

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Djilali Bounaama Khemis Miliana



Faculté des Sciences et de la Technologie
Département de Technologie

Mémoire du Projet de Fin d'Etudes
Pour l'obtention du diplôme de

Master

En

«Télécommunications »

Option :

« Systèmes de Télécommunications »

Titre :

**Evaluation des Protocoles de routages pour
l'interconnexion des réseauxVANETs**

Réalisé par

Mr. Nezar Soufyane & Mr. Makhlouf Ouahchia Noureddine

Rapporteur

Mme F. Benbekhti

Année Universitaire: 2019/2020

Dédicace 1

*Tout d'abord, je tiens à remercier Dieu tout puissant
Clément et miséricordieux de m'avoir donné la force
Et la diligence providentielle d'avoir élaboré ce projet d'étude.*

*À ceux qui m'ont donné le courage pour atteindre ce
Modeste travail.*

*Je remercie également par cette occasion tous les enseignants et
Les administrateurs du département d'Électronique.*

*À mon cher père et ma chère mère pour leur esprit de sacrifice,
Leur encouragement, leur générosité et leur soutien moral
Qu'ils m'ont apporté durant mon enfance et tout le long de mes*

Études,

Que Dieu me les garde.

Je souhaite que ce travail soit à la hauteur de leurs attentes.

À ma très chère femme et mon fils Aniss

A mes chers frères Ahmed, Moussa et Hamou

A mes chères sœurs

À toute la famille, mes oncles et mes tantes, mes cousins et cousines.

Tous les amis en particulier.

Soufyane

Dédicaces 2

Avec grande joie, je dédie ce modeste travail

A ceux qui m'ont aidé

*A ceux qui sont la source de mon inspiration et mon
courage*

A ceux qui m'ont encouragé et soutenu moralement

A ceux que j'aime et respecte.

A ma très chère mère.

A mon cher père.

A mes très chères sœurs

A mes très chers frères.

A toute ma famille.

A tous mes amis.

A tous mes enseignants

Noureddine

Remerciements

Nous remercions tout d'abord le grand Dieu pour l'achèvement de ce mémoire.

Nous remercions Mme Benbekhti, notre encadreur, pour ses conseils et suggestions avisés qui nous aidés à mener à bien ce travail, et d'avoir rapporté à ce mémoire ces remarques et conseils.

Nous exprimons nos gratitudees à Monsieur le président de jury d'avoir accepté examiné ce mémoire.

Nous remercions Messieurs les membres de jury, d'avoir accepté de prendre part à ce jury ainsi que pour l'intérêt qu'ils l'ont portés à ce travail.

Résumé :

VANET (Vehicular Ad-hoc Networks) est une nouvelle technologie émergente qui intègre les fonctionnalités de la nouvelle génération de réseaux sans fil pour les véhicules. Le principal objectif de ce type de réseau est de faire circuler l'information de la manière la plus efficace possible entre les véhicules eux-mêmes ; et entre les véhicules et les infrastructures afin d'améliorer la sécurité routière et d'apporter le confort aux conducteurs et aux passagers. Les réseaux véhiculaires reposent sur des protocoles qui assurent l'échange d'informations entre les véhicules dont sont caractérisés par une forte mobilité. L'amélioration de la communication revient à la détermination de l'efficacité de ces protocoles.

Ce projet de fin d'études comprend une étude comparative des protocoles de routage: AODV, DSDV, DSR, OLSR sur deux scénarios qui décrivent la circulation automobile urbaine dans la ville de Sidi hamou, à Aindefla. Les scénarios ont été générés avec SUMO et simulés avec NS3 en utilisant de vraies cartes. Le réseau Ad-hoc véhiculaire (VANET) était configuré avec des interfaces sans fil selon la norme IEEE 802.11p.

L'évaluation VANET est réalisée sous les métriques suivantes: taux de livraison des paquets (*PDR*), le délai moyen de bout en bout (*EED*), la charge de routage normalisée (*NRL*) et le nombre de paquets perdus (*PLR*) et le débit (*throughput*). Ces métriques sont présentées comme résultat de simulations à travers des tableaux et des graphiques. Les performances du réseau sont optimales lors de l'obtention de niveaux élevés de débit et de *PDR*, avec de faibles niveaux de retard et de *NRL*.

Mots clés : VANET, NS3, SUMO, OLSR, DSDV, DSR, AODV.

ملخص

شبكات للمركبات المخصصة (VANET) هي تقنية ناشئة جديدة تدمج وظائف الجيل الجديد من الشبكات اللاسلكية مع المركبات. الهدف الرئيسي من هذا النوع من الشبكات هو تعميم المعلومات بأكثر الطرق فعالية بين المركبات بحد ذاتها أو بين البنية التحتية والمركبات لتحسين السلامة على الطرق وتوفير الراحة للسائقين والركاب. تعتمد شبكات المركبات على بروتوكولات تضمن تبادل المعلومات بين المركبات المتميزة بالحركة العالية. حيث أن تحسين الاتصال يسهل تحديد فعالية هذه البروتوكولات.

يتضمن مشروع نهاية الدراسة هذا، دراسة مقارنة لبروتوكولات التوجيه AODV - DSR - DSDV - OLSR على سيارتين يوهين يصفان حركة مرور السيارات في المناطق الحضرية في مدينة سيدي حمو بولاية عين الدفلى. تم إنشاء

السيناريوهات باستخدام SUMO وتمت محاكاتها باستخدام NS3 باستخدام خرائط حقيقية. تم تكوين الشبكة المخصصة للمركبات (VANET) بواجهات لاسلكية وفقاً لمعيار 802.11b.

يتم إجراء تقييم VANET وفقاً للمقاييس: معدل تسليم الحزم (PDR) ومتوسط التأخير من طرف إلى طرف (EED) و شحنة التوجيه المنتظمة (NRL) وعدد الحزم المفقودة (PLR) والتدفق. هذه المقاييس يتم تقديمها نتيجة عمليات المحاكاة؛ من خلال الجداول والرسوم البيانية. أداء الشبكة هو الأمل عند تحقيق مستويات عالية من التدفق و معدل تسليم الحزم مع مستويات منخفضة من التأخير و شحنة التوجيه المنتظمة.

الكلمات المفتاحية : VANET, NS3, SUMO, DSDV, AODV, OLSR, DSR.

Abstract

Vehicular Ad.hoc Network is new emerging technology that integrates the functionalities of the new generation of wireless networks for vehicles. The main objective of this model of network is to circulate information in the most efficient way possible between vehicles; between infrastructure and vehicles to improve road safety and provide comfort to drivers and passengers. Vehicular networks are based on protocols that ensure the exchange of information between vehicles that are characterized by high mobility. Improving communication come down to determining the effectiveness of these protocols.

This end-of-study project includes a comparative study of routing protocols: AODV, DSDV, DSR, and OLSR on two scenarios that describe urban automobile traffic in the city of Sidi hamou, at Aindefla. The scenarios were generated with SUMO and simulated with NS3 using real maps. The Ad.hoc vehicular network (VANET) has been configured with wireless interfaces according to the IEEE 802.11p standard.

VANET evaluation is performed under the metrics: packet delivery rate (PDR), average end-to-end delay (EED), normalized routing load (NRL) and number of lost packets (PLR), and throughput which are presented as the result of simulations; through tables and charts. Network performance is optimal when achieving high levels of throughput and PDR, with low levels of delay and NRL.

Keywords: VANET, NS3, SUMO, OLSR, DSDV, DSR, AODV.

Liste des abréviations

AODV	Ad hoc O n Demand D istance V ector.
A-STAR	Anchor-based Street and Traffic Aware R outing.
DSDV	D estination- S equenced D istance- V ector.
DSR	D ynamic Source R outing « Routage à Source Dynamique».
GPS	G lobal P osition S ystem.
GPSR	G reedy P erimeter Stateless R outing.
GSR	G lobal State R outing.
GyTAR	Improved G reedy T raffi c-Aware R outing protocol.
IARP	I ntrA zone R outing P rotocol.
IERP	I ntEr zone R outing P rotocol.
IETF	I nternet E ngineering T ask F orce.
IP	I nternet P rotocol
MAC	M edia A ccess C ontrol.
MANET	M obile ad hoc n etworks
MPR	M ulti- P oint R elaying.
NDP	N eighbour D iscovery P rotocol.
NS3	N etwork S imulator 3 .
OLSR	O ptimized L ink S tate R outing.
OTcl	O bject T ool C ommand L angage.
RREP	R oute R epl y M essage.
RREQ	R oute R equ e st M essage.
RSU	R oad S ide U nits.

UMB	UrbanMulti hop Broadcast Protocol.
Unicast	Monodiffusion.
VADD	Vehicle-Assisted Data Delivery.
VANET	Vehicular Ad-Hoc Network.
WAVE	Wireless Access in Vehicular Environments
WiFi	Wireless Fidelity.
ZRP	Zone Routing Protocol.
SUMO	Simulation of Urban Mobility.
V2I	Vehicle To Infrastructure “Véhicule à Infrastructure”.
V2V	Vehicle To vehicle “Véhicule à Véhicule”.

Listes des figures

Chapitre 01

figure 1. 1. classification des reseaux.....	6
figure 1. 2. basic service set	7
figure 1. 3. independant basic service set.....	7
figure 1. 4. reseau vanet	8
figure 1. 5. exemple de vehicule intelligent	9
figure 1. 6. communication v2v	10
figure 1. 7. communication v2i	11
figure 1. 8. communication hybride	11
figure 1. 9. protocoles de routage dans les reseaux vanets	13

Chapitre 02

figure 2. 1. architecture de ns3	33
--	----

Chapitre 03

figure 3. 1. methode de construction d'une route.....	37
figure 3. 2. principe de detection d'un lien defaillant.....	38
figure 3. 3. densite des vehicules.....	41
figure 3. 4. vue d'une partie du village sidi hamou sur osm avant l'importer sur sumo	42
figure 3. 5. zone d'etude apres importation.....	43
figure 3. 6. etapes de la simulation	44
figure 3. 7. fichier mobilite.tcl.....	45
figure 3. 8. commande utilise pour la simulation	47
figure 3. 9. visualisation de 40 noeuds avec net anim.....	49
figure 3. 10. resultats de simulation du protocole aodv pour le scenario 1	49
figure 3. 11. pdr vs densite	50
figure 3. 12. delai vs densite.....	51
figure 3. 13. nrl vs densite	52
figure 3. 14. debit vs densite.....	53
figure 3. 15. pl vs densite	55
figure 3. 16. pdr vs vitesse.....	56
figure 3. 17. temps vs vitesse	57
figure 3. 18. nrl vs vitesse.....	58
figure 3. 19. debit vs vitesse	59
figure 3. 20. plr vs vitesse.....	60

Liste des tableaux

Chapitre 01

tableau 1. 1. protocoles proactifs vs reactifs	16
tableau 1. 2. grille d'analyse	19

Chapitre 02

tableau 2. 1. comparaison entre simulateurs de la mobilite.....	26
tableau 2. 2. comparaison entre simulateurs de reseaux.....	29
tableau 2. 3. comparaison entre les simulateurs integres.....	31

Chapitre 03

tableau 3. 1. parametres de simulation de mobilite.	42
tableau 3. 2. parametres de simulation generale	46
tableau 3.3 : parametres utilises dans la simulation du reseau de vehicules	48

Table des matières

Résumé :	i
Liste des abréviations	iii
Listes des figures	v
Liste des tableaux	vi
Introduction générale	1

Chapitre 1

État de l'art des VANETs

1.1. Introduction	5
1.2. Réseaux sans fil.....	5
1.2.1. Définition d'un réseau	5
1.2.2. Définition d'un réseau informatique	5
1.2.3. Objectif d'un réseau	5
1.2.4. Classification des réseaux :	6
1.2.5. Définition de réseau sans fil	6
1.2.6. Architecture de réseau sans fil.....	6
1.2.6.1. Mode avec infrastructure.....	6
1.2.6.2. Mode sans infrastructure ou réseau ad hoc.....	7
1.2.6.3. Réseau ad hoc mobile MANET.....	8
1.3. Réseaux VANET	8
1.3.1. Définition.....	8
1.3.2. Nœuds d'un réseau VANET.....	9
1.3.3. Entités de communication	9
1.3.3.1. Unité côté route (RSU).....	9
1.3.3.2. Unité embarquée(OBU).....	9
1.3.3.3. Unité d'application (AU)	10
1.3.4. Architecture des réseaux VANET	10
1.3.4.1. Mode de communication V2V	10
1.3.4.2. Mode de communication (V2I)	10
1.3.4.3. Mode de communication hybride	11
1.3.5. Caractéristiques des réseaux VANETs.....	12
1.3.5.1. Collecte d'informations.....	12
1.3.5.2. Capacité d'énergie.....	12

1.3.5.3. Environnement de déplacement	12
1.3.5.4. Forte mobilité	12
1.4. Protocoles de routage dans VANETs	12
1.4.1. Routage dans Ad-hoc	12
1.4.2. Routage dans VANETs	13
1.4.3. Classification des protocoles de routages	13
1.4.3.1 .Classification selon la topologie.....	13
1.4.3.2. Classification selon la géographie.....	17
1.4.4. Grille d'analyse	18
1.5. Conclusion.....	20

Chapitre2

analyse comparative des simulateurs des VANETs

2.1. Introduction.....	22
2.2. Avantage de la simulation	22
2.3. Objectif de la simulation	23
2.4. Déroulement de la simulation.....	23
2.5. Présentation des simulateurs	23
2.5.1. Préambule.....	23
2.5.2. Simulateurs de mobilité des VANETs	24
2.5.2.1. Simulateur SUMO.....	24
2.5.2.2. Simulateur MOVE.....	24
2.5.2.3. Simulateur CITY MOB	24
2.5.2.4. VANET MOBISIM.....	25
2.5.2.5. Simulateur STRAW	25
2.5.2.6. Logiciel BOONMOTION	25
2.5.3. Comparaison entre les simulateurs de mobilité.....	26
2.5.4. Simulateurs de réseaux	26
2.5.4.1. Simulateur de réseau NS2	26
2.5.4.2. Simulateur de réseau NS3	27
2.5.4.3. MATLAB.....	27
2.5.4.4. OMNET++	27
2.5.4.5. OPENET.....	28
2.5.4.6. Simulateur GLOMOSIM.....	28
2.5.5. Comparaison entre les simulateurs de réseaux.....	28

2.5.6. Simulateurs intègres	29
2.5.6.1. Simulateur NCTUns	29
2.5.6.2. Simulateur TRANS	30
2.5.6.3. Simulateur GROOVENET	30
2.5.7. Comparaison entre les simulateurs intègres	30
2.5.8. Analyse des simulateurs	31
2.6. Synthèse et choix retenu	31
2.7. Vue générale sur NS3 et SUMO	32
2.7.1. NS3	32
2.7.2. Simulateur SUMO	33
2.8. Conclusion	33

Chapitre 03

Simulation et analyse des résultats

3.1. Introduction	35
3.2. Etude détaillée du protocole AODV	35
3.2.1. Définition	35
3.2.2. Principe de fonctionnement	35
3.3. Critère et métrique d'évaluation	38
3.3.1. Taux de livraison de paquets (PDR)	38
3.3.2. Délai moyen de bout en bout (EED)	39
3.3.3. Charge de routage normalisé (NRL)	39
3.3.4. Débit moyen	40
3.3.5. Paquet perdu (PLR)	40
3.3.6. Scenarios utilisés	40
3.4. Déroulement de la simulation	41
3.4.1. Matériels et outils	41
3.4.2. Scenario de simulation	41
3.4.3. Étapes de la simulation	42
3.4.3.1. Étape 1 : création du Map	42
3.4.3.2. Étape 2 : programme NS3	45
3.4.3.3. Exécution du programme	48
3.2. Analyses des résultats de simulation	50
3.5.1. Scenario 1 : Interprétation des résultats selon la variation de nombre des nœuds	50
3.5.1.1. PDR vs Densité	50

3.5.1.2. Délai vs densité	51
3.5.1.3. NRL vs densité	52
3.5.1.4. Débit vs densité	53
3.5.1.5. Paquets perdus vs densité	55
3.5.2. Scenario 2 : Interprétation des résultats selon la variation de vitesse des nœuds.	56
3.5.2.1. PDR vs vitesse.....	56
3.5.2.2. Délai vs vitesse.....	57
3.5.2.3. NRL vs vitesse.....	58
3.5.2.4. Débit vs vitesse.....	59
3.5.2.5. PL vs vitesse.....	60
3.6. Conclusion	61
conclusion generale et perspectives	62
Bibliographie	VII
Annexe A : installation des outils	X
Annexe B : étapes de la simulation.....	XV
Annexe C : script vanet-routing-compare.....	XVIII

Introduction

générale

Introduction générale

Ces dernières années, les réseaux sans fil ont connu une croissance fulgurante et connaissent aujourd'hui un succès. Leur apparition a conduit, d'une part au progrès des technologies de la communication et de l'information et d'autre part, a donné naissance au «système de transport intelligent», qui vise à en faire un système routier efficace.

Les automobiles occupent une place très importante dans la vie des gens, entretemps elles sont le moyen de transport le plus dangereux parmi les autres. Pour contourner ce problème, la communauté scientifique s'est concentrée sur cette question, pour obtenir des solutions afin de rendre ce moyen de transport plus sécurisé.

Pour atteindre ce noble objectif, les chercheurs ont développé un concept, basé sur la communication entre véhicules, comme la construction d'un réseau sans fil dynamique (VANET) où les nœuds sont des véhicules.

L'interconnexion des réseaux (VANET), est considérée comme l'une des dernières technologies de communication sans fil. Les automobiles actuels ne sont plus que de simples moyens de transport, elles sont dotées de sources de renseignement par interaction avec l'environnement routier, à partir des équipements informatiques embarqués et aux équipements informatiques, intégrés aux routes et autoroutes le long du site.

Les véhicules peuvent communiquer entre eux, via des appareils fixes ou mobiles. En tirant parti de leurs capacités de communication récentes, la création d'un réseau dédié peut réduire le temps, tout à bord en améliorant la sécurité routière.

Depuis l'avènement du VANET, de nombreuses normes, applications et mécanismes de routage, ont été proposés pour répondre aux spécifications de ce nouveau type de réseau. Le routage est un élément essentiel dans un réseau VANET. Il représente un grand défi à relever pour sa conception en raison de la forte mobilité des véhicules, de la diversité spatio-temporelle de la densité du trafic et de la propagation des ondes radio, en environnement extérieur défavorable à l'établissement des communications sans fil.

Différents protocoles de routage ont été conçus, font l'objet d'un grand effort de tests, d'évaluations et d'améliorations. Pour réaliser ces tâches dans un environnement réel avec une topologie complexe, comme celle des réseaux véhiculaires ça peut être difficile, aussi bien sur le plan économique que logistique. Pour contourner ce problème, la simulation est le moyen le plus facile, moins cher et le plus largement utilisé.

Dans le cadre de ce mémoire, l'évaluation de certains aspects de performances de protocoles de routage, sera abordée par des simulations sous NS3 et SUMO. Elle permet de faire une comparaison efficace entre les protocoles, en évoluant quelques paramètres comme, le débit de transmission, les pertes des paquets et le délai de transmission ainsi que la charge de routage normalisé et le taux de livraison des paquets dans l'environnement de VANETs.

Ce mémoire est structuré en trois chapitres :

Dans le premier chapitre, nous présentons un état de l'art et une vue d'ensemble sur les réseaux véhiculaires Ad-Hoc VANET, leurs caractéristiques et les protocoles de routage les plus utilisés.

Dans le deuxième chapitre, nous présentons les divers logiciel de simulation des réseaux VANETs, une petite comparaison est donnée, entre les simulateurs de réseau, et enfin une présentation du simulateur de mobilité SUMO a été aborder.

Le troisième chapitre est consacré à l'évaluation des protocoles AODV et OLSR, DSR et DSDV. Nous effectuons en premier lieu une présentation détaillé des protocoles de routage utilisé, ensuite nous procédons au comparaison de performances entre ces protocoles en termes de taux de livraison de paquets, délai de bout en bout, nombre de paquets perdus et débit moyen et charge de routage. Ainsi nous présentons les résultats obtenus au cours de la simulation.

Nous terminons ce mémoire par une conclusion pour résumer tous ce qui est essentiels de notre étude et perspective de recherche.

Chapitre 1

État de l'art des VANETs

1.1. Introduction

Les réseaux VANETs (en anglais: Vehicular Ad-hoc Network), constituent une nouvelle forme des Réseaux MANET (en anglais : Mobile Ad-hoc Network), ou les nœuds de réseau VANET sont des véhicules. Ils permettent d'établir des communications entre les véhicules, ou bien les véhicules avec l'infrastructure, située aux bords de routes. Les réseaux VANETs, sont caractérisés par une topologie dynamique et une forte mobilité des nœuds. Ils offrent des services liés à la sécurité routière et les services liés au confort.

Ce chapitre présente les réseaux sans fil et les réseaux VANETs, leurs modes de communications, leurs caractéristiques et leurs objectifs. Ensuite, une représentation des protocoles de routage utilisés pour l'interconnexion des VANETs.

1.2. Réseaux sans fil

1.2.1. Définition d'un réseau

Un réseau est un ensemble d'appareils et de programmes qui sont liés entre eux et modifiés pour pouvoir communiquer. Un réseau se caractérise par le nature de la technologie ou de protocole de transmission des données qu'il utilise [1].

1.2.2. Définition d'un réseau informatique

Un réseau désigne une série de machines ou nœuds interconnectés par des liens et chemins de communication (câble, sans fil ou par onde radio) permettant l'échange de données et de ressources communes. Les réseaux peuvent eux-mêmes s'interconnecter avec les autres réseaux et contenir des sous-réseaux. Les réseaux informatiques sont utilisés dans plusieurs domaines tels que l'industrie, automobiles, etc.

1.2.3. Objectif d'un réseau

Les objectifs d'un réseau sont classiquement les suivants [1] :

- permettre l'exploitation à distance des systèmes informatiques;

- La réduction des coûts.
- Développer et harmoniser les méthodes
- L'échange de l'information ;
- Création des liens de Communication entre les entités.
- Aider à la création des bases des données numériques.

1.2.4. Classification des réseaux :

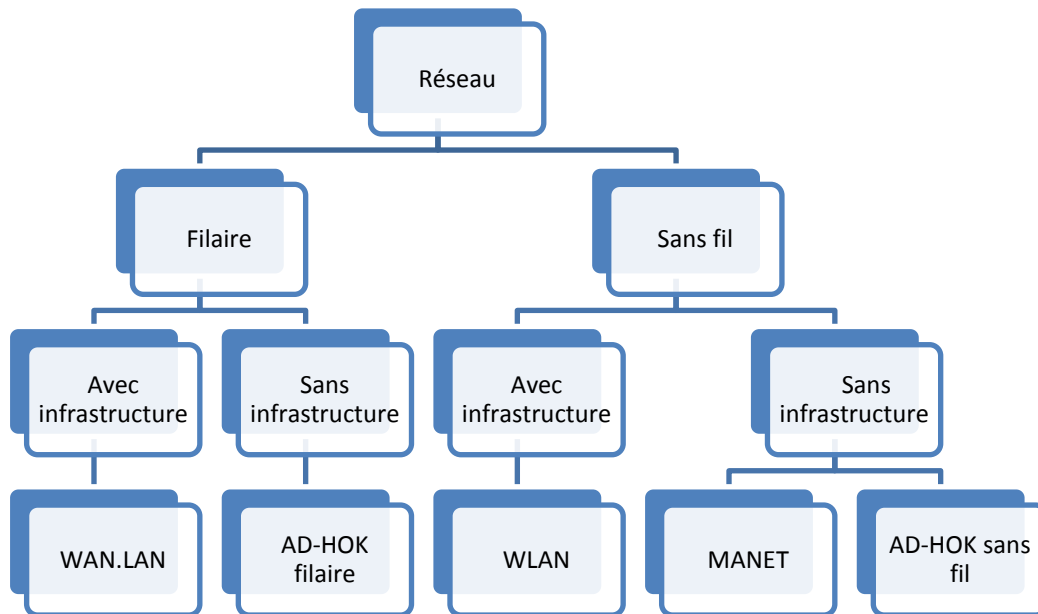


Figure 1. 1. Classification des réseaux. [2]

1.2.5. Définition de réseau sans fil

Un réseau est dit sans fil lorsque la communication entre ces éléments (téléphones portable, ordinateurs, etc) n'utilisent pas de fils conducteurs électriques, donc ce réseau permet de connecter différents nœuds sans l'aide d'une connexion physique mais qui établit la communication par des ondes électromagnétiques, ce qui représente plus de confort par rapport aux réseaux filaires.

