

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER PROFESSIONNEL

En

Informatique

Option

Administration et Sécurité des Réseaux

Thème

**Simulation et évaluation des performances
des protocoles de routage AODV, OLSR et GPCR
pour les réseaux VANETs sous NS-3 et SUMO**

Présenté par : Mme MESSAOUDI Yamina.

Soutenu 2 Juillet 2017 devant le jury composé de :

Rapporteur	Mme N. TASSOULT	M.A.A	U. A/Mira Bejaia.
Président	Mme B. GASMI	M.A.A	U. A/Mira Bejaia.
Examineur	M H. KHENOUS	M.A.A	U. A/Mira Bejaia.
Examineur	Mme A. BOUDRIOUA	Doctorante	U. A/Mira Bejaia.

Promotion 2016-2017.

Remerciements

En préambule à ce mémoire je remercie ALLAH le tout puissant et miséricordieux, qui ma a donné la force et la patience d'accomplir ce modeste travail.

Je souhaite adresser mes remerciements les plus sincères à mes parents qui tout au long de ce travail, ils m'ont apporté leurs précieux soutien ainsi que leur encouragements.

*Je tiens à remercier sincèrement mon encadreur **Mme. TASSOULT Nadia** pour sa disponibilité, son aide et le temps consacré qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.*

Mon vifs remerciements vont également aux membres du jury qui ont accepter d'examiner mon travail et de l'enrichir par leurs propositions.

mes sincères reconnaissances à tous mes enseignants pour les efforts fournis durant toute la période d'étude.

Enfin, je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

A mes parents, pour leurs sacrifices déployés à mon égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de mes profondes affections et de mon attachements indéfectibles, nulle dédicace ne puisse exprimer ce que je leur doit.

A mes frères *Lemnouar, Hamza* et à mes sœurs *Hakima, Samia, Aicha, Touns, Rima* je tiens également à remercier ma belle sœur *Fatima* et tous mes amis, *Lynda, Samia, Sabrina, Warda, Soraya* pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Yamina

Table des matières

Table de matières	i
Liste des figures	ii
Liste des tableaux	iv
Liste des abréviations	v
Introduction générale	1
1 Vue générale sur les réseaux véhiculaires ad hoc	3
1.1 Réseaux ad-hoc	3
1.2 Réseaux VANET	4
1.2.1 Nœud du réseau VANET	5
1.2.2 Messagerie et architectures des réseaux sans fil véhiculaires . .	6
1.2.2.1 Types de message	7
1.2.3 Architecture de communication	8
1.2.3.1 Vehicle-to-vehicle V2V	8
1.2.3.2 Vehicle-to-Infrastructure V2I	9
1.2.3.3 Vehicle-to-road V2R (Hybrides)	9
1.2.4 Application des réseaux VANET	9
1.2.5 Caractéristiques les réseaux VANETs	13
1.2.6 Technologies de transmission	13
1.2.7 Travaux dans le domaine des VANETs	15
1.2.7.1 Sécurité	15
1.2.7.2 L'accès au canal	15
1.2.7.3 Localisation des véhicules	15

1.2.7.4	Problèmes de congestion	15
1.2.7.5	Mobilité dans la simulation des réseaux	16
1.2.7.6	Routage	16
2	Protocoles de routage	17
I	Protocoles de routage dans les réseaux véhiculaires	18
2.1	Protocoles de routages dans les VANETs	19
2.1.1	Classification	19
2.1.2	Différents types de routages	20
2.1.2.1	Protocoles Unicast Ad-hoc et leurs adaptations pour les VANETs	21
2.1.2.2	Protocoles Unicast de groupe	22
2.1.2.3	Protocoles Unicast basé sur la position	23
2.1.2.4	Protocoles Broadcast	23
2.1.2.5	Protocoles Multicast et Geocast	24
II	Études des Protocoles AODV, OLSR et GPCR	25
2.2	Principe fonctionnement de protocole OLSR	26
2.2.1	Processus de construction des routes dans OLSR	26
2.3	Principe fonctionnement de protocole AODV	29
2.3.1	Méthode de construction d'une route	29
2.3.2	Maintenance de route	30
2.4	Principe fonctionnement de protocole GPCR	31
2.4.1	Routage greedy restaint	31
2.4.2	Stratégie de réparation	31
2.5	Comparaisons entre les protocoles OLSR, AODV et GPCR	32
3	Simulation et évaluation des protocoles	34
3.1	Présentation des simulateurs	35
3.1.1	Simulateur de trafic routier	35
3.1.1.1	JOSM (Java Open Street Maps)	35
3.1.1.2	eWorld.1.0.0	36
3.1.1.3	SUMO.28 (Simulator of Urban Mobility.28)	36
3.1.2	Simulateur réseau	37

3.1.2.1	NS3.19(Network Simulator 3.19)	37
3.2	Intégration de GPCR à NS-3 et les problèmes rencontrés	38
3.3	Étapes de la simulation	39
3.3.1	Simulation du trafic routier	40
3.3.1.1	Préparation de la carte avec eWorld	41
3.3.1.2	Création des scénarios avec SUMO	41
3.3.2	Génération des fichiers traces de mobilité	43
3.3.3	Collecte d'informations	45
3.3.3.1	Fonction de génération du trafic	45
3.3.3.2	Fonction ReceivePacket	45
3.4	Critères d'évaluation	46
3.4.1	Taux de livraison de paquets	46
3.4.2	Délai de bout en bout	46
3.4.3	Paquets perdus	47
3.4.4	Débit moyen	47
3.5	Interprétation et évaluation des résultats	47
3.5.1	Scénarios	47
3.5.1.1	Premier scénario	48
3.5.1.2	Deuxième scénario	50
3.5.1.3	Troisième scénario	53
	Bibliographie	59
	ANNEXE I : Étape de JOSM à SUMO	i
	ANNEXE II : Les résultats de la simulation	vi
	ANNEXE III :Installation des outils	viii
	ANNEXE IV : Tableau comparatif entre les simulateurs	x
	ANNEXE V : l'outil utilisé pour établir les graphes de simulation	xi
	ANNEXE VI : Script de simulation	xii

Table des figures

1.1	<i>Exemple de transmission d'un message dans un réseau ad-hoc [5].</i>	4
1.2	<i>Hiérarchie des réseaux sans fil [20].</i>	5
1.3	<i>Véhicule intelligent [20].</i>	5
1.4	<i>Exemple de réseau véhiculaire [18].</i>	6
1.5	<i>Mode de communication V2V [2].</i>	8
1.6	<i>Mode de communication V2I.</i>	9
1.7	<i>Mode de communication V2R [12] [24].</i>	9
1.8	<i>Application de sûreté et de sécurité routière [21].</i>	11
1.9	<i>Alerte du piéton [21].</i>	12
2.1	<i>Différents types de routages [13].</i>	20
2.2	<i>Classifications des protocoles de routages dans les VANETs [13].</i>	21
2.3	<i>Cluster-based Routing [13].</i>	22
2.4	<i>GPCR Routing [10].</i>	23
2.5	<i>diffusion classique (a) et diffusion optimisé à travers les MPR (b) [9].</i>	27
2.6	<i>Diffusion de messages avec MPR [9].</i>	28
2.7	<i>Méthode de construction d'une route [16].</i>	29
2.8	<i>Principe de détection d'un lien défaillant [16].</i>	30
2.9	<i>Routage greedy restaint [10].</i>	32
3.1	<i>JOSM.</i>	35
3.2	<i>eWorld.</i>	36
3.3	<i>SUMO.</i>	37
3.4	<i>NS3.</i>	38
3.5	<i>Schéma récapitulatif des étapes de la simulation.</i>	40
3.6	<i>Nod.xml.</i>	41

3.7	<i>Edge.xml</i>	42
3.8	<i>Flow.xml</i>	42
3.9	<i>sumo.cfg</i>	43
3.10	<i>Simulation de protocole GPCR</i>	44
3.11	<i>Taux de livraison de paquets</i>	48
3.12	<i>Délai moyen de bout-en-bout</i>	49
3.13	<i>Débit en (bit/s)</i>	50
3.14	<i>Nombre de paquets perdus</i>	50
3.15	<i>Taux de livraison de paquets</i>	51
3.16	<i>Délai</i>	52
3.17	<i>Débit</i>	52
3.18	<i>Nombre de paquets perdus</i>	53
3.19	<i>Taux de livraison de paquets</i>	53
3.20	<i>Délai</i>	54
3.21	<i>Débit</i>	55
3.22	<i>Nombre de paquets perdus</i>	55
3.23	<i>Recherche un lieu</i>	i
3.24	<i>Télécharger la carte</i>	ii
3.25	<i>Importation de la carte</i>	ii
3.26	<i>Ajout des points de départ et d'arrivée</i>	iii
3.27	<i>Exporter la carte vers SUMO</i>	iv
3.28	<i>SUMO</i>	v
3.29	<i>EXCEL</i>	xi

Liste des tableaux

2.1	Tableau comparatif.	33
3.1	Les paramètres utilisés dans les différents scénarios.	48
3.2	Taux de livraison des paquets (premier scénario).	vi
3.3	Débit (bit/s) (premier scénario).	vi
3.4	Délai (ms) (premier scénario).	vi
3.5	Nombre de paquets perdus (premier scénario).	vi
3.6	Taux de livraison des paquets (deuxième scénario).	vii
3.7	Débit (bit/s) (Deuxième scénario).	vii
3.8	Délai(ms)(Deuxième scénario).	vii
3.9	Nombre de paquets perdus (Deuxième scénario).	vii
3.10	Taux de livraison des paquets (troisième scénario).	vii
3.11	Débit (bit/s) (troisième scénario).	vii
3.12	Délai(ms)(troisième scénario).	viii
3.13	Nombre de paquets perdus (troisième scénario).	viii
3.14	Tableau comparatif des différents Simulateurs.	x

Liste des abréviations

AODV	Ad-hoc On-demand Distance Vector .
CPU	Central Processing Unit .
DSDV	Dynamic Destination Sequence Distance Vector .
DSRC	Dedicated Short Range Communication .
DVD	Digital Versatile Disc .
DSR	Dynamic Source Routing .
GPCR	Greedy Perimeter Coordinator Routing .
GPSR	Greedy Perimeter Stateless Routing .
GPS	Globale Position System .
HV-TRADE	History-enhanced V-TRADE .
IVG	Inter Vehicular Geocast .
IHM	L'interface Homme-Machine .
JOSM	Java Open Street Maps .

MAC	Medium Acces Control.
MPRs	Multi Point Relays.
NS-3	NETwork Simulator 3.
NS-2	Network Simulator 2.
OBU	One Boaed Unit.
OLSR	Optimized Link State Routing Protocol.
OSI	Open Open Systems Interconnection .
PDR	Packet Delivery Ratio.
PRW	Road Sid Unit.
4G	Quatrième Génération.
RSU	Road Side Unit.
RREQ	Route- Request.
RREP	Route-Reply.
RERR	Request- Error.
STI	Système de Transport Intelligent Delivery.
SUMO	Simulator of Urban Mobility.
TC	Topology Control.
3G	Troisième Génération.

VANET	V ehicular A d hoc NET work.
V2I	V éhicule T o I nfrastructure.
V2V	V éhicule T o V éhicule.
V2R	V ehicle- T o- R oad.
V-TRADE	V ector-based TR Acking D etection.
WiMax	W orldwide I nteroperability for M icrowave access.
WAVE	W ireless A ccess in V ehicular E nvironment.
Wi-Fi	W ireless F idelity.
ZRP	Z one R outing P rotocol.

Introduction générale

Le domaine de la télécommunication sans fil a connu une évolution importante et cela est due aux besoins actuels en termes de disponibilité et d'accès aux données à n'importe quel moment et depuis n'importe quel endroit. De plus en plus de foyers possèdent au moins un véhicule. Cette situation a conduit à une croissance du trafic routier qui a fortement aggravé le problème de congestion des systèmes de transport et qui a créé un risque significatif pour la sécurité des personnes. C'est dans cet esprit que sont apparues un certain nombre de recherches qui visent non seulement de réduire le nombre de morts sur les routes et améliorer les conditions de la circulation, mais aussi de diminuer les embouteillages et la pollution.

Un réseau véhiculaire mobile (VANETs) est un réseau de type ad-hoc. Un réseau ad-hoc est capable de se mettre en place de façon autonome. Ils peuvent utiliser, de façon opportuniste, les communications avec des infrastructures, permettant ainsi un accès à d'autres réseaux et donc aussi, à Internet. Les différentes infrastructures utilisées dans le cadre des VANETs sont appelées (RSU)"Unités de Bords de Routes". Ces unités peuvent être des feux de circulation, des parcomètres, ou toute autre borne placée au bord de la route.

Aujourd'hui des nombreuses applications ont vu pour améliorer notre vie : dans nos maisons, nos sociétés, nos voitures . . . etc. Une des applications de ce concept consiste à renforcer la prévention routière et à doter nos voitures et nos routes de capacités permettant de rendre la route plus sûre (les informations sur le trafic, les accidents, les dangers, les déviations possibles, les informations météorologiques... etc.), améliorer le confort des passagers et rendre le temps passé sur les routes plus conviviale (accès à internet, jeux interactifs entre les passagers des véhicules proches... etc.). Cette application est l'exemple type de ce que nous appelons les

systèmes de transports intelligents ITS (Intelligent Transportation System) et dont le but est : premièrement, l'amélioration de la sécurité des déplacements, deuxièmement, l'amélioration de l'efficacité globale du système de transport en réduisant les temps de parcours et les congestions, et enfin l'amélioration du confort de l'utilisateur en lui fournissant une multitude de services d'information, d'aide à la décision, de guidage et d'accès à Internet. Les caractéristiques des réseaux VANETs, font que les protocoles de routage dédiés aux réseaux MANET sont inefficaces. Par conséquent, plusieurs chercheurs dans le domaine des réseaux mobiles veulent concevoir le protocole qui pourrait être le standard. Afin d'étudier les VANETs, le déploiement sur terrain n'est malheureusement pas envisageable à ce jour, d'où le recours à la simulation. Plusieurs simulateurs ont été mis à la disposition des chercheurs dans ce but (NS2, NS3, . . .).

Dans ce mémoire nous avons choisi trois protocoles de routage AODV, OLSR et GPCR afin de d'évaluer leurs performances. Pour ce faire nous avons utilisé NS-3 (Network Simulator 3) version 3.19. Pour simuler le trafic routier, nous avons utilisé une carte de réseau routier que nous avons configurée dans l'outil eWorld après l'avoir importé par JOSM, puis nous l'avons exporté vers le Simulateur SUMO (Simulator Urban Mobility) version 0.28.0 pour générer le modèle de mobilité. Nous l'avons ensuite intégré à NS-3 à l'aide de Ns2MobilityHelper.

Notre mémoire est composé de trois chapitres. Dans le premier chapitre nous présentons de manière générale les réseaux véhiculaires (caractéristiques, architectures, applications). Dans le deuxième chapitre nous avons détaillé les différents protocoles de routage dans les réseaux véhiculaires et leur fonctionnement. Le troisième chapitre est consacré à l'évaluation des protocoles AODV, OLSR et GPCR, Nous effectuons en premier lieu une comparaison de performances entre ces protocoles en termes de taux de livraison de paquets, délai de bout en bout, nombre de paquets perdus et débit moyen. Ainsi nous présentons les résultats obtenus au cours de la simulation. Nous terminons par une conclusion générale, et quelques perspectives.

Introduction

Les réseaux VANET ne sont qu'une application des réseaux ad-hoc mobiles(MANET). Ils constituent le noyau d'un Système de Transport Intelligent(STI) ayant comme objectif principal l'amélioration de la sécurité routière en tirant profit de l'émergence de la technologie de communication et la baisse du coût des dispositifs sans-fil. En effet grâce à des capteurs installés au sein de véhicules ou bien situés au bord des routes et des centres de contrôle les communications véhiculaires permettront aux conducteurs d'être avertis suffisamment tôt de dangers éventuels [5].

Ces réseaux sont utilisés pour répondre aux besoins de communication appliquée aux réseaux de transport pour améliorer la conduite et la sécurité routière aux utilisateurs de la route [7].

1.1 Réseaux ad-hoc

Les réseaux ad-hoc sont des réseaux sans-fil capables de s'organiser spontanément et de manière autonome dans l'environnement dans lequel ils sont déployés sans infrastructure définie préalablement. La tâche de la gestion du réseau est répartie sur l'ensemble d'entités communicantes par liaison sans-fil, ces entités sont souvent appelées «nœuds». Dans ces réseaux les entités envisagées sont des terminaux légers et de taille réduite qui fonctionnent sur batterie, donc elles ont des capacités de traitement et de mémoire limitées, Les réseaux ad-hoc dans leur configuration mobile sont connus sous le nom de MANET (pour Mobile Ad-hoc NETWORKS) [5]. La FIGURE 1.1 montre un exemple de transmission d'un message dans un réseau ad-hoc entre deux équipements distants A et C

L'entité A veut communiquer avec C. Puisqu'elles sont hors de portée directe de transmission, A transmet son message vers B, qui à son tour le relaie vers C, comme ces deux derniers ne peuvent pas communiquer directement à cause de la portée limitée de supports de transmission utilisés, alors ils utilisent l'équipement B comme relai.

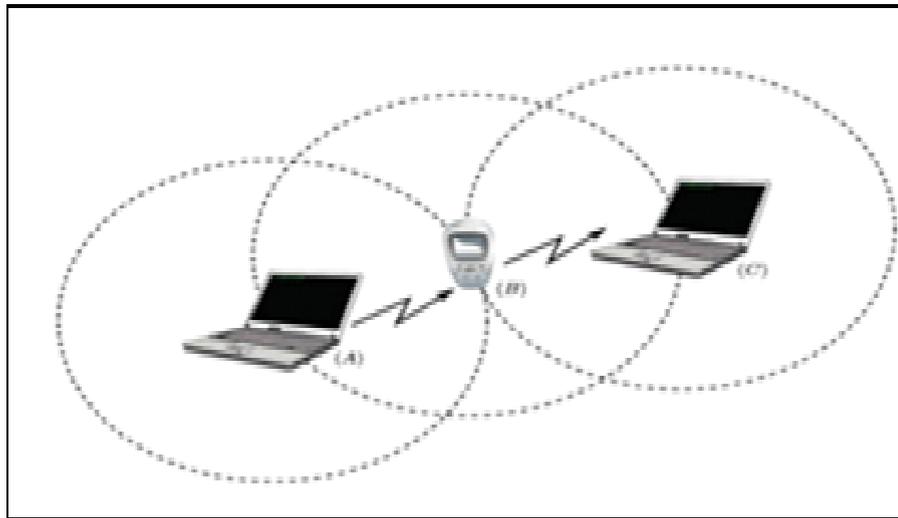


FIGURE 1.1 – *Exemple de transmission d'un message dans un réseau ad-hoc [5].*

1.2 Réseaux VANET

Un réseau VANET est une particularité des réseaux MANET où les nœuds mobiles sont des véhicules (intelligents) équipés de calculateurs, de cartes réseau et de capteurs. Comme tout autre réseau Ad-hoc les véhicules peuvent communiquer entre eux (pour échanger les informations sur le trafic par exemple) ou avec des stations de base placées tout au long des routes (pour demander des informations ou accéder à internet...).

Les réseaux véhiculaires regroupent deux grandes classes d'applications, à savoir les applications qui permettent de bâtir un système de transport intelligent ITS (Intelligent transport System) et celles liées au confort ou avertissement du conducteur et des éventuels passagers [20].

La FIGURE 1.2 représente la hiérarchie des réseaux sans fil où elle schématise l'inclusion des réseaux véhiculaires ad-hoc VANET dans les réseaux mobile Ad-hoc

MANET, les MANET dans les réseaux Mobiles ainsi que les réseaux mobiles dans les réseaux sans fil [20].

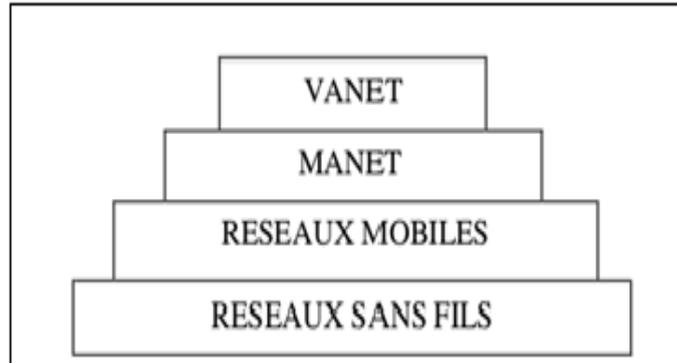


FIGURE 1.2 – Hiérarchie des réseaux sans fil [20].

1.2.1 Nœud du réseau VANET

Un nœud d'un réseau VANET est un véhicule équipé de terminaux tels que les calculateurs les interfaces réseaux ainsi que des capteurs capables de collecter les informations et de les traiter. On parle de la notion de « véhicule intelligent ». La FIGURE 1.3 modélise un véhicule intelligent [20].

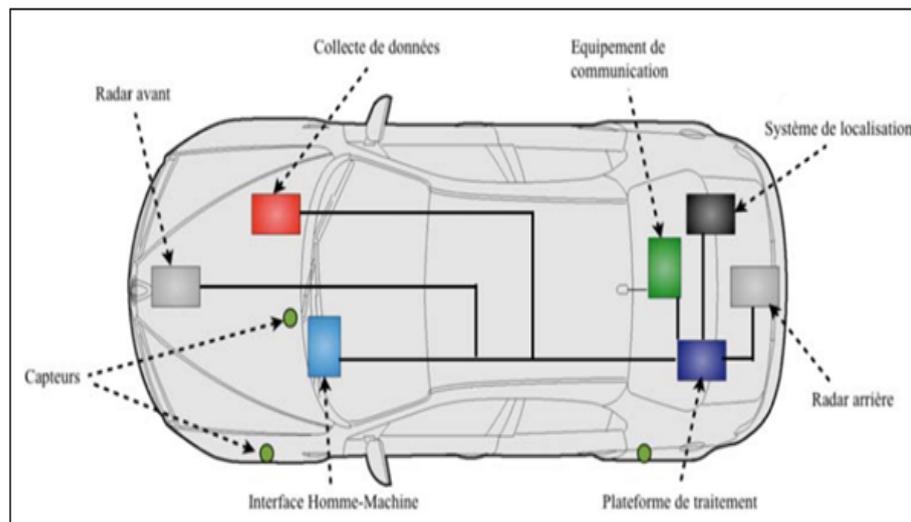


FIGURE 1.3 – Véhicule intelligent [20].

