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

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

MÉMOIRE DE MASTER PROFESIONNEL

En

Informatique

Option

Administration et Sécurité des Réseaux

Thème

Simulation et évaluation des performances des protocoles de routage (GPSR, DSR et DSDV) dans les réseaux VANETs

---

Présenté par : Mme Djellab Samia
Mme Mouradi Warda

Soutenu le 02 Juillet 2017 devant le jury composé de :

Rapporteur	**Mme N. TASSOULT**	**M.A.A**	**U. A/Mira Bejaia.**
Président	**M A. ACHROUFENE**	**M.A.A**	**U. A/Mira Bejaia.**
Examinateur	**Mme D. KESSIRA**	**M.A.A**	**U. A/Mira Bejaia.**
Examinateur	**Mme A. BOUDRIOUA**	**Doctorante**	**U. A/Mira Bejaia.**

Promotion 2016-2017.

Remerciements

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

Nous souhaitons adresser nos remerciements les plus sincères à nos parents qui tout au long de ce travail, nous ont apporté leurs précieux soutien ainsi que leur encouragements.

Nous tenons à remercier sincèrement notre 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.*

Nos vifs remerciements vont également aux membres du jury qui ont accepté d'examiner notre travail et de l'enrichir par leurs propositions.

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

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

Dédicaces

A nos parents, pour leurs sacrifices déployés à notre 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 et à mes sœurs et tous mes amis, ***Lynda, Soraya, Nadjet, Yamina et Sabrina*** pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Samia

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 mes attachements indéfectibles, nulle dédicace ne puisse exprimer ce que je leur doit.

A mes chers frères ***Redouane et Mahrez*** qui ont étaient d'une grande aide tout au long de cette période,ma belle sœur ***Mounia*** et tous mes amis et plus particulièrement à ***Soraya, Lynda, Yamina*** et ***Sabrina*** pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Warda

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

A-STAR	**A**nchor-based **S**treet and **T**raffic **A**ware **R**outing .
AODV	**A**d hoc **O**n-demand **D**istance **V**ector .
AU	**A**pplication **U**nit.
DSR	**D**ynamic **S**ource **R**outing .
DSDV	**D**estination **S**equence **D**istance **V**ector.
DSRC	**D**edicated **S**hort **R**ange **C**ommunications.
FCC	**F**ederal **C**ommunications **C**ommission.
GPS	**G**lobale **P**osition **S**ystem.
GSR	**G**eografic **S**ource **R**outing **F**il.
GSR	**G**lobal **S**tate **R**outing.
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing.
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing.
HSR	**H**ierarchical **S**tate **R**outing.

IEEE	**H**ierarchical **S**tate **R**outing.
ITSA	**I**ntelligent **T**ransportation **S**ociety Of **A**merica.
IVC	**I**nter-**V**ehicule **C**ommunication.
ITS	**I**ntelligent **T**ransporation **S**ystem.
JOSM	**J**ava **O**pen **S**treet **M**aps.
MANET	**M**obile **A**d hoc **NET** work.
NS-3	**NET**work **S**imulator **3**.
OBU	**O**n **B**aord **U**nit.
OLSR	**O**ptimized **L**ink **S**tate **R**outing **P**rotocol.
QRY	**Q**uery .
RLS	**R**eactive **L**ocation **S**ervice.
RSU	**R**oad **S**id **U**nit.
RLS	**R**eactive **L**ocation **S**ervice.
SUMO	**S**imulation of **U**rban **MO**bility.
SN	**S**equence **N**umber.
TORA	**T**emporally-**O**rdered **R**outing **A**lgorithm.
UPD	**U**pdate.
VADD	**V**ehicle-**A**ssisted **D**ata **D**elivery.

V2I	**V**éhicule **T**o **I**nfrastructure.
V2V	**V**éhicule **T**o **v**éhicule.
VANET	**V**ehicular **A**d hoc **NET**works.
WAVE	**W**ireless **A**ccess in **V**ehicular **E**nvironment.
WSM	**W**awe **S**hort **M**essages.
WSMP	**W**AVE **S**hort **M**essages **P**rotocol.
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 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.

Pour acheminer les informations d'un véhicule à un autre à travers un réseau composé de beaucoup de véhicules, il est nécessaire d'effectuer un routage rapide et efficace de l'information. Le routage est une méthode d'acheminement des informations vers la bonne destination à travers un réseau de connexion de donnée. Il consiste à assurer une stratégie qui garantit, à tout moment, un établissement de routes qui soient correctes et efficaces entre n'importe quelle paire de nœuds appartenant au réseau.

Dans ce mémoire nous avons choisi trois protocoles de routage GPSR (Greedy Perimetre Stateless Routing), DSR (Dynamic Source Routing) et DSDV (Destina-

tion Sequence Distance Vector) afin de d'évaluer leurs performances. Pour se faire nous avons utilisé NS-3 (Network Simulator 3) 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 (Java Open Street Maps), puis nous l'avons exporté vers le simulateur SUMO (Simulator Urban Mobillity) pour générer le modèle de mobilité. Nous l'avons ensuite intégré à NS-3 à l'aide de Ns2 Mobility Helper.

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 allons détailler les déférents protocoles de routage dans les réseaux véhiculaires et leur fonctionnement. Le troisième chapitre est consacré à l'évaluation des protocoles GPSR, DSR et DSDV. Nous effectuons en premier lieu une comparaison de performances entre ces protocoles en terme 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 cour de la simulation.

Nous terminons par une conclusion générale, et quelques perspectives.

Chapitre 1

Généralités sur les réseaux VANETs

Introduction

Les réseaux VANETs (Vehicular Ad hoc NETworks) constituent une nouvelle forme de réseaux ad hoc mobiles permettant d'établir des communications entre les véhicules ou bien avec une infrastructure située aux bords de routes.

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 .

Dans ce chapitre nous allons donner une vue générale sur les réseaux VANETs.

1.1 Réseau Ad-hoc

Les réseaux ad-hoc sont des réseaux sans fil capables de s'organiser sans infrastructure. Au lieu de communiquer via un point d'accès centralisé, chaque entité communique directement avec tous les nœuds qui se situent dans la portée de la couverture radio comme l'indique la FIGURE 1.1 [30].

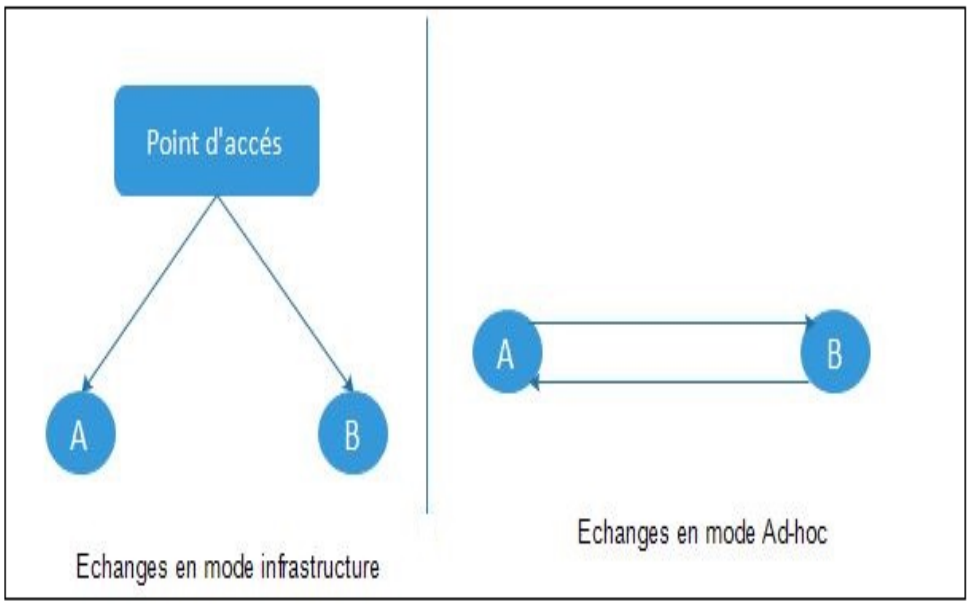

FIGURE 1.1 – Réseau Ad-hoc [30].

1.1.1 Caractéristiques des réseaux Ad-Hoc

Les réseaux mobiles Ad Hoc présentent plusieurs caractéristiques, parmi elles :

- ***Absence d'infrastructure***
  Les nœuds d'un réseau Ad Hoc travaillent dans un environnement pair à pair totalement distribué, ce qui leur permet de se déplacer librement. Ces nœuds agissent en tant que routeurs pour relayer des communications ou générer leurs propres données [30].

- ***Routage par relais***
  Dans un réseau Ad hoc, un terminal peut communiquer directement avec les terminaux à sa portée (ses voisins). Lorsqu'une machine veut communiquer avec une autre se trouvant hors de sa portée, chaque nœud actif du réseau sert de routeur pour ses voisins[30] .

- ***Topologie dynamique***
  Une particularité très importante qui distingue les réseaux mobiles Ad Hoc des réseaux filaires est la mobilité de ses nœuds. Les nœuds sont libres de se déplacer arbitrairement, des routes peuvent se créer et disparaître très souvent, ce qui provoque des changements fréquents dans la topologie du réseau. Ces modifications doivent être prises en compte par le protocole de routage. Cette caractéristique rend la topologie de ce type du réseau sans fil très dynamique[30].

- ***Multi-sauts***
  Les réseaux Ad Hoc utilisent souvent des sauts multiples pour éviter les obstacles, minimiser la consommation d'énergie ou pour joindre un nœud qui n'est pas dans la portée de communication de l'émetteur[30].

1.2 Réseau MANET

Le réseau mobile ad hoc, appelé généralement MANET (Mobile Ad hoc NETwork) est un système autonome se compose d'un nœud mobiles dynamiques interconnectés par des liens sans fil sans l'utilisation de l'infrastructure fixe et sans gestion centralisée [9]. Les nœuds sont libres de se déplacer de façon aléatoire et, par conséquent, peuvent changer la structure du réseau rapidement et de manière imprévisible.

1.3 Réseaux VANETs

Un réseau VANET (Vehicule Ad-Hoc NETworks) fait partie de réseau MANET (Mobile Ad -hoc Network). Dans le réseau VANET, les nœuds sont les véhicules intelligents appartenant au réseau. Qui équipés des calculateurs, périphériques réseau et de différents types de capteurs. Nous avons les véhicules peuvent communiquer entre eux V2V (Véhicule to Véhicule) par exemple pour échanger les informations sur le trafic ou avec des stations de base placées sur des routes V2I (Véhicule to Infrastructure), exemple pour demander des informations ou accéder à internet.

L'objectif principal des réseaux VANETs est d'améliorer la sécurité routière tout en élaborant des routes plus sûres et efficaces en fournissant des informations opportunes aux conducteurs [21]. Un exemple de réseaux VANET urbain est illustré dans la FIGURE 1.2 suivante :

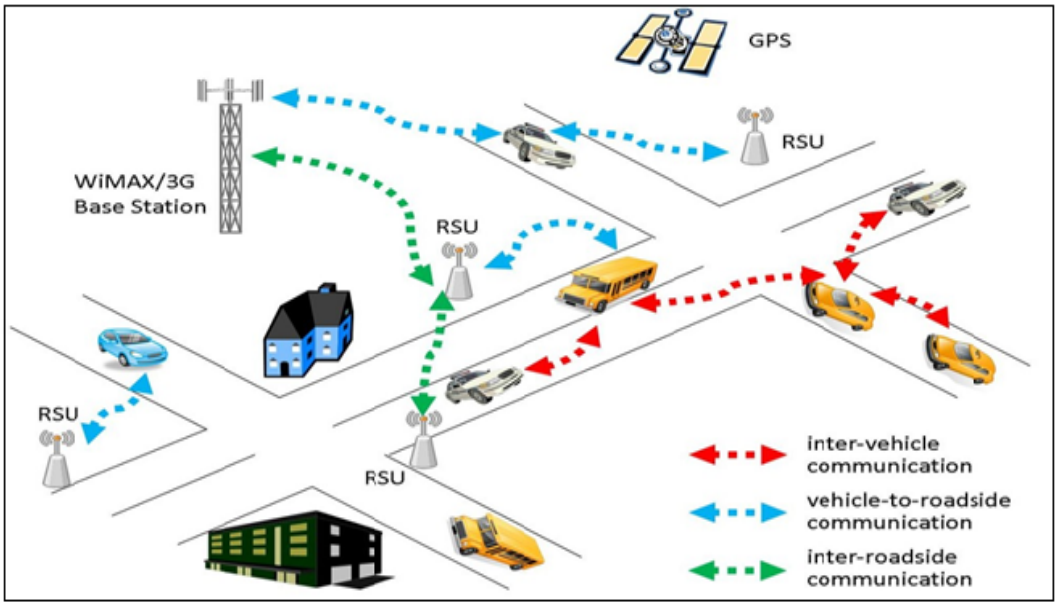

FIGURE 1.2 – *Réseau VANET [21].*

1.3.1 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. Nous parlons de la notion de « véhicule intelligent ». La FIGURE 1.3 modélise un exemple de véhicule intelligent [2] .

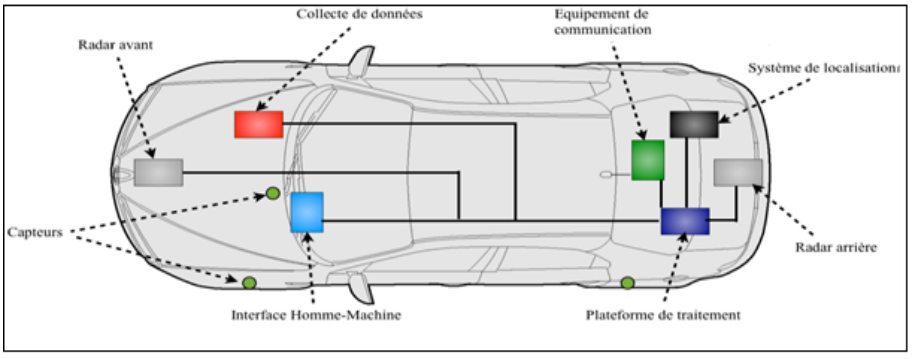

FIGURE 1.3 – *Véhicule intelligent [17].*

1.3.2 Entités de communication

Dans un réseau véhiculaire sans fil, il existe plusieurs entités permettant la communication qui réalise le dispositif sans fil appelé WAVE (Wireless Access in Vehicular Environment) les principaux composants de ce système sont les : AU (Application Unit), OBU (On Baord Unit) et RSU (Road Sid Unit) [2][1].

- ***OBU (On Baord Unit)*** : Un OBU est un équipement WAVE mobile qui permet des communications OBU à OBU en mode ad hoc, et des communications OBU à RSU en mode infrastructure. Les OBU sont également reliés à une gamme de capteurs et d'actionneurs au sein du véhicule. Ceci facilite la surveillance efficace des véhicules pour rassembler des informations comme la vitesse du véhicule et son accélération.

- ***AU (Application Unit)*** : Connectée à l'OBU ce périphérique intégré à l'intérieure du véhicule est dédiée aux applications liées à la sécurité, au confort et aux loisirs.

- ***RSU (Road Sid Unit)*** : C'est un périphérique WAVE son rôle est de connecter le véhicule au réseau commun, qui ensuite les connectent au cœur central du réseau. Les RSUs sont habituellement installées sur des infrastructures existantes telles que les feux de circulation et les panneaux routiers ou encore les lampadaires.

1.3.3 Architecture des réseaux VANET

Les progrès dans les communications mobiles et les tendances actuelles dans les réseaux ad hoc permettent différentes architectures de déploiement pour les réseaux de véhicules dans les autoroutes, les milieux urbains et ruraux pour soutenir de nombreuses applications avec des exigences de qualité de service différentes [35].

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 :

1.3.3.1 Mode de communication de véhicule à véhicule (V2V)

Le mode de communication de véhicules à véhicules V2V (Vehicle to Vehicle) Appelé également IVC (Inter-Vehicule Communication), est un mode décentralisé car il est composé uniquement d'OBU (véhicules légers, poids lourds, véhicules de secours, etc.)[33] où les véhicules intelligents communiquent les informations, chacun avec les véhicules à portée, d'une façon autonome sans utiliser une quelconque infrastructure [12].

En effet, un véhicule peut communiquer directement avec un autre véhicule s'il se situe dans sa zone radio voir la FIGURE 1.4, ou bien par le biais d'un protocole multi-sauts qui se charge de transmettre les messages de bout en bout en utilisant les nœuds voisins qui les séparent comme des relais.

La communication de véhicule à véhicule elle donne une communication moins coûteuse et très efficace pour le transfert des informations concernant les services liés à la sécurité routière, mais elle ne garantis pas une connectivité permanente entre les véhicules.

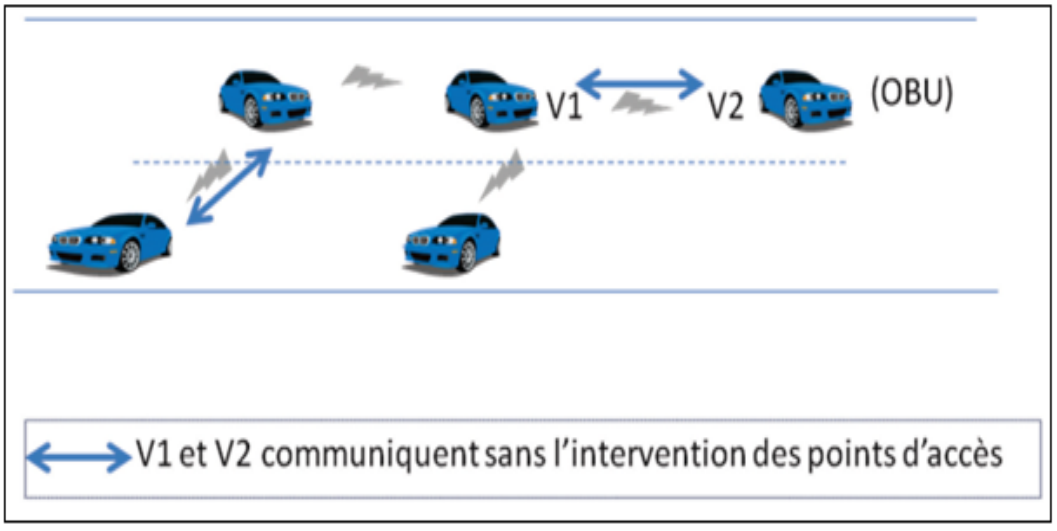

FIGURE 1.4 – *Communication véhicule à véhicule [5].*

1.3.3.2 Mode de communication de Véhicule à infrastructure (V2I)

Dans ce mode de véhicule à infrastructure il se compose sur des stations de base ou point d'accès RSU (Road Sid Unit). Cette approche repose sur le modèle

client/serveur où les véhicules sont les clients et les stations installées le long de la route sont les serveurs. Ces serveurs sont connectés entre eux via une interface filaire ou sans fil voir la FIGURE 1.5. Toute communication doit passer par eux. Ils peuvent aussi offrir aux utilisateurs plusieurs services concernant le trafic, accès à internet, échange de données de voiture à domicile et même la communication de voiture à garage pour le diagnostic distant.