1.2.6. Architecture de réseau sans fil

1.2.6.1. Mode avec infrastructure

Ce mode désigne un réseau avec infrastructure permettant l'échange d'information entre les différentes stations du réseau, Cette infrastructure est basée sur un matériel qui fournit plusieurs services. Ce matériel est appelé un point d'accès (PA) ou station de base. Le point

d'accès est doté d'une interface de communication sans fil établir la communication avec les stations mobiles [2].

L'ensemble composé par le point d'accès et les stations situés dans sa zone de couverture est appelé ensemble de services de base (en anglais basic service set, noté BSS) et constitue une cellule [3].

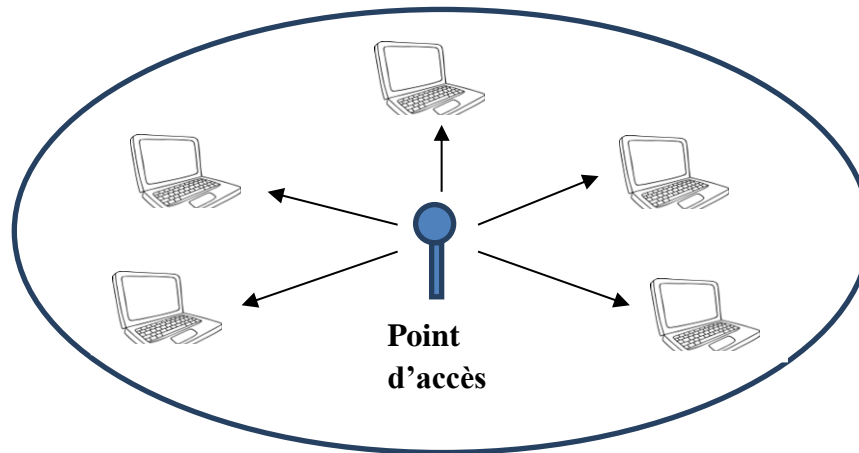


Figure 1. 2. Basic Service Set [3]

1.2.6.2. Mode sans infrastructure ou réseau ad hoc

Dans ce mode les points d'accès n'est pas nécessaires, et les machines sans fils clientes se connectent les unes aux autres pour constituer un réseau point à point (peer to peer en anglais), dans ce mode chaque machine joue en même temps le rôle de client et le rôle de point d'accès.

L'ensemble formé par les différentes stations est appelé ensemble de services de base indépendants (en anglais indépendant basic service set, abrégé en IBSS) [3].

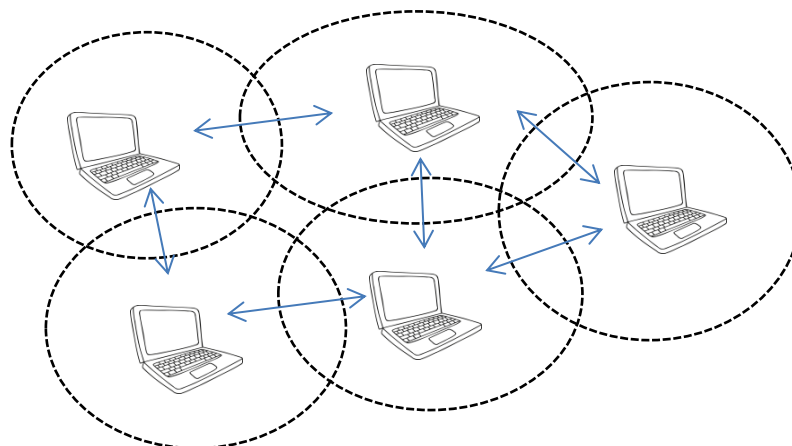


Figure 1. 3. Indépendant Basic Service Set [3]

1.2.6.3. Réseau ad hoc mobile MANET

Un réseau mobile ad-hoc ou réseau MANET, est un réseau qui utilise la transmission avec ondes radio et sans infrastructure dans lesquels tous les nœuds peuvent être mobiles. Ce réseau marche d'une façon autonome composé de stations mobiles ou de nœuds, qui peuvent interconnecter entre eux s'ils sont à portée de la zone de couverture [2.3].

1.3. Réseaux VANET

1.3.1. Définition

VANET (Vehicular Ad-Hoc Network) c'est une technologie sous forme de MANET qui utilise les voitures intelligentes comme nœuds dans un réseau pour créer un réseau mobile. VANET transforme chaque voiture participante en un routeur ou nœud sans fil, permettant aux voitures situées à environ 100 à 300 mètres l'une de l'autre de se connecter et, à leur tour, de créer un réseau avec une large portée.

D'autre part, VANET devient un domaine de recherche émergent en raison de son rôle dans la conception de système de transport intelligent afin de réduire les accidents de la circulation et de rendre la conduite plus confortable [4].

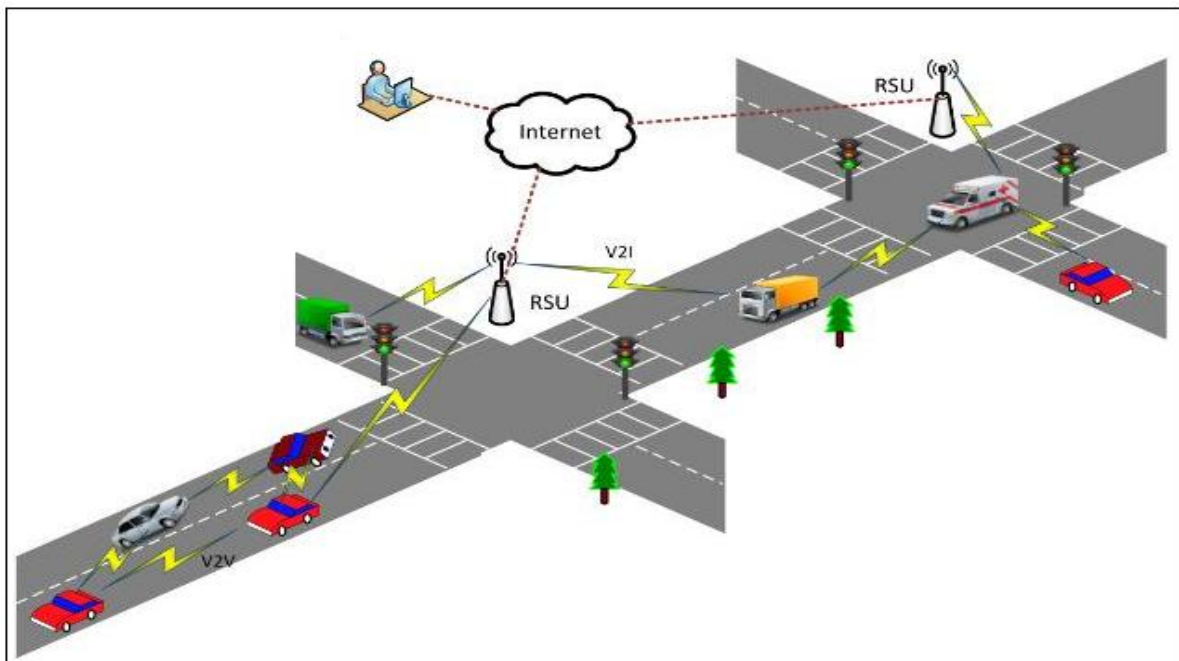


Figure 1. 4. Réseau VANET [4]

1.3.2. Nœuds d'un réseau VANET

Le véhicule intelligent représenté dans la figure 1.5, est un nœud d'un réseau VANETS, il est équipé de plusieurs appareils, tel que des interfaces réseau, capteurs, ordinateurs, afin de communiquer et traiter les informations avec d'autres véhicules.

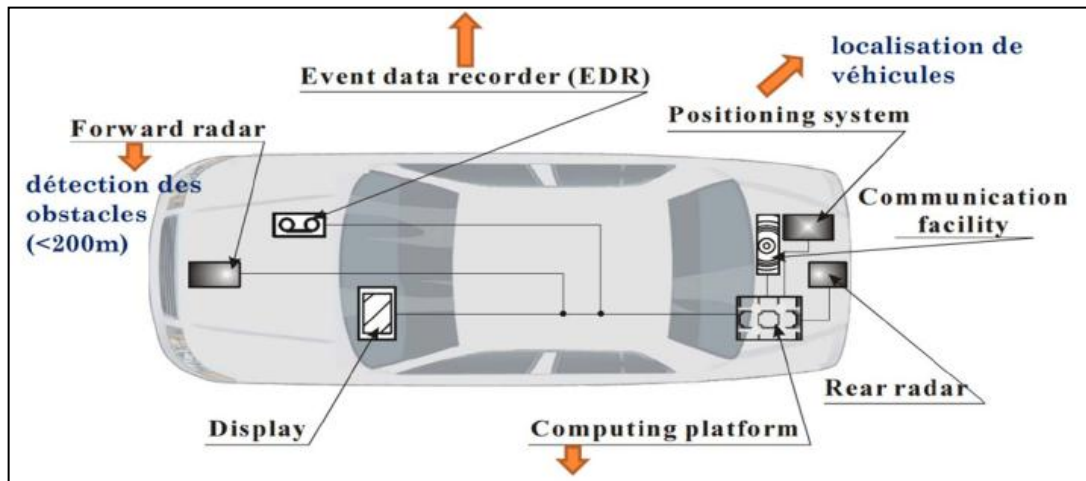


Figure 1. 5. Véhicule intelligent [4]

1.3.3. Entités de communication

Les véhicules sont considérés comme entités du réseau VANETS. Ils possèdent de nombreux capteurs et unités de calcul à bord permettant de gérer et traiter les informations reçues. Les véhicules sont équipés d'une unité embarquée « On Board Unit» (OBU), l'unité d'application AU (application unit) et l'unité du route RSU (Road Side Unit) [4].

1.3.3.1. Unité côté route (RSU)

Le RSU (Road Side Unit) est un petit dispositif installé au bord de la route utilisé comme un point d'accès pour assurer les communications avec d'autres RSUs et échanger les informations relatives à l'image du trafic routier avec les utilisateurs de la route. Les RSU sont installés sur les feux de circulation, les panneaux routiers et aussi les lampadaires. Les RSU disposent d'un émetteur/récepteur DSRC et d'un processeur qui offre des services de sécurité et autre informations pour les multiples OBU situées dans la zone de couverture [6].

1.3.3.2. Unité embarquée(OBU)

Les véhicules communiquent entre eux via l'OBU (On Board Unit). Ces derniers représentent un ensemble d'équipements installés dans les véhicules, tels que: radars, caméras, système de localisation GPS, plateforme de traitement [4].

1.3.3.3. Unité d'application (AU)

L'AU est un dispositif électronique installé dans les véhicules pour assurer les communications avec l'autorité de confiance (CA), connecté à l'OBU afin d'exécuter des applications pour le confort et sécurité [6].

1.3.4. Architecture des réseaux VANET

1.3.4.1. Mode de communication V2V

C'est un mode de communication véhicule à véhicule, V2V, la communication est rétablis directement sans infrastructure, le coût de communication est plus bas et plus flexible.

Les communications V2V sont très efficaces pour le transfert des informations de la sécurité routière, mais elles ne garantissent pas une connectivité permanente entre les véhicules [1, 8].

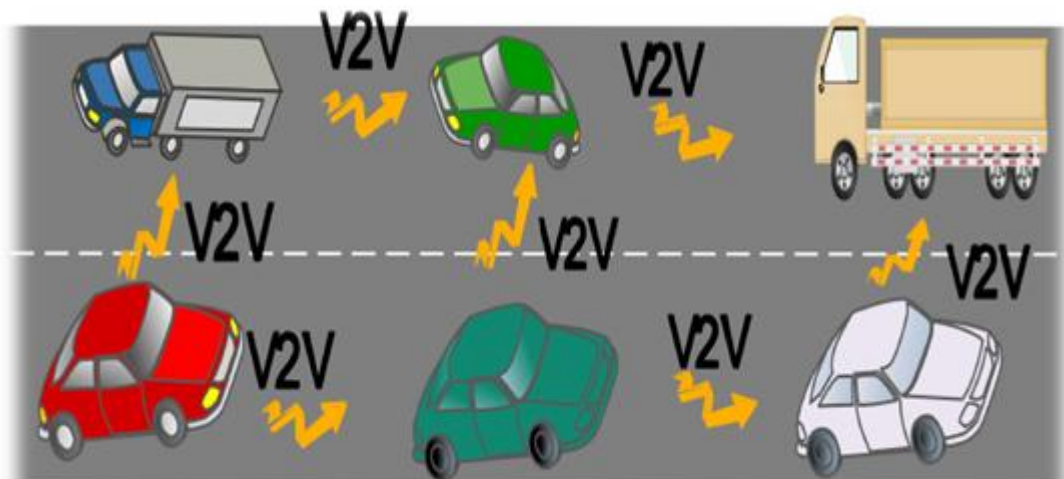


Figure 1. 6. Communication V2V [4]

1.3.4.2. Mode de communication (V2I)

C'est un mode de communication véhicule à infrastructure (V2I). Ce mode de communication est assuré grâce aux différentes entités du réseau VANETS, à savoir les OBUs, les RSUs placés aux bords des routes et aussi les TA (Trusted Authority) .tous les entités du réseau sont utilisées pour assurer les communications dans le réseau véhiculaire.

Le mode V2I permet aux véhicules d'accéder à internet et de rapatrier les informations météorologiques [8.5].

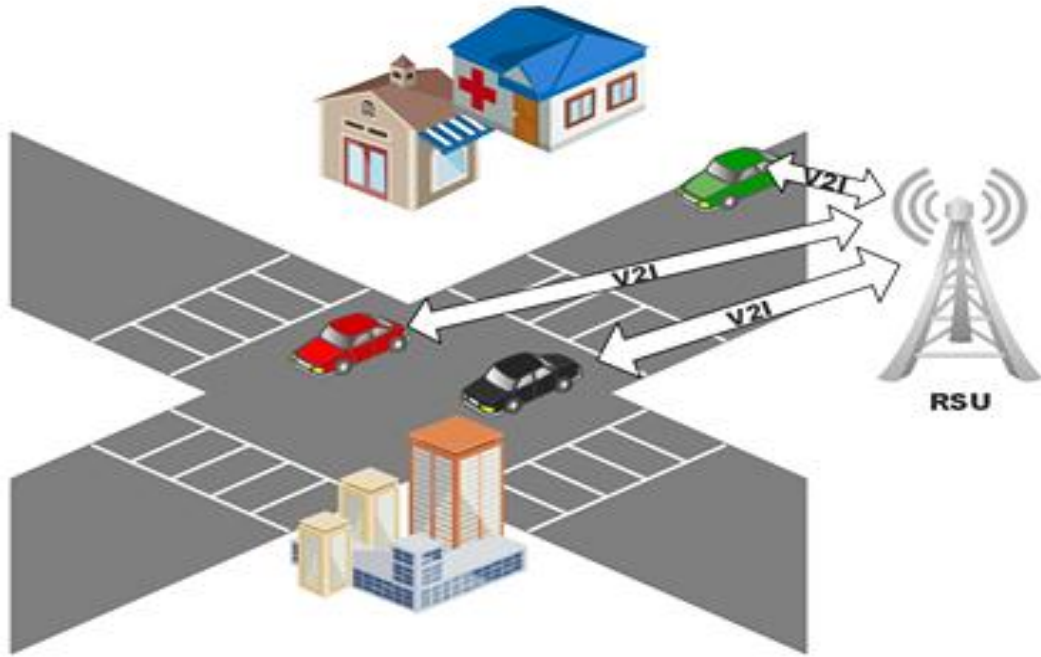


Figure 1. 7. Communication V2I [5]

1.3.4.3. Mode de communication hybride

La communication hybride rassemble les deux modes : véhicule infrastructure et véhicule-véhicule. Les véhicules se comportent comme des relais de capture et de partage de l'information. Le mode V2V complète V2I et grâce à cette combinaison la couverture des infrastructures sera maximal [5].

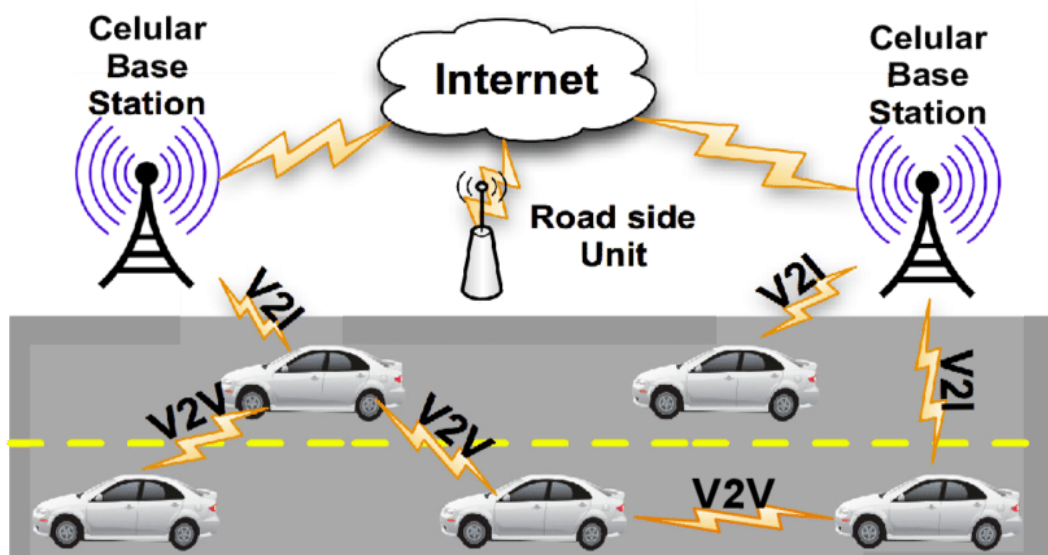


Figure 1. 8. Communication hybride [5]

1.3.5. Caractéristiques des réseaux VANETs

1.3.5.1. Collecte d'informations

Afin de rendre la conduite plus sécurisée et confortable les véhicules intelligents sont équipés de plusieurs capteurs tel que les capteurs de pollution, caméras, sondes, etc. tous ces capteurs servent à récolter les informations du trafic routier [5].

1.3.5.2. Capacité d'énergie

Vue l'utilisation des batteries à grande capacité dans les véhicules intelligents, la capacité d'énergie pour les systèmes VANETs est suffisante pour l'utilisation de plusieurs équipements électroniques [5].

1.3.5.3. Environnement de déplacement

La variation du type d'environnement des réseaux véhiculaire impose des contraintes affectent considérablement le modèle de mobilité et la qualité des transmissions des informations [8].

1.3.5.4. Forte mobilité

La variation de la vitesse des véhicules dans certains environnements provoque une coupure de communication et la rupture des liens entre les éléments de réseau VANETs [5].

1.4. Protocoles de routage dans VANETs

Pour que les véhicules puissent échanger les informations entre eux, les protocoles de routage doivent être bien définis. Puisque les nœuds et stations de base ne sont pas à une portée de transmission radio directe, le routage est exigé pour réussir la communication entre les véhicules.

1.4.1. Routage dans Ad-hoc

Chaque élément dans les réseaux Ad Hoc peut émettre et recevoir des informations, et se comporte comme relais de l'information assurant la circulation des messages dans le réseau de proche en proche. Chaque nœud du réseau, doit supporter les capacités de routage Ad Hoc. Avec ce routage, la zone de couverture radio d'un nœud, peut être étendue en utilisant les nœuds voisins comme relais de l'information.

La conception de protocoles de routage c'est le point le plus important dans les réseaux Ad-Hoc. Cette propriété permet aux protocoles de présenter des bonnes performances, lorsque la topologie du réseau ou le nombre de nœuds varie.

1.4.2. Routage dans VANETs

Le routage dans les VANET est très important, puisque tous les services se basent sur des communications à plusieurs saut pour l'acheminement des informations, le transfert de données, etc. Les communications multidiffusion, sont utilisées dans les applications de gestion de trafic et de sécurité.

1.4.3. Classification des protocoles de routages

Les protocoles de routage dans les réseaux VANETs sont divisés en deux classes : les protocoles basés sur la topologie (réactifs, proactifs et hybrides) et les protocoles basés sur la localisation (géographique).

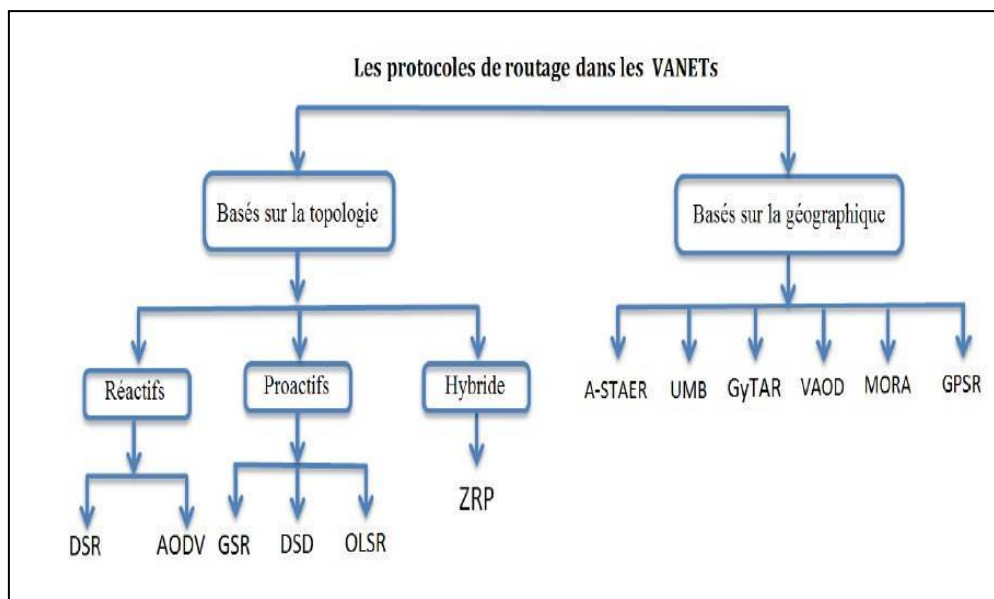


Figure 1. 9. Protocoles de routage dans les réseaux VANETs [6]

1.4.3.1. Classification selon la topologie

A. Protocoles réactifs

Ils adoptent des algorithmes classiques. Les itinéraires sont configurés uniquement sur demande et seuls les itinéraires utilisés sont conservés. Dans ce cas, un temps supplémentaire est nécessaire au début de chaque session pour trouver le chemin

Si un nœud viens d'envoyer des paquets, une étape de découverte de parcoure est lancée avec diffusion de route de recherche de message. Chaque nœud qui reçoit ce message et n'a pas de données sur la cible transfère le message aux nœuds proches. [6].

A.1. Protocole AODV

Le protocole AODV (*Ad hoc On-demand Distance Vector*), procède a la création des routes à la demande et a l'utilisation du principe de numéro de séquence, pour utiliser les nouvelles routes. En plus il utilise le nombre de sauts, pour choisir entre plusieurs chemins disponibles.

les paquets utilisés par AODV sont : les paquets de requête de chemin RREQ (*Route Request Message*), les paquets de réponse de chemin RREP (*Route Reply Message*) et les paquets erreur route RERR (*Rout Error Message*). AODV utilise aussi des paquets de contrôle HELLO qui servent à vérifier la connectivité des routes.

AODV utilise deux principes : la découverte de route qui permet de trouver le chemin pour atteindre une destination, et le deuxième principe c'est la maintenance de route afin de détecter et signaler les erreurs et la perte des liens de communications [7].

