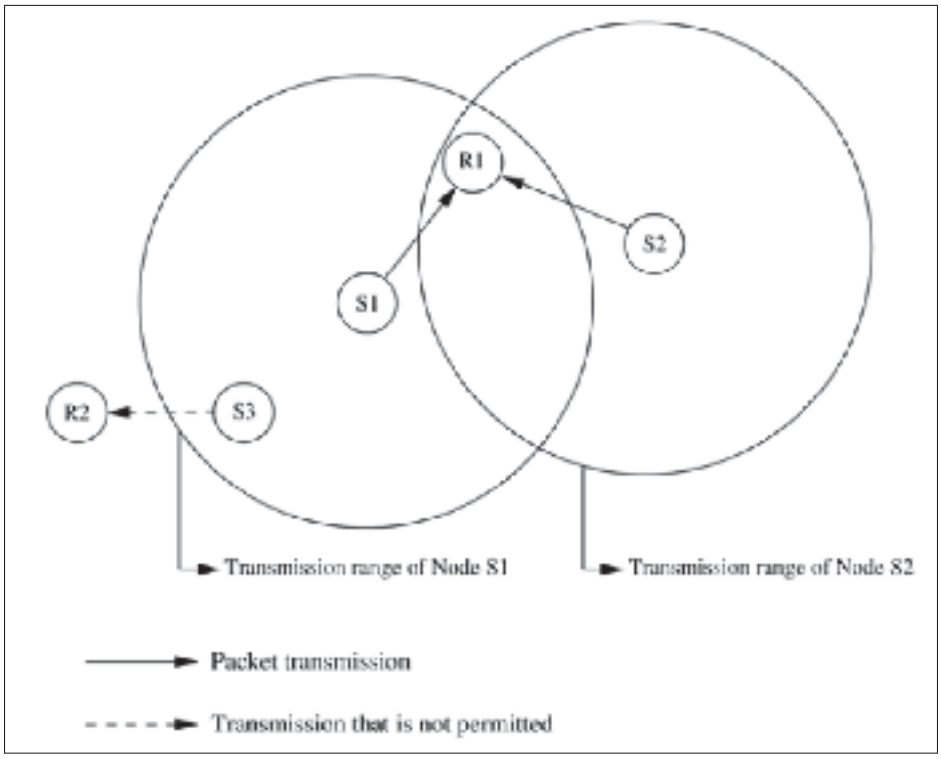

Figure 1.6 Le problème du terminal caché et du terminal exposé.
Tirée de (Murthy & Manoj, 2004)

- Débit : Il est influencé par plusieurs facteurs dans le réseau comme l'occurrence des collisions et la disponibilité du canal. Ce défi s'aggrave avec le déploiement de certains services comme la communication vidéo et l'IPTV.

- Énergie : Les nœuds dans le réseau ad hoc sont caractérisés par des ressources d'énergie limitées. La consommation d'énergie accroit avec la mobilité des nœuds et l'acheminement multi sauts des paquets. Un contrôle et gestion de la consommation énergétique est important afin de garantir une continuité des services.

- La bande passante : Dans les réseaux sans fil, la bande passante est partagée par tous les nœuds. Ainsi sa disponibilité est affectée par le nombre des nœuds et les trafics à envoyer.

- Routage : La mobilité dans les réseaux sans fil ad hoc cause des variations aléatoires dans le voisinage immédiat de chaque nœud et dans la topologie du réseau entier. Ainsi, un défi est dans la conception des protocoles de routage capables de s'adapter rapidement à cette dynamique et à reconfigurer des trajets optimaux avec les minimums délais d'accès.

1.2 Le standard IEEE 802.11

Le standard IEEE 802.11, apparu en 1997, est le standard international caractérisant la couche MAC et la couche physique dans les réseaux sans fil LAN. Le rôle principal de ce standard est de spécifier les mécanismes de gestion de connexion, de contrôle de fiabilité de liens et de contrôle de consommation d'énergie.

Débutant par l'architecture, l'IEEE 802.11 définit deux architectures pour le réseau sans fil LAN : avec et sans point d'accès. L'architecture avec point d'accès est configurée sous forme d'un ensemble appelé service de base *Basic Service Set* (BSS). Les connexions entre les terminaux d'un BSS sont gérables par le point d'accès, équipé généralement d'une carte réseau filaire permettant, par exemple, d'avoir une connexion internet. La deuxième architecture fonctionne en mode ad hoc. L'ensemble est appelé ensemble de services de base indépendante *Independent Basic Service Set* (IBSS). Il est possible d'assembler un BSS et un IBSS par l'intermédiaire de *Distribution System* (DS) afin de former un ensemble de services étendus *Extended Service Set* (ESS).

Afin d'aboutir à un bon déploiement du réseau sans fil sous ses différentes architectures, l'IEEE 802.11 organise la couche physique sous forme de deux sous-couches, *Physical Medium Dependent Sublayer* (PMD) et *Physical Layer Convergence Protocol* (PLCP). PMD se charge du codage, d'encodage et de modulation de signal. PLCP offre la fonctionnalité de l'écoute de porteuse à la couche MAC. Le mécanisme *Clear Channel Assessment* (CCA), intégré au niveau de PLCP, signale l'occupation du canal après une détection de bits dans l'air ou la vérification de *Received Signal Strength indicator* (RSSI). Le CCA aide le MAC à implémenter le protocole *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) qui contrôle l'accès au canal avec évitement de collision. Dans le mode ad hoc, CSMA/CA est basé sur le mécanisme *Distributed Coordination Function* (DCF). Ainsi, un accès aléatoire est établi à l'aide du déploiement d'intervalles de temps nommés espace inter-trames et aussi un algorithme de *backoff*. L'intervalle inter-trames définit l'intervalle entre la transmission de deux trames successives d'un terminal. Pour le réseau ad hoc, l'accès au médium est organisé essentiellement au moyen de deux intervalles, *Short Inter Frame Space* (SIFS) et *Distributed Inter frame Space* (DIFS).

SIFS : est l'écart le plus court. Il donne une priorité d'accès au canal afin que les transmissions des fragments et des acquittements entre l'émetteur et le récepteur soient terminées (IEEE 802.11, 2012, section 9.3.4.5). La valeur de SIFS est fixée par la couche physique en se basant principalement sur le délai de traitement au niveau de MAC et le délai de commutation en mode réception et émission. La Figure 1.7 illustre la transmission des fragments et des acquittements d'un même dialogue.

DIFS : est l'intervalle pendant lequel le terminal doit attendre la disponibilité du canal. Si le canal n'est pas libre, un algorithme de *backoff* se met en place, consistant en une fixation aléatoire d'un *backoff* entre *contention window min* (CWmin) et *contention window max* (CWmax). Après un DIFS et dans le cas où le canal est libre, le *backoff* décrémente. La transmission débute quand le backoff atteint la valeur zéro. Si le canal n'est pas libre, la procédure est suspendue (IEEE 802.11, 2012, section 9.3.4.3). La Figure 1.8 décrit une transmission utilisant le DIFS.

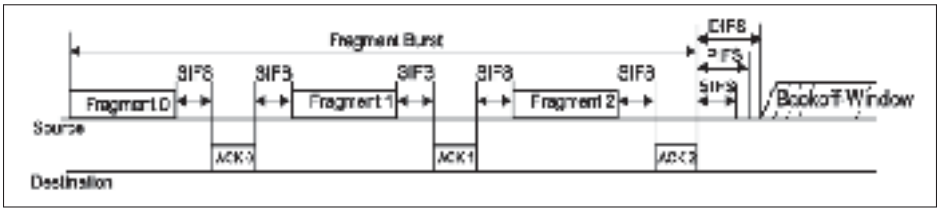

Figure 1.7 Transmission des fragments avec SIFS.
Tirée de (IEEE 802.11, 2012)

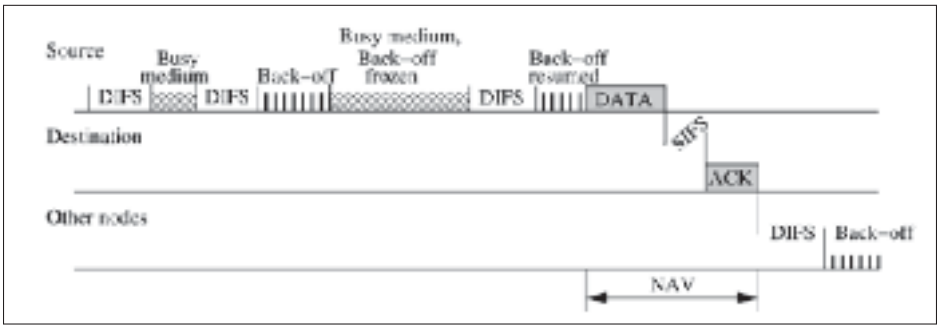

Figure 1.8 Transmission avec DIFS.
Tirée de (Murthy & Manoj, 2004)

1.3 Les protocoles de routage

Les caractéristiques du réseau ad hoc obligent les protocoles de routage à s'orienter vers certains critères. En fait, un protocole de routage adéquat pour le réseau ad hoc doit adopter le mécanisme distribué et épouser le dynamisme de la topologie. Cela garantit une convergence et une validité des données et ainsi permet de construire des trajets optimaux sélectionnés selon des critères bien choisis. Par exemple, le choix du trajet le plus court en termes de nombre de sauts permet un acheminement rapide des trafics de données. Son déploiement aide ainsi à respecter les limites de ressources disponibles (mémoire, batterie, bande passante) vu qu'un nombre minimal de nœuds est inclus dans le routage. Le critère de nombre de sauts peut être associé avec d'autres comme la qualité de liaison et le critère d'interférence.

Plusieurs travaux ont essayé de concilier les exigences de routage du réseau ad hoc en adoptant différentes approches. Les protocoles proposés peuvent être classifiés selon divers critères (Murthy & Manoj, 2004). En premier lieu, on peut citer les protocoles de routage basés sur

la mise à jour de l'information. Une collecte de données topologiques et de voisinage se met en place. Différents mécanismes de mise à jour de l'information peuvent être exploités. Un mécanisme périodique est présenté par une diffusion périodique de messages de contrôle. Un mécanisme sous demande consiste à une diffusion de messages par un nœud source en cas de besoin d'envoi des données à une destination (Dhenakaran & Parvathavarthini, 2013). En deuxième lieu, on peut mentionner le critère temporel. En fait, la direction de certains protocoles est d'exploiter des données déjà collectées afin de calculer les trajets les plus courts. Dans un autre cas, une durée de vie estimée de lien ou une prédiction de localisation peuvent être demandées. Un traitement des données estimées dans le futur prend place alors dans la détermination de la stratégie du routage (Baccour *et al.*, 2015). En troisième lieu, une autre direction de recherche sur certains protocoles consiste à accorder la priorité aux ressources limitées d'énergie. Ce qui mène à prendre des décisions de routage visant à réduire la consommation d'énergie. Cette approche est exploitée principalement dans les réseaux de capteurs (Vazifehdan *et al.*, 2014).

Ainsi, plusieurs stratégies sont adoptées afin d'aboutir à un routage efficace des données dans le réseau ad hoc. Mais, les mécanismes de la mise à jour de l'information restent le critère basique qui détermine la bonne convergence et la validité de données nécessaires pour prendre les décisions adéquates de routage. Les protocoles utilisant ce mécanisme sont décrits plus précisément dans les sections suivantes.

1.3.1 Les protocoles proactifs

Avec les protocoles proactifs, chaque nœud dans le réseau vise à fixer les routes optimales en termes de nombre de sauts vers tous les autres nœuds dans un tableau de routage disponible à tout moment. Une mise à jour périodique des informations de topologie est maintenue afin de garantir un calcul correct et précis des trajets. Des techniques de sélection de trajets peuvent se différencier d'un protocole à un autre. Certains protocoles choisissent comme métrique, seulement, le trajet le plus court comme le protocole *Destination Sequenced Distance Vector* (DSDV) et le protocole *Wireless Routing Protocol* (WRP). Ces protocoles sont nommés pro-

tocoles à vecteur de distance. Autres se dirigent vers la sélection des trajets en se basant sur l'état de liens comme *Optimized Link State Routing Protocol* (OLSR), ils sont désignés comme protocoles à état de lien.

WRP est présenté en 1996 par Murthy et Garcia (*an efficient routing protocol*). La sélection du trajet est basée sur l'algorithme de *Bellman-Ford* permettant d'avoir le trajet le plus court. Pour un maintien précis et consistant des informations, chaque nœud organise les données en quatre tableaux : tableau de routage, tableau de distance, tableau de coût de liens (nombre de sauts) et tableau MRL *(Message Retransmission List)*. Un échange de messages de contrôle est établi au niveau du voisinage de chaque nœud. Le message contient essentiellement les identifications des destinations, la distance à chaque destination et les identifications de l'avant-dernier *(penultimate node)* et le dernier *(successor)* nœuds par rapport à la destination . Lors de retransmission d'un message mis à jour, une entrée dans le MRL est ajoutée. Elle contient le numéro de séquence du message, le compteur de retransmission, un vecteur d'acquittement et la liste de mises à jour envoyée. Après chaque retransmission, le compteur de retransmission décrémente. Si le compteur atteint zéro et on n'a pas d'acquittement reçu, on conclut une défaillance dans le lien et le nœud concerné envoie un message contenant un coût de lien défaillant égal à l'infini à son voisinage. Après la réception et le traitement de messages, le tableau de distance ajoute et met à jour les distances minimales aux destinations qui sont accessibles par l'intermédiaire de voisins. Ainsi, le tableau de routage va contenir pour chaque destination son identification, la distance à parcourir, l'avant-dernier et le dernier nœud le long du trajet sélectionné par l'algorithme de routage et il est disponible à tout moment. Afin d'éviter le problème de comptage à l'infini *(counting to infinity)*, WRP oblige tous les nœuds de vérifier la validité des informations de l'avant dernier et dernier nœuds rapportées par tous les voisins afin de garantir une convergence rapide.

OLSR est un protocole à états de lien connu, développé à l'Institut National de Recherche en Informatique et en Automatique (INRIA, France) et standardisé par *Internet Engineering Task Force* (IETF) dans la RFC 3626 (Clausen & Jacquet, 2003). Une deuxième version a été publiée et standardisée dans la RFC 7181 (Clausen et al., 2014). OLSR est une optimisation

de protocole *Link State Routing* (LSR) (Mcquillanir *et al.*, 1980). Une diffusion périodique des messages de contrôle appelés Hello permettant un échange des états de lien entre les nœuds d'un même voisinage. Ainsi, chaque Hello contient les identifications de nœuds voisins et leurs états de lien. Vu des changements non prévus de la technologie radio et la mobilité aléatoire des nœuds, la bidirectionnalité de liens n'est pas toujours valide. OLSR permet une sauvegarde temporelle des liens unidirectionnels, en attendant l'établissement de la bidirectionnalité. Un deuxième message de contrôle, appelé *Topology Control* (TC), permet de diffuser périodiquement les données de voisinage de chaque nœud dans le réseau entier. C'est possible pour chaque nœud d'avoir une vision sur la topologie du réseau entier.

Afin de minimiser la charge de trafic de contrôle, le protocole permet une diffusion optimisée à l'aide des relais multipoints (MPR). Chaque nœud choisit des nœuds de son voisinage direct (1 saut), lui permettant d'accéder aux différents autres nœuds placés dans le voisinage de deux sauts avec le minimum de transmissions. La Figure 1.9 illustre une diffusion classique vs une diffusion avec MPR. On constate clairement la diminution de nombre de retransmissions requis pour atteindre les nœuds éloignés de trois sauts par rapport au nœud source. Ce mécanisme de diffusion peut être exploité dans un réseau dense et vaste, ce qui mène à une amélioration dans les performances du réseau. Après la collecte des données de voisinage direct et de topologie du réseau, chaque nœud sera capable de construire son tableau de routage, disponible immédiatement et contenant les routes optimales vers toutes les destinations disponibles. Généralement, l'algorithme de *Dijkstra* (Dijkstra, 1959) est exploité afin de déterminer les trajets les plus courts en termes de nombre des sauts. Ce protocole est vu plus en détail au chapitre 2.

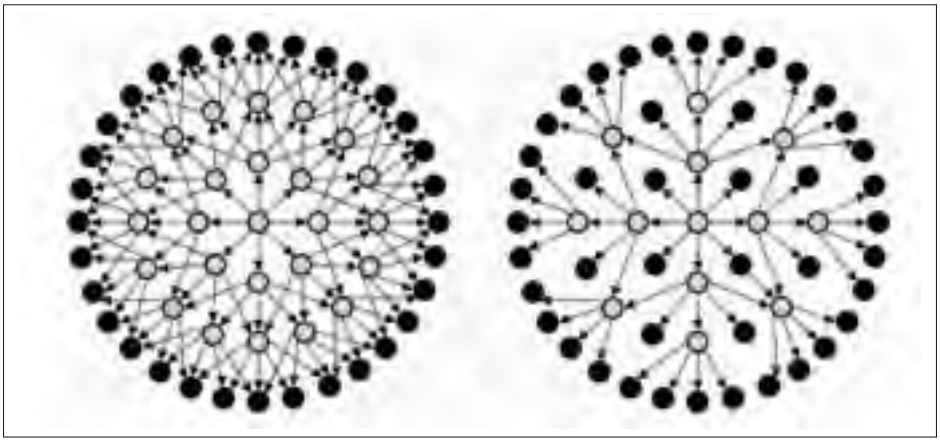

Figure 1.9 Diffusion à l'aide de MPR.
Tirée de (Voorhaen & Blondia, 2006)

1.3.2 Les protocoles réactifs

Le deuxième mécanisme de recherche des routes est le mécanisme réactif. En effet, la recherche de route est établie sous demande. Les nœuds ne nécessitent pas de sauvegarde des trajets dans des tableaux de routage contenant toutes les destinations dans le réseau. Le but de cette approche est d'éliminer la charge de trafic de contrôle dans le réseau, causé par l'envoi périodique des messages de contrôle. On peut citer deux protocoles adoptant cette approche : *Dynamic Source Routing Protocol* (DSR) et *Ad Hoc On Demand Distance Vector* (AODV).

DSR est présenté par (Johnson *et al.*, 2001). Il fait partie du projet *Monarch* à l'université *Carnegie Mellan* (Johnson & Maltz, 1996). Le mécanisme de DSR est divisé en deux étapes ; la découverte de routes et la maintenance de routes.
La première étape est déclenchée lors d'une demande d'un nœud source d'envoyer un paquet à un nœud destination. Un message de demande de route (*Route Request*) (RREQ) sera diffusé dans le réseau identifiant la source, la destination et un paramètre indicatif de demande. Chaque nœud différent de destination rediffuse le message de demande, en ajoutant son identification dans la route sauvegardée dans le paquet. La Figure 1.10 montre un exemple de demande de route entre nœud A (la source) et nœud E (la destination).

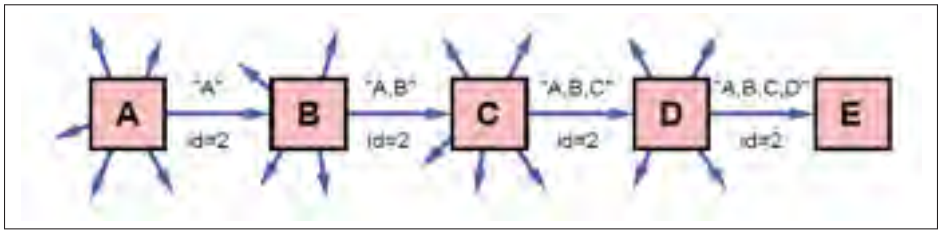

Figure 1.10 La procédure de demande de route pour DSR.
Tirée de (Johnson *et al.*, 2001)

Chaque destination recevant avant une demande de route (RREQ) sauvegarde la route suivie par (RREQ) dans un cache de route. Un message de réponse (*Reply Message*)(RREP) sera envoyé par la destination ayant déjà trouvé une route sauvegardée vers la source dans son cache de route. La route suivie par le message de demande de route (RREQ) est incluse dans (RREP). Dans le cas de non-disponibilité d'enregistrement, un message de demande de route (destination, source) est associé avec (RREP). Dans le cas de la réception de plusieurs messages de réponse, la source sélectionne le trajet optimal en termes de nombre de sauts.
La deuxième étape qui est la maintenance des routes consiste à confirmer que le paquet est reçu par le prochain saut. Dans le cas de non-réception d'acquittement, on peut aboutir à un nombre limité de retransmissions. En les dépassant, un message d'erreur sera envoyé au nœud source identifiant le lien défaillant et le nœud marque la route comme non valide dans son cache de route.

AODV (*Ad Hoc On Demand Distance Vector)* détaillé dans la RFC 3561 par (Perkins, Belding-Royer & Das, 2003) est basé sur les vecteurs de distance. Dans le cas de non-disponibilité de route pour une destination, la source diffuse un message de demande de route *RouteRequest* (RREQ) dans le réseau. Ce message peut contenir l'adresse de la source (*SrcID*), l'identification de la destination (DestID), un numéro de séquence pour la source (SrcSeqNum), un numéro de séquence pour la destination (DestSeqNum) et la durée de vie de paquet (TTL). Quand un nœud intermédiaire reçoit le message de demande de route, il rediffuse le message dans le cas de non-disponibilité de route mémorisée et valide. Cette validité est vérifiée par une comparaison entre un numéro de séquence mémorisé et le (DestSeqNum) inclus dans le

message. Lorsque le message de demande parvient à la destination ou à un nœud intermédiaire avec une route disponible, une réponse RREP sera envoyée vers la source selon le chemin inverse. Chaque nœud intermédiaire sauvegarde le dernier nœud envoyant le RREP dans son tableau de routage local afin de sauvegarder une mise à jour et une route valide vers la destination. Le nœud source vérifie la récence de route en vérifiant le (DestSeqNum) et le trajet idéal est le plus court en termes de nombre de sauts. Ainsi, le numéro de séquence (DestSeqNum) est la clé d'AODV pour éviter les problèmes de boucles et comptage à l'infini connus dans les protocoles basés sur les vecteurs de distance. Dans le cas de déplacement de la destination ou un nœud intermédiaire en dehors de la portée, un message RRER (*Route Error*) sera transmis aux nœuds sélectionnés dans le trajet. Quand le nœud source reçoit le RRER, il peut initialiser un RREQ, si c'est nécessaire.

La Figure 1.11 illustre la découverte de route par AODV.

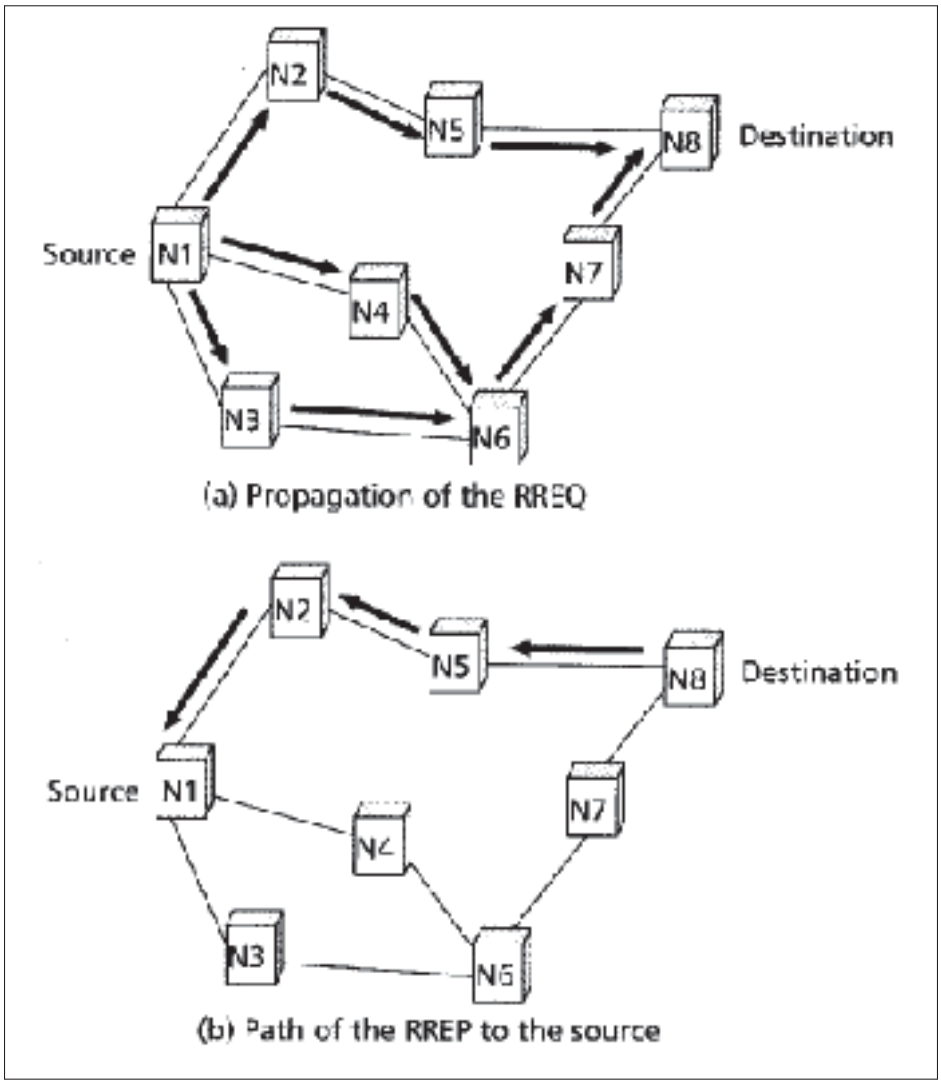

Figure 1.11 La procédure de recherche de route par AODV.
Tirée de (Royer & Toh, 1999)

1.3.3 Les protocoles hybrides

Les protocoles de routage hybrides essayent de combiner l'approche proactive et l'approche réactive dans l'établissement des routes afin d'en tirer les avantages de déploiement de deux mécanismes. Parmi les premiers protocoles adoptant cette direction, on peut citer *Zone Routing Protocol* (ZRP) et *Relative Distance Micro-Discovery Ad Hoc Routing* (RDMAR).