1.2.2 Messagerie et architectures des réseaux sans fil véhiculaires

Un réseau sans fil véhiculaire est un ensemble d'entités communicantes organisées selon une architecture de communication. Ces entités embarquées peuvent rencontrer différents environnements (urbain, autoroutier) ayant leurs contraintes propres [18].

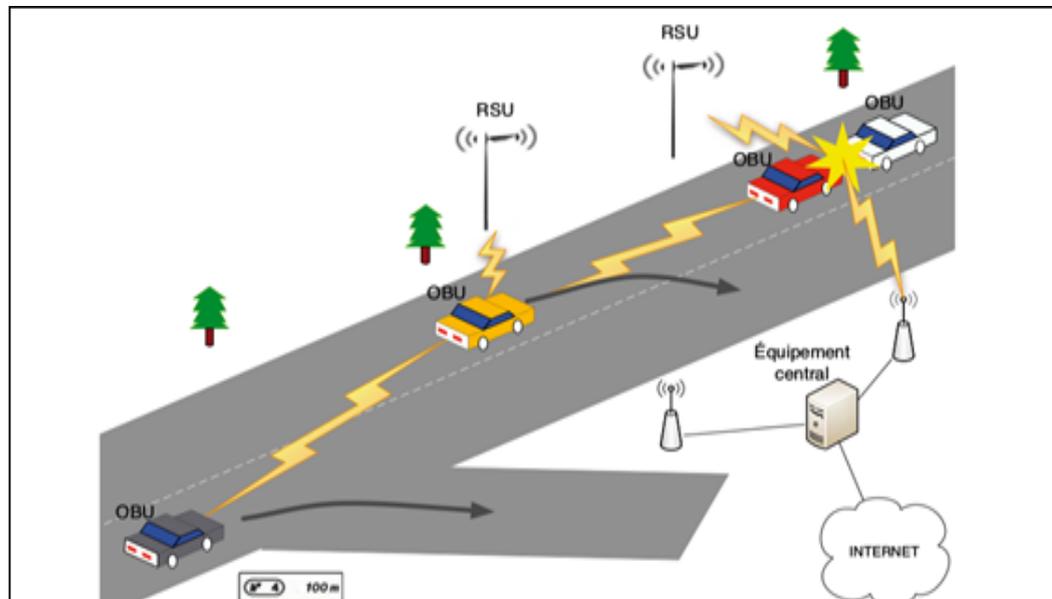


FIGURE 1.4 – Exemple de réseau véhiculaire [18].

Dans un réseau sans fil véhiculaire, il existe quatre entités communicantes : l'équipement personnel, le véhicule, l'équipement de bord de route et l'équipement central. La FIGURE 1.4 illustre un exemple de réseau véhiculaire faisant intervenir les différentes entités lors d'un accident de la route.

- Les équipements personnels sont les équipements qui peuvent être apportés par l'utilisateur à l'intérieur de son véhicule. Cela peut être un téléphone portable, un ordinateur portable ou encore un GPS autonome. Ces équipements peuvent interagir avec le véhicule. De nos jours en activant l'interface Bluetooth du téléphone portable on peut utiliser son téléphone portable par commande vocale (en utilisant les microphones intégrés au véhicule) ou par le biais de l'interface Homme-Machine (IHM) du véhicule.
- Les véhicules modernes sont équipés d'un ensemble de processeurs connectés à une plateforme centrale de calcul qui dispose d'interfaces filaires et sans fil.

Les véhicules intelligents sont des véhicules équipés d'une unité nommée On-Board Unit (OBU). Cette unité peut enregistrer, calculer, localiser et envoyer des messages sur une interface réseau.

- Les entités de bord de route sont appelées Road-Side Unit (RSU). Ces unités peuvent informer les véhicules à proximité en diffusant les conditions de trafic, météorologiques ou spécifiques à la route (vitesse maximale, autorisation de dépassement, etc.). Les RSU peuvent aussi jouer le rôle de station de base en relayant l'information envoyée par un véhicule.
- L'équipement central se situe du côté « serveur ». Il est transparent pour l'utilisateur. Cet équipement central pourra être un serveur de stockage, un point d'entrée à un réseau filaire (Internet) ou un serveur de transaction (télépéage par exemple).

1.2.2.1 Types de message

Les entités formant un réseau sans fil véhiculaire vont générer et s'échanger des messages. En fonction de l'application et du contexte environnemental un véhicule peut envoyer (ou recevoir) un message de contrôle, d'alerte ou « autre » [18].

- ***Message de contrôle***

Le message de contrôle est généré à intervalle régulier. Conventionnellement chaque véhicule émet un message de contrôle toutes les 100 ms. Ce message appelé aussi « beacon » contient la position, la vitesse, la direction et l'itinéraire du véhicule émetteur. Grâce aux messages de contrôle chaque véhicule se crée une vue locale de son voisinage [18].

- ***Message d'alerte***

Le message d'alerte est généré lorsqu'un événement est détecté. Cela peut être la détection d'un accident, d'un obstacle ou la réception d'un autre message d'alerte. Le message d'alerte doit être émis à intervalle régulier afin d'assurer la pérennité de l'alerte. Ainsi le ou les véhicules désignés pour la retransmission des messages émettront des alertes à instants réguliers. Les messages d'alerte doivent donc être de taille réduite pour être transmis le plus rapidement possible. Les messages contiennent en particulier les coordonnées du lieu de l'accident et les paramètres de la zone de retransmission [18].

- **Autres messages**

Ce type de message contient tous les messages qui ne sont pas des messages d’alerte ou de contrôle. Ces messages ne sont généralement pas répétés à intervalle régulier. En effet, cela peut être par exemple un message de transaction financière ou l’envoi de courrier électronique. Tous les messages reçus seront stockés dans un « cache des messages récemment reçus ». Chaque message se verra associer une durée de vie dans le cache [18].

1.2.3 Architecture de communication

L’objectif d’une architecture VANET est de permettre la communication entre les véhicules à proximité et entre les véhicules et les équipements routiers fixes menant aux trois possibilités suivantes :

1.2.3.1 Vehicle-to-vehicle V2V

Une approche purement ad hoc pour des communications véhicule à véhicule a un saut ou a plusieurs sauts (voir FIGURE 1.5) [2].

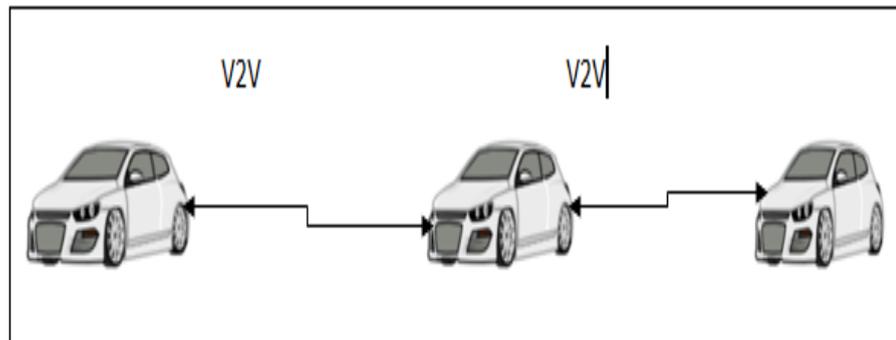


FIGURE 1.5 – Mode de communication V2V [2].

1.2.3.2 Vehicle-to-Infrastructure V2I

Une approche purement centralisée pour des communications de véhicule à infrastructure uniquement à un saut (voir FIGURE 1.6)[12].

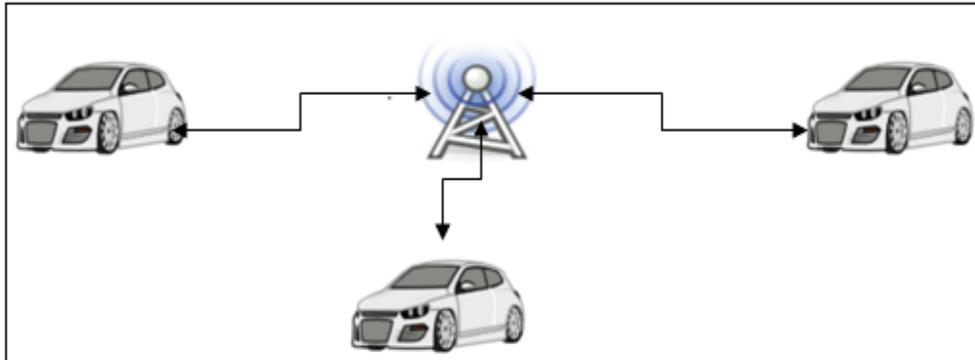


FIGURE 1.6 – Mode de communication V2I.

1.2.3.3 Vehicle-to-road V2R (Hybrides)

Une approche hybride permettant a un véhicule de communiquer avec un autre véhicule et/ou avec une infrastructure à un saut ou à multi-sauts, incluant les V2V et V2I (voir FIGURE 1.7)[12] [24].

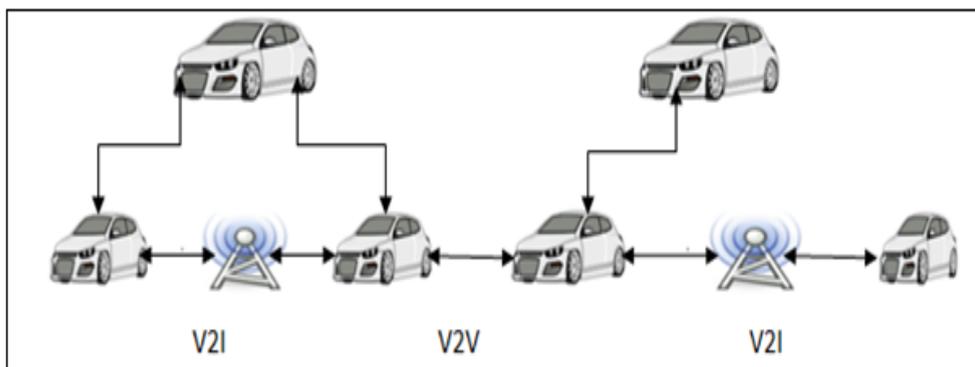


FIGURE 1.7 – Mode de communication V2R [12] [24].

1.2.4 Application des réseaux VANET

Les communications V2V et V2I permettent le développement d'un grand nombre d'applications et peuvent fournir un large éventail d'informations aux

conducteurs et aux voyageurs. Plusieurs véhicules ont la capacité à collecter, traiter et diffuser des informations sur eux-mêmes et leurs environnements à d'autres véhicules. Cela a conduit à l'amélioration de la sécurité routière et le confort des passagers. Les applications VANET sont classées selon leurs principaux objectifs en [14] [21] :

a. Applications du confort des conducteurs et des passagers : Cette catégorie d'applications visent à améliorer les conducteurs et les passagers des niveaux de confort. L'accès à Internet est la technologie clé pour la plupart des applications de confort et, par conséquent, la plupart des travaux dans cette catégorie se concentre sur elle. Par exemple, des fichiers multimédias, des DVD, de la musique peuvent être téléchargés sur le système de divertissement de voiture. le conducteur pourrait recevoir de l'information concernant les restaurants, les hôtels, les annonces de stations d'essence [21].

b. Applications de sécurité : Ces applications utilisent la communication sans fil entre les véhicules ou entre les véhicules et les infrastructures, afin d'améliorer la sécurité routière et éviter les accidents, l'intention étant de sauver la vie des gens et de fournir un environnement sans accidents.

Les applications des VANETs de communication utilisées entre véhicule ou avec infrastructure conduit à une augmentation du niveau de la sécurité routière. Les applications de sécurité en utilisant la communication V2V ou la communication V2I, ou les deux peuvent être classés comme suit [21] :

b.1 Avertissement des risques de collision : L'amélioration des systèmes pour éviter les collisions conduira à éviter de nombreux accidents de la route ; ce système est basé sur I2V ou communication V2I. Les capteurs à l'infrastructure collectent, traitent et analysent les informations provenant des véhicules se déplaçant à proximité de l'intersection [21], en fonction de l'analyse des données, s'il y a une probabilité d'un accident ou d'une situation dangereuse, un message d'avertissement est envoyé aux véhicules dans la zone d'intersection pour les avertir de la possibilité de l'accident afin qu'ils puissent prendre les mesures appropriées pour l'éviter. Il existe de nombreuses applications qui relèvent des collisions aux intersections des systèmes. Ces applications sont les suivantes :

b.2 Sécurité publique : les applications de sécurité publique visent à aider les

conducteurs en cas d'accident et pour soutenir les équipes d'urgence en réduisant leur temps de voyage et de fournir leurs services, pour la plupart des véhicules d'urgence, le temps de réponse est gaspillé dans leur chemin vers la destination. Les applications les plus connues dans cette catégorie sont :

b.2.1 Avertissement d'incidents : Cette application vise à prévenir des incidents potentiels suite à une situation dangereuse. Par exemple, s'il y a du brouillard sur la voie, un véhicule en panne, suite à des problèmes mécaniques ou à un accident, envoie des messages d'avertissement aux véhicules à l'approche circulant dans la même direction ou en direction opposée, en utilisant des communications V2V et V2I, afin de les informer de sa situation et de sa localisation, un exemple de cet application est illustré dans la FIGURE 1.8 suivante.

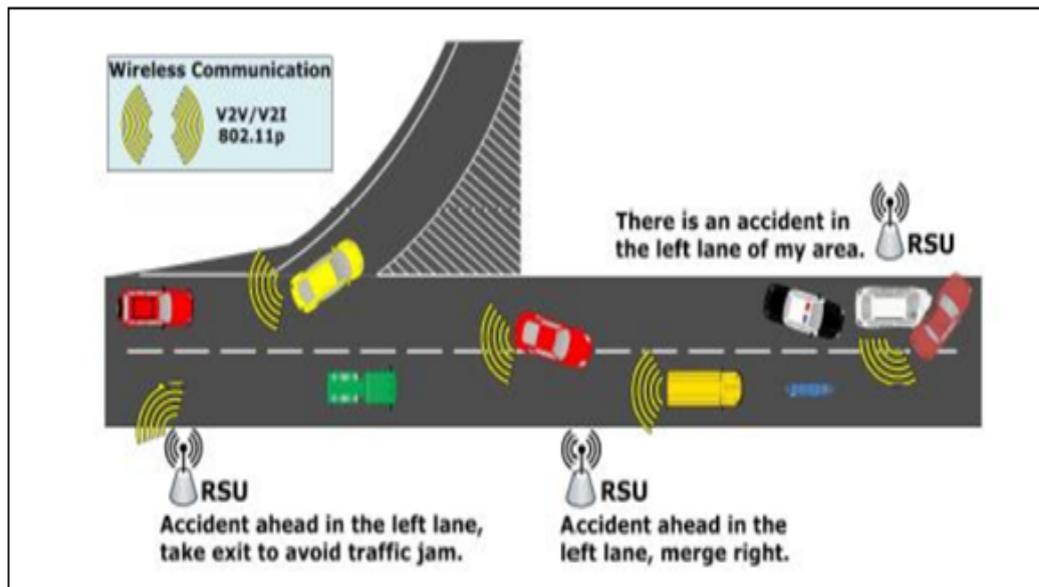


FIGURE 1.8 – Application de sûreté et de sécurité routière [21].

C. Informations sur d'autres véhicules : Ce type d'application utilisent les communications à courte portée entre un véhicule et d'autres véhicules à proximité, repose sur la communication V2V, V2I ou bien les deux pour exécuter des applications [21].

Les applications d'informations peuvent être classifiées comme suit [21] :

C.1 Avertissement de changement de voie : cette application est conçue pour éviter les accidents qui pourraient se produire en raison de changement de voie qui est une décision prise par le conducteur. Le système recueille des données

sur le véhicule et les véhicules environnants. Telles que la vitesse, la direction et la position du véhicule, et lorsque le conducteur décide de changer sa voie actuelle, le système traite les données recueillies et évalue si la décision va conduire à un accident. Le système émet alors un avertissement pour alerter le conducteur sur la situation potentiellement dangereuse et utilise la communication V2V pour alerter les autres véhicules.

C.2 Etat d'alerte de la route : Ce système alerte les véhicules sur les mauvaises conditions routières causées par des substances de la route, afin de prévenir les accidents. Les capteurs de bord de route recueillent des données relatives à la route afin de déterminer s'il y a des conditions dangereuses, puis diffuse des messages d'avertissement aux véhicules, ce qui suggère un ajustement de vitesse pour éviter les accidents.

Les collisions avec les piétons, résultent souvent lorsque les pilotes se précipitent leurs virages à gauche et à droite. La FIGURE 1.10 montre L'avertissement du piéton (PRW) permet aux véhicules de prédire les collisions avec les piétons. (a) Véhicule 1 décide d'exécuter son virage à gauche entre les véhicules 2 et 3, mais, avec sa rapidité, ne voit pas le piéton qui traverse la route. (B) Avec PRW, le conducteur du véhicule 1 est alerté du piéton traversant la chaussée. Véhicule 1 retarde alors son virage à gauche jusqu'à ce que le véhicule 3 et le piéton traversant l'intersection.

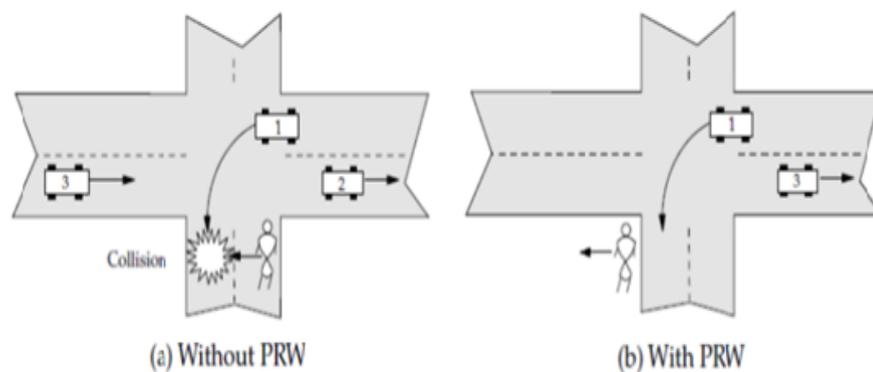


FIGURE 1.9 – *Alerte du piéton [21].*

1.2.5 Caractéristiques les réseaux VANETs

Les réseaux véhiculaires mobiles ont les caractéristiques intrinsèques suivantes [22] [3] [15] :

- *Ils sont très dynamiques* : En effet le réseau doit pouvoir gérer des véhicules allant de 0 à 160 km/h. Ainsi si deux véhicules sont en sens inverse et qu'ils roulent tous les deux à des vitesses excessives, le temps d'interconnexion pourra être très court.
- *Leurs environnements sont très variés* : L'environnement peut être une route, une autoroute, ou une ville (beaucoup plus complexe). De plus une situation d'embouteillage peut mener à l'encombrement du réseau tandis qu'une route de campagne peut conduire à la disparition des liens du réseau surtout la nuit.
- *Leur mobilité peut être prédite et modélisée* : En effet la circulation sur une route ou une autoroute est facilement prévisible, car le véhicule doit en général rester sur la route. Connaissant la position actuelle du véhicule et sa vitesse, il est ainsi possible de prédire la position du véhicule à condition de connaître la carte de l'environnement.
- *Ils n'ont pas de problème d'énergie ou de stockage* : Les équipements du réseau étant installés dans le véhicule, il n'y a pas de problème lié à la durée de vie du réseau.

1.2.6 Technologies de transmission

Les réseaux véhiculaires mobiles sont en théorie capables d'utiliser un grand nombre de technologies de transmission, dont celles issues de la 3e génération (3G) et 4e génération (4 G) de téléphonie mobile, le Wi-Fi, le WiMax, et le DSRC/WAVE [3]. Les réseaux mis en place pour les véhicules peuvent utiliser une ou plusieurs de ces technologies. Nous allons décrire les principales technologies ci-dessous en terminant par la plus utilisée dans les VANETs : DSRC/WAVE [19].

- *3G et 4G*

Les VANETs utilisent les connexions ad-hoc. Afin de pouvoir se connecter au

réseau global, et donc à Internet, le réseau cellulaire mobile peut être utilisé. Ce dernier utilise un certain nombre de transmissions, parmi lesquelles : 3G, 4G. Ces technologies de transmission sont plus ou moins performantes et leurs débits théoriques dépendent fortement de la mobilité des utilisateurs. Ainsi puisque les réseaux véhiculaires mobiles sont hautement dynamiques, ces technologies sont peu adaptées. De plus la nécessité d'appartenance au réseau via une carte SIM et donc un forfait avec un opérateur n'est pas intéressante [19].

- ***Wi-Fi et WiMax***

Les technologies Wi-Fi et WiMax, à travers les standards 802.11 [19] et 802.16 [19] peuvent être utilisés dans les VANETs. Un protocole Wi-Fi, 802.11p [19], a été établi spécialement pour les VANETs, il est pris en compte par la technologie DSRC. De son côté WiMax permet une transmission portant sur une zone géographique étendue. Il reste donc intéressant comme moyen de communication entre les véhicules et les infrastructures afin de permettre un accès à Internet au réseau ad-hoc formé par les véhicules. Cependant WiMax n'est pas non plus idéal dans les situations de très grande mobilité. C'est pour quoi on pourra l'utiliser pour les infrastructures mais on préférera DSRC/WAVE pour les véhicules Cela peut poser des problèmes de mise en place les véhicules nécessitant un bloc de réception/transmission capable d'utiliser à la fois DSRC/WAVE et WiMax [19].

- ***DSRC/WAVE***

La technologie DSRC/WAVE est déjà utilisée pour certaines applications liées à l'automobile, tel que le paiement électronique des péages sans s'arrêter. Nous allons présenter en détail cette technologie. Les communications dédiées à courtes portées, DSRC, ont été créées spécialement pour les réseaux véhiculaires mobiles. Elles regroupent les standards WAVE ainsi que la norme 802.11p [19] qui est la norme Wi-Fi pour les réseaux véhiculaires mobiles. WAVE regroupe les standards IEEE 1609.x [19] (1,2,3 et 4) qui sont les standards des couches 2, 3, 4 et 7 du modèle Open System Interconnections (OSI). IEEE 802.11p [19] gère la couche Physique (couche 1) de ce même modèle. Ils sont développés spécialement pour les réseaux véhiculaires mobiles [19].