L'inconvénient majeur de cette approche est que l'installation des stations le long des routes est une tâche coûteuse et prend beaucoup de temps, sans oublier les coûts relatifs à la maintenance des stations [26].

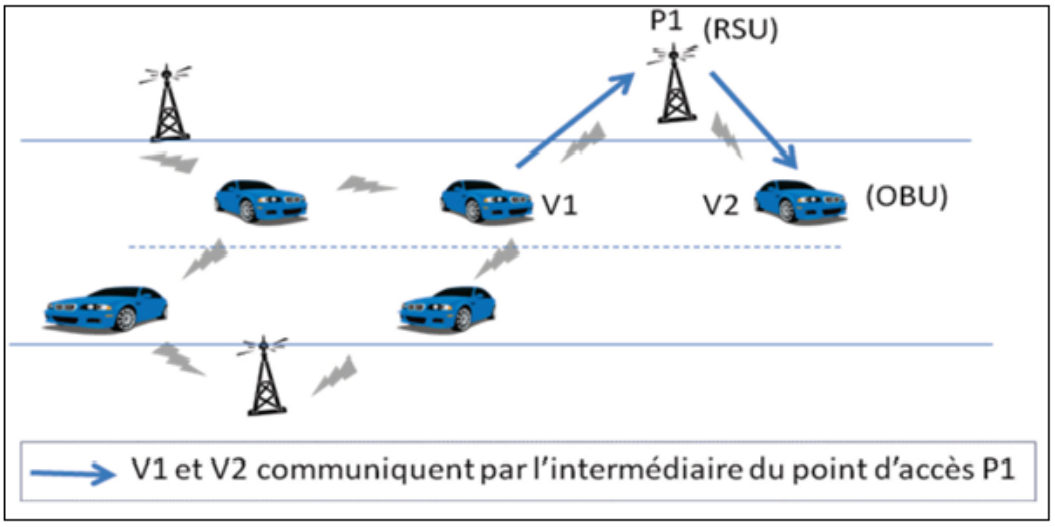

FIGURE 1.5 – *Communication véhicule à infrastructure [5].*

1.3.3.3 Mode de communication hybride

Les communications hybrides combinent les modes véhicule infrastructure et véhicule-véhicule. Ce mode permet de couvrir un maximum d'infrastructures grâce à la communication V2V où les véhicules servent de relais pour étendre les informations voir la FIGURE 1.6. Cela permet d'économiser sur les coûts des infrastructures : un nombre beaucoup moins élevé d'infrastructures peut être suffisant, davantage sur les autoroutes. En plus, ce mode permet de répondre à la problématique de connectivité de longue distance dans les réseaux V2V.

Les infrastructures, n'ayant pas des contraintes de mobilité, servent à leur tour de relais fixes afin d'étendre la distance de communication inter-véhicules. Les deux

premiers modes se montrent complémentaires, leur combinaison est donc très intéressante [33].

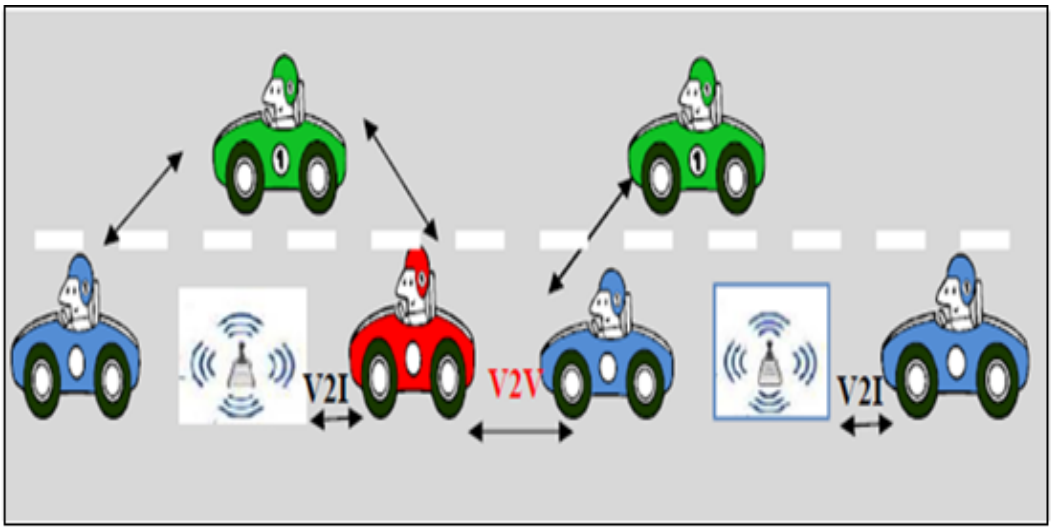

FIGURE 1.6 – *Communication hybride [6].*

1.3.4 Application de réseau VANET

Après avoir présenté les entités de communication de réseau VANET et les modes de communication, nous détaillons maintenant les applications qui peuvent être déployées sur ce type de réseau. On distingue deux types d'applications : application pour la sécurité du trafic routier et application de confort.

1.3.4.1 Application de sécurité du trafic routier

Application de sécurité du trafic routier. Cette d'application vise à améliorer la sécurité routière. Il s'agit d'améliorer le champ de vision du conducteur en lui proposant une aide à la conduite. On a Le conducteur pourra ainsi anticiper et agir pour rendre la conduite plus sûre et aussi il pourra être informé qu'un véhicule vient de passer un feu rouge ou qu'un piéton est en train de traverser la route. Dans ce type, on retrouve les applications qui utilisent les informations des autres véhicules : l'alerte d'état de la route (verglas, obstacle), l'aide au dépassement (calcul des distances, vérification de l'angle mort), l'alerte de freinage ou de collision en amont du trajet.

On remarque donc que les applications de sécurité du trafic routier ont un rôle majeur dans la réduction du nombre d'accidents [33].

Nous allons citer les principaux cas où les applications de sécurités sont vraiment bénéfiques.

- ***Dans le cas d'un accident*** : Les voitures circulent à grande vitesse sur les routes, ne laissant qu'un petit moment de réaction au conducteur face aux véhicules devant lui. Si un accident venait à surgir, les véhicules derrières, entrent souvent en plein choque. Les applications de sécurité peuvent être utilisées dans ce cas pour avertir le conducteur d'un accident qui s'est produit plus loin dans la route ce qui éviterait un carambolage. Ils peuvent être utilisés pour envoyer un avertissement précoce au pilote et éviter qu'un accident ne surgisse [42].

- ***Dans le cas de ralentissement anormal (embouteillage, travaux, intempéries, etc.)*** : Ce service permet d'avertir les automobilistes de situations de circulation particulières. L'information, quelle que soit la nature des difficultés de circulation, renseigne l'automobiliste de la menace et qu'il est nécessaire de ralentir. Le message d'alerte est émis par un véhicule détectant les difficultés de circulation (freinage important, déclenchement des feux de détresse, pluie par exemple). Un véhicule banalisé effectuant des travaux peut également être à l'origine du message d'alerte. Comme pour le message d'alerte informant d'un accident, le message d'alerte informant d'un ralentissement doit être transmis aux autres véhicules de façon efficace et rapide [33].

- ***Trafic routière*** : Les applications de sécurité elle permet d'aider les pilotes à déterminer la meilleure route à prendre pour leur destination, ce qui réduira les circulations et démunira de manière indirecte le nombre d'accidents dans la route [33].

1.3.4.2 Application de confort

L'objectif principal de cette application est de rendre les voyages plus agréables qui permettant aux passagers de communiquer soit avec d'autres véhicules qui peuvent s'échanger des messages ou partager des données (vidéo, musique, itinéraire,

jeux en réseau)ou avec des stations fixes comme l'accès à internet, la messagerie, le chat inter – véhicule, etc. La vie des usagers pourra aussi être facilitée par le contrôle à distance de véhicule de manière électronique (vérification du permis de conduire, contrôle technique, plaque d'immatriculation) pour les services compétents (police, douane, gendarmerie) [33].

1.3.5 Travaux de standardisation et de normalisation

Nous citons les différents travaux de standardisation et de normalisation :

1.3.5.1 DSRC (Dedicated Short Range Communications)

C'est Les premiers standards définis pour les communications sans fil dans les STI utilisent la bande de fréquence de 915MHz essentiellement pour assurer des services tels que, le péage électronique, l'accréditation et la surveillance des opérations des véhicules commerciaux.

Cette bande de fréquence étant trop étroite et polluée pour supporter l'évolution Envisagée pour les applications dans les réseaux véhiculaires, l'ITSA (Intelligent Transportation Society Of America) a sollicité la FCC (Federal Communications Commission) pour l'allocation d'une bande passante de 75MHz dans la gamme de fréquences 5,850-5,925GHz pour les communications à courte portée dédiées aux STI aux USA. Cette demande a été accordée par la FCC en 1999 et a donné naissance àla technologie DSRC [2].

1.3.5.2 WAVE (Wireless Access in Vehicular Environments)

Le modèle de communication à courte portée DSRC est conforme à l'architecture OSI mais avec une importance accordée à trois couches suffisantes pour les besoins de la communication. C'est la couche 1 dite « Physique » basée sur le lien micro-ondes à 5.8 GHz, et la couche 2 dite « Liaison » permettant la gestion des données et les communications les véhicules et l'infrastructure et la couche 7 dite « Application » comportant un ensemble de commandes qui traitent les applications envisagées i.e. les applications embarquées sur le dispositif [18][6]. Le reste de la pile protocolaire de WAVE se situant entre la couche liaison de données et la couche application représente l'architecteur de WAVE que le groupe IEEE1609 à standardiser sous

quatre catégories comme la FIGURE 1.7 le montre :

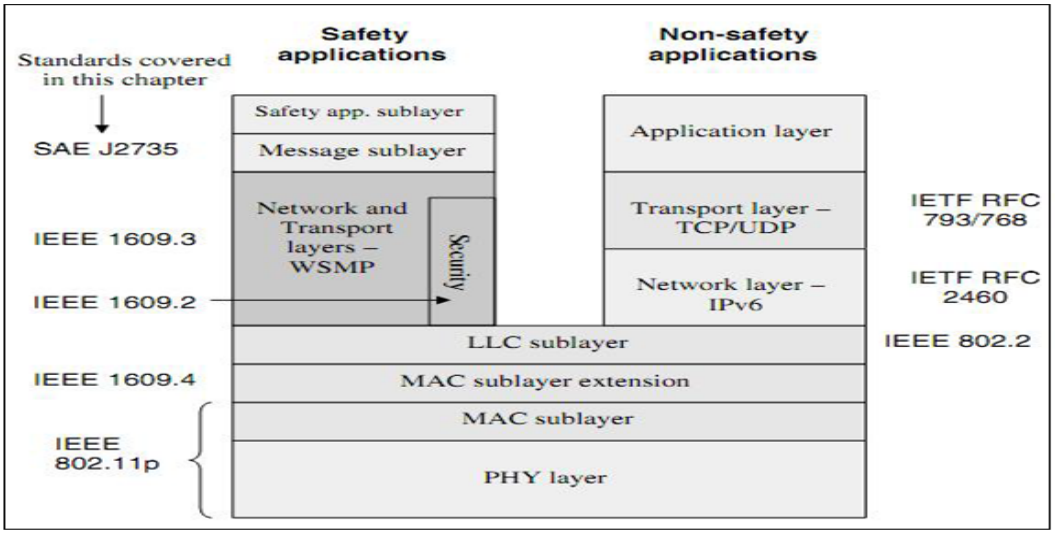

FIGURE 1.7 – *Pile protocolaire [33].*

IEEE 1609.1 - WAVE Resource Manager : Permet couche application représente l'architecture de WAVE que le groupe IEEE1609 à standardiser sous quatre aux applications distantes une bonne gestion de ressources de OBU telle que : La mémoire et l'interface utilisateur en garantissant l'interopérabilité [38].

IEEE 1609.2 – WAVE Security Services for Applications and Management Messages : Décrit le format des paquets et les fonctions de sécurité dans un système WAVE pour les messages de sécurité, données et gestion [39].

IEEE 1609.3 - WAVE Networking Services : Définit les services de niveau réseau et transport incluant l'adressage, le routage et la transmission, l'envoie est basé sur le protocole WSMP (WAVE Short Messages Protocol) qui permet un échange efficace des messages WSM (WAVE Short Messages) [40].

IEEE 1609.4 - WAVE Multi-Channel Operation) : Définit les mécanismes d'accès en priorité, la coordination et la gestion des 7 canaux DSRC lors de routage et transmission de données [41].

1.3.5.3 Norme IEEE 802.11p

La norme IEEE 802.11p [19] est un amendement du standard IEEE 802.11 que le groupe de travail IEEE (TGP, task group p) a commencé à développer en 2004 pour

l'accès sans fil dans les systèmes de transport intelligents. Il définit les spécifications des couches MAC et PHY dans le cadre des réseaux véhiculaires.

1.3.6 Caractéristiques des réseaux VANETs

Les VANETs possèdent un nombre de caractéristiques spécifique qui les différencient des autres types de réseaux sans infrastructure. Ces caractéristiques peuvent se traduire par des contraintes ou des points forts ayant un impact sur les communications [15]. Les caractéristiques principales des réseaux ad hoc véhiculaires, lesquelles doivent être prises en compte par toute solution dédiée, sont :

- ***Collecte d'informations et la perception de l'environnement proche***
  La collecte d'informations se fait en utilisant différents capteurs de toutes catégories par exemple (caméras, capteurs de pollution, capteurs de pluies, capteurs de l'état de la route et de voiture) qui permettent au conducteur à bord de son véhicule de disposer d'un certain nombre d'informations et d'une meilleure visibilité pour pouvoir réagir d'une manière adéquate aux changements de son environnement proche [27] [31].

- ***Energie***
  A l'opposé des autres réseaux ad hoc où l'énergie est limitée, les nœuds VANET disposent d'une très grande capacité d'énergie (La batterie de la voiture) nécessaire aux différents traitements de la voiture intelligente. Or de nos jours les véhicules doivent être écologiques et l'ajout de nouveaux équipements nécessaires à la communication sans fil complexifie la gestion énergétique du véhicule il faut donc prendre ce problème en considération lors du déploiement des VANET [33].

- ***Forte mobilité***
  Les réseaux VANETs sont caractérisés par la forte mobilité des nœuds (véhicules), la vitesse des voitures varie selon l'environnement, elle est en moyenne de 50km/h en zones urbaines et peut atteindre 130km/h sur autoroute. Bien que les mouvements des véhicules soient relativement prédictibles, un nœud peut rejoindre ou quitter le réseau en un temps très court, ce qui rend les changements de topologie très fréquent [2].

- ***Environnement de déplacement et modèle de mobilité***
  Dans VANET les environnements sont très variés. Ce dernier 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. La mobilité dans les réseaux véhiculaires 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 connaitre la carte de l'environnement [34].

1.3.7 Défis des réseaux VANETs

Les VANETs découlent plusieurs défis que l'on peut résumer en ces points [2] [33].

- ***Sécurité***
  Les exigences en sécurité doivent être prises en compte aussi bien dans la conception architecturale du réseau que dans la conception des protocoles de communication. Elles diffèrent en fonction des applications et comprennent principalement la confidentialité, l'authentification, la cohérence et l'intégrité des données et la disponibilité. La satisfaction de ces exigences dans des systèmes aussi dynamiques et mobiles que les réseaux véhiculaires est difficile mais particulièrement importante étant donné que des vies humaines sont concernées.

- ***Routage***
  Le routage dans les réseaux VANET c'est un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs. Pour que les véhicules puissent communiquer, il est nécessaire de faire passer leurs informations par d'autres qui se chargeront de les acheminer. Pour cela, il est primordial que les véhicules soient capables de construire des routes entre eux : c'est le rôle des protocoles de routage, qui permettent un acheminement optimal des informations depuis un émetteur à un destinataire précis.

- ***Normalisation vis-à-vis de la flexibilité***
  Il est évidemment nécessaire d'uniformiser les communications afin de permettre aux véhicules conçus par différents fabricants de pouvoir collaborer. Cependant, en raison des enjeux commerciaux, il est probable que les constructeurs voudront créer une certaine différenciation des standards.

- ***Canal radio fiable***
  Le rôle des mécanismes de gestion du canal radio est d'offrir des transmissions fiables et robustes et un partage équitable du médium de communication. Pour atteindre cet objectif dans le cas des réseaux véhiculaires, il est nécessaire de définir des méthodes qui permettent de faire face aux deux problèmes majeurs des transmissions qui sont, les interférences inter-symboles dues à la propagation des ondes par trajets multiples et l'effet Doppler causé par le mouvement des véhicules.

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, en détaillant quelques protocoles spécifiques tel que GPSR, DSR et DSDV.

Chapitre 2

Routage dans les réseaux véhiculaires

Introduction

Un VANET (Vehicular Ad hoc Network) est un réseau sans infrastructure fixe, constitué d'un ensemble de véhicules dénommés nœuds mobiles. Ces nœuds établissent entre eux une communication dite à un-saut lorsqu'ils sont à portée radio l'un de l'autre. Une communication dite multi-sauts peut également être établie entre deux nœuds distants (nœuds n'étant pas dans une même zone de transmission). Une communication multi-sauts est réalisable grâce à la mise en place d'un chemin de routage reliant le nœud source au nœud destinataire et impliquant un ou des nœuds intermédiaires, dits aussi nœuds relais.

Les chemins de routage multi-sauts peuvent être construits par des protocoles de routage de types variés, qui permettent d'assurer la transmission des paquets vers un (ou des) nœud(s)destinataire(s) [5].

Ce chapitre est composé de deux parties. Dans la première partie nous allons présenter le routage dans les VANETs de manière globale et classifier les différents protocoles de routage selon les deux critères : basé sur la topologie et ceux basé sur la position (géographique). Dans la deuxième partie nous allons présenter le principe et le fonctionnement des protocoles que nous avons choisi pour évaluer leurs performances.