ZRP (Haas, 1997) divise le réseau en deux zones déployant deux différents mécanismes. En fait, la première zone est nommée *Intrazone Routing Protocol* (IARP) où les nœuds exploitent le mécanisme proactif. Ainsi, chaque nœud essaye de détecter son voisinage et construit sa zone de routage (*Routing Zone*) limitée par un nombre défini X de sauts. Pour chaque zone de routage, on caractérise deux types de nœuds : intérieurs et périphériques. Les périphériques se trouvent aux frontières de la zone. Chaque nœud maintient les informations de tous les nœuds dans sa zone de routage par l'envoi périodique de paquets de contrôle. Une disponibilité immédiate de toutes les routes est ainsi possible.

La deuxième zone est nommée *Interzone Routing Protocol* (IERP) et s'étend au reste du réseau. Le mécanisme réactif est exploité, d'où l'envoi des messages de demande de route (RREQ) lors de la tentative de transmissions des paquets de données.

La Figure 1.12 clarifie la répartition du réseau en deux zones IERP et IARP, de point de vue du nœud S. Les nœuds H,G,J et I sont les nœuds périphériques de la zone IERP.

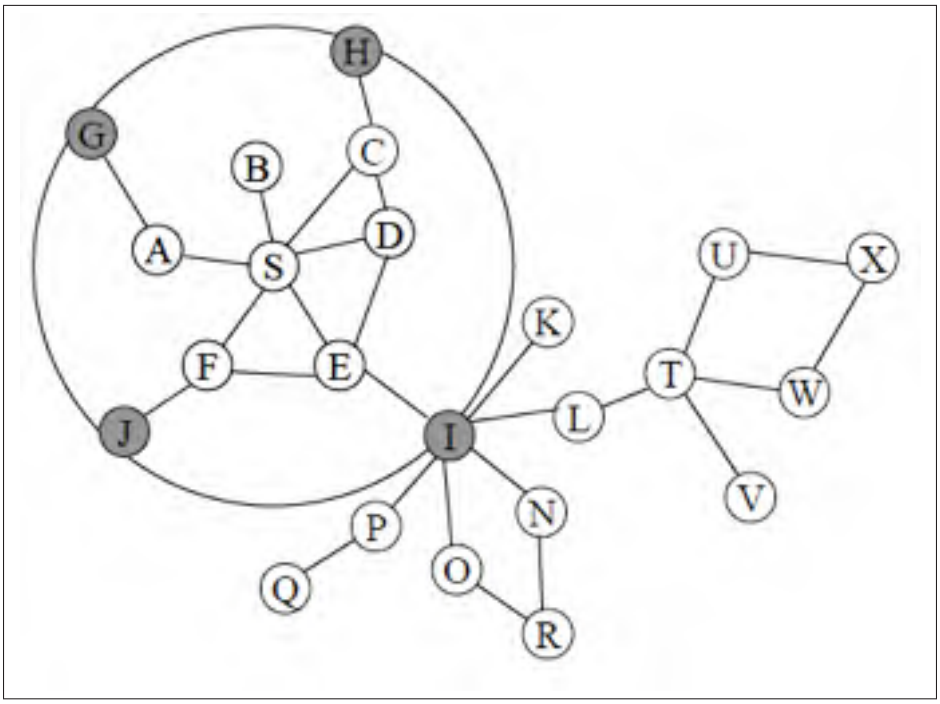

Figure 1.12 IERP et IARP selon le protocole ZRP
Tirée de (Beijar, 2002)

L'apport ajouté de ZRP est la mise en place de concept (*Bordercast*) qui s'illustre clairement dans l'envoi de RREQ. En effet, si un nœud cherche la route vers une destination hors la zone IERP, il envoie des demandes de routes (RREQ) vers les nœuds périphériques, au lieu de diffuser (RREQ) dans le réseau. Si un nœud périphérique connaît la route vers la destination, il transmet un message de réponse (RREP) à la source, sinon le (*Bordercast*) continue jusqu'à la localisation de la destination.

RDMAR (Aggelou & Tafazolli, 1999) présente une similarité avec les protocoles réactifs dans la découverte et la maintenance des routes. En fait, une diffusion des messages de demande de routes est accomplie, mais en respectant un nombre limité des sauts. Ce nombre est déterminé en fonction de la distance relative entre source et destination, calculée en fonction de la dernière distance enregistrée. Aussi deux autres facteurs sont considérés, soit le temps de la dernière mise à jour de route et la vitesse estimée de la destination. L'adaptation de l'approche proactive s'illustre dans le maintien d'un tableau de routage contenant le nœud suivant le long du trajet vers la destination.

La maintenance de la route prend place dans le cas d'erreur d'envoi. Si un nœud a un problème dans l'envoi de paquet des données à cause d'un lien ou nœud défaillant, un nombre de retransmissions du même paquet prend place. Si le problème continue à exister, la phase de découverte de route est initialisée en augmentant le nombre de sauts à respecter.

CHAPITRE 2

LE CHOIX DE PROTOCOLE DE ROUTAGE

2.1 Étude comparative des protocoles réactifs, proactifs et hybrides

Une diversité dans les approches de routage présentées nous mène à poser la question du choix optimal d'un protocole de routage. Plusieurs articles traitent cette problématique sous différents scénarios, multiples environnements et divers paramètres de simulation. Plusieurs facteurs peuvent changer en fonction du but d'exploitation de la technologie ad hoc, comme la mobilité, le nombre de nœuds et la quantité de trafic à acheminer. Ainsi, une comparaison qualitative peut aider à filtrer les comportements des protocoles. Parmi les protocoles de routage multi-sauts, les plus populaires dans les travaux de recherche et les applications industrielles sont essentiellement les protocoles proactifs OLSR et DSDV (*Destination Sequenced Distance Vector*) et les protocoles réactifs AODV, DSR. La comparaison est effectuée selon les critères suivants : le taux de paquets livrés, le délai de bout en bout, le volume de trafic de contrôle, la longueur de route, le débit et le taux de perte de paquets.

2.1.1 Taux de paquets livrés

Le rôle primordial de chaque protocole de routage est la livraison de paquets de données envoyés, qui peut être exprimée à l'aide de la métrique taux de paquets livrés. Les facteurs de mobilité et de quantité de trafic à acheminer influencent la métrique. Dans (Clausen et al., 2010), le résultat montre qu'OLSR a une bonne stabilité en ce qui concerne le taux de paquets livrés en fonction du nombre de flot de trafic (*traffic streams*), tandis que AODV et DSR connaissent un abaissement. En fait, avec l'augmentation du trafic de données à envoyer simultanément dans le réseau, le besoin de maintenir les routes dans l'approche réactive a créé une charge supplémentaire. Ceci rend la bande passante moins disponible et les collisions plus fréquentes.

La mobilité des nœuds ajoute plus de complexité à la livraison de paquets aux destinations. D'une part, la rupture des liens résultante rend les routes plus longues (source, destination). D'autre part, donner un avis à la source au sujet de cette rupture est évidemment retardé à cause des délais de propagations des messages de contrôle. Ainsi, le taux des pertes des paquets est de plus en plus augmenté. Dans l'article (Clausen et al., 2010), les courbes taux de paquets livrés en fonction de mobilité montrent une décroissance des paquets livrés pour OLSR par rapport aux deux autres protocoles AODV et DSR. Le recours aux files d'attente dans chaque nœud dans l'approche réactive explique cette différence de performance. Dans le cas de non-disponibilité de route valide vers la destination, le paquet IP doit prendre place dans la file d'attente, ce qui diminue le risque de paquets perdus.

Arun Kumar *et al.* (2008) évaluent les performances du réseau MANET en soulignant un flux de données variable (*Variable Bit Rate*) au lieu d'un flux de données constant qui est le cas le plus fréquent dans les travaux de recherche. Les performances sont évaluées en fonction du nombre de nœuds. La comparaison était menée entre les protocoles OLSR, DSDV, DSR et AODV. Ainsi, une comparaison entre l'approche proactive et l'approche réactive est encore établie. En augmentant le nombre de noeuds, OLSR garde un taux de paquets livrés satisfaisant, au-dessus de 0,8. Les taux de paquets livrés d'AODV et de DSR sont proches de celui d'OLSR. DSDV a la moins bonne performance avec un taux de paquets livrés au-dessous de 0,8.

Dans (Desai & Patil, 2014), une évaluation des protocoles réactifs AODV, DSR et AOMDV et des protocoles proactifs OLSR et DSDV, est établie en fonction du nombre de nœuds dans un scénario caractérisé par une mobilité variante entre 10 m/s et 50 m/s. AOMDV *(Ad Hoc Multipath Distance Vector*) vise à découvrir plusieurs routes entre la source et la destination. Le taux de paquets livrés est évalué pour les cinq protocoles. Pour un nombre de nœuds allant de 50 à 100, le taux de paquets livrés atteint la valeur maximale pour AODV et DSR. OLSR et AOMDV gardent aussi un haut taux de paquets livrés supérieur à 0,91. DSDV possède le taux

le plus bas, soit à inférieur à 0,9. Plus le nombre de nœuds dépasse100, plus le taux de paquets livrés pour les cinq protocoles baisse. Avec la mobilité et le nombre croissant de nœuds, le taux de paquets livrés pour AODV est diminué jusqu'à la valeur 0,4. Pour un nombre de nœuds supérieur à 150, le taux d'OLSR dépasse celui d'AODV, d'AOMDV et de DSR.

Subramanya Bhat *et al.* (2011) évaluent les trois types de protocoles de routage en fonction du nombre de nœuds. L'apport de cet article est la comparaison des comportements des protocoles selon deux scénarios : statique et mobile. L'approche réactive est présentée par AODV, DSR et LAR. L'approche proactive est présentée par OLSR. L'approche hybride est présentée par ZRP, décrit au chapitre 1. Avec les paramètres de simulation indiqués, AODV, DSR et LAR conservent pratiquement le même niveau de taux de paquets livrés dans les deux scénarios statique et mobile. OLSR maintient un bon taux de paquets livrés comparable à celui d'AODV et de DSR dans le scénario statique. Dans le scénario mobile, OLSR a un abaissement de taux de paquets livrés jusqu'à 0,7 à cause de la nature de l'approche proactive. ZRP a le taux de paquets livrés le plus bas pour les deux scénarios. ZRP présente une amélioration de taux de paquets livrés dans le scénario mobile grâce au déploiement de l'approche hybride combinant les avantages des approches réactifs et proactifs.

Une comparaison des trois approches de routage est établie dans l'article (Qasim *et al.*, 2008) souligne l'effet de transmissions TCP sur les performances de trois protocoles choisis. L'approche proactive est présentée par OLSR. L'approche réactive est présentée par AODV. L'approche hybride est présentée par *Temporary Ordered Routing Algorithm Protocol* (TORA). Le choix de transmissions TCP vise à évaluer l'effet de la congestion due au trafic de contrôle sur les performances de trois protocoles. Ainsi, une évaluation de la fiabilité et l'efficacité de mécanismes de routage peuvent être obtenues. Concernant le taux de paquets livrés, cette performance est évaluée dans un scénario défini par 50 noeuds mobiles. Pour OLSR, le réseau est divisé en *clusters* et chaque *cluster* exploite le mécanisme MPR. Dans un intervalle de temps de 900 secondes, OLSR possède le taux de paquets livrés le plus élevé avec des valeurs égales ou supérieures à 0,8. Pour AODV, le temps d'expiration d'une route active est égal à 30 se-

condes. Une augmentation de *Time To Live* (TTL) vise à diminuer la fréquence de la recherche des routes. Le taux de paquets livrés par AODV est moins qu'OLSR avec des valeurs qui ne dépassent pas le 0,54. TORA a le plus faible taux de paquets livrés qui est expliqué par l'augmentation de congestion liée au protocole TCP. TORA a ainsi une difficulté de convergence, ce qui mène à un abandon élevé des paquets de données.

2.1.2 Délai de bout en bout

Comme indiqué dans (Clausen *et al.*, 2010), l'approche proactive gagne évidemment ce défi grâce à la disponibilité immédiate de routes. Quand un paquet arrive à un nœud intermédiaire, il est retransmis ou laissé tomber immédiatement. Tandis que les protocoles réactifs ont recours aux files d'attente. Dans le scénario caractérisé par une mobilité croissante, la courbe délai de bout en bout en fonction de la mobilité montre qu'OLSR possède le délai minimal par rapport aux AODV et DSR.

Dans (Dugaev *et al.*, 2015), la courbe délai en fonction du nombre de connexions (*number of connection*) montre que DSDV possède la plus basse performance. Malgré que DSDV déploie le mécanisme proactif, la valeur de l'intervalle de temps entre deux transmissions successives de message de contrôle topologique est large (5 seconde). Dans le cas de rupture d'un lien le long du trajet, DSDV est lent dans sa réaction à cause de l'intervalle de mise à jour topologique. Ce qui augmente le délai. OLSR performe bien grâce à l'intervalle de mise à jour topologique convenable (4 seconde).

En augmentant le nombre de noeuds et avec un flux de données variable, Arun Kumar *et al.* (2008) montrent une autre fois la correspondance entre le délai minimal et l'approche proactive. OLSR et DSDV maintiennent des délais minimaux, soit au-dessous de 0,2 secondes. AODV possède le délai de bout en bout le plus élevé. La performance de DSR est proche de celle

d'OLSR et de DSDV.

Desai & Patil (2014) confirment de plus que l'approche proactive mène à un délai minimal. Avec une mobilité et une augmentation du nombre de nœuds, OLSR est meilleur qu'AODV. Pour un nombre de nœuds égal à 100, AODV a le délai le plus élevé, soit égal à 0,07 ms. Pour le même nombre, OLSR a le délai le plus court, soit inférieur à 0,015 ms. AOMDV et DSR ont des délais plus courts qu'AODV.

Dans (Subramanya Bhat *et al.*, 2011), l'évaluation de délai de bout en bout est établie en fonction de la densité variante du réseau dans deux scénarios mobile et stationnaire. La mise à jour périodique de tableau de routage dans l'approche proactive minimise le délai de la maintenance des trajets et ainsi minimise le délai de bout en bout. OLSR présente le délai de bout en bout minimal pour les nombres de noeuds 25, 50 et 75 dans le scénario mobile. ZRP a aussi un délai de bout en bout proche d'OLSR dans le scénario mobile pour les mêmes nombres de nœuds.

Comme indiqué avant, Qasim *et al.* (2008) évaluent les trois approches de routage dans un scénario caractérisé par des transmissions TCP afin d'évaluer l'effet de la congestion due aux trafics de contrôle. OLSR possède le plus court délai de bout en bout avec une valeur approximative de 0,4 ms. AODV présente un moyen égal à 1,5 ms. Le protocole hybride TORA est le plus influencé par la congestion dans le réseau. Son délai de bout en bout est égal approximativement à 3,2 ms.

Mbarushimana & Shahrabi (2007) évaluent les performances de trois protocoles OLSR, AODV et DSR avec un débit binaire constant (*Constant Bit Rate*) en analysant l'effet des différents facteurs du réseau ad hoc comme la densité du réseau, la mobilité et le nombre de flux de données dans le réseau. Ainsi, l'évaluation établie touche la majorité des contraintes rencontrées dans le réseau ad hoc. Concernant le délai de bout en bout, la comparaison de trois protocoles

est basée sur l'effet des contraintes mentionnées. Le premier scénario est basé sur 10 flux CBR dont la valeur de débit varie de 12.5 Kbps à 150 Kbps. Le nombre de nœuds est fixé à 50. Les nœuds ont des vitesses entre 0 m/s et 20 m/s. L'avantage de la disponibilité immédiate des routes laisse OLSR avoir le délai le plus court indépendamment de la charge de trafic (*load*). Contrairement à OLSR, l'augmentation de charge de trafic dans le réseau est accompagnée par une augmentation de délai pour AODV et DSR. Avec la congestion dans le réseau, DSR possède le délai le plus élevé avec un trafic modéré ou important. Ce délai est expliqué dans l'article (Mbarushimana & Shahrabi, 2007) par le mécanisme de découverte de routes lent à cause que DSR a besoin de répondre à tous les RREQs afin de sauvegarder les routes alternatives dans son cache de route. AODV montre un délai de bout en bout plus court que DSR, vu qu'AODV a besoin de répondre uniquement au premier RREQ. Le deuxième scénario vise à évaluer l'effet du nombre de flux sur le délai de bout en bout. Ainsi, le nombre de flux CBR varie entre 5 et 30. Chaque flux génère 25 Kbps. Pratiquement, le délai de bout en bout d'OLSR n'est pas influencé par l'augmentation du nombre de flux. OLSR garde le délai le plus court dans le deuxième scénario. DSR montre le délai le plus élevé. AODV est meilleur que DSR, mais son délai dépasse le délai d'OLSR. Pour le troisième scénario décrit par une augmentation du nombre de nœuds entre 25 et 100, l'ordre reste le même, mais avec une augmentation du délai pour les trois protocoles. Dans le scénario défini par une mobilité croissante, OLSR possède le délai le plus court en gardant approximativement une valeur constante. Contrairement aux scénarios précédents, DSR est meilleur que AODV dans le cas de mobilité moyenne des noeuds, soit inférieure à 10 m/s. En augmentant la vitesse, DSR et AODV ont des délais proches.

2.1.3 Volume du trafic de contrôle

La congestion au niveau de la bande passante, à cause de la redondance du trafic de contrôle, est parmi les défis de la technologie ad hoc. L'envoi périodique des messages de contrôle par le mécanisme proactif introduit une utilisation constante de la bande passante, indépendamment de la mobilité et de trafic de données. Le volume de trafic de contrôle d'AODV et de DSR est

augmenté en fonction de la mobilité et le nombre du trafic. Ce qui est confirmé par la courbe volume de trafic de contrôle en fonction du nombre de trafic (*traffic streams*) et la courbe volume du trafic de contrôle en fonction de la mobilité dans (Clausen *et al.*, 2010).

Il est prévisible que le volume de trafic de contrôle augmente avec le nombre de noeuds. Dans (Arun Kumar *et al.*, 2008), une comparaison de deux protocoles proactifs OLSR et DSDV et deux protocoles réactifs AODV et DSR est établie. Les courbes montrent qu'OLSR a le volume de trafic de contrôle le plus bas *(Routing overhead)* par rapport à DSDV et AODV. La courbe d'AODV a une allure croissante en fonction du nombre de nœuds et avec une vaste différence par rapport aux autres protocoles à cause de la diffusion de paquets de routage. DSR a une performance proche des protocoles proactifs expliquée par l'effet de son cache de route qui fournit des routes sauvegardées.

Comme indiqué avant, Mbarushimana & Shahrabi (2007) comparent les trois protocoles OLSR, AODV et DSR en soulignant l'effet de la mobilité, de nombre de flux, de la densité du réseau et la quantité de trafic. Dans les différents scénarios traités, OLSR a le trafic de contrôle le plus élevé, mais ce trafic haut est le coût des performances élevées d'OLSR côté débit et délai de bout en bout. Pour l'approche réactive, DSR a généralement un trafic de contrôle plus bas qu'AODV. En augmentant le nombre de flux, les sources génèrent plus de trafics (sources trafic). Plus de routes vers un nombre plus grand des destinations vont être recherchées. Ainsi, le volume de trafic de contrôle est augmenté chez les protocoles réactifs AODV et DSR. Dans le cas de mobilité élevée, le volume de trafic de contrôle de DSR est supérieur à celui d'AODV.

2.1.4 Longueur de la route

Chaque protocole de routage cherche à fournir le trajet le plus court pour l'acheminement de trafic de données. Les algorithmes distribués de routage sont intégrés dans la conception de protocoles de routages, mais la réaction de protocoles vis-à-vis la mobilité et le trafic à ache-

miner diffère d'une approche à une autre. Dans (Clausen *et al.*, 2010), OLSR garde les trajets les plus courts indépendamment de nombre de trafic (*traffic streams*) en maintenant un taux de paquets livrés élevé. Tandis qu'AODV, la procédure de demande de route (RREQ) peut aboutir à une sélection du trajet plus long que le trajet optimal en termes de nombres de sauts, à cause de la congestion fort probablement accrue dans le réseau. En augmentant la mobilité, OLSR maintient les trajets les plus courts. Dans le cas de mobilité intense, les messages de contrôle de topologie (TC) peuvent ne pas être diffusés correctement. Des lacunes peuvent avoir lieu dans la base de données topologique de chaque nœud n'ayant pas reçu tous les messages TC diffusés. Ce qui explique la limitation relative des trajets enregistrés dans le tableau de routage. Quand l'acheminement d'un paquet vers une destination trop lointaine est demandé, il existe un grand risque que le paquet soit laisser tombé. En comparant les trajets d'OLSR et AODV, les trajets d'ADOV sont plus longs, même dans le cas d'une mobilité associée avec un taux de paquets livrés élevé chez OLSR. AODV peut acheminer des trafics vers des destinations lointaines en suivant un trajet plus long que l'optimal.

2.1.5 Débit

Dugaev *et al.* (2015) comparent les protocoles OLSR, AODV, DSR et DSDV dans un scénario défini par une mobilité réduite et une génération arbitraire des paquets de données. La courbe de débit montre qu'OLSR, AODV et DSDV ont des résultats proches. DSR présente un débit plus élevé grâce à la possibilité de déploiement de routes alternatives sauvegardées dans son cache de route dans le cas de défaillance de liens.

Varshney *et al.* (2016) se concentrent sur la comparaison de performances de deux protocoles proactifs OLSR et STAR. Comparer deux protocoles proactifs vise à comparer deux protocoles n'ayant pas le même volume de trafic de contrôle. STAR, contrairement à OLSR, n'exige pas le déploiement du trajet le plus court praticable *(Shortest practicable paths)*. Un abaissement dans le trafic de contrôle est ainsi possible. Les scénarios de simulation sont décrits par une augmentation du nombre de noeuds, une mobilité variante et un flux de données constant. La

courbe de débit montre qu'OLSR présente un débit plus élevé que STAR, mais en augmentant le nombre de nœuds jusqu'au 50, OLSR connaît un abaissement de débit par rapport à STAR.

L'étude de (Kulla *et al.*, 2010) nous aide à avoir une description de performances d'OLSR et AODV pour quatre scénarios dont le nombre de nœuds est faible, soit égal à 7. Le choix d'un petit nombre de nœuds a pour but de laisser la focalisation des résultats sur le comportement élémentaire de chaque nœud. Le premier scénario est un scénario statique où le débit pour les deux protocoles est maximal grâce à la réussite de l'établissement des routes pour les deux protocoles. Le deuxième scénario est caractérisé par le mouvement de source vers la destination. Quand la source et la destination perd la ligne de vue (*Line of Sight*), le débit a atteint zéro car la source est incapable d'établir une route vers la destination. Cette diminution de débit est plus remarquable chez AODV. Le troisième scénario est défini par le mouvement de la destination. Les mêmes résultats que le deuxième scénario sont conclus. Le quatrième scénario où la source et la destination sont toutes les deux en mouvement confirme plus la sensibilité d'AODV envers le changement et la perte de routes en raison de son besoin de rétablir la route entière avant l'envoi des données.

Le débit est influencé par la congestion du réseau et la non-disponibilité de route, ce qui cause généralement un abandon des paquets. Desai & Patil (2014) comparent le débit d'OLSR, AODV, AOMDV, DSR et DSDV dans un scénario caractérisé par une mobilité et un nombre de noeuds variant. Jusqu'à un nombre de 60 nœuds, tous les protocoles ont montré des performances proches. Au-delà de 60 noeuds, AODV, AOMDV et DSR dépassent légèrement OLSR et DSDV.

Dans les deux scénarios statiques et mobiles ainsi qu'avec une densité de noeuds croissante, le débit est évalué pour les trois approches de protocoles de routage (Subramanya Bhat *et al.*, 2011). Le protocole proactif OLSR et les trois protocoles réactifs AODV, DSR et LSR montrent

pratiquement le même débit dans le scénario statique avec une densité de noeuds croissante. Quand la mobilité associée avec l'augmentation de la densité du réseau prend place, le débit est diminué pour les quatre protocoles. Le protocole hybride ZRP a un débit plus inférieur que les autres protocoles indiqués. Dans le cas de mobilité et une densité de noeuds croissante, le débit est amélioré grâce à l'activation de la procédure de recherche de route dans l'inter-zone et l'intra-zone.