1.2.7 Travaux dans le domaine des VANETs

Les propriétés des réseaux véhiculaires offrent des challenges importants ce qui rend les VANETs s'ouvrent à plusieurs domaines de recherche dont nous citons les plus importants [20] :

1.2.7.1 Sécurité

La sécurité est un défi majeur ayant un grand impact sur le futur déploiement des réseaux véhiculaires ainsi que leurs applications. En raison de la sensibilité des domaines d'utilisation des VANETs, une intrusion d'un véhicule malicieux aurait des conséquences graves sur l'ensemble des véhicules interconnectés. C'est pour cette raison que beaucoup de travaux de recherche ont été réalisés pour développer un mécanisme de sécurité instituant les relations de confiance entre les nœuds communicants et garantissant le contrôle d'accès aux services [20].

1.2.7.2 L'accès au canal

Les réseaux véhiculaires utilisent des communications radio. Par conséquent il est important de concevoir des solutions spécifiques aux réseaux VANETs qui permettent d'apporter de la qualité de service et de gérer les priorités en résolvant les problèmes d'interférences radio, des problèmes de propagation à multi-trajets des ondes ainsi que les irrégularités électromagnétiques [20].

1.2.7.3 Localisation des véhicules

Si l'un des véhicules du réseau doit être localisé (dans le cas d'un accident par exemple), les autres doivent être informés de sa position. Le problème est que tous les véhicules ne sont pas équipés d'un système de repérage par satellite (GPS). Pour cette raison un mécanisme de localisation sans utilisation de GPS est nécessaire [20].

1.2.7.4 Problèmes de congestion

L'un des problèmes des VANETs est que chaque véhicule communique avec tous ceux qui sont dans sa zone de couverture. Ceci entraîne une dégradation de la qualité de service avec l'augmentation du nombre de véhicules ce problème a fait l'objet de plusieurs études.

1.2.7.5 Mobilité dans la simulation des réseaux

Dans la simulation des VANETs le facteur mobilité a longtemps été négligé. On ne considérait pas la différence de mouvements entre les nœuds des VANETs et des MANETs, ce qui pouvait biaiser les résultats de la simulation. Pour cette raison de plus en plus d'équipes de recherche s'intéressent à l'étude de la mobilité dans les VANETs Avec un bon simulateur plus le modèle de mobilité est réaliste plus les résultats de la simulation sont proches de la réalité. D'où l'impact direct des modèles de mobilité sur la réussite d'une simulation [20].

1.2.7.6 Routage

Le routage dans les réseaux VANETs est un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs. Pour que les véhicules puissent communiquer entre eux, un protocole de routage doit être défini. En effet quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules [20].

Conclusion

Dans ce chapitre, nous avons tout d'abord présenté les réseaux véhiculaires VANET qui sont devenus un domaine prometteur de la recherche une fois que le monde progresse vers la vision des systèmes de transport intelligents. Ensuite nous avons exposé les différents modes de communication dans les réseaux véhiculaires à savoir V2I, V2V et hybride qui servent à améliorer la sécurité routière en échangeant des messages entre les véhicules. Enfin nous avons cité les applications et les caractéristiques de réseau VANET.

Dans le chapitre suivant, nous présenterons le routage dans les réseaux VANETs, les types et de la classification des protocoles de routage, ainsi que des détails sur les protocoles spécifiques et leur fonctionnement.

Introduction

Le routage joue un rôle très important dans les VANETs puisque tous les services supportés, unicast ou multicast, se basent sur des communications multi-saut pour l'acheminement des données. Les transferts de fichiers et les jeux. Les communications multicast sont utilisées dans les applications de sécurité et de gestion de trafic telles que l'avertissement de collision. Pour réaliser les échanges, les protocoles de routage utilisent des informations locales sur le voisinage immédiat ou globales, concernant tout le réseau, de déterminer les nœuds relais qui participent à l'acheminement des données [4].

Ce chapitre est composé de deux parties. Dans la première partie nous allons focaliser sur les protocoles de routage les plus adaptés aux réseaux ad hoc de véhicules. Dans ce contexte, nous proposons une classification de ces protocoles en deux grandes catégories : Les protocoles de routage multipoints et géographiques (1ère catégorie). Ces protocoles peuvent être mis en œuvre dans la phase de la dissémination de requêtes Les protocoles de routage unicast pour le transfert de données volumineuses (2ème catégorie).

Dans la deuxième partie nous allons présenter le principe et le fonctionnement des protocoles les plus adaptés dans le réseau ad-hoc véhiculaire.

Première partie

Protocoles de routage dans les réseaux véhiculaires

2.1 Protocoles de routages dans les VANETs

Afin de présenter les principaux protocoles de routages dans les VANETs, nous avons choisi de commencer par la classification des différents protocoles existant dans la littérature. La partie suivante présente donc ces différentes méthodes de routages. En effet un grand nombre de protocoles de routage existent dans la littérature [13][6]. Nous n'avons donc pas pu tous les retenir. C'est aussi dans ce but que nous avons choisi de commencer par une classification.

2.1.1 Classification

La classification que nous vous présentons est illustrée ci-dessous .Cependant, elle est issue d'une première analyse du routage de l'information qui est la suivante : le routage de l'information consiste à diriger les messages du nœud source vers les nœuds destinataires, à travers le réseau. Les communications peuvent se faire de deux manières différentes, directement avec un seul saut (single-hop) ou par étape avec plusieurs sauts (multi-hop). Avec un seul saut, la source et la destination sont en fait connectées directement l'un avec l'autre, tandis qu'avec plusieurs sauts, la communication se fait via l'intermédiaire des nœuds positionnés entre la source et la destination.

Chaque fois que le message passe d'un nœud à un autre il se déplace d'un saut. Lors d'une communication multi-sauts, le chemin est choisi en fonction de divers facteurs tel que sa longueur, sa bande passante, ou encore sa durée de vie. Cela est valable pour n'importe quel type de routage qui sera présentés ci-dessous avec la FIGURE 2.1. Ensuite, il existe deux classes de protocoles de routage :

- Routage Proactif : Les nœuds maintiennent en permanence des tables de routage afin de connaître la topologie du réseau en tout temps ; qu'un nœud voisin en fait la demande ou qu'il doit démarrer une transmission. Ils ne connaissent donc pas la topologie du réseau lorsqu'ils sont inactifs.
- Routage Réactif : Les nœuds construisent une table de routage seulement lors qu'on peut noter que des techniques de type hybrides existent, par exemple, le protocole ZRP (Zone Routing Protocol) [8]. Ce protocole crée des groupes d'utilisateurs. Le routage se fait de façon réactive au sein du groupe, et il se fait de façon proactive entre les différents groupes. Ces classes de protocoles

coexistent au sein des différents types de routage.

2.1.2 Différents types de routages

- Routage Unicast : l'information transite d'un nœud source vers un nœud destinataire.
- Routage multicast : le multicast (qu'on pourrait traduire par « multidiffusion ») est une forme de diffusion d'un émetteur (source unique) vers un groupe de récepteurs. Les termes « diffusion multipoint » ou « diffusion de groupe » sont également employés.
- Routage de type Broadcast : l'information du nœud source est envoyée à tous les nœuds présents autour de ce dernier.
- Routage de géocast : c'est une technique de routage de paquet sur un réseau Manet qui a pour but de transmettre des données à l'ensemble des nœuds situés dans une zone géographique donnée. Il a principalement été développé pour les réseaux MANET, dans lesquels chaque nœud peut être un routeur et chaque nœud peut se déplacer changeant ainsi très fréquemment la topologie du réseau.

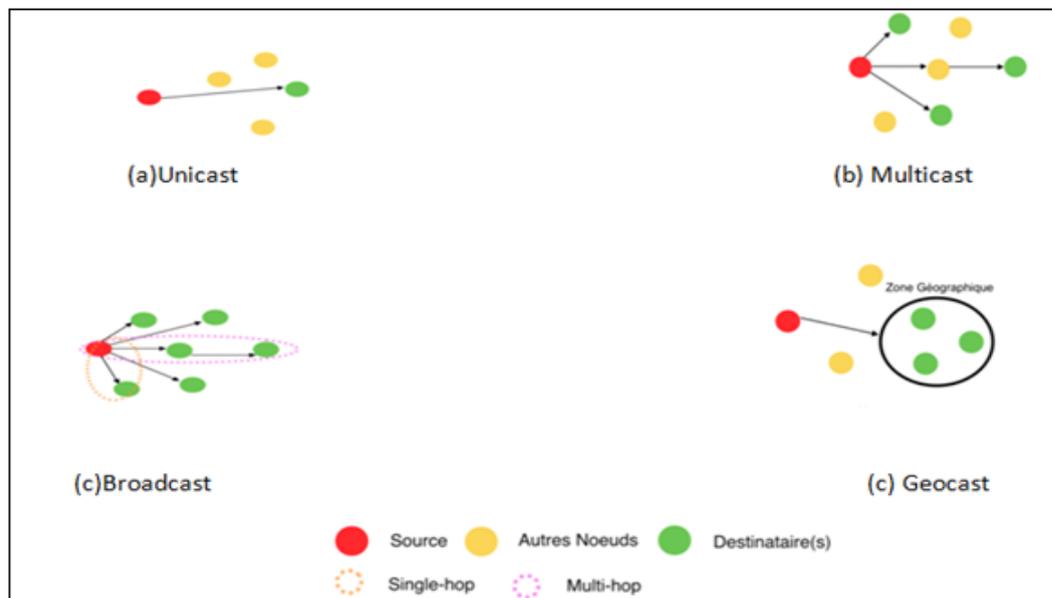


FIGURE 2.1 – Différents types de routages [13].

Ainsi, ils ont été divisés les différents protocoles de routages dans les VANETs en trois grands types de routage (voir la FIGURE 2.2), et nous avons présenté aux

protocoles unicasts, en fonction des spécificités des réseaux véhiculaires mobiles et des protocoles qui ont été créés pour eux, trois sous-catégories : les protocoles Ad-hoc, les protocoles basés sur la position et les protocoles de groupes. Dans toutes ces catégories de routage, la source a été définie unique vers un ou plusieurs destinataire (one-to-many). Cependant, il est possible dans certains cas qu'un nombre multiple de nœuds sources émettent des paquets de données à plusieurs destinations (many-to-many).

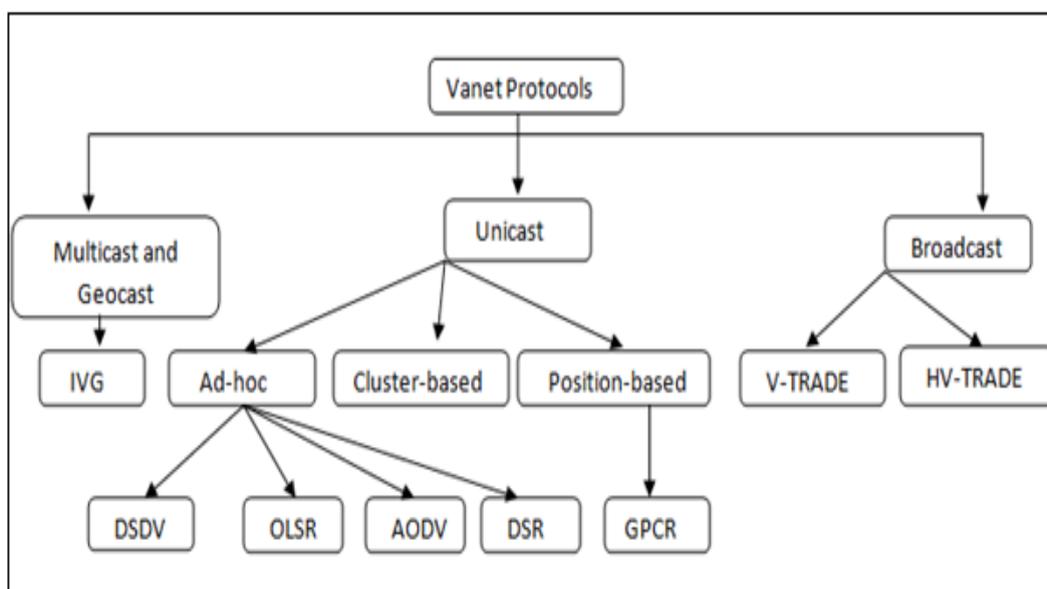


FIGURE 2.2 – Classifications des protocoles de routages dans les VANETs [13].

2.1.2.1 Protocoles Unicast Ad-hoc et leurs adaptations pour les VANETs

Dans la littérature, quatre protocoles reviennent de façon récurrente lorsque l'on parle de réseau ad-hoc mobile. Deux sont proactifs, Dynamic destination-Sequenced Distance Vector (DSDV) [17] et Optimized Link State Routing Protocol (OLSR) [9], et les deux autres sont réactifs, Ad-hoc On-Demand Distance Vector Routing (AODV) [16] et Dynamic Source Routing (DSR) [10].

DSDV : Utilisation de l'algorithme de Bellman-Ford afin de trouver le chemin optimal du nœud source au nœud destinataire. La mise à jour des tables se fait en fonction du temps (de façon périodique grâce à des timers), ou en fonction d'événement (mauvais débit de connexion, délais trop élevés, liens qui disparaissent) [17].

OLSR : Ce protocole définit des nœuds qu'il nomme «multi-relais», seuls ces nœuds possèdent les tables de routage complètes. Cela réduit la diffusion excessive et le gaspillage de la bande passante par rapport au protocole DSDV [9].

AODV : Ce protocole a été développé pour les MANETs. Lorsqu'un nœud doit commencer une transmission, AODV diffuse un paquet de découverte de façon broadcast. La destination va utiliser le chemin emprunté par le premier paquet de découverte qui l'a atteint. Il envoie alors un paquet réponse afin d'annoncer ce chemin à la source. L'envoi de données peut alors commencer. En cas de disparition d'un lien sur le chemin choisi, un paquet d'erreur est généré [16].

DSR : Reprends le principe de DSDV (notamment l'algorithme de Bellman-Ford) mais construit les tables de routages de façon réactive, et se basent sur l'état des liens. En effet, la modification de la table se fait seulement lorsque l'état des liens varie [10].

2.1.2.2 Protocoles Unicast de groupe

Le routage par groupe aussi appelé «Cluster-based Routing» est une technique efficace sur les autoroutes. Elle consiste à créer des sous-réseaux virtuels tels que la montre la FIGURE 2.3 [13]. Chaque groupe élit à sa tête un nœud header qui va s'occuper de gérer la coordination et le management de communications inter et intra-groupes.

Dans un groupe les communications ne peuvent être que directes (1-hop).

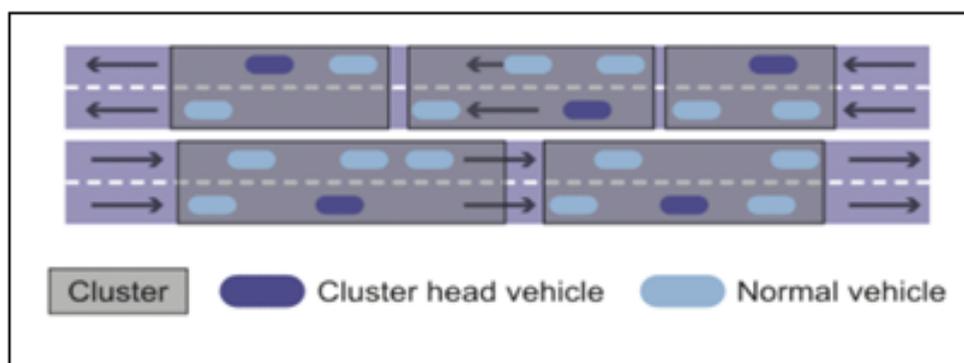


FIGURE 2.3 – Cluster-based Routing [13].

2.1.2.3 Protocoles Unicast basé sur la position

L'idée de protocole basé sur la position pour les VANETs vient naturellement lorsque l'on considère leurs spécificités. En effets, les déplacements des véhicules sont limités par les routes et l'environnement. On peut donc sans trop de difficulté effectuer un graphe en deux dimensions en utilisant les données de positionnement des véhicules afin de trouver le meilleur chemin de routage de l'information entre la source et la destination [10].

Le protocole basé sur la position le plus connue est GPCR (Greedy Perimeter Coordinator Routing), il place des nœuds appelés «Coordinateurs» aux intersections. Ainsi le message, une fois arrivé aux abords de l'intersection, n'est plus envoyé de façon gourmande, mais au coordinateur. Cela permet d'éviter un grand nombre de situations de minimums locaux rencontrés avec GPSR. On remarque sur la FIGURE 2.4 [10], que N3 est un minimum local du protocole GPSR. Grâce à GPCR, le message est envoyé à C1, le nœud coordinateur, Qui lui n'est plus un minimum local.

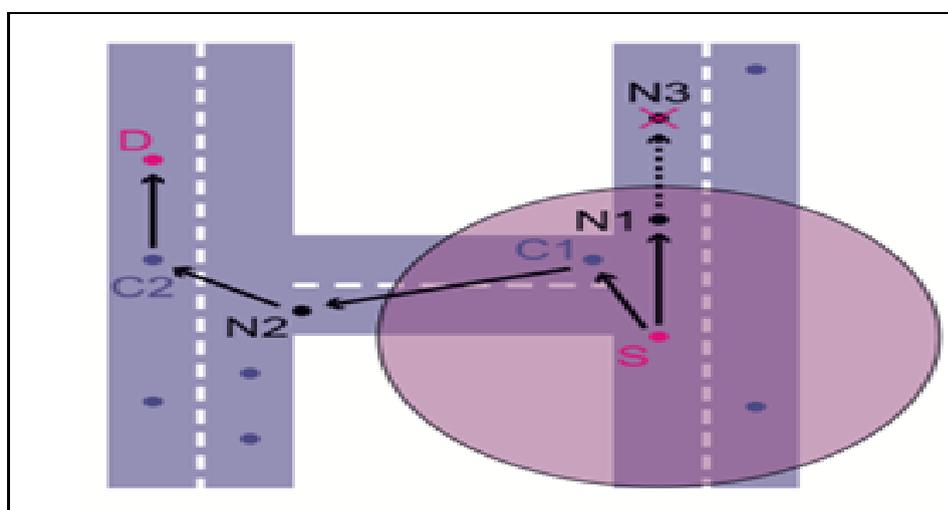


FIGURE 2.4 – GPCR Routing [10].

2.1.2.4 Protocoles Broadcast

Les protocoles basés sur la diffusion (broadcast) ne sélectionnent plus un nœud en particulier à qui transmettre l'information. Elle est envoyée du nœud source vers tous les nœuds du réseau. Cette méthode est beaucoup utilisée dans les VANETs, et utilise le plus souvent la méthode d'inondation («flooding») [13]. Cette méthode

consiste à envoyer le message à tous les nœuds qui sont à portée de transmission, et chaque nœud réitère l'opération pendant un temps donné. L'inondation est efficace dans des réseaux de petite taille, mais mène rapidement à une surcharge du réseau lorsque celui-ci est grand.

On peut citer le Vector-based TRACKing Detection (V-TRADE) et History-enhanced V-TRADE (HV-TRADE) [13], utilisent la position ainsi que le mouvement des nœuds. Ils divisent le réseau en plusieurs petites zones et sélectionnent ensuite les nœuds aux frontières de ces zones de transmission, puis les inondent en rapportant le message au centre.

2.1.2.5 Protocoles Multicast et Geocast

Les protocoles Multicast / Geocast diffusent le message, non plus d'une source vers un destinataire (Protocole Unicast), ou plusieurs destinataires (Protocole Multicast), mais d'une source vers une destination géographique. La zone desservie peut aller de quelques mètres à plusieurs kilomètres. Là encore, la plupart des protocoles tentent de réduire la surcharge du réseau causé par l'inondation [13]. Nous présentons ci-dessous le protocole qui utilise ce principe.

Le protocole Inter Vehicular Geocast (IVG) [13] met en place un temps d'attente avant de rediffuser les messages. Si durant ce temps d'attente il ne reçoit pas à nouveau ce même message, alors il le transmettra aux nœuds qui l'entourent. Ce temps d'attente est proportionnel à la distance entre le nœud dont il a reçu le message et lui-même. De plus, il limite le nombre de sauts qu'un message peut effectuer.

Deuxième partie

Études des Protocoles AODV, OLSR et GPCR

Dans cette partie nous allons présenter les fonctionnements des protocoles AODV, OLSR et GPCR.

2.2 Principe fonctionnement de protocole OLSR

Le protocole OLSR (Optimized Link-State Routing) est une version optimisée du principe de routage à état de liens, destinée aux réseaux sans fil. Les problèmes du principe de routage à état de liens est le risque de congestion lié à l'échange de messages. Le principal objectif de ce nouveau protocole est de limiter cet échange de messages tout en garantissant que chaque nœud ait en sa possession les informations suffisantes pour choisir les meilleurs next-hops [9].

2.2.1 Processus de construction des routes dans OLSR

Les protocoles proactifs sont caractérisés par le fait que les nœuds tiennent à jour une table de routage de l'ensemble du réseau. Chaque nœud envoie périodiquement des messages sur l'ensemble du réseau afin d'y informer des variations de topologie [9].

Le protocole OLSR propose de sélectionner pour chaque nœud du réseau un sous-ensemble de voisins qui se chargeront de retransmettre les messages de contrôle de topologie reçus appelé MPR. Les autres nœuds, qui ne font pas partie de ce sous-ensemble devront uniquement traiter l'information reçue sans la retransmettre [6].

De plus le nombre de voisins déclarés dans chaque message est limité. Grâce à ces optimisations, la charge transitant sur le réseau est sensiblement réduite. Le processus de construction des routes dans OLSR passe par les étapes suivantes [9] :

- a) Découverte du voisinage
- b) Élection des relais multipoints
- c) Distribution de la topologie
- d) Calcul des tables de routage

A) **Découverte du voisinage**

La découverte du voisinage se fait par l'intermédiaire de messages HELLO. Périodiquement, chaque nœud broadcaste un message HELLO contenant l'ensemble de ses voisins unidirectionnels et l'ensemble de ses voisins choisis comme MPR. Ces messages ne doivent jamais être retransmis par les nœuds qui les reçoivent. Les messages HELLO ont deux rôles :

- Déterminer le type de liens existant entre un nœud et ses voisins (lien symétrique ou asymétrique)
- Déterminer les voisins qui serviront de MPR.