A.2. Protocole DSR

Le protocole DSR (*Dynamic Source Routing*) est responsable de la création des itinéraires à la demande comme le protocole AODV avec une méthode de routage source.

La source est introduise dans l'en-tête du paquet sur tout le chemin. Les nœuds intermédiaires n'ont pas besoin de contenir d'informations sur le chemin parcouru depuis le chemin complet. Le protocole DSR se compose de deux mécanismes: la détection des itinéraires et la maintenance des itinéraires.

B. Protocoles proactifs

Les nœuds dans Les protocoles proactifs construisent les tables de routage avant que la demande soit effectuée. Ils enregistrent à chaque instant la topologie du réseau.

B.1. Protocole OLSR

Le protocole OLSR (*Optimised Link State Roting*) fait partie des protocoles de routage proactifs. Il utilise la même philosophie que les protocoles de routage utilises dans les réseaux filaires conventionnel : méthode par états de lien (*link state routing*) et méthode par vecteur de distance (*distance vector routing*).

Chaque nœud découvre ses voisins, et informe tout le réseau de son voisinage par diffusion afin de fournir des routes de plus court chemin vers une bonne destination.

OLSR économise les ressources radio pendant les transmissions. Ceci est réalisable avec l'utilisation de la technique des relais multipoint (MPR). Chaque nœud construit un sous-réseau avec ses voisins appelé MPR. Les nœuds de ce groupe sont utilisés pour l'acheminement et la retransmission des messages reçus. [19-21].

B.2. Protocole DSDV

Le protocole DSDV (Destination-Sequenced Distance-Vector) est un protocole proactif Basé sur L'algorithme de vecteur de distance de Bellman-Ford (Table-Drive)

Chaque nœud stocke la table de routage dans sa mémoire et gère le routage contenant les informations sur les destinations disponibles sur le réseau. Ces informations contiennent le numéro du prochain nœud utilisé pour atteindre la destination. Ce numéro de séquence est utilisé pour choisir entre les anciens et les nouveaux itinéraires.

Chaque nœud envoie périodiquement toute sa table de routage à ses nœuds voisins. Vu le changement de la topologie du réseau, d'autres paquets seront envoyés. Ces paquets contiennent les entrées de la table modifiée et sont utilisés pour partager les informations de routage rapidement. [17, 18].

A chaque fois les nœuds faites une comparaison entre les informations dans leurs tables de routage et les paquets de mise à jour reçus. [17, 18].

B.3. Protocole GSR

Le protocole GSR (*Global Routing State*) est un protocole similaire au protocole DSDV. Ce protocole utilise et améliore la technique de routage basé dur l'état des liens afin d'éviter le mécanisme non efficace de l'inondation des messages de routage.

Les nœuds connaissent la topologie du réseau, et ne partagent pas leurs états de liens à tout le réseau, mais ils les envoyer aux voisins seulement. Aussi, La taille des paquets de mise à jour augmente si le réseau contient un grand nombre de nœuds [7].

C. Protocoles hybrides : Protocole ZRP

Le Protocole de routage de zone (Zone Routing Protocol) est un protocole de routage hybride. Il met en place, simultanément, un routage proactif et un routage réactif, afin de combiner les avantages des deux approches, il découpe le réseau en différentes zones de

roulage de fait que Le routage au sein d'une zone se fait de manière proactive, et le routage vers les nœuds extérieurs de la zone se fait de façon réactive [17, 22].

ZRP utilise le protocole de contrôle d'accès au support (MAC) afin de connaître les voisins immédiats et exploite le protocole NDP (*Neighbour Discovery Protocol*) pour partager les messages HELLO.

D. Comparaison entre les protocoles proactifs et réactifs

Les nœuds ne conservent pas les informations sur la topologie globale du réseau pour les protocoles réactifs. Elles ne stockent que les informations sur les routes actives.

Par contre dans un protocole de routage proactif, la topologie du réseau est connue et les routes sont disponibles immédiatement.

Protocoles proactifs		Protocoles réactifs	
avantages	inconvénients	avantages	inconvénients
La topologie du réseau est connue et les routes sont disponible immédiatement	Diffusion régulièrement des informations sur le changement de topologie du réseau	Les nœuds ne conservent pas les informations sur la topologie globale du réseau : seules les informations sur les routes actives sont stockées	/
Disposition en permanence d'une route pour chaque destination dans le réseau	Volume de signalisation plus grand	Ils génèrent un volume plus faible de signalisation	Ils engendrent un délai lors de la construction des routes et produisent plus difficilement des routes optimales

Tableau 1. 1. Protocoles proactifs vs réactifs [9]

1.4.3.2. Classification selon la géographie

a. Protocole A-STAR

Le protocole A-STAR (*Anchor-based Street and Traffic Aware Routing*) utilise la position pour la communication dans les réseaux véhiculaire. Il est spécialement conçu pour les scénarios de veille, Il traduit les informations sur les itinéraires des bus de la ville afin d'identifier une route, et ceci donnent une idée sur l'état du réseau véhiculaire dans chaque rue [7.8].

b. Protocole UMB

Le protocole UMB (*Urban Multi hop Broad cast Protocol*), est un protocole basé sur la diffusion multi-saut, pour les réseaux avec infrastructure, il permet de réduire les collisions et d'utiliser de la bande passante d'une façon efficace. Le protocole UMB sécurise les opérations d'envoi et la reconnaissance des paquets aux nœuds les plus éloignés, sans connaître les informations sur la topologie du réseau.

UMB utilise deux types de diffusion, la première appelée diffusion directionnelle, où la sélection des nœuds se fait selon la direction de diffusion pour l'interconnexion sans connaître la topologie. La deuxième c'est la diffusion multidirectionnelle aux intersections pour dispatcher les paquets dans toutes les directions [8].

c. Protocole GyTAR

Le protocole de routage GyTAR (*improved Greedy Traffic-Aware Routing protocol*) est basé sur la localisation géographique dans les réseaux véhiculaires.

GyTAR prend en considération le routage des données de proche en proche, et chaque nœud connaît sa position courante à partir du GPS. Aussi, un nœud source connaît la position du destinataire, pour pouvoir prendre des décisions de routage avec le service de localisation GLS (*Grid Location Service*) et peut déterminer la position des intersections voisines à travers des cartes électroniques.

d. Protocole VADD

Le protocole VADD (*Vehicle-Assisted Data Delivery*) utilise le contexte des réseaux de véhicules et exploite le déplacement des nœuds, pour retransmettre ou non les messages. Il traite les messages sur le trafic routier pour calculer le délai d'un paquet. Les paquets seront orientés sur le long d'un chemin avec un faible délai de bout en bout.

1.4.4. Grille d'analyse

Le tableau ci-dessous représente une analyse détaillé sur les protocoles de routages les plus utilisé dans les réseaux véhiculaires.

Contrainte de performance	AODV	DSR	OLSR	DSDV	GSR	ZRP	GPSR
Catégorie	Réactif	Réactif	Proactif	Proactif	Proactif	Hybride	Géographique (réactif)
Type de protocole	Vecteur de distance	Schéma à état de lien	Schéma à état de lien	Vecteur de distance	Schéma à état de lien	Indéfini	Vecteur de distance
Routes maintenues dans	Table de routage	Table de routage	Table de routage	Table de routage	Table de topologie	Table de routage	Table de position
Liberté de boucle	Oui	Oui	Oui	Oui	Non	Oui	Non
Routes multiples	Non	Oui	Non	Oui	Non	Oui	Oui
Multicast	Oui	Non	Oui	Oui	Oui	Oui	Oui
Surcharge réseau	Modérée	Modérée	Minimale	Minimale	Modérée	Modérée	Modérée
Diffusion périodique	Possible	Possible	Possible	Possible	Pas possible	Possible	Possible
Requiert des séquences de données	Oui	Non	Non	Non	Oui	-	Non
Méthode de reconfiguration des routes	Suppression des routes & notification à la source	Message d'erreur et suppression du chemin sauvegardé	Messages de contrôles envoyés en avance afin d'augmenter la réactivité	Des paquets de mises à jour. Numéro de séquence estampillé	Paquets de mise à jour (table de topologie)	Ce dépend du protocole réactif utilisé	Supprimer le nœud n'appartient pas dans sa zone
Résumé	Découverte des routes, expansion en anneau, recherche, poursuite du chemin.	routes à la demande, routage à la source, la découverte de route et la maintenance de route	Messages de contrôle pour détection de liaison, détection des voisins (MPR), détection de multiples interfaces, calcul des routes.	Des informations sur les destinations numéro de séquence. Envoie périodiquement aux les voisins la totalité de table de routage	Chaque nœud connaît-la topologie globale. L'envoyer aux voisins uniquement.	Chaque nœud connaît les voisins, IARP pour découvrir les routes, IERP est utilisé à la demande pour chercher les routes, BRP utilise les données de la topologie fournies	Même plan. l'acheminement des paquets de données ou de contrôle géographiquement. utilisé deux modes: «Greedy Forwarding» le «Perimeter Forwarding». détermine la route à suivre en minimisant les distances entre les nœuds et la destination.

Tableau 1. 2. Grille d'analyse [10]

1.5. Conclusion

De nombreux débats ont lieu sur l'analyse des performances et l'évolution des protocoles de routage dans l'interconnexion des réseaux VANETs. Dans ce chapitre, les différents protocoles du routage dans les réseaux VANETs ont été cités et analysés. Pour choisir l'un le plus efficace, selon leur classification soit géographique ou topologie, sachant que la réalisation de ces tâches dans un environnement réel avec une topologie complexe, comme celle des réseaux véhiculaires ça peut être difficile, aussi bien sur le plan économique que logistique. Pour contourner ce problème, la simulation est le moyen le plus facile, moins cher et le plus largement utilisé, ce qui sera étudié dans les prochains chapitres.

Chapitre2

**ANALYSE
COMPARATIVE
DES
SIMULATEURS DE
VANETs**

2.1. Introduction

Avant de publier une nouvelle solution, elle doit subir un processus, d'évaluation et de validation. La manière idéale d'accomplir cette tâche, est de pouvoir tester, dans des environnements réels. Cependant, la simulation est favorisée, car c'est la méthode la plus utilisée en raison de la nature de l'environnement distribué et de la topologie complexe des réseaux de véhicules, à travers laquelle les performances de toute solution sont analysées et évaluées.

La simulation d'un réseau VANET contient deux composants qui sont la simulation de la mobilité des véhicules et la simulation du réseau de communication. Le simulateur de la mobilité est responsable de la création des itinéraires qui seront exploités par des nœuds mobiles (véhicules), et le simulateur de réseau tente de générer et de gérer le trafic de données entre les nœuds.

La simulation est réalisée à l'aide d'un ou plusieurs programmes, et chaque programme présente des avantages et des inconvénients, et le choix des programmes est dépend de l'étendue de la compréhension et de la connaissance des langages de programmation, ainsi qu'à la nature des protocoles qui seront étudiés.

Dans ce chapitre, les divers simulateurs des réseaux de communication sont mentionnés et discutés. Mais notre choix de simulateur parmi les autres cités, est tombé sur le simulateur réseau NS3 et le simulateur de mobilité SUMO. Une petite comparaison entre les simulateurs est mentionnée.

2.2. Avantage de la simulation

La simulation est le moyen idéal pour analyser, évaluer et vérifier les performances de toute nouvelle application ou solution. C'est parce qu'elle est la plus facile, la moins chère et la plus sûre. Les simulations sont utilisées, pour capturer le mouvement réaliste des véhicules et reproduire des environnements réels. Elles sont, souvent les plus utilisées, pour tester de nouveaux modèles.

2.3. Objectif de la simulation

Le but principal de la simulation est d'analyser et d'étudier l'effet des modèles de mobilités sur les performances des protocoles de routage appliqués aux réseaux VANET. Afin d'obtenir des résultats de simulation efficaces, le modèle de mobilité doit être aussi logique que possible. A la fin de la simulation, des statistiques et des informations, sont collectées pour l'analyse et l'évaluation, dans le but de réduire les accidents dans le système réel.

2.4. Déroulement de la simulation

La première étape de la simulation commence par la définition de la carte, qui est responsable de définir les voies qui limitent le mouvement du véhicule, ainsi que l'itinéraire emprunté par le véhicule traversé. A la fin de cette partie, les entrées nécessaires pour la deuxième étape sont générées responsable de la simulation du réseau.

Dans le simulateur de réseau, le protocole sur lequel les données seront transmises est spécifié, étant essentiel pour l'évaluation des performances dans le scénario créé. Enfin, la dernière étape de la simulation d'un réseau de véhicules est l'évaluation des résultats générés par l'union des étapes précédentes.

2.5. Présentation des simulateurs

2.5.1. Préambule

Le simulateur VANET intègre des modèles de mobilité, des couches de protocole et une propagation radio. Les appareils de simulation se divisent en deux catégories, la première est celle des appareils de simulation, basés sur la modélisation de mobilité et reposés sur des modèles simplifiés et incomplets, de couches de protocole et de propagation radio. Il intègre des modèles de comportement. Il est spécialement conçu pour VANET.

La seconde catégorie, repose sur leur idée de base d'intégration, dans des simulations de réseaux existants et établis, de propagation radio et de modèles de mobilité spécifiques aux environnements de véhicules. La plupart des simulateurs appartiennent à ce dernier. Le simulateur modélise les mouvements du véhicule à l'aide du modèle de mobilité.

Il existe trois types de simulateurs:

- Simulateurs de trafic: ces simulateurs de route permettent de cartographier le réseau routier et de générer le trafic souhaité (types de véhicules, événements routiers, vitesses, etc.).
- Simulateurs de réseau: ils sont utilisés pour évaluer les performances des protocoles et sont responsables de divers services de réseau.
- Simulateurs intègres.

2.5.2. Simulateurs de mobilité des VANETs

2.5.2.1. Simulateur SUMO

Le simulateur SUMO (Simulation of Urban Mobility) est un simulateur de Mobilité Urbaine développé par le centre aérospatial Allemagne (DLR) en 2001 [22], largement utilisé dans les études des systèmes de transport open source, qui permet une évolution constante par la communauté des développeurs.

C'est un simulateur microscopique qui comprend un ensemble des applications, qui permet de rendre chaque véhicule individuel, la collecte d'informations comme position, vitesse, type de véhicule, voie que vous utiliserez et même créer le réseau routier pour la simulation via l'application «netgen» ou importer des cartes numériques disponibles sur Internet via «net convert» [22]. Cela permet à SUMO de simuler les simples réseaux de trafic et aussi avec environnements plus complexes.

2.5.2.2. Simulateur MOVE

Le Modèle de générateur de Mobilité pour réseaux Véhiculaires MOVE [16] est un simulateur de mobilité à source ouverte, exécuté avec JAVA qui est basé sur la plateforme de simulateur SUMO, avec l'objectif de simplifier la création de modèle de mobilité.

Grâce à MOVE, il est possible de générer deux types de modèles, le premier c'est le modèle de mobilité et le deuxième c'est le modèle de trafic. Il permet à l'utilisateur de créer une topologie personnalisée ou d'importer des cartes.

L'application MOVE contient une interface graphique afin de permettre à l'utilisateur de générer des scénarios de simulation real rapidement.

2.5.2.3. Simulateur CITY MOB

Citymob est un simulateur générateur de modèle de mobilité, basé sur une interface graphique, généralement utilisé dans les réseaux véhiculaires (VANET). CityMob peut

générer des graphes pour les scénarios VANET à l'aide de trois modèles de mobilité : Simple, Manhattan et Downtown.

Modèle simple (SM): c'est un modèle de mobilité verticale et horizontale, sans changements de direction.

Modèle Manhattan (MM): reproduise la ville comme une grille de style Manhattan, avec une taille de bloc uniforme dans toute la zone de simulation.

Modèle Downtown (DM): Ce modèle présente les mêmes caractéristiques que le Modèle de Manhattan, comme les feux de circulation et la limitation de la route, mais ajoute la densité du trafic routier [17].

2.5.2.4. VANET MOBISIM

Le simulateur VANETMOBISIM [18] est un simulateur à code ouvert basé sur la technologie JAVA qui permet la génération de modèles de mobilité une fois macroscopiques et autre fois microscopique. Le niveau macroscopique, il permet d'utiliser les cartes depuis la base de données Encodage géographique typologiquement intégré (TIGER).

Le niveau microscopique, VANETMOBISIM prend en charge différents modèles de mobilité, tels que Système modèle de conduite avec gestion des intersections, le modèle de dépassement.

2.5.2.5. Simulateur STRAW

Le simulateur de mobilité STRAW (Street Random Waypoint) [19] fait partie du projet C3 (*Car-to-car Cooperation*) et est basé sur le modèle *Random Way Point* (RWP), un modèle largement utilisé dans les réseaux MANET, mais contrairement au modèle RWP traditionnel, ce modèle détermine l'itinéraire d'un véhicule sur chaque intersection. STRAW est un modèle de trafic réaliste utilise les cartes réelles.

2.5.2.6. Logiciel BOONMOTION

Le simulateur Boon Motion [27] est un logiciel basé sur la technologie JAVA qui permet de créer et analyser la mobilité, il est utilisée beaucoup plus dans l'étude des réseaux mobiles ad hoc. Il prend en charge plusieurs modèles de mobilité : Random Waypoint, Gauss-Markov, Manhattan Grid, etc.

La création des modèles de mobilité est plus simple, permettant la configuration de plusieurs paramètres.

2.5.3. Comparaison entre les simulateurs de mobilité

Le tableau 2.1, ci-dessous présente un résumé comparatif des principales caractéristiques des simulateurs de mobilité [16.10].

	SUMO	MOVE	City Mob	Vanet Mobisim	STRAW	BoonMOTION
LANGUAGE	C++	JAVA	C	JAVA	JAVA	JAVA
Code ouvert	✓	✓	✓	✓	✓	✓
Interface graphique	✓	✓	✓	✓	✓	non
Simulateur réseau	Omnet++ NS2 NS3	NS2-NS3- GloMosi m- Omnet++	NS2-NS3	NS2-NS3- GloMosim- Omnet++	SWANS	NS2-NS3- GloMosim
Exemple disponible	✓	✓	non	✓	non	✓
Calcule des routes	✓	✓	non	✓	✓	non
Scenari réel	✓	✓	non	✓	✓	non
Défini par l'utilisateur	✓	✓	non	✓	non	✓

Tableau 2. 1. Comparaison entre simulateurs de la mobilité

2.5.4. Simulateurs des réseaux

Les simulateurs de réseau sont utilisés pour analyser le comportement du réseau sur les différentes conditions, donc il est possible d'identifier et d'étudier les points d'amélioration des protocoles de routage.

2.5.4.1. Simulateur de réseau NS2

Le simulateur NS2 (Network Simulator 2) [6] est un simulateur a code libre gratuit développé spécialement pour les études des réseaux de communication, il est doté de

plusieurs modules intégrés avec divers composants du réseau, comme: technique de routage, protocoles de transport, applications, modèle de mobilité, etc.

Deux langues de programmation sont utilisées dans NS-2: C++ pour la structure de base et aussi le langage OTCL (*Oriented Object Tool Command Language*) pour l'interface.

Ces deux langages permettant la rapidité et l'efficacité du traitement des grandes quantités de données.

Les résultats récapitulés par le simulateur NS2 sont enregistrés dans un fichier de texte qui facilite visualisation graphiquement avec l'outil NAM (*Network Animator*).

2.5.4.2. Simulateur de réseau NS3

Le simulateur NS3 (Network Simulator 3) [12] ce n'est pas une extension du NS2 mais un nouveau simulateur gratuit et open source. Le NS3 exploite le langage C et parfois Python. Il a de nouvelles fonctionnalités, tels que: la possibilité d'exploitation du plusieurs interfaces réseau, l'utilisation de l'adressage IP, compatible avec les architectures et protocoles Internet comme le modèle 802.11.

Dans tous les cas, les modèles NS2 peuvent être exploités sur le NS3. Les résultats de la simulation peuvent également être visualisés à travers l'outil NAM, aussi, le NS3 génère les fichiers de trace des paquets dans un fichier pcap avec la possibilité d'utiliser d'autres outils pour l'analyse des paquets.

2.5.4.3. MATLAB

Le simulateur MATLAB (*MATrix LABoratory*), est un environnement de développement, utilisé à des fins de calcul numérique avec un langage de programmation répétitif pour analyser les données et résoudre les problèmes scientifiques et techniques, chaque ligne d'un programme Matlab est lue, interprétée et exécutée. MATLAB nécessite une licence payante, il est encore largement utilisé, dans le milieu universitaire, c'est un outil utilisé aussi pour l'évaluation et la modélisation des performances protocoles VANET.

2.5.4.4. OMNET++

Le simulateur OMnet++ (*Objective Modular Network Testbed in C++*), est un simulateur de réseaux utilise le langage C++ [13]. Il offre un environnement de développement basé sur la plateforme Eclipse utilisant de nombreux outils et une interface graphique. Sa principale caractéristique est l'approche par composants, appelée modules, qui permettent l'interconnexion dans des ensembles formant des modules

composites. Cette connexion et cette configuration du réseau de simulation ont lieu dans le NED (Network Description Language) Langage de description de réseau OMNET ++.

Le simulateur OMNeT++ est beaucoup plus utilisé par le simulateur intégré de communication véhiculaire VEINS (VEhicule In Network Simulation) en association avec le simulateur de mobilité SUMO.

2.5.4.5. OPENET

Le simulateur OPNET (*OPTimized Network Engineering Tools*), est un logiciel payant de conception, de modélisation et d'analyse des réseaux de communication [14].

Il a une version académique gratuite, connue sous le nom d'OPNET IT (Guru Academic Edition), capable de simuler le comportement des appareils de un réseau entier : les routeurs, commutateurs, protocoles, serveurs et applications individuelles. En plus il permet de prendre en charge diverses normes de technologie de communication sans fil.

OPENET donne la possibilité de crée manuellement les chemins des nœuds ou d'importer les modèles de déplacements aléatoires. Il est moins utilisé que NS et OMNET++, à cause du coût de la licence [14].

2.5.4.6. Simulateur GLOMOSIM

Le simulateur de système d'information (GLOMOSIM) [20] est un simulateur des réseaux de communication écrit en PARSEC, un langage basé sur C, et développé pour les réseaux MANET. Il utilise les techniques de parallélisme afin de simuler un grand nombre de nœuds.

Ce simulateur possède une interface graphique édité en JAVA permettant l'utilisation de plusieurs Modèles de mobilité exploitent les points de routes aléatoires et les traces. GLOMOSIM ne prend pas en charge la norme IEEE 802.11p, mais peut être ajouté par programmation.

2.5.5. Comparaison entre les simulateurs de réseaux

Le tableau 2.2, présente un résumé comparatif des principales caractéristiques des simulateurs de réseau présentés [16].

<i>Simulateurs</i>		<i>NS2/NS3</i>	<i>MATLAB</i>	<i>Omnet</i> ++	<i>OPNET</i>	<i>glomosim</i>
Divers caractéristiques	Développement en cours	✓	✓	✓	✓	✓
	Langage	C++/otcl	C/C++/java	C++	C++/java	C/PARSE C
	Multi-plateforme	✓	✓	✓	✓	✓
	Licence libre	✓	non			✓
	Interface graphique	✓	✓	✓	✓	✓
	Animation	✓	✓	✓	✓	✓
usage	Prise en main	-	++	-	+	-
	Exemples fournis	✓	✓		✓	✓
	Installation	+++	+++	++	+	+
	Facilité d'usage	-	+	-	-	+
fonction	Rapidité simulation	✓	✓	✓	✓	✓
	Passage à l'échelle	-	++	+	+	++
	Modèle 802.11p	✓		✓	✓	✓

Tableau 2. 2. Comparaison entre simulateurs du réseaux

2.5.6. Simulateurs intégrés

Dans l'approche de simulation de réseau VANET utilisant des simulateurs de réseau et mobilité il est nécessaire de charger le fichier trace généré par le simulateur de mobilité dans le simulateur de réseau, puis effectuez la simulation. Cependant, de cette manière, la mobilité ne peut pas être modifiée après l'importation.