Dans un scénario avec 50 nœuds mobiles et avec l'exploitation de protocole TCP, Qasim *et al.* (2008) évaluent le débit pour les trois protocoles OLSR, AODV et TORA. OLSR possède le plus haut débit, approximativement stable autour de la valeur 1620 Kbps de paquets de données. Avec l'augmentation de nœuds sources déployés par AODV, une croissance de paquets de routage est résultante jusqu'à la valeur de 8 Kbps de paquets de données dans un intervalle de temps de 900 secondes. Le débit ne dépasse pas la valeur 7 Kbps qui présente un grand écart par rapport au débit d'OLSR. Pour le mécanisme de découverte de route, TORA déploie l'envoi périodique de message Hello chaque 3 secondes. Le débit enregistré par TORA est meilleur que celui d'AODV, avec une valeur qui ne dépasse pas 14,5 Kbps de paquets de données.

Une évaluation des débits des trois protocoles OLSR, AODV et DSR pour différents scénarios est établie dans l'article (Mbarushimana & Shahrabi, 2007). L'apport de cet article concerne l'analyse des effets de la majorité des contraintes rencontrées dans le réseau ad hoc. Le premier scénario est décrit par la variation de trafic CBR. Le nombre du flux de données est fixé à 10 et le nombre de noeuds est égal à 50 ayant une vitesse constante. OLSR a un débit croissant linéairement, même pour un trafic CBR élevé. Ce qui présente un bon avantage. Pour un trafic CBR léger autour de 50 Kbps, le débit de DSR est comparable à OLSR. En augmentant le trafic, le débit de DSR n'a pas dépassé le 75 Kbps. AODV obtient une meilleure performance que DSR. Le débit s'est saturé à 125 Kbps. En augmentant le nombre du flux, comme deuxième scénario, OLSR continue à avoir le débit le plus élevé avec une croissance linéaire continue. Contrairement à OLSR, le débit de DSR s'est saturé après 15 flux. AODV montre un débit

croissant en fonction du nombre du flux sans aucune saturation, contrairement à DSR. En variant le nombre de nœuds de 25 à 100, le trafic de routage devient plus redondant, ce qui mène à une congestion dans le réseau. Dans un tel cas et avec un réseau étendu, la communication entre les noeuds peut être restreinte à une communication locale. Le débit pour chaque nœud devient ainsi limité. Une diminution de débit total du réseau résulte. En augmentant le nombre de noeuds, AODV montre le débit le plus bas. DSR et OLSR ont des débits proches. Dans le dernier scénario mentionné, les vitesses de différents noeuds augmentent. Cette mobilité croissante peut causer un abandon fréquent des paquets à cause de non-disponibilité des routes pour les différents protocoles. OLSR présente un débit relativement stable en fonction de la vitesse. Son débit est plus élevé par rapport à celui de DSR et d'AODV. Cette supériorité est expliquée par la capacité d'OLSR de détecter les défaillances de liens. Pour des basses vitesses, DSR a un débit comparable à celui d'OLSR. Mais en augmentant la vitesse, un abaissement de débit a lieu. Ce comportement peut être expliqué par la non-validité des routes sauvegardées dans le cache de route de DSR. Pour des vitesses supérieures à 12 m/s, la performance d'AODV est meilleure que celle de DSR.

2.1.6 Taux de perte de paquets

Dugaev *et al.* (2015) montrent l'intérêt de la détection périodique des états de liens par le message de contrôle dans l'approche proactive. Vu que chaque noeud est au courant des changements des états de lien dans son voisinage et le réseau entier, le tableau de routage est basé sur une base de données fiable et valide. OLSR et DSDV ont ainsi un bas taux de perte de paquets. AODV présente le taux de perte de paquet le plus élevé.

Comme décrit avant, dans l'étude de (Kulla *et al.*, 2010), AODV montre un taux de perte de paquets plus élevé qu'OLSR quand un changement ou perte de routes prennent place.

2.1.7 Conclusion

La revue de littérature montre qu'il n'y a pas un protocole idéal. Chaque protocole a ses avantages et ses désavantages. Les comportements des protocoles diffèrent selon les scénarios choisis. Le but d'exploitation de la technologie ad hoc diffère d'un réseau à l'autre. Pour les réseaux ad hoc exigeant un débit élevé, AODV est un candidat approprié. OLSR montre aussi le maintien d'un bon débit en fonction de la quantité de trafics et la densité du réseau. Pour une congestion élevée, OLSR présente un débit meilleur que AODV et le protocole hybride TORA. La mobilité intense est le facteur qui peut diminuer le débit de la majorité des protocoles. L'approche proactive est le choix approprié si le but d'exploitation de la technologie ad hoc est d'avoir un délai de bout en bout minimal. Comme le montre la revue de littérature, OLSR est le meilleur choix pour garantir un délai de bout en bout minimal grâce à la disponibilité immédiate des trajets. Concernant le taux de paquets livrés en fonction de la quantité de trafic, OLSR a un taux de paquets livrés plus important que celui d'AODV et DSR. En augmentant le nombre de noeuds dans le réseau, OLSR garde toujours un taux de paquets livrés satisfaisant. Mbarushimana & Shahrabi (2007) concluent que si le réseau ad hoc n'exige pas la conservation énergétique, OLSR est le meilleur choix, car il donne les performances les plus élevées.

Tableau 2.1 Scénario1 : Nombre de noeuds augmente

	Taux de paquets livrés	**Délai de bout en bout**	**Volume de trafic de contrôle**	**Débit**
Optimized Link State Routing (OLSR)	Très satisfaisant	Très satisfaisant	Moins satisfaisant	Très satisfaisant
Destination Sequenced Distance Vector (DSDV)	Satisfaisant	Satisfaisant	Satisfaisant	-
Ad Hoc On Demand Distance Vector (AODV)	Satisfaisant	Moins satisfaisant	Moins satisfaisant	Satisfaisant
Dynamic Source Routing (DSR)	Satisfaisant	Moins satisfaisant	Satisfaisant	Très satisfaisant
Zone Routing Protocol (ZRP)	Moins satisfaisant	Satisfaisant	-	Moins satisfaisant

Tableau 2.2 Scénario2 : Nombre de flux de trafic augmente

	Taux de paquets livrés	**Délai de bout en bout**	**Volume de trafic de contrôle**	**Débit**	**Longueur de la route**
Optimized Link State Routing (OLSR)	Très satisfaisant	Très satisfaisant	Satisfaisant	Très satisfaisant	Très satisfaisant
Ad Hoc On Demand Distance Vector (AODV)	Satisfaisant	Moins satisfaisant	Moins satisfaisant	Satisfaisant	Satisfaisant
Dynamic Source Routing (DSR)	Satisfaisant	Moins satisfaisant	Moins satisfaisant	-	Moins satisfaisant

Tableau 2.3 Scénario3 : Trafic de contrôle augmente

	Taux de paquets livrés	**Délai de bout en bout**	**Débit**
Optimized Link State Routing (OLSR)	Très satisfaisant	Très satisfaisant	Très satisfaisant
Ad Hoc On Demand Distance Vector (AODV)	Moins satisfaisant	Satisfaisant	Moins Satisfaisant
Temporary Ordered Routing Algorithm (TORA)	Moins satisfaisant !	Moins satisfaisant	Satisfaisant

2.2 Le protocole OLSR

OLSR est parmi les protocoles les plus cités dans les travaux de recherche traitant le routage dans le réseau ad hoc. Il constitue une référence pour l'approche proactive aidant à établir des comparaisons valides. Comme décrit dans les sections précédentes de ce chapitre, OLSR garde généralement de bonnes performances en fonction de la variation des différents facteurs influençant le réseau ad hoc. L'idée générale est que OLSR est adéquat pour un réseau large et

dense grâce à son déploiement du mécanisme MPR. Un mécanisme permet d'alléger la charge de trafic de contrôle par la diffusion des messages de contrôles par l'intermédiaire des nœuds choisis. Chaque nœud a la possibilité d'accéder à n'importe quel voisin se plaçant au voisinage de 2 sauts et ayant un lien symétrique avec un nœud MPR. Malgré le mécanisme MPR est déployé dans OLSR, l'approche proactive est réputée pour une charge de trafic de contrôle élevée à cause de la diffusion périodique des messages de contrôles. Mais cette diffusion périodique permet à OLSR de détecter facilement et rapidement les défaillances de liens très fréquentes dans le réseau ad hoc. Ainsi, on garantit la validation des trajets plus courts établis par l'algorithme de routage. OLSR est désigné spécifiquement pour le MANET, en respectant les caractéristiques du réseau ad hoc mobile comme la limite de la bande passante et la dynamique de la topologie. Concernant la perte de messages de contrôle, OLSR présente une flexibilité. La première version d'OLSR est présentée par la RFC 3626 qui était énormément exploitée dans le routage dans le réseau ad hoc. La RFC 3626 englobe la procédure de la découverte de voisinage, la diffusion MPR, les formats de messages de contrôle, le traitement de données de la topologie et le calcul des trajets. La découverte de voisinage est établie par l'envoi périodique de messages Hello. L'échange de ces messages dans le voisinage d'un saut de chaque nœud permet de construire l'ensemble de liens, l'ensemble de voisins qui fixe les voisins avec liens symétriques et l'ensemble de voisins à 2 sauts. Cet ensemble présente le premier pas pour la sélection des nœuds MPR et la sélection de trajets. Une mise à jour de l'état de lien dans chaque uplet de lien est établie en fonction des informations reçues et le facteur de temps. OLSR est parmi les protocoles qui acceptent de sauvegarder les liens unidirectionnels en attente de l'établissement de la bidirectionnalité. Le message Hello est composé de plusieurs champs visant à identifier le nœud source et les nœuds voisins. La Figure 2.1 illustre le format proposé du message Hello dans la RFC 3626.

OLSR dans sa première version donne la possibilité d'utiliser plusieurs interfaces pour chaque nœud. L'adresse principale est l'adresse de l'interface appliquant l'OLSR. Un message appelé *Multiple Interface Declaration* (MID) sert à la déclaration des configurations des interfaces. La Figure 2.2 décrit ce message MID.

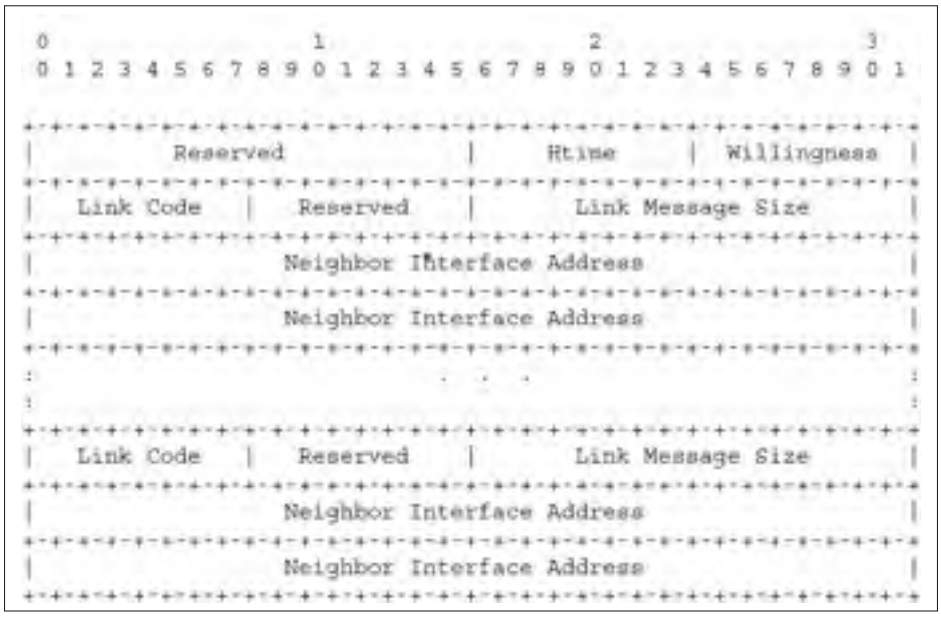

Figure 2.1 Le format proposé du message Hello dans la RFC 3626.
Tirée de RFC 3626

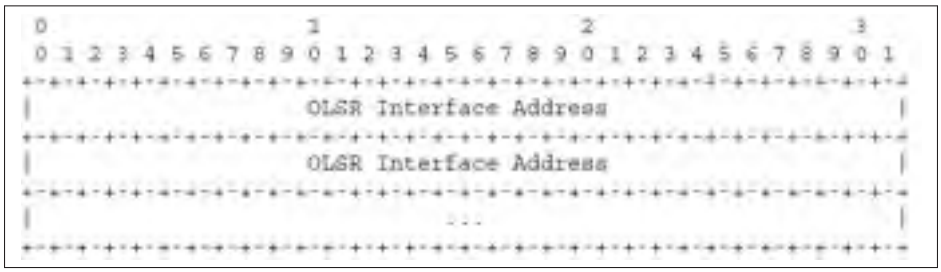

Figure 2.2 Le format proposé du message MID dans la RFC 3626.
Tirée de RFC 3626

Le troisième type du message de contrôle exploité dans la RFC 3626 est le message de contrôle de topologie (message TC) assurant la diffusion des informations topologiques dans le réseau. La Figure 2.3 illustre le format proposé dans la RFC 3626 de message de contrôle de topologie.

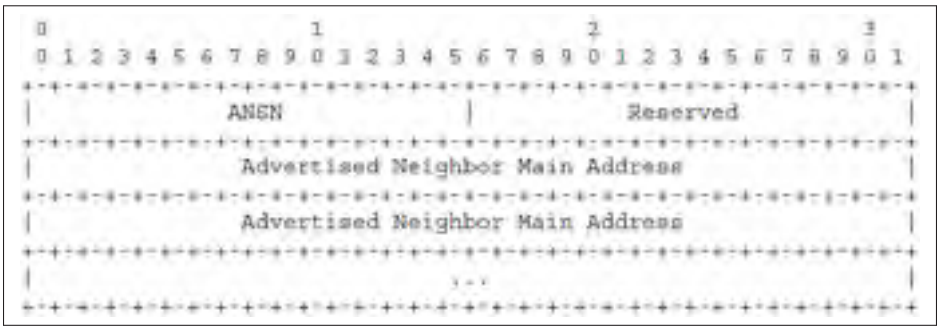

Figure 2.3 Le format proposé du message TC dans la RFC 3626.
Tirée de RFC 3626

Advertised Neighbor Sequence Number (ANSN) est un indicatif de la fraîcheur des informations incluses dans le message TC. Plus la valeur d'ANSN est élevée, plus les informations incluses sont plus récentes. Le champ *Advertised Neighbor Main Address* contient l'adresse principale (*main address*) d'un voisin de nœud source. La diffusion de message est périodique à chaque intervalle TC. En plus de l'envoi périodique, une transmission du message peut avoir lieu comme réaction des défaillances des liens et le changement arbitraire de la topologie locale. Après la validation et le traitement de message, le nœud récepteur crée un uplet *(Topology Tuple*) contenant la destination à ajouter, le nœud prédécesseur et le temps d'expiration de l'uplet.

Concernant la sélection de trajets, OLSR vise à sauvegarder toutes les destinations dans le tableau de routage de chaque nœud et rend les trajets les plus courts disponibles immédiatement. Le tableau de routage est structuré en des uplets contenant l'adresse de la destination, l'adresse de voisin symétrique de la destination et le nombre de sauts séparant la source de la destination. OLSR donne la liberté pour chaque nœud d'être sélectionné ou non dans le trajet selon la valeur associée au paramètre (*Willingness*). Le routage est établi en suivant un algorithme du trajet le plus court en termes du nombre de sauts. Chaque variation dans l'ensemble de liens ou l'ensemble des voisins engendre des nouveaux calculs des trajets.

2.2.1 OLSRV2

L'exploitation continue d'OLSR encourage les chercheurs à améliorer ses mécanismes et ses procédures. Une deuxième version (OLSRV2) est apparue en 2014 et est standardisée par IETF dans la RFC 7181. Le premier but d'OLSRV2 est d'améliorer la capacité de l'adaptation à la topologie dynamique. Le deuxième but est d'identifier toutes les destinations dans le réseau et avoir des données suffisantes de liens permettant d'appliquer l'algorithme de routage pour les destinations disponibles. Comme différence par rapport à OLSRV1, les mécanismes et les procédures à appliquer lors de la découverte de voisinage sont définis dans un autre protocole appelé *Mobile Ad Hoc Neighborhood Discovery Protocol* (NHDP) apparu en 2011 et présenté dans la RFC 6130. Ce qui rend la découverte de voisinage et la détection des états de liens indépendantes d'OLSR et permet leurs exploitations dans d'autres contextes ou leurs associations aux autres protocoles. OLSRV2 exploite NHDP pour la préparation d'ensemble de liens, d'ensemble de voisins à 1 saut et l'ensemble de voisins à 2 sauts pour chaque nœud. Une base de données est donc conçue pour chaque nœud donnant une idée de la topologie locale (2 sauts). OLSRV2 donne la possibilité à apporter des modifications aux ensembles de NHDP par l'ajout d'autres champs dans chaque uplet de l'ensemble, par exemple, un champ décrivant la métrique associée à chaque adresse ou la qualité de lien ajouté aux uplets de l'ensemble de liens. Aussi, l'exploitation des données issues de la couche liaison est possible quand ces données sont disponibles et valides. OLSRV2 garde toujours la sélection des noeuds MPR. La sélection de base est mentionnée dans NHDP, mais OLSRV2 ajoute la possibilité pour le nœud de choisir d'être un MPR de routage (*Routing MPR*) ou un MPR de diffusion (*flooding MPR*). Les noeuds présentant des MPRS de routage ne sont pas obligés de diffuser les états des liens. Dans OLSRV1, la structure topologique est présentée par un seul ensemble composé par des uplets, chacun définit le nœud destination, l'adresse de nœud permettant d'accéder à la destination dans un seul saut, un numéro de séquence et le temps d'expiration. La structure topologique dans OLSRV2 est plus riche et contenant plusieurs ensembles visant à sauvegarder les informations valides dans le temps des noeuds distants originaires des messages TC, les liens entre les noeuds comme décrits dans les messages TC reçus et les adresses routables

(*routable address*) disponibles comme des IP destinations.
Des améliorations aux formats des messages de contrôle sont présentées dans OLSRV2. Plusieurs structures et mécanismes proposés peuvent être associés aux messages de contrôle déployés afin de garantir une signalisation fiable dans le voisinage de chaque nœud et par conséquent le réseau entier. Les définitions et le guide d'utilisation de ces mécanismes sont détaillés dans la RFC 5444. OLSRV2 accepte l'extensibilité interne de message par l'ajout des données ou l'extensibilité externe par l'utilisation d'autre types de messages si c'est nécessaire, comme indiqué dans la RFC 5444. Les messages de contrôle mentionnés dans OLSRV2 sont essentiellement le message Hello et le message TC.

2.2.1.1 Le message de contrôle de lien Hello

En raison de plusieurs facteurs comme la mobilité et l'interférence, le changement continu de la connectivité dans le réseau ad hoc entraîne une variation continue des états de lien entre les différents noeuds. La diversité engendrée (lien unidirectionnel, lien symétrique, lien perdu) nécessite un moyen d'échange des informations capable de diffuser localement les données utiles et suffisantes. Ainsi, le message Hello, selon la RFC 5444, inclut des champs servant à donner l'identité du nœud transmetteur et les paramètres de message comme le nombre de sauts et le numéro de séquence. Ces champs sont essentiellement.

- HELLO.msg_orig_addr : L'adresse de routeur ou de l'interface MANET qui a généré le message. Il est obligatoire d'inclure ce champ.
- HELLO.msg_hop_limit : le Hello est transmis une seule fois par le nœud source, donc la valeur de ce champ doit être égale à 1.
- HELLO.msg_hop_count : C'est le nombre de sauts achevés.La valeur maximale doit être égale à 1.
- HELLO.sequence_number : C'est un indicateur unique indiquant le numéro de séquence de message Hello généré par un nœud. A chaque fois, un message Hello est généré, le numéro de séquence est incrémenté de 1. Ce numéro est utilisé pour s'assurer que le message n'est pas transmis plus qu'une fois par le nœud.

Le Hello sert du mécanisme TLV (*type length value*) pour associer des attributs au message, selon la RFC 5497. Ces mécanismes présentent des structures permettant d'organiser des données de temps et de voisinage. Parmi les structures TLV utilisées dans le Hello, le message TLV sert essentiellement à agréger des valeurs de temps au message. Deux champs sont associés au message Hello.

- HELLO.message TLV.Validity_time : indiquant la durée de temps pour laquelle le contenu de message est valide. Ce paramètre de temps contribue à la mise à jour de durée de validité des informations dans la base de données du nœud récepteur, d'où l'obligation de son intégration dans le message. Dans le cas contraire, le message est non valide.
- HELLO.message TLV.Interval_time : signalant la durée séparant la transmission de deux messages Hello issus de la même source. Cette valeur peut être relative à un nœud ou identique pour tous les nœuds. La valeur proposée dans la RFC 6130 est 2 seconde pour tous les nœuds.

Un autre mécanisme est utilisé également dans le format proposé du message qui est le bloc TLV à adresse *(address block TLV)*. En effet, le protocole NHDP permet d'exploiter trois types de blocs d'adresse dont l'inclusion est obligatoire ou potentielle.

- Local_IF TLV : Les adresses des interfaces de routeur sauvegardées dans l'ensemble des interfaces locales doivent être inclues dans un bloc TLV à adresse de type (LOCAL_IF) (RFC 6130). La valeur associée à ce type est égale à (THIS_IF) si l'interface est celle originaire du message Hello. Dans le cas contraire, la valeur est égale à (OTHER_IF).
- LINK_STATUS TLV : C'est le deuxième type de bloc TLV à adresse indiquant l'état de lien entre le nœud émetteur et les nœuds voisins (symétrique, unidirectionnel, perdu).
- OTHER_NEIGHB : C'est le troisième type de bloc TLV à adresse présentant les adresses des voisins du routeur originaire du message Hello. Si le voisin a un lien symétrique, l'adresse est associée à la valeur symétrique. Dans le cas contraire, la valeur est indiquée comme perdu.

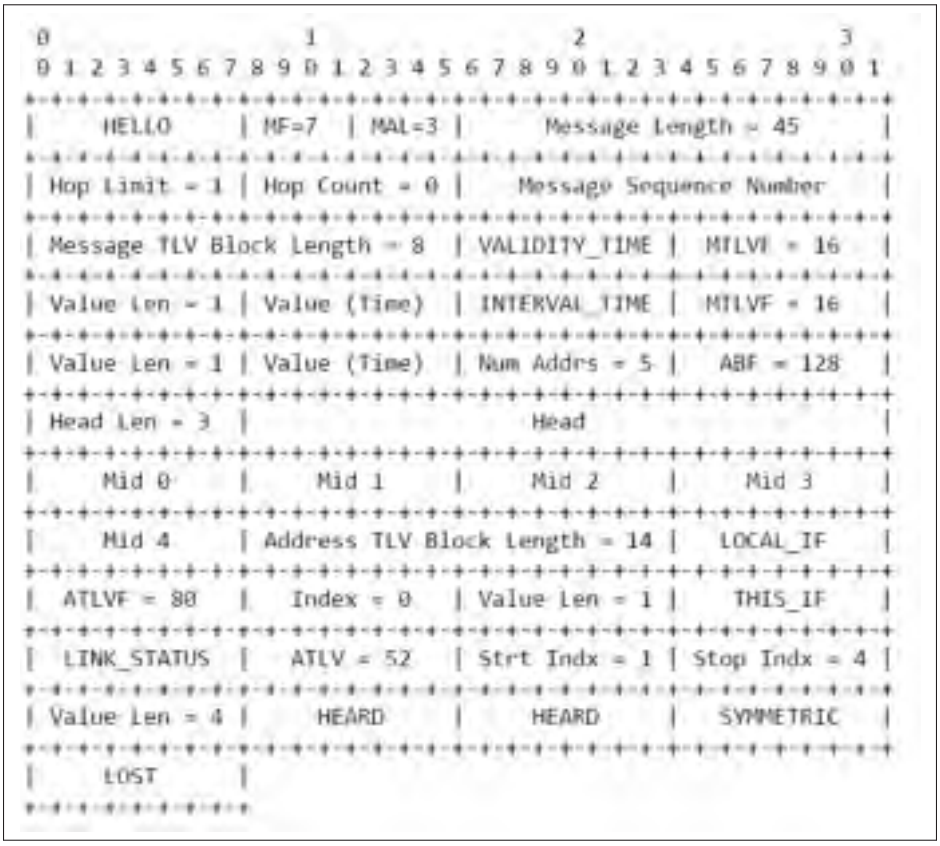

Figure 2.4 Le format du message Hello dans la RFC 6130.
Tirée de RFC 6130

2.2.1.2 le message de contrôle de topologie

Le protocole OLSRV2 identifie le processus de la mise à jour de données topologiques pour chaque nœud. La méthode de la collecte et de sauvegarde de données est présentée dans la RFC 7181. Après la réception d'un message TC, chaque nœud récepteur le transmet en respectant le nombre de sauts permis, après avoir exploité les informations valides incluses dans le message et mis à jour sa base de données topologique. Le message de contrôle de topologie contient les champs suivants :

- TC.msg_orig_addr : L'adresse de routeur originaire du message TC.
- TC.msg_seq_num : C'est le numéro de séquence du message TC.