B) **Sélection des relais Multipoint**

Les relais multipoints sont des nœuds chargés de la retransmission des messages de contrôle de topologie. Tout nœud n du réseau doit choisir un ou plusieurs de ses voisins comme MPR de telle sorte que tous les nœuds situés à deux sauts du nœud d'origine n soient accessibles via un de ses MPR. Chaque nœud choisi comme MPR ajoute le nœud n à sa liste de sélecteurs MPR. La principale difficulté réside dans le choix des MPR [9].

La FIGURE 2.5 montre la différence entre la diffusion classique et la diffusion optimisée à travers les MPR. Par exemple, dans le cas d'une diffusion classique, un message TC diffusé par le nœud source S , a besoin de 24 retransmissions de ce message pour atteindre tous les nœuds à 3 sauts (voir la FIGURE 2.5 (a)). Par contre, en utilisant les relais multipoints MPR (la FIGURE 2.5 (b)), il suffit de retransmettre ce message 12 fois pour atteindre les nœuds à 3 sauts.

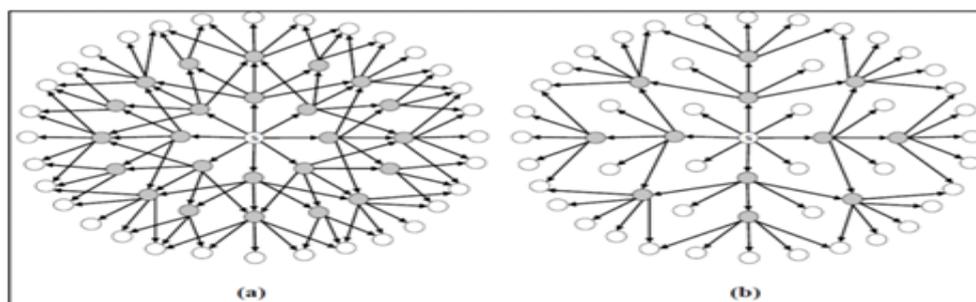


FIGURE 2.5 – diffusion classique (a) et diffusion optimisé à travers les MPR (b) [9].

C) Distribution de la topologie

La distribution de la topologie du réseau se fait par l'intermédiaire des messages TC (Topology Control). À intervalles réguliers, chaque MPR broadcaste un message TC contenant l'ensemble de ses voisins qui l'ont choisi comme MPR. Chaque message contient également un numéro de séquence pour reconnaître les messages périmés. Seul les voisins MPR rediffusent le message TC reçu pour éviter l'inondation, les nœuds n'ayant pas été choisis MPR ne génèrent pas le message TC. Les messages TC ont 2 rôles :

- Distribuer la topologie du réseau.
- Détecter les changements de topologie du réseau.

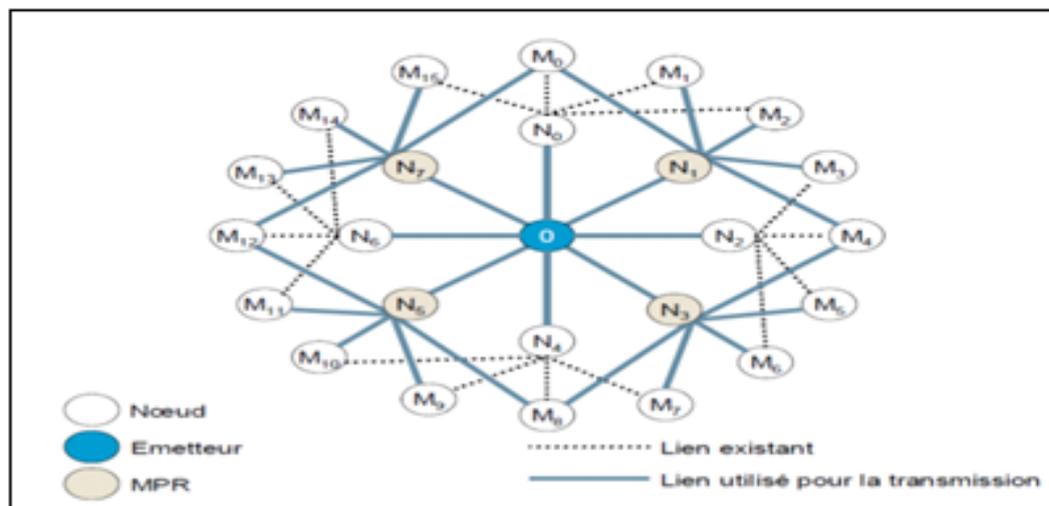


FIGURE 2.6 – Diffusion de messages avec MPR [9].

D) Calcul de la table de routage

Chaque nœud dans le réseau calcule sa table de routage pour permettre l'acheminement des informations vers tous les autres nœuds du réseau, ce calcul est basé sur les informations de voisinages et de topologie [9].

2.3 Principe fonctionnement de protocole AODV

Le protocole AODV (Ad hoc On-demand Distance Vector) représente essentiellement une amélioration de l'algorithme DSDV dans le contexte réactif. Il est spécialement conçu pour les réseaux mobiles pour créer et découvrir les liaisons entre la source et la destination. Il est utilisé pour des routages unicast et multicast en utilisant des requêtes de type (route request / route reply). Il offre une adaptation rapide aux conditions dynamiques des liens [16].

2.3.1 Méthode de construction d'une route

Le protocole Ad-hoc On-Demand Distance Vector Routing (AODV) [16], utilise quatre types de messages afin de construire et contrôler l'état des liens lors d'une communication : RREQ, RREP, HELLO et RERR. Le message de requête (RREQ) est diffusé par le nœud source afin de découvrir le réseau. Le message de réponse (RREP) est initié par le nœud destination ou par un nœud qui connaît le chemin jusqu'à la destination afin de rapporter à la source les informations à propos du chemin choisi. Ce processus est illustré à la FIGURE 2.7.

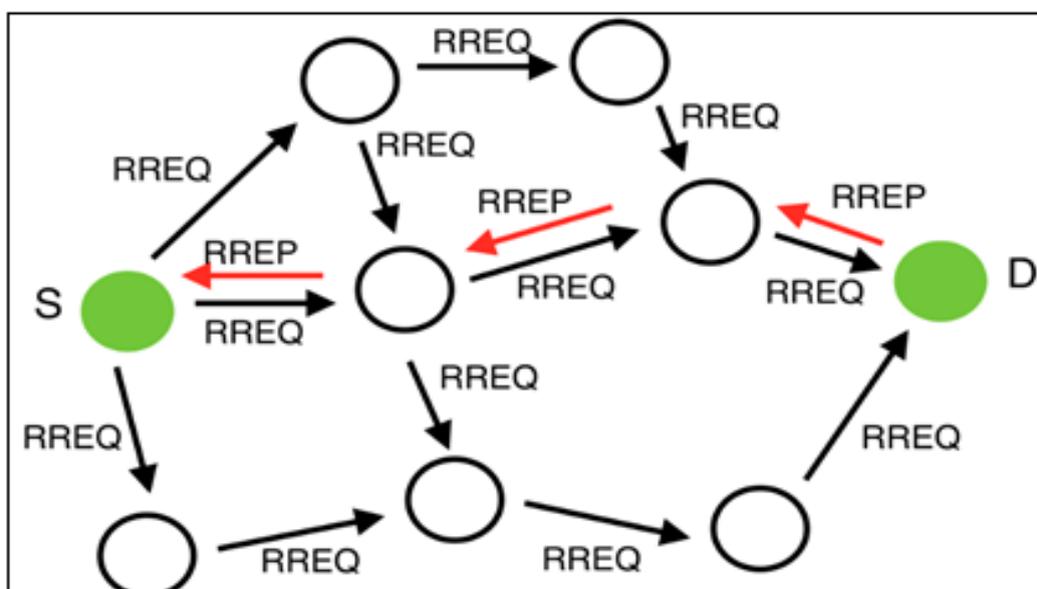


FIGURE 2.7 – Méthode de construction d'une route [16].

2.3.2 Maintenance de route

La maintenance des routes est réalisée en deux étapes. La première étape consiste en la détection de la perte d'une route. Quand un nœud sur une route établie se déplace, les routes qui passent par ce nœud peuvent être rompues. Les nœuds en amont, détectant la perte de connectivité grâce aux paquets HELLO, préviennent les sources affectées en émettant une requête d'erreur RERR. A la réception de ce paquet, le nœud source engage la deuxième étape de la maintenance des routes. Il entame une nouvelle phase de découverte des routes, si un chemin est toujours nécessaire [16].

Quand un nœud détecte une rupture de lien, il envoie une requête d'erreur RERR à toutes les sources affectées voir la FIGURE 2.8, si un nœud source reçoit une requête d'erreur, il peut décider de lancer à nouveau une découverte de route dans éventualité où il a toujours du trafic à envoyer.

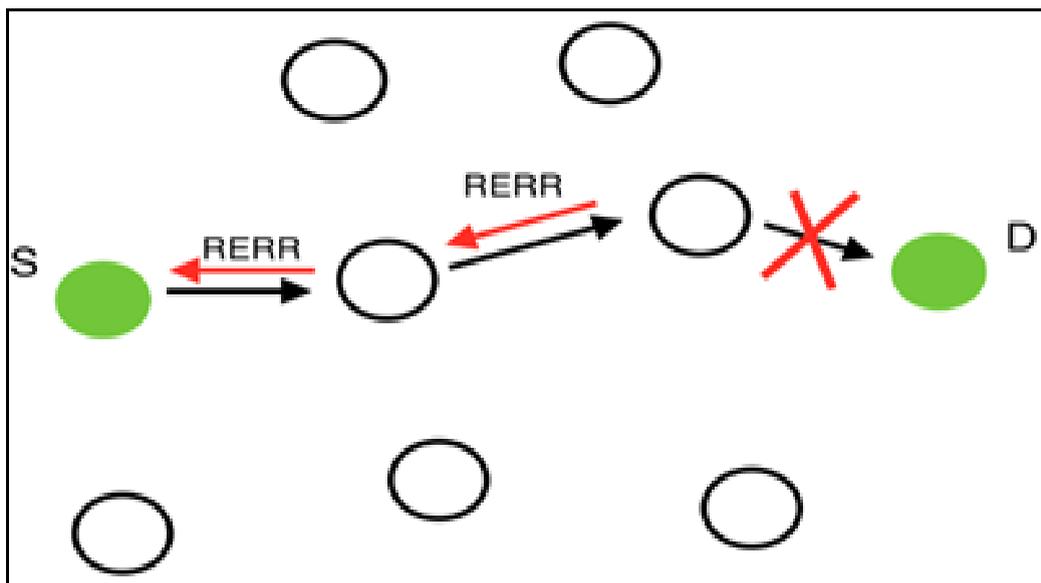


FIGURE 2.8 – Principe de détection d'un lien défaillant [16].

2.4 Principe fonctionnement de protocole GPCR

GPCR (Greedy Perimeter Coordinator Routing) est un autre protocole de routage unicast pour les VANETs. L'idée principale de GPCR est d'utiliser les routes et les intersections. Il consiste en l'utilisation dans un premier temps d'une procédure de greedy forwarding restreinte puis d'une stratégie de réparation basée sur la topologie réelle du réseau routier [10].

2.4.1 Routage greedy restreint

Une forme spéciale de greedy forwarding est utilisée pour acheminer le paquet d'information vers sa destination. Et comme les obstacles bloquent les signaux radio, les paquets sont sensés être acheminés le long des routes. Les intersections sont donc les seuls endroits où les décisions de routage sont prises. Par-contre, les paquets sont censés être relayés vers un nœud (véhicule) dans une intersection et non pas uniquement traverser l'intersection. Ceci est illustré dans la FIGURE 2.9 où le nœud u désire relayer un paquet à travers l'intersection au nœud 1a si la technique du greedy forwarding simple est utilisée. En relayant le paquet au nœud 2a, un autre chemin vers la destination pourra être trouvé. Les auteurs ont appelé un nœud situé dans une intersection, un coordinateur.

2.4.2 Stratégie de réparation

La stratégie de réparation de GPCR évite d'utiliser les graphes planaires par la prise de décisions de routage sur la base des routes et des intersections au lieu des nœuds individuels et de leurs connectivités. En conséquence, la stratégie de réparation de GPCR se scinde en deux parties :

- Dans chaque intersection, il est nécessaire de décider quelle route est censée suivre le paquet.
- Entre les intersections, le routage greedy vers la prochaine intersection peut être utilisé.

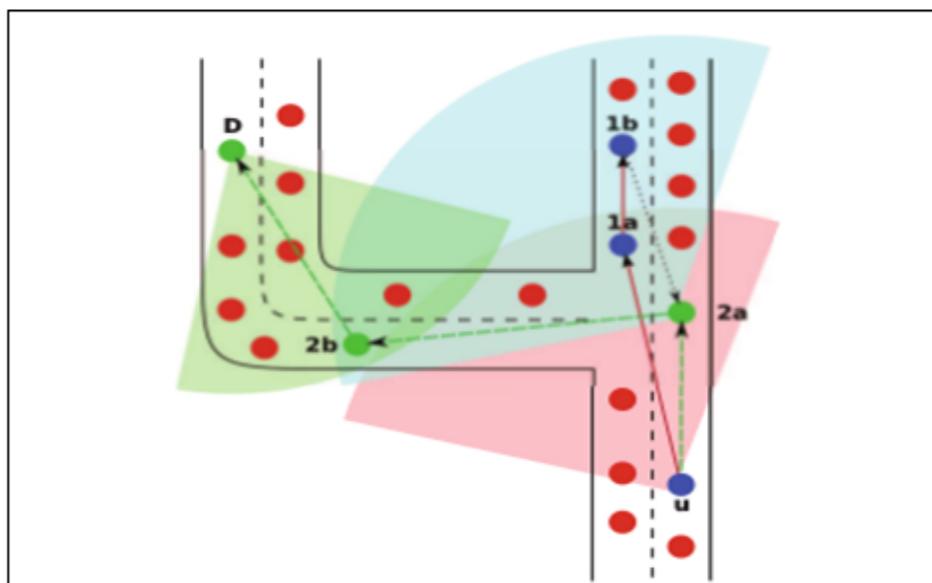


FIGURE 2.9 – *Routage greedy restreint* [10].

2.5 Comparaisons entre les protocoles OLSR, AODV et GPCR

Des études comparatives montrent que certains protocoles sont plus performants que d'autres selon les caractéristiques du réseau. La TABLE 2.1 suivant qui montrent les différents Avantages et les Inconvénients pour les trois protocoles OLSR, AODV et GPCR [1].

Protocoles	Avantages	Inconvénients
AODV	<ul style="list-style-type: none"> – Trafic de contrôle faible – Adaptés aux grands réseaux – Consommation d'énergie réduite 	<ul style="list-style-type: none"> – Temps de réaction long – Problème en cas de forte mobilité
OLSR	<ul style="list-style-type: none"> – Adaptés aux réseaux à forte mobilité – Adaptés aux réseaux denses de taille moyenne 	<ul style="list-style-type: none"> – Capacité d'échange du réseau limitée – Consommation d'énergie plus importante
GPCR	<ul style="list-style-type: none"> – La position des nœuds est connue – Limitation du trafic 	<ul style="list-style-type: none"> – Utilisation d'un GPS – Ressources CPU et mémoire

TABLE 2.1 – Tableau comparatif.

Conclusion

Dans ce chapitre nous avons présenté selon les différents critères de la classification des protocoles et leur types de routage dans les VANETs, nous avons constaté que les différentes caractéristiques supportées dans les VANETs comme la forte mobilité qui entraîne une topologie très dynamique, la détermination des routes stables entre les sources et les destinations des données. En effet, cela nous a permis de conclure que le choix de protocole de routage dépend de plusieurs contraintes.

Enfin, nous nous sommes basés sur certains protocoles de routage qui sont de base défini dans les MANETs ainsi déployé dans les VANETs, à savoir OLSR , AODV et GPCR Dans le prochain chapitre, nous présenterons l'évaluation des performances des protocoles de routage dans les VANETs.

Introduction

L'évaluation des protocoles développés pour les réseaux est indispensable avant de pouvoir les déployer dans les routeurs et les terminaux d'un réseau réel. Cette évaluation est nécessaire à la fois pour vérifier un fonctionnement correct d'un protocole dans divers scénarios, mais surtout pour mesurer ses performances et les comparer à d'autres protocoles existants. Les performances d'un protocole sont très sensibles aux changements rapides de la topologie du réseau, comme dans un réseau fortement dynamique [11].

La simulation est couramment utilisée pour mesurer les performances d'un réseau informatique, et plus particulièrement les performances d'un protocole. Elle est nécessaire quand l'expérimentation est trop coûteuse et l'étude théorique trop complexe. Les simulateurs permettent d'étudier le comportement des entités des réseaux véhiculaires sans fil et d'évaluer les performances des protocoles conçus pour ces réseaux [11].

Dans ce chapitre nous allons présenter les trois simulateurs de trafic routier JOSM, eWorld, SUMO et le simulateur réseau NS3 comme des environnements de simulation de notre travail, ainsi les étapes de la simulation. Ensuite nous allons effectuer une comparaison des performances entre les protocoles de routage : AODV, OLSR et GPCR.

3.1 Présentation des simulateurs

L'étude des performances d'un protocole de réseau véhiculaire sans fil, fait appel à deux types de simulateurs : le simulateur de trafic routier et le simulateur réseau. Le simulateur de trafic routier permet de générer la mobilité des véhicules sur une carte. Le simulateur réseau modélise le comportement des différentes entités du réseau ; c'est-à-dire qu'il permet de gérer les interactions entre les différents nœuds du réseau [11].

Dans cette partie de réalisation, nous avons utilisé comme environnement de simulation du trafic routier : JOSM pour la récupération des données géographiques d'une partie de la carte de Bejaia (CNEP, Cité seghir), eWorld pour l'extraction du réseau routier de ces deux quartiers et sumo pour la visualisation et la simulation de ce réseau routier. Comme simulateur de réseau sans fil nous avons opté pour le simulateur NS3 qui offre de meilleures performances comparées aux autres simulateurs (NS-2 et OMNET++ voir la Tab 3.14 ANNEXE IV).

3.1.1 Simulateur de trafic routier

3.1.1.1 JOSM (Java Open Street Maps)

C'est une application Java qui permet d'exploiter les services de cartes libre OSM (OpenStreetMaps) [23]. La FIGURE 3.1 représente le JOSM qui permet la récupération, la modification et la mise à jour des cartes sur osm. Dans cette partie on a eu recours à JOSM pour l'extraction de la carte des deux quartiers de Bejaia (voir ANNEX I).



FIGURE 3.1 – JOSM.

3.1.1.2 eWorld.1.0.0

C'est un outil qui permet d'importer des cartes proposées par des fournisseurs tels que Open Street map, les enrichir avec des événements et les exporter vers un simulateur de trafic routier, dans notre cas SUMO. Voir FIGURE 3.2 qui représente l'eWorld [23].

Nous utilisons l'eWorld pour l'extraction du réseau routier pour les deux quartiers (CNEP et Cité seghir, voir ANNEX I).

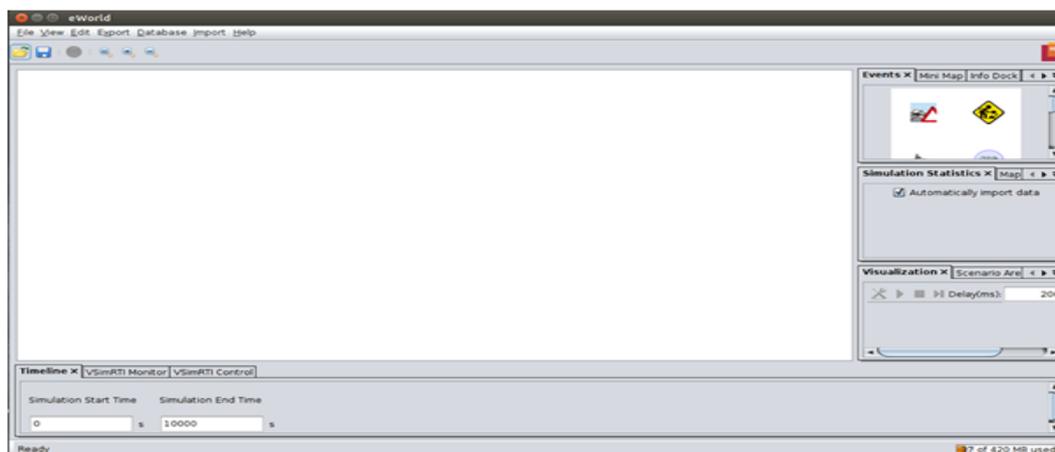


FIGURE 3.2 – *eWorld*.

3.1.1.3 SUMO.28 (Simulator of Urban Mobility.28)

SUMO (Simulation of Urban Mobility) est un logiciel de simulations de trafic routier open source sous licence ONU public (OPL), dont le développement a commencé en 2002. L'objectif des développeurs est de mettre à la disposition du monde académique un outil leur permettant de modéliser le réseau routier aussi bien en milieu urbain.

La FIGURE 3.3 montre le progiciel SUMO qui contient une suite d'applications qui aident à préparer et à exécuter la simulation d'un scénario de trafic routier. Il est très portable et nécessite uniquement l'installation des bibliothèques C++. Dans SUMO, chaque véhicule a son propre chemin, le comportement du véhicule est vivant comme le changement de voie. Les routes dans SUMO sont présentées sous forme de plusieurs voies. Il y a des intersections à base de règles de circulation et d'autres à priorité [23].

Nous choisissons le sumo pour la récupération des données géographiques d'une

partie de la carte de Bejaia (CNEP, Cité seghir).