Les simulateurs intégrés rassemblent la simulation de réseau avec la simulation de mobilité de manière intégrée afin de modifier les scénarios sans importer à chaque fois le fichier de trace. Voici les simulateurs intégrées les plus utilisés par les chercheurs.

2.5.6.1. Simulateur NCTUns

Le simulateur de réseau national NCTUns (National Chiao Tung University Network Simulator), a été développé pour la simulation des réseaux de communication, mais à partir de la version 5.0, il intègre un simulateur de trafic routier. NCTUns Ecrit en langage C et possède une interface graphique animé qui facilite l'implémentation des véhicules et des

protocoles. Il présente les avantages de prendre en charge plusieurs protocoles réseau à la fois pour la communication sans fil et pour les réseaux câblés [15].

Le simulateur NCTUns fonctionne seulement sur la plate-forme Linux

2.5.6.2. Simulateur TRANS

Le simulateur d'environnement et de trafic TRANS (Realistic Joint Traffic and Network Simulator) c'est un simulateur intégré qui a été effectué l'intégration entre un simulateur de trafic et de réseau (SUMO et NS2) pour générer un environnement réaliste dans les réseaux VANET [28].

C'est un logiciel gratuit basé sur les langages de programmation C ++ et Java, prennent en charge la norme IEEE 802.11p et permettent la génération de cartes via base de données TIGER. Il fait la simulation des événements sur les routes (accidents, fermetures des routes, etc).

2.5.6.3. Simulateur GROOVENET

Le simulateur hybride pour les réseaux de véhicule à véhicule GROOVENET, c'est un simulateur intégré prend en charge l'analyse et l'étude des réseaux de communication véhicules- véhicules et véhicules-infrastructure.

Avec GROOVENET les véhicules simulés peuvent communiquer avec les véhicules réels, en plus les cartes de la base de données TIGER qui sont disponible librement peuvent être importées.

GROOVENET prend en charge trois formes de communication: une communication multi-hop entre véhicules à travers des canaux DSRC. Une communication avec des nœuds fixes (infrastructure), et une communication mobile dans les applications V2V et V2I [21].

2.5.7. Comparaison entre les simulateurs intègres

Le tableau ci-dessus représente une brève comparaison entre les simulateurs intègres [16].

	NCTUns	TraNS	GrooveNet
gratuit	✓	✓	✓
Générateur trafic	NCTUns	SUMO	GrooveNET
Interface graphique	✓	✓	✓
configuration	difficile	modérer	modérer
utilisation	facile	modérer	difficile
langage	C++	Java	C++
IEEE 802.11P	✓	✓	✓

Tableau 2. 3. Comparaison entre les simulateurs intégrés.

2.5.8. Analyse des simulateurs

Tous les simulateurs ont montré de bonnes caractéristiques soit couplés ou intégrés. Mais le simulateur le plus démarqué était le simulateur NS3, qui avec le NS2 est l'un des plus utilisés par les chercheurs et universitaires dans l'évaluation et les études des réseaux de communication.

Le NS3 est un pur simulateur de réseau et des modèles de mobilité fournis par lui ne rentrent pas dans les réseaux VANET, nécessitant l'utilisation d'un simulateur de mobilité. Et lors de l'évaluation des simulateurs présentés, le SUMO a les meilleures caractéristiques, et de bonnes fonctionnalités, permettant la configuration de nombreux des paramètres qui permettent la création de bons scénarios de simulation, en plus d'être entièrement compatible avec le simulateur NS3.

2.6. Synthèse et choix retenu

En conclusion, et pour le moment, il n'a pas de consensus sur la recommandation d'utiliser un simulateur plutôt qu'un autre. Pour les raisons présentées dans les paragraphes précédents, le simulateur de réseau choisis pour ce travail ainsi que le simulateur de mobilité ont été, le NS3 et SUMO respectivement.

2.7. Vue générale sur NS3 et SUMO

2.7.1. NS3

2.7.1.1. Composants d'un réseau sous NS :

Le modèle de réseau sous NS comprend [6]:

- **Nœud** : il existe deux types : les nœuds unicast et les nœuds multicast.
- **Liens de communication** : pour faire une liaison entre les nœuds.
- **Agents de communication** pour échanger les données entre les nœuds.
- **Application**: utiliser pour la génération du trafic.

2.7.1.2. Définition :

Le NS3 (Network Simulator 3) est un simulateur de réseau de communication, dont le but principal est la recherche et l'éducation. NS3 est un logiciel libre de la troisième version de NS. Mais NS3 n'est pas une version mise à jour des versions précédentes de NS2 mais plutôt un nouveau simulateur. [11.12].

2.7.1.3. Éléments de NS-3 :

1. Outils de visualisation des scénarios de simulation
2. Outils de traçages d'information
3. Outils pour tracer le trafic

2.7.1.4. Domaine d'utilisation de NS-3

- Simulation de protocoles : TCP, UDP, IPv4, IPv6, OLSR, AODV, etc.
- Simulation de médias : Ethernet, WIFI, WiMAX, etc.
- Définition de la topologie du réseau : Statique ou dynamique.
- Exécution d'une implémentation d'un protocole dans NS3

2.7.1.5. Architecture NS-3 :

NS3 utilise le langage de programmation C ou Python

Le NS3 a une bibliothèque C++ qui fournit des modèles de simulation de réseau, qui sont exécutés comme des objets C++ enveloppés avec Python.

Les paquets dans NS3, sont stockés sous forme des tampons de paquets d'octets.

Le simulateur NS3 dispose d'un système intégré, basé sur les attributs, pour gérer les valeurs par défaut et les valeurs pour chaque instance de paramètres de simulation. Il dispose également d'une puissante API de bas niveau, qui permet aux utilisateurs premium d'utiliser la flexibilité pour croire les choses de différentes manières [12].

Le diagramme des composants de ns3 est donné sur la figure 2.1 [12].

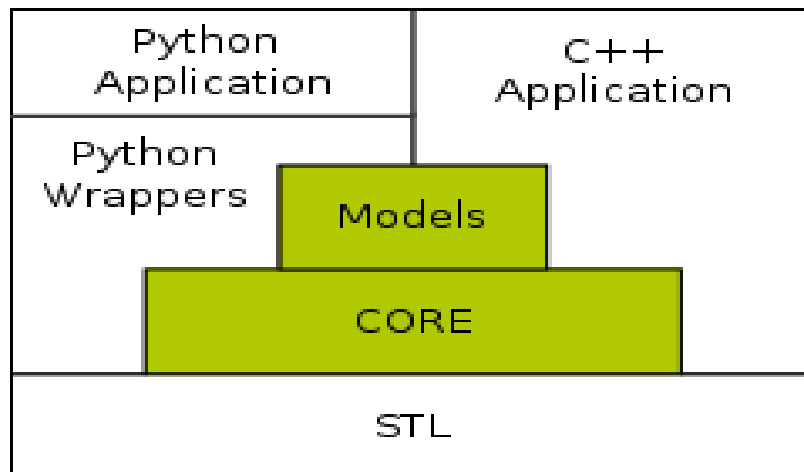


Figure 2. 1. Aarchitecture de NS3

2.7.2. Simulateur SUMO

C'est un logiciel de simulation de trafic routier open source, Il permet de générer la mobilité sur une zone géographique de la taille d'une ville. Le premier but de SUMO est de pouvoir générer la mobilité basée sur des critères de décision qui ne sont pas aléatoires et qui imitent presque le comportement humain [22].

2.8. Conclusion

Dans ce chapitre, nous avons présenté la plupart des simulateurs des réseaux véhiculaires Ad Hoc. L'étude comparative réalisée entre les divers simulateurs, montre que tous les simulateurs ont de bonnes caractéristiques, qu'ils soient couplés ou intégrés.

Cependant, celui qui s'est le plus démarqué était le simulateur NS3, qui avec le NS2 est l'un des plus utilisés par les chercheurs et universitaires dans l'évaluation et les études des réseaux de communication VANET. aussi le NS3 et SUMO excellent dans la présentation d'améliorations et progrès dans certaines caractéristiques.

Dans le chapitre suivant, nous allons présenter les diverses étapes de simulation des protocoles AODV, DSR, DSDV et OLSR avec une comparaison et analyse des résultats de simulation obtenus.

Chapitre 03

Simulation et analyse des résultats

3.1. Introduction

Ce chapitre évalue les protocoles de routage, qui sont extrêmement importants dans un réseau de communication pour assurer la fiabilité de la livraison des paquets. Cependant, les réseaux VANET ont des caractéristiques particulières, en fonction de la mobilité, ce qui rend le routage difficile.

Au départ, le chapitre commence par une présentation plus détaillée du protocole AODV, afin de bien comprendre les étapes de la simulation. Après cela, une méthodologie et critère d'évaluation sont également mentionnés. Ensuite un détail des étapes de simulation ainsi que le programme sous NS3 sont effectués afin d'exécuter la simulation et d'interpréter les résultats.

3.2. Etude détaillée du protocole AODV

3.2.1. Définition

Le protocole "Routage avec Vecteur de Distance à la Demande" (AODV: Ad-hoc On demande Distance Vector), représente essentiellement une amélioration de l'algorithme DSDV. Il réduit le nombre de diffusions de messages en créant les routes au besoin, contrairement au DSDV qui maintient la totalité des routes. L'AODV est basé sur l'utilisation des deux mécanismes "Découverte de route" et "Maintenance de route" (utilisés par le DSR), en plus du routage nœud-par-nœud, du principe des numéros de séquence et de l'échange périodique du DSDV [23].

3.2.2. Principe de fonctionnement

Ce protocole utilise le principe des numéros de séquence afin de maintenir la consistance des informations de routage. A cause de la mobilité des nœuds dans les réseaux ad hoc, les routes changent fréquemment ce qui fait que les routes maintenues par certains nœuds deviennent invalides. Les numéros de séquence permettent d'utiliser les routes les plus nouvelles ou autrement dit les plus fraîches (*fresh routes*).

Comme le fait DSR, l'AODV utilise une requête de route dans le but de créer un chemin vers une certaine destination. Cependant, l'AODV maintient les chemins d'une façon distribuée en gardant une table de routage au niveau de chaque nœud de transit appartenant au chemin cherché. Une entrée de la table de routage contient essentiellement :

- L'adresse de la destination.
- Le nœud suivant.
- La distance en nombre de nœud (i.e. le nombre de nœud nécessaire pour atteindre la Destination).
- Le numéro de séquence destination.
- Le temps d'expiration de l'entrée de la table.

Quand un nœud de transit envoie le paquet de la requête à un voisin, il sauvegarde aussi l'identificateur du nœud à partir duquel la première copie de la requête est reçue. Cette information est utilisée pour construire le chemin inverse (figure 3.1) qui sera traversé par le paquet réponse de route (cela induit que l'AODV ne supporte que les liens symétriques). Puisque le paquet réponse de route va être envoyé à la source, les nœuds appartenant au chemin de retour vont modifier leurs tables de routage suivant le chemin contenu dans le paquet de réponse.

Un nœud diffuse une requête de route (RREQ : Route REQuest) dans le cas où il aurait besoin de connaître une route vers une certaine destination et qu'une telle route n'est pas disponible comme montre la fig. 3.1. Cela peut arriver si la destination n'est pas connue au préalable, ou si le chemin existant vers la destination est devenu défaillant (i.e. la métrique qui lui est associée est infinie) [23].

Le champ numéro de séquence destination du paquet RREQ contient la dernière valeur connue du numéro de séquence associé au nœud destination, cette valeur est recopiée de la table de routage.

Si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut. Le numéro de séquence source du paquet RREQ contient la valeur du numéro de séquence du nœud source.

Comme nous avons déjà dit, après la diffusion du RREQ, la source attend le paquet réponse de route (RREP : Route REPLY). Si ce dernier n'est pas reçu pendant une certaine période (appelée RREP_WAIT_TIME), la source peut rediffuser une nouvelle requête

RREQ. A chaque nouvelle diffusion, le champ Broadcast ID du paquet RREQ est incrémenté. Si la requête RREQ est rediffusée.

Un certain nombre de fois (RREQ_RETRIES) sans la réception de réponse, un message d'erreur est délivré à l'application.

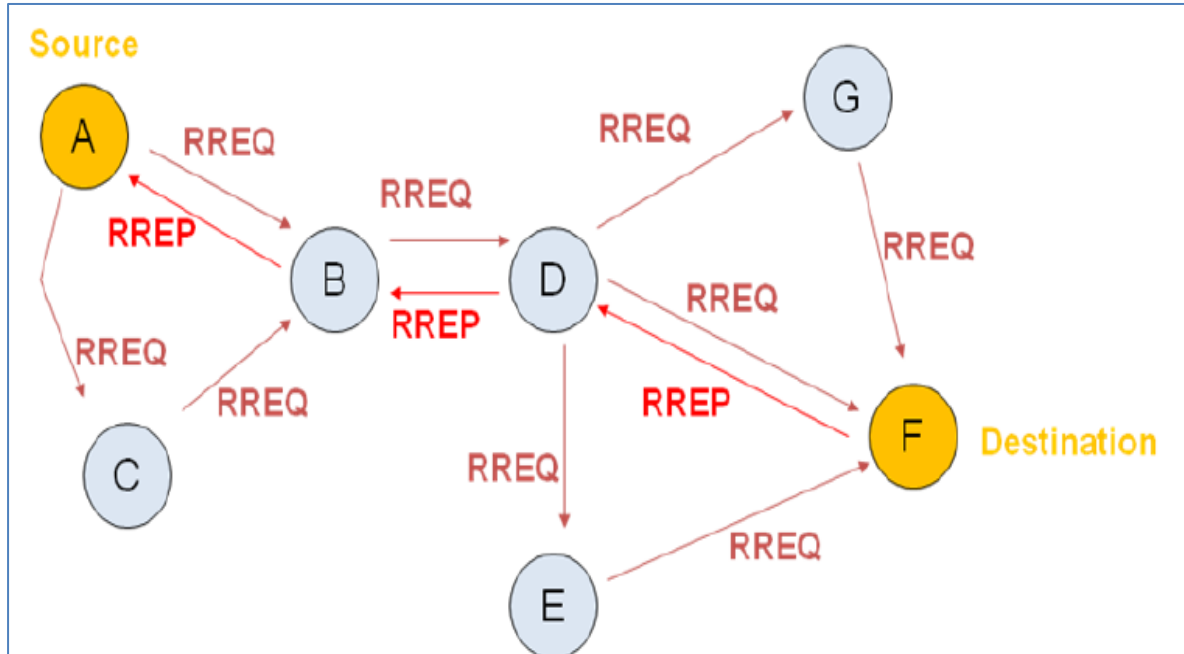


Figure 3. 1. méthode de construction d'une route

Afin de maintenir des routes consistantes, une transmission périodique du message "HELLO" est effectuée. Si trois messages "HELLO" ne sont pas reçus consécutivement à partir d'un nœud voisin, le lien en question est considéré défaillant. Les défaillances des liens sont, généralement, dû à la mobilité du réseau ad hoc. Les mouvements des nœuds qui ne participent pas dans le chemin actif n'affectent pas la consistance des données de routage.

Quand un lien reliant un nœud p à un nœud qui le suit dans le chemin de routage devient défaillant, le nœud p diffuse un paquet UNSOLICITED RREP, avec une valeur de numéro de séquence égale à l'ancienne valeur du paquet RREP incrémentée de un, et une valeur infinie de la distance. Le paquet UNSOLICITED RREP est diffusé aux voisins actifs jusqu'à ce qu'il arrive à la source. Une fois le paquet est reçu, la source peut initier le processus de la découverte de routes.

L'AODV maintient les adresses des voisins à travers lesquels les paquets destinés à un certain nœud arrivent. Un voisin est considéré actif, pour une destination donnée, s'il

délivre au moins un paquet de donnée sans dépasser une certaine période (appelée *active time out period*). Une entrée de la table du routage est active si elle est utilisée par un voisin actif. Le chemin reliant la source et la destination en passant par les entrées actives des tables de routage est dit un chemin actif.

Dans le cas de défaillances de liens, toutes les entrées des tables de routage participant au chemin actif et qui sont concernées par la défaillance sont supprimées. Cela est accompli avec la diffusion du message d'erreur RERR (figure 3.2) entre nœuds actifs.

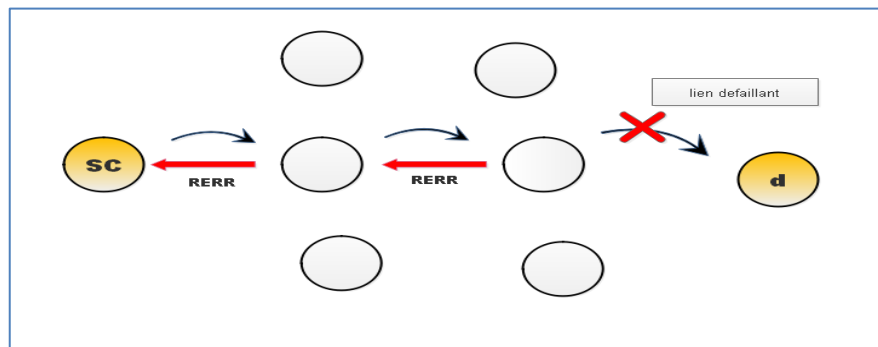


Figure 3. 2. Principe de détection d'un lien défaillant

Le protocole de routage AODV (tout comme le protocole DSR), n'assure pas l'utilisation du meilleur chemin existant entre la destination et la source. Cependant, des évaluations de performances récentes ont montré qu'il n'y a pas de grandes différences (en terme d'optimisation) entre les chemins utilisés par le protocole AODV et celles utilisées par les protocoles basés sur les algorithmes de recherche des plus courts chemins. De plus, le protocole AODV ne présente pas de boucle de routage et évite le problème "counting to infinity" de Bellman-Ford, ce qui offre une convergence rapide quand la topologie du réseau ad hoc change [23].

3.3. Critère et métrique d'évaluation

Notre objectif dans ce mémoire réalisé sous NS-3 est d'évaluer les performances des quatre protocoles de routages AODV et OLSR, DSR et DSDV selon les métriques suivantes :

3.3.1. Taux de livraison de paquets (PDR)

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.

Le taux de livraison de paquet PDR (*Packet Delivery Ratio*) peut être obtenu 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 [52, 54]. Cela est traduits mathématiquement par l'équation 1:

$$PDR = \frac{\sum \text{Nbr des paquets recus par la destinatio}}{\sum \text{Nbr des paquets envoyer par tous les noeuds}} * 100 \quad \text{Eq (3.1)}$$

3.3.2. Délai moyen de bout en bout (EED)

Le délai de bout en bout (*EED, End to End Delay*), est le temps moyen nécessaire à un paquet de données pour atteindre la destination. Cela inclut tous les retards possibles causés par la mise en mémoire tampon pendant la latence de découverte d'itinéraire, la mise en file d'attente dans la file d'attente d'interface, le délai de retransmission au MAC et le délai de propagation.

Cette métrique est calculée en soustrayant le moment auquel le premier paquet a été transmis par la source du moment auquel le premier paquet de données est arrivé à destination. Mathématiquement, il peut être défini comme:

$$\text{Average EED} = \frac{S}{N} \quad (\text{s}) \quad \text{Eq (3.2)}$$

Où **S** est la somme du temps passé à livrer les paquets pour chaque destination, et **N** est le nombre de paquets reçus par tous les nœuds de destination.

3.3.3. Charge de routage normalisé (NRL)

La charge de routage normalisée (*NRL, Normalized Routing Load*) définit le nombre total des paquets de contrôle transmis pendant la simulation par rapport au nombre de paquets de données reçus par les destinations. Elle montre donc à quel point un protocole consomme de la bande passante avec ses messages de routage. Dans les protocoles où le nombre de paquets utilisés par la couche de routage est très élevé, les performances du réseau sont mauvaises.

$$NRL = \frac{\text{nombre total de paquets de routage envoyés par tous les nœuds}}{\text{nombre de paquets de données reçus par les destinations}} (\%) \quad \text{Eq (3.3)}$$

Cette métrique est utilisée pour déterminer la quantité de trafic de routage effectué par le protocole afin de recevoir des paquets de données. Il est calculé comme le rapport du

nombre total de paquets de routage envoyés par tous les nœuds sur le nombre de paquets de données reçus par les destinations

3.3.4. Débit moyen

Le débit des protocoles peut être décrit comme pourcentage du nombre total de paquets reçus par la destination parmi les paquets transmis par la source. Il est la quantité de données par unité de temps qui est fournie à partir d'un nœud à un autre via une liaison de communication. Le débit est mesuré en bits par seconde.

L'unité de mesure de cette métrique est paquet/unité, où TIL est la longueur de l'intervalle de temps [24]. Cela est traduit mathématiquement par l'équation 4 :

$$Debit\ moyen = \frac{\text{taille du packet recus}}{(\text{temps final} - \text{temps initial})} * \frac{8}{1000} \quad (kbit/s) \quad \text{Eq (3.4)}$$

3.3.5. Paquet perdu (PLR)

La perte des paquets (*Packet Loss Ratio*) est le rapport du nombre de paquets perdu au nombre de paquets provenant de la source. Mathématiquement, peut être représenté par l'équation (5 et 6).

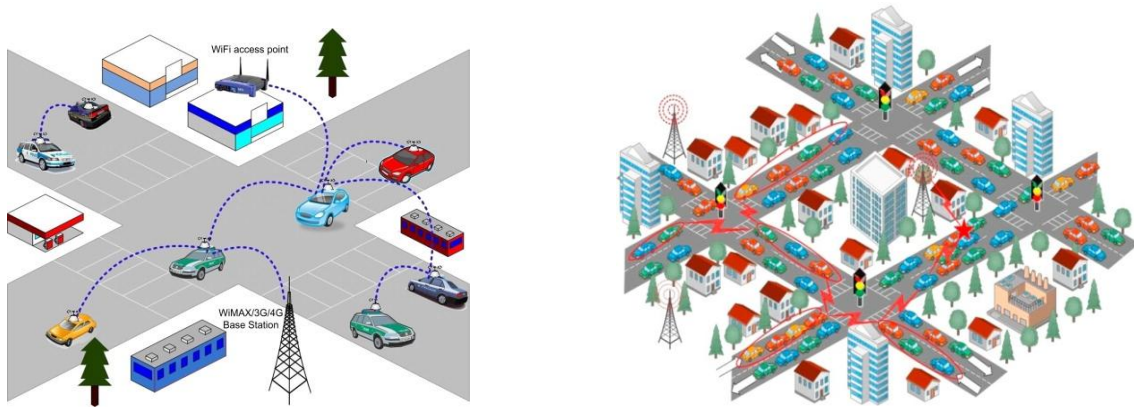
$$PL = \frac{\text{Nbr paquet transmis} - \text{Nbr paquet reçus}}{\text{Nbr paquet transmis}} \quad \text{Eq (3.5)}$$

Le taux de perte de paquets (PLR) est mesuré à travers l'application UDP en faisant varier la distance entre les nœuds et les taux de transmission. Notez que le PLR donne une idée sur la portée de transmission, si le PLR est plus proche de 0, les nœuds communicants sont dans la même plage de transmission, sinon hors de portée.

3.3.6. Scenarios utilisés

1^{er} scénario

La variation des paramètres de performances selon la variation du nombre des nœuds (véhicules 40, 50,60).



(a) Faible densité

(b) haute densité

Figure 3.3. Densité des véhicules**2^{eme} scénario**

La variation des paramètres de performances suivant la variation de la vitesse des nœuds (20, 25, 30,35 km/h).

3.4. Déroulement de la simulation**3.4.1. Matériels et outils**

Les deux simulateurs utilisés dans cette étude, NS3 et SUMO, sont pris en charge par le système d'exploitation Linux UBONTU 18.04. Pour cette étude, un ordinateur a été utilisé avec les configurations suivantes: Processeur Intel® Core™ i3 à 2,33 GHz x 2/64 bits, 4 Go de RAM, 500 Go de stockage interne et système d'exploitation UBONTU 18.04.

3.4.2. Scénario de simulation

Pour la simulation d'un réseau VANET, il est nécessaire de créer un scénario qui représente mieux l'environnement véhiculaire, où le déplacement des nœuds (véhicules) est restreint par les rues et la topologie du réseau a tendance à changer rapidement en raison de la vitesse de déplacement de nœuds.