Première partie

Protocoles de routage dans les réseaux véhiculaires

2.1 Routage dans les VANETs

Le routage joue un rôle très important dans les VANET puisque tous les services supportés, unicast ou multicast, se basent sur des communications multi-saut pour l'acheminement des données. Les communications unicast sont généralement utilisées dans les applications de confort telles que le transfert des 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, afin de déterminer les nœuds relais qui participent à l'acheminement des données. Les informations sont obtenues par des échanges de paquets de contrôle entre nœuds. La fréquence des échanges et les informations spécifiées dans les paquets diffèrent d'un protocole à l'autre tout comme les mécanismes utilisés pour déterminer les nœuds relais [2].

2.2 Classification des protocoles de routage dans les réseaux VANETs

Les réseaux VANETs utilisent plusieurs protocoles de routage qui aident à l'établissement des routes entre un groupe de nœud afin d'assurer un échange de paquet de manière continue et efficace, plusieurs articles de recherches répartissent les protocoles de routage dans les réseaux VANETs selon leurs caractéristique en deux catégories : la première catégorie est celle des protocoles qui se basent sur des informations de la topologie du réseau (Unicast) qui sont divisés en protocoles proactifs, réactifs et hybrides et les protocoles basés sur la localisation (géographique) qui utilisent la position physique des nœuds mobiles pour configurer le routage [4] [7].

2.2.1 Protocoles de routage basés sur la topologie

Les protocoles de routage de cette classe sont des protocoles où les nœuds n'ont aucune connaissance de leur position géographique ni de celle des autres nœuds. Ils utilisent les informations sur les liens qui existent dans le réseau afin d'acheminer

les paquets vers leurs destinataires. Ces protocoles découvrent la route en question, la maintiennent dans des tables de routages avant que l'émetteur ne commence à envoyer les données. Les protocoles de routage basés sur la topologie ayant trois catégories : proactifs, réactifs et hybrides, comme l'indique la FIGURE 2.1.

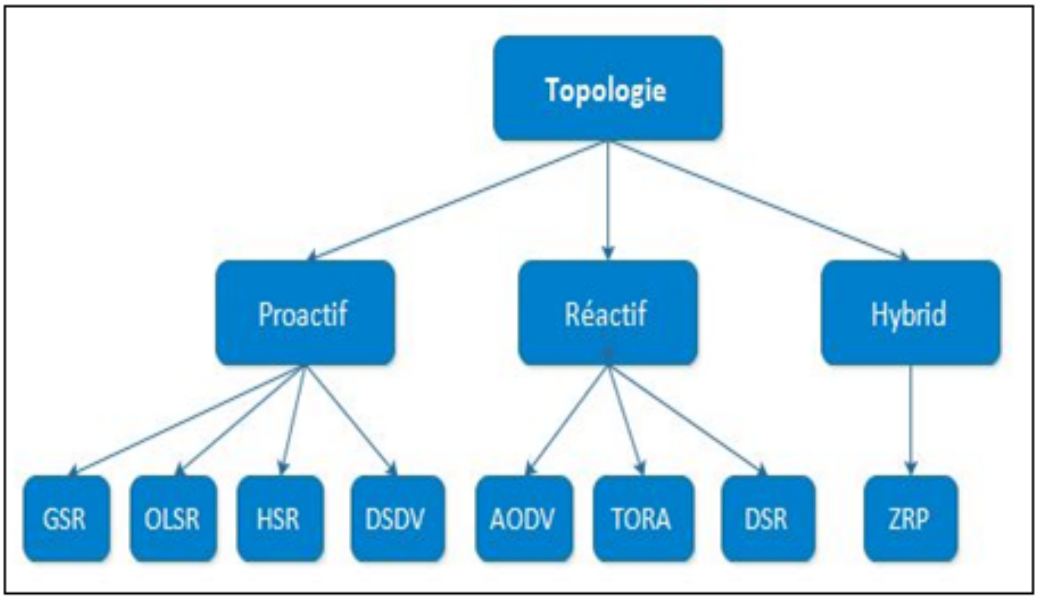

FIGURE 2.1 – *Protocoles de routage basés sur la topologie [4].*

2.2.1.1 Protocoles proactifs

Les principales méthodes utilisées par ces protocoles sont : l'état de lien (Link state), et le vecteur de distance (Distance Vector). Les protocoles proactifs maintiennent les meilleurs chemins existants vers toutes les destinations possibles. Les routes sont sauvegardées mêmes si elles ne sont pas utilisées. La sauvegarde permanente de ces routes est assurée par un échange continu des messages de mise à jour. Ceci induit un contrôle excessif, surtout dans le cas des réseaux de grande taille.

Le premier avantage de ce type de protocole, c'est que les routes sont immédiatement disponibles quand les nœuds ont en besoin. Cependant, les échanges réguliers de messages de mise à jour engendrent une consommation inutile de la bande passante [4]. Il existe plusieurs protocoles proactifs, tels que :

- ***GSR(Global State Routing)*** : Dans le protocole GSR, chaque nœud maintient essentiellement la liste de voisinages, la table de la topologie du

réseau, la table des nœuds suivants qui contient l'adresse du nœud suivant, retransmetteur du paquet vers chaque nœud destinataire,enfin la table de distance comportant le chemin le plus court vers chaque destination. Lors d'un changement des états des liens dans le réseau et grâce aux messages de contrôle diffusés dans le réseau, toutes les tables maintenues sont mises à jour. Mentionnons également que les mises à jour sont appliquées uniquement lorsque le numéro de séquence est supérieur au numéro de séquence précédemment sauvegardée dans la table [**?**].

- ***OLSR (Optimized Link State Routing Protocol)*** : Est une optimisation de l'algorithme à état de liens. Ce protocole utilise aussi l'algorithme du plus court chemin pour la sélection des routes. Dans l'algorithme à état de liens, chaque nœud du réseau déclare son voisinage à tout le réseau. Par contre dans OLSR, les nœuds ne déclarent qu'un sous-ensemble de leur voisinage grâce à la technique des relais multipoints (MPR) [8].

- ***HSR (Hierarchical State Routing)*** : Dans le protocole HSR, le réseau est partitionné en un ensemble de groupes, dont l'union forme le réseau entier. Dans chaque groupe, un nœud est élu responsable pour gérer le groupe et assurer la communication avec d'autres représentants des autres groupes. Dans la décomposition en groupes, nous distinguons 3 types de nœuds : des nœuds représentants de groupe (appelés aussi Cluster Heads), des nœuds liaison qui relient deux groupes, et des nœuds internes n'ayant aucun rôle spécifique au sein du groupe [22].

- ***DSDV (Destination Sequence Distance Vector)*** : Est un protocole proactif unicast mobile ad hoc qui est basé sur l'algorithme de Bellman-Ford. Dans les tables de routage de DSDV on trouve :
  - Toutes les destinations possibles.
  - Le nombre de nœuds (ou de sauts) nécessaire pour atteindre la destination.
  - Le numéro de séquences (SN) qui correspond à un nœud destinataire.

Les numéros de séquence sont utilisés dans DSDV pour distinguer les anciennes et nouvelles routes et pour éviter la formation de boucles de parcours. Chaque nœud transmet périodiquement des mises à jour, y compris des informations de routage à ses voisins immédiats [14].

2.2.1.2 Protocoles réactifs

Les protocoles de routage réactif sont caractérisés par le fait que, contrairement aux protocoles proactifs les nœuds ne tiennent pas à jour de tables de routage. Lorsqu'un nœud souhaite contacter un autre nœud, il fait une demande de route à ses voisins, jusqu'à tomber sur un nœud connaissant l'information. Le principal inconvénient est le temps nécessaire pour obtenir l'information sur une route. Par contre, l'avantage est que ce type de protocole réagit très bien aux changements de topologie : le nœud faisant la demande de route obtient toujours une route à jour [35].

Dans ce qui suit, nous allons présenter les protocoles réactifs les plus connus dans le réseau VANET.

- ***AODV (Ad hoc On-demand Distance Vector)*** : Le protocole de routage AODV est un protocole réactif multi-saut et très utilisé dans les réseaux 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 qu'il 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 [17].

- ***TORA (Temporally-Ordered Routing Algorithm)*** :Le protocole TORA [14] est un protocole de routage adapté aux réseaux à topologie changeante de façon rapide et aléatoire. Il est conçu pour découvrir des routes à la demande, aussi fournir des voies multiples vers une destination et de minimiser la surcharge de la communication réseau par la localisation des changements topologiques lorsque cela possible. Le protocole TORA dispose de trois opérations principales : la création des routes, la maintenance des routes, et la suppression des routes. Trois paquets sont utilisés à cet effet :
  - Requête QRY (Query) : pour la création des routes.
  - Mise-à-jour UPD (Update) : pour la création et la maintenance des routes.
  - Suppression CLR (Clear) : pour la suppression des routes.

- ***DSR (Dynamic Source Routing)*** : DSR fonctionne de la même manière que AODV, c'est un protocole réactif qui n'utilise aucun échange périodique d'informations de contrôle, seules les routes actives sont maintenues dans les tables de routage. La différence entre les deux protocoles réside dans le fait que, dans AODV la route est déterminée de proche en proche, alors que DSR utilise la technique de routage par la source. Dans cette technique, le nœud source détermine la suite complète de nœud à traverser qu'il insère dans chaque paquet de données avant sa transmission. Cette séquence est utilisée par chaque nœud de la route pour déterminer le prochain relai vers le destinataire [23].

2.2.1.3 Protocoles hybrides

Les protocoles de routage hybrides combinent les avantages des deux protocoles de routage proactifs et réactifs. Le routage est établi initialement avec des itinéraires réalisés au préalable de manière proactive puis répond à la demande et de manière réactive aux autres nœuds. Un des protocoles de routage hybrides est ZRP (Zone Routing Protocol) [11].

- ***ZRP (Zone Routing Protocol)*** : Le protocole de routage ZRP divise le réseau en différentes zones. Pour chaque nœud, il définit une zone de routage exprimée en nombre de sauts maximal ainsi, la zone de routage d'un nœud inclut tous les nœuds qui sont à une distance au maximum de sauts. Les nœuds qui sont exactement à sauts sont appelés nœuds périphériques. À l'intérieur de cette zone, ZRP utilise un protocole proactif et à l'extérieur de cette zone de routage, il fait appel à un protocole réactif [13] [14].

2.2.2 Protocoles de routage basés sur la position

Dans ce type de protocole, chaque nœud connait la position géographique de ses nœuds voisins. Ceci, à l'aide d'un système de localisation tel que le GPS (Globale Position System). Ce type de protocole n'échange aucune information sur l'état des liens avec ses voisins et ne maintient aucune table de routage. Lorsqu'un nœud source désire envoyer un paquet vers un nœud destinataire, les informations provenant du système de localisation sont stockées dans l'en-tête de la trame. Ces informations servent à transmettre le paquet à son destinataire sans voir recours à la procédure de découverte ou de maintenance des routes [28] [10].

Nous allons présenter quelque exemple de ces protocoles dans la FIGURE 2.2 suivante :

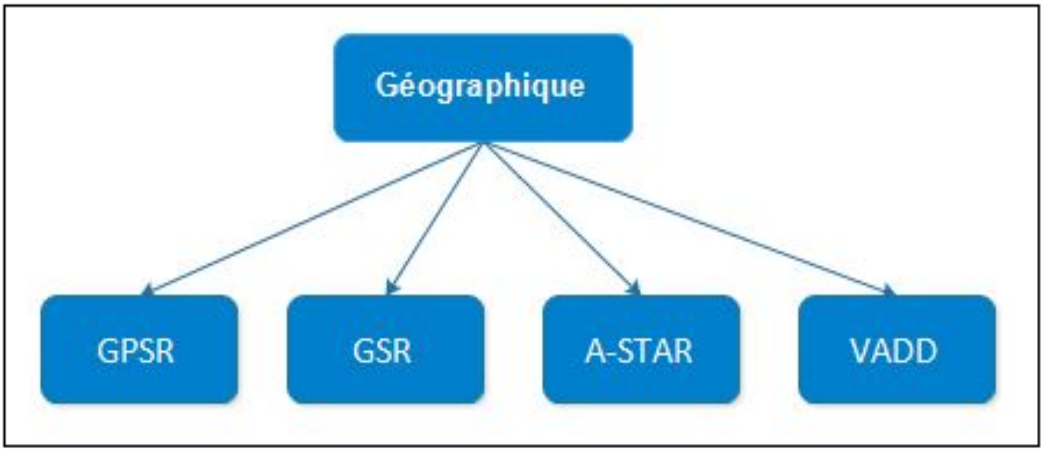

FIGURE 2.2 – *Protocoles de routage basés sur la position.*

- ***GPSR (Greedy Perimeter Stateless Routing)*** : Le protocole de routage GPSR c'est le plus connu dans les réseaux VANETs qui exploite la correspondance entre la position géographique et la connectivité dans un réseau sans fil afin de prendre des décisions de transfert de paquets. La transmission des paquets dans le GPSR se fait grâce à deux méthodes : le greedy forwaeding utilisé pour envoyer des données vers le nœud le plus proche du destinataire qui se trouve en portée radio. Et la deuxième méthode Perimiter Forwarding, le protocole GPSR fait appelle à cette technique de transmission quand la méthode par défaut n'aboutit pas [25] [16].

- ***GSR (Geografic Source Routing)*** : Est un autre protocole basé sur la localisation pour les réseaux VANETs. GSR suppose l'existence d'une carte numérique du réseau routière. Cette carte est utilisée pour connaître la topologie de la ville. GSR utilise un mécanisme appelé RLS (Reactive Location Service) pour avoir la position d'une destination. GSR combine le routage géographique avec la connaissance de la topologie : l'émetteur détermine les intersections qui doivent être traversées par le paquet et ce dernier est acheminé entre chaque deux intersections avec du routage basé sur la localisation. Le protocole est destiné aux environnements urbains [29].

- ***A-STAR(Anchor-based Street and Traffic Aware Routing)*** : Est un protocole de routage basé sur la position pour un environnement véhiculaire métropolitain. Il utilise particulièrement les informations sur les itinéraires d'autobus de ville pour identifier une route d'ancre avec une connectivité élevée pour l'acheminement des paquets. A-STAR est similaire au protocole GSR en adoptant une approche de routage basée sur l'ancrage qui tient compte des caractéristiques des rues. Cependant, contrairement à GSR il calcule les "anchorpaths" en fonction du trafic (trafics de bus, véhicules, etc..). Un poids est assigné à chaque rue en fonction de sa capacité (grande ou petite rue qui est desservie par un nombre de bus différent). Les informations de routes fournies par les bus donnent une idée sur la charge de trafic dans chaque rue. Ce qui donne une image de la ville à des moments différents [36].

- ***VADD(Vehicle-Assisted Data Delivery)*** : Est un protocole de routage qui prend en considération le contexte des réseaux de véhicules et exploite le mouvement prévisible des véhicules pour décider de retransmettre ou non le message. Il utilise particulièrement les informations sur le trafic routier au niveau d'une route pour estimer le délai mis par un paquet pour parcourir un tel segment. Par conséquent, les paquets seront acheminés le long d'un chemin ayant le plus faible délai de bout en bout [43].

  Dans ce qui suit nous détaillons le fonctionnement de trois protocoles GPSR, DSR et DSDV.

Deuxième partie

Fonctionnement des protocoles GPSR, DSR et DSDV

2.3 Protocole GPSR (Greedy Perimeter Stateless Routing)

GPSR est un protocole de routage géographique (Position Based). Dans un réseaux VANET, les nœuds sont capables de se déplacer, il est donc nécessaire d'utiliser un mécanisme qui permet à chaque nœud de connaître la position de ses voisins, afin de signaler leur présence et leur localisation.

Le protocole GPSR utilise deux mécanismes pour transmettre un paquet à son destinataire : le « Greedy Forwarding » et le « Perimeter Forwarding » [25] [20] .Dans ce qui suit nous allons présenter et détailler ces deux mécanismes.

2.3.1 Greedy Forwarding

Greedy Forwarding c'est un mécanisme appelé GF il permet de construit un chemin grâce aux nœuds qui se trouvent entre la source et la destination. Chacun des nœuds intermédiaires peut faire localement un choix gourmand (greedy) optimal dans le choix du prochain saut pour un paquet. Un nœud de retransmission/a' relayage achemine le paquet qu'il reçoit et le fait parvenir à son voisin le plus proche géographiquement à la destination du paquet. Ce mécanisme est répété de manière récursive jusqu'à ce que la destination soit atteinte.la FIGURE 2.3 ci-dessus montre le fonctionnement de cette technique.

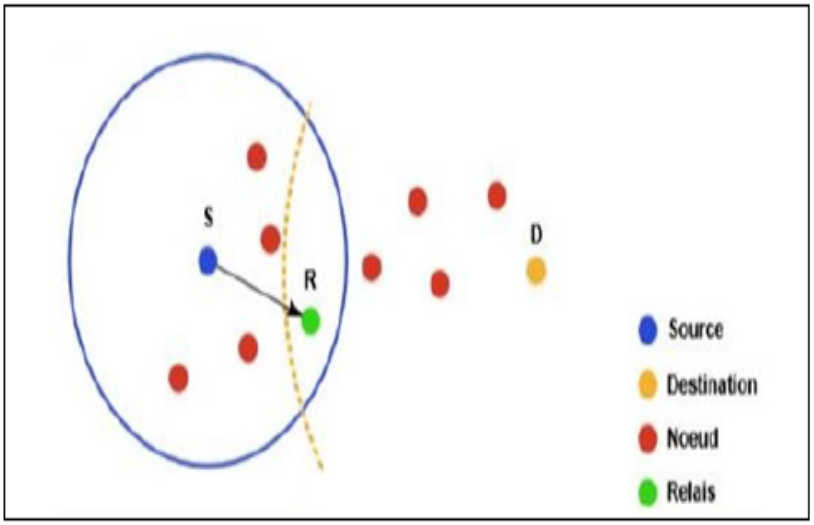

FIGURE 2.3 – *Technique du Greedy Forwarding [25].*

La FIGURE 2.3 représente un exemple de choix gourmand du prochain saut. Ici, le nœud S reçoit ou génère un paquet destiné à D. La zone de couverture de S est désignée par le cercle plein de couleur bleue autour de S et l'arc dont le rayon est égal à la distance entre R et D est représenté par une ligne discontinue. S relaye le paquet à R, car la distance entre R et D est inférieure à celle entre D et n'importe quel autre voisin de S. Ce processus de transfert gourmand se répète jusqu'à ce que le paquet atteigne D.

Afin qu'un nœud puisse connaitre la position de ses voisins et construire sa table de voisinage, un algorithme dit Beaconing algorithme est utilisé. Chaque nœud informe son voisin de sa position, et signale sa présence grâce à des messages de contrôle(messages Hello) qui contiennent la position et l'identifiant du nœud. Cet échange périodique de paquets permet de construire une table qui contient l'ID du nœud etsa position.