- TC. msg_hop_limit : C'est la limite de sauts que le message peut avoir. On a la flexibilité pour un seul nœud d'avoir de différentes limites de sauts. La valeur indiquée dans la RFC 7181 est 255 sauts.
- TC.msg_hop_count : C'est le nombre de sauts déjà parcourus par le message TC.
  La valeur de temps introduise dans la structure TLV (*type length value*) définit la durée appropriée séparant deux transmissions consécutives du message TC. Cette durée est flexible selon la spécification RFC 5497 qui donne un guide sur la fixation des intervalles de temps dans le message. En effet, si le réseau ad hoc mobile présente un changement de topologie très dynamique dans le temps, un écart court entre deux messages est approprié. Dans le cas d'une topologie stable, un espacement des heures est accepté (RFC 5497). Ce qu'aide à réduire le volume de trafic de contrôle. Selon RFC 5497, il y a deux types de messages TLV inclus dans le message de topologie de contrôle dont le contenu se spécifie comme suit :
- TC.msg_tlv.type_1=cont_seq_num : Il contient la valeur ANSN issue de la base de données de protocole NHDP, permettant d'indiquer la fraîcheur des informations issues de cette base.
- TC.msg_tlv.type_2= validity_time : Il définit la durée de validation des informations continues dans le message TC reçu.
- TC.msg_tlv.type_3= interval_time : Il définit le temps maximal avant que le prochain message de contrôle de topologie issu du même nœud puisse être transmis.
- TC.address_block_TLV : On doit inclure les adresses des voisins sauvegardés dans le(s) uplet(s) de voisin(s) avec (n_advertised) est vraie indiquant la symétrie de liens. Les blocs TLV à adresse sont de type (NBR_ADDR_TYPE).

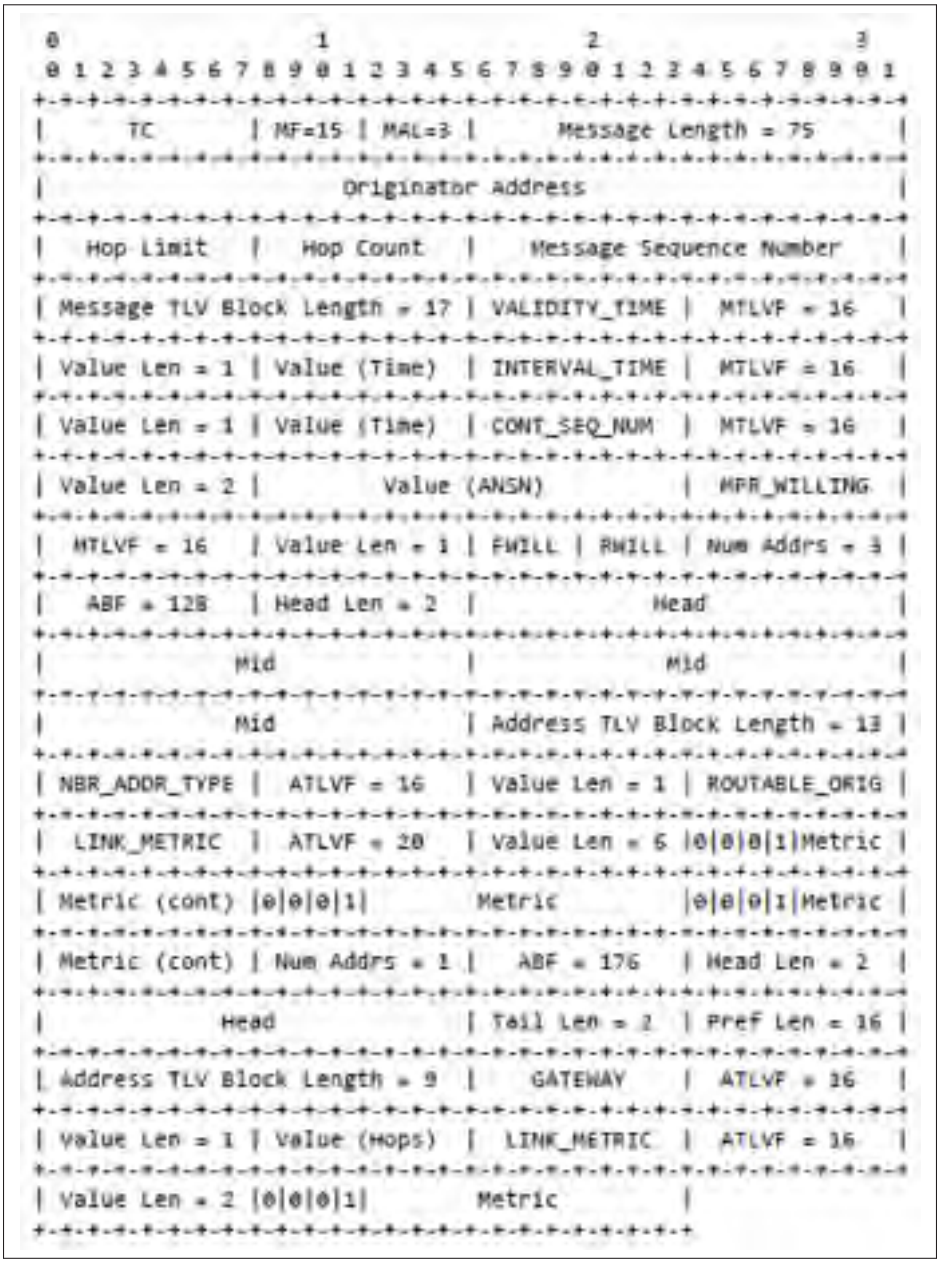

Figure 2.5 Le format de message de contrôle de topologie dans la RFC 7181.
Tirée de RFC 7181

2.3 Conclusion

Le but de ce chapitre est de faire un choix convenable d'un protocole de routage adéquat pour le réseau ad hoc afin de l'intégrer dans un simulateur WIFI fourni. L'intégration sera sous la

forme d'un code Matlab. D'après l'étude comparative, OLSR présente un choix approprié. Les résultats des scénarios traités montrent qu'OLSR garde généralement des bonnes performances côté débit, taux de paquets livrés, délai de bout en bout et taux de paquets perdus. Grâce à sa structure de données riche, on peut avoir une vision mise à jour sur la topologie locale de chaque nœud et sur les états de liens. Par conséquent, une vision sur la topologie globale du réseau peut être conçue. La deuxième version OLSRV2 présente des améliorations et des enrichissements dans le mécanisme MPR et la structure topologique. Un autre avantage d'OLSRV2 est la flexibilité de sa structure des messages qui peut être extensible grâce aux blocs TLV à adresse contenant les données valides de voisinage de chaque nœud. Une extensibilité est possible aussi au niveau des ensembles de données de NHDP. Si c'est conclu comme nécessaire dans l'implémentation, c'est possible d'ajouter un champ dans l'ensemble de lien décrivant la qualité de liaison en fixant une limite pour que le lien puisse être considéré comme accepté. Un choix d'intégrer des métriques est aussi possible, mais la spécification est hors la RFC 7181. Ainsi, c'est possible de concevoir des bases de données de voisinage et de topologie complètes, précises et valides permettant de suivre et analyser les changements dans le réseau ad hoc.

CHAPITRE 3

L'IMPLÉMENTATION DE NHDP ET D'OLSRV2

Ce chapitre introduit les différentes étapes de fonctionnement des deux protocoles NHDP et OLSRV2. Une réalisation sous forme de simulation est conçue à l'aide du logiciel Matlab afin de rendre son utilisation plus conviviale pour la majorité des usagers.

3.1 Implémentation de protocole NHDP

Chaque routeur envoie à son voisinage, périodiquement ou suite à un changement éventuel, ses messages de contrôle de liens, les messages Hello. Ainsi, un moyen de contrôle et d'échange des données est bien établi dans le voisinage immédiat de chaque routeur. Après les mises à jour adoptées par NHDP, lors de la réception des Hello, une base de données valide et bien structurée est mise en place. Ainsi, une vision sur la topologie locale de chaque routeur est conçue.

3.1.1 Implémentation de message de contrôle de lien Hello

On a choisi d'implémenter le message Hello à l'aide de la fonction, **Générer le message Hello**, qui produit les différents champs et les blocs. La Figure 3.1 décrit le bloc de cette fonction.

Figure 3.1 La fonction générant le message Hello

Cette fonction fait appel à la sous-fonction, **Déterminer les blocs TLV à adresse dans le message Hello**, permettant d'inclure les informations issues de la base de données sauvegardées par NHDP dans les blocs TLV à adresse inclus dans le message Hello. La Figure 3.2 illustre le bloc de cette sous-fonction.

Figure 3.2 La sous-fonction générant les blocs TLV à adresse dans le message Hello

3.1.2 La base de données de NHDP

Afin d'accomplir son objectif, NHDP collecte et sauvegarde les données issues des messages Hello reçus et met à jour sa base de données suivant les informations reçues. NHDP organise sa base de données en trois ensembles principaux.

1. **Base d'information locale** *(Local information base)* : Cet ensemble contient les informations issues de la configuration du routeur comme la configuration de ses interfaces et de leurs adresses. La base d'information locale est composée de deux ensembles.
   - L'ensemble des interfaces locales *(Local interface set)* : Cet ensemble contient un ou plusieurs uplets qui sont en rapport avec la configuration du routeur. Les informations incluses sont indépendantes du facteur temporel. Chaque uplet est associé à une interface et est composé de deux champs comme le montre le Tableau 3.1.

Tableau 3.1 Uplet d'interface locale *(Local Interface Tuple)*

I_local_iface_-addr_list	**I_manet**
Liste des adresses de l'interface.	Variable booléen indiquant si l'interface est MANET ou non.

- Liste des adresses éliminées des interfaces *(Removed Interface Address List)* : Cet ensemble contient les adresses qui sont déjà utilisées par les interfaces du routeur. Il peut être vide dans le cas où les adresses des interfaces sont fixes. Chaque uplet est composé de deux champs comme le montre le Tableau 3.2.

Tableau 3.2 Uplet des adresses éliminées d'interface (*Removed Interface Address Tuple*).

IR_local_iface_-addr	**IR_time**
Adresse récemment utilisée par une interface.	Temps d'expiration de cet uplet.

2. **Base d'information des interfaces** *(Interface Information Bases)* : Chaque routeur maintient une base d'information pour chaque interface. Le rôle principal de cette base est de sauvegarder les liens et les voisins symétriques à deux sauts. Cette base est composé de deux ensembles.

   - L'ensemble de liens *(Link set)* : Cet ensemble joue un rôle important car il sert à sauvegarder les liens avec les autres routeurs qui sont des voisins à 1 saut. Chaque uplet correspond à un lien et est composé des champs suivants comme le montre le Tableau 3.3.

Tableau 3.3 Uplet de lien (*Link tuple*)

L_neighbor_iface_addr_list	**L_HEARD_time**	**L_SYM_time**	**L_quality**	**L_pending**	**L_lost**	**L_time**
Adresse de l'interface du voisin à 1 saut.	Temps durant lequel le lien unidirectionnel entre le voisin et le routeur est établi.	Temps durant lequel le lien entre le voisin et le routeur est considéré symétrique.	Nombre entre 0 et 1 décrivant la qualité de liaison.	Paramètre booléen montrant que le lien est un candidat qui peut être établi ou non.	Paramètre booléen montrant que le lien est perdu ou non.	Temps d'expiration de cet uplet.

- L'ensemble de voisins à 2 sauts *(2 Hop Set)* : Il sauvegarde les voisins symétriques à 2 sauts qui ont des liens symétriques avec les voisins symétriques à 1 saut. Cet ensemble peut être composé de plusieurs uplets où chacun présente les informations collectées d'un voisin symétrique à 2 sauts comme le montre le Tableau 3.4.

Tableau 3.4 Uplet de voisin à 2 sauts (*2 Hop Tuple*)

N2_neighbor_iface_addr_list	**N2_2hop_addr**	**N2_time**
Adresse de l'interface MANET du voisin symétrique à 1 saut originaire de cette information.	Adresse du voisin symétrique à 2 sauts qui a un lien symétrique avec le voisin symétrique à 1 saut.	Temps d'expiration de cet uplet.

3. **Base de données de voisins** *(Neighbor Information Base)* : Chaque routeur conserve une base de données qui sauvegarde les informations collectées sur les autres routeurs qui sont actuellement ou récemment voisins à 1 saut. Cette base est composée de deux autres ensembles :

- L'ensemble de voisins *(Neighbor Set)* : Le rôle de cet ensemble est de sauvegarder les adresses de voisins à 1 saut et l'état de lien. Chaque uplet maintient les données d'un seul voisin comme décrit dans le Tableau 3.5.

Tableau 3.5 Uplet de voisin *(Neighbor Tuple)*

N_neighbor_-addr_list	**N_symmetric**
Adresse du voisin à 1 saut.	Paramètre booléen décrivant si le voisin est symétrique ou non.

- L'ensemble de voisins perdus *(Lost Neighbor Set)* : Il sauvegarde les voisins qui ont été récemment perdus et avertis comme perdus, comme décrit dans le Tableau 3.6.

Tableau 3.6 Uplet du voisin perdu (*Lost Neighbor Tuple*)

NL_neighbor_addr	**NL_time**
Adresse du routeur qui était récemment un voisin symétrique.	Temps d'expiration de cet uplet.

3.1.3 L'organigramme de protocole NHDP

La collecte des informations du voisinage et la mise à jour de chaque base de données se font en suivant des sections bien déterminées dans la RFC 6130. L'organigramme global pour organiser ces changements est illustré à la Figure 3.3.

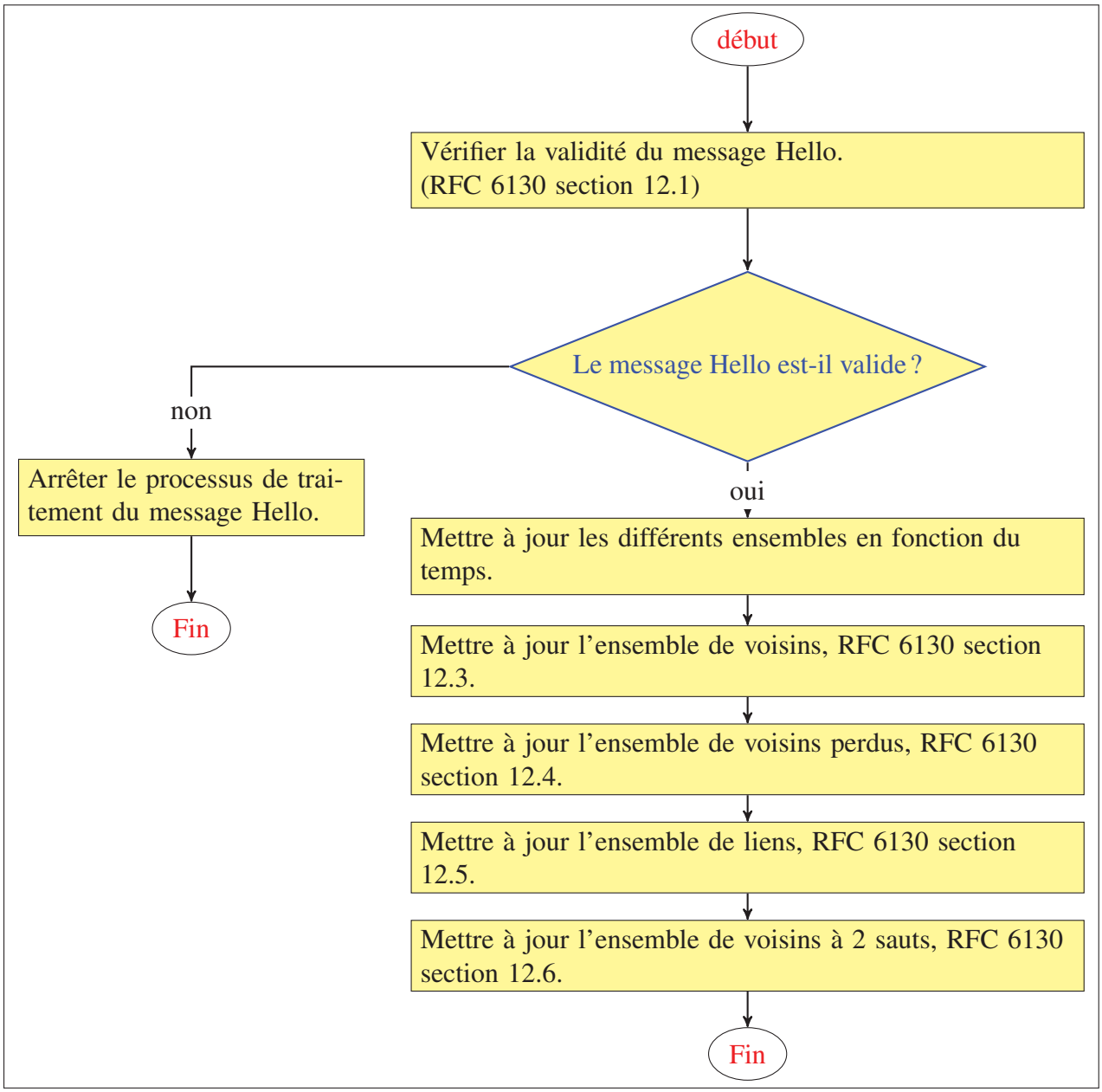

Figure 3.3 L'organigramme de protocole NHDP

À l'intérieur de chaque changement d'un ensemble, il pourra y avoir une modification des autres ensembles des autres bases de données. Dans le NHDP, l'ensemble de liens *(Link Set)* est l'ensemble capital car il capte les états de liens entre tous les noeuds du réseau. Par conséquent, chaque variation dans cet ensemble suscite une mise à jour de la majorité des autres ensembles comme l'ensemble de voisins, l'ensemble de voisins perdus et l'ensemble de voisins à 2 sauts.

La variation la plus considérable dans un uplet de l'ensemble de liens est le changement de l'état de lien vers ou à partir de l'état symétrique. Ce changement peut susciter des mises à jour, des créations et des suppressions des uplets correspondants, comme illustré à la Figure 3.4.

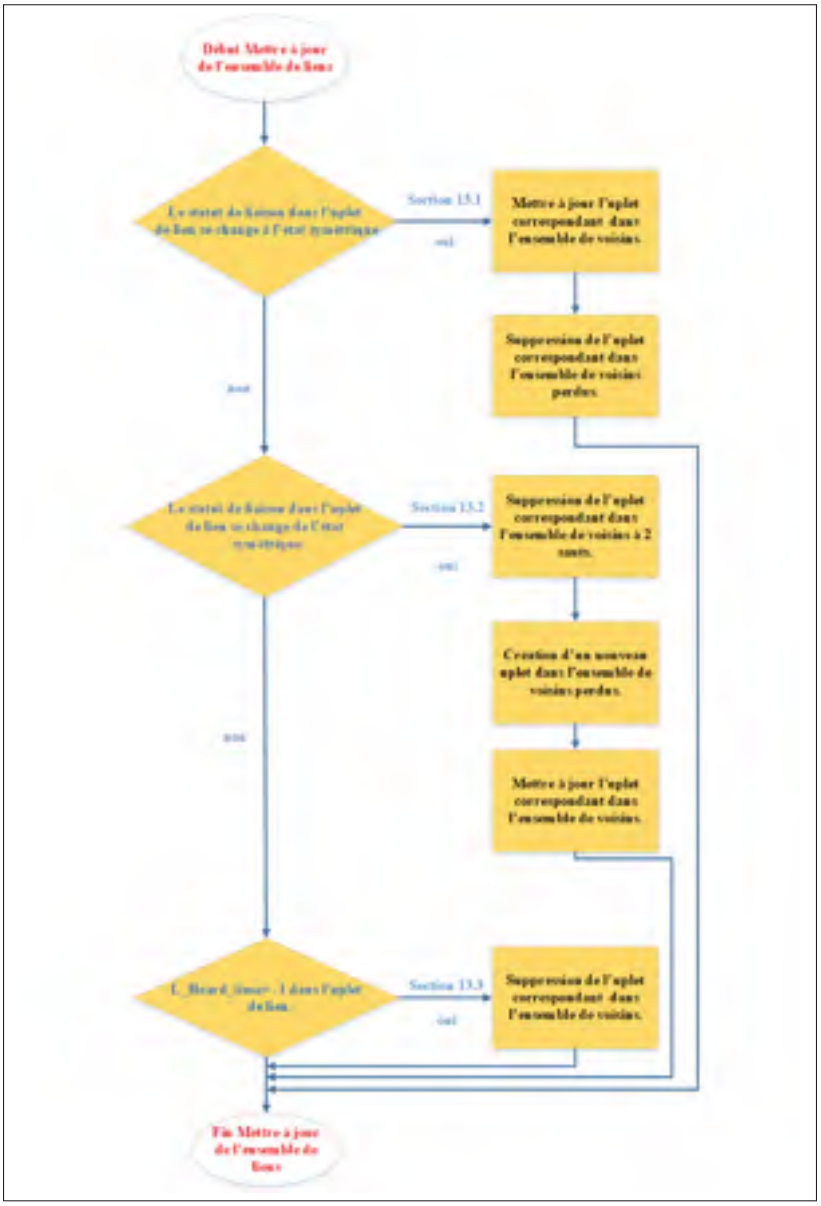

Figure 3.4 L'organigramme de mise à jour de l'ensemble de liens

3.1.4 La réalisation sous forme de code de NHDP

On a choisi d'implémenter le NHDP à l'aide de trois fonctions principales, tel qu'illustré à la Figure 3.5.

- sim_hello_list : C'est la fonction qui déclenche le traitement des messages Hello reçus. Elle contient les identifications du nœud source et du nœud destination du paquet et permet de télécharger la base de données de NHDP.
- run_hello_list : C'est la fonction qui permet d'identifier l'évènement à exécuter.
- action_hello_list : C'est la fonction qui contient tous les évènements à exécuter pour compléter ou arrêter le traitement de la mise à jour des différents ensembles de NHDP.

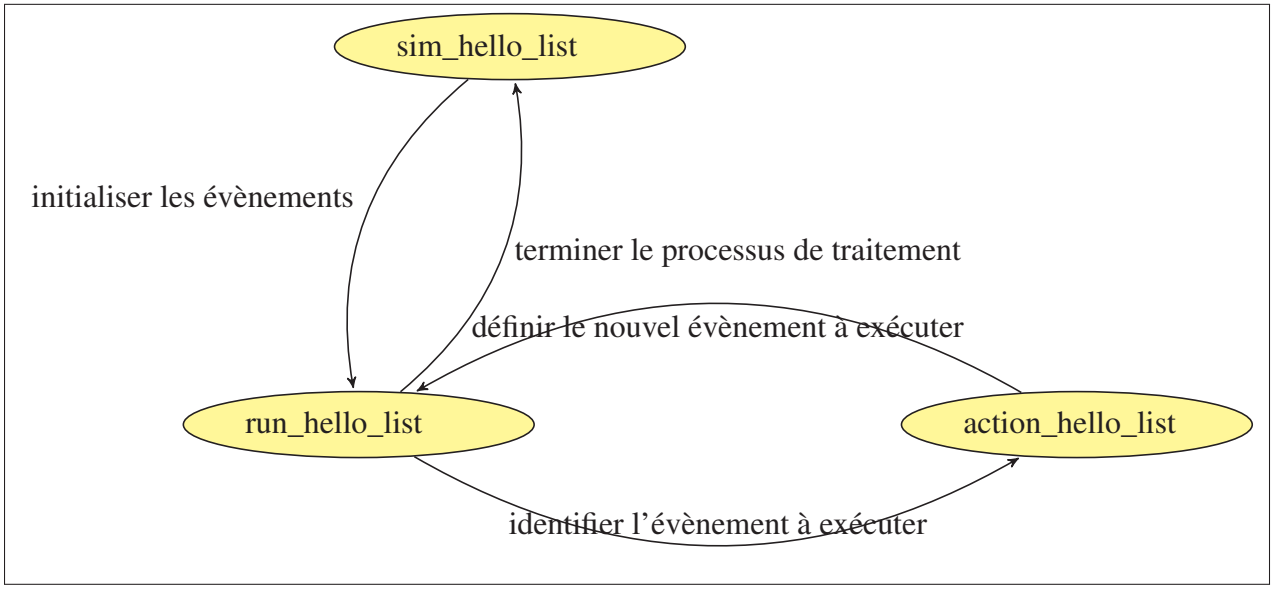

Figure 3.5 Les fonctions de NHDP

Par exemple, on peut prendre l'échange de messages Hello entre le nœud 1 et le nœud 2. le nœud 1 envoie le Hello à nœud 2. Après la vérification de la validité du message, le nœud 2 met à jour ses ensembles. Dans un premier temps, la mise à jour de l'ensemble de voisins est effectuée, tel qu'illustré à la Figure 3.7. Dans un second temps, la mise à jour de l'ensemble de liens et l'évolution de statut de lien sont réalisés, comme le montre la Figure 3.8.