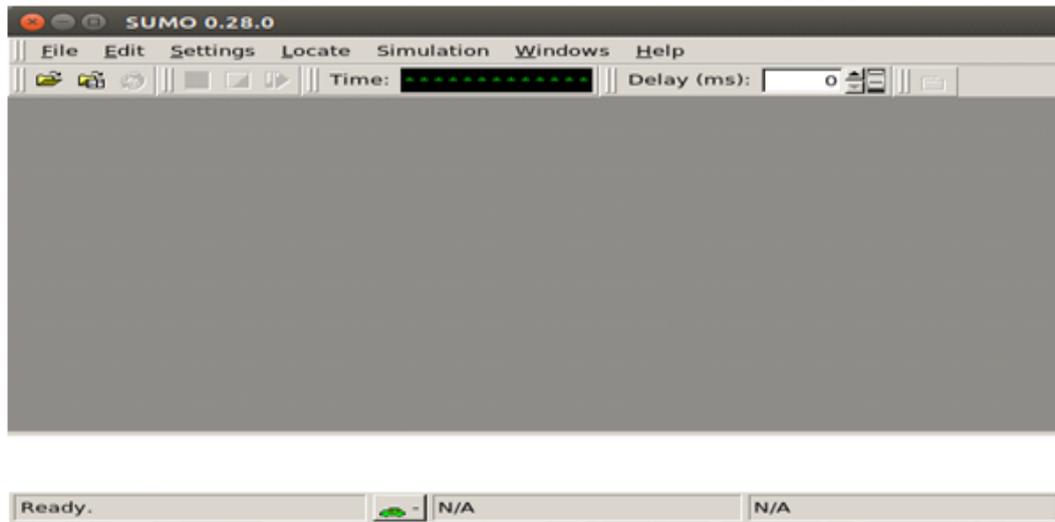


FIGURE 3.3 – *SUMO*.

3.1.2 Simulateur réseau

3.1.2.1 NS3.19(Network Simulator 3.19)

Un simulateur de réseau se compose d'une large gamme de technologies de réseau et de protocoles. Il est conçu pour aider les utilisateurs à créer des réseaux complexes à partir des classes représentant les modules de base pour construire un réseau. Nous avons choisi de faire notre simulation avec le simulateur de réseau NS-3 (Network Simulator 3). Les raisons de ce choix sont justifiées par la robustesse du simulateur NS-3. En effet, ce simulateur offre de meilleures performances en termes de rapidité de calcul et en temps de réponse. De plus, il garantit un passage à l'échelle jusqu'à 2000 nœuds (voir la FIGURE 3.4) [23].

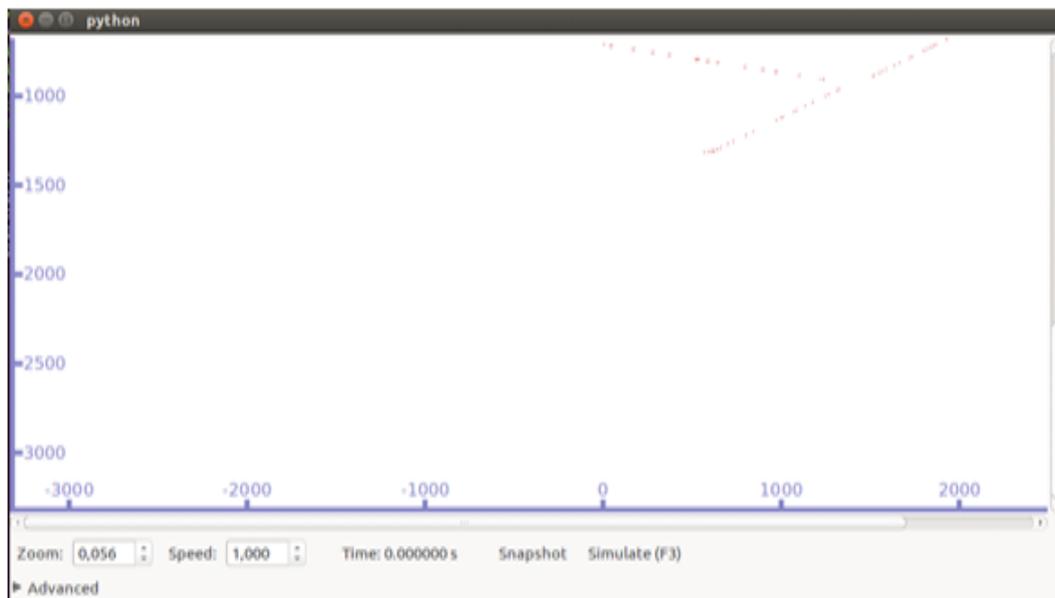


FIGURE 3.4 – NS3.

3.2 Intégration de GPCR à NS-3 et les problèmes rencontrés

Les protocoles OLSR et AODV font partie des modules (protocoles de routages) intégrés dans NS-3, alors que le protocole GPCR ne fait pas partie de ces modules. Dans ce qui suit, nous avons montré comment nous avons intégré le protocole GPCR. Dans un terminal, exécuter les commandes suivantes :

```
Sudo cd NS-3/ns-3.19/src  
Sudo ./create-module.py gpcr
```

Pour télécharger la dernière version de GPCR et du protocole de localisation-locationservice, il faut exécuter :`URL :https://github.com/fengyoulong/ns3-gpcr/tree/master/src` .nous avons copié les sous répertoires de `ns3/ns3.19/src/gpcr/*` dans le dossier crée précédemment avec `create-module.py` (dossier GPCR). Afin d'inclure les modules ajoutés dans la compilation, il faut exécuter les commandes ci-dessous :

```
Sudo ./waf -d debug --enable-examples--enable-  
tests configure
```

Toutefois, nous avons rencontré des erreurs lors de cette opération. Vous allez sans doute les rencontrer si vous utilisez la version **NS-3.19**. Par conséquent, nous procédons aux corrections de script dont le nom est `Wscript` qui se trouvent dans le répertoire `ns3/ns3.19/src/gpcr/wscript` comment suite :

◇ Il faut remplacer

```
headers=bld.new$\_ $task$\_ $gen(features=['ns3header'])
```

par

```
headers=bld(features='ns3header')
```

◇ Puis nous avons copié fichier `ipv4-l4-protocole.h` dans `ns3/ns-3.19/build/ns3`.

◇ Enfin on va supprimer la variable `addr` non utilisée du script `ns3/ns3.19/src/gpcr/model/gpcr.cc` puis nous avons exécuté la commande :

```
Sudo ./build.py
```

3.3 Etapes de la simulation

Le processus de simulation d'un VANET se fait en plusieurs étapes, illustrées à la FIGURE 3.5.

Dans un premier temps, SUMO nécessite des fichiers d'entrées au format xml. Ces fichiers sont réunis dans un fichier de configuration dont le format est propre à SUMO : (nom fichier.sumo.cfg). Le fichier de sortie est alors au format xml. Il est

ensuite traité par un programme en langage C++ (fourni par SUMO) qui permet de créer un fichier exploitable par NS-3 .

À la fin de la simulation avec NS-3, il y a deux fichiers de sortie : un fichier contenant la trace de toutes les communications entre les nœuds de la simulation(activity.tcl), et un fichier comportant les données de déplacement des nœuds durant la simulation(mobility.tcl), et pouvant être exécuté dans l'interface graphique de NS-3. Ce processus est détaillé dans les paragraphes suivants.

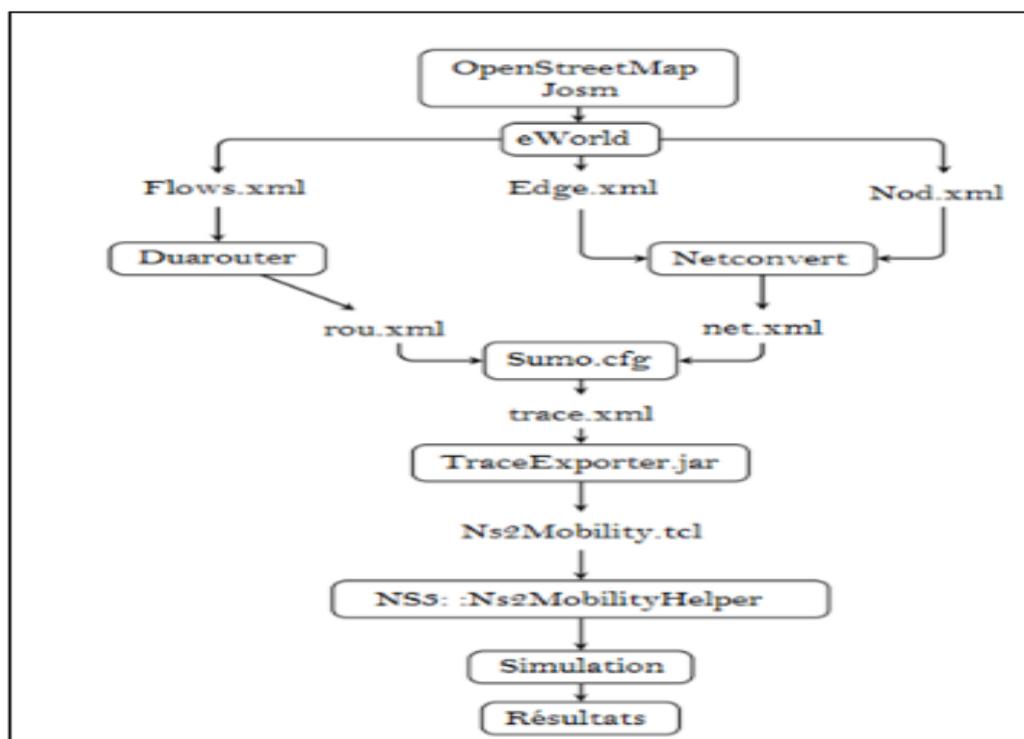


FIGURE 3.5 – Schéma récapitulatif des étapes de la simulation.

3.3.1 Simulation du trafic routier

Pour une évaluation cohérente des performances d'un protocole de routage pour les Réseaux véhiculaires, il est nécessaire de créer un scénario proche de la réalité basé sur un réseau routier déterminé. C'est dans cette optique que nous avons eu recours aux simulateurs présentés précédemment.

3.3.1.1 Préparation de la carte avec eWorld

Dans cette étape, nous avons importé une carte de la ville de Bejaia (la CNEP ,cité seghir) de l'éditeur de carte (JOSM) vers eWorld où nous avons configuré les points de départ des véhicules et leurs points d'arrivée. Nous avons précisé les types des véhicules (nœuds) et les intersections à feux de circulation et ceux à priorité (voir ANNEXE I).

3.3.1.2 Création des scénarios avec SUMO

Les fichiers .xml générés à partir d'eWorld qui sont nod.xml, edg.xml et fows.xml, contiennent les données du réseau routier, ces informations sont utilisées pour créer un scénario avec SUMO.

◊ Node file :

C'est un fichier composé d'un ensemble de nœud du réseau routier, ces derniers sont définis par les coordonnées x et y et un identificateur (id) FIGURE 3.6.



```

*final.nod.xml (-/Fin) - gedit
Ouvrir Enregistrer
<nodes>
  <node id="9654969f-d114-4d69-bfaf-9d3a29eaeae4" x="683332.2365218404" y="4068591.6260637203" type="priority" />
  <node id="4bc09c8d-ef41-483c-861d-a855ec4f6ce9" x="683895.4725838212" y="4068770.9231481142" type="priority" />
  <node id="0f181934-aab0-4e13-89af-456a0baf2f6a" x="683761.3624189183" y="4068703.542174601" type="priority" />
</nodes>
XML Largeur des tabulations : 8 Lig 6, Col 9 INS

```

FIGURE 3.6 – *Nod.xml*.

◊ Edge file : C'est un fichier qui relie les nœuds par des routes. Il est spécifié par un identifiant unique (id), un nœud source, nœud destination, priorité, nombre de voies et la vitesse maximal autorisée (voir la FIGURE 3.7).



```

<?xml version="1.0" encoding="UTF-8"?>
<edges>
  <edge id="97eb8be6-378b-4d23-b831-e5a489b0aefc" from="856b8219-19d6-46b4-9965-1e8cdd75eea" to="6202b901-f9b8-4665-b978-e6a5891c0700" priority="2" numLanes="1" speed="13.888888888888889" />
  <edge id="0006225e-5301-41b3-acee-8d1823917469" from="6202b901-f9b8-4665-b978-e6a5891c0700" to="856b8219-19d6-46b4-9965-1e8cdd75eea" priority="2" numLanes="1" speed="13.888888888888889" />
  <edge id="e07bef20-6927-4924-8162-754d3afddf65" from="bd74d335-5349-45fb-a525-65f828e375bb" to="c80d292b-7cde-468a-827d-519cdf358436" priority="2" numLanes="1" speed="13.888888888888889" />
  <edge id="43397e58-def9-401f-be9a-5035e22462d0" from="c80d292b-7cde-468a-827d-519cdf358436" to="eeba48e2-0706-42a2-a514-a9e00445f4a6" priority="2" numLanes="1" speed="13.888888888888889" />
</edges>

```

FIGURE 3.7 – *Edge.xml*.

◇ Net file :

C'est un fichier généré à partir de la commande NETCONVERT qui prend en argument les fichiers edg.xml et nod.xml. La commande de NETCONVERT permet de lire ces fichiers et de les convertir dans le format SUMO.

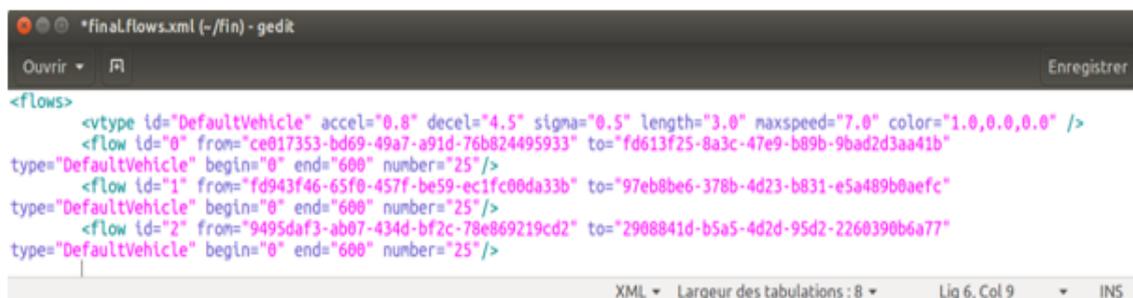
```

Sudo netconvert -n=nom-fichier.nod.xml -e=nom-fichier.edg.xml
-output-file=nom-fichier.net.xml

```

◇ Flows file :

C'est un fichier qui représente le flux des véhicules entre deux points nœud (à savoir une spécifiant le nœud de départ et le nœud d'arrivée, le nombre de véhicules, temps de début et temps de fin voir la FIGURE 3.8.



```

<flows>
  <vtype id="DefaultVehicle" accel="0.8" decel="4.5" sigma="0.5" length="3.0" maxspeed="7.0" color="1.0,0.0,0.0" />
  <flow id="0" from="ce017353-bd69-49a7-a91d-76b824495933" to="fd613f25-8a3c-47e9-b89b-9bad2d3aa41b" type="DefaultVehicle" begin="0" end="600" number="25" />
  <flow id="1" from="fd943f46-65f0-457f-be59-ec1fc00da33b" to="97eb8be6-378b-4d23-b831-e5a489b0aefc" type="DefaultVehicle" begin="0" end="600" number="25" />
  <flow id="2" from="9495daf3-ab07-434d-bf2c-78e869219cd2" to="2908841d-b5a5-4d2d-95d2-2260390b6a77" type="DefaultVehicle" begin="0" end="600" number="25" />

```

FIGURE 3.8 – *Flow.xml*.

◇ Route file :

C'est un fichier généré à partir de la commande DUAROUTER qui prend en

argument deux fichiers : net.xml et fows.xml. La commande DUAROUTER permet de calculer les itinéraires les plus rapides à travers le réseau.

```
duarouter -flows=--xx.flows.xml -net=xx.net.xml -output-
file=newxx.rou.xml.
```

◇ *Sumocfg file* :

C'est le fichier résultant des deux fichiers Route file et Net file, Sumocfg est donc le fichier de configuration de SUMO voir FIGURE 3.9.



```
final.sumo.cfg (~/Bureau/fln3) - gedit
Ouvrir Enregistrer
<configuration>
  <net-file value="final.net.xml"/>
  <route-files value="newfinal.rou.xml"/>
  <additional-files value="final.tls.xml;final.evt.xml;final.add.xml"/>
  <junction-files value=""/>
  <output-file value=""/>

  <simulation
    begin="0"
    end="600"
  />

  <reports
    verbose=""
    print-options=""
  />
</configuration>
Texte brut Largeur des tabulations : 8 Lig 1, Col 1 INS
```

FIGURE 3.9 – *sumo.cfg*.

3.3.2 Génération des fichiers traces de mobilité

L'étape de la création des fichiers trace est la plus importante afin de créer un scénario pour un simulateur de réseau, NS-3 dans notre cas. Générer un fichier trace implique l'utilisation de TraceExporter.py qu'on trouve par défaut dans sumo. L'exportation ce fait en deux étapes :

◇ En premier lieu nous allons créer un fichier trace pour sumo qui est sumoTrace.xml en utilisant la commande suivante :

```
sumo -c xx.sumo.cfg -fcd-output sumoTrace.xml.
```

◊ En deuxième lieu, nous convertissons le fichier trace généré dans la première étape en un format trace qu'on peut utiliser dans un simulateur réseau. Dans notre cas, nous avons besoin des fichiers tcl qui sont spécifiques à NS-2. La commande suivante renvoie trois fichiers de format tcl : **config.tcl**, **mobility.tcl** et **activity.tcl**.

```
Sudo Python traceExporter.py -i sumoTrace.xml --nxx.net.xml
--ns2activity-output=activity.tcl --ns2config-output=
config.tcl--ns2mobility output=mobility.tcl.
```

◊ Puis nous utilisons la commande qui permet la visualisation de la simulation FIGURE 3.10 d'un protocole :

```
Sudo ./waf --run GPCR --vis.
```

Ensuite nous étudions les performances des trois protocoles AODV, OLSR et GPCR en fonctions des quatre métriques de performances suivantes (le taux de livraison des paquets, le délai de bout en bout, nombre des paquets perdus et le débit). Ses métriques caractérisent un protocole de routage dans les réseaux VANETs. Nous avons fait trois scénarios dans lesquels nous varions des différents paramétrés (le nombre des nœuds, temps de pause, émetteur/récepteur) respectivement (voir TABLE.3.1).

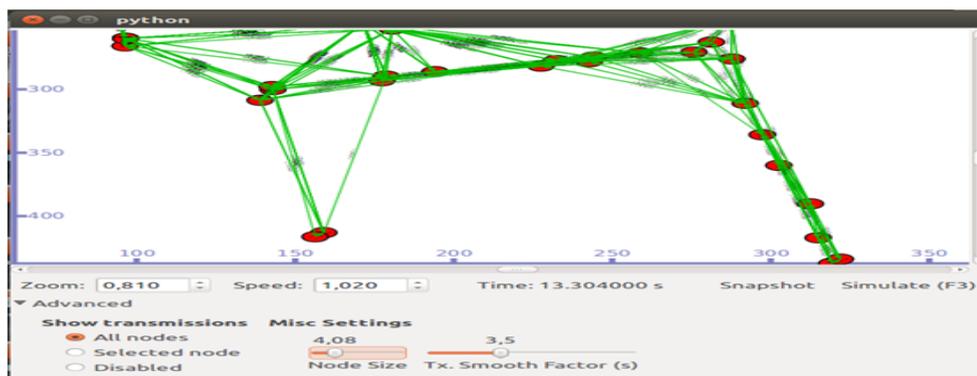


FIGURE 3.10 – Simulation de protocole GPCR.

3.3.3 Collecte d'informations

La simulation peut prendre quelques minutes pour GPCR et OLSR, voir des dizaines de minutes pour le protocole AODV. Ceci dépend de la taille du réseau et du protocole à simuler.

Une fois que la simulation d'un scénario est achevée, les résultats sont affichés à l'écran. Pour cela, nous avons défini deux fonctions :

- Fonction pour générer le trafic,
- Une autre pour faire des calculs.

3.3.3.1 Fonction de génération du trafic

Dans le code de l'application UdpClientServer, on a fait appel à la fonction `GenerateTrafic` pour envoyer les paquets. Dans cette fonction, on crée un paquet et on soustrait la taille de l'entête `SeqTsHeader`, car il est défini ainsi dans l'implémentation de la classe `UdpClient`. Voir le script `udp-client.cc` dans `.../src/applications/-model`, particulièrement la fonction `Send`. Ceci pour calculer le délai moyen de bout en bout. On a récupéré aussi le temps où le premier paquet est envoyé.

3.3.3.2 Fonction `ReceivePacket`

Cette fonction se déclenche à chaque fois qu'un paquet est reçu. Elle est constituée de deux étapes, la première consiste à extraire l'entête du paquet. La deuxième, permet d'obtenir les informations collectées de l'entête et des données et de faire les calculs ci-dessous :

- **Débit moyen**

Pour calculer le débit moyen, il suffit de récupérer la taille des données reçues avec la fonction `GetSize()` de la classe `racket` et le temps entre le premier et le dernier paquet reçu. Enfin appliquer la formule définie dans 3.4.4.

- **Délai moyen de bout en bout**

Pour calculer le délai moyen de bout en bout, il suffit de calculer les délais cumulés entre l'envoi et la réception de chaque paquet. Pour le faire, on utilise la fonction `Simulator::Now()` qui retourne le temps de l'évènement actuel et la fonction `GetTs()` qui retourne le temps d'encapsulation de l'entête `SeqTsHeader`.

- **Nombre de paquets reçus**

Pour calculer le nombre de paquets reçus, il suffit tout simplement d'incrémenter le nombre de paquets reçus dans la fonction `ReceivePacket()`. Pour les autres métriques d'évaluation, nous avons exploité les résultats précédents pour les calculer.

3.4 Critères d'évaluation

Notre objectif dans ce projet réalisé sous NS3 est d'analyser les métriques de performances des trois protocoles de routages OLSR, AODV et GPCR selon les métriques suivantes [1] :

3.4.1 Taux de livraison de paquets

C'est un facteur très important pour évaluer les performances d'un protocole de routage dans n'importe quel type de réseau. Ces performances dépendent des différents paramètres choisis pour la simulation. Les facteurs les plus importants sont la taille du paquet, le nombre de nœuds, la portée de communication et la structure du réseau. On peut obtenir le taux de livraison de paquet "PDR" (Packet Delivery Ratio) à partir de la somme de nombre de paquets reçus par le destinataire ce dernier divisé par la somme de paquets émis par tous les nœuds émetteurs.

$$\text{PDR} = \frac{\sum \text{nbr de paquets recus par la destination}}{\sum \text{nbr de paquets envoyés par tous les noeuds source}} \quad (i)$$

3.4.2 Délai de bout en bout

C'est le temps moyen pour qu'un paquet de donnée soit guidé par le réseau à sa destination. Il peut être obtenu en calculant les délais des messages de bout en bout transmis avec succès.