La période d'émission des messages Hello dépend du taux de mobilité dans le réseau ainsi que de la portée radio des nœuds. En effet, lorsqu'un nœud ne reçoit pas de messages Hello d'un voisin après un temps T, il considère que le voisin en question n'est plus dans sa zone de couverture et l'éfface de sa table de position. Cependant, Il existe des topologies où cette stratégie peut échouée dans le cas où il n'existe pas de nœuds voisins proche de la destination que le nœud lui-même. Un exemple d'une telle topologie est représentée dans la FIGURE 2.4 suivante :

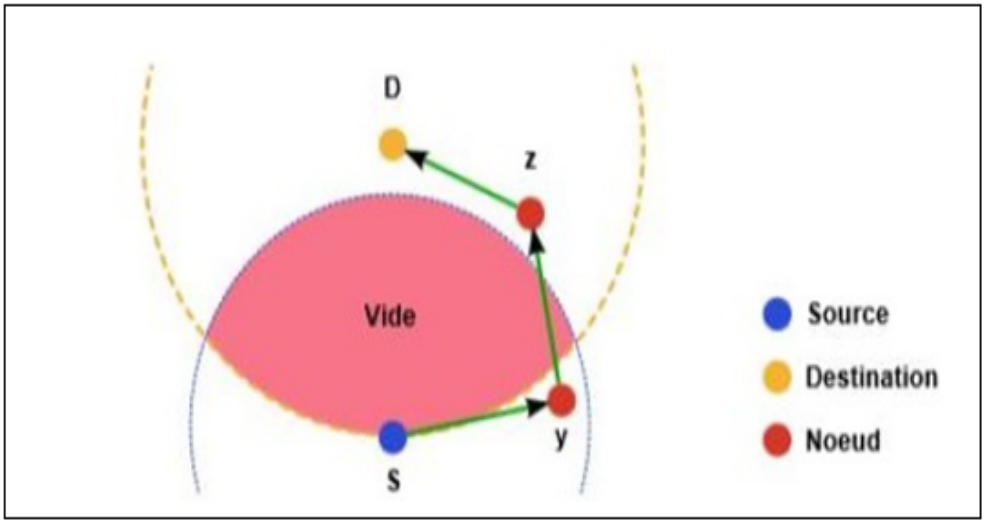

FIGURE 2.4 – *Scénario d'un maximum local [25].*

Sur cette figure, S est plus proche de D que son voisin y. Ainsi le chemin ($S \longrightarrow Y \longrightarrow Z \longrightarrow D$) existe vers D, S ne choisira pas de transmettre à y avec la technique du Greedy Forwarding. Pour résoudre ce problème, un autre mécanisme doit être utilisé pour transmettre les paquets dans ce genre de situations.

2.3.2 Perimeter Forwarding ou règle de la main droite

La méthode Perimeter Forwarding est utilisée dans le cas où l'algorithme Greedy Forwarding a échoué. C'est à dire quand il n'existe pas de nœuds voisins proches de la destination que le nœud lui-même. Le mode Perimiter Forwarding fait appel à la règle de lamain droite. Lorsqu'un paquet arrive à un nœud S, le chemin à suivre est le prochain qui se trouve dans le sens inverse des aiguilles d'une montre en partant de S et par rapport au segment [Source Destination] tout en évitant les liens déjà parcoururent.

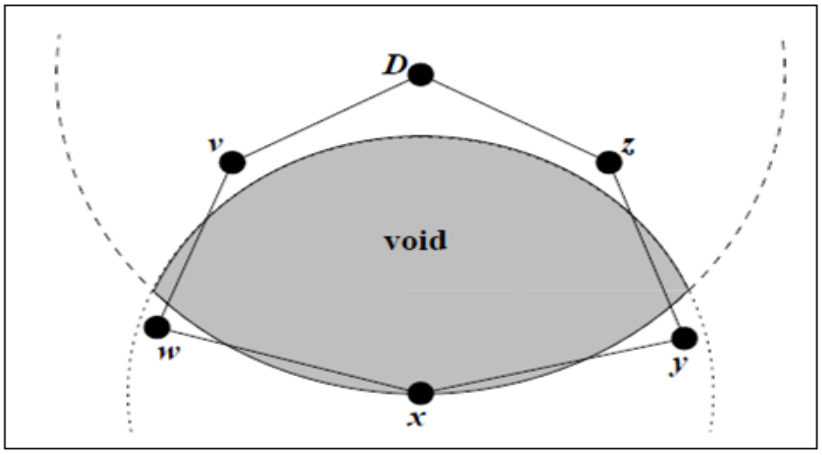

FIGURE 2.5 – *Exemple d'utilisation de la règle de la main droite [25].*

On constate dans la FIGURE 2.5 que l'intersection du cercle qui définit la portée radio de nœud émetteur "X" et celle du destinataire "D" ne contient aucun nœud voisin. Dans ce cas, le protocole fait appel à la méthode de la main droite à fin d'envoyer le paquet reçu. L'application de cette règle sur l'exemple de cette FIGURE donne le chemin suivant : $x \longrightarrow w \longrightarrow v \longrightarrow D \longrightarrow z \longrightarrow y \longrightarrow x$. Cette combinaison de nœud traversée est appelée Périmètre. Le paquet est alors envoyé par la route :$x \longrightarrow y \longrightarrow z \longrightarrow D$ *ou* $x \longrightarrow w \longrightarrow v \longrightarrow D$.

2.4 Protocole DSR (Dynamic Source Routing)

Le protocole DSR est basé sur l'utilisation de la technique "routage source". Dans cette technique, le trajet parcouru par le paquet est inclus dans l'en-tête du paquet de données à partir de la source. Les deux opérations de base du protocole DSR sont : la découverte de routes (route discovery) et la maintenance de routes (route maintenance). Un nœud source qui veut atteindre un nœud cible débute l'opération de découverte par la diffusion d'un paquet requête de route. Si cette opération de découverte est réussite, le nœud initiateur reçoit un paquet réponse de route qui liste la séquence de nœuds à travers lesquels la destination peut être atteinte. En plus de l'adresse de la source, le paquet requête de route contient un champ enregistrement de route, dans lequel est accumulée la séquence des nœuds visités durant la propagation de la requête de route dans le réseau voir la FIGURE 2.6 (a). Le paquet requête de route, contient aussi un identificateur unique de la requête. Dans le but de détecter les duplications de réceptions de la requête de route, chaque nœud du réseau ad hoc maintient une liste de couples (adresse source, identificateur de requête), des requêtes récemment reçues. Pour réduire le coût de la découverte de routes, chaque nœud garde les chemins appris à l'aide des paquets de réponses FIGURE 2.6 (b). Ces chemins sont utilisés jusqu'à ce qu'ils soient invalides [24].

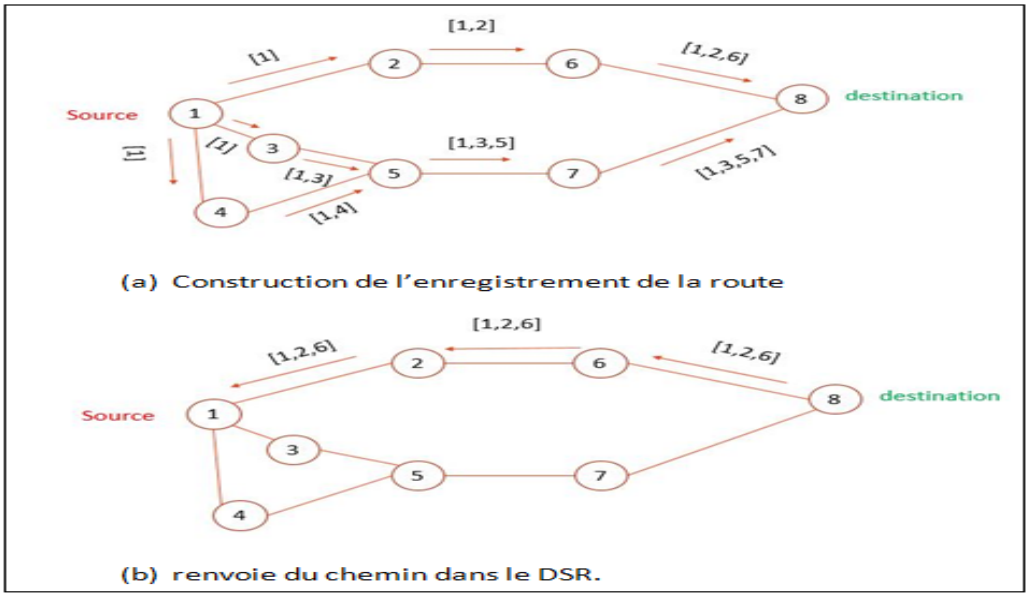

FIGURE 2.6 – *Découvert de la route dans DSR [24].*

2.5 Protocole DSDV (Destination Sequence Distance Vector)

Le protocole DSDV est basé sur l'algorithme distribué de Bellman-Ford qui utilise les vecteurs de distance. Dans ce cas, la métrique utilisée est le nombre de sauts. La FIGURE 2.7 montre un exemple de réseau ayant comme métrique le nombre de sauts. Les routes choisies sont celles présentant le moins de sauts.

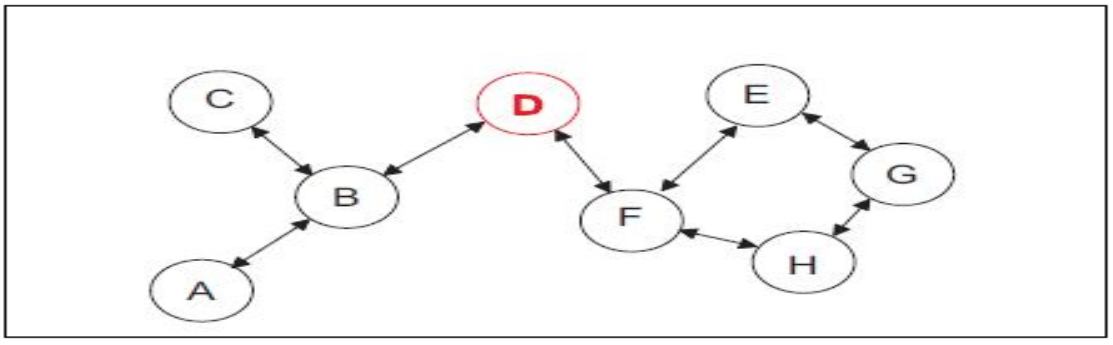

FIGURE 2.7 – Exemple de réseau utilisant le protocole DSDV.

Chaque nœud du réseau maintient une table de routage à jour comportant tous les nœuds du réseau joignable, le nœud intermédiaire suivant sur la route, le nombre de sauts, et le numéro de séquence de la destination.

La Table 2.1 représente la table de routage du nœud D du réseau de la FIGURE 2.7. Le numéro de séquence permet d'éviter les phénomènes de boucle en ne conservant que l'information la plus récente (numéro de séquence le plus élevé). Les tables de routage sont mises à jour périodiquement mais également lors d'événements particuliers (la découverte d'une route invalide, la mobilité d'un nœud...). Sur des réseaux de grande dimension, la mise à jour des tables en cas de mobilité des nœuds peut être lente car elle est initiée par la destination.

Destination	Saut suivant	Métrique	Sequence
D	D	0	S344_D
A	B	2	S120_A
B	B	1	S550_B
C	B	2	S700_C
E	F	2	S390_E
F	F	1	S072_F
G	F	3	S120_G
H	F	2	S050_H

TABLE 2.1 – Table de routage d'un noeud de réseau DSDV.

2.6 Différents avantages et inconvénients des trois protocoles

La TABLE 2.2 suivante montre les différents avantages et inconvénients pour les trois protocoles GPSR, DSR et DSDV.

Protocole	Avantage	Inconvénient
GPSR	– la source connait sa position, la position de ses voisins et la position du destinataire en question. – Il marche mieux sur les grandes routes avec un partage égal des nœuds, car sans obstacles il obtient de très bonnes performances.	– le "Greedy Forwarding" est restreint à cause des obstacles. – Les paquets sont parfois envoyer dans de mauvaises direction, ce qui augmente les délais.
DSR	– Découvre les routes à la demande en inondant le réseau avec un paquets de requête.	– La taille des paquets de données très grande quand le nombre de nœud dans réseau est grand.
DSDV	– Fournit à tout moment des routes valables ver toutes les destinations du réseau.	– L'inondation des paquets de mise à jour cause une charge de contrôle importante au réseau.

TABLE 2.2 – Tableau comparatif.

Conclusion

Dans ce chapitre nous avons présenté le routage dans les réseaux VANETs. Le routage joue un rôle très important et un bon protocole de routage est celui qui est capable de livrer un paquet dans un temps très court et un minimum de bande passante. Ensuite, nous avons présenté la classification des protocoles selon différents critères.

Ainsi nous avons présenté le mode de fonctionnement de quelques protocoles (GPSR ,DSR et DSDV).

Dans le prochain chapitre, nous allons évaluer les performances de ces trois protocoles.

Chapitre 3

Simulation et évaluation des protocoles GPSR, DSR, DSDV

Introduction

L'existence d'un réseau véhiculaire ouvre la voie à une large gamme d'applications pour résoudre plusieurs problèmes de circulation et l'évaluation de protocoles et d'applications composées d'un grand nombre de nœuds VANETs, est seulement possible en utilisant des outils de simulation qui permettent de simuler une topologie dans laquelle il est possible de faire transiter des données.

Les véhicules dans les réseaux VANETs ne peuvent se déplacer que sur les routes ou les chemins définis dans cette topologie et doivent obéir aux règles de circulation, et d'autre part, par la prise en compte de l'interaction entre véhicules. Alors, il y'a une liberté dans leurs mouvements et ils sont limités par les signalisations routières telles que des panneaux d'arrêt, feux de circulation, etc.

Dans ce dernier chapitre nous allons présenter l'environnement de simulation et la configuration des paramètres d'entré, ainsi que les critères d'évaluation. Enfin l'interprétation des résultats des simulations.

3.1 Environnement de simulation des VANETs

L'émergence de réseaux de véhicules a encouragé la conception d'un ensemble de nouvelles applications et de protocoles spécifiquement pour ces types de réseaux. L'évaluation de ces protocoles en plein air, en utilisant des réseaux à grande échelle pour obtenir des résultats significatifs, est extrêmement difficile en raison des coûts de construction, c'est pourquoi la simulation est devenue un outil indispensable car il permet de modéliser un VANET et ensuite de récupérer des données statistiques sur l'utilisation du réseau au cours de la simulation afin de mesurer les performances des protocoles.

Nous allons présenter deux types de simulateurs : les simulateurs de trafic routier sont utilisés pour la réalisation de la carte du réseau routier et la génération de la mobilité souhaitée (types de véhicules, vitesses, évènements de la route, etc.), et les simulateurs réseaux se chargent d'évaluer les performances des protocoles et les différents services réseaux. Pour réaliser la simulation nous allons utiliser les outils suivant : JOSM, eWorld, SUMO et NS-3.

3.1.1 JOSM (JavaOpen Street Maps)

C'est une application Java qui permet d'exploiter les services de cartes libre OSM (Open Stree tMaps). JOSM permet entre-autres la récupération, la modification et la mise à jour des cartes sur OSM, dans ce travail nous avons eu recours à JOSM pour l'extraction de la carte des deux quartiers de béjaia.

FIGURE 3.1 – *JOSM.*

3.1.2 eWorld

C'est un projet réalisé par des étudiants allemands de l'institut Hasso Plattner. C'est un outil qui permet d'importer des cartes proposées par des fournisseurs tels que (Open Street Map), les visualiser, les éditer et enrichir avec des événements, puis l'exporter vers un simulateur de trafic routier [6].

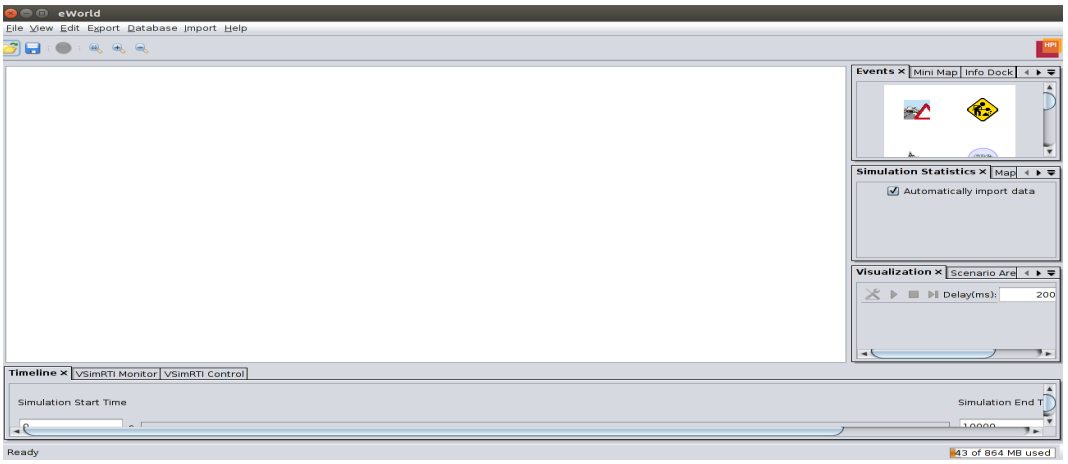

FIGURE 3.2 – *eWord* .

3.1.3 SUMO (Simulation of Urban MObility)

Le logiciel ouvert SUMO est le simulateur de trafic gratuit le plus communément utilisé dans les récents travaux de recherche sur les réseaux véhiculaires. C'est un simulateur hautement portable capable de fournir des schémas de mobilité précis. Il prend en compte plusieurs types de véhicules, les feux de signalisation, les intersections avec priorité (priorité à droite par exemple), le changement de voies, les connexions de voie à voie, etc. SUMO utilise le modèle du carfollowing Krauss Model pour simuler le comportement des conducteurs et Random Way point pour le mouvement sur les routes [37].