De cette manière, le logiciel de mobilité SUMO a été utilisé, ce qui permet la création d'un scénario véhiculaire simplifié. On a introduit le modèle de mobilité urbain du village SIDI HAMOU pour la configuration des paramètres suivants: durée de la simulation, nombre des nœuds, largeur et hauteur de la zone de simulation. Les paramètres utilisés dans les simulations sont indiqués dans le tableau 3.1.

Paramètres de simulation	valeur
Nombre des nœuds	40, 50,60
Vitesse (m/s)	20, 25, 30,35
Temps de simulation	60 s
Zone de simulation m ²	1000 m ,2000 m

Tableau 3. 1. Paramètres de simulation de mobilité.

3.4.3. Étapes de la simulation

3.4.3.1. Étape 1 : création du Map

Pour cette étape, nous importons une partie de la carte géographique d'une zone urbaine existante dans notre ville «Aindefla» à partir de site Open Street Map [25], le simulateur SUMO collecte des données des routes, voies ferrées, les rivières, les forêts, les bâtiments, etc.

Sur le terminal dans le répertoire [sumo/Tools] et avec la commande suivante: « python osmWebWizard », nous obtenons la figure suivante :

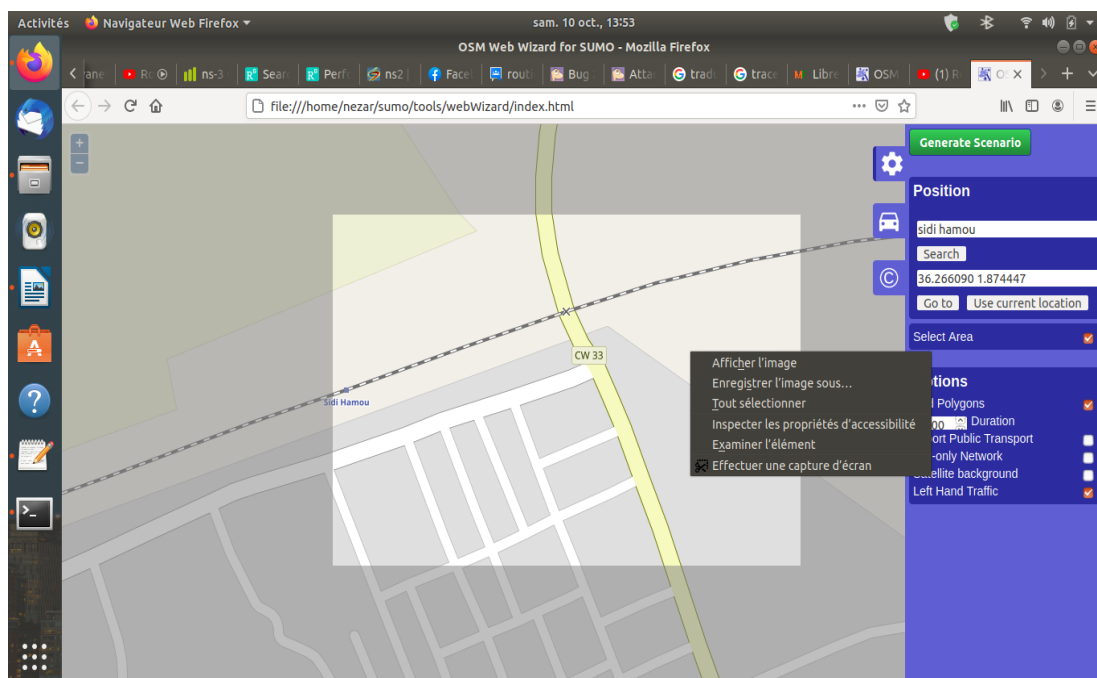


Figure 3. 4. vue d'une partie du village Sidi hamou sur OSM avant l'importer sur SUMO

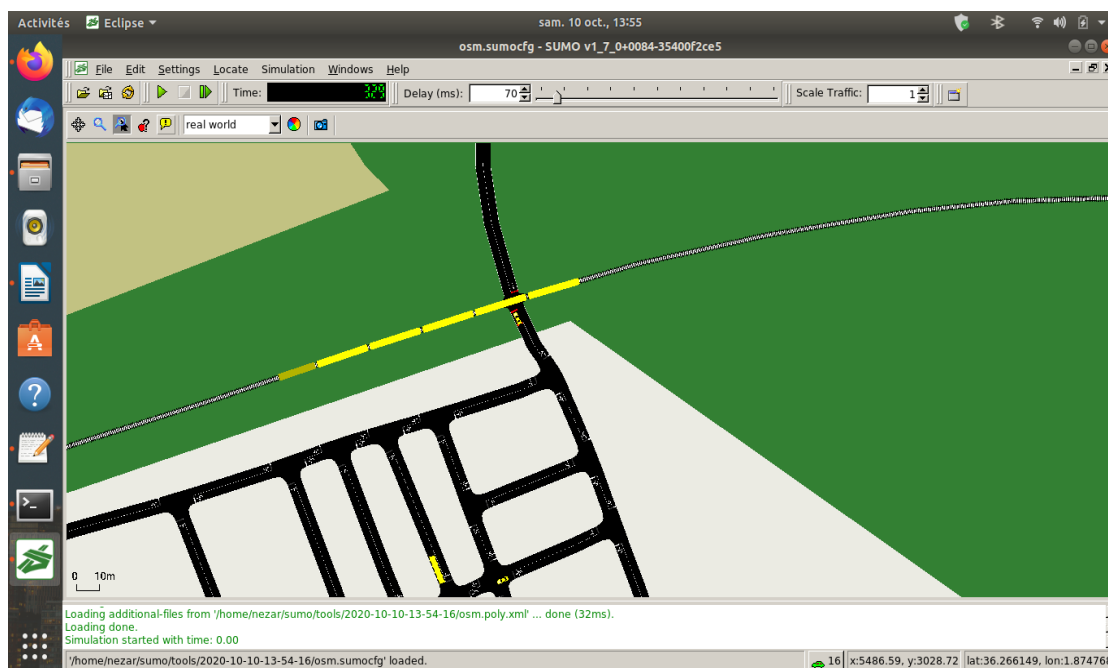


Figure 3. 5. Zone d'étude après importation

Le processus de simulation d'un VANET se fait en plusieurs étapes, illustrées à la figure 3.8. 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 NS3.

A la fin de la simulation avec NS-3, deux fichiers de sortie seront enregistrés: 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 NS3. Ce processus est illustré dans la figure suivante 3.8 et détaillé à l'annexe.

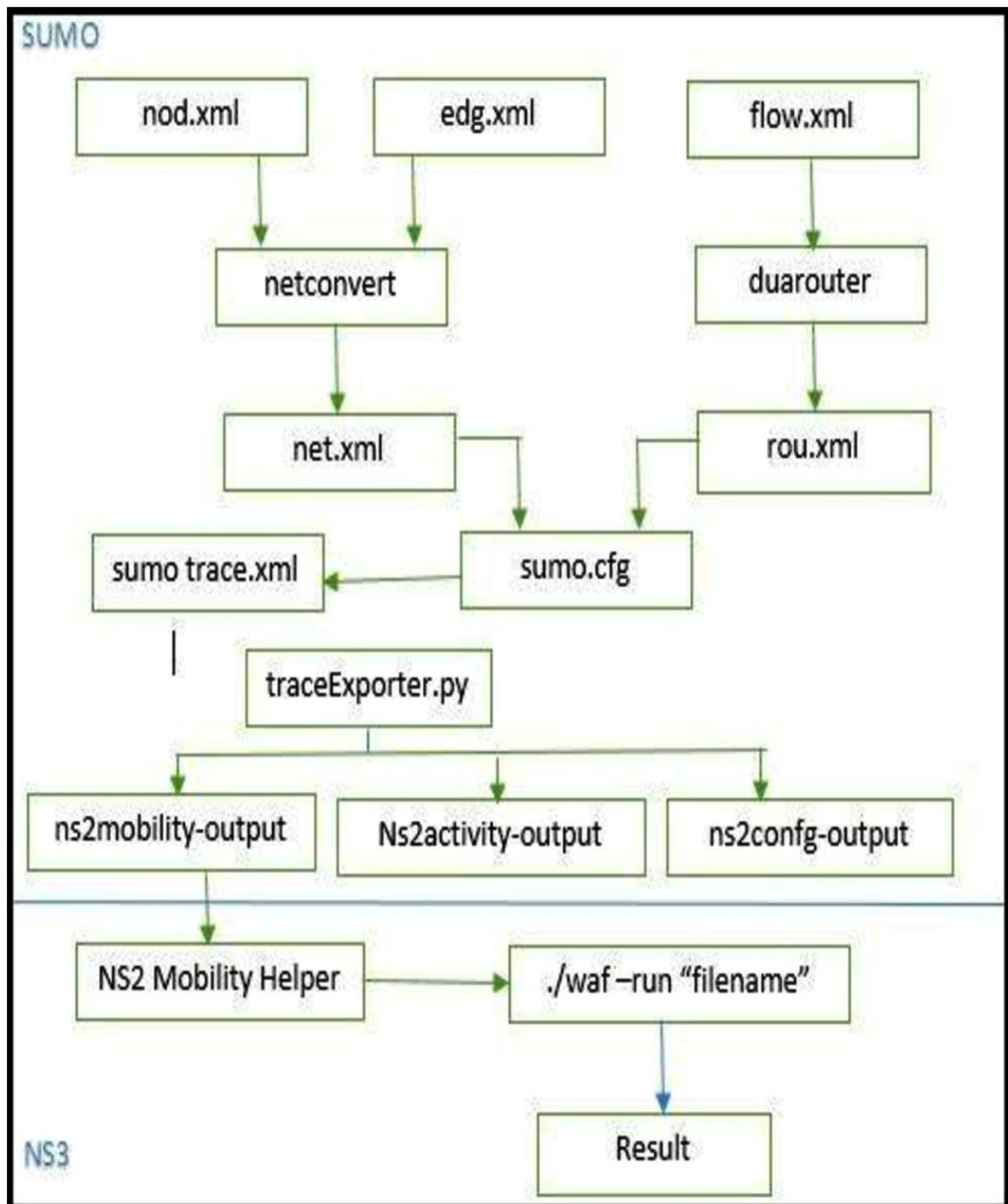


Figure 3. 6. Etapes de la simulation

Plusieurs étapes ont été faite avec SUMO afin d'obtenir le fichier « mobility.tcl » comme le montre la figure 3.9 (voir annexe b) :

Commande de simulation du script vanet-routing-compare.cc				
paramètre	explication	valeurs	Valeur par défaut	Commande sur terminal
Total time	temps de simulation	<double>	300.01 (s)	./waf --run "scratch/vanet-routing-compare totaltime=100"
verbose	sortie	0=quiet; 1=verbose	0	./waf --run "scratch/vanet-routing-compare --verbose=1"
flowmon	flow mon statistiques	0=off; 1=on	0	./waf --run "scratch/vanet-routing-compare --flowmon=1"
routing_tables	Vider les tables de routage	0=no; 1=yes	0	./waf --run "scratch/vanet-routing-compare --routing_tables=1"
ascii_trace	Vider ASCII trace	0=no; 1=yes	0	./waf --run "scratch/vanet-routing-compare --ascii_trace=1"
pcap	Créer fichier pcap	0=no; 1=yes	0	./waf --run "scratch/vanet-routing-compare --pcap=1"
CSV file Name	Nom du fichier csv	/	Vanet-routing-compare.csv	./waf --run "scratch/vanet-routing-compare --CSVfileName=AODV.csv"
scenario	Scenari prédéfini	1=Realistic 2=51 nodes, RWP, 100s; 3=101 nodes, RWP, 10s	1	./waf --run "scratch/vanet-routing-compare --scenario=2"
mobilité	model	1=trace; 2=RWP	/	./waf --run "scratch/vanet-routing-compare --mobility=2"
nodes	Nombre des nœuds	/	156	./waf --run "scratch/vanet-routing-compare --nodes=51"
Trace File	Ns2 mouvement trace file (Trace playback sauf, i.e. mobilité=1)	ns-2 mobility trace file	./scratch/ct- unterstrass- 1day.filt.5.ad j.mov	./waf --run "scratch/vanet-routing-compare --traceFile=trace.out"
speed	vitesse (m/s) (RWP only, i.e. mobilité=2)	<int>	20	./waf --run "scratch/vanet-routing-compare --speed=15"
txp	Energie de transmission (<int>	7.5	./waf --run "scratch/vanet-routing-compare --txp=20"
802-11 Mode	802.11 mode (i.e. b/p)	1=802.11p; 2=802.11b	/	./waf --run "scratch/vanet-routing-compare --80211Mode=2"
Protocol	Protocol de routage	1=OLSR; 2=AODV; 3=DSDV; 4=DSR	2=AODV	./waf --run "scratch/vanet-routing-compare --protocol=4"

Tableau 3. 2. Paramètres de simulation générale

Le script vanet-routing-compare.cc envoie des messages de sécurité de base (BSM) de 200 octets 10 fois par seconde à un débit de 6 Mbps vers tous les nœuds du réseau, via le canal de contrôle (CH) 10 MHz avec accès continu.

De plus, tous les nœuds essaient en permanence d'acheminer les paquets de 64 octets à un taux d'application de 2048 bps pour les autres nœuds sélectionnés comme nœuds récepteurs.

Par défaut, si nous effectuons la simulation de script sans modifier aucun paramètre, le scénario 1 sera exécuté, qui a 10 secondes de simulation, avec un total de 40 nœuds (40 véhicules), avec un déplacement selon le modèle de mobilité aléatoire Waypoint et avec une vitesse de 20 m/s et sans temps de pause dans une région de 300 x 1500 mètres. Avec le protocole de routage AODV en standard et la propagation à deux rayons. La puissance d'émission est réglée sur 20 dB et la plage transmission de 145 mètres.

Cependant, le modèle de mobilité *Random Waypoint* n'est pas adapté à un environnement véhiculaire, puisque les nœuds se déplacent au hasard sans aucune restriction. Afin de faire une bonne simulation pour les deux scénarios avec l'outil SUMO on doit modifier le script, les changements sont présentés à l'annexe.

La configuration de la mobilité est entièrement réalisée par le simulateur SUMO où des scénarios ont été créés avec un model urbain, tandis que la partie de création de trafic de données et la structuration du réseau est entièrement effectuée dans le simulateur NS-3. Les paramètres réseau sont indiqués dans le tableau 3.4.

La syntaxe de ligne de commande pour exécuter les simulations de notre programme est indiquée ci-dessous:



```
nezar@dell:~/repos/ns-allinone-3.29/ns-3.29$/waf --run "scratch/simulation
--totaltime=100
--CSVfileName2=<output>.csv
--scenario=3
--mobility=1
--nodes=<qt.Nós (30, 40, 50)>
--traceFile=scratch/<cenários>.ns_movements
--logFile=ns2-mob.log
--txp=20
--lossModel=3
--80211Mode=1
--fading=1
--protocol=<OLSR (1), AODV (2), DSDV (3)>
--sinks=10"
```

Figure 3. 8. Commande utilisé pour la simulation

Pour la simulation du réseau à travers NS3, nous avons utilisé les paramètres illustrés au tableau suivant :

paramètre	valeur
simulateur	Ns3.29
Nombre des nœuds	40, 50,60
Temps de simulation(s)	60
Distance de transmission(m)	250
Protocol MAC	802.11p
Protocole de routage	OLSR-DSDV- AODV-DSR
Vitesse de transmission	6 Mbps
Taille du paquet	200 byte
Model de mobilité	urbain

Tableau 3.3 : Paramètres utilisés dans la simulation du réseau de véhicules

3.4.3.3. Exécution du programme

Plusieurs modifications ont été apportées au script `vanet-routing-compare` (voir annexe) afin d'évaluer les protocoles selon notre besoin, et avec les commandes présentées sur la figure 3.10, nous avons pu générer tous les tests pour notre travail. Pour tous les protocoles il faut répéter la simulation 5 fois pour chacune.

Nous avons stocké les résultats des différentes simulations dans des fichiers d'extension `csv`, où les informations sur le taux de livraison des colis, le débit moyen et la charge de routage normalisée ainsi que le taux de perte des paquets.

Pour collecter des données sur le temps de retard, nous avons utilisé les outils `Flow Monitor` et `trace métrique` du simulateur NS3 lui-même, qui génère des fichiers de sortie au format `xml`.

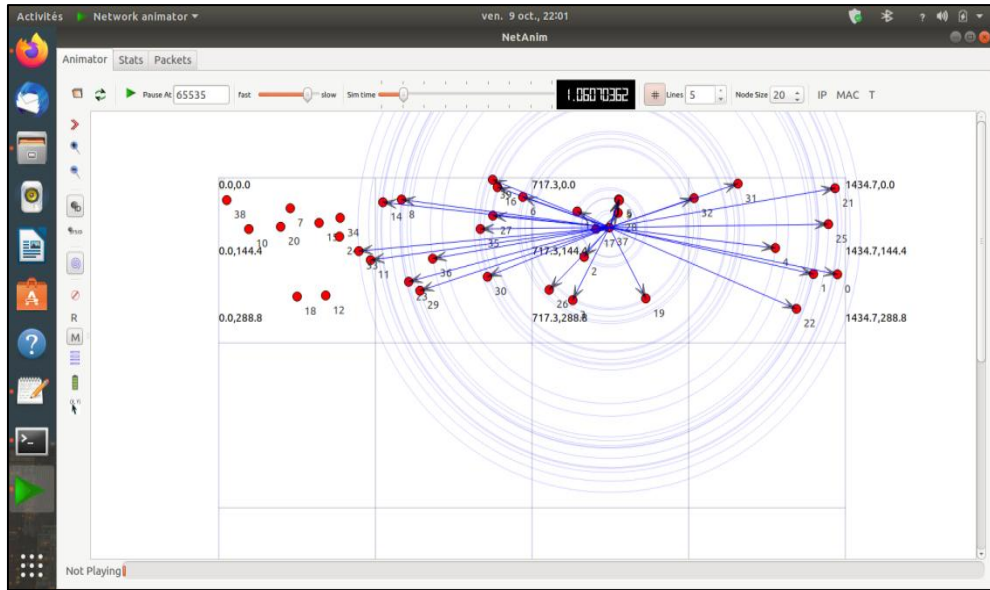


Figure 3. 9. Visualisation de 40 nœuds avec Net Anim.

Les analyses sur ces données sont présentées dans la section suivante.

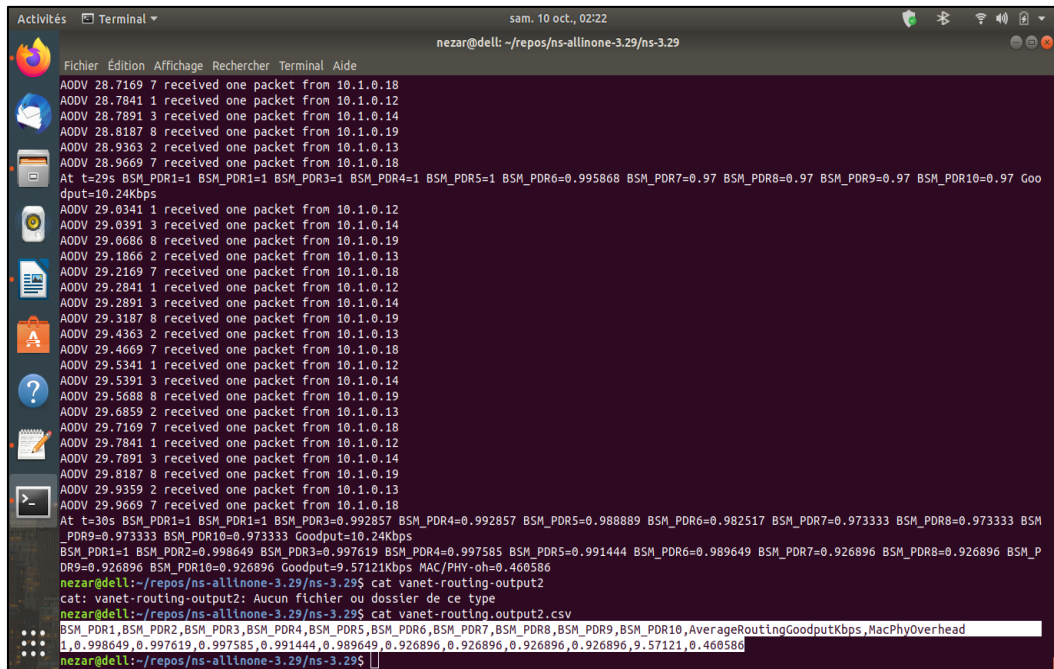


Figure 3. 10. Résultats de simulation du Protocol AODV pour le scenario 1

Remarques

Il ya des résultats collectés directement comme montre la figure 3.12, et les autres sont obtenus avec des calculs selon les équations mentionnées sur la section 3.3. Les analyses des données sont présentées dans la section 3.5.

3.2. Analyses des résultats de simulation

3.5.1. Scenario 1 : Interprétation des résultats selon la variation de nombre des nœuds

3.5.1.1. PDR vs Densité

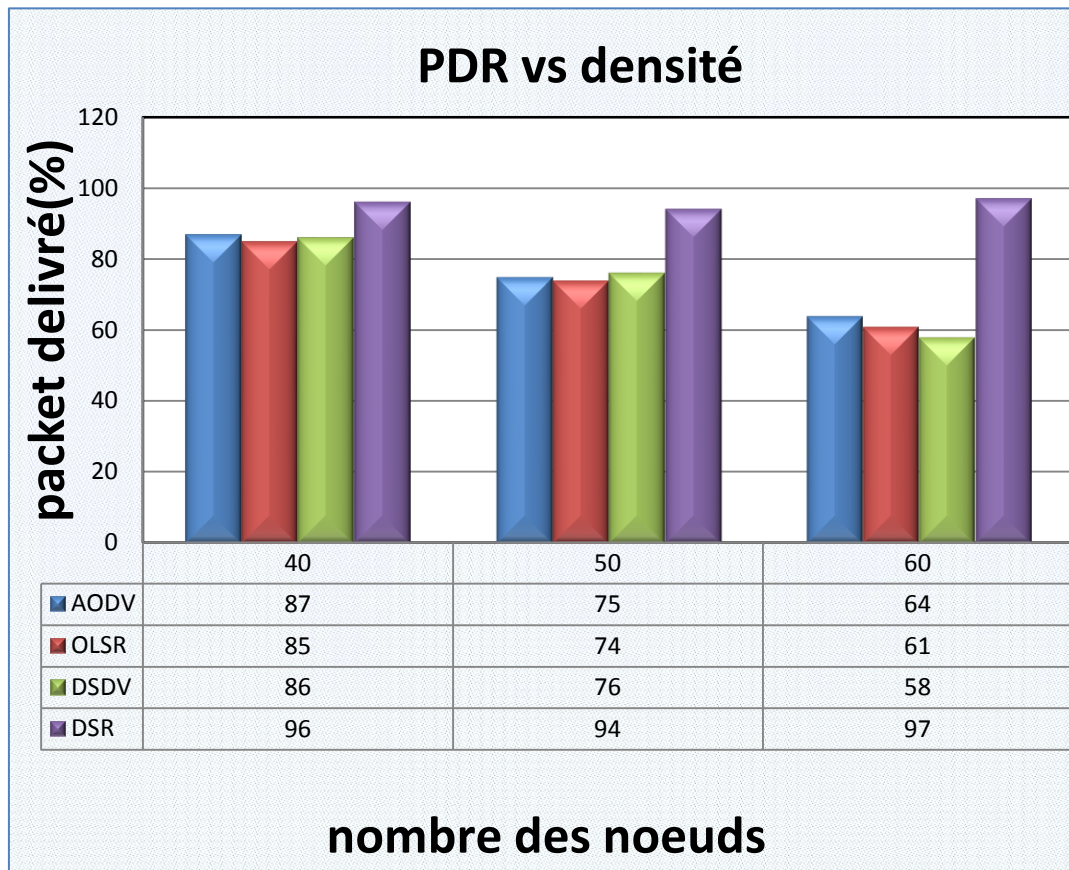


Figure 3. 11. PDR vs densité

En évaluant les résultats présentés à la figure 3.13, nous notons que la densité des véhicules ait une influence directe sur le taux de livraison des paquets. Puisque le nombre de nœuds augmente et le PDR diminue, ça c'est due au fait que lorsqu'une route est invalidée un message RERR est envoyé qui réduit le taux d'envoi jusqu'à la découverte d'un nouvel itinéraire.