Le délai moyen de bout en bout comprend tous les retards possibles dans le réseau tampon c-à-dire la découverte de route de latence, les retards de retransmission à la MAC, et la propagation et délai de transmission. Mathématiquement il peut être démontré sous l'équation suivante (ii) :

$$D = \frac{1}{n} \sum_{k=1}^n (T_{ri} - T_{si}) * 100[ms] \quad (ii)$$

3.4.3 Paquets perdus

Ce sont les paquets qui n'ont pas pu atteindre leur destination. Cela est traduits mathématiquement par l'équation suivante (iii) :

$$D = \frac{\text{nbr paquets envoyés} - \text{nbr paquets recus}}{\text{nbr paquet envoyés}} \quad (iii)$$

3.4.4 Débit moyen

L'unité de mesure de cette métrique est paquet/unité TIL(TimeIntervalLenght), où TIL est la longueur de l'intervalle de temps. Cela est traduit mathématiquement par l'équation (iv).

$$\text{débit}_{\text{moyen}} = \sum_{k=1}^n \frac{\text{taille du paquet reçu}}{(\text{temps reception} - \text{temps emission})} \quad (iv)$$

3.5 Interprétation et évaluation des résultats

Pour notre simulation, les protocoles que nous avons simulés sont OLSR, AODV et GPCR. Pour évalués ces derniers, on revient sur les métriques citées dans 3.4. Ensuite, on applique les trois différents scénarios que nous avons définis, suivi des paramètres d'évaluations. Enfin, on interprète les graphes tracé à partir des résultats collectés lors de la simulation.

3.5.1 Scénarios

Le modèle de mobilité à utiliser est défini dans la carte du réseau routier de la ville de Bejaïa. L'application définie pour les deux premiers scénarios consiste à faire des échanges de paquets entre deux véhicules suivant le modèle client/serveur. Pour le troisième scénario, l'échange se fait entre plusieurs véhicules. Ceci pour tester d'éventuelles collisions. Les différents paramètres sont illustrés dans le TABLE 3.1.

Paramètres	Scénario 1	Scénario 2	Scénario 3
Nombre de nœuds	40, 60, 100 et 150	100	100
Temps de simulation	600 sec	1000 sec	600 sec
Portés de communication	80 m	80 m	80 m
Temps de pause	3 sec	1,2,5,8 sec	3 sec
Taille de paquet	1024 ko	1024 ko	1024 ko
Nombre émetteur/ récepteur	1/2	1/2	1/2,2/3, 4/5,7/8
Nombre de paquets de nœuds	100	100	100

TABLE 3.1 – Les paramètres utilisés dans les différents scénarios.

3.5.1.1 Premier scénario

Dans ce scénario, le nombre de nœuds connectés dans le réseau varie. Les résultats sont conclus lors de ces variations. Nous avons obtenu les résultats représentés dans les figures suivantes :

- **Taux de livraison des paquets**

Les résultats obtenus FIGURE 3.11 confirment que le taux de paquets délivrés par le protocole GPCR est plus élevés par rapport aux protocoles AODV et OLSR. En observant le graphe, on remarque que le PDR varie de la même manière pour les trois protocoles. Cela peut être à cause des nœuds intermédiaires qui coopèrent à la transmission des paquets au destinataire.

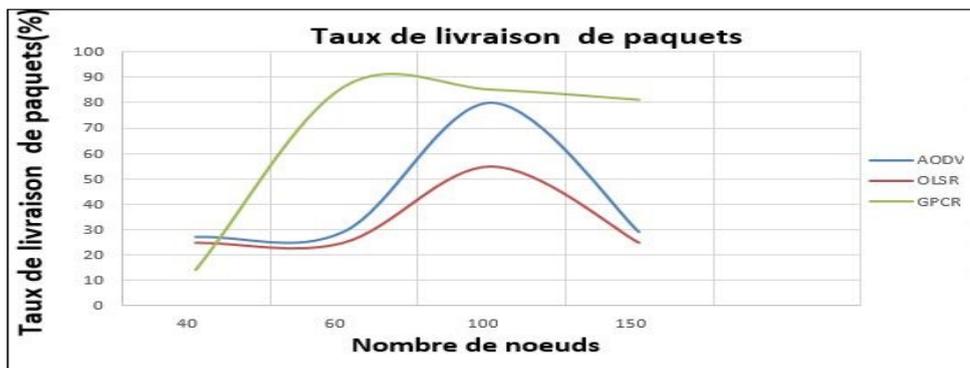


FIGURE 3.11 – Taux de livraison de paquets.

- **Délai moyen**

Les résultats obtenus FIGURE 3.12 confirment que les performances du protocole GPCR restent plus ou moins stables comme le montre sa courbe. Tandis que les performances d'OLSR et AODV augmente puis diminues en augmentant le nombre de noeuds. La courbe de GPCR reste en dessous des deux courbes AODV et OLSR, cela est expliqué par le fait que avant la transmission des paquets le protocole AODV fait appel à une procédure de découverte de route et le protocole OLSR entame son processus de sélection des noeud sMPRs.

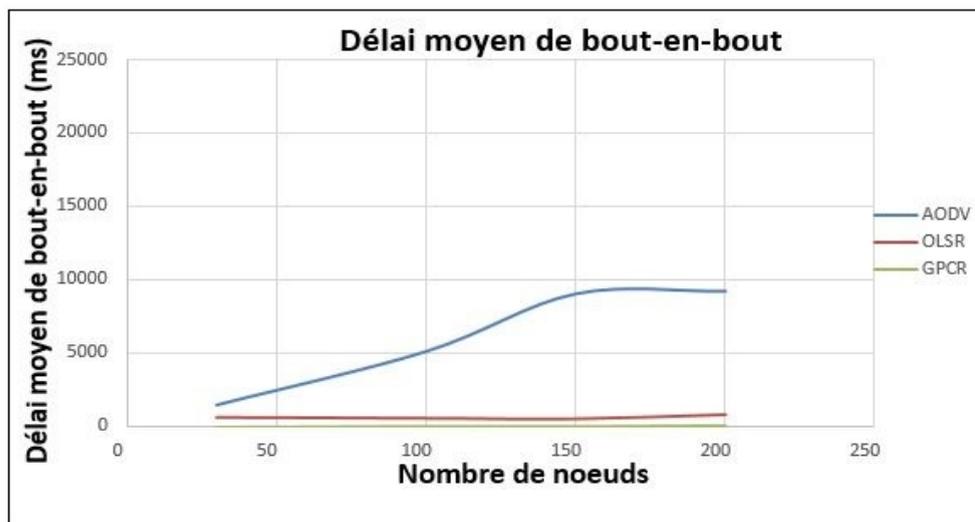


FIGURE 3.12 – Délai moyen de bout-en-bout.

- **Débit**

D'après la FIGURE 3.13 qui représente le débit en fonction du nombre de noeud, on constate que le débit de GPCR est nettement au dessus des deux autres protocoles AODV et OLSR.



FIGURE 3.13 – Débit en (bit/s).

- **Nombre de paquets perdus**

On constate que GPCR présente moins de perte de paquets que les protocoles AODV et OLSR FIGURE 3.14.

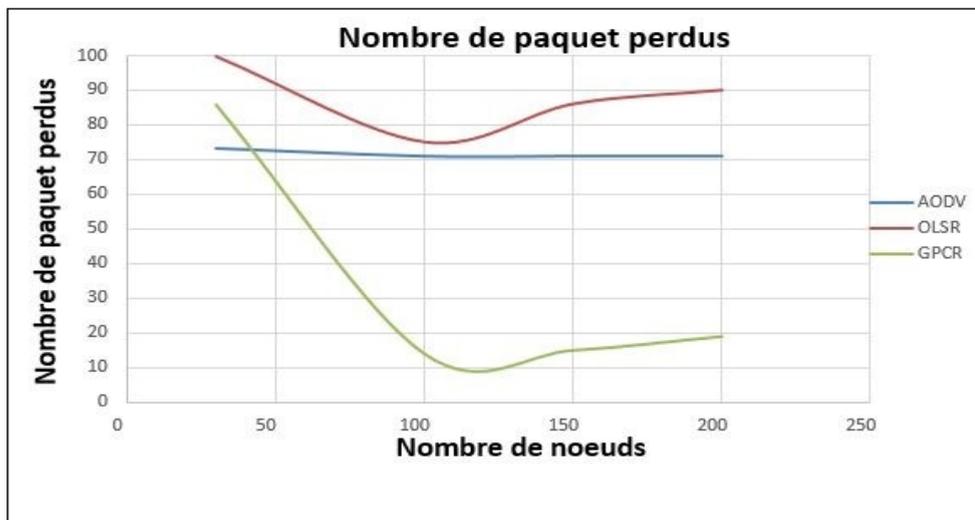


FIGURE 3.14 – Nombre de paquets perdus.

3.5.1.2 Deuxième scénario

Dans ce scénario le nombre total de véhicules dans le réseau est fixé et le temps de pause varie, Le temps de pause est l'intervalle de temps qui sépare deux transmissions de paquets successives. Nous avons obtenu les graphes suivants :

- **Taux de livraison des paquets**

Les résultats obtenus FIGURE 3.15 confirment que entre 3 et 4 second de temps de pause, les résultat d'OLSR sont élevés par rapport a AODV et GPCR en terme de taux de livraison de paquet. A partir de 5 seconde GPCR prend le dessus, avantageé par sa technique (géographique) de routage.

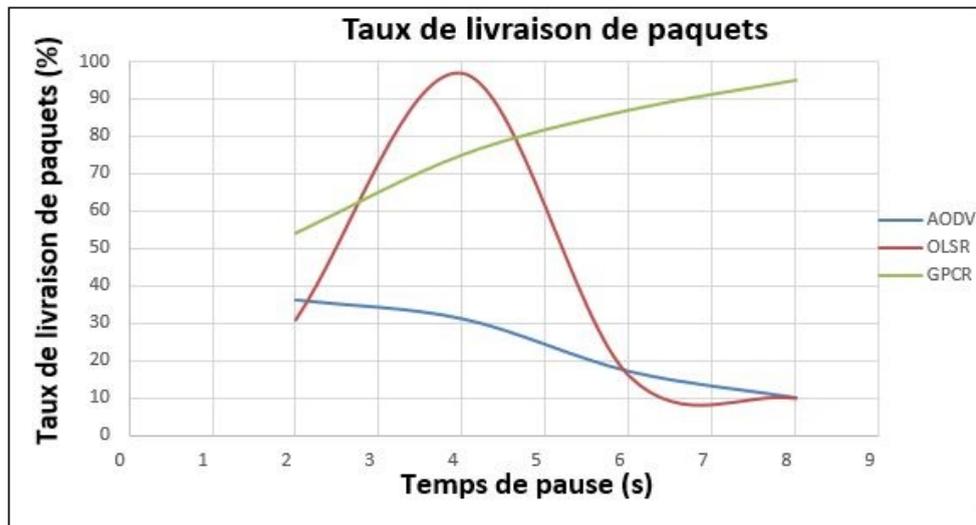


FIGURE 3.15 – Taux de livraison de paquets.

- **Délai**

D'après la FIGURE 3.16, le délai de bout en bout que présente GPCR reste stable dans l'intervalle de 2 à 8 ms. Les résultats d'AODV sont élevés au début de la simulation mais s'améliorent en augmentant le temps de pause avantageé par sa nature proactive (processus de sélection noeuds MPRs). OLSR présente un délai de bout en bout proportionnel à l'augmentation du temps de pause, pénalisé par sa nature réactive qui l'oblige à chaque transmission de procéder à une recherche de route.

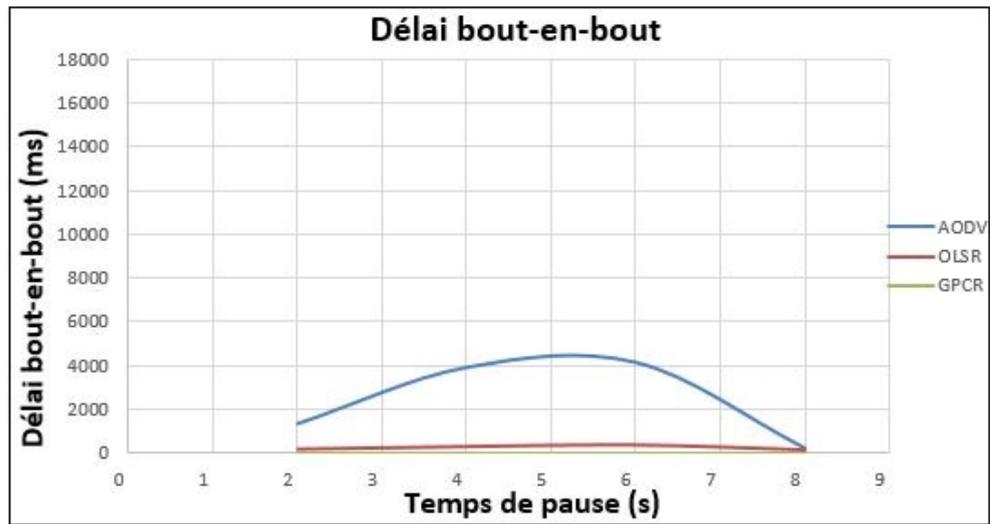


FIGURE 3.16 – Délai.

- Débit

Le graphe de FIGURE 3.17 représente le débit en fonction de temps de pause, on remarque que pour le temps de pause entre 2 et 8 second GPCR présente de meilleurs résultats que ceux des deux autres protocoles.

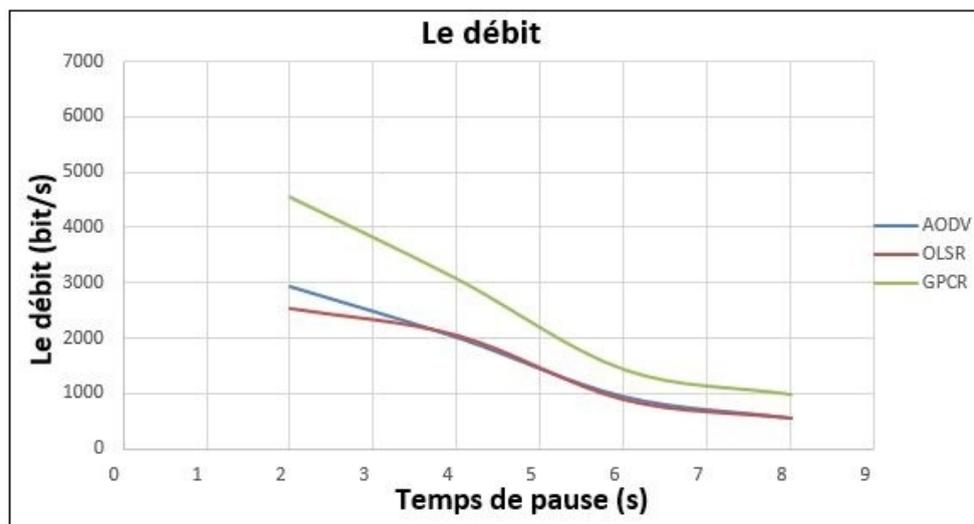


FIGURE 3.17 – Débit.

- **Nombre de paquets perdus**

Les résultats obtenus FIGURE 3.18 confirment que le GPCR présente moins de perte que le AODV et OLSR.

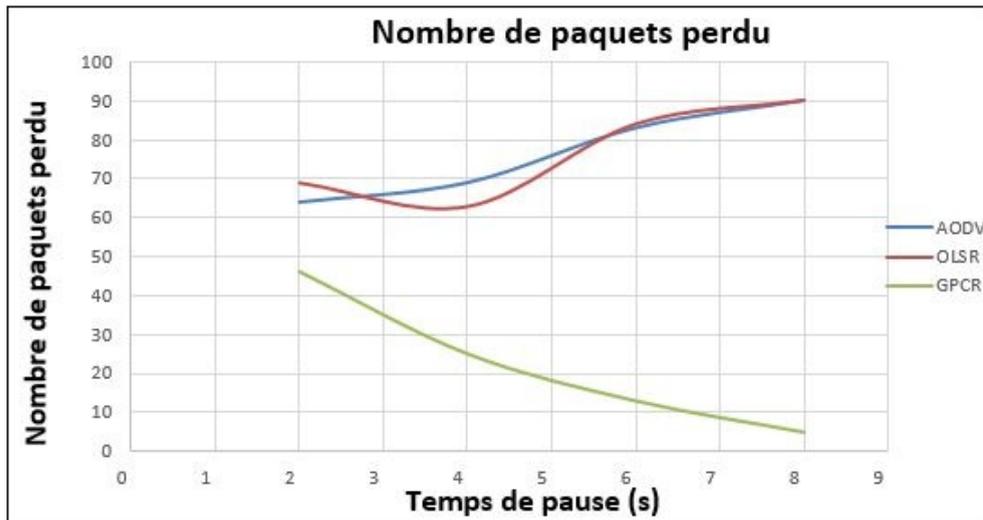


FIGURE 3.18 – Nombre de paquets perdus.

3.5.1.3 Troisième scénario

Dans ce scénario, nous proposons de varier le nombre émetteur/récepteur dans les protocoles.

- **Taux de livraison des paquets**

D'après la FIGURE 3.19 la courbe du protocole GPCR prend le dessus à partir de la première application (émetteur=1 et récepteur=2) par rapport aux deux autres courbes d'AODV et OLSR.

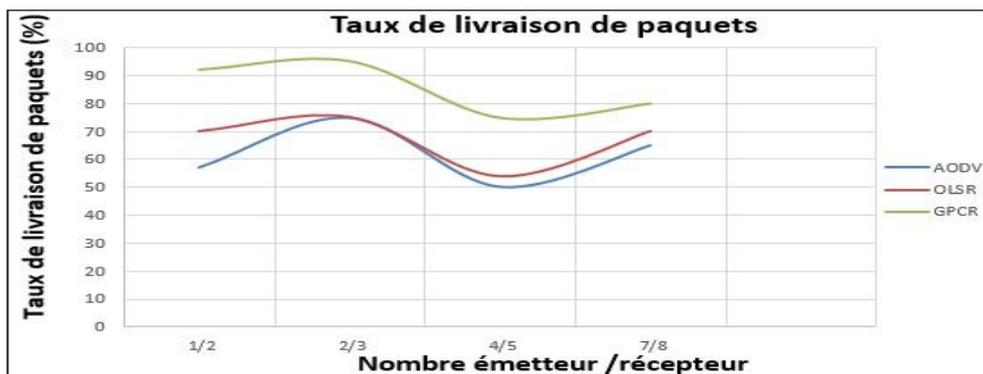


FIGURE 3.19 – Taux de livraison de paquets.

- Délai

On remarque que la courbe de AODV diminue de 8000 ms jusqu'à 1000 ms entre la première application (émetteur=1, récepteur=2) et la deuxième application (émetteur=2, récepteur=3), le protocole AODV augmente tandis que le OLSR se diminue. Par contre la courbe du GPCR reste stable durant toutes les applications FIGURE 3.20.

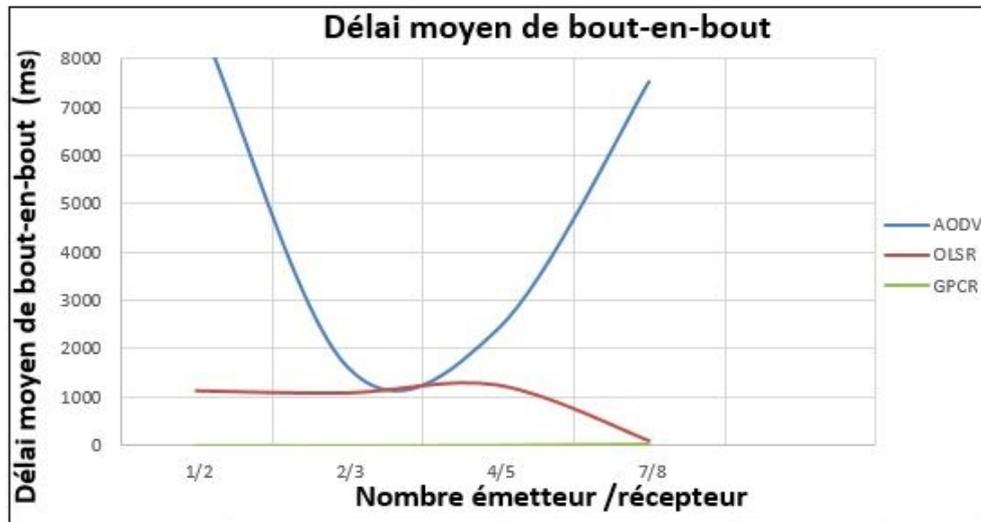


FIGURE 3.20 – Délai.

- Débit

A travers cette FIGURE 3.21, on remarque que GPCR présente une augmentation par rapport aux deux protocoles AODV et OLSR, à partir du nombre (émetteur=2 et récepteur=3), les courbes des trois protocoles diminuent puis augmentent mais toujours la courbe de GPCR se trouve au-dessus des deux autres.

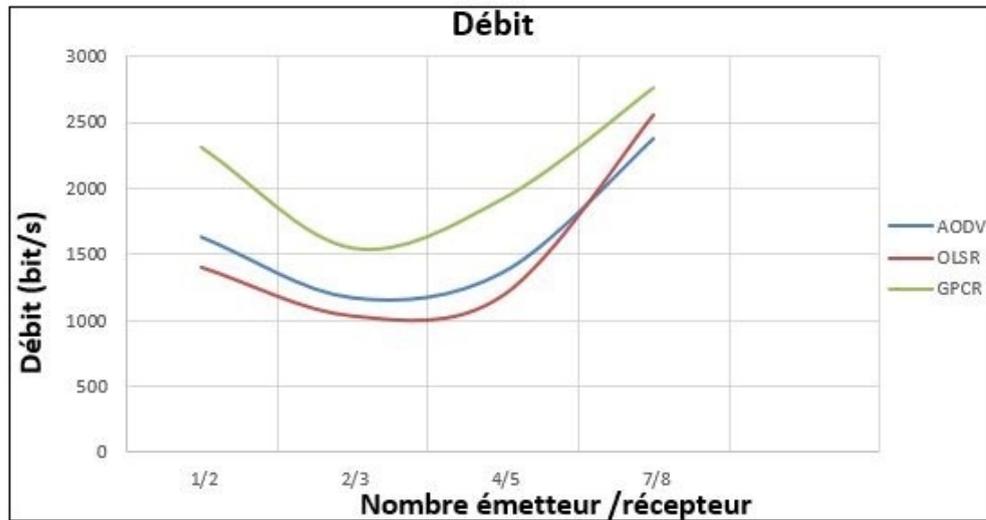


FIGURE 3.21 – Débit.