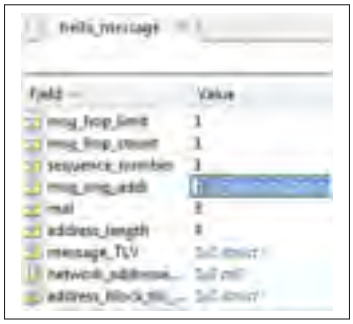

Figure 3.6 La transmission du message Hello du nœud 1 vers nœud 2

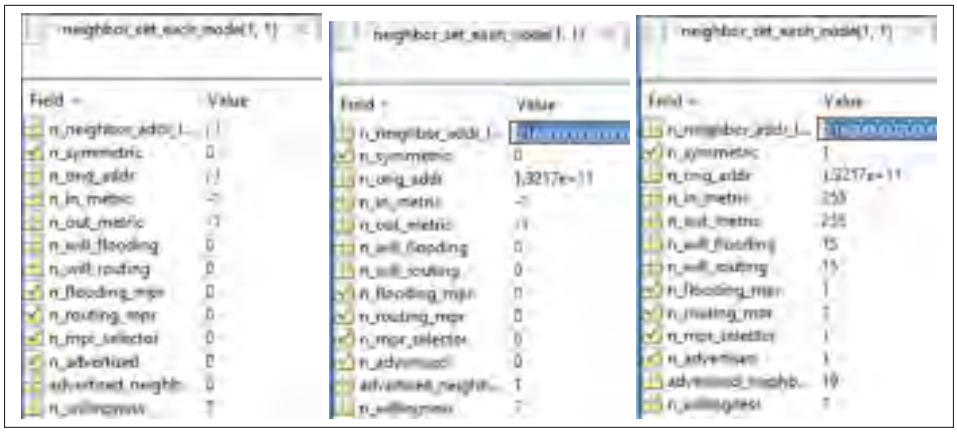

Figure 3.7 La mise à jour de l'ensemble de voisins du nœud récepteur

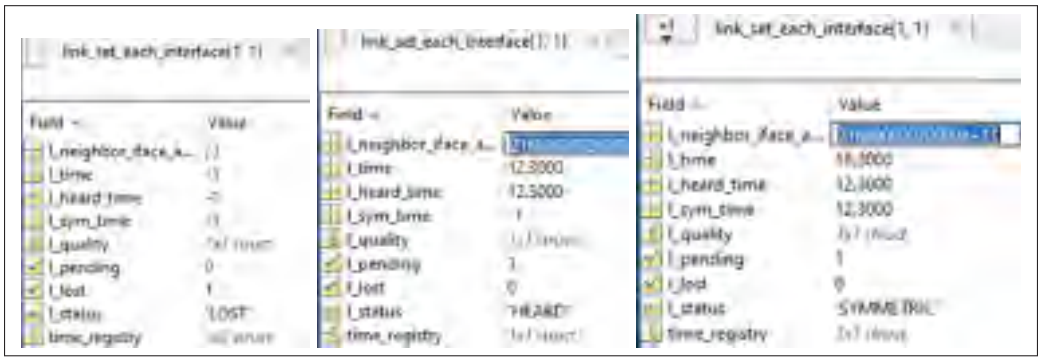

Figure 3.8 La mise à jour de l'ensemble de liens du nœud récepteur

3.2 L'implémentation de protocole OLSRV2

3.2.1 L'implémentation de message de contrôle de topologie

On a choisi d'implémenter le message de contrôle de topologie à l'aide de la fonction, **Générer le message de contrôle de topologie**, qui produit les différents champs et les blocs et les met à jour. La Figure 3.9 illustre le bloc de cette fonction. Dans notre implémentation, on a choisi une limite de sauts (*msg hop limit*) identique pour tous les nœuds du réseau et égale à 255 sauts. Ainsi le temps de validité inclus dans le message de contrôle de topologie doit encoder la valeur (t_ hold_ time) qui est un multiple de 3 de l'intervalle de topologie de contrôle, comme indiqué dans la RFC 7181.

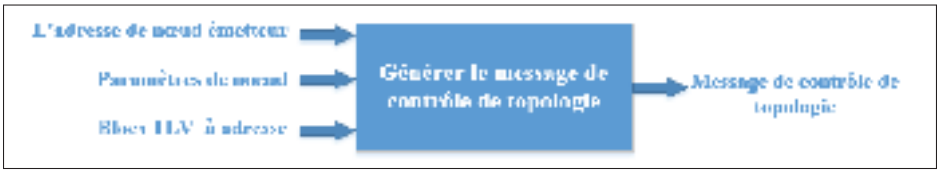

Figure 3.9 La fonction générant le message de contrôle de topologie

Cette fonction fait appel à la sous-fonction, **Déterminer les blocs TLV à adresse inclus dans le message de contrôle de topologie**, permettant d'inclure essentiellement les informations issues de l'ensemble de voisins sauvegardé par le NHDP, dans les blocs TLV à adresse inclus dans le message.

Figure 3.10 La sous-fonction générant les blocs TLV à adresse dans le message de contrôle de topologie

3.2.2 La base de données de topologie

La base de données de topologie *(Topology Information Base)* est organisée sous la forme de divers ensembles, aidant les différents nœuds d'avoir une image fiable de la topologie du réseau. Les ensembles se définissent comme suit :

- L'ensemble de routeurs distants annoncés *(Advertising Remote Router Set)* : Il sert à sauvegarder chaque routeur loin et originaire d'un message de contrôle de topologie reçu. Son but principal est de déterminer si les informations dans les messages TC reçus sont expirées. Dans ce cas, les données dans le message sont ignorées et son traitement est arrêté, comme indiqué dans la RFC 7181.

  Chaque ensemble est constitué d'un nombre d'uplets identifiant des routeurs distants annoncés. Ce nombre peut accroître ou décroître selon la mise à jour adoptée lors de la réception du message de contrôle de topologie et aussi selon le facteur temps, comme l'illustre le Tableau 3.7.

Tableau 3.7 Uplet de routeur distant annoncé
(Advertising Remote Router Tuple)

AR_orig_addr	**AR_seq_number**	**AR_time**
Adresse d'origine d'un message de contrôle de topologie reçu.	Plus grand ANSN parmi tous les ANSN dans les messages TC reçus et associés à l'adresse AR_orig_addr.	Temps d'expiration et de suppression de cet uplet.

- L'ensemble topologique de routeurs *(Router topology set)* : C'est l'ensemble qui donne au routeur une idée pertinente sur la topologie dans le réseau ad hoc entier en sauvegardant les liens collectés comme décrits dans les messages de contrôle de topologie reçus, tel qu'indiqué dans la RFC 7181 page 14. L'ensemble est composé des uplets, *(Router topology*

tuple), qui changent du nombre et du contenu selon les informations reçues. Le Tableau 3.8 définit les champs de cet uplet.

Tableau 3.8 Uplet Topologique de routeur *(Router Topology Tuple)*

TR_from_orig_-addr	**TR_to_orig_addr**	**TR_time**
Adresse d'un routeur auquel il peut être accédé à l'aide du routeur d'adresse originaire TR_to_-orig_addr en un seul saut.	Adresse d'un routeur auquel il peut être accédé par le routeur d'adresse TR_from_orig_addr en un seul saut	Temps d'expiration et de suppression de cet uplet

- L'ensemble des adresses routables de topologie *(Routable Address Topology Set)* : Il sert à conserver les données topologiques en les associant aux adresses routables dans le réseau ad hoc, signifie les adresses qui peuvent être des destinations pour un paquet IP. Aussi, il détermine quel routeur peut accéder directement à ces adresses (en un seul saut), comme indiqué dans le message TC reçu. L'ensemble est composé d'un nombre des uplets *(Routable Address Topology tuple)*. Chaque uplet est composé des champs tel qu'illustré dans le Tableau 3.9.

Tableau 3.9 Uplet topologique d'adresse routable *(Routable Address Topology Tuple)*

TA_from_orig_-addr	**TA_dest_addr**	**TA_seq_number**	**TA_time**
Adresse d'un routeur qui accède au routeur d'adresse TA_dest_addr en un seul saut.	Adresse routable d'un routeur auquel il peut être accédé par le routeur d'adresse TA_-from_orig_addr en un seul saut	Plus grand ANSN parmi tous les ANSN dans les messages TC reçus et associés à l'adresse TA_from_orig_addr	Temps d'expiration et de suppression de cet uplet.

- Le tableau de routage *(Routing Set)* : Il présente tous les trajets sélectionnés et issus du routeur désigné vers tous les routeurs dans le réseau, suivant un algorithme choisi. La RFC 7181 différencie entre le tableau IP de routage et le tableau de routage car chaque interface peut avoir une adresse fixe et des adresses variantes. Ainsi, l'adresse qui peut être enregistrée comme adresse destination ou adresse intermédiaire peut ne pas être l'adresse IP du routeur. Aussi, un routeur peut avoir plusieurs interfaces. On a choisi d'associer au chaque routeur une seule interface. En conséquence, le tableau de routage présente le tableau IP de routage. Dans l'implémentation, on a choisi de le valider après la validation de la structure topologique.

3.2.3 La base de données des messages reçus

- L'ensemble des messages reçus *(received set)* : Cet ensemble sauvegarde tous les messages reçus pour chaque interface appliquant le protocole OLSRV2. La sauvegarde signale le type du message, l'adresse originaire et le numéro de séquence. Chaque uplet est composé de champs comme le montre le Tableau 3.10.

Tableau 3.10 Uplet du message reçu *(received tuple)*

Rx_type	**Rx_orig_addr**	**Rx_seq_num**	**Rx_time**
Type du message reçu.	Adresse originaire du message	Numéro de séquence du message reçu	Temps d'expiration d'uplet

- L'ensemble des messages transmis *(forwarded set)* : Cet ensemble sauvegarde les messages transmis par le routeur. Chaque uplet est composé de champs tel qu'illustré au Tableau 3.11.

Tableau 3.11 Uplet du message transmis *(Forwarded tuple)*

F_type	**F_orig_addr**	**F_seq_num**	**F_time**
Type du message transmis.	Adresse originaire du message	Numéro de séquence du message transmis	Temps d'expiration d'uplet

Dans le code, on a opté pour l'intégration de la mise à jour de cette base de données dans l'organigramme de la structure topologique, vu que la retransmission est désignée uniquement pour le message de contrôle de topologie.

3.2.4 L'organigramme de la structure topologique

La Figure 3.11 illustre les différentes étapes de la mise à jour de la structure topologique. Après la réception du message de contrôle de topologie, une vérification de la validité du message est effectuée. Cette vérification inclut essentiellement l'adresse originaire du message, le numéro de séquence et le nombre de sauts parcourus. L'absence du temps de validité implique aussi l'arrêt de traitement du message. La mise à jour de chaque ensemble de la structure topologique est décrite dans les sections correspondantes dans la RFC 7181.

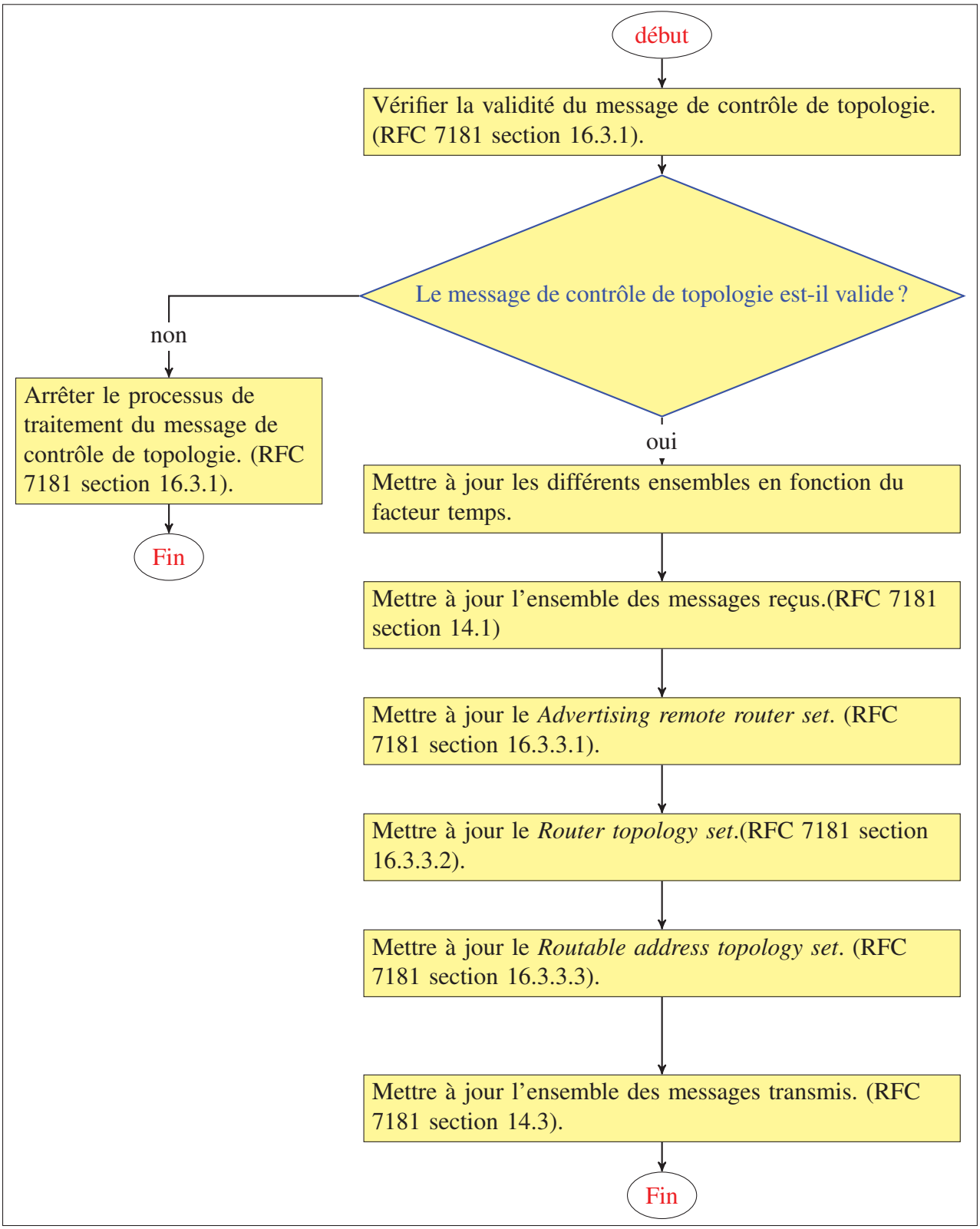

Figure 3.11 L'organigramme de la structure topologique

À l'intérieur de chaque mise à jour de chaque ensemble, on doit comparer les nouvelles informations reçues avec les données enregistrées dans l'ensemble désigné. On peut recourir à un rafraîchissement d'un ou plusieurs champs dans l'uplet ou une élimination de l'ensemble global. Cette suppression peut être accompagnée d'une élimination des autres uplets liés à l'uplet désigné, dans d'autres ensembles. La Figure 3.12 décrit la machine d'état de la mise à jour d'uplet de la structure topologique.

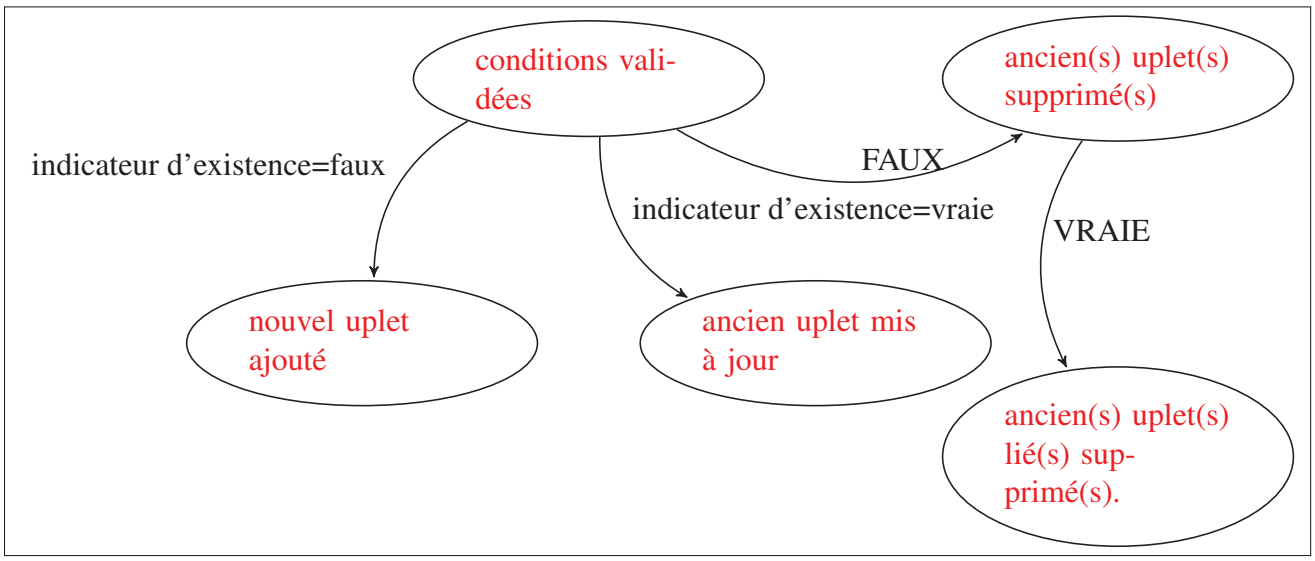

Figure 3.12 La machine d'état de la mise à jour d'un uplet de la structure topologique

3.2.5 La réalisation sous forme de code de la structure topologique

On a choisi de gérer le traitement de la structure topologique à l'aide de trois fonctions principales, tel qu'illustré à la Figure 3.13.

- sim_tc_list : C'est la fonction qui déclenche le traitement du message de contrôle de topologie reçu. Elle contient les identifications de nœud source et de nœud destination. Ceci permet de télécharger la base de données de NHDP et d'OLSRV2 de nœud correspondant.
- run_tc_list : C'est la fonction qui permet d'identifier l'évènement à exécuter.

- action_tc_list : C'est la fonction qui contient tous les évènements à exécuter pour compléter ou arrêter le traitement de la mise à jour des différents ensembles de la structure topologique.

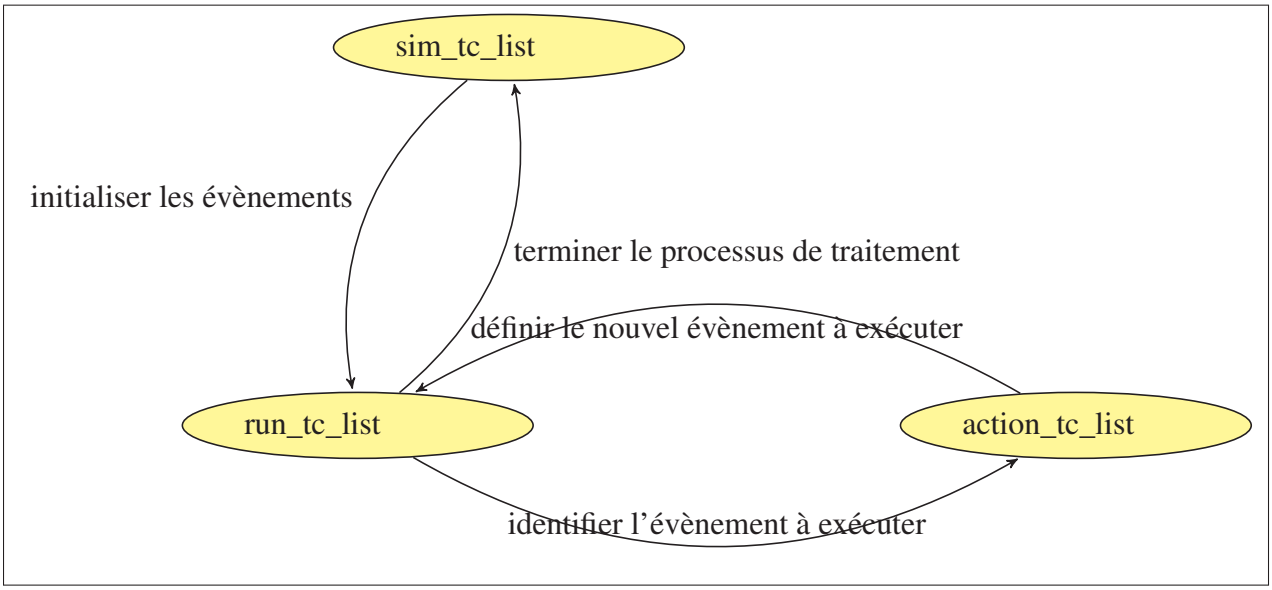

Figure 3.13 Les fonctions de la structure topologique

3.2.5.1 Le graphe de topologie

En recevant les messages de contrôle de topologie, chaque routeur dans le réseau peut concevoir sa vision sur la topologie du réseau entier. En fait, on peut présenter cette vue sous forme d'un graphe orienté. Dans (Sigward, 2002), un graphe orienté G(S,A) est défini par :

- Un ensemble S fini de sommets (vertices, un sommet).
- Un ensemble A d'arcs *(edges)*. Chaque arc a une origine (source) et une extrémité finale *(target)* dans S.

Principalement dans OLSRV2, on a trois types d'arêtes permettant de concevoir pas à pas le graphe de topologie pour chaque routeur.
Le premier type d'arête présente les arcs voisins par rapport à chaque routeur. Ces arcs présentent les liens symétriques enregistrés et encore valides. La Figure 3.14 illustre ce type d'arête.

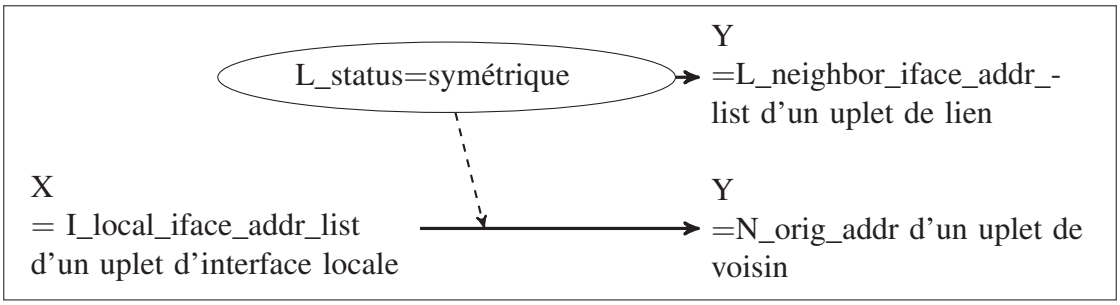

Figure 3.14 Le premier type d'arête dans le graphe de topologie

Le deuxième type d'arête aide principalement à illustrer les arcs non voisins à l'aide de données enregistrées et valides dans l'ensemble topologique de routeurs *(router topology set)*. Ainsi, un graphe étendu peut être réalisé. La Figure 3.15 décrit ce type d'arête.

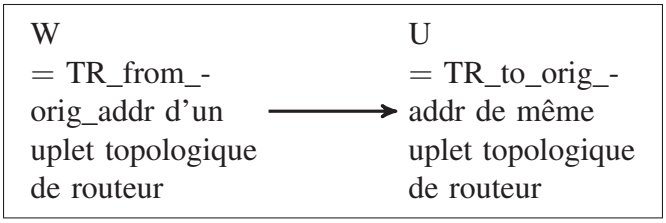

Figure 3.15 Le deuxième type d'arête dans le graphe de topologie

Le troisième type d'arête met en place les données enregistrées dans l'ensemble des adresses routables de topologie. La Figure 3.16 décrit l'établissement de ce type d'arête.

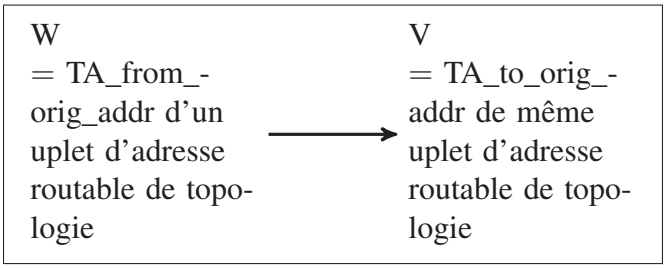

Figure 3.16 Le troisième type d'arête dans le graphe de topologie

On prend, par exemple, le nœud 1 qui a réussi à construire son graphe orienté dans un réseau de 8 nœuds déplacés aléatoirement. La Figure 3.17 illustre cette vision du réseau de la part de nœud 1.

Figure 3.17 Le graphe de topologie de nœud1

Comme on peut voir dans la Figure 3.17, le nœud 1 n'a pas de lien direct avec le nœud 5. Après les traitements de messages de contrôle de topologie, le nœud 1 peut avoir des informations sauvegardées dans son ensemble topologique de routeurs *(router topology set)* concernant les liens valides du nœud 5 avec ses voisins. La Figure 3.18 présente l'uplet contenant le lien valide issu de nœud 5 vers le nœud 8.