Le protocole DSDV a montré de bons résultats en aboutissant à un faible nombre de nœuds, mais dans des réseaux à densité comme c'est le cas pour le troisième scénario avec un total de 60 nœuds sa performance est réduite. Cela parce que c'est un environnement à haute densité et le nombre de liens est coupé en raison de la mobilité.

Le protocole AODV a montré des taux élevés de livraison des paquets, et ceci ne se traduit pas par de bonnes performances de communication car le protocole AODV envoie des messages répétitifs (messages de découverte d'itinéraire).

Le protocole de routage OLSR a également subi une baisse de performance dans les scénarios de haute densité, mais a maintenu un bon taux de livraison des paquets, qui sont utiles et non répétitifs, comme le cas avec AODV.

En comparant les protocoles proactifs (OLSR et DSDV) entre eux, nous constatons que le protocole OLSR présente un meilleur comportement car il essaie d'éliminer les routes obsolètes en utilisant le message hello, donc dans des conditions où le nombre de routes est élevé, OLSR délivre plus de paquets et utilise donc efficacement la bande passante et par conséquent le PDR est plus grand que pour DSDV.

Par la suite, nous constatons que le Protocole unicast DSR est le meilleur, puisque celui qui présente des valeurs très élevées de taux de livraison des paquets.

3.5.1.2. Délai vs densité

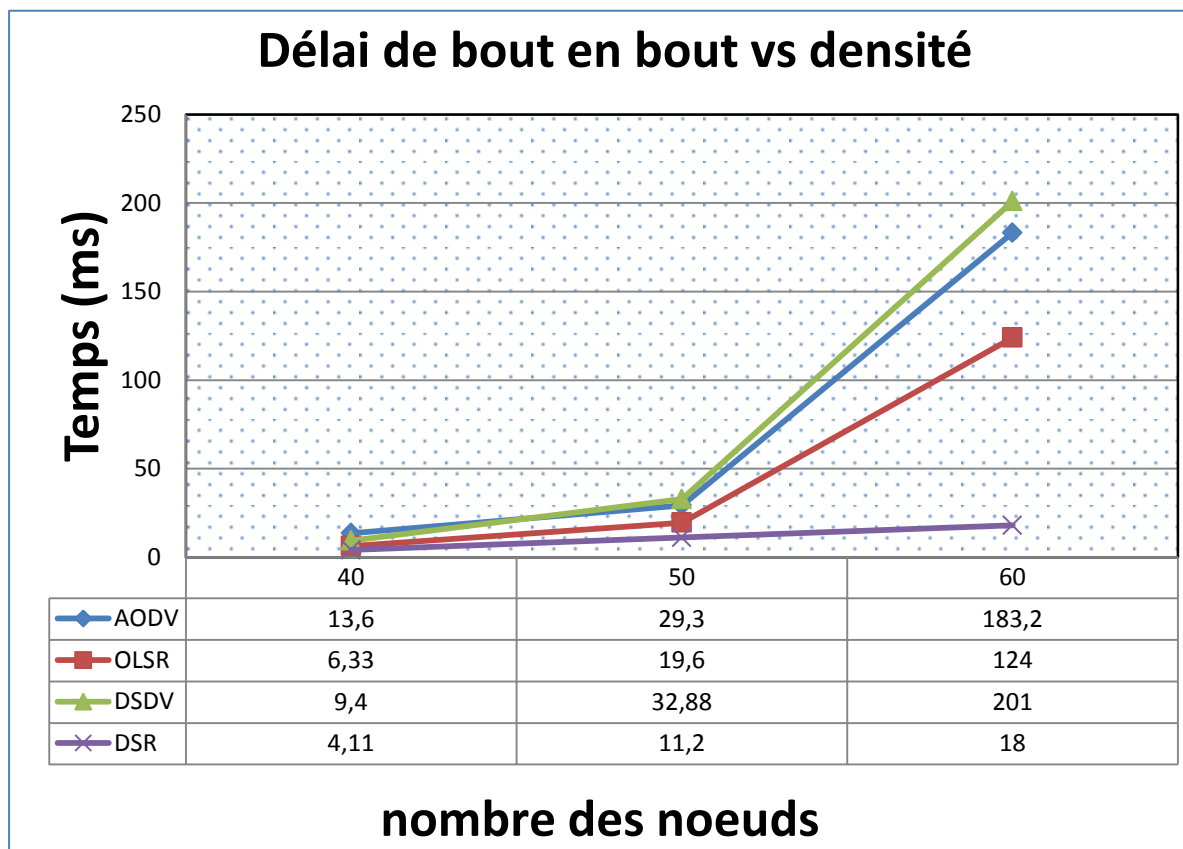


Figure 3. 12. Délai vs densité

Le taux de retard mis par chaque protocole, est représenté dans la figure 3.14. Ce paramètre fait référence au temps de retard de transmission d'un paquet de données de la source à la destination, et il est extrêmement important car les principales applications du réseau VANET sont axées sur la sécurité et la demande de faible temps de réponse.

En regardant la figure 3.14, nous avons remarqué le temps de retards augmente à mesure que la densité du réseau augmente. Dans le premier scénario avec un montant de 40 véhicules, nous notons que le protocole avec le plus long retard est AODV, cela est dû à l'inondation causée sur le réseau par le grand volume de messages de contrôle. Cependant, plus la densité de véhicules augmente plus le retard donné par le protocole DSDV est important.

Et dans les trois scénarios, le protocole OLSR a montré de bons résultats par rapport a AODV et DSDV. Nous pouvons aussi vérifier que DSR est le meilleur puisque celui qui présente des valeurs très faibles.

3.5.1.3. NRL vs densité

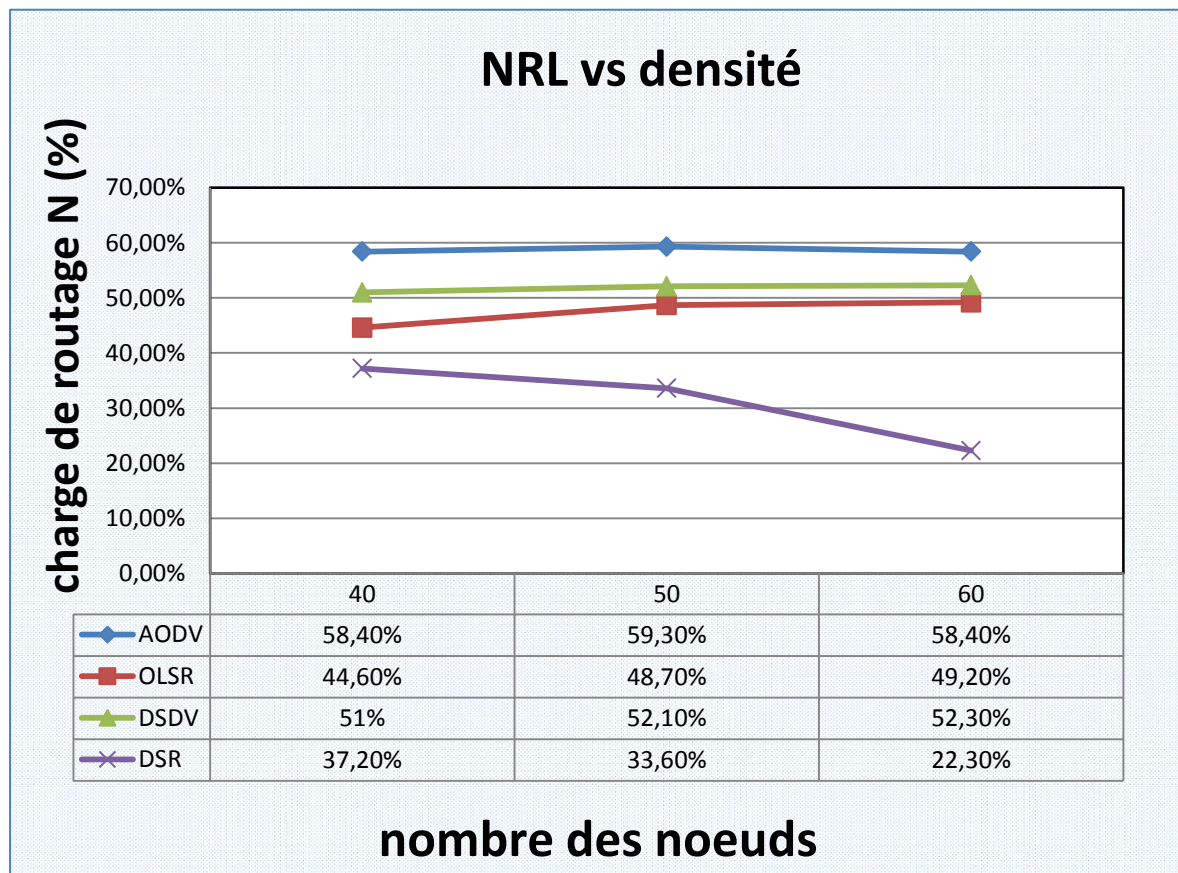


Figure 3. 13. NRL vs densité

La figure 3.15, représente les résultats de charge de routage normalisée obtenus pour chacun des protocoles, en raison de l'augmentation de la densité du réseau. Nous notons que le protocole qui génère la plus grande valeur de charge de routage est AODV en raison des procédures de découverte et de la maintenance des itinéraires.

Le protocole DSDV a également des taux de NRL élevés en raison de la maintenance de tables de routage. Pour le protocole OLSR, la réduction de charge de routage est due au plus petit nombre de retransmissions. En comparant ces protocoles réactifs, on peut voir que DSR est le meilleur.

3.5.1.4. Débit vs densité

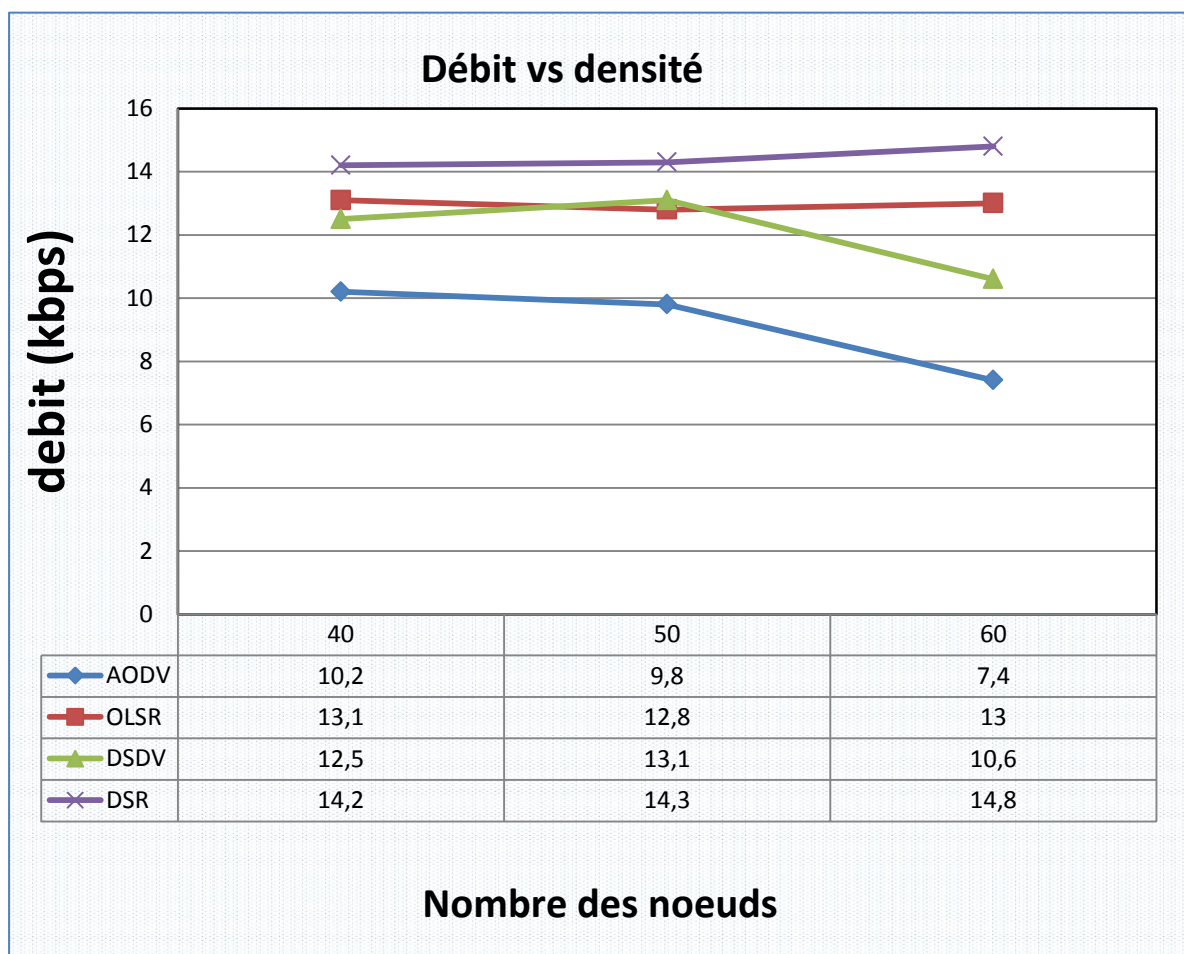


Figure 3. 14. Débit vs densité

La figure 3.16, représente la variation du débit en fonction de la quantité des véhicules. Le débit se traduit par la quantification des données uniquement reçus, et les données de retransmission seront rejetées.

En évaluant les résultats présentés dans le graphique de la figure 3.16, nous avons remarqué que chaque protocole a présenté des réponses différentes aux taux de débit en fonction du nombre de véhicules.

Le protocole DSDV a bien fonctionné jusqu'au scénario de densité moyenne avec 50 nœuds qui a eu après des performances réduites dans les topologies à haute densité, en raison du faible taux de livraison des paquets.

Le protocole AODV est celui qui a les valeurs de débit les plus bas, malgré les taux élevés du PDR présenté précédemment. En effet, le nombre de paquets livrés ne traduit pas les informations utiles, car AODV est un protocole réactif et envoie un grand nombre de messages de découverte d'itinéraire, ce qui finit par générer un nombre de collisions affectant directement le taux de débit.

Aussi le protocole OLSR a montré de meilleurs résultats par rapport à AODV et DSDV en raison de l'augmentation de la densité du réseau, en raison de sa bonne caractéristique de gestion grâce à l'utilisation de technique MPR, qui évite d'inonder le réseau d'informations répétées.

Nous pouvant voir que le protocole ayant les meilleures performances est le DSR.

3.5.1.5. Paquets perdus vs densité

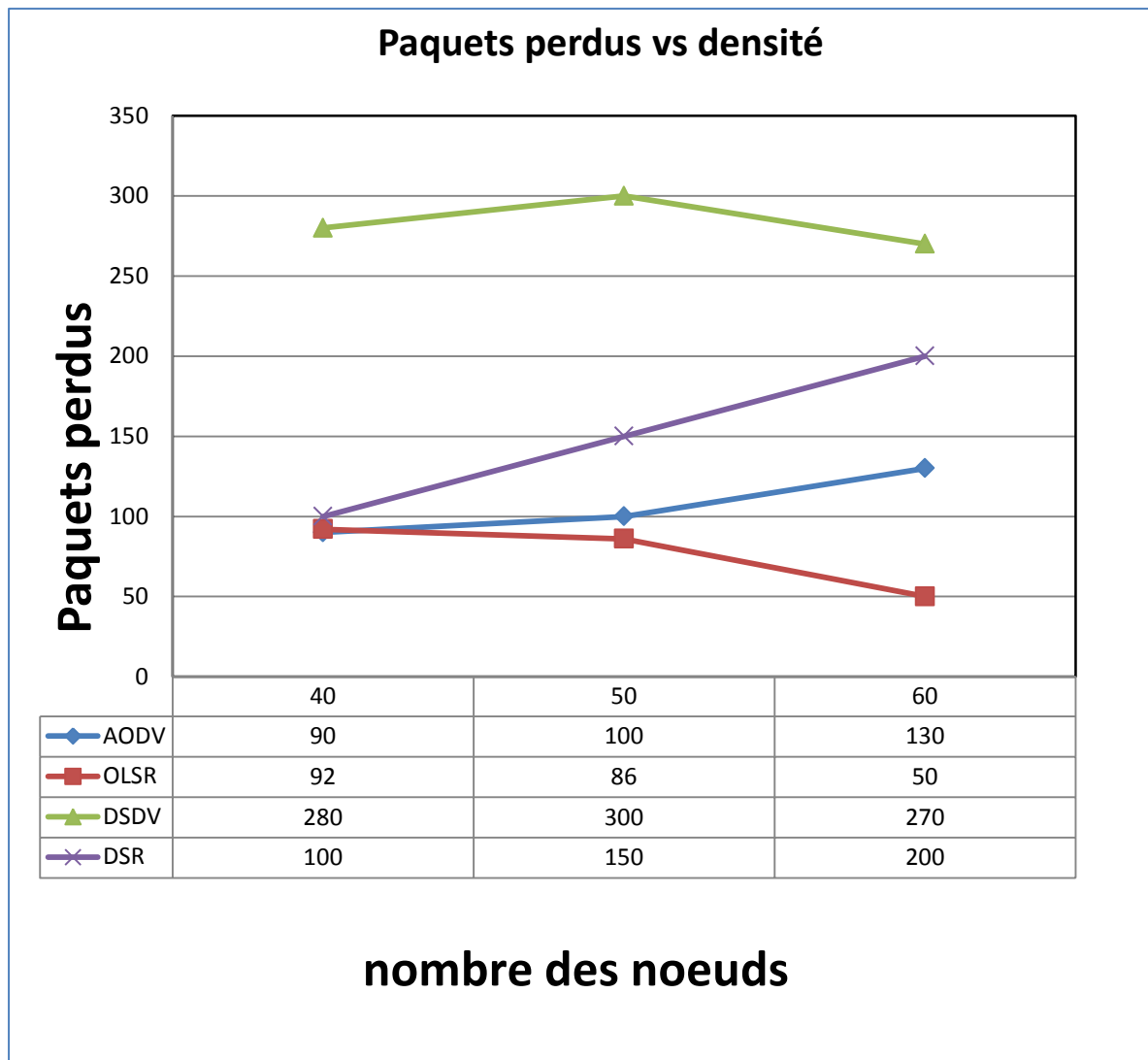


Figure 3. 15. PL vs densité

Dans la figure3.17, nous constatons que pour un certain nombre de nœuds, le protocole qui présente le bon comportement concernant le nombre de paquets perdus est OLSR (proactif). Par rapport à un protocole de même catégorie qu’OLSR, le protocole DSDV, il y a une différence importante, car OLSR supprime toutes les routes de ses tables de routage obsolète de sorte que la plupart des paquets perdus dans OLSR sont dus à l’expiration de temps.

Dans le cas de protocoles proactifs, par exemple AODV et DSR, et sous la condition d’un numéro des nœuds bas le protocole AODV présente un meilleur comportement que le DSR.

3.5.2. Scenario 2 : Interprétation des résultats selon la variation de vitesse des nœuds.

Afin de faire une analyse plus approfondie sur les protocoles de routage, en évaluant ces derniers en fonction de la vitesse de déplacement de nœuds (véhicules). La vitesse de déplacement ait une influence directe sur certains paramètres, en raison de la de coupures liens de communication. Pour cela, le scénario a une densité constante (30 nœuds) et les vitesses de déplacement testées changent, une vitesse de 15 m/s (54 km/h) et une autre vitesse de 35 m/s (126 km / h).

3.5.2.1. PDR vs vitesse

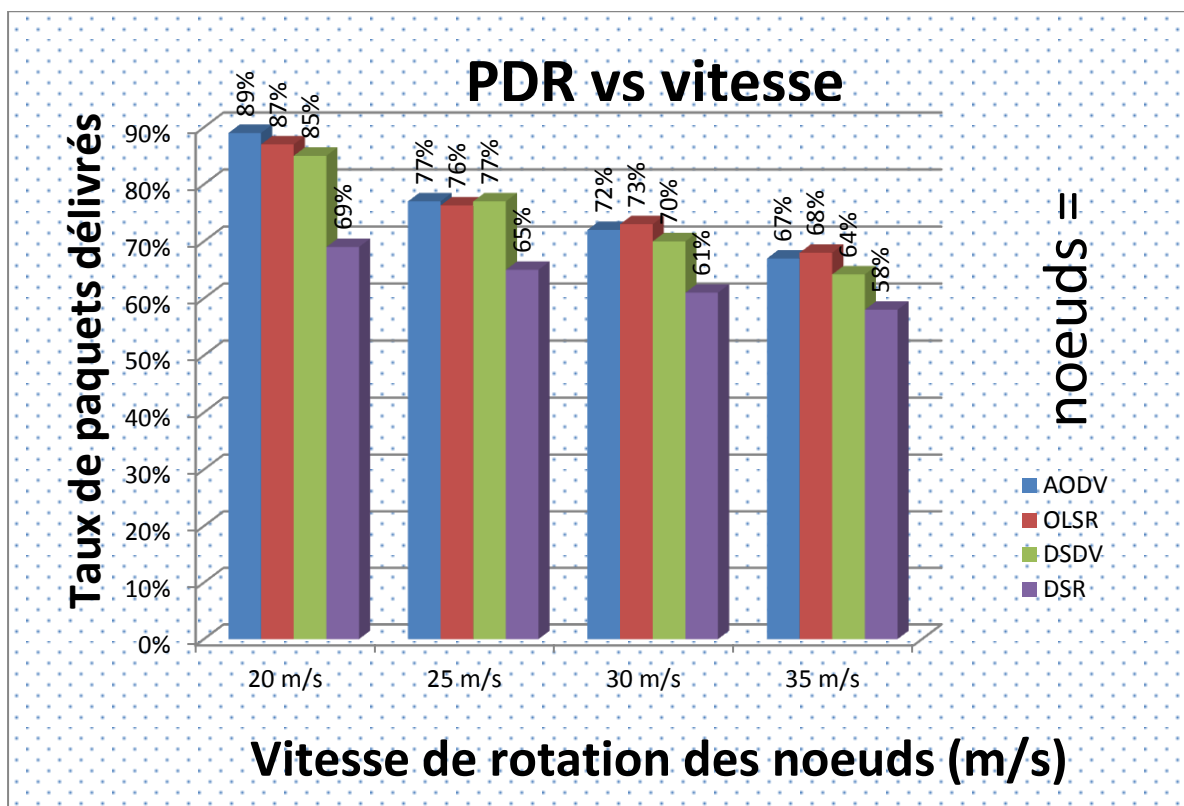


Figure 3. 16. PDR vs vitesse

Le protocole qui a montré le meilleur support pour augmenter la vitesse était OLSR maintenant un taux de livraison plus élevé que les autres. Le protocole AODV est meilleur dans le scénario avec vitesse 20 m/s.