- Nombre de paquets perdus

On constate que GPCR présente des résultats sensiblement meilleurs que les protocoles AODV et OLSR dans la FIGURE 3.22.

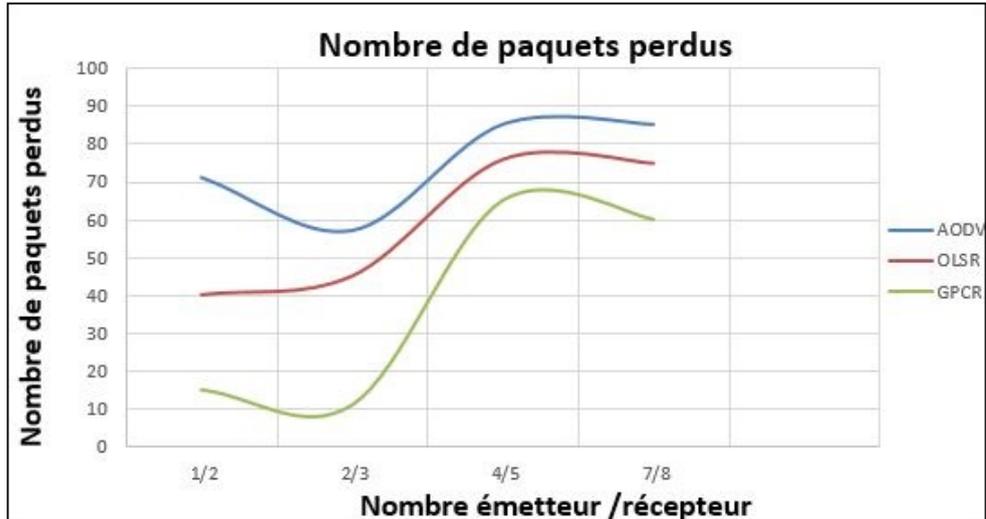


FIGURE 3.22 – Nombre de paquets perdus.

Conclusion

Dans ce chapitre nous avons évalué de quelques protocoles de routage AODV, GPCR et OLSR dans les VAENTs. Nous avons constaté que le protocole GPCR offre de meilleurs résultats en termes de délai moyen de bout-en-bout et en termes de PDR. Cependant, nous avons remarqué que le débit qu'offre le GPCR est relativement faible.

Donc on peut conclure que parmi les trois protocoles étudiés GPCR répond le mieux au besoin des réseaux véhiculaire et particulièrement en environnement urbain qui sont à la base de nos simulations.

Conclusion générale

La recherche dans le domaine des réseaux véhiculaires est en plein essor. En effet, le travail présenté dans ce mémoire porte sur la simulation des VANETs, notamment sur l'étude des performances des protocoles OLSR, AODV et GPCR.

Nous avons décrit les différentes technologies et techniques utilisées dans les réseaux VANETs. Nous avons également présenté les simulateurs les plus utilisés dans ce domaine tels que le simulateur de réseau NS-3 et le simulateur de trafic routier SUMO. Nous avons jugé utile de faire notre simulation sur une carte de réseaux routiers réels. L'objectif est d'obtenir des résultats d'évaluation cohérents avec ce type de réseaux mobiles.

Nous avons testé les capacités de création des routes, et l'acheminement des paquets sur les protocoles AODV, OLSR et GPCR. Dans l'ensemble de la littérature, il est admis que les protocoles géographiques sont les plus performants [4]. En effet, notre analyse a démontré que le protocole GPCR est meilleur que AODV ainsi que le protocole OLSR, ceci en zone urbaine.

Durant les deux phases de notre étude théorique et pratique, nous nous sommes vite rendu compte que le domaine de recherche dans les réseaux véhiculaire est tout jeune et qu'il a un long chemin à parcourir. Plusieurs groupes scientifiques s'intéressent à ce type de réseaux qui promet beaucoup pour l'avenir. En effet nous avons eu l'occasion de nous familiariser avec le langage C++ ainsi que les simulateurs NS-3 et SUMO.

Perspectives

Pour l'avenir, nous envisageons :

Adapté nos simulation a un environnement autoroutier, afin de conclure si ces protocoles (AODV, OLSR et GPCR) répondent de la même manière qu'aux environnements urbains.

Apporter une amélioration au protocole GPCR en proposons d'améliorer le temps de maintien de route qui est fixé par défaut a 2 second.

Utiliser des scénarios plus complexe et encore plus proche de la réalité, avec un passage à l'échelle en utilisant un maximum de nœud.

Enfin, nous souhaitons que notre travail apporte une contribution aux étudiants de notre université qui désire s'initier au domaine de la recherche dans les réseaux VANETs.

Bibliographie

- [1] S. A LLAL. l'optimisation des échanges dans le routage géocast pour les réseaux de véhicules ad hoc vanets. *Thèse présentée pour obtenir le grade de Docteur de l'Université Paris 13 en Sciences de l'ingénieur Mention Réseaux et Techniques Informatiques, Université Paris 13*, 10 Décembre 2014.
- [2] K. AIT ALI. Thèse doctorat modélisation et étude de performances dans les réseaux vanets. *École doctorale des sciences pour l'ingénieur et micro-techniques, Université de technologies Belfort Montbeliard*, 2013.
- [3] Conti M. Basagni, S. and I. Stojmenovic. Mobile ad hoc networking. *The Cutting Edge Directions, chapter 14 - 19. Wiley - IEEE PRESS, 2nd edition*, March 2013.
- [4] A. Benchabana. Dynamic source routing in ad hoc wireless networks. *analyse des protocoles de routage dans les réseaux VANET Université Kasdi Merbah-Ouargla*, 25 Juin 2014.
- [5] N. Chaib. Mémoire présenté en vue de l'obtention du diplôme de magister en informatique option : Ingénierie des systèmes informatiques sur la sécurité des communications dans les réseaux vanet. *Université Elhadj Lakhder - Batna*, 25 Juin 2010.
- [6] P. Duddalwar, A. Deshmukh, and S. Dorle. A comparative study of routing protocol in vehicular ad hoc network. *International Journal of Emerging Technology and Advanced Engineering*, March 2012.
- [7] M. Erritali. Thèse doctorat contribution à la sécurisation des réseaux ad hoc véhiculaires. *Université de technologie Rabat*, 10 Octobre 2013.
- [8] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. *In Universal Personal Communications Record, 1997. Conference Record, 6th International Conference pages 2553*, Octobre ,1997.

- [9] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Hoptimized link state routing protocol for ad hoc networks. *In Multi Topic Conference, Technology for the 21st Century. Proceedings.IEEE International*, 2001.
- [10] B. Johnson and A. Maltz. Dynamic source routing in ad hoc wireless networks. *In Mobile Computing, Kluwer Academic Publishers*, 1996.
- [11] F. KAISSER. thèse de doctorat :communications dans les réseaux fortement dynamiques,ecole doctorale sciences et technologies de l’information des télécommunications et des systèmes. *université de paris sud11*, 21 juin 2010.
- [12] M. Kamel, M.and Khaled and K. Islam. geographical opportunistic routing for vehicular networks. *IEEE international Symposium Word of Wireless Mobile and Multimedia Network*, 2010.
- [13] J. Luo, X. Gu, T. Zhao, and W. Yan. A mobile infrastructure based vanet routing protocol in the urban environment. *TIn Communications and Mobile Computing (CMC), International Conference volume 4, pages 2451–2455*, April 2010.
- [14] A. Maria and M. Biagi. Smart vehicles, technologies and main applications in vehicular ad hoc networks in vehicular technologies. *Deployment and Applications, Ed*, 2013,InTech.
- [15] H. Moustafa and Y. Zhang. Vehicular networks. *Techniques,Standards and Applications. CRC Press*, 2009.
- [16] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. in mo-bile computing systems and applications, proceedings. *WMCSA '99. Second IEEE Workshop*, February 1999.
- [17] E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. *In SIGCOMM '94 Proceeedings of the conference on Communications architectures, protocols and applica-tions*, 1994.
- [18] J. Petit. Thèse doctorat sur coût de l’authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires. *Université Toulouse III Paul Sabatier*, 13 Juillet 2011.
- [19] O. Rivaton. Thèse doctorat le routage de l’information dans les réseaux véhiculaires mobiles. *Université de technologies laval*, 2016.

- [20] G. Sofiane. Ingénierie des systèmes complexes et multimédia université d'oran. *Mémoire présenté en vue de l'obtention du diplôme de Magister en informatiques Option*, 04 Novembre 2015.
- [21] P. Toor, Y. Muhlethaler and A. Laouiti. Vehicle ad hoc networks. *applications and related technical issues, IEEE Commun*, 2008.
- [22] X. Wang. Thèse doctorat mobile ad-hoc networks : Applications. *Université de INTECH*, January 2011.
- [23] E. Weingartner, H. Lehn, and K. Wehrle. ta performance comparison of recent network simulators. *Department of Computer Science University of Southern California*, 2008.
- [24] D. Xavier, G Etienne, and L. DEBAT. Etude des simulateurs de routage pour les réseaux sans fil. *Internet-draft, IETF MANETWorking Group*, 2008.

ANNEXE I : Étape de JOSM à SUMO

I.1 JOSM

- Ouverture de JOSM.
- Dans la barre d'outils cliqué sur télécharger les donnée de la carte.
- Donner le nom du lieu à rechercher FIGURE 3.23.
- sélectionner la zone voulue puis enregistrer la Carte. FIGURE 3.24.

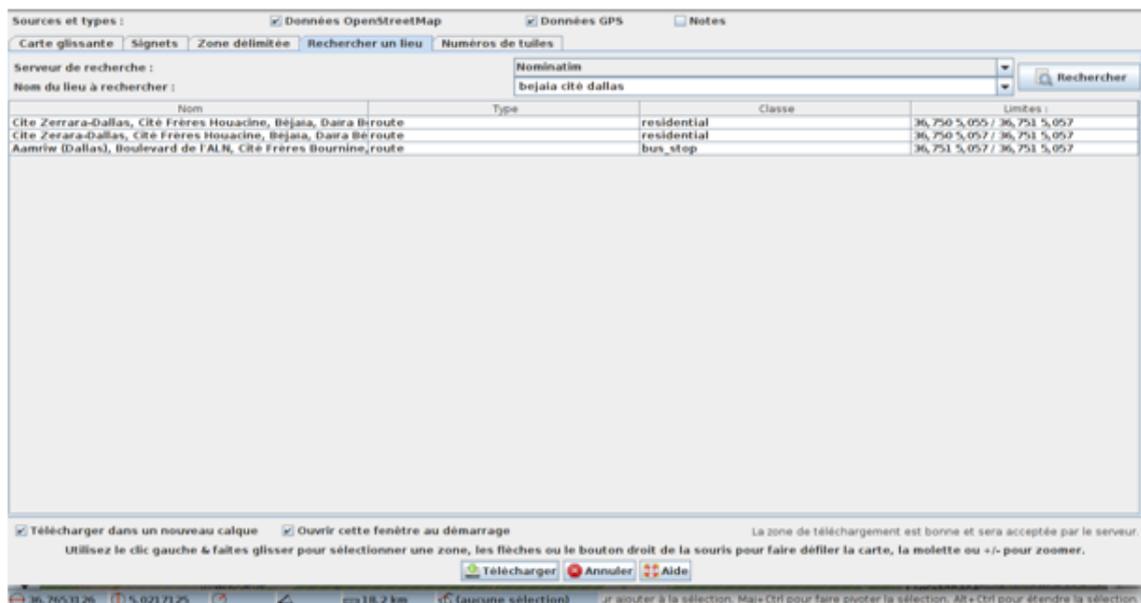


FIGURE 3.23 – Recherche un lieu.

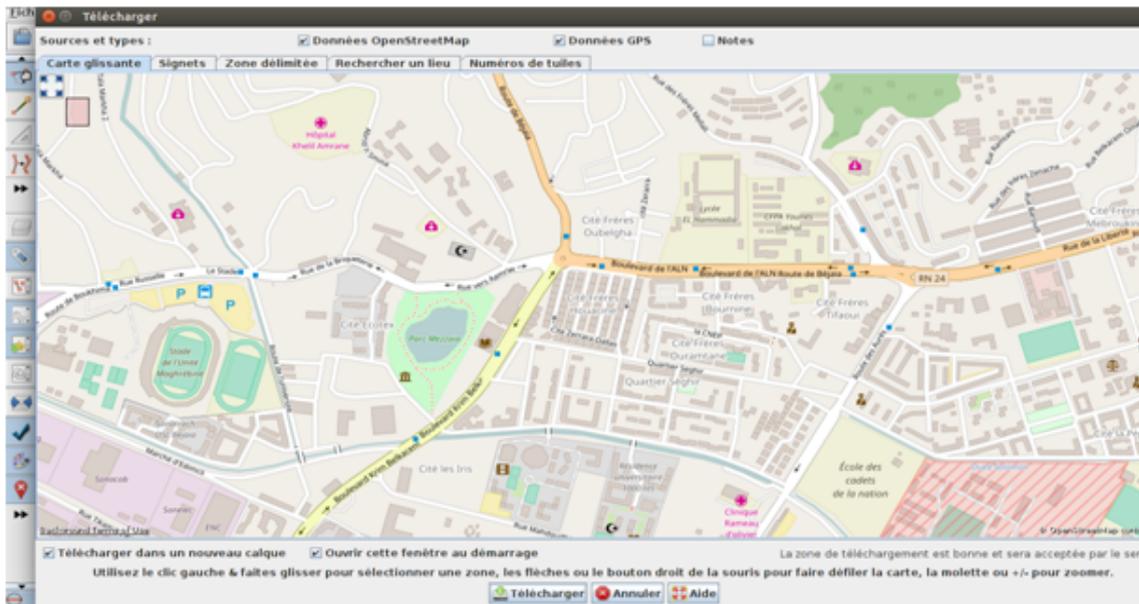


FIGURE 3.24 – *Télécharger la carte.*

I.2 eWorld

- pour ouvrir eWorld, on tape la commande suivante sur un terminal. `./eWorld.sh`
- Import->Import from OSM file

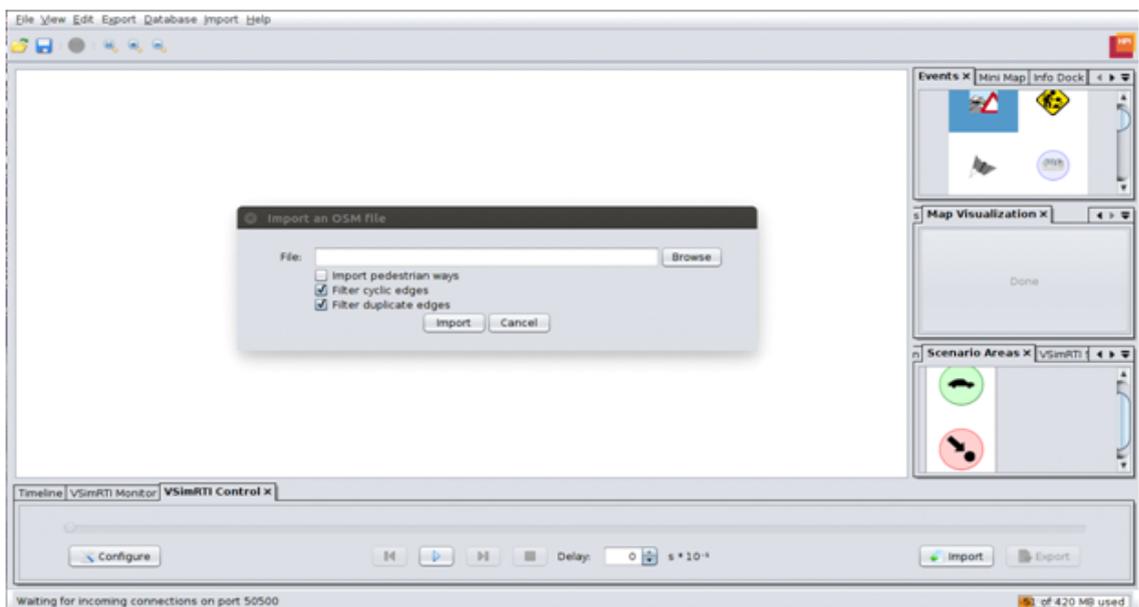


FIGURE 3.25 – *Importation de la carte.*

- Ajouter des points de départ et d'arrivée en prenant compte du nombre de véhicule et le temps de simulation.

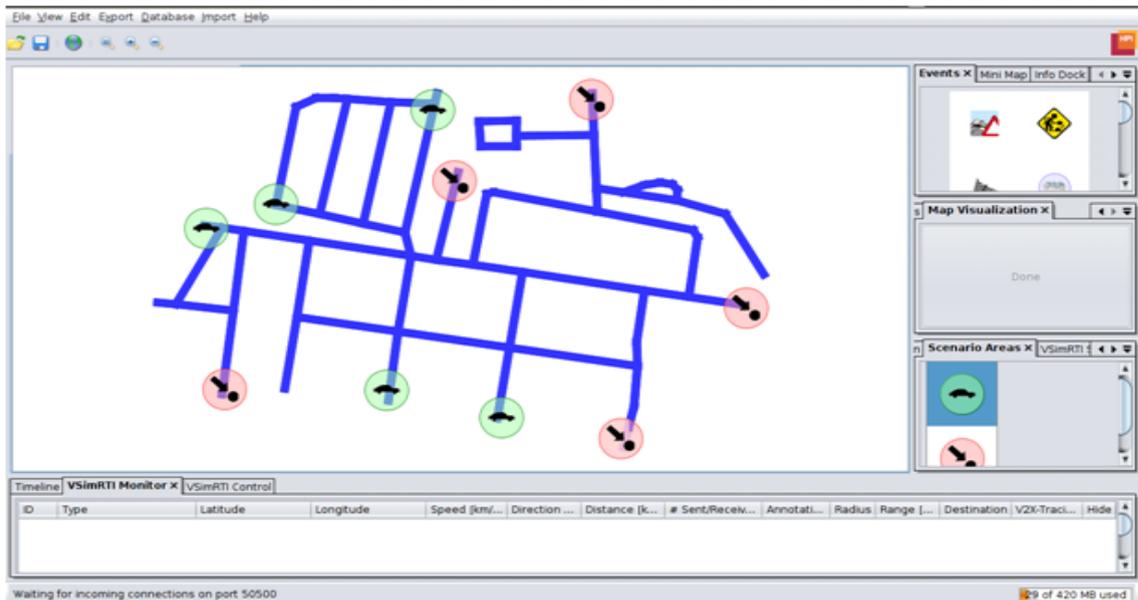


FIGURE 3.26 – Ajout des points de départ et d'arrivée.

- Export->Export to SUMO

Après avoir spécifié les paramétré suivant FIGURE 3.27 :

- (a) Location : à spécifier
- (b) Name : à spécifier
- (c) Case à cocher : Export traffic light system logics defined in eWorld
- (d) Case à cocher : Generate Net-File
- (e) Location of netconvert : /usr/bin/netconvert
- (f) Case à cocher : Generate routes with duarouter...
- (g) Location of duarouter : /usr/bin/duarouter
- (h) case à cocher : Generate scenarios
- (i) Export

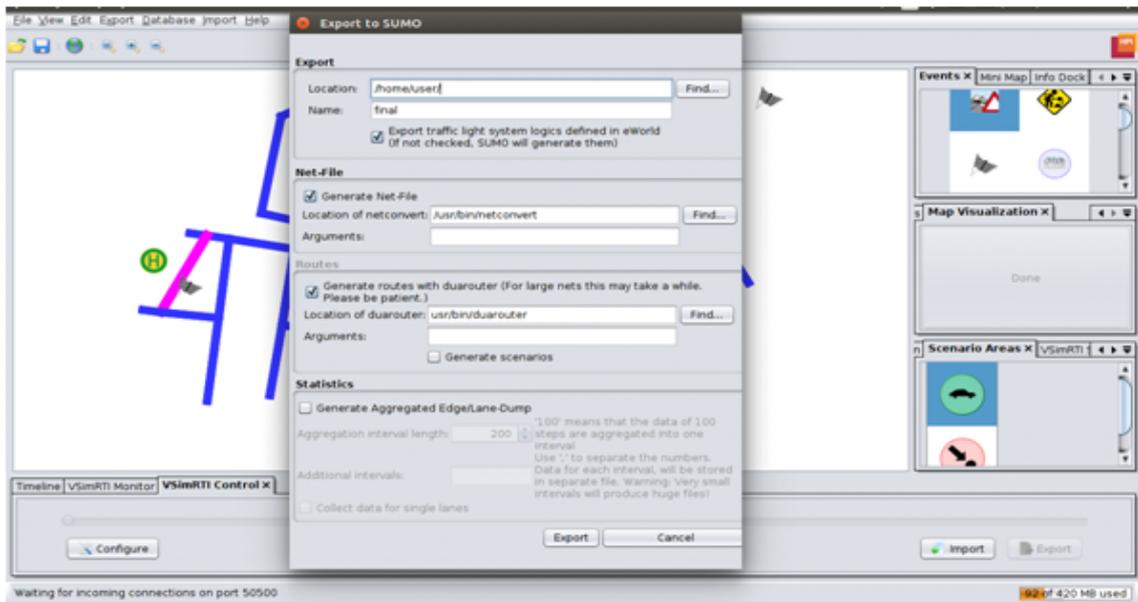


FIGURE 3.27 – Exporter la carte vers SUMO.

I.2.1 SUMO

- Effectuer quelques modifications sur les fichiers générés comme suit : Dans le fichier final.fows.xml :
Changer `no=" "/>` par `number=" "/>` dans toutes les occurrences du fichier.
exécuter la commande suivante :

```
duarouter --flows=final.flows.xml --net=final.net.xml
--output-file=finalNew.rou.xml
```

Supprimer `type="DefaultVehicle"` dans toutes les occurrences de `<vehicle id="0.0" type="DefaultVehicle" depart="0.00">`.