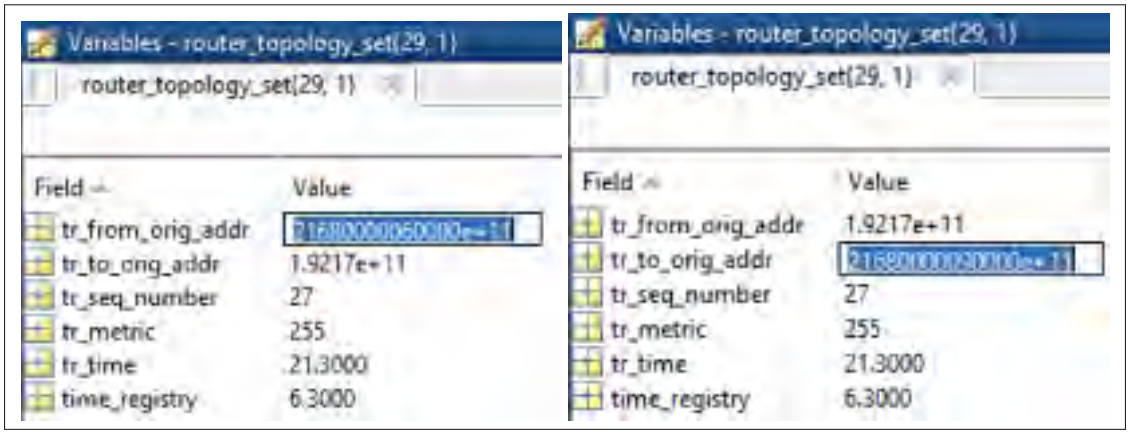

Figure 3.18 Un exemple d'uplet sauvegardé dan l'ensemble *router topology set* du nœud1.

3.2.6 Le tableau de routage

Vu la nature proactive du protocole OLSRV2, le tableau de routage contient tous les trajets issus du nœud considéré vers tous les autres noeuds dans le réseau. Ce tableau de routage ne subit pas l'influence du facteur temps et il est disponible à tout moment. La validation de son contenu dépend essentiellement des changements du voisinage et de topologie dans le réseau. Chaque changement dans n'importe quel ensemble fondamental (l'ensemble de lien, l'ensemble de voisins, l'ensemble topologique de routeurs, l'ensemble des adresses routables de topologie) engendre une mise à jour du tableau. Cette actualisation n'exige pas une transmission des messages, mais un nouveau calcul des trajets suivant l'algorithme choisi sera nécessaire. Selon OLSRV2, la sélection basique du trajet est celle du trajet le plus court en terme du nombre de sauts. Chaque uplet de routage enregistre le premier saut le long du trajet pour chaque destination. Le Tableau 3.12 décrit les différents champs dans l'uplet de routage.

Tableau 3.12 Uplet de routage *(Routing Tuple)*

R_dest_addr	**R_next_iface_addr**	**R_local_iface_addr**	**R_dist**
Adresse de la destination.	Adresse du prochain saut dans le trajet sélectionné vers la destination	Adresse de l'interface par laquelle on doit transmettre le paquet IP	Nombre de sauts dans le trajet sélectionné jusqu'à la destination.

3.2.6.1 L'algorithme de routage

L'algorithme de routage choisi garantissant le trajet le plus court en terme de nombre de sauts est l'algorithme de Dijkstra. OLSRV2 donne la possibilité de choisir un autre algorithme et propose une version d'algorithme de Dijkstra. En effet, on peut trouver plusieurs versions de cet algorithme adaptées aux particularités des projets. Dans ce projet, on a choisi d'avoir recours à l'algorithme de Dijkstra cité dans la RFC 7181 vu son harmonie avec la structure topologique et la structure de données du protocole NHDP. La version proposée présente plusieurs étapes afin d'accomplir le remplissage du tableau de routage. Dans le code utilisé, la fonction qui englobe toutes les phases et qui présente le tableau de routage final est nommée, **Déterminer le tableau de routage pour chaque nœud**. Elle prend comme données l'adresse du nœud et le tableau de routage initialisé ou sa dernière version, tel qu'illustré à la Figure 3.19.

Figure 3.19 La fonction générant le tableau de routage

1. La première étape de l'algorithme de Dijkstra :

   Selon RFC 7181 section C, on doit en premier lieu remplir le tableau de routage par les voisins symétriques. En parcourant l'ensemble de voisins, chaque uplet du voisin avec un lien symétrique valide engendre la création d'un nouveau uplet de routage dont le remplissage de ses champs s'établit de la manière suivante, comme le montre la Figure 3.20. Cette étape est implémentée à l'aide de la sous-fonction, **Ajouter les nœuds voisins**, qui prend comme entrées l'ensemble de voisins déjà préparé par le NHDP et qui crée le premier niveau de tableau de routage. La Figure 3.21 définit cette sous-fonction

   .

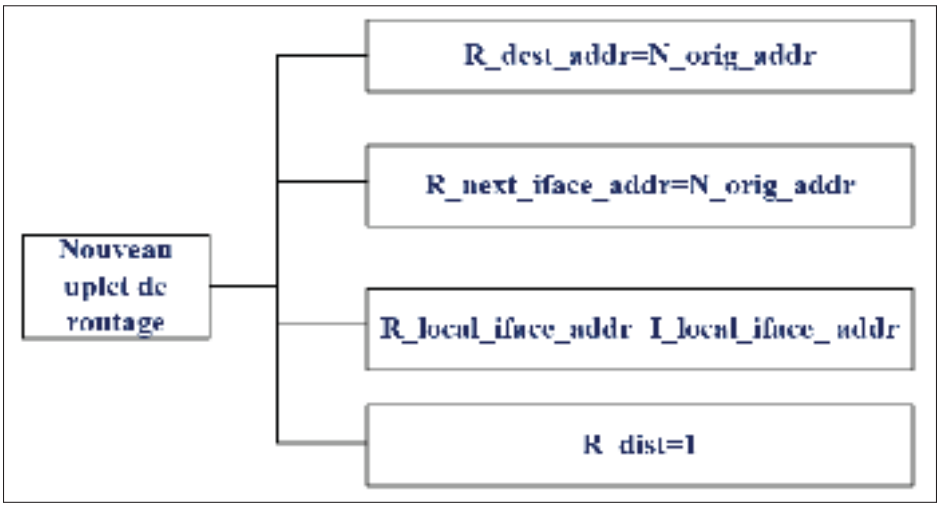

Figure 3.20 La première étape de l'algorithme de Dijkstra

Figure 3.21 La sous-fonction présentant la première étape de l'algorithme de Dijkstra

2. La deuxième étape d'algorithme de Dijkstra :

   Cette étape consiste à ajouter les routeurs distants (plus qu'un saut) en suivant l'algorithme présenté dans la section C3 de la RFC 7181. Elle sert à déclencher une itération qui vise à ajouter et à remplir les uplets de routage en se basant sur les données valides dans l'ensemble topologique de routeurs *(router topology set)*. Le critère du choix de l'uplet de routage à mettre à jour ou à ajouter est la distance minimale en terme du nombre de sauts. La Figure 3.22 illustre l'organigramme de cette étape.

   La sous-fonction permettant d'accomplir cette phase est nommée, **Ajouter les noeuds distants**. Elle prend comme entrée l'ensemble topologique de routeurs qui permet de construire le reste de tableau de routage (trajets de plus qu'un saut). La Figure 3.23 illustre le bloc de cette sous-fonction.

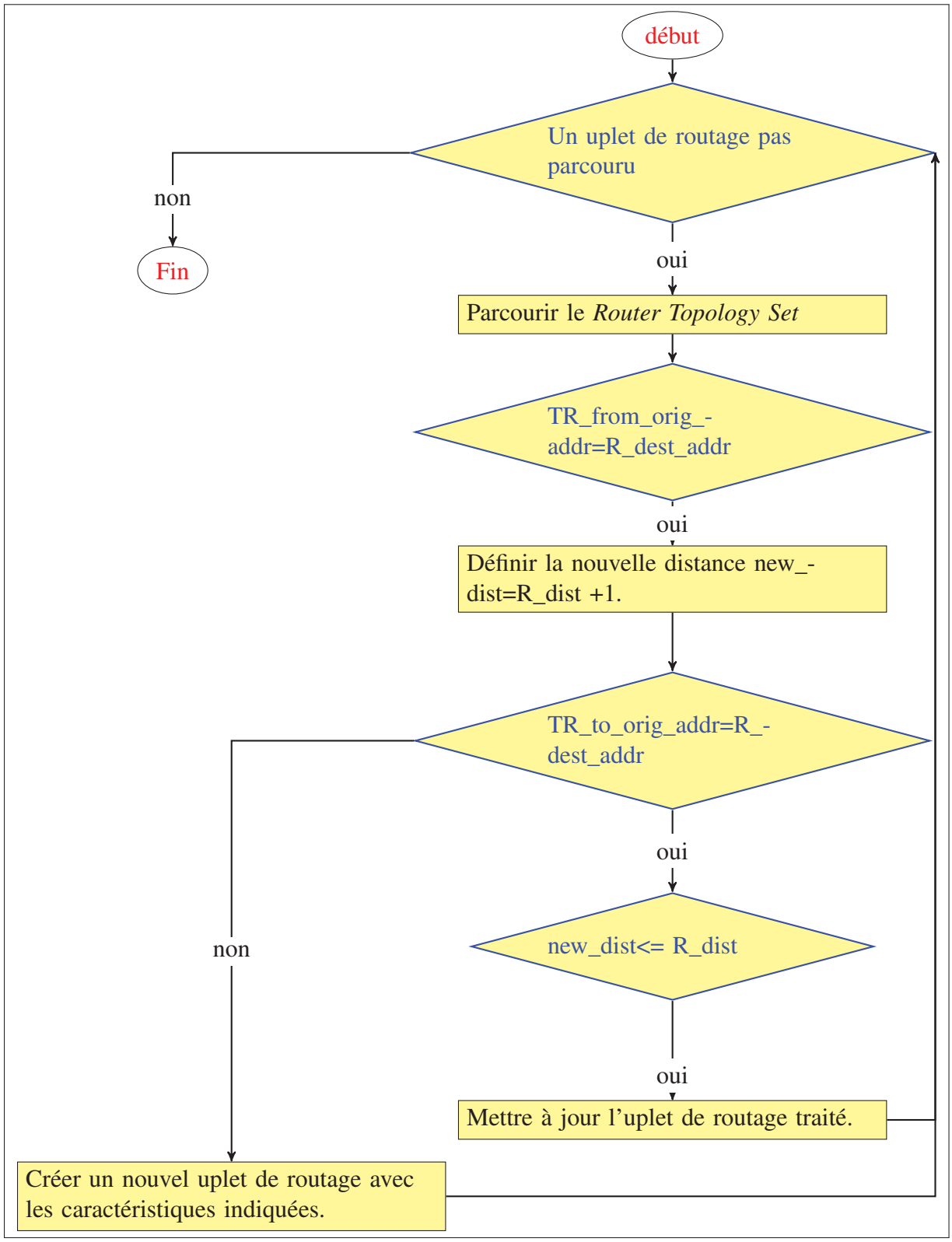

Figure 3.22 L'organigramme de deuxième étape d'algorithme de Dijkstra

Figure 3.23 La sous-fonction présentant la deuxième étape de l'algorithme de Dijkstra

Afin de concrétiser la description d'algorithme de routage codé, on prend l'exemple d'un réseau ad hoc de 8 noeuds où tous les liens directs sont établis. La Figure 3.24 illustre la topologie du scénario traité.

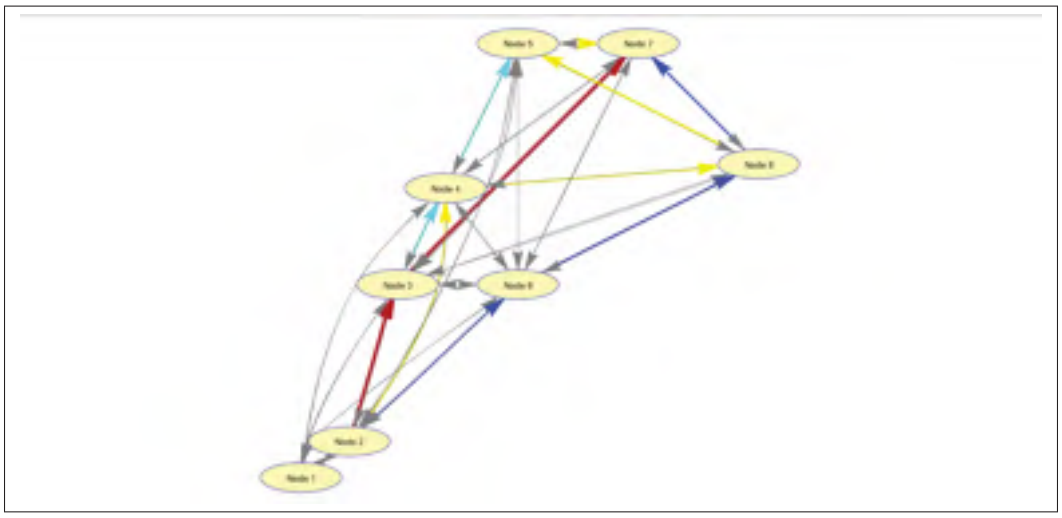

Figure 3.24 La topologie du scénario traité

On prend comme source le nœud 2 et comme destination le nœud 7. La Figure 3.24 montre plusieurs trajets disponibles. On peut citer l'exemple du trajet passant par le nœud 4 et le noeuds 5. L'algorithme implémenté a réussi à choisir le trajet le plus court possible qui est le trajet passant par le nœud 3. Ainsi, la longueur de la route construite est de longueur 2 sauts qui est le nombre minimal à cause que la source n'a pas de lien direct avec la destination. La Figure 3.25 montre l'enregistrement du trajet choisi dans le tableau de routage du nœud 2.

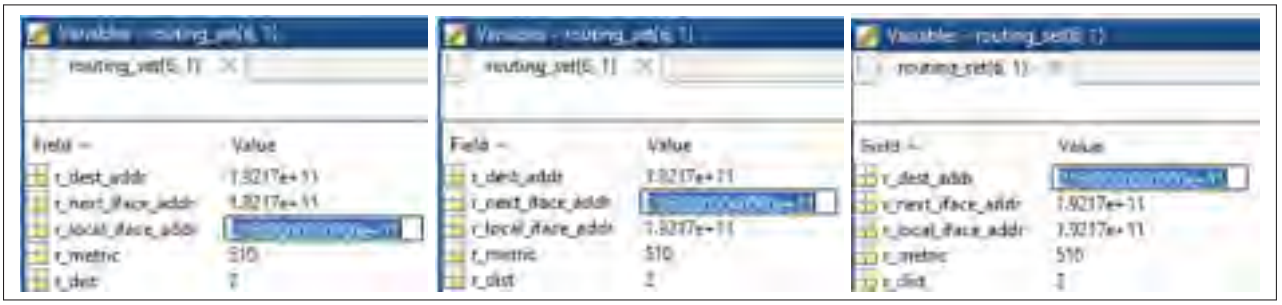

Figure 3.25 Un exemple du trajet le plus court

La structure proposée par le NHDP englobe la configuration locale (2 sauts) pour chaque nœud. Chaque uplet du voisin sauvegarde les données valides d'un noeud voisin. Cet uplet du voisin est associé à un uplet du lien où le statut temporel du lien est bien précis. On a la possibilité d'avoir l'information précise sur le temps durant lequel le lien est considéré symétrique ou unidirectionnel. Ainsi, une vision sur la topologie locale peut être conçue. La structure topologique détaillée par la RFC 7181, vise à étendre cette vision en rassemblant les données concernant les liens valides entre les noeuds lointains. Une base de données topologiques sauvegarde les liens pouvant être établis pour atteindre des destinations lointaines. En rassemblant ces données, on peut avoir recours à l'algorithme de Dijkstra afin de sélectionner le trajet le plus court en terme du nombre de sauts.

CHAPITRE 4

ÉVALUATION DE PROTOCOLE OLSRV2

Dans ce chapitre, on vise à évaluer le protocole OLSRV2 en analysant les performances résultantes après son intégration dans le simulateur.

4.1 Environnement de simulation

L'architecture de l'environnement de simulation et sa capacité d'intégrer les configurations demandées présentent des facteurs qui influencent nécessairement la fiabilité des résultats de simulation obtenus. De ce fait, plusieurs travaux et améliorations ont été réalisés afin d'obtenir des simulateurs capables d'évaluer les performances du réseau et d'identifier les nœuds et les autres paramètres du réseau.

4.1.1 Les environnements de simulation existants

Il existe un bon nombre de simulateurs disponibles qui diffèrent dans leurs architectures, le temps d'exécution et la langage d'implémentation, comme ns_2, ns_3, OMNeT++, JiST et SimPy.

- **ns_2 :** Simulateur à évènements discrets. En raison du grand nombre de protocoles et de générateurs de trafic disponibles gratuitement pour ns_2, il est devenu un standard de simulation du réseau. ns_2 est composé d'un code C++ utilisé pour modéliser le comportement de nœuds dans le réseau et un script oTcl qui permet de contrôler les modifications et la topologie dans le réseau. Cette conception aide à gagner du temps de recompilation dans le cas de changements dans la configuration. Le défaut de ns_2 est sa mémoire limitée et son temps d'exécution long.
- **ns_3 :** Successeur de ns_2. Il est composé d'un code C++ pour l'implémentation de modèles de simulation. ns_3 n'utilise pas le script oTcl pour le contrôle des changements dans le réseau. Ceci évite le problème de combinaison du code C++ et oTcl dans ns_2. La si-

mulation peut être ainsi implémentée en C++. ns_3 englobe un code de simulateur GTNetS permettant de souligner le facteur de mise à l'échelle du réseau. ns_3 peut supporter un code d'implémentation à temps réel au moyen des API standards comme *Berkley sockets* et *POSIX threads*.

- **OMNeT++ :** Il n'est pas un simulateur désigné pour le réseau, mais il est un simulateur à évènements discrets avec un progiciel INET contenant une collection de modèles et de protocoles d'internet. Il a aussi recours au progiciel *OMNeT++ (mobility framewok)* désigné pour la configuration de la mobilité dans le réseau ad hoc. La simulation prend place à l'aide de *network description language* (NED) qui permet une configuration dynamique des paramètres du réseau.
- **JiST (*Java in simulation Time*) :** Il utilise le langage Java. La plupart de temps, il est utilisé avec SWANS qui est un simulateur du réseau ad hoc mobile implémenté architecturalement au-dessus de JiST. Il est composé des entités représentant chacune un paramètre du réseau. Le temps de simulation est uniquement piloté par l'interaction entre ces entités. Ceci détermine la synchronisation de simulation et facilite l'exécution parallèle du code.
- **SimPy :** Il est implémenté à l'aide de Python. Il est composé de processus exécutables en parallèle, qui peuvent échanger des objets Python. Les tâches de synchronisation et de contrôle de données sont dirigées par des instructions bien déterminées dans SimPy.

Le nombre de simulateurs disponibles nous mène à poser la question du choix d'un simulateur. L'utilisateur du simulateur réseau cherche généralement un temps d'exécution court et un usage optimal de mémoire. Afin d'évaluer les compétences de chaque simulateur selon ces deux critères, Weingatner et al. (2009) choisissent d'implémenter un scénario simple basé sur une topologie carrée où chaque nœud génère un paquet chaque seconde et le diffuse à son voisinage. Après une seconde, les nœuds voisins reçoivent les messages. Le délai de propagation est contrôlé par les délais d'exécution des évènements dans le simulateur concerné et les paquets sont transmis au canal avec une probabilité identique pour chaque liaison. Les résultats de comparaison mènent à conclure que SimPy n'a pas un temps d'exécution faisable pour un réseau étendu. Par exemple, dans le cas d'un réseau de 3025 nœuds, SimPy prend 1225 seconde pour accomplir la simulation. Pour le même scénario, JiST termine la simulation

dans une durée égale à 1/14 de l'intervalle enregistré par SimPy. Avec cette performance, JiST dépasse ns_3 et OMNeT++ grâce à l'architecture basée sur l'exécution parallèle et l'optimisation du temps d'exécution. L'amélioration architecturale dans ns_3, lui permet aussi d'avoir un temps d'exécution satisfaisant. Concernant la deuxième voie de la comparaison, JiST est le premier dans l'usage de mémoire à cause de son mécanisme de récupération. Plus le réseau s'étend, plus la différence entre JiST et les autres simulateurs augmente. Les usages de mémoire de ns_3, d'OMNeT++ et de SimPy augmentent linéairement en fonction du nombre de nœuds. En conclusion, ns_3 montre la meilleure performance globale qui touche les deux voies (temps d'exécution et usage de mémoire). OMNeT++ peut être considéré comme l'alternative appropriée de ns_3.

4.1.2 Le simulateur fourni

Le simulateur fourni est un simulateur à évènements discrets pour le réseau WIFI. Sa philosophie est proche de celle des simulateurs mentionnés. Il est codé à l'aide du logiciel Matlab. Ceci rend son exploitation plus facile pour la majorité des utilisateurs. Son objectif est de simuler le modèle OSI pour le réseau WFI. Le fichier qui déclenche la simulation est *sim1scratch*. Ensuite, on a l'appel du fichier *sim1_test1*. Dans ce dernier fichier, il y a une initialisation et une définition de plusieurs paramètres du réseau comme le trafic bidirectionnel, le nombre de nœuds et la quantité du trafic. Puis, on a la fonction *run* qui définit l'évènement à exécuter en choisissant celui avec l'indicateur du temps minimal. Le fichier *action* englobe la structure qui organise l'implémentation de différentes couches du modèle OSI. Donc on a un ensemble d'événements contrôlés par la structure *switch*. Dans chaque accès au fichier *action*, on a une exécution de l'événement choisi par le fichier *run*. Des fichiers (*trace file*) sont générés et permettent une sauvegarde des évènements pour chaque source et chaque destination. La Figure 4.1 décrit l'architecture du simulateur fourni.

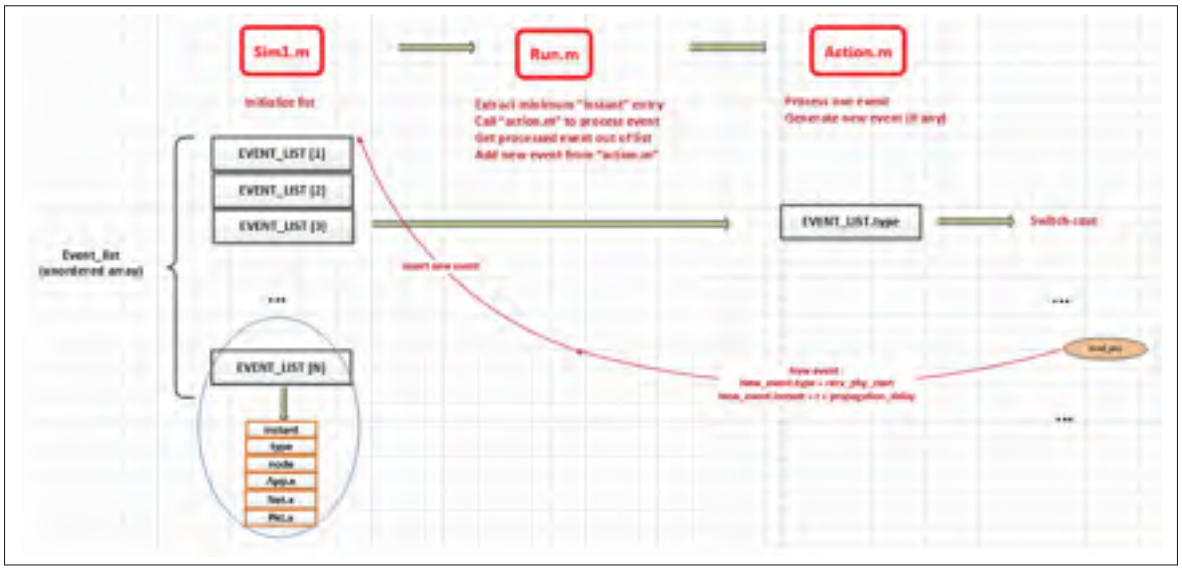

Figure 4.1 L'architecture du simulateur Matlab.
Tirée de (Groleau, 2012)

4.2 Les paramètres d'évaluation

Vu que le rôle principal de chaque protocole de routage est la livraison du plus grand nombre de paquets possible dans des délais minimums et en gardant un débit moyen élevé, on choisit d'évaluer OLSRV2 en soulignant ces trois paramètres.

- **Taux de paquets livrés :**

  La capacité de livrer les paquets aux destinations, en suivant le nombre de sauts minimal ou autre métrique, est le rôle primordial de chaque protocole de routage. Le paramètre de taux de paquets livrés permet d'expliciter cette capacité. Le taux de paquets livrés est défini comme pourcentage des paquets livrés à leurs destinations par rapport aux paquets transmis.