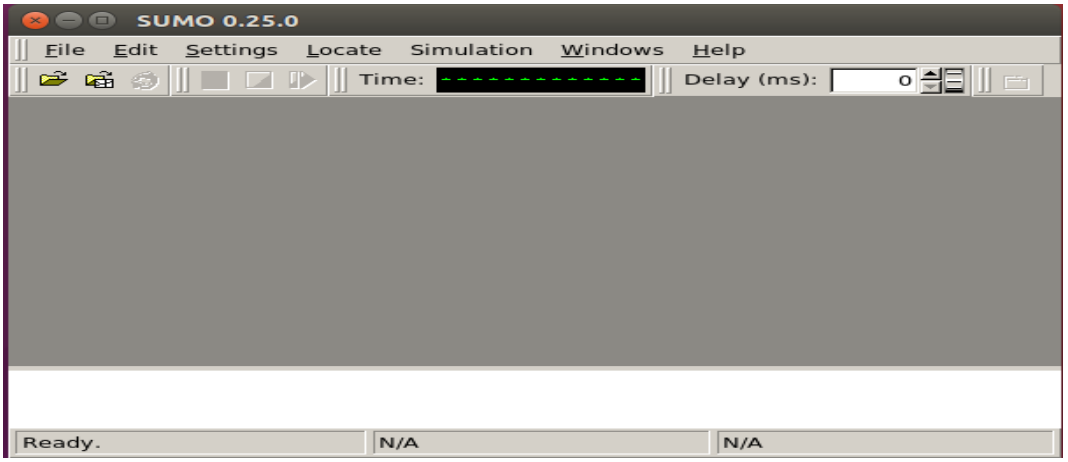

FIGURE 3.3 – *Sumo.*

3.1.4 NS-3 (Network Simulator 3)

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 représentants 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 3000 nœuds [32].Voir le tableau comparatif entre les différents simulateurs TABLE 3.14, Annexe V.

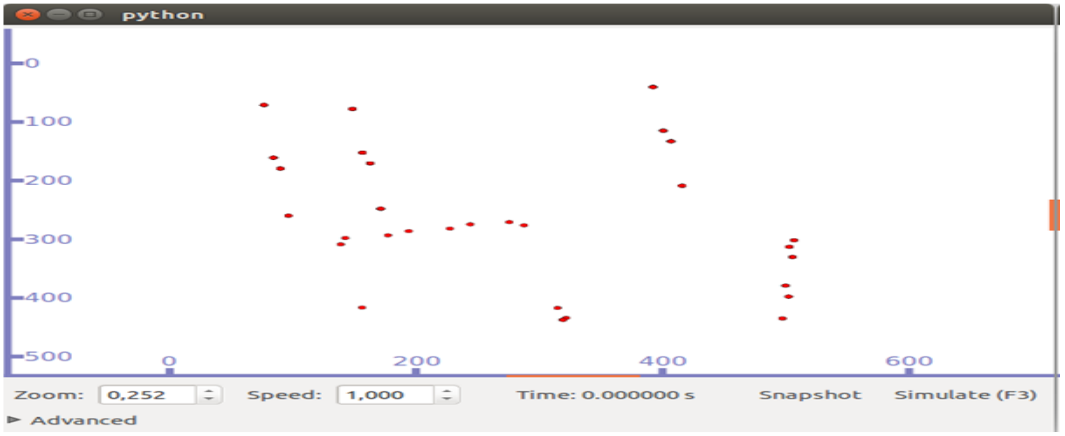

FIGURE 3.4 – *NS-3.*

3.2 Configuration des paramètres d'entrées

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, et un fichier comportant les données de déplacement des nœuds durant la simulation, et pouvant être exécuté dans l'interface graphique de NS-3. Ce processus est détaillé dans les paragraphes suivants.

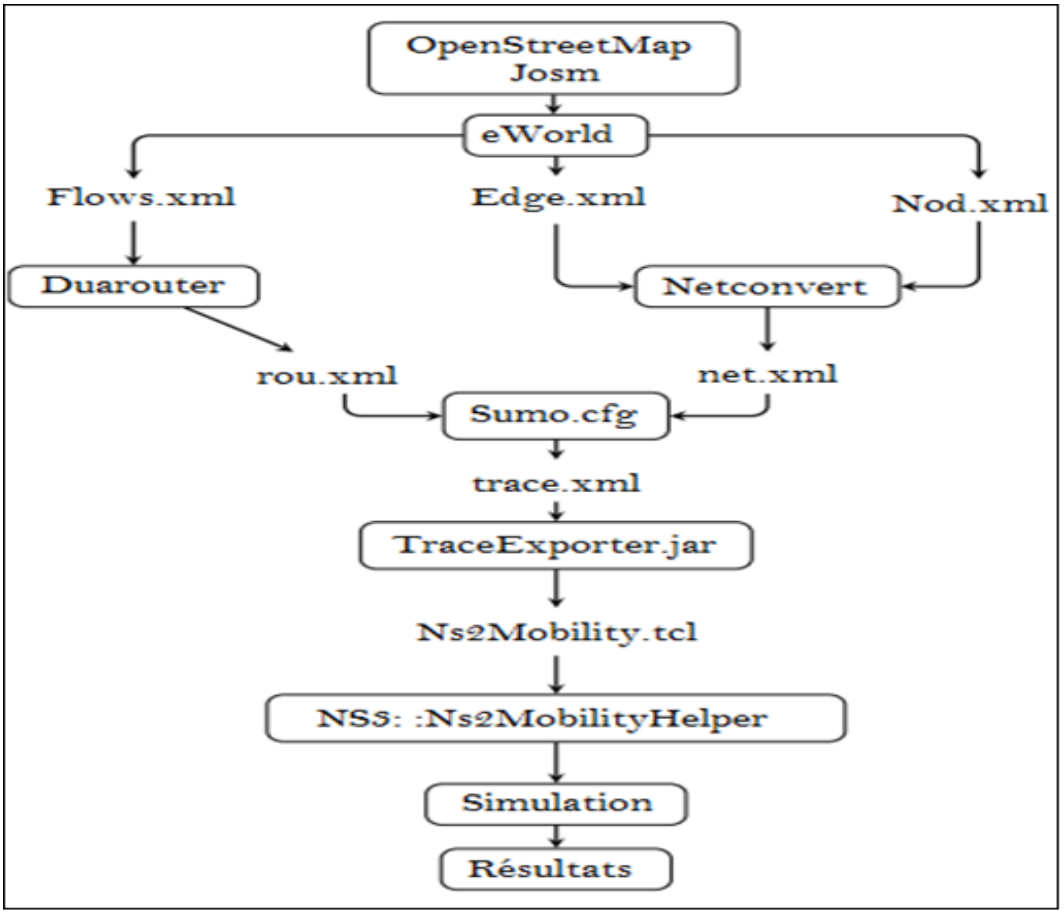

FIGURE 3.5 – *Schéma récapitulatif des étapes de la simulation [32].*

3.2.1 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.

3.2.1.1 Préparation de la carte avec eWorld

Dans cette étape, nous avons importé une carte de la ville de Bejaia (Aamriw Cité dallas) 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 défini les types des véhicules (nœuds) et les intersections à feux de circulation et ceux à priorité.

3.2.1.2 Création des scénarios avec SUMO

Les fichiers .xml (nod.xlm, edg.xml et flows.xml) générés à partir de eWorld contiennent les données du réseau routier, ces informations sont utilisées pour crées un scénario avec SUMO.

- ***Node file*** : C'est un fichier composé d'un ensemble de noeud du réseau routier, ces derniers sont définis par les coordonnées x et y et un identificateur (id).

```
<nodes>
<node id="f5c0f2ef-bda1-41f6-9b1f-5d58839739dd" x="683565.0875753529" y="4068494.394027818" type="priority" />
<node id="02c3be64-fa5a-4b3b-92e1-18f60cf68d92" x="683650.0336496166" y="4068568.461753604" type="priority" />
<node id="19434db3-229f-4042-90a3-837615d47497" x="683780.7023073143" y="4068816.7336738934" type="priority" />
<node id="dacc385e-26b8-4a0d-982f-1e2fa176ea28" x="683780.936049643" y="4068810.4345484837" type="priority" />
<node id="bdb3e916-c79d-42f9-bb6a-1b7b02621169" x="683467.6332438139" y="4068549.3933440167" type="priority" />

k/nodes>
```

FIGURE 3.6 – *Node.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.

FIGURE 3.7 – *Edg.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 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 (a savoir une route). Spécifiant le nœud de départ et le nœud d'arrivée, le nombre de véhicules, temps de début et temps de fin.

```
<flows>
 <vtype id="DefaultVehicle" accel="0.9" decel="1.5" sigma="0.5" length="3.0" maxspeed="8.0" 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"/>
 <flow id="3" from="c23057eb-2a5b-4019-995a-ce13f2e635ca" to="a7e91e56-3479-4b30-b71e-49e656fd1f23" type="DefaultVehicle" begin="0"
end="600" number="25"/>
 <flow id="4" from="3e15caa4-8e54-4040-8cad-d4bb69bbaffb" to="c0d91be1-8e44-4c6a-8814-d7cd811f1d0f" type="DefaultVehicle" begin="0"
end="600" number="8"/>

</flows>
```

FIGURE 3.8 – *Flows.xml.*

- ***Route file*** : C'est un fichier généré à partir de la commande DUAROUTER qui prend en argument deux fichiers : net.xml et flow.xml. La commande DUAROUTER permet de calculer les itinéraires les plus rapides à travers le réseau.

duarouter −−flows=nom-fichier.flows.xml ˘−net=nom-fichier.net.xml −−output-file=newnom-fichier.rou.xml.

- ***Sumocfg file*** : C'est le fichier résultant des deux fichiers Route file et Net file, Sumo.cfg est donc le fichier de configuration de SUMO.

```
 <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>
```

FIGURE 3.9 – *Sumo.cfg.*

3.2.2 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 on va créer un fichier trace pour sumo qui est ***sumoTrace.xml*** en utilisant la commande suivante :

```
sumo -c nom-fichier.sumo.cfg –˘fcd-
output  sumoTrace.xml.
```

En deuxième lieu, nous allons convertir 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 renvois trois fichiers de format tcl : ***config.tcl, mobility.tcl et activity.tcl.***

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

Puis nous utilisons la commande ci-dessous permettant la visualisation de la simulation d'un protocole exemple GPSR :

```
Sudo ./waf −−run GPSR −−vis
```

Après l'execution de la commande précedente nous obtienderons le résultat représenté par la FIGURE 3.10.

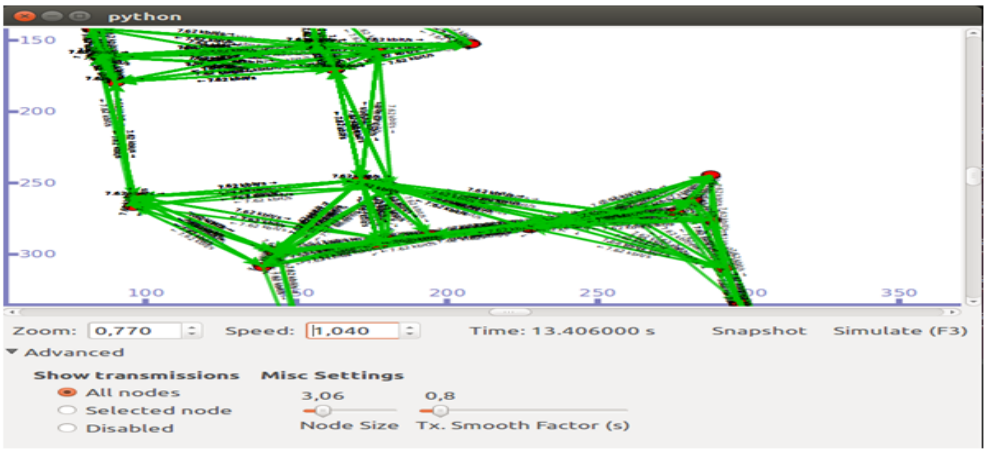

FIGURE 3.10 – *Simulation de protocole GPSR.*

3.2.3 Collecte d'informations

La simulation peut prendre quelques minutes pour GPSR et DSR, voir des dizaines de minutes pour le protocole DSDV. 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. Une pour générer le trafic, une autre pour faire des calculs.

3.2.3.1 Fonction de génération du trafic

Objectif de cette fonction est d'envoyer les paquet. Ceci pour calculer le délai moyen de bout en bout et pour récupéré le temps où le première paquet est envoyé.

3.2.3.2 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.

- 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, nous utilisons 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.

3.3 Critères d'évaluation

Notre objectif dans ce mémoire réalisé sous NS-3 est d'évaluer les performances des trois protocoles de routages GPSR, DSR et DSDV selon les métriques suivantes.

3.3.1 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 delivraison 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 [21][3].Cela est traduits mathématiquement par l'équation i :

$$\mathbf{PDR} = \frac{\sum nbr\ de\ paquets\ recus\ par\ la\ destination}{\sum nbr\ depaquets\ envoyes\ par\ tous\ les\ noeuds\ source} \qquad (i)$$

3.3.2 Délai de bout en bout

Le délai moyen de bout en bout, c'est-à-dire la mesure du délai entre l'envoidu message par le nœud source et sa réception par le nœud destinataire, est le paramètre principal que l'on vise à améliorer lors de l'évaluation d'un protocole de routage. Un bon protocole à des délais moyens de bout en bout qui sont les plus bas possible [21][3]. Le délai moyen de bout en bout d'un paquet i entre une paire de nœud Source-Destinataire est :

$$D = (T_{ri} - T_{si}) \qquad (ii)$$

Avec D le délai moyen de bout en bout, Trile temps au moment de la réception du paquet par le nœud destinataire r, et Tsi le temps au moment de l'émission du paquet par le nœud source S. On fait ensuite la moyenne pour chacun des paquets envoyés par chaque pair de nœuds tout au long de la simulation, afin d'obtenir le délai moyen de bout en bout [21][3]. Mathématiquement il peut être démontré sous l'équation iii :

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

Avec n, le nombre total de paquets reçus pendant la simulation.

3.3.3 Nombre de Paquets perdus

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

$$D = \frac{nbr\,paquets\,envoyes - nbr\,paquets\,recus}{nbr\,paquet\,envoyes} \qquad (iv)$$

3.3.4 Débit moyen

L'unité de mesure de cette métrique est paquet/unité TIL Time IntervalLenght, où TILest la longueur de l'intervalle de temps [21][3]. Cela est traduit mathématiquement par l'équation v :

$$debit_{moyen} = \frac{taille\,du\,paquet\,recu}{(temps\,final\ -\ temps\,initial)} * \frac{8}{100} \qquad (v)$$

3.3.5 Résultats des simulations

3.3.5.1 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 3.11, 3.12, 3.13, 3.14 et 3.15.

Taux de livraison des paquets

La FIGURE 3.11 montre que le taux de livraison de paquets délivré pour les deux protocoles de routages (GPSR et DSR), varie de 15% à 90% en fonction de variation de nombres de nœuds. Par contre le protocole DSDV reste toujours dans le même rythme de 25%.Et cela est dûs moment que le nombres de paquets émis et reçus dans le protocole GPSR est moindre que les deux autres.

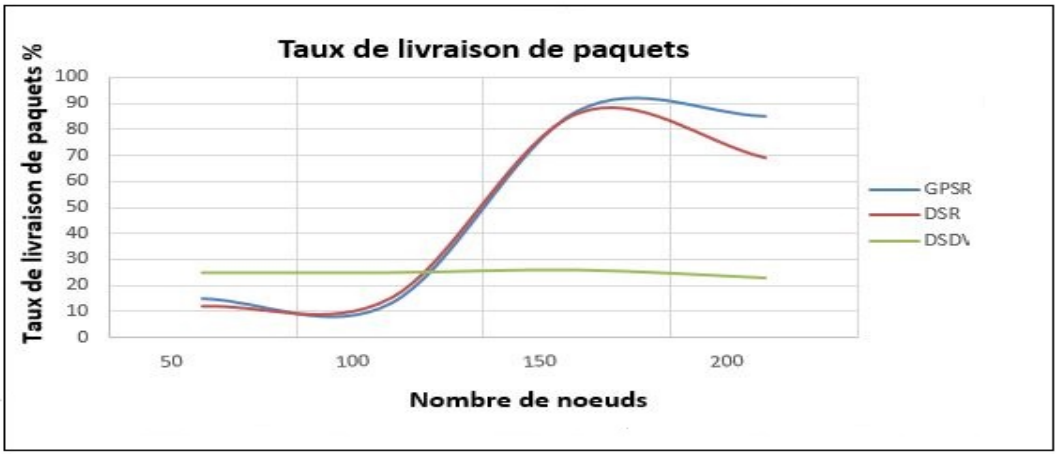

FIGURE 3.11 – *Taux de livraison de paquets.*

Débit

D'après la FIGURE 3.12 qui représente le débit en fonction du nombre de nœuds, on constate que le débit de GPSR est nettement au-dessus des deux autres protocoles DSR et DSDV.Et cela revient au fait que la taille des paquets dans GPSR et plus petite que chez les deux autre.

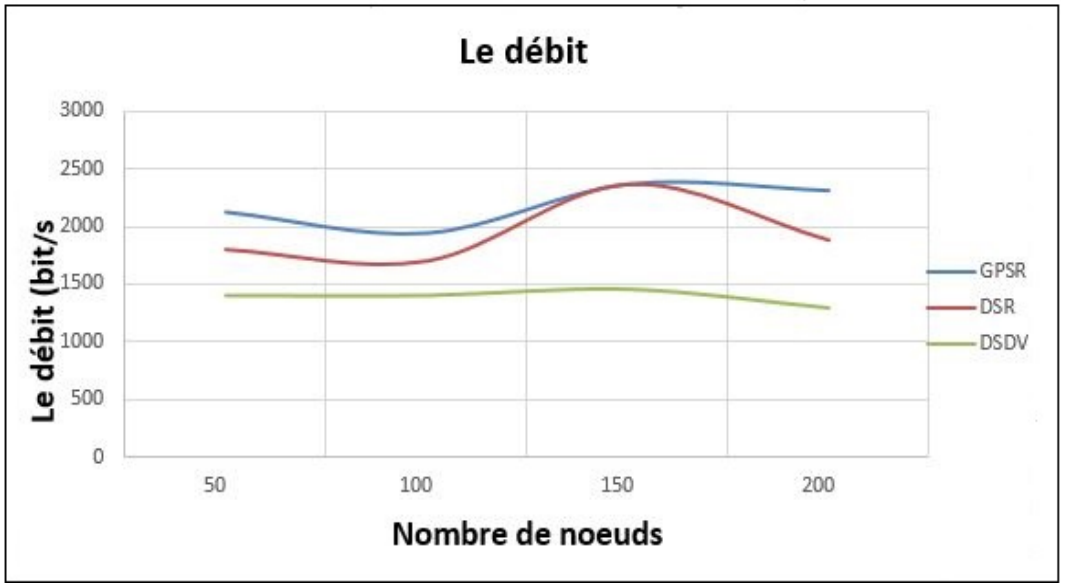

FIGURE 3.12 – *Débit en (bit/s).*

Délai moyen

D'après la FIGURE 3.13, le délai moyen de bout en bout que présente DSR s'accroît en fonction de la variation de nombres de nœuds. Par contre la courbe de DSDV reste stable puis elle augmente légèrement à un moment.