- Lancer SUMO et ouvrir le fichier de configuration final.sumo.cfg Mettre Delays (ms) à 580.
- Démarrer la simulation FIGURE 3.28.

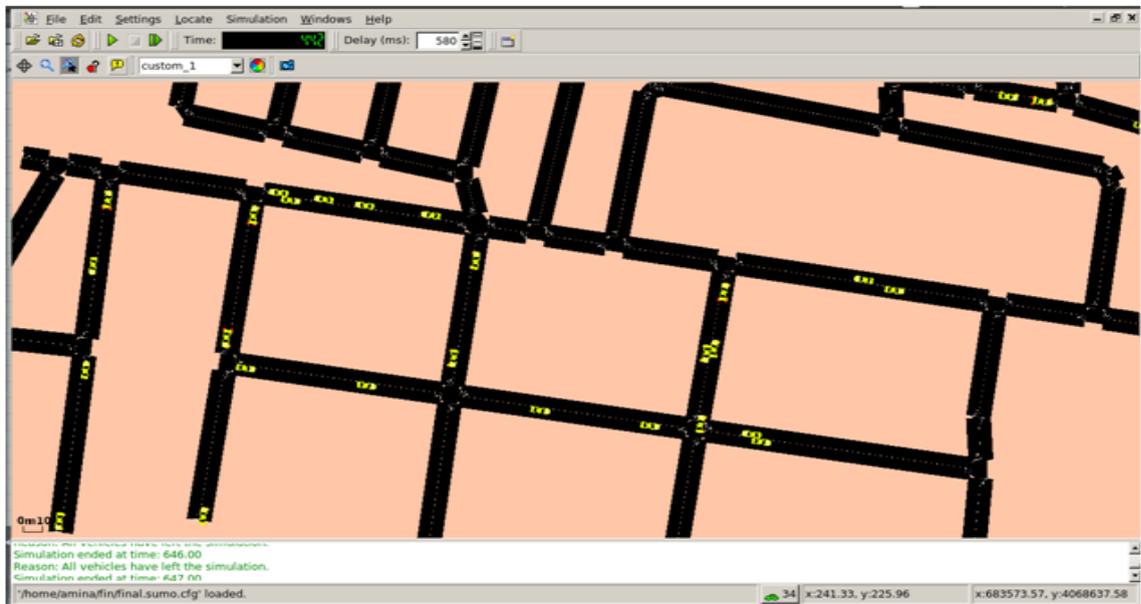


FIGURE 3.28 – SUMO.

ANNEXE II : Résultats de simulation

Nombre de noeuds	40	60	100	150
AODV	27	29	80	29
OLSR	25	25	25	26
GPCR	14	86	85	81

TABLE 3.2 – Taux de livraison des paquets (premier scénario).

Nombre de noeuds	40	60	100	150
AODV	1517.83	1630.18	1630.41	1665.1
OLSR	1405.53	1405.53	1405.53	1461.75
GPCR	1988.07	2344.23	2317	2207.96

TABLE 3.3 – Débit (bit/s) (premier scénario).

Nombre de noeuds	40	60	100	150
AODV	1397	5055	8989	9175
OLSR	96	96	131	121
GPCR	5.42857	9.17442	8.62353	12.3457

TABLE 3.4 – Délai (ms) (premier scénario).

Nombre de noeuds	40	60	100	150
AODV	73	71	71	71
OLSR	99.75	75	86	90
GPCR	86	14	15	19

TABLE 3.5 – Nombre de paquets perdus (premier scénario).

Temps de pause	1	2	5	8
AODV	36	31	17	10
OLSR	31	37	16	10
GPCR	54	75	87	95

TABLE 3.6 – Taux de livraison des paquets (deuxième scénario).

Temps de pause	1	2	5	8
AODV	2943.91	2023.14	948.944	562.093
OLSR	2535.03	2051.68	893.333	562.168
GPCR	4551.16	3066.59	1422.88	971.105

TABLE 3.7 – Débit (bit/s) (Deuxième scénario).

Temps de pause(s)	1	2	5	8
AODV	1348	3909	4134	241
OLSR	124	168	105	73
GPCR	19.8148	10.64	7.26437	7.48421

TABLE 3.8 – Délai(ms)(Deuxième scénario).

Temps de pause	1	2	5	8
AODV	64	69	83	90
OLSR	99.69	99.63	99.84	99.9
GPCR	46	25	13	5

TABLE 3.9 – Nombre de paquets perdus (Deuxième scénario).

Nombre émet- teur/récepteur	1/2	2/3	4/5	7/8
AODV	29	43	50	87
OLSR	25	38	40	83
GPCR	85	19	100	13

TABLE 3.10 – Taux de livraison des paquets (troisième scénario).

Nombre émet- teur/récepteur	1/2	2/3	4/5	7/8
AODV	1630.41	1172.14	1362.95	2371.55
OLSR	1405.53	1035.85	1090.36	2262.51
GPCR	2317	1553.42	2725.92	1168.81

TABLE 3.11 – Débit (bit/s) (troisième scénario).

Nombre émet- teur/récepteur	1/2	2/3	4/5	7/8
AODV	8989	8647	3428	7538
OLSR	131	86	1246	115
GPCR	8.62353	11.0526	11.36	36.4615

TABLE 3.12 – Délai(ms)(troisième scénario).

Nombre émet- teur/récepteur	1/2	2/3	4/5	7/8
AODV	71	57	50	13
OLSR	99.95	99.62	99.6	99.17
GPCR	15	81	6	87

TABLE 3.13 – Nombre de paquets perdus (troisième scénario).

ANNEXE III : Instalation des outils

Nous présentons ici, les différents outils de simulations installé sous Ubuntu 16.04 /64 bit.

III.1 Installation de JOSM

III.1.1 Installation de l'environnement Java Exécuter la ligne suivante dans un terminal :

- Sudo apt-get install default-jdk
- Sudo apt-get install default-jre

III.1.2 Téléchargement et lancement de JOSM Il faut se rendre sur le site :<http://josm.openstreetmap.de/>

- Cliquer sur le lien “télécharger la dernière version stable”

III.3 Installation d'eWorld

- Pour l'installer, il faut le télécharger du site :<http://eworld.sourceforge.net/>
- Pour commencer à travailler avec eWorld, décompresser l'archive téléchargée puis dans le répertoire eWorldAllInOne exécuter la commande : sudo ./eWorld.sh

III.4 Installation de SUMO Pour l'installer, ouvrir le terminal de Linux et exécuter la commande :`sudo apt-get install sumo`

III.5 Installation de NS-3 NS-3 est conçu pour fonctionner sur la plate-forme Linux .L'installation comprends les étapes suivantes :

III.5.1 Téléchargement

- Télécharger la dernière version <http://www.nsnam.org/releases/>
- Décompresser l'archive :`tar -jxf ns-3.19.tar.bz2`

III.5.2 Validation Tester l'exécution d'un script

- Sudo `./waf --run scrach/test --vis`

ANNEXE IV : Tableau comparatif entre les simulateurs

Dans la littérature, il existe plusieurs simulateurs réseaux, propriétaires et libres. Le tableau TABLE 3.14 dresse un comparatif des trois simulateurs réseau libres les plus utilisés pour la simulation des réseaux véhiculaires.

Critères d'évaluations	NS-3	NS-2	OMNET++
Interface Utilisateur (Entrée / Sortie)	✓✓	✓✓	✓✓
Acceptabilité dans le milieu de la recherche scientifique	✓✓	✓✓	✓✓
Génération de trafic de données	✓✓	✓	✓
Modèles de mobilité	✓✓	✓✓	✓
Documentation	✓✓	✓✓	✓
Portabilité	✓✓	✓	✓
Stabilité	✓✓	✓	✓
Performance du moteur de simulation	✓✓	✓	✓
évolutivité	✓✓	✓✓	✓✓

✓ Moyen
 ✓✓ Meilleur
 0 Resultats fiable
 0 Fiabilité non vérifiée

TABLE 3.14 – Tableau comparatif des différents Simulateurs.

ANNEXE V : L'outil utilisé pour établir les graphes de simulation

Nous avons utilisé l'Excel version 2016 pour établir les graphes de trois protocoles GPCR, OLSR et AODV.

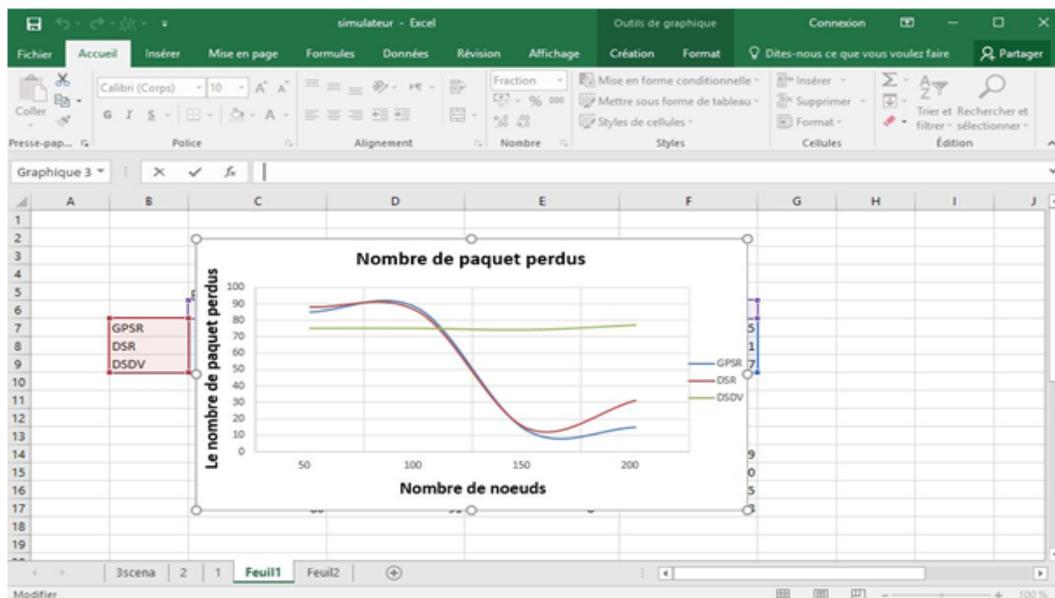


FIGURE 3.29 – EXCEL

ANNEXE VI : Script de simulation

```
*****      Importation des bibliothèques nécessaires*****  
#include "ns3/core-module.h"  
#include "ns3/network-module.h"  
#include "ns3/internet-module.h"  
#include "ns3/mobility-module.h"  
#include "ns3/wifi-module.h"  
#include "ns3/gpcr-module.h"  
// nous modifions "ns3/gpcr-module.h" par "ns3/aodv-  
module.h", "ns3/olsr-module.h" pour les deux protocoles (AODV et OLSR)  
respectivement.  
#include "ns3/applications-module.h"  
#include "ns3/propagation-loss-model.h"  
#include "ns3/ocb-wifi-mac.h"  
#include "ns3/wifi-80211p-helper.h"  
#include "ns3/wave-mac-helper.h"  
#include "ns3/stats-module.h"  
#include "ns3/point-to-point-module.h"  
#include "ns3/wifi-module.h"  
#include<sstream>  
#include<iostream>  
#include<string.h>  
#include<math.h>  
// nous ajoutons « #include <iostream> », « #include <cmath> » ,  
#include "ns3/udp-echo-server.h", « #include "ns3/udp-echo-client.h"  
» ,« #include "ns3/udp-echo-helper.h" » « #include "ns3/flow-monitor-  
module.h" », #include "myapp.h" » , «#include "ns3/propagation-loss-
```

model.h"», «#include "ns3/wave-mac-helper.h"» «#include "ns3/flow-monitor-module.h" » pour les deux protocoles AODV et OLSR

```
NS_LOG_COMPONENT_DEFINE("GpcrTest");
NS_LOG_COMPONENT_DEFINE("GpcrTest"); // nous modifications "GpcrTest" par "AodvTest", "OlsrTest" pour les deux protocoles (AODV et OLSR) respectivement.
```

```
using namespace ns3;
```

```
.....
```

```
int packetsReceived = 0;
```

```
long double throughput = 0.0;
```

```
long double bytesTotal = 0;
```

```
int m_lossCounter = 0;
```

```
long double xx; //le temps de génération de trafic sera utiliser dans le calcul de débit.
```

```
long double senttime = 0.0;
```

```
long double delay = 0.0;
```

```
long double rcv; //le temps de reception d'un paquet//le temps de réception d'un paquet.
```

```
long double sqhd; //Le temps de séquence.
```

```
uint32_t currentSequenceNumber; // num de seq en cours.
```

*******La fonction received paquet*******

```
void ReceivePacket(Ptr< Socket > socket)
```

```
NSLOGUNCOND("ReceivedOnepacket") Ptr< Packet > packet;
```

```
While(packet = socket->Recv())
```

```
SeqTsxHeader seqTsx;
```

```
packet->RemoveHeader(seqTsx); //séparer l'enteteSeqTsxHeader du paquet.
```

```
currentSequenceNumber = seqTsx.GetSeq(); // retourne le numéro de séquence.
```

```
bytesTotal += packet->GetSize(); // taille des donnée du paquet (sans entête).
```

```
packetsReceived += 1; // calculer le nombre de paquets reçus.
```

```
rcv = Simulator : :Now().GetMilliSeconds(); //temps de reception du paquet en cours.
```

```
sqhd = seqTsx.GetTs().GetMilliSeconds(); // le temps d'envoi du paquet.
```

```
std : :cout <<"SeqNo" <<currentSequenceNumber<<"Temps : " <<sqhd<< // vérifier le
```

numéro de séquence et le temps d'envoi du paquet.

delay = delay + (rcv - sqhd); // calcul du délai (le temps de réception - le temps d'envoi).

xx = rcv - senttime; // temps de réception du dernier paquet - temps de réception du premier paquet.

throughput = (bytesTotal * 1000 * 8) / (xx); // débit d'envoi calculé en bit/s.

*******La fonction génère trafic*******

```
static void GenerateTra_c(Ptr<Socket> socket, uint32_t pktSize, uint32_t pktCount, Time pktInterval)
```

```
if (pktCount > 0)
```

```
SeqTsHeader seqTs;
```

```
seqTs.SetSeq(pktCount); // car il commence du dernier paquet.
```

```
Ptr<Packet> p = Create<Packet>(pktSize - (8 + 4)); // créer un paquet "p" et soustrait 8+4 c'est la taille de l'entête SeqTs.
```

```
p->AddHeader(seqTs); // ajouter l'entête au paquet "p".
```

```
socket->Send(p); // la source envoie le paquet "p".
```

```
NS_LOG_UNCOND ("Envoi du paquet numéro "<< pktCount << " \n");
```

```
if (pktCount == 100)
```

```
senttime = Simulator::Now().GetMilliSeconds(); // retourne le temps d'envoi du premier paquet.
```

```
Simulator::Schedule(pktInterval, &GenerateTra_c, socket, pktSize, pktCount - 1, pktInterval);
```

```
else
```

```
socket->Close(); // fin de l'envoi.
```

//***** **la fonction main** *****

```
int main(int argc, char *argv[])
```

```
double duration = 600.0; // temps de la simulation.
```

```
std : :string phyMode("OfdmRate6MbpsBW10MHz"); // type de modulation OFDM 6Mbps BP=10MHZ (802.11p)
```

```

uint32_t packetSize = 1024; // en byte.
uint32_t numPackets = 100; // nombre de paquet.
uint32_t numNodes = 150; //nombre de nœuds.
uint32_t sourceNode = 1; // nœud émetteur.
double interval = 3; // temps inter paquet en secondes.
int range = 80; // porté de communication des nœud.
double hello = 1; // période d'émission des messages hello.
bool perimeter = true; // activer ou non le mode perimeter.
// Convertir la variable Seconds en objet de temps.
Time interPacketInterval = Seconds(interval);
// disable fragmentation for frames below 2200 bytes
Config : :SetDefault("ns3 : :Wi_RemoteStationManager : :FragmentationThre-
shold", StringValue("2200"));
// turn off RTS/CTS for frames below 500 bytes
Config : :SetDefault("ns3 : :WifiRemoteStationManager : :RtsCtsThreshold",
StringValue("500"));
// Fix non-unicast data rate to be the same as that of unicast
Config : :SetDefault("ns3 : :WifiRemoteStationManager : :NonUnicast-
Mode", StringValue(phyMode)); NodeContainer c; // la structure de nœud.
c.Create(numNodes); // créer un nombre = numNodes de nœud
//***** Configuration du canal WIFI*****
WifiHelper wifi;
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper : :Default();

//***** configuration du model de propagation*****
YansWifiChannelHelper wifiChannel; wifiChannel.SetPropagationDelay("ns3 : :ConstantS-
peedPropagationDelayModel"); //le temps.
wifiChannel.AddPropagationLoss("ns3 : :TwoRayGroundPropagationLossMo-
del", "SystemLoss", DoubleValue(1), "HeightAboveZ", DoubleValue(1.5)); //model
de propagation.
wi_Channel.AddPropagationLoss("ns3 : :FriisPropagationLossMo-
del", "MinDistance", DoubleValue(range)); //portée de communication.
wi_Phy.SetChannel(wi_Channel.Create()); // création du canal wifi.
//***** Configuration du protocole mac802.11p (WAVE)*****
NqosWaveMacHelper wi_80211pMac = NqosWaveMacHelper : :Default();

```

```

Wifi80211pHelper wifi80211p = Wifi80211pHelper : :Default();
wifi80211p.SetRemoteStationManager("ns3          :          :ConstantRateWifiMana-
ger","DataMode",StringValue(phyMode),          "ControlMode",          StringVa-
lue(phyMode));

    //*****instalation de couche phy et mac pour chaque
noeud***** NetDeviceContainer devices = wifi80211p.Install(wifiPhy,
wifi80211pMac, c);

    //*****Définir la mobilité des noeuds Ns2MobilityHelper et le fichier
mobility.tcl***** Ns2MobilityHelper ns2 = Ns2MobilityHelper("/...../NS-
3/ns-3.19/scratch/mobility.tcl"); ns2.Install(); // configure le mouvement des
noeuds contenu dans le fichier trace (mobility.tcl).
//***** Configuration du protocole de routage ad-Hoc*****
// activation de protocole.
nom_protocoleHelper nom_protocole; // activation de protocole.
InternetStackHelper internet;
internet.SetRoutingHelper(nom_protocole);
internet.Install(c); // installer la pile protocolaire.

    //*****Adressage IP*****
Ipv4AddressHelper ipv4;
NS_LOG_INFO ("Assign IP Addresses.");
ipv4.SetBase("10.1.0.0",          "255.255.0.0"); Ipv4InterfaceContainer i =
ipv4.Assign(devices); // assigner une adresse à chaque noeud.
nom_protocole.Set("HelloInterval",
TimeValue(Seconds(hello)));
nom_protocole.Install(); //
*****Aplication*****
uint32_t sinkNode = 2; // noeud recepteur
TypeId tid =
TypeId : :LookupByName("ns3 : :UdpSocketFactory");
Ptr<Socket> recvSink =
Socket : :CreateSocket(c.Get(sinkNode), tid); //receiving
sink at node "sinkNode"

```

```
InetSocketAddress local =
InetSocketAddress(Ipv4Address : :GetAny(), 80);
recvSink->Bind(local);
recvSink->SetRecvCallback(MakeCallback(&ReceivePacket));
Ptr<Socket> source =
Socket : :CreateSocket(c.Get(sourceNode), tid);
InetSocketAddress remote =
InetSocketAddress(i.GetAddress(sinkNode, 0), 80);
source->Connect(remote);
NS_LOG_UNCOND ("Testing " « numPackets « " packets sent ");
Simulator : :Schedule((Seconds(20)), &GenerateTra_c,
source, packetSize, numPackets, interPacketInterval);
Simulator : :Stop(Seconds(duration));
Simulator : :Run();
Simulator : :Destroy();
//écrire les résultats de la simulation
100 std : :cout « "Le Taux de livraison de paquet est : " « packetsReceived /
numPackets « "% \n" ; // les paquets reçus
std : :cout « "Le débit est " « throughput « " bit/sec \n" ; // le débit
std : :cout « "Délai est " « delay / packetsReceived « "ms \n" ; // délais de
propagation
point-à-point
std : :cout « "Le nombre de paquets perdus est : " «
numPackets - packetsReceived « " \n" ;
return 0;
```

Résumé

Les réseaux véhiculaires représentent aujourd'hui un intérêt certain pour l'industrie automobile, les opérateurs des réseaux, les organisations et même les particuliers. Le but des réseaux véhiculaire est d'améliorer la sécurité et l'efficacité des transports routiers afin de diminuer les accidents et fournir un environnement confortable aux conducteurs et à leurs passagers. Afin de choisir le protocole de routage qui convient le mieux aux VANETs parmi d'autres protocoles Ad hoc, ces derniers doivent être appliqués, un par un, sur le réseau. Mais vu le coût de réalisation concrète d'un VANET, et afin de pouvoir produire différents scénarios de tests, la simulation devient nécessaire avant de mettre ce système réellement en œuvre. Notre but, à travers ce travail, est d'évaluer les performances des protocoles de routage AODV, OLSR et GPCR appliqués aux réseaux VANET dans l'environnement spécifique d'une ville. Afin de choisir celui qui convient le mieux en terme de taux de livraison de paquet, délai de bout en bout, paquets perdus et le débit moyen.

Mots clés : VANET, SUMO, AODV, OLSR, GPCR, NS3.

Abstract

Nowadays, vehicular networks represent great interest in the industry of automobile, for network operators, the organizations, individuals as well. The purpose of the vehicular networks is to improve the security, and the efficiency of the road transport to decrease the accidents and to provide better conditions for drivers and passengers. In order to choose the convenient routing protocol among other protocols Ad Hoc, they need to be applied one by one in the network. To perform different scenarios of tests, the simulation of this system becomes necessary before its implementation. Our objective in this work is to evaluate routing protocol performances AODV, OLSR and GPCR applied on VANET network in specified environment of the city, in order to choose the best convenient one regarding the reception rate of packets, the end to end delay, lost packets and medium debit.

Keywords : VANET, SUMO, AODV, OLSR, GPCR, NS3.