- **Délai de bout en bout :**

  C'est la durée du temps nécessaire pour qu'un paquet atteigne sa destination après sa transmission de la source. C'est la transmission de bout en bout qui compte. Ce délai peut être influencé par le délai de traitement et le délai d'envoi au canal. Un délai de bout en bout court est considéré comme une bonne performance du protocole de routage.

- **Débit moyen :**

  Le débit moyen (*Average throughput*) est défini comme le rapport des bits reçus sur le temps total séparant l'émission et la transmission des paquets.

4.3 Les paramètres de simulation

On a choisi d'évaluer les performances d'OLSRV2 dans le réseau WIFI avec l'interface de type IEEE 802.11n, comme le montre le Tableau 4.1. La fonction DCF est incluse pour organiser l'accès au canal. Une distribution aléatoire de nœuds est choisie pour une évaluation concrète du protocole. La charge utile est identique pour tous les noeuds.

Tableau 4.1 Paramètres de simulation

Type de réseau	**IEEE 802.11n**
DCF	inséré
Modèle de propagation	FRIIS
Nombre de nœuds	maximum de 20 nœuds
Distribution de nœuds	Aléatoire, pas de mobilité
La charge utile (Payload)	800 bits
Trafic	Continu et identique pour tous les nœuds
Nombre d'exécutions pour chaque simulation	100

4.4 Les résultats de simulation

Après l'intégration du protocole OLSRV2 dans le simulateur fourni, les scénarios de simulation consistent à évaluer les performances du réseau en fonction du nombre de nœuds et du nombre de trafics à acheminer. L'augmentation du nombre de nœuds donne une idée de la capacité du protocole à garder un facteur d'échelle satisfaisant et à montrer un maintien dans le bon

fonctionnement de ses mécanismes. La variation du nombre de trafics à acheminer dans le réseau provoque plus de congestion. Ceci mène à une évaluation concrète de la capacité du protocole pour supporter un trafic intense.

Le simulateur fourni est installé sur l'ordinateur de laboratoire ayant un processeur de fréquence égale à 3 GHz et une mémoire vive égale à 8 GB. En augmentant le nombre de nœuds, le temps moyen d'exécution augmente, comme le montre le Tableau 4.2. Pour le scénario avec un nombre de nœuds égale à 20, on a le temps d'exécution le plus élevé avec une valeur égale à 0,689567368 seconde.

Tableau 4.2 Variation du temps d'exécution en fonction du nombre de nœuds

5 noeuds	0,047389 (s)
10 noeuds	0,188946273 (s)
15 noeuds	0,527507238 (s)
20 noeuds	0,689567368 (s)

L'augmentation du nombre de paquets transmis a engendré aussi une augmentation du temps d'exécution. Le Tableau 4.3 illustre cette variation.

Tableau 4.3 Variation du temps d'exécution en fonction du nombre de paquets

100 paquets	0,3327586 (s)
200 paquets	0,689567368 (s)
300 paquets	1,05 (s)
400 paquets	1,26853536 (s)

L'utilisation de la mémoire physique est pratiquement stable en fonction du nombre de nœuds. L'utilisation est de l'ordre de 3 GB. Ceci est équivalent à un taux d'utilisation égal à 0,4. En augmentant le nombre de paquets transmis, le cas de 400 paquets transmis a l'utilisation de mémoire la plus élevée avec un taux d'utilisation égal à 0,42. Le Tableau 4.4 illustre cette stabilité de l'utilisation de mémoire physique en fonction du nombre de nœuds. Le Tableau 4.5 montre le taux d'utilisation de mémoire en fonction du nombre de paquets transmis.

Tableau 4.4 Taux moyen d'utilisation de mémoire physique en fonction du nombre de nœuds

5 noeuds	0,4
10 noeuds	0,4
15 noeuds	0,4
20 noeuds	0,4

Tableau 4.5 Taux moyen d'utilisation de mémoire physique en fonction du nombre de paquets

100 paquets	0,4
200 paquets	0,4
300 paquets	0,41
400 paquets	0,42

Débit en fonction du nombre de nœuds

En fixant le nombre de trafics à 10 et en augmentant le nombre de nœuds de 5 jusqu'à 20, nous obtenons la variation de débit telle qu'illustrée à la Figure 4.2. Lorsque nous augmentons le nombre de nœuds et par conséquent le nombre de trafics total circulant dans le réseau, le débit augmente progressivement jusqu'à une valeur de 2713,528 Kbps pour un nombre de nœuds égal à 15. Pour un nombre de nœuds égal à 20, le débit a un abaissement jusqu'à la valeur 2347,669 Kbps. Pour ce scénario, le nombre de nœuds recevant les paquets est égal à 15, avec un nœud recevant un nombre de paquets double (20 paquets). Quatre nœuds n'ont reçu aucun paquet. Par exemple, le nœud 10 a échoué à transmettre tous les paquets. A cause de collisions et d'interférences, le nœud 10 a eu un nombre de retransmissions de chaque paquet à transmettre. Chaque paquet peut être retransmis 7 fois. Ce nombre est défini comme le nombre limite de retransmissions. Après cette limite, le nœud décide de laisser tomber le paquet.

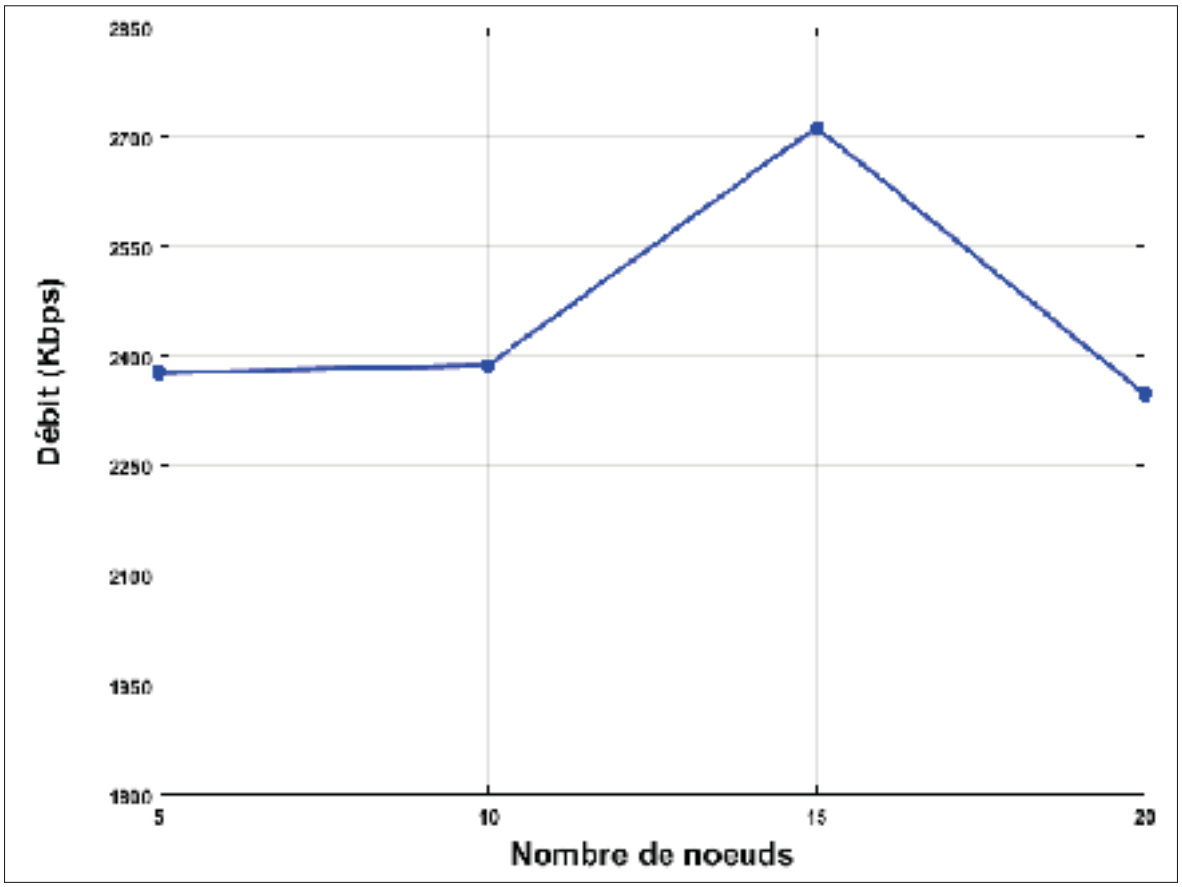

Figure 4.2 Le débit en fonction du nombre de nœuds .

Délai de bout en bout en fonction du nombre de nœuds

Le délai de bout en bout augmente progressivement avec le nombre de noeuds, comme le montre la Figure 4.3. Les valeurs enregistrées n'ont pas dépassé le 0,0035 secondes. Le scénario avec 15 noeuds possède le délai le plus élevé, mais on a une livraison de tous les paquets transmis. Dans ce scénario, il y a eu un nombre de retransmissions pour certains nœuds à cause de collisions et d'interférences. Le nombre maximal de retransmissions sauvegardées est 5 retransmissions pour le cinquième paquet au cinquième nœud. Le délai de bout en bout pour le scénario de 20 nœuds est diminué par rapport au scénario de 15 nœuds. Dans le scénario de 20 nœuds, certains nœuds ont décidé de laisser tomber les paquets. Ceci explique la diminution générale de délai de bout en bout. D'une façon générale, les délais de bout en bout enregistrés confirme que l'approche proactive garantit les délais minimums possibles.

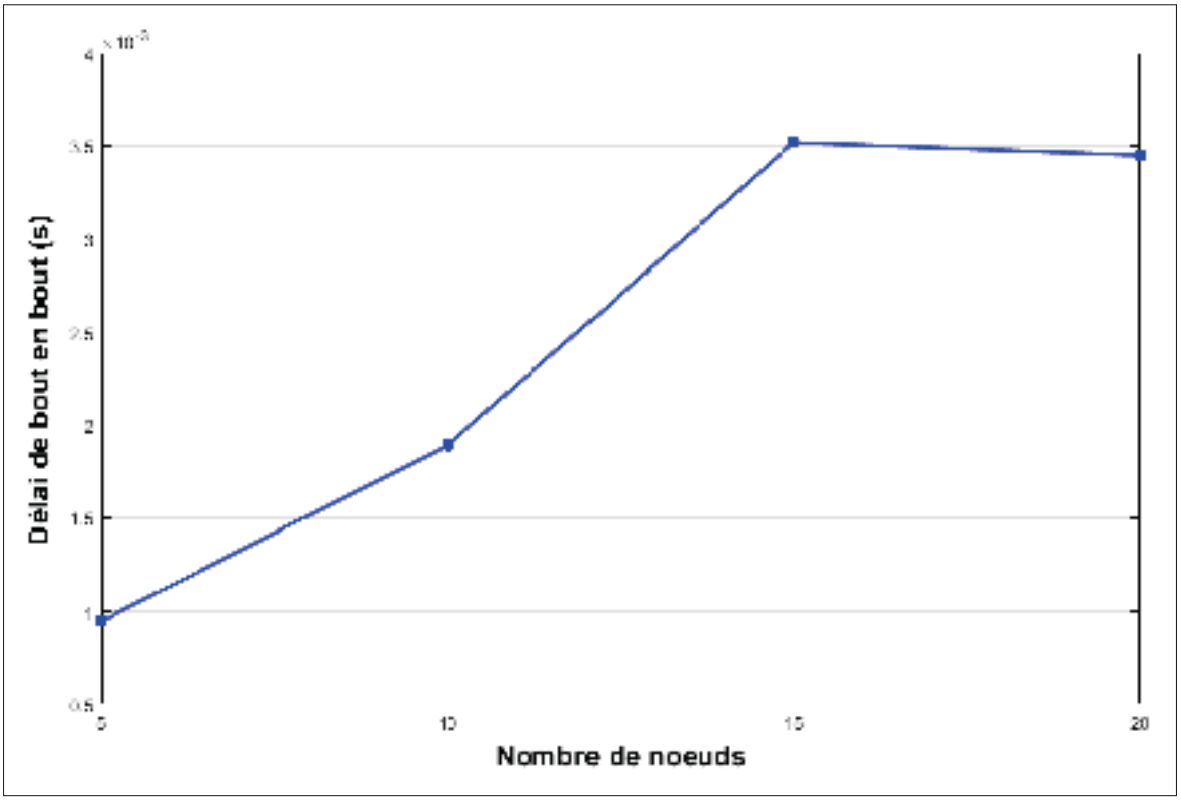

Figure 4.3 Le délai de bout en bout en fonction du nombre de nœuds

Taux de paquets livrés en fonction du nombre de nœuds

Pour les différents nombres de nœuds, le taux de paquets livrés a gardé un taux satisfaisant avec des valeurs supérieures ou égales à 0,8, comme le montre la Figure 4.4. Pour un nombre de nœuds égal à 5 et à 15, le taux de paquets livrés atteint sa valeur maximale. A cause du nombre de paquets laissés tomber, le scénario de 20 nœuds a enregistré un taux de paquets livrés égale à 0,8 qui est une valeur satisfaisante malgré la perte de paquets. Pour le nombre de nœuds égale à 10, on a deux nœuds qui ont échoué à transmettre leurs paquets. Le nœud 9 a échoué à transmettre tous les paquets au nœud 1 après 7 retransmissions échouées pour chaque paquet. Le nœud 3 a échoué à transmettre tous ses paquets au nœud 5. Les résultats enregistrés montrent l'efficacité des mécanismes déployés par OLSRV2 à détecter les liens fiables et, par conséquent, les trajets optimums afin de garantir un taux de paquets délivrés satisfaisant.

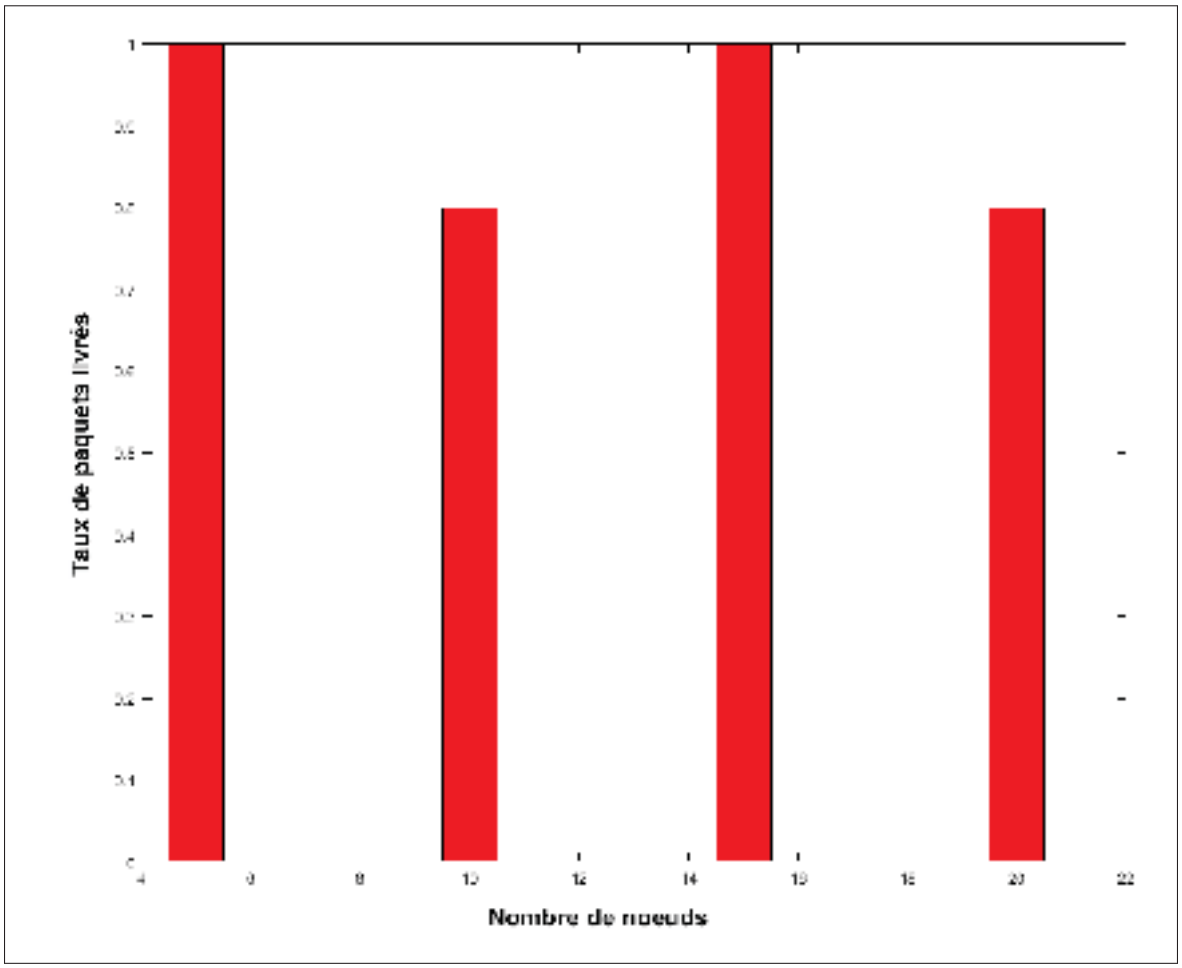

Figure 4.4 Taux de paquets livrés en fonction du nombre de nœuds

Débit en fonction du nombre de paquets

La capacité de protocole de routage d'acheminer un trafic important dans le réseau est une qualité demandée. La Figure 4.5 montre la variation de débit en fonction de nombre de paquets. Le nombre de paquets transmis est entre 100 paquets et 400 paquets. OLSRV2 a réussi à garder une augmentation progressive de débit en fonction du nombre de paquets. La valeur maximale enregistrée est de l'ordre de 2577,02665 Kbps pour 400 paquets transmis.

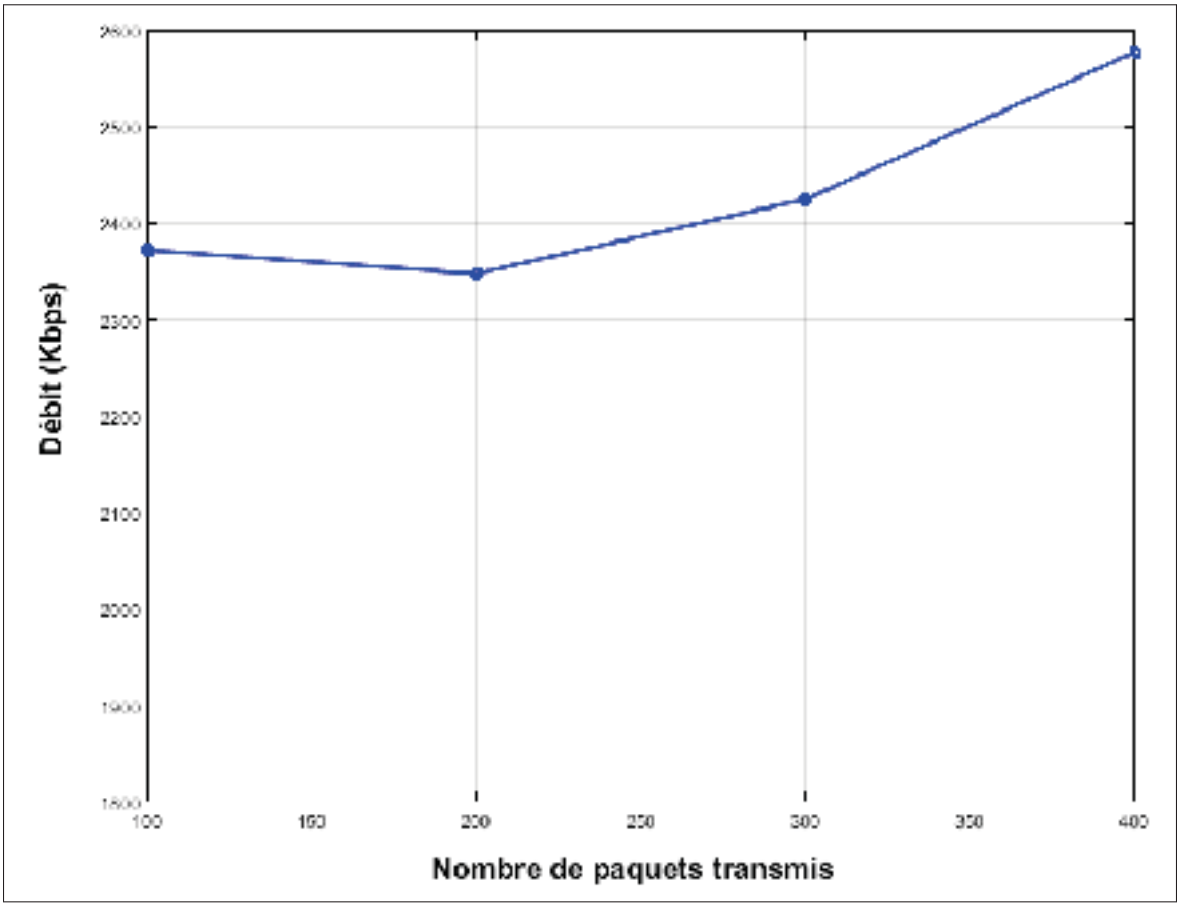

Figure 4.5 Débit en fonction du nombre de paquets .

Délai de bout en bout en fonction du nombre de paquets

Quand on augmente le nombre de paquets à transmettre, le délai de bout en bout est pratiquement stable. L'écart entre la valeur maximale et la valeur minimale est de l'ordre de 0,0003 secondes. Le délai augmente légèrement jusqu'au nombre de 300 paquets. Pour un nombre égal à 400 paquets, un abaissement léger a eu lieu. Pour le scénario de 400 paquets, quatre nœud n'ont pas reçu les paquets transmis, le nœud 11, le nœud 16, le nœud 17 et le nœud 19. OLSRV2 garantit un délai de bout en bout, pratiquement peu influencé par l'augmentation du nombre de paquets transmis.

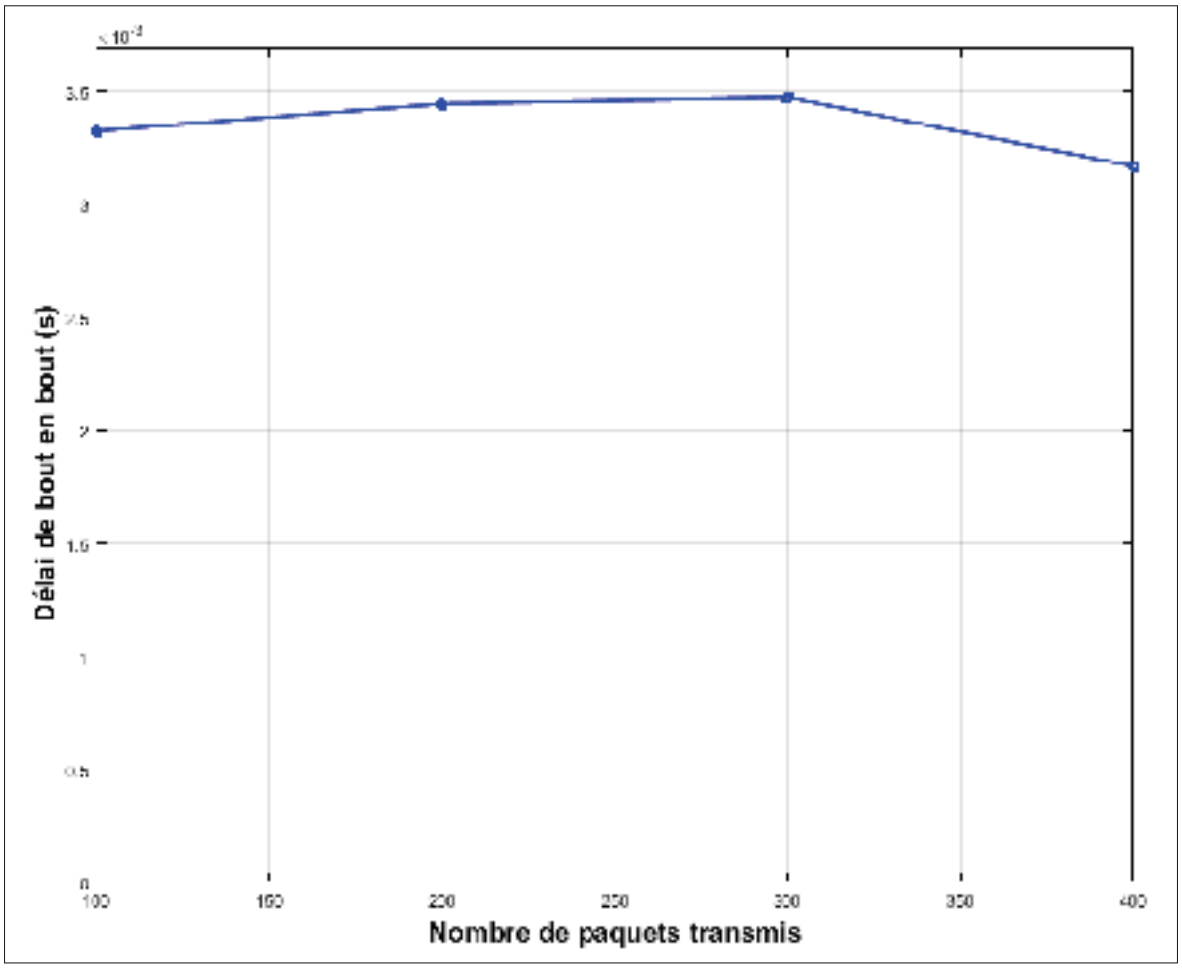

Figure 4.6 Délai de bout en bout en fonction du nombre de paquets

Taux de paquets livrés en fonction du nombre de paquets

OLSRV2 a réussi à garder un taux de paquets livrés stable en fonction du nombre de paquets transmis avec une valeur satisfaisante égale à 0,8, comme le montre la Figure 4.7. Des collisions et des interférences ont mené à des pertes de paquets. Certains nœuds sources ont décidé de laisser tomber les paquets après le nombre limite de retransmissions. La stabilité du taux de paquets livrés, indépendamment du nombre de paquets transmis dans le réseau, prouve la capacité d'OLSRV2 à supporter la congestion dans le réseau et à choisir des liens robustes garantissant des transmissions fiables.