3.5.2.2. Délai vs vitesse

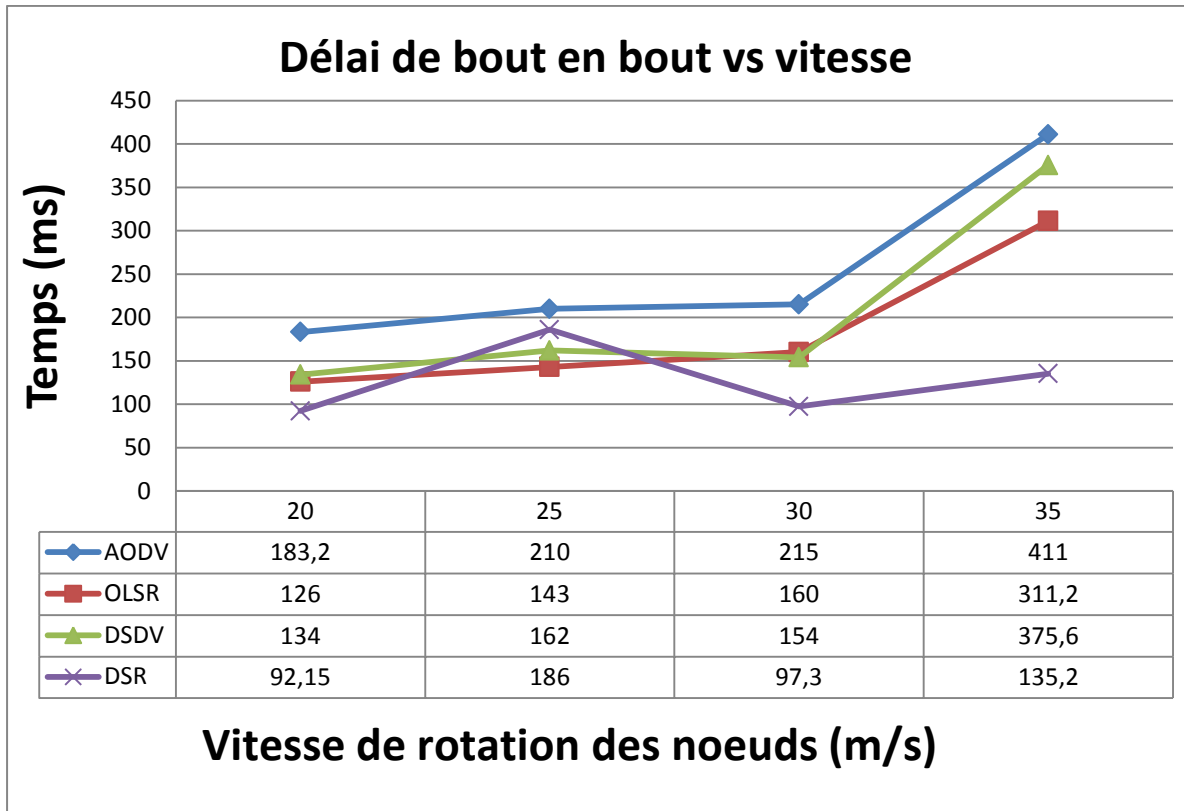


Figure 3. 17. Temps vs vitesse

La figure 3.19, montre le délai de livraison des paquets en fonction de la vitesse de déplacement des nœuds. Nous remarquons que pour les faibles vitesses, le retard dans la livraison des paquets est similaire pour tous les protocoles. Nous pouvons voir que, selon l’augmentation de la vitesse, le temps de retard augmente, cela est dû à la rupture des liens de communication en raison de la mobilité.

Cependant, le protocole OLSR était le protocole avec la meilleure réponse lorsque la vitesse de déplacement augmentait. Ceci est directement lié au bon taux de livraison obtenu par ce protocole.

Par conséquent, après avoir analysé les différents paramètres sur les protocoles de routage, soit en fonction de la densité du réseau ou de la vitesse de déplacement, le protocole OLSR s’est avéré supérieur aux autres lorsqu’il est appliqué dans un réseau VANET, avec une bonne fonctionnalités d’assistance à la fois pour augmenter la densité et la vitesse du réseau avec de bons taux de transmission et un temps de retard plus court par rapport aux autres protocoles étudiés. Ces facteurs étant extrêmement importants pour les applications des réseaux VANET.

3.5.2.3. NRL vs vitesse

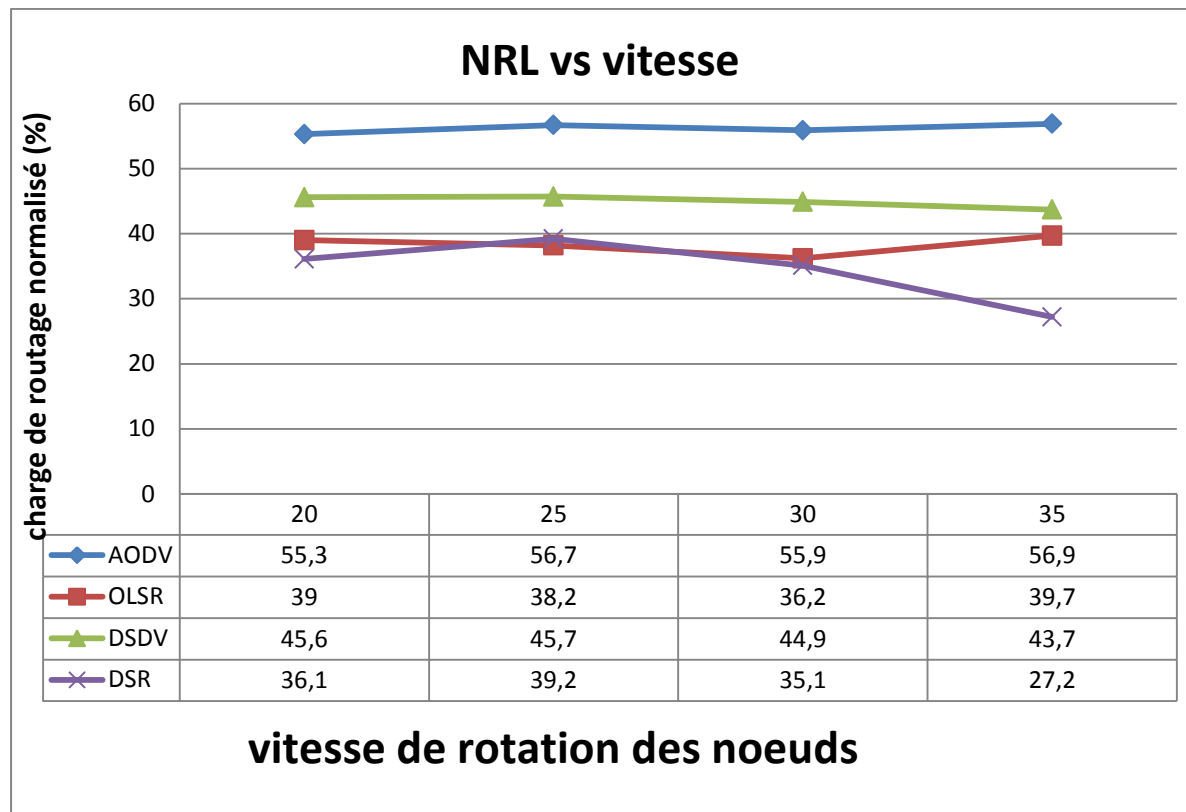


Figure 3. 18. NRL vs vitesse

La figure 3.20, montre les résultats du NRL en fonction de la vitesse de déplacement de nœuds (véhicules).

- Le protocole AODV a une fonction réactive qui peut envoyer un grand nombre de messages de contrôle d’itinéraire, et a un taux de surcharge élevé.
- Étant suivi par le protocole DSDV dont la charge de routage est générée en raison de la maintenance du routage. Et tout comme dans les simulations impliquant une densité de réseau élevée.
- Le Protocole OLSR a donné de faibles taux de charge de routage grâce à l’utilisation de la technique des MPR.
- Le protocole DSR est le meilleur, avec un fonctionnement impeccable.

3.5.2.4. Débit vs vitesse

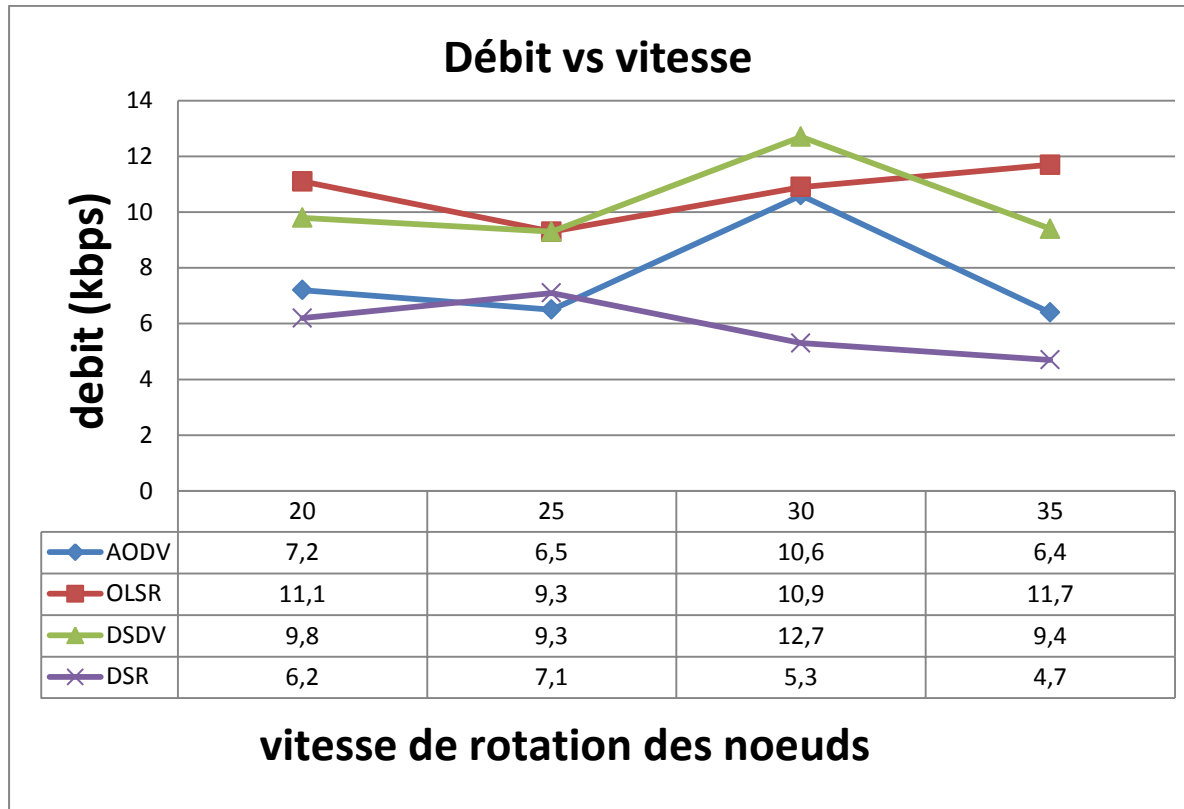


Figure 3. 19. Débit vs vitesse

La figure 3.21, montre les résultats de débit en fonction de la vitesse de déplacement.

Le protocole le plus affecté est l'AODV ayant un taux de transmission inférieur aux autres. Le protocole DSDV a montré de bons résultats en général lorsqu'il était soumis à des vitesses élevées.

Le protocole OLSR, était celui qui présentait les variations les plus légères et le meilleur comportement lorsqu'il est traité dans un scénario à forte mobilité. Cependant, le scénario avec une vitesse de 35 m/s, qui a atteint le taux de transmission le plus élevé. La justification de l'amélioration du débit métrique à la vitesse est due à la portée de transmission, ceci puisque dans les vitesses inférieures, les nœuds sont éloignés (le scénario simulé a une superficie de 1 km²).

Encore en fonction de l'augmentation de la vitesse à des valeurs supérieures à 30 m/s, l'indice de débit se diminue à nouveau du fait qu'à partir de cette valeur la mobilité du réseau est très élevée, ce qui entraîne une instabilité des liaisons de communication.

3.5.2.5. PL vs vitesse

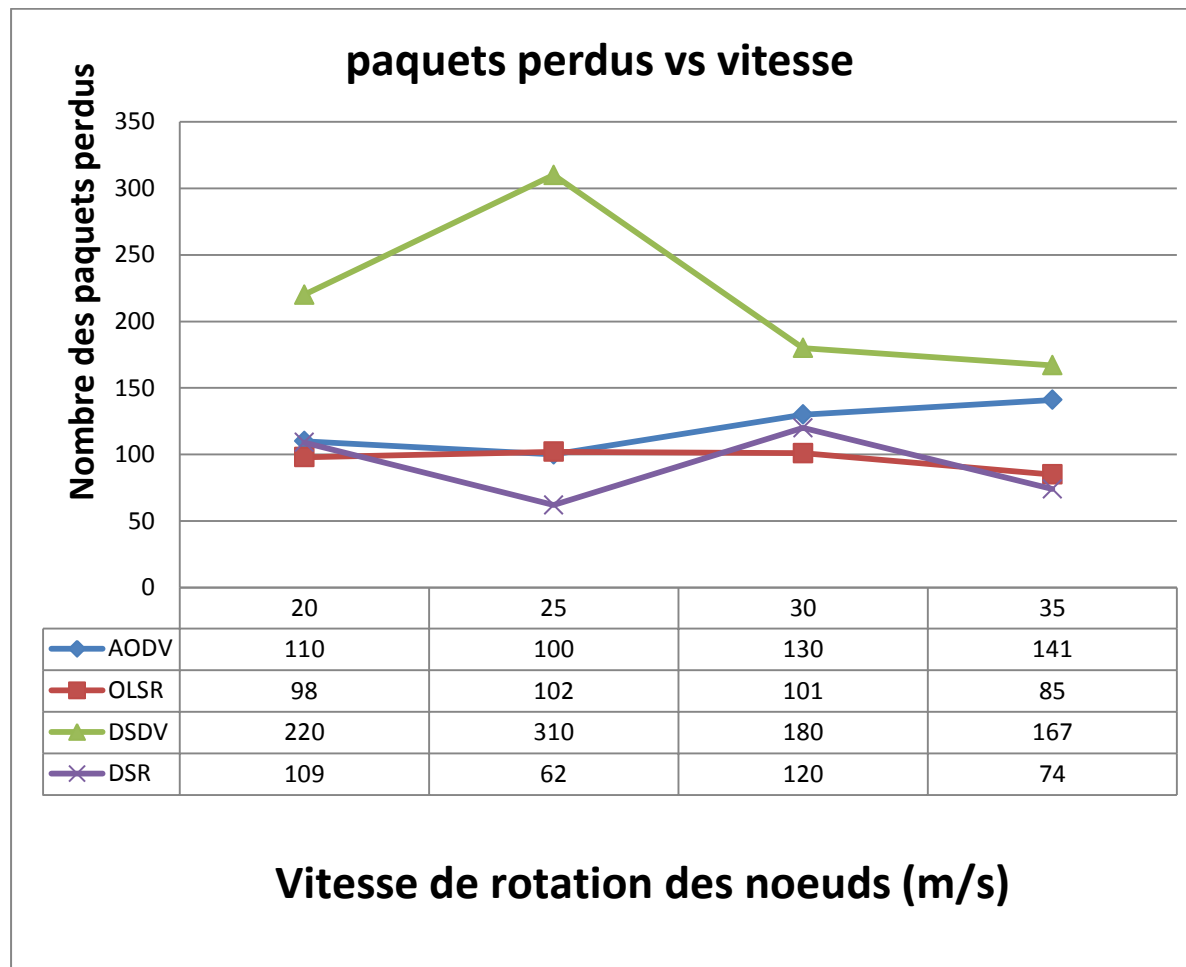


Figure 3. 20. PLR vs vitesse

La figure 3.20, représente la variation de la perte des paquets en fonction de la variation de la vitesse, ce paramètre est plus important afin d'évaluer la performance des protocoles de routage des VANETs.

En premier lieu, nous constatons que le protocole OLSR présente des meilleurs résultats par rapport aux DSDV et AODV.

Aussi, DSR fonctionne bien dans le deuxième scenario.

3.6. Conclusion

Dans ce chapitre nous avons introduit les étapes des simulations et l'évaluation des performances des quatre protocoles choisis (AODV, OLSR, DSR, DSDV).

Après avoir analysé les différents paramètres des protocoles de routage, soit en fonction de la densité du réseau ou de la vitesse de déplacement, le protocole OLSR s'est avéré le meilleur par rapport aux autres lorsqu'il est appliqué dans un réseau VANET, avec une bonne fonctionnalité d'assistance à la fois pour augmenter la densité et la vitesse du réseau avec de bons taux de transmission et un temps de retard plus court par rapport aux autres protocoles étudiés. Ces facteurs étant extrêmement importants pour les applications des réseaux VANET. Et il a également été démontré que ce protocole avait de faibles taux de charge de routage, cela démontre donc qu'il s'agit d'un protocole évolutif et prend en charge l'expansion du réseau.

CONCLUSION GENERALE

Conclusion générale et perspectives

L'étude des réseaux des véhicules est un domaine très large, avec un potentiel de recherche extrêmement large. Les règles qui régissent sur ces réseaux sont en état de maturation, en constante rénovation et amélioration, toutes ces études visant à analyser le trafic de données sur ces réseaux pour s'assurer que les applications qu'ils fournissent sont satisfaites.

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, nous parlons 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, 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 NS3 et le simulateur de trafic routier SUMO que nous avons utilisé pour évaluer les protocoles choisis (AODV, OLSR, DSR et DSDV).

l'évaluation réalisée dans cette étude sur ces protocoles de routage a permis d'examiner les performances de ces protocoles en termes des métriques de qualité de service, taux de livraison des paquets (PDR), débit, charge de routage normalisée et le temps de retard ainsi que le taux de perte des paquets.

Parmi les protocoles évalués, OLSR était le meilleur, en raison de sa bonne caractéristique de gestion, à travers la technique MPR qui permet d'éviter l'inondation du réseau de données répétées, sans occuper toute la bande passante de transmission et

permettant de supporter une bonne évolutivité du réseau. OLSR est celui qui a donné des meilleurs résultats pour être utilisé dans un réseau de communication véhiculaire.

Cependant, il convient de mentionner que ces protocoles ont été conçus pour les réseaux MANET, qui présentent une faible mobilité, ce qui fait que le réseau n'a pas beaucoup de changements dans la topologie et un plus petit nombre de messages de contrôle du Web. Contrairement aux réseaux VANET, qui malgré leur inclusion dans MANET présentent des caractéristiques spécifiques en terme de la mobilité.

Durant ce travail, nous avons constaté que le domaine de recherche dans les réseaux véhiculaire est vaste et qu'il a un long chemin à parcourir.

Plusieurs groupes scientifiques s'intéressent à ce type de réseaux qui promet beaucoup pour l'avenir. En effet, nous avons eu l'occasion d'utiliser le langage C++ ainsi que les simulateurs NS3 et SUMO.

Perspectives

Dans le cadre de travaux futurs, il serait intéressant de vérifier les performances de ces protocoles, avec des scénarios plus réalistes avec la présence de feux de signalisation, de doubles voies, de sentiers opposés, et avec un nombre important des nœuds.

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

Bibliographie



Bibliographie

- [1] Muriel.Aubert « les réseaux informatique » article. Université de nice sophia antipolis.2014. <http://members.unine.ch/muriel.aubert/images/uninice.pdf>
- [2] Khadîdja AYAD, «Sécurité du routage dans les réseaux Ad Hoc Mobile», thèse de magister ; Ecole nationale Supérieure en Informatique (ESI) Oued-Essemr Alger, Novembre 2012.
- [3] Fatma LAIDOUI, «Approche basée sur la confiance pour l'établissement des routes dans les réseaux Ad Hoc mobiles», thèse de magister ;Ecole nationale Supérieure d'informatique (ESI) Oued-Essemr/Alger, 19 décembre 2013.
- [4] Djellab.S, Mouradi.W « Simulation et évaluation des performances des Protocoles de routage GPSR, DSR et DSDV Dans les réseaux VANETs ». Mémoire Master Université A/Mira de Bejaia, 2017.
- [5] Ghrici.A, BOUTELLIS.M.W. « L'étude des performances des protocoles de routage appliqués aux réseaux Véhiculaires Cloud Computing dans une zone urbaine Ville de Khemis Miliana» Mémoire. Université khemis Miliana. (2018).
- [6] BOUZEBIBA Hadjer, Yasmina BOUIZEM. « Impact des modèles de mobilités sur les performances des protocoles de routage en milieu urbain réaliste dans les réseaux VANET (V2V)» mémoire fin d'étude, université Tlemcen. Diss. 2015
- [7] Charles E. Perkins, Samir Ranjan Das «Ad hoc On-Demand Distance Vector (AODV) Routing». Article .California RFCk3561 .2003
<https://www.researchgate.net/publication/221611438>
- [8] Meraihi Yassine « Routage Dans Les Réseaux Véhiculaires (Vanet) Cas D'un Environnement Type Ville » .Mémoire de Magister En Génie Electrique Université M'Hamed Bougara.2011
- [9] Chaudet Claude, Guérin Lassous Isabelle. « Routage QoS et réseaux Ad Hoc : de l'état de lien à l'état de nœud » article. Institut National De Recherche En Informatique Et En Automatique. R-4700, INRIA. 2015

- [10] Benchabana Ayoub et Ramla Bensaci « Analyse des protocoles de routage dans les réseaux VANET » mémoire de Master de l'Université Kasdi Merbah Ouargla 2014.
- [11] BOUABDALLAH. Imane Karima, Yasmina. BOUDERBANE. « Étude de la norme 802.11 v dans les réseaux WLAN mobile sous l'environnement de simulation NS-3 » mémoire de Master en Réseaux Mobiles et Services de télécommunications. Université Tlemcen .16/06/2015
- [12] Christos. Profentzas « Studying routing issues in vanets by using ns3 » these. Institute de Thessaloniki. Grèce. Novembre (2012).
- [13] Open Sim. Omnet++ documentation. [Https: « www.omnetpp.org »](https://www.omnetpp.org) visité septembre 2020
- [14] Vasil. Hnatyshin, Hristo.Asenov. «Design and implementation of an opnet model for simulating geo aodv manet routing protocol». Article.In Proc. of the OPNETWORK 2010 International Conference, Session: Wireless Ad Hoc and Wireless Personal Area Networks, Washington DC, 2010.
- [15] Nitin.Maslekar, Mounir Boussejra « C-drive clustering based on direction in vehicular environment » article IEEE Intelligent Vehicles .Germany. 2011
- [16] Shifali e Y. Juneja, « Simulation Environment for VANET » article. International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, n. 5, pp. 492 - 497, 2015.
- [17] F. J. Martinez, « City Mob: A Mobility Model Pattern Generator for VANETs » article .ICC Workshops -IEEE International Conference on Communications University of Zaragoza, Spain 2008.
- [18] J. Härri, M. Fiore, F. Filali e C. Bonnet, «Vanet MobiSim: generating realistic mobility patterns for VANETs» article.institut Eurécom Torino, 2006
- [19] F.E.Bustamante, J.Otto, D.Choffnes « straw street random waypoint vehicular mobility model for network simulations » disponible sur:
<http://www.oldaqualab.cs.northwestern.edu/projects/144-straw-street-random-waypoint-vehicular-mobility-model-for-network-simulations-e-g-car-networks>

- [20] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia e M.Gerla, «GloMoSim: A Scalable Network Simulation Environment » article de l'université de Californie, Los Angeles, 1999.
- [21] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalig e F. Bai, «GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks » article. International Conférence des Mobiles et système automobiles .université Carnegie Mellon, U.S.A 2006.
- [22] D. Krajzewicz, G. Hertkorn, P. Wagner and C.Rossel, «SUMO Simulation of urban mobility » article. Institute de Systèmes de transport Allemand. Aerospace Center.2015
- [23] BOUZITE Bilal « Evaluation des performances des protocoles de Routage GPSR et AODV dans Les Réseaux VANETs » mémoire de Master ,systèmes intelligents et réseau.univercité Sidi Mohamed Ben Abdallah. juin 2017.
- [24] V. Jindal and P. Bedi.« Modélisation et étude de performances dans les réseaux Vanets». Article du Journal International des sciences des ordinateurs (IJCSI), volume 13, Numéro 2, page 44, 2012.
- [25] Open Street Map France, <http://openstreetmap.fr> visité en septembre 2020.
- [26] SE Carpenter, ML Sichitiu, DA Underwood, M. Patwardhan, S Starr. «Evaluating VANET Performance in ns-3». Article. In Workshop on ns-3 (WNS3), 2014.
- [27] N. Aschenbruck, R. « BonnMotion: a mobility scenario generation and analysis tool » article.3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools'10), Malaga, 2010.
- [28] M. PiórkowskiM. Raya. A. Lezama « TraNS: Realistic Joint Traffic and Network Simulator for VANETs » article. Laboratoire de Communications et Applications, Swiss. 2008.