On ce qui concerne la courbe de GPSR reste en dessous des deux courbes DSR et DSDV, cela est expliqué par le fait qu'avant la transmission des paquets le protocole DSR utilise la technique de routage par la source et le protocole DSDV exploite l'algorithme Bellman Ford pour trouver le plus court chemin. Alors que le protocole GPSR emploi des messages Hello pour rafraichir périodiquement satable de voisinage.

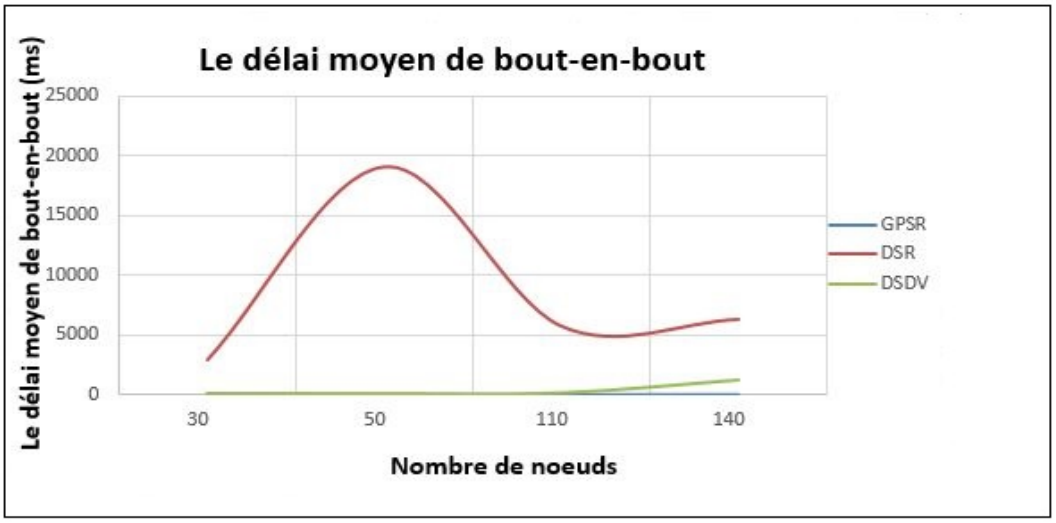

FIGURE 3.13 – *Délai moyen de bout-en-bout.*

Nombre de paquets perdus

Dans la FIGURE 3.14 suivante on constate que GPSR présente moins de perte de paquets que les protocoles DSR et DSDV. Il revient au fait que le protocole GPSR est plus fiable que DSDV et DSR.

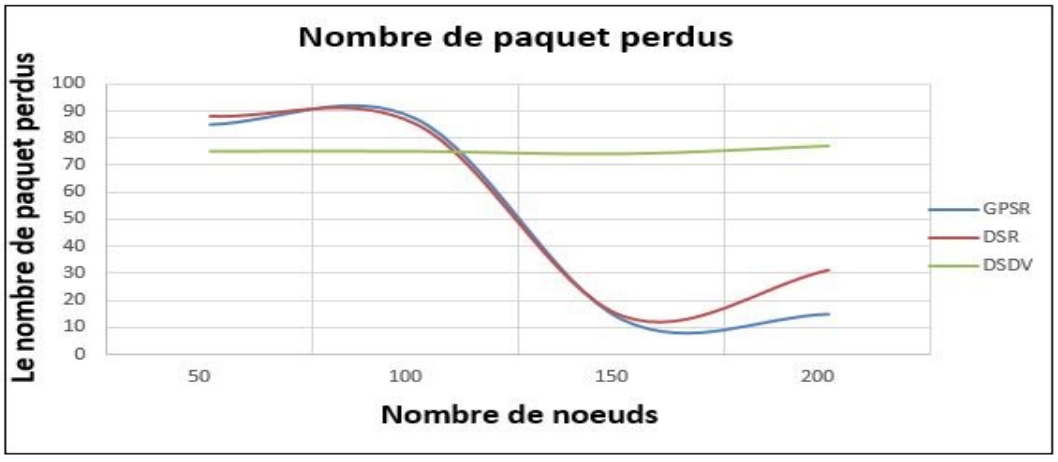

FIGURE 3.14 – *Nombre de paquets perdus.*

3.3.5.2 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

La FIGURE 3.15 ci-dessous présente les résultats de taux de livraison de paquets en fonction de temps de pause, remarquant que les deux protocoles DSDV et DSR diminuent du taux de livraison qui étais respectivement égale à 90% et 40% pour les deux protocoles et cela à partir d'un temps de pause qui est égale à 2s cependant quand ils arrivent à un égale 6s ils se stables. En parallèle le GPSR commence a augmenté d'un temps de livraison de 65%.Et cela est dûs moment que le nombres de paquets émis et reçus dans le protocole GPSR est moindre que les deux autres.

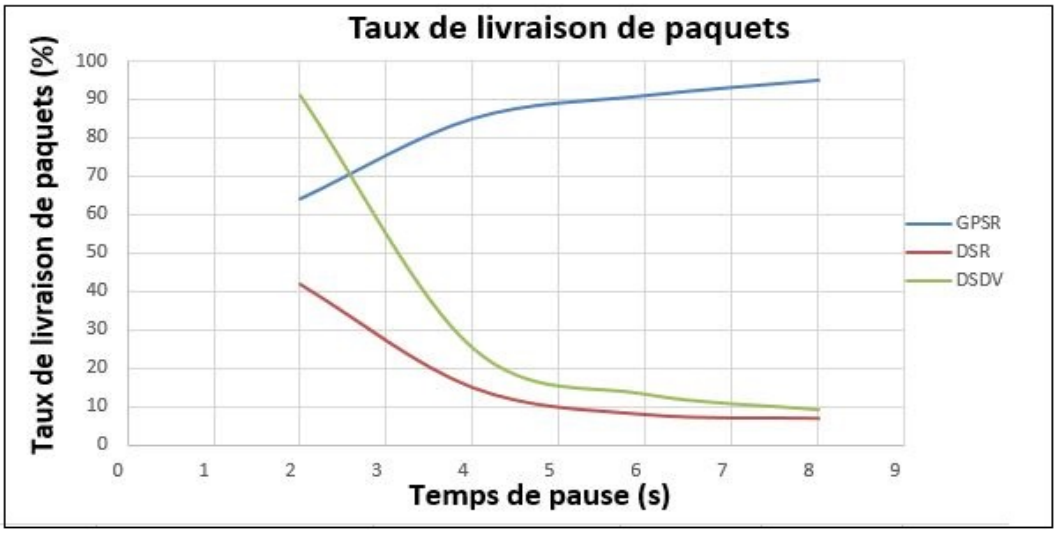

FIGURE 3.15 – *Taux de livraison de paquets.*

Débit

La FIGURE 3.16 représente le débit en fonction de temps de pause, on remarque que pour les trois courbes de protocoles diminuent à partir de 2s de temps de pause, et un débit qui est totalement différents d'un protocole à un autres cependant GPSR reste toujours le plus grand. Et cela revient au fait que la taille des paquets dans GPSR et plus petite que chez les deux autre.

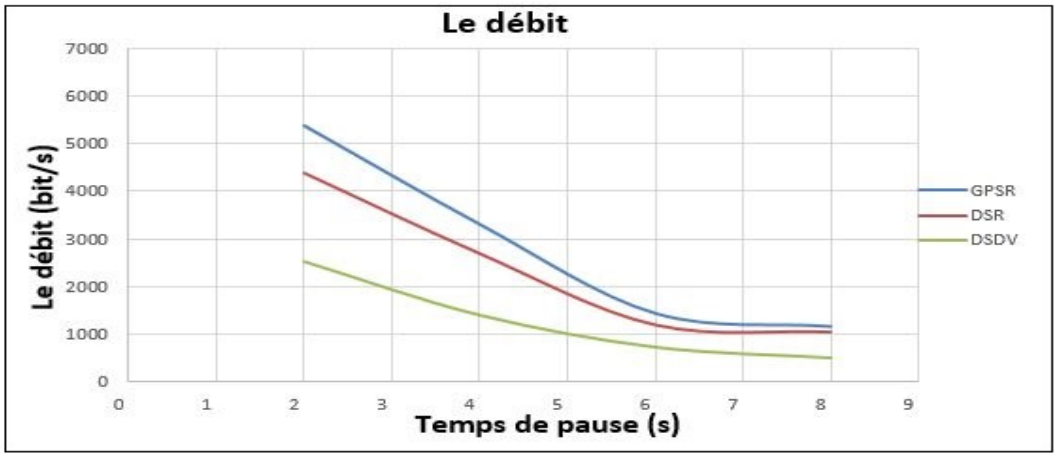

FIGURE 3.16 – *Débit.*

Délai

Cette FIGURE 3.17 illustre que la courbe DSR augmente légèrement d'un délai 16500ms avec un temps de pause de 2s jusqu'elle arrive a 3.5 second elle commence à diminuer part la suite elle reprend à nouveau l'augmentation. En revanche les courbes de DSDV et GPSR restent stable durant toute l'application.

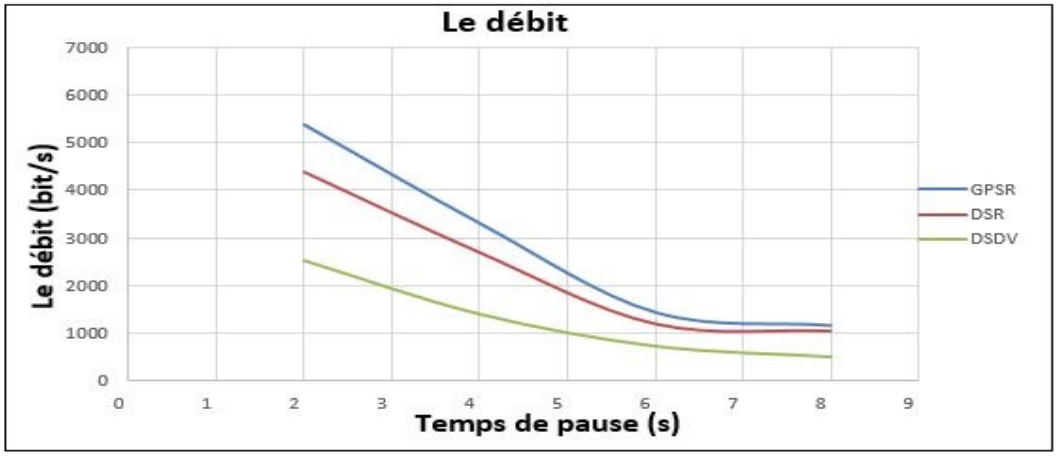

FIGURE 3.17 – *Délai.*

Nombre de paquets perdus

Les résultats obtenus dans la FIGURE 3.18 confirment que le GPSR présente moins de perte que le DSR et DSDV.Il revient au fait que le protocole GPSR est plus fiable que DSDV et DSR.

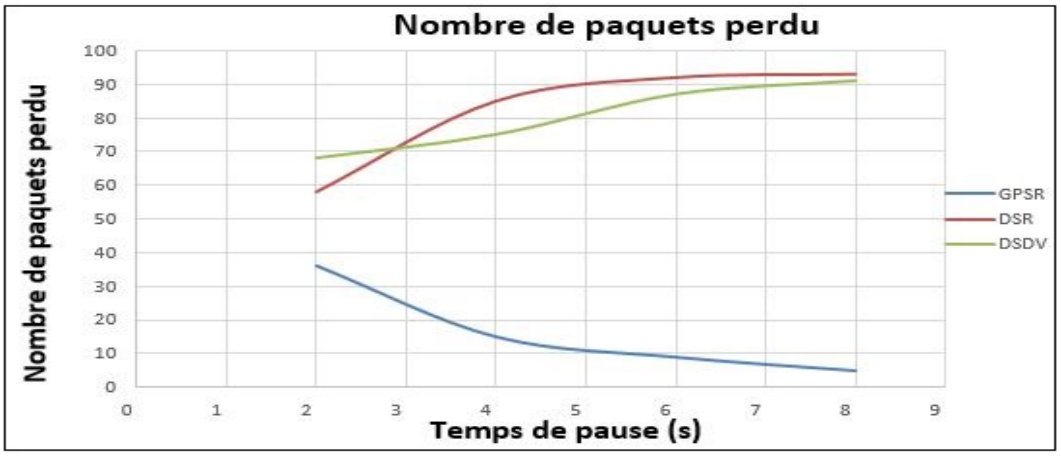

FIGURE 3.18 – *Nombre de paquets perdus.*

3.3.5.3 Troisième scénario

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

Taux de livraison des paquets

D'après la FIGURE 3.19 la courbe du protocole GPSR prend le dessus dans toute l'application par rapport aux deux autres courbes DSR et DSDV.Et cela est dûs moment que le nombres de paquets émis et reçus dans le protocole GPSR est moindre que les deux autres.

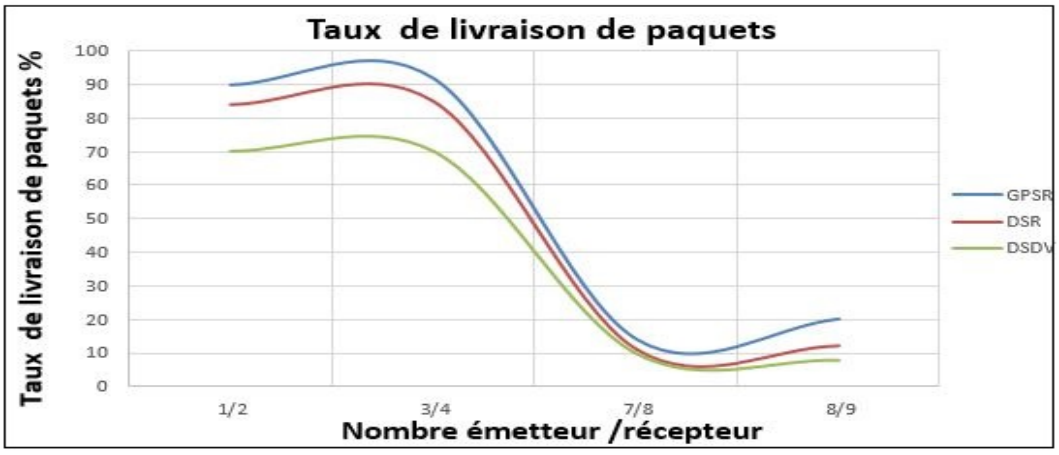

FIGURE 3.19 – *Taux de livraison de paquets.*

Débit

A travers cette FIGURE 3.20, on remarque que GPSR présente une légère augmentation par rapport aux deux protocoles DSDV et DSR, à partir du nombre (émetteur= 1 et récepteur=2), les courbes des trois protocoles diminuent puis augmentent à partir du nombre (émetteur= 7 et récepteur=8) mais toujours la courbe de GPSR se trouve au-dessus des deux autres.Et cela revient au fait que la taille des paquets dans GPSR et plus petite que chez les deux autre.

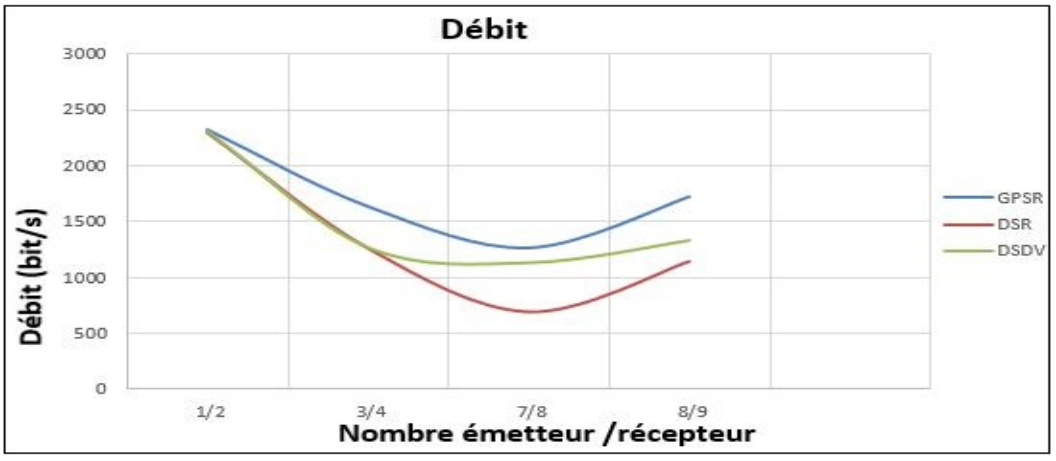

FIGURE 3.20 – *Débit.*

Délai

On remarque que la courbe de DSR diminue de 2000 ms jusqu'à 1000 ms entre la première application (émetteur=1, récepteur=2) et la deuxième application (émetteur=3, récepteur=4), la courbe DSDV diminue de 1200 ms jusqu'à ce qu'elle se

stabilise tandis que DSR augmente. Par contre la courbe du GPSR reste stable durant toutes les applications représentées dans la FIGURE 3.21.

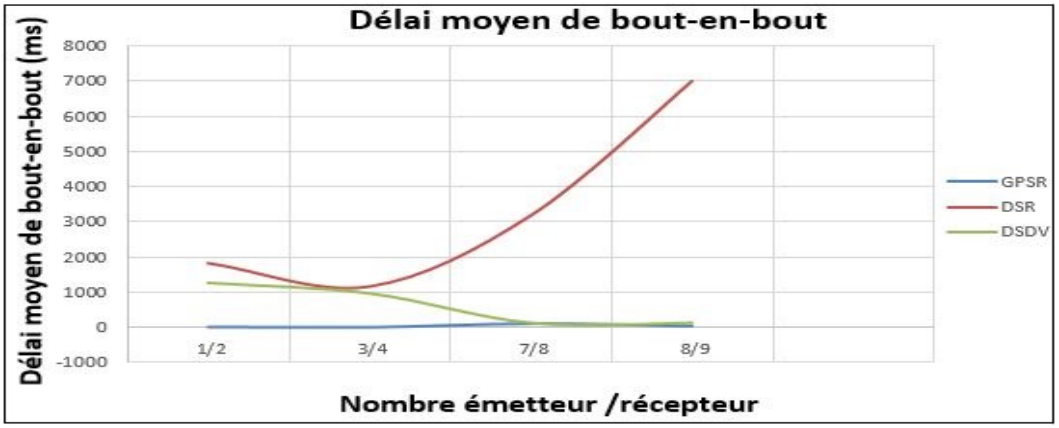

FIGURE 3.21 – *Délai.*

Nombre de paquets perdus

On constate que GPSR présente des résultats sensiblement meilleurs que les protocoles DSR et DSDV dans la FIGURE 3.22.Il revient au fait que le protocole GPSR est plus fiable que DSDV et DSR.