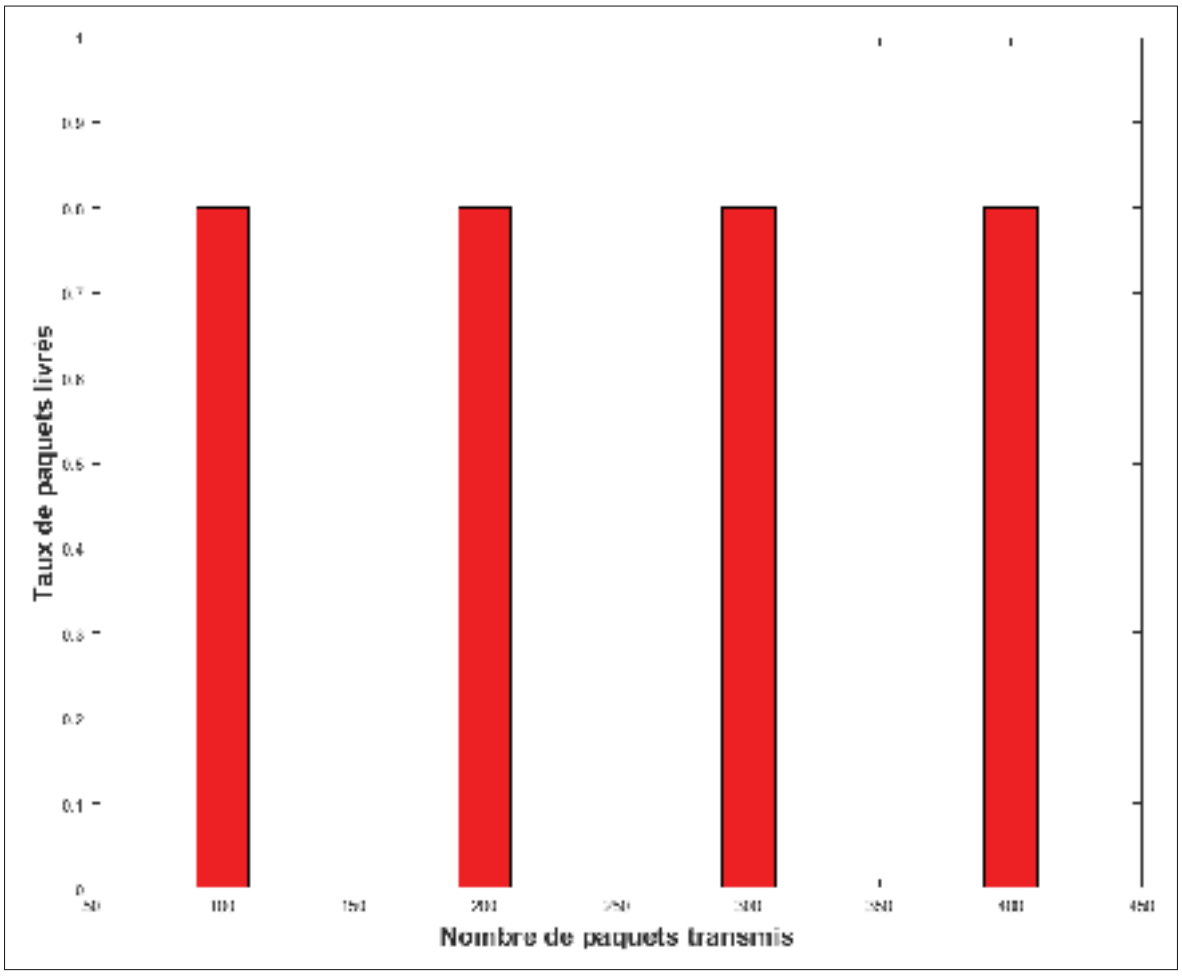

Figure 4.7 Taux de paquets livrés en fonction du nombre de paquets

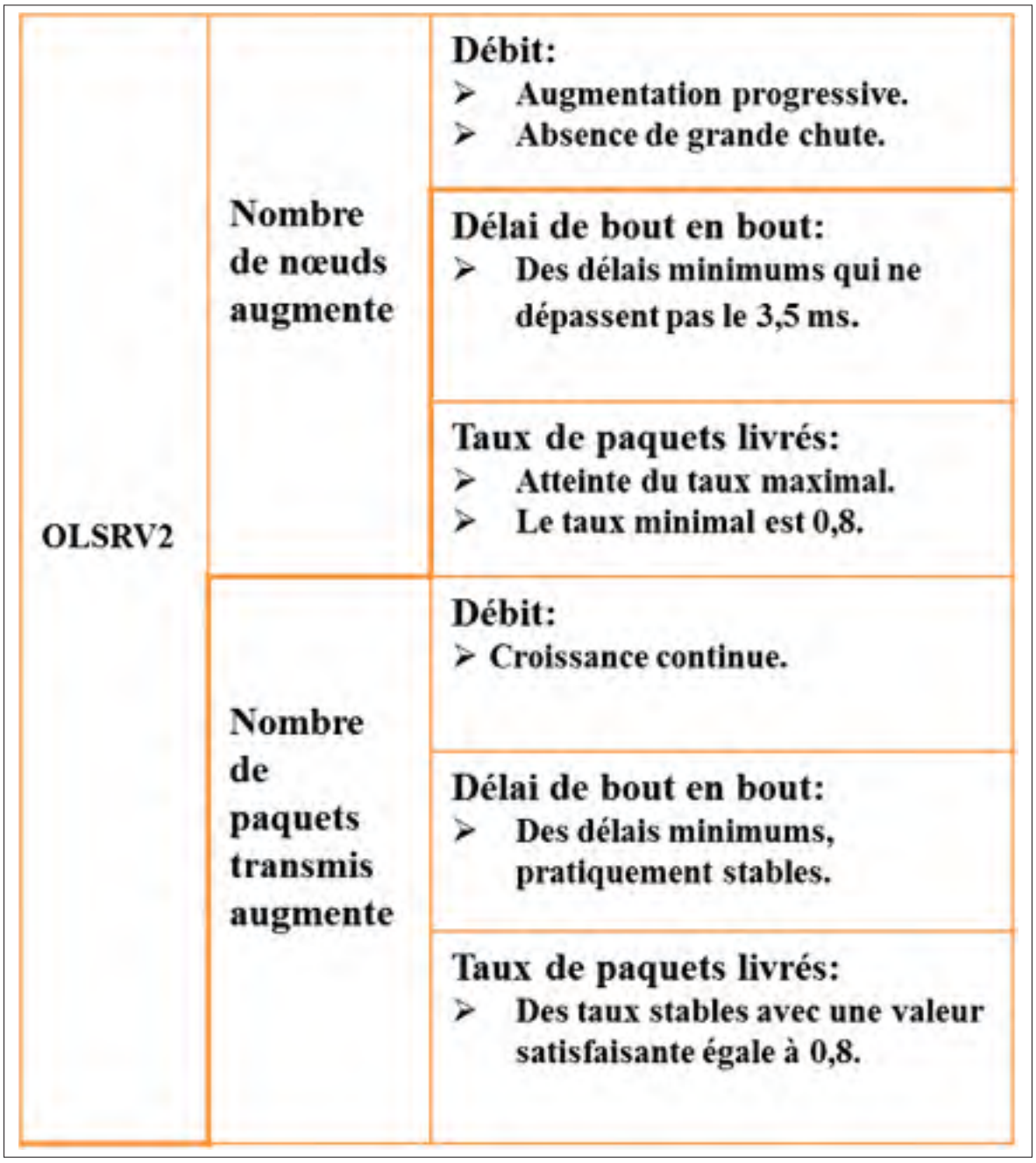

Figure 4.8 Résumé des performances .

CONCLUSION ET RECOMMANDATIONS

La technologie ad hoc présente une évolution dans la communication numérique. L'auto-organisation associée à la robustesse et la fiabilité des liens établis donnent la possibilité d'exploiter cette technologie dans divers domaines (le domaine militaire, le domaine industriel, etc). Le niveau de routage est un des niveaux à considérer lors du déploiement de cette technologie. L'autonomie et la mobilité caractérisant les nœuds dans le réseau ad hoc imposent aux protocoles de routage d'épouser l'aspect distributif et le dynamisme de la topologie. Plusieurs approches ont été proposées afin de garantir une convergence de données et le choix optimal des trajets. L'approche proactive avec ses mécanismes de mise à jour périodique vise à fixer toutes les routes optimales vers tous les autres nœuds dans le réseau et à rendre le tableau de routage disponible à tout moment. L'approche réactive avec son mécanisme de recherche de route sous demande a pour but de réduire la charge de trafic de contrôle dans le réseau. L'approche hybride essaye de combiner les mécanismes des approches réactives et proactives afin de créer plus de flexibilité face aux changements imprévisibles dans le réseau ad hoc.

Dans ce projet, notre but était de choisir un protocole de routage pour le réseau ad hoc et de l'intégrer dans le simulateur fourni afin d'analyser les performances résultantes. Une étude comparative qualitative a été effectuée dans le but de garantir le bon choix. La sélection a été basée sur les critères les plus importants caractérisant la qualité de déploiement de la technologie ad hoc. Ces critères sont le taux de paquets livrés, le délai de bout en bout, le volume de trafic de contrôle, la longueur de route, le débit et le taux de perte de paquets. Après une analyse des différents scénarios et des conclusions tirées de différents articles cités, le protocole de routage OLSR a été choisi. OLSR est le protocole de routage garantissant un taux de paquets livrés et un débit satisfaisants dans un réseau dense et avec un trafic intense. Un délai de bout en bout minimal est un résultat assuré du déploiement de l'approche proactive et donc d'OLSR. OLSR a subi des améliorations et des enrichissements de ses structures en raison de son exploitation continue. Une comparaison des versions disponibles a mené au choix de la deuxième version

présentée dans la RFC 7181. Une base de données bien structurée permet d'assurer le sauvegarde des informations valides du voisinage de chaque nœud. Divers ensembles permettent d'organiser les données concernant les états de liens et la validité temporelle de chaque statut. Une structure topologique vise à permettre à chaque nœud de concevoir sa vision valide sur le réseau entier.

Pour évaluer les performances résultantes, nous avons intégré le protocole OLSRV2 dans le simulateur fourni. Le but était d'évaluer le débit, le taux de paquets livrés et le délai de bout en bout en fonction du nombre de nœuds et du nombre de trafics. L'évaluation établie englobe les critères critiques lors de déploiement de la technologie ad hoc.Le temps d'exécution a augmenté en fonction du nombre de nœuds avec une valeur maximale égale à 0,689567368 (s) pour 20 nœuds. L'augmentation de temps d'exécution est plus clair en fonction du nombre de paquets. La valeur maximale est 1,26853536 (s) pour 400 paquets, par rapport à 0,3327586 (s) pour 100 paquets. L'utilisation de mémoire est pratiquement stable en fonction du nombre de nœuds et de paquets, avec une moyenne égale à 0,4. Après 100 exécutions pour chaque simulation, on a pu conclure que OLSRV2 garantit des performances satisfaisantes dans les deux scénarios traités. Dans le premier scénario, le débit a conservé des valeurs acceptables en fonction du nombre de nœuds. Le taux de paquets livrés a atteint pour un certain nombre de nœuds la valeur maximale. Dans le cas de perte de paquets, le taux de paquets livrés a gardé la valeur 0,8 ce qui est une valeur satisfaisante. Les différentes valeurs de délai de bout en bout enregistrées n'ont pas dépassé le 0,0035 (s). Dans le deuxième scénario, on a visé à évaluer les performances d'OLSRV2 dans un réseau avec un trafic croissant. Des bons résultats ont été relevés. Le débit a gardé une croissance en fonction du nombre de paquets transmis. Le délai de bout en bout a été pratiquement stable. Le taux de paquets livrés a gardé la valeur 0,8 pour tous les nombres de paquets.

La réalisation sous forme de code d'OLSRV2 aide à l'exploitation d'un protocole de routage garantissant des performances satisfaisantes. Des évaluations plus diversifiées peuvent être établies en ajoutant des métriques lors de la sélection du trajet optimal. On peut citer la métrique EDR (*Expected Data Rate*) qui vise à sélectionner le trajet associé au débit le plus élevé. Une autre métrique peut être exploité, soit (*Airtime Link Metric*), qui reflète le nombre de canaux utilisés. Des algorithmes de multi-trajets peuvent être exploités, également, dans le cas où plusieurs trajets disponibles sont nécessaires.

BIBLIOGRAPHIE

Abdellaoui, R. (2009). *SU-OLSR une nouvelle solution pour la sécurité du protocole OLSR.* (Mémoire de maîtrise, ets, Montreal).

Abolhasan, M., Wysocki, T. & Dutkiewicz, E. (2004a). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1), 1–22. doi : 10.1016/S1570-8705(03)00043-X.

Abolhasan, M., Wysocki, T. & Dutkiewicz, E. (2004b). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1), 1–22. doi : 10.1016/S1570-8705(03)00043-X.

Aggelou, G. & Tafazolli, R. (1999). RDMAR : A bandwidth-efficient Routing Ad hoc Networks Protocol for Mobile. 26–32.

Anjana, N. R., Valiveti, S., Garg, S. & Kotecha, K. (2010). Topology Management in Ad Hoc Network. *International Journal of Computer Applications*, 10(8), 1–5. doi : 10.5120/1505-2023.

Arun Kumar, B. R., Reddy, L. C. & Hiremath, P. S. (2008). Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols. *European Journal of Scientific Research*, 8(6), 337–343. doi : 10.1109/INMIC.2004.1492924.

Baccour, N., Koubâa, A., Youssef, H. & Alves, M. (2015). Ad Hoc Networks Reliable link quality estimation in low-power wireless networks and its impact on tree-routing. *Ad Hoc Networks*, 27, 1–25. doi : 10.1016/j.adhoc.2014.11.011.

Beijar, N. (2002). Zone Routing Protocol (ZRP). *Networking Laboratory, Helsinki University of Technology, Finland*, 1–12.

Biskupski, B., Dowling, J. & Sacha, J. (2007). Properties and mechanisms of self-organizing MANET and P2P systems. *ACM Transactions on Autonomous and Adaptive Systems*, 2(1), 1–es. doi : 10.1145/1216895.1216896.

Bisnik, N. (2007). Stochastic and information theoretic models for design and performance evaluation of mobile ad hoc and sensor networks. 2007(August 2007), 281.

Chen, P. Y.-s. (2006). Chapter 9 . Broadcast Storm Problem in a Mobile Ad Hoc Network. 153–167.

Clausen,Tand Dearlove, C. & Jacquet,Pand Herberg, U. (2014). RFC7181.

Clausen, T. & Dearlove, C. (2009). RFC 5497 Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs). 1–14.

Clausen, T. & Jacquet, P. (2003). RFC 3626 - Optimized Link State Routing Protocol (OLSR). 1–75.

Clausen, T., Jacquet, P. & Viennot, L. (2002). Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks. *Med-Hoc-Net' 02*, 1–10.

Clausen, T., Dearlove, C., Dean, J. & Adjih, C. (2009). RFC 5444 Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format. 1–60.

Clausen, T., Dearlove, C. & Dean, J. (2011). RFC 6130 - Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP).

Clausen, T., Jacquet, P. & Viennot, L. (2010). Comparative Study of Routing Protocols for Mobile Ad Hoc Networks To cite this version : HAL Id : inria-00471702 Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks.

Cooper, D. & Polk, W. (2008). Network Working Group. *Best Current Practice*, 1–7. doi : 10.1017/CBO9781107415324.004.

Corson, S. & Macker, J. (1999). RFC 2501 Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations. 1–12.

Das, S., Perkins, C. & Belding-Royer, E. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. 1–37.

Desai, R. & Patil, B. P. (2014). Performance Analysis of Ad Hoc Routing Protocols. 147(14), 41–46. doi : 10.5120/ijca2016911277.

Dhenakaran, S. & Parvathavarthini, a. (2013). An Overview of Routing Protocols in Mobile Ad-Hoc Network. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(2), 251 —- 259. Repéré à http://www.ijarcsse.com/docs/papers/Volume{_}3/2{_}February2013/V3I2-0201.pdf.

Dijkstra, E. W. (1959). A Note on T w o Problems in Connexion with Graphs. 271, 269–271.

Dijkstra, E. W. (1986). A belated proof of self-stabilization. *Distributed Computing*, 1(1), 5–6. doi : 10.1007/BF01843566.

Dressler, F. (2008). A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications*, 31(13), 3018–3029. doi : 10.1016/j.comcom.2008.02.001.

Dugaev, D., Zinov, S., Siemens, E. & Shuvalov, V. (2015). A survey and performance evaluation of ad-hoc multi-hop routing protocols for static outdoor networks. *2015 International Siberian Conference on Control and Communications, SIBCON 2015 - Proceedings*. doi : 10.1109/SIBCON.2015.7147048.

Fifer, W. C. & Bruno, F. J. (1987). The Low-Cost Packet Radio. *Proceedings of the IEEE*, 75(1), 33–42. doi : 10.1109/PROC.1987.13703.

Frodigh, M., Johansson, P. & Larsson, P. (2000). Wireless ad hoc networking - the art of networking without a network. *Ericsson Review (English Edition)*, 77(4), 248–263.

Groleau, R. (2012). *Notes_802_11*.

Haas, Z. J. (1997). New routing protocol for the reconfigurable wireless networks. 562–566.

Johnson, D. B. & Maltz, D. A. (1996). Dynamic source routing in a d hoc wireless networks. Dans *Mobile computing*.

Johnson, D. B., Maltz, D. a. & Broch, J. (2001). The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. *Computer Science Department Carnegie Mellon University Pittsburgh, PA*, 15213–3891. doi : 10.1007/BF01193336.

Jubin, J. & Tornow, J. (1987). The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1), 21–32. doi : 10.1109/PROC.1987.13702.

Karia, D. C., Jadiya, A. & Kapuskar, R. (2014). Review of routing metrics for wireless mesh networks. *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*, 47–52. doi : 10.1109/ICMIRA.2013.16.

Kulla, E., Hiyama, M., Ikeda, M., Barolli, L., Kolici, V. & Miho, R. (2010). MANET performance for source and destination moving scenarios considering OLSR and AODV protocols. *Mobile Information Systems*, 6(4), 325–339. doi : 10.3233/MIS-2010-0106.

Kumar Sarkar, S., Basavaraju, T. & Puttamadappa, C. (2013). *Ad Hoc Mobile Wireless Networks*.

Leiner, B. M., Ruth, R. J. & Sastry, A. R. (1996). Goals and challenges of the DARPA GloMo program. *IEEE Personal Communications*, 3(6), 34–42. doi : 10.1109/98.556477.

Leite, J. R. E., Ursini, E. L. & B, P. S. M. (2017). Ad-hoc, Mobile, and Wireless Networks. 10517, 199–209. doi : 10.1007/978-3-319-67910-5.

Loo, J., Khan, S. & Khwildi, A. N. A. (2016). Mobile Ad Hoc Network. Dans Jonathan Loo Jaime Lioret Mauri, J. H. O. (Éd.), *Mobile Ad Hoc Networks Current Status and Future Trends* (éd. 3, vol. 4, pp. 220–242). Boca Raton : CRC Press.

Man, L. A. N., Committee, S. & Computer, I. (2012). Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) S pecifications.

Mbarushimana, C. & Shahrabi, A. (2007). Comparative study of reactive and proactive routing protocols performance in mobile ad hoc networks. *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, 679–684. doi : 10.1109/AINAW.2007.123.

Mcquillanir, J. M., Ira, R. & Eric C, R. (1980). The New Routing Algorithm for the ARPANET. C(5), 711–719.

Mnif, K. (2006). *Construction et Maintenance d'une dorsale virtuelle dans les réseaux ad hoc mobiles*. (Mémoire de maîtrise, ets, Montreal).

Moussaoui, A. & Boukeream, A. (2015). A survey of routing protocols based on link-stability in mobile ad hoc networks. *Journal of Network and Computer Applications*, 47, 1–10. doi : 10.1016/j.jnca.2014.09.007.

Murthy, C. & Manoj, B. (2004). *Ad Hoc Wireless Networks : Architectures and Protocols*. PRENTICE HALL.

Murthy, S. & Garcia-Luna-Aceves, J. J. (1996). An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2), 183–197. doi : 10.1007/BF01193336.

Narayan, P. & Syrotiuk, V. (2003). Evaluation of the AODV and DSR routing protocols using the MERIT tool. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2865, 25–36.

Neumann, A., López, E. & Navarro, L. (2015). Evaluation of mesh routing protocols for wireless community networks. *Computer Networks*, 93, 308–323. doi : 10.1016/j.comnet.2015.07.018.

Partial, I. N., For, F., Degree, T. H. E. & Doctor, O. F. (2017). Link failure detection , network recovery , and network reliability in multi-hop wireless networks by.

Prakash, C. (2014). Mobile Ad hoc Network. 1–4.

Qasim, N., Said, F. & Aghvami, H. (2008). Mobile Ad Hoc Networks Simulations Using Routing Protocols for Performance Comparisons. *World Congress on Engineering*, I.

Qiu, T., Chen, N., Li, K., Qiao, D. & Fu, Z. (2017). Ad Hoc Networks Heterogeneous ad hoc networks : Architectures , advances and challenges. *Ad Hoc Networks*, 55, 143–152. doi : 10.1016/j.adhoc.2016.11.001.

Rashid, B. & Husain, M. (2016). Journal of Network and Computer Applications Applications of wireless sensor networks for urban areas : A survey. *Journal of Network and Computer Applications*, 60, 192–219. doi : 10.1016/j.jnca.2015.09.008.

Redi, J. (2002). A BRIEF OVERVIEW OF AD Hoc NETWORKS :. (May), 20–22.

Royer, E. M. & Toh, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2), 46–55. doi : 10.1109/98.760423.

Sanchez-Garcia, J., Garcia-Campos, J. M., Toral, S. L., Reina, D. G. & Barrero, F. (2016). A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios. *Proceedings - 2015 International Conference on Developments in eSystems Engineering, DeSE 2015*, 35–40. doi : 10.1109/DeSE.2015.12.

Sigward, E. (2002). *Introduction à la théorie des graphes*.

Subramanya Bhat, M., Shwetha, D. & Devaraju, J. (2011). A Performance Study of Proactive , Reactive and Hybrid Routing Protocols using Qualnet Simulator. 28(5), 10–17.

Varshney, P. K., Agrawal, G. & Sharma, S. K. (2016). Relative Performance Analysis of Proactive Routing Protocols in Wireless Ad hoc Networks using Varying Node Density. *Invertis Journal of Science & Technology*, 9(3), 161. doi : 10.5958/2454-762X.2016.00015.9.

Vazifehdan, J., Prasad, R. V. & Niemegeers, I. (2014). Energy-efficient reliable routing considering residual energy in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 13(2), 434–447. doi : 10.1109/TMC.2013.7.

Voorhaen, M. & Blondia, C. (2006). Analyzing impact of neighbor sensing on performance of OLSR protocol. *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt 2006*. doi : 10.1109/WIOPT.2006.1666446.

Walikar, G. A. & Biradar, R. C. (2017). A survey on hybrid routing mechanisms in mobile ad hoc networks. *Journal of Network and Computer Applications*, 77(August 2016), 48–63. doi : 10.1016/j.jnca.2016.10.014.

Weingärtner, E., Vom Lehn, H. & Wehrle, K. (2009). A performance comparison of recent network simulators. *IEEE International Conference on Communications*. doi : 10.1109/ICC.2009.5198657.

Weiser, M., Welch, B., Demers, A. & Shenker, S. (1996). *Mobile Computing*. doi : 10.1007/b102605.

Yang, L. T. & Wang, G. (2012). Journal of Network and Computer Applications. *Journal of Network and Computer Applications*, 35(3), 865–866. doi : 10.1016/j.jnca.2012.02.003.

Zafar, H., Alhamahmy, N., Harle, D. & Andonovic, I. (2011). Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks. *International Journal of Communication Networks and Information Security (IJCNIS)*, 3(3), 193–216.