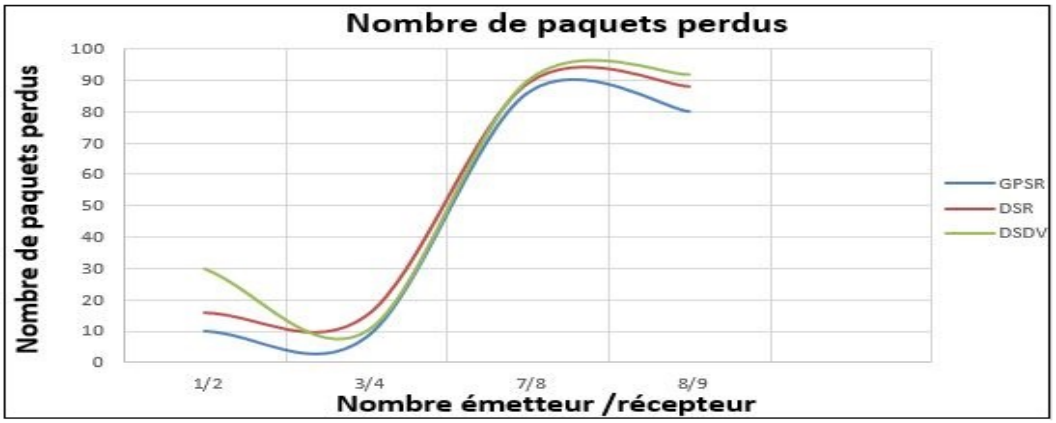

FIGURE 3.22 – *Nombre de paquets perdus.*

3.4 Interprétation des résultats

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 choisis pour chaque scénario sont illustrés dans le TABLE 3.1.

Paramètres	Scénario 1	Scénario 2	Scénario 3
Nombre de nœuds	30, 50, 110 et 140	100	100
Temps de simulation	600 sec	10000 sec	600 sec
Portés de communication	80 m	80 m	80 m
Temps de pause	3 sec	2, 3, 6 sec	3 sec
Taille de paquet	1024 ko	1024 ko	1024 ko
Nombre émetteur/ récepteur	1/2	1/2	1/2,3/4, 7/8,8/9
Nombre de paquets de nœuds	100	100	100

TABLE 3.1 – Les paramètres utilisés dans les différents scénarios.

Conclusion

Dans ce chapitre nous avons introduit les outils choisis pour la simulation et l'évaluation des performances des tois protocoles choisi (GPSR, DSR, DSDV). Après l'évaluation, nous avons constaté que le protocole GPSR 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 GPSR est relativement faible.

Enfin nous concluons que parmi les trois protocoles étudiés, le protocole GPSR répond mieux au besoin des réseaux véhiculaires et particulièrement en environnement urbain qui sont à la base de nos simulations.

Conclusion générale

A travers ce manuscrit nous avons dans un premier lieu présenté une vue générale sur les réseaux VANETs qui sont considérés comme une particularité des réseaux MANET (Mobile Ad hoc NETwork) où les nœuds mobiles sont des véhicules intelligents, on parle de la notion de « véhicule intelligent » quand un véhicule est équipé de calculateurs, dispositifs de communications sans fil, cartes réseau et de capteurs. Alors, un réseau VANET est formé de plusieurs véhicules communiquant entre eux ou avec des stations fixes afin d'offrir une conduite collaborative sécurisée et un environnement plus sûr.

Dans la suite du manuscrit nous avons présenté le concept de routage dans les réseaux VANETs et nous avons déduit que le routage joue un rôle très important et un bon protocole de routage est celui qui est capable de livrer un paquet dans un temps très court et un minimum de bande passante. 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 que nous avons utilisé pour évalué les trois protocoles choisi (GPSR, DSR, DSDV).

Après avoir calculé le taux de livraison de paquet, le nombre de paquet perdu et le délai moyen de bout en bout, les résultats auxquels nous avons abouti montrent que le protocole de routage GPSR offre des meilleures performances que DSR et DSDV.

Perspectives

Pour l'avenir, nous envisageons :
Adapter nos simulations à un environnement autoroutier, afin de conclure si ces protocoles (DSDV, DSR et GPSR) répondent de la même manière qu'aux environnements urbains. Utiliser des scénarios plus complexe et encore plus proche de la réalité, avec un passage à l'échelle en utilisant un maximum de nœud.

Apporter une amélioration au protocole géographique GPSR pour augmenter ses capacités ceci en ajoutant une fonction d'auto-configuration de la zone de couverture d'un véhicule.

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

Bibliographie

[1] S.M. Al-Sultann, A . Al-Doori, M.and H. Al-Bayatti, and H. Zedan. A comprehensive survey on vehicular ad hoc network. *Software Technology Research Laboratory, De Montfort University, Bede Island Building, Western Bou-levard, Leicester LE2 7EW, UK*, 2012.

[2] K. Ali. Vehicular ad-hoc networks : Introduction, standards, routing protocols and challenges. *Thése de doctorat. In Université de Technologie de Belfort-Montbeliard*, 2016.

[3] S. Allal. optimisation des échanges dans le routagegéocast pour les réseaux de véhicules ad hoc vanets. *Thèse de doctorat , université paris 13*, 10 Décembre 2014.

[4] M. Altayeb and I. Mahgoub. A survey of vehicular ad hoc networks routing-protocols. *Journal of Innovation and AppliedStudies*, July 2013.

[5] T. Atéchian. Protocole de routage géo-multipoint hybride et mécanismed'acheminement de données pour les réseaux ad hoc de véhicules (vanets). *thèse de doctorat*, 24 septembre 2010.

[6] D. Bektache. Application et modélisation d'un protocole de communication-pour la sécurité routière. *Thése de doctorat. In Université Badji Mokhetar AnnabaAlgerie*, 2014.

[7] P. Bijan and J. Islam-Mohammed. Survey over vanet routing protocols for vehicle to vehicle communication. *Department of Computer Science and Engineering Shahjalal University of Science and Technology, Sylhet, Bangladesh*, 2000.

[8] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). *RFC 3526, Mobile Ad Hoc Networking Working Group*, octobre 2003.

[9] S. Corson and J. Macker. Mobile ad hoc networking (manet) routing protocol performance issues and evaluation considerations. *Request for Comments (RFC) 2501, IETF*, January 1999,Maryland.

[10] Pucha H. Das, S. and Y. Hu. Performance comparison of scalable location services for geographic ad hoc routing. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and CommunicationsSocieties. Proceedings IEEE, vol. 2*, 2005.

[11] H. Ehan and Z.A. Uzmi. performance comparaison of ad-hoc wireless network routing protocols. *IEEE 8th International Multi to-pic conference , Proceedings of INMIC, page 457-465*, December 2004.

[12] M. Farah. Méthodes utilisant des fonctions de croyance pour la gestion des informations imparfaites dans les réseaux de véhicules. *Thèse de doctorat de l'Université d'Artois*, 2014.

[13] B. Guizani. Algorithme de clusterisation et protocoles de routage dans les réseaux ad hoc. *Thèse de doctorat de l'université de Technologie de Belfort-Montbéliard Tunisie.176. pp 85-115*, 04 Avril 2012 .

[14] Zygmuntand R. Marc-P. Haas, J. and P. Samar. The zone routing protocol (zrp) for ad hoc networks. *Internet-draft, IETF MANETWorking Group*, juillet 2002.

[15] N. Haddadou. réseaux ad hoc véhiculaires : vers une dissémination de données efficace, coopérative et fiable. *Thèse de doctorat de l'université PARIS-EST*, 2014 .

[16] Oussama M.R. hidi, O. and M. Erritali. Uml modeling of geographicrouting-protocol. *,reedy Perimeter Stateless Rou-ting ,for itsintegrationinto the Java Network Simulator , International Journal of Advanced Research in Computer Science and Software Engineering*, 2012 .

[17] J.P. Hubaux. Vehicular networks : How to secure theme. *MiNeMa Summer School,Klagenfurt, Germany*, July 2005.

[18] M. Jerbi. Protocoles pour les communications dans les réseaux devéhicules en environnement urbain :routage et geocast basés sur les intersections. *Thése de doctorat. In Université Paris*, 6 Novembre 2008.

[19] D. Jiang and L. Delgrossi. Ieee 802.11 p : Towards an international standard forwireless access in vehicular environments. *In Vehicular Technology Conference,pages 2036–2040*, 2008.

[20] Y. Jin Kim and R. Govidan. Greedy perimeter stateless routing (gpsr) detailed-level design specification (dds) version 0.1. *Department of Computer Science University of Southern California*, July 6, 2003.

[21] V. Jindal and P. Bedi. Modélisation et étude de performances dans les réseaux vanets. *In International Journal of Computer ScienceIssues (IJCSI), volume 13, Numéro 2, page 44*, 2012.

[22] M. Joa-NG and T. Lu. A peer-to-peer zone-based two-level link state routing for mobile ad-hoc networks. *In : IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, pp.1415-25*, 1999.

[23] Maltz D.A. Johnson, D. and Y.C. Hu. The dynamic source routingprotocol for mobile ad hoc networks (dsr) ietf internet draft. *work in progress,draft-ietf-manet-dsr-09.txt.*, April 2003.

[24] Maltz D.A. Johnson, D.B. and Y-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). *Internet Draft*, 19 Juillet 2004 .

[25] B. Karp and H.T. kung. Greedyprimeterstatelessrouting for wireless networks . *in proceedings of the sixthannual ACM/IEEE International conferenceon mobile computing and networking, pages 243–254*, 2000.

[26] Menouar H. Khaled, Y. and Y. Challa. Reactive and adaptative protocol for inter vehicle communication (rap-ivc). *UMR-CNRS, France*, 2004.

[27] Ekici E. Korkmaz, G. and F. Ozguner. An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems. *Department of Electrical and Computer Engineering, The Ohio State University*, 2006.

[28] I. Leontiadis and C. Mascolo. Review composition based protocol in vehicular ad hoc network. *American journal of engineering research*, 2013.

[29] Hartenstein H. Tian-J. Herrman D. Fubler H. Lochert, C. and M. Mauve. A routing strategy for vehicular ad hoc networks in city envi-ronlents. *IEEE intellihent vehicles symposium, IV*, 2003.

[30] R. Meraihi. Gestion de la qualité de service et contrôle de topologie dans les réseaux ad hoc. *Thèse doctorat. Ecole Nationale Supérieure des Télécommunications de Paris*, 2005.

[31] Senouci M. Meraihi, R. and M. Djebri. Réseau mobile ad hoc et réseaux de capteurs sans fil. *chapitre de livre Edition Hermes*, 2006.

[32] NS-3. https ://www. *nsnam.org/*, juin 2017.

[33] J. Petit. Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires. *Thèse de doctorat de l'université de toulouse*, 2011.

[34] O. Rivaton. Le routage de l'information dans les réseaux véhiculaires mobiles. *université de Québec, Canada*, 2008.

[35] S. SAFAE. evaluation des performances des protocoles de routage aodv et olsr dans un environnement vanet. *master science et techniques système intelligents et réseaux*, July 2013.

[36] Liu G. Lee-B.-S. Foh C. H. Wong K. J. Seet, B.C. and K.-K. Lee. A-star : A mobile ad hoc routingstrategy for metropolisvehicular communications. *NETWORKING 2004, 989-999*, 2004 .

[37] SUMO. Simulation of urban mobility. *http ://sumo.sourceforge.net/*, juin 2017 .

[38] IEEE P1609.1 SWG. Ieee trial-use standard for wireless access in vehicular environments (wave). *Resource Manager, IEEE Computer Society*, 2006.

[39] IEEE P1609.2 SWG. Ieee trial-use standard for wireless access invehicular environments (wave). *Security Services for Applications and Management Messages, IEEE Computer Society*, 2006.

[40] IEEE P1609.3 SWG. Ieee standard for wireless access in vehicular environments (wave). *Networking Services. IEEE Computer Society*, 2010.

[41] IEEE P1609.4 SWG. Ieee standard for wireless access in vehicular environments (wave), multichannel operation. *IEEE Computer Society*, 2010.

[42] Muhlethaler P. Toor, Y. and A. Laouiti. Vehicle ad hoc networks : applications and related technical issues. *In Communications and Surveys Tutorials, volume10, Numéro 3, pages 74–88*, 2008.

[43] J. Zhao and G. Cao. Vadd : Vehicle-assisted data delivery in vehicular ad hoc networks. *In Proceedings of the 25th Conference on Computer Communications (INFOCOM06)*, 2006.

ANNEXE I : Intégration de GPSR à NS-3

Les protocoles DSR et DSDV font partie des modules (protocoles de routages) intégrés dans NS-3, alors que le protocole GPSR ne fait pas partie de ces modules. Dans ce qui suit, nous allons montrer comment nous avons intégré le protocole GPSR. Dans un terminal, exécuter le script :

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

Pour télécharger la dernière version de GPSR et les paquets manquants, il faut exécuter :URL :https ://github.com/nandanwarchetan/ns3-gpsr/tree/master/src. Nous avons copié les sous répertoires de ns3 /ns3.19/src/gpsr/* dans le dossier crée Précédemment avec create-module.py (dossier GPSR). 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 rencontrées 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/gpsr/wscript** comment suite :

Il faut remplacer

```
headers=bld.new_task_gen(features=['ns3header'])
```

par

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

◇ Puis nous avons copié le fichier **ipv4-l4-protocole.hdans ns3/ns-3.19/build/ns3.**

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

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

ANNEXE II : Étape de JOSM à SUMO

II.1 JOSM

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

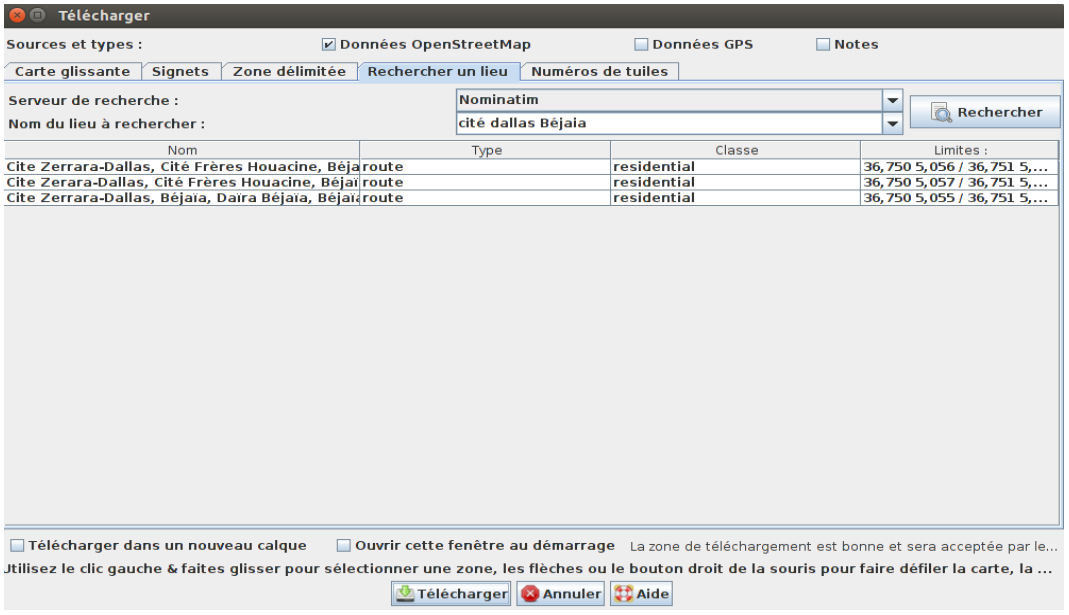

FIGURE 3.23 – *Recherche d'un lieu .*

- Sélectionner la zone voulue puis enregistrer la Carte.

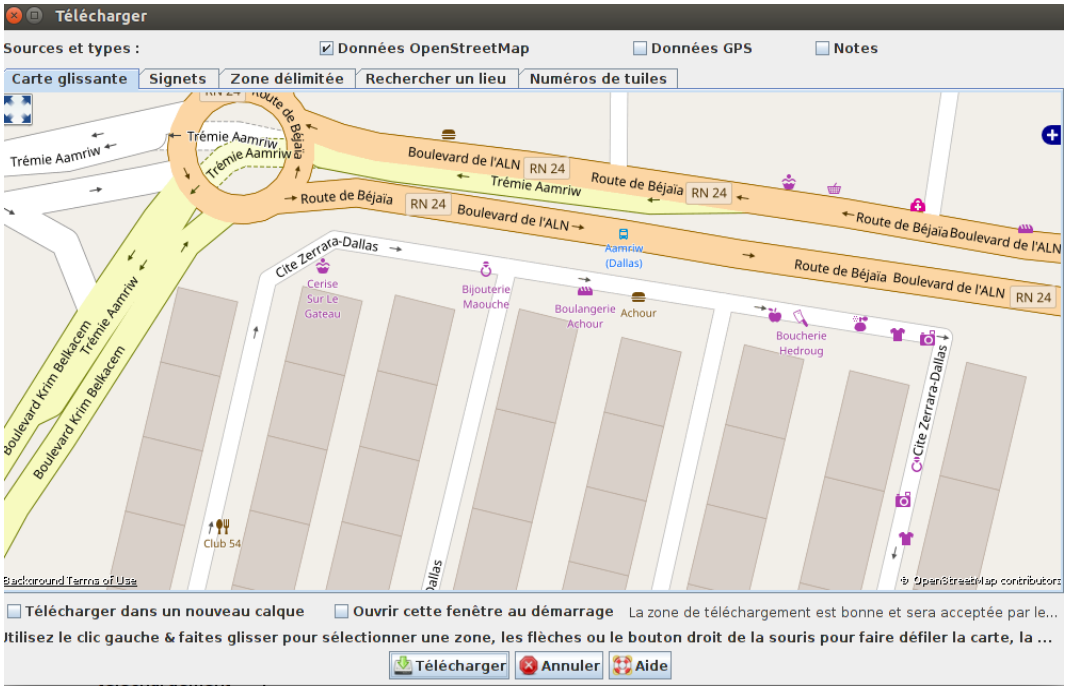

FIGURE 3.24 – *Télécharger la carte JOSM.*

II.2 eWorld

- OuvrireWorld,
- Taper la commande suivante sur un terminal ./eWorld.sh
- *Importer* $\longrightarrow$ *Import* from OSM file

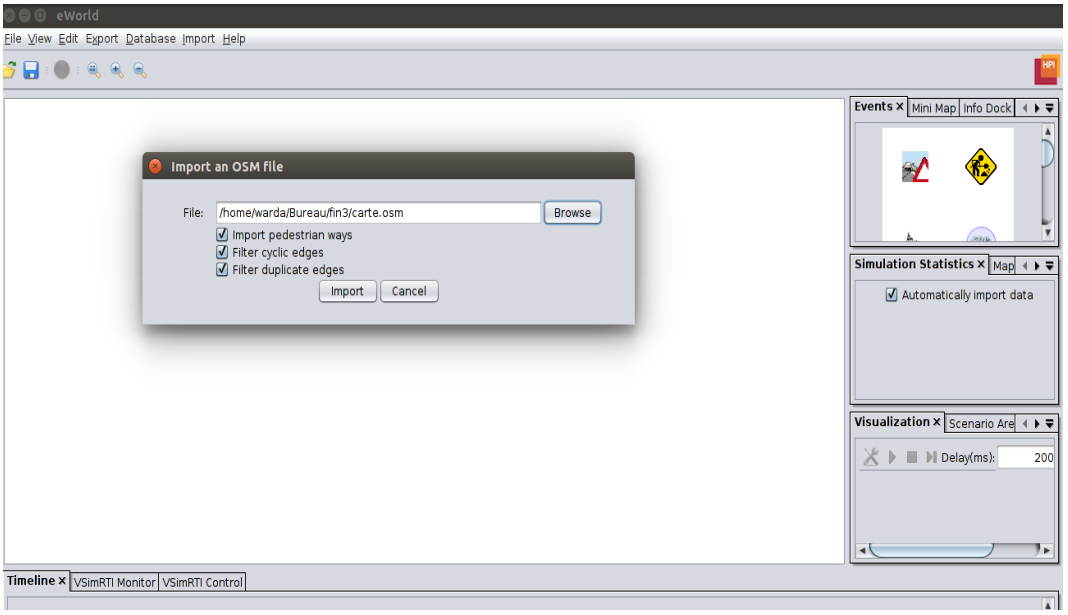

FIGURE 3.25 – *Importation de la carte.*

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

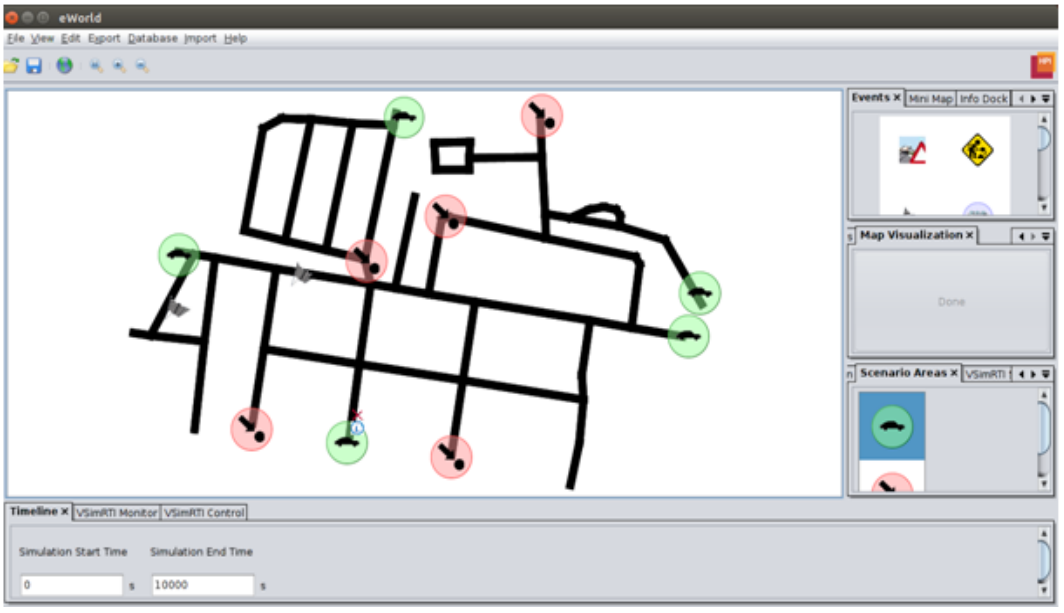

FIGURE 3.26 – *Ajout des points de départ et d'arrivée.*

- Export->Export to SUMO...

(a) Location : à spécifier

(b) Name : à spécifier

(c) Case à cocher : Export trafic light system logicsde_ned in eWorld

(d) Case à cocher : Generate Net-File

(e) Location of netconvert : /usr/bin/netconvert

(f) Case à cocher : Generate routes withduarouter...

(g) Location of duarouter : /usr/bin/duarouter

(h) case à cocher : Generate scenarios

(i) Export

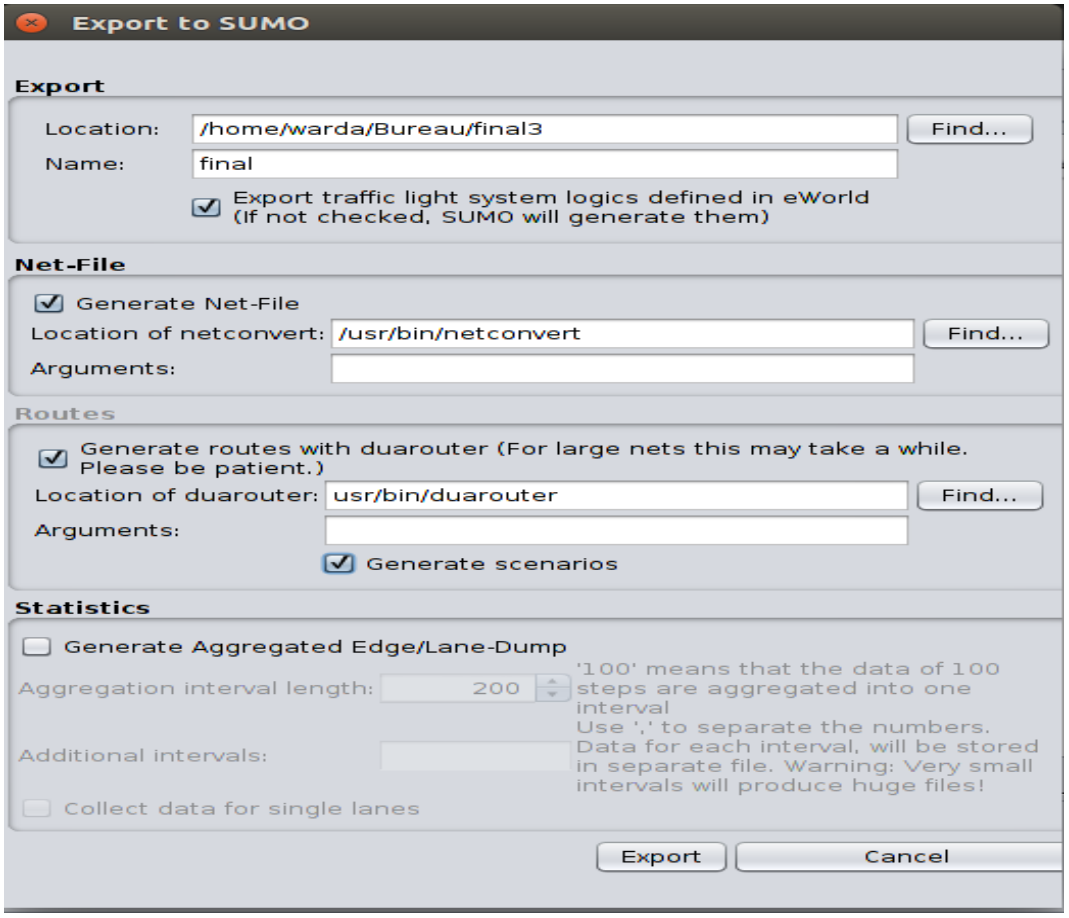

FIGURE 3.27 – *Exporter la carte vers SUMO.*

II.3 SUMO

- Appliquer quelques modifications sur les fichiers généré comme suit : final.flows.xml Changer no=" "/> par number=" "/> dans toutes les occurrences du fichier.

Ensuite exécuter la commande suivante pour créer le fichier rou :

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

En plus nous portons dans ce dernier cette modification : final.sumo.cfg<route-files value="finalNew.rou.xml"/>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 350.
- Démarrer la simulation FIGURE 3.28.

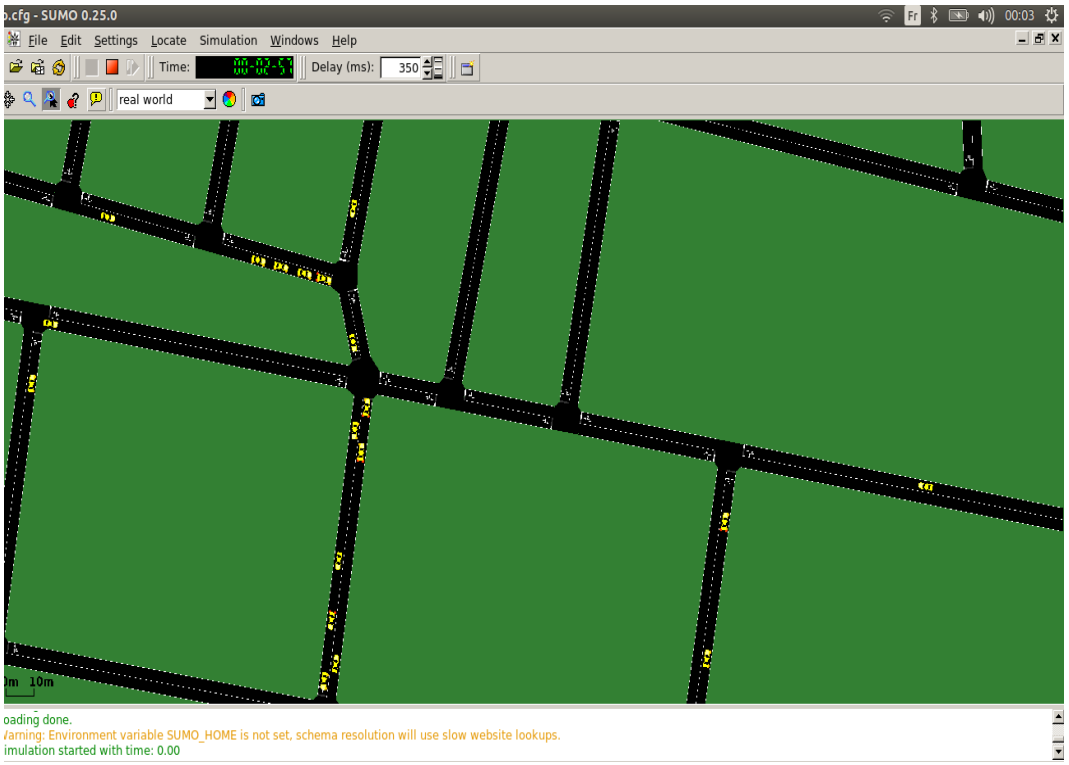

FIGURE 3.28 – *SUMO.*

ANNEXE III : Résultats de simulation

Nombre de noeuds	30	50	110	140
GPSR	15	13	87	85
DSR	12	15	86	69
DSDV	25	25	26	23

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

Nombre de noeuds	30	50	110	140
GPSR	2130.3	1948.5	2371.52	2317
DSR	1798.81	2698.61	2368.13	1880.69
DSDV	1405.53	1405.53	1461.75	1293.08

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

Nombre de noeuds	30	50	110	140
GPSR	42.5333	5.84615	8.71264	12.2
DSR	2849	19151	5682	6266
DSDV	99	105	159	1150

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

Nombre de noeuds	30	50	110	140
GPSR	85	87	13	15
DSR	88	85	14	31
DSDV	75	75	74	77

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

Temps de pause	1	3	6	9
GPSR	64	85	91	95
DSR	42	15	8	7
DSDV	31	25	13	9

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

Temps de pause	1	3	6	9
GPSR	5394,8	3317	1440,29	1163,205
DSR	4391.84	2698.61	1199.25	1049.38
DSDV	2535.03	1405.53	730.874	505.989

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

Temps de pause(s)	1	3	6	9
GPSR	9.79688	12.2	9.2967	8.50526
DSR	16699	17151	5720	5728
DSDV	500	105	65	50

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

Temps de pause	1	3	6	9
GPSR	36	15	9	5
DSR	58	85	92	93
DSDV	69	75	87	91

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

Nombre émetteur/récepteur	1/2	3/4	7/8	8/9
GPSR	85	25	13	21
DSR	84	23	6	14
DSDV	75	83	83	77

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

Nombre émetteur/récepteur	1/2	3/4	7/8	8/9
GPSR	2317	1729.46	1168.81	1715.74
DSR	2289,74	1861.78	2262.51	1144.85
DSDV	1405.53	2262.5	1125,92	2098.96

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

Nombre émetteur/récepteur	1/2	3/4	7/8	8/9
GPSR	8.62353	11.68	36.4615	28.619
DSR	3816	3967	2917	6987
DSDV	131	225	115	100

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

Nombre émetteur/récepteur	1/2	3/4	7/8	8/9
GPSR	15	75	87	79
DSR	16	77	96	86
DSDV	25	17	17	23

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

ANNEXE IV : Installation des outils

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

IV.1 Installation de JOSM

IV.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

IV.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 »
◇ Faire un clic droit sur le fichier josm-tested.jar téléchargé.
◇ Cliquer sur propriétés et sélectionner l'onglet « ouvrir avec » cocher la case « Sun Java 6 run time », fermer la fenêtre.
Maintenant pour chaque lancement il suit d'un double clic sur l'icône josm-tested.jar.

IV.2 Installation de 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 :
./eWorld.sh

IV.3 Installation de SUMO : Pour l'installer, ouvrer le terminal de Linux et exécuter la commande :
sudo apt-get install sumo

IV.4 Installation de NS-3 : NS-3 est conçu pour fonctionner sur les plate-formes Linux. Toute fois il peut être installer sur MAC OS et Ms Windows en utilisant Cywing. Pour les instructions détaillées et les prérequis, consulter le site : http ://www.nsnam.org/wiki/index.php/Installation
L'installation comprends les étapes suivantes :

IV.4.1 Téléchargement

◇ Télécharger la dernière version http ://www.nsnam.org/releases/

◇ Décompresser l'archive : tar -jxf ns-3.19.tar.bz2

IV.4.2 Compilation

◇ Entrer dans le répertoire de ns-3 et éxécuter./build.py

IV.4.3 Validation

◇ Tester l'exécution d'un script
./waf −−run scrach/test ˘−vis

ANNEXE V : script 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/gpsr-module.h"//**nous modifions "ns3/gpsr-module.h" par "ns3/dsr-module.h", "ns3/dsdv-module.h" pour les deux protocoles (DSR et DSDV) 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**

DSR et DSDV

****

NS_LOG_COMPONENT_DEFINE("GpsrTest") ;") ;// nous modifions ''GpsrTest" par ''DsrTest'', ''DsdvTest" pour les deux protocoles (DSR et DSDV) respectivement.

usingnamespace ns3 ;

.....

intpacketsReceived = 0 ;

long double throughput = 0.0 ;

long double bytesTotal = 0 ;

intm_lossCounter = 0 ;

long double xx ; //le temps de génration 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 **********

voidReceivePacket(Ptr< Socket >socket)

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

While(packet = socket->Recv())

SeqTsHeaderseqTsx ;

packet->RemoveHeader(seqTsx) ; //séparer l'enteteSeqTsHeader 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 générate trafic**********
staticvoidGenerateTra_c(Ptr<Socket> socket, uint32_t pktSize, uint32_t pktCount, Time pktInterval)

if (pktCount> 0)
SeqTsHeaderseqTs ;
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'enteteSeqTs.
p->AddHeader(seqTs) ;//ajouter l'entête au paquet "p".
socket->Send(p) ; // la source envoi 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(intargc, 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.
doubleinterval = 3 ; // temps inter paquet en secondes.
int range = 80 ; // porté de communication des nœud.
double hello = 1 ; // période d'émission des messages hello.
boolperimeter = 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 : :FragmentationThreshold", 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 : :NonUnicastMode",StringValue(phyMode)) ; NodeContainerc ;// la structure de noeud.
c.Create(numNodes) ;// créer un nombre = numNodes de nœud
//********** **Configuration du canal WIFI**********
WifiHelperwifi ;
YansWifiPhyHelperwifiPhy = YansWifiPhyHelper : :Default() ;

//********** **configuration du model de propagation**********
YansWifiChannelHelperwifiChannel ;wifiChannel.SetPropagationDelay("ns3 : :ConstantSpeedPropagationDelayModel") ;//le temps.
wifiChannel.AddPropagationLoss("ns3 : :TwoRayGroundPropagationLossModel","SystemLoss", DoubleValue(1),"HeightAboveZ", DoubleValue(1.5)) ; //model de propagation.
wi_Channel.AddPropagationLoss("ns3 : :FriisPropagationLossModel","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", StringValue(phyMode)) ;

//**********installation de couche phy et mac pour chaque noeud**********
NetDeviceContainerdevices = 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 nœuds contenu dans le fichier trace (mobility.tcl).
//********** **Configuration du protocole de routage ad-Hoc**********
// activation de protocole.
nom_protocoleHelpernom_protocole ;// activation de protocole.
InternetStackHelperinternet ;
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 nœud.
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) ; //receving
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

VANET (Vehicular Adhoc Networks) est une nouvelle technologie émergente qui intègre les fonctionnalités de la nouvelle génération de réseaux sans fil pour les véhicules. 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. Les réseaux ad hoc véhiculaires reposent sur des protocoles qui assurent l'échange d'informations et la communication entre les véhicules puisqu'il est évident que l'amélioration de la communication entre les véhicules revient à la détermination de l'efficacité de ces protocoles. Notre objectif, à travers ce mémoire, est d'évaluer les performances de protocoles de routage GPSR, DSR et DSDV appliqués aux réseaux dans un environnement Urbain, en utilisant NS3 et SUMO, pour déterminer le meilleur protocole en termes de taux de livraison de paquet, débit de bout en bout, paquets perdus, débit moyen. Les résultats de simulation ont montré que le protocole GPSR offre des meilleures performances que DSR et DSDV.

Mots clés : VANET, DSR, GPSR, DSDV, NS3 , SUMO.

Abstract

VANET (Adhoc Vehicular Networks) is a new emerging technology that incorporates the functionality of the new generation of wireless networks for vehicles. 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. Vehicular ad hoc networks rely on protocols that ensure information exchange and communication between vehicles since it's obvious that improved communication between the vehicle returns to the determination of the effectiveness of these protocols. Our objective, in this work, is to evaluate routing protocol performances GPSR, DSR and DSDV applied to VANET network in specified environment of the city, using NS3 and SUMO, in order to choose the best convenient one in term of the reception rate of packets, the end to end delay, lost packets and medium debit. The simulation results showed that the GPSR protocol offers better performances than DSR and DSDV.

Key words : VANET, DSR, GPSR, DSDV, NS3, SUMO.