Annexes



Annexe A : installation des outils

Introduction

Afin d'étudier et d'évaluer les protocoles de routage dans les réseaux VANET, Nous avons utilisé le simulateur NS3 dans lequel on a utilisé le langage C++ pour écrire les différents scripts ainsi que l'apport de plusieurs modifications à plusieurs niveaux. La plateforme logicielle qu'on a utilisée pour le simulateur NS3 est le système d'exploitation Linux Ubuntu 18.04.5

Nous avons présenté dans ces annexes les étapes à suivre pour installer le simulateur NS3 et le NETanim de Vis animator qui permet de visualiser les scénarios de simulation à l'aide d'un fichier de trace [XML]. Ensuite la méthode d'installation de SUMO avec quelques images capturées à partir de l'écran. Et nous avons présenté aussi les étapes de simulation et le script « vanet-routing-compare.cc »

1-Installation de NS-3 dans Ubuntu 18.04 .5

Système d'exploitation utilisé: Ubuntu 8.04.5 ns3 version: ns3 version 3.31

La même procédure sera utile pour le système : Debian, Linux Mint.

1.1. Avant l'Installation :

Ns3 a besoin autant de dépendances, de bibliothèques de développement, de pilotes, etc.

Les commandes suivantes sont suffisantes pour installer ce qui est nécessaire.

```
$] sudo apt update
```

```
$] sudo apt upgrade
```

```
$] sudo apt-get install build-essential autoconfautomakelibxmu-dev python-pygoocanvas  
python-pygraphvizcvsvs mercurial bzrgit cmake p7zip-full python-matplotlib python-tk  
python-dev python-kiwi python-gnome2 python-gnome2-desktop-dev python-rsvg qt4-dev-  
tools qt4-qmake qt4-qmake qt4-default gnplot-x11 wireshark
```

```
$] sudo apt install build-essential autoconfautomakelibxmu-dev python-pygraphvizcvsvs  
mercurial bzrgitcmake p7zip-full python-matplotlib python-tk python-dev python-kiwi  
python-gnome2 python-gnome2-dev qt4-dev-tools qt4-qmake qt4-qmake qt4-default  
gnplot-x11 wireshark
```

La commande ci-dessus prend un certain temps à télécharger, compiler et installer, soyez patient.

1.2. Téléchargement et décompression de NS3 :

A- téléchargement : il y'a deux méthodes



methode1 : téléchargement direct avec le site <https://www.nsnam.org>

-Faites un clic droit sur le fichier téléchargé et décompressez-le dans le dossier (/home/nezar/).la place préférée pour l'installation.

Allez à l'emplacement du dossier de téléchargement et copiez le fichier dans le dossier de départ et ouvrez le terminal et donnez la commande

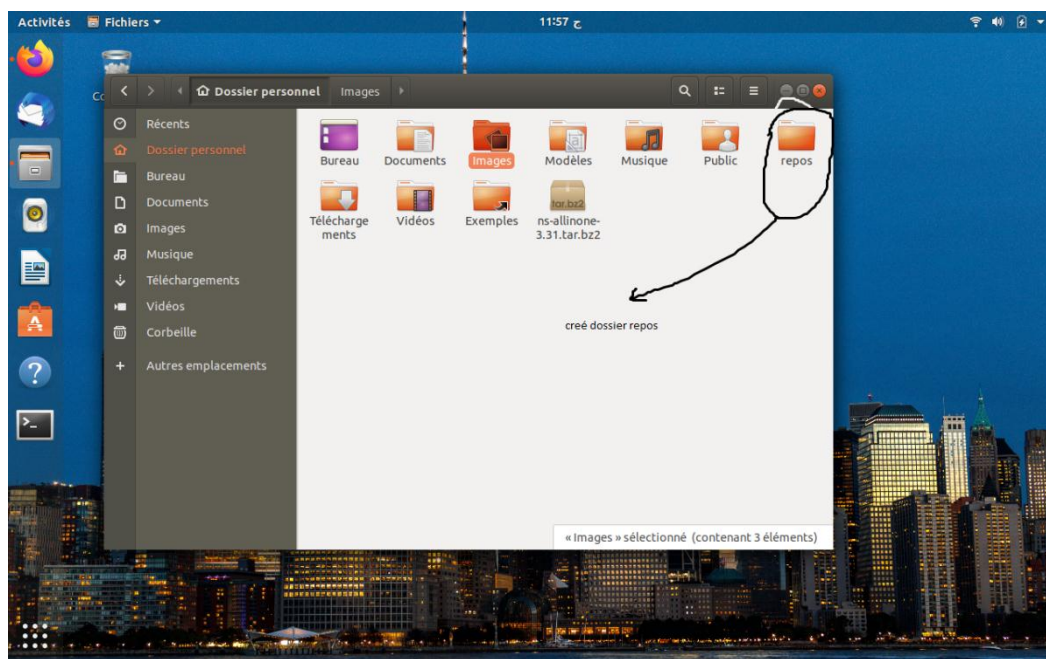
Méthode 2 : avec le terminal

Entrée les commande suivant sur le terminal pour le téléchargement.

\$] cdmkdir repos (créé un dossier sous nom repos, c'est pas obligatoire)

\$] cd repos (entrée dans le fichier repos)

\$] wget<http://www.nsnam.org/releases/ns-allinone-3.31.tar.bz2>



B- Décompression :

Pour décompresser et convertir le fichier tar en fichier bz2 faites entrer la commande suivante :

\$] tarjxvf ns-allione-3.27.tar.bz2

C- compilation :

\$] cd ns-allinone-3.31/

\$] ./build.py --enable-examples --enable-tests

Cela prendra un certain temps pour obtenir la compilation et la construction. Une fois l'installation réussie, vous obtiendrez un écran comme indiqué ci-dessous.

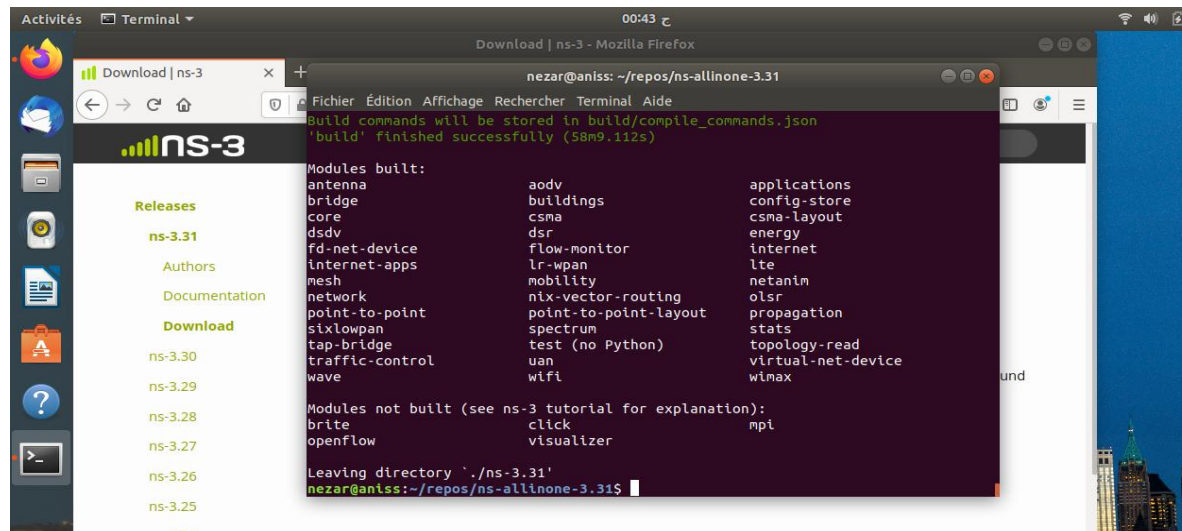


Figure : ns3 installation

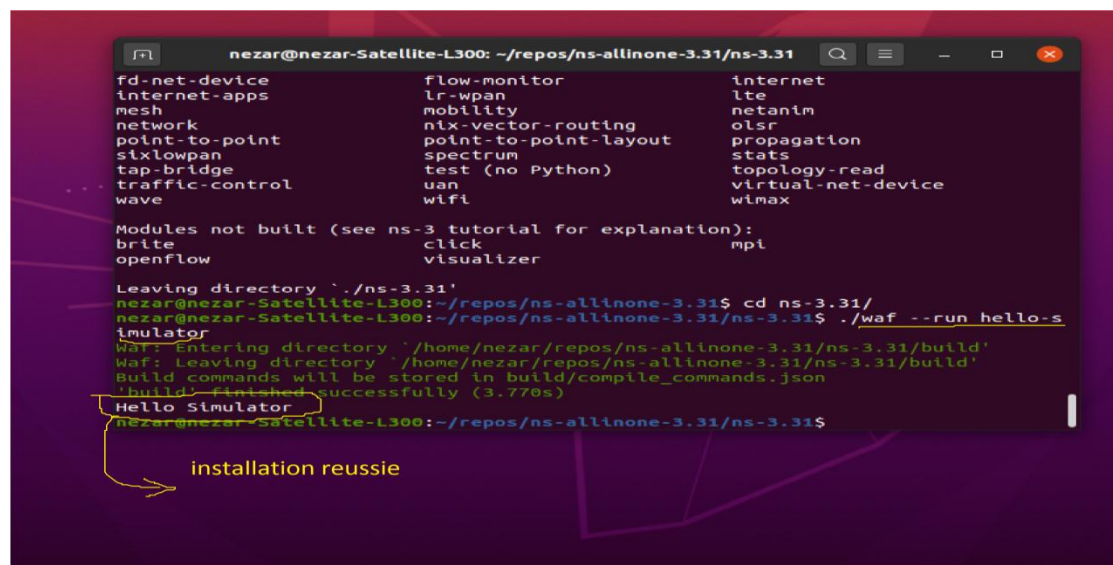
Cela indique que ns3 est construit avec succès.

Pour vérifier les applications. Faire les étapes suivantes

```
$] cd ns-3.27/
```

```
$] ./waf --run hello-simulator
```

Cela imprimera le simulateur Hello qui indique que ns3 est installé avec succès.



2 : installation de simulateur SUMO :

Etapes d'installation :

Etape 1:telechargement des prérequis

```
$] sudo apt update
```

```
$] sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev
```

```
libgl2ps-dev swig
```



```

nezar@dell: ~/sumo/build/cmake-build
Fichier Édition Affichage Rechercher Terminal Aide
Atteint :3 http://ppa.launchpad.net/yannubuntu/boot-repair/ubuntu bionic InRelease
Atteint :4 http://fr.archive.ubuntu.com/ubuntu bionic InRelease
Atteint :5 http://fr.archive.ubuntu.com/ubuntu bionic-updates InRelease
Atteint :6 http://fr.archive.ubuntu.com/ubuntu bionic-backports InRelease
Réception de :7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Atteint :8 http://fr.archive.ubuntu.com/ubuntu bionic-proposed InRelease
88,7 ko réceptionnés en 2s (42,8 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
3 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
nezar@dell:~$ sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev
libproj-dev libl2ps-dev swig
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
python est déjà la version la plus récente (2.7.15-rc1-1).
cmake est déjà la version la plus récente (3.10.2-1ubuntu2.18.04.1).
g++ est déjà la version la plus récente (4:7.4.0-1ubuntu2.3).
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  linux-hwe-5.4-headers-5.4.0-42
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
Les paquets supplémentaires suivants seront installés :
  default-libmysqldclient-dev gdal-data hdf5-helpers libaec-dev libaec0
  libarmadillo-dev libarmadillo8 libarpack2 libarpack2-dev libblas-dev
  libcupss2-dev libcupssfilters-dev libcupssimage2-dev libcurl4-gnutls-dev
  libldap-dev libldap25 libldapclient6v5 libldapserver7v5 libepsilon-dev
  libepsilon1 libfox-1.6-0 libfreexl-dev libfreexl1 libfyba-dev libfyba0
  libgdal20 libgeos-3.6.2 libgeos-c1v5 libgeos-dev libgeotiff-dev libgeotiff2

```

Étape 2 :

```
$] sudo apt install git
```

```
$] git clone --recursive https://github.com/eclipse/sumo
```

```

nezar@dell: ~/sumo/build/cmake-build
Fichier Édition Affichage Rechercher Terminal Aide
Paramétrage de libgdal20 (2.2.3+dfsg-2) ...
Paramétrage de libgdal-dev (2.2.3+dfsg-2) ...
Paramétrage de libfox-1.6-dev (1.6.56-1) ...
update-alternatives: utilisation de « /usr/bin/fox-config-1.6 » pour fournir « /usr/bin/fox-config » (
fox-config) en mode automatique
update-alternatives: utilisation de « /usr/bin/reswrap-1.6 » pour fournir « /usr/bin/reswrap » (reswr
ap) en mode automatique
Traitement des actions différées (« triggers ») pour libc-bin (2.27-3ubuntu1.3) ...
Traitement des actions différées (« triggers ») pour man-db (2.8.3-2ubuntu0.1) ...
nezar@dell:~$ sudo apt install git
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
git est déjà la version la plus récente (1:2.17.1-1ubuntu0.7).
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  linux-hwe-5.4-headers-5.4.0-42
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 3 non mis à jour.
nezar@dell:~$ git clone --recursive https://github.com/eclipse/sumo
Clonage dans 'sumo' ..
remote: Enumerating objects: 15090, done.
remote: Counting objects: 100% (15090/15090), done.
remote: Compressing objects: 100% (2628/2628), done.
remote: Total 936698 (delta 13983), reused 12722 (delta 12461), pack-reused 921518
Réception d'objets: 100% (936698/936698), 650.61 MiB | 775.00 KiB/s, fait.
Résolution des deltas: 100% (784366/784366), fait.
Extraction des fichiers: 100% (76505/76505), fait.
Sous-module 'build/brw' (https://github.com/DLR-TS/homebrew-sumo) enregistré pour le chemin 'build/br
ew'
Sous-module 'tools/contributed/saga' (https://github.com/lcodeca/SUMOActivityGen) enregistré pour le c
hemin 'tools/contributed/saga'
Sous-module 'tools/contributed/trac4matlab' (https://github.com/pipecosta/trac4matlab) enregistré p

```

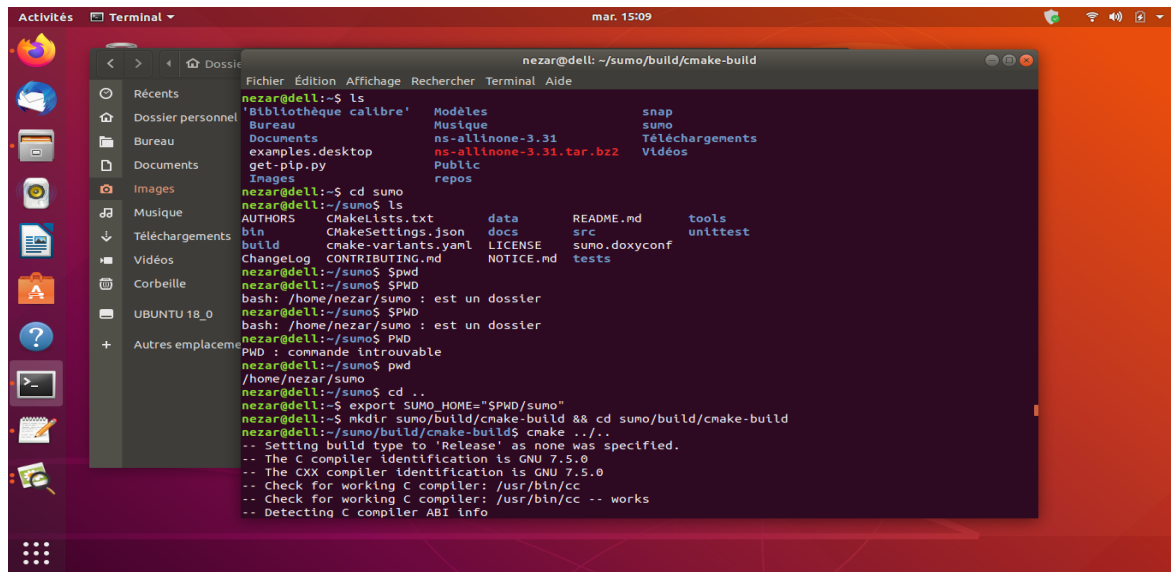
Étape 3 : - définir le chemin

```
« export SUMO_HOME="$PWD/sumo »
```

Étape 4 : compilation

```
mkdir sumo/build/cmake-build && cd sumo/build/cmake-build
```

```
$] cmake ../..
```



```
nezar@dell: ~/sumo/build/cmake-build
nezar@dell:~$ ls
'Bibliothèque calibre'  Modèles          snap
Bureau                 Musique          sumo
Documents              ns-allinone-3.31 Téléchargements
examples.desktop      ns-allinone-3.31.tar.bz2 Vidéos
get-pip.py             Public
Images                repos
nezar@dell:~/sumo$ cd sumo
nezar@dell:~/sumo$ ls
AUTHORS  CMakeLists.txt  data  README.md  tools
bin      CMakeSettings.json  docs  src        unittest
build   cmake-variants.yaml  LICENSE  sumo.doxyconf
Changelog  CONTRIBUTING.md  NOTICE.md  tests
nezar@dell:~/sumo$ pwd
bash: /home/nezar/sumo : est un dossier
nezar@dell:~/sumo$ pwd
bash: /home/nezar/sumo : est un dossier
nezar@dell:~/sumo$ PWD
PWD : commande introuvable
nezar@dell:~/sumo$ pwd
/home/nezar/sumo
nezar@dell:~/sumo$ cd ..
nezar@dell:~$ export SUMO_HOME="$PWD/sumo"
nezar@dell:~$ mkdir sumo/build/cmake-build && cd sumo/build/cmake-build
nezar@dell:~/sumo/build/cmake-build$ cmake ../..
-- Setting build type to 'Release' as none was specified.
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
```

\$] make -j8

Annexe B : étapes de la simulation

1-import du map :

1-sumo

`cdsumo/tools`

dans l'arborescence suivante `:/home/nezar/sumo/tools`

2-on va ouvrir le fichier `osmWebWizard.py`

avec la commande sur terminal:`pythonosmWebWizard.py`

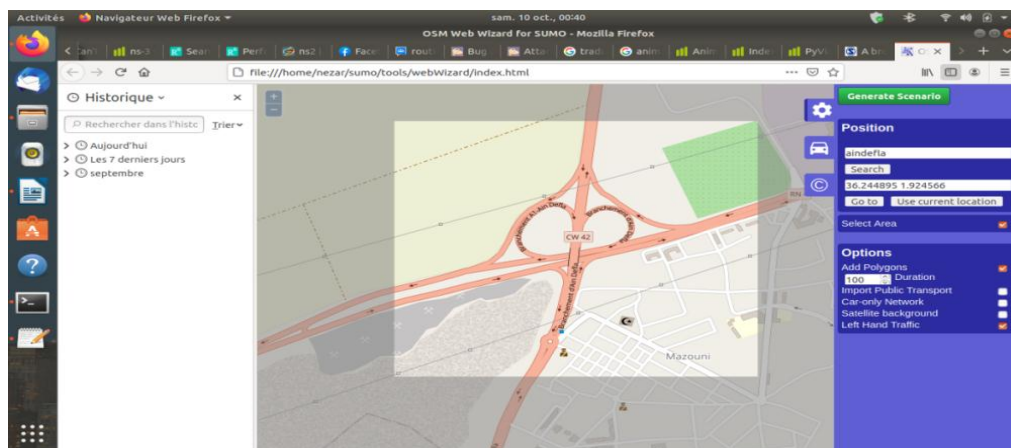
➤ En obtient la figure suivante en appliquant les modifications suivantes :

-selectionner la surface,

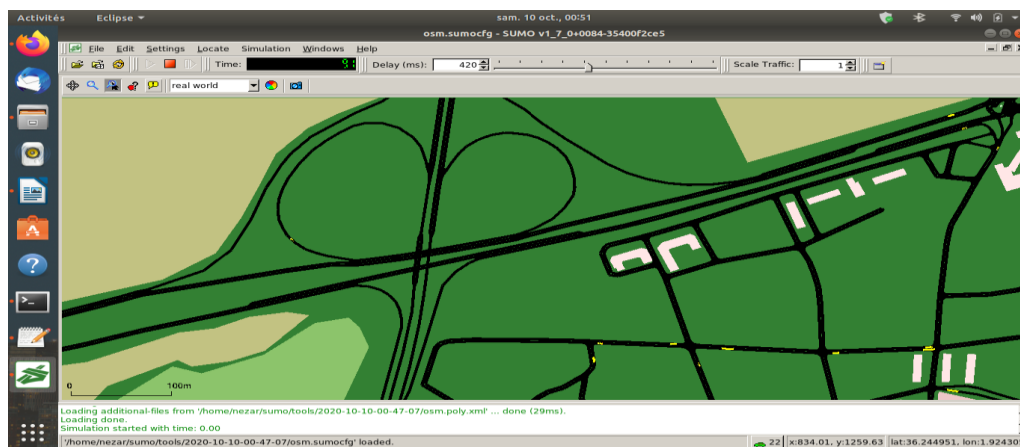
-temps de simulation =100s

-nombre:vehicule(5),bus(5),trax(5),

-on a choisi l'intersection de la route n°4 et la rocade de bourached situer près de village MAZOUNI a aindefla



➤ Après on doit générer le map

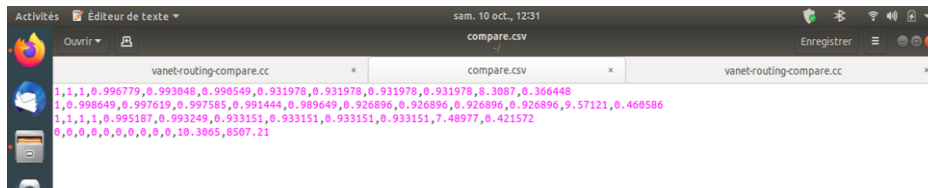


➤ après la simulation un fichier a été créé sur le répertoire `sumo/tools` :`2020-10-10-00-47-07` qui contient les fichiers suivants :

`build.bat` `osm.netccfgosm.polycfg` `osm.truck.trips.xml`

`osm_bbox.osm.xml` `osm.net.xml` `osm.poly.xml` `osm.view.xml`

A chaque fois on refait la simulation après chaque changement et on obtient des fichiers de type CSV.



Exemples des résultats de simulation

OLSR 30 NODE

BSM_PDR1=1 BSM_PDR2=0.998575 BSM_PDR3=0.998001 BSM_PDR4=0.996447
 BSM_PDR5=0.992814 BSM_PDR6=0.981683 BSM_PDR7=0.914528 BSM_PDR8=0.914528
 BSM_PDR9=0.914528 BSM_PDR10=0.914528 Goodput=8.9841Kbps MAC/PHY-oh=0.353935

AODV 30 NODE

BSM_PDR1=0.99823 BSM_PDR2=0.996201 BSM_PDR3=0.992618 BSM_PDR4=0.990849
 BSM_PDR5=0.988079 BSM_PDR6=0.977005 BSM_PDR7=0.910555 BSM_PDR8=0.910555
 BSM_PDR9=0.910555 BSM_PDR10=0.910555 Goodput=10.0079Kbps MAC/PHY-oh=0.425461

DSDV 30 NODE

BSM_PDR1=0.99823 BSM_PDR2=0.995726 BSM_PDR3=0.994309 BSM_PDR4=0.992249
 BSM_PDR5=0.989855 BSM_PDR6=0.979772 BSM_PDR7=0.913401 BSM_PDR8=0.913401
 BSM_PDR9=0.913401 BSM_PDR10=0.913401 Goodput=6.10032Kbps MAC/PHY-oh=0.418504

DSR 30 NODE

BSM_PDR1=0 BSM_PDR2=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0
 BSM_PDR7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=10.3065Kbps
 MAC/PHY-oh=8507.21

Les autres métriques à savoir NRL et PL et débit seront calculées selon l'équation présentée dans la section 3.3.

Annexe C : script vanet-routing-compare

Nous avons présenté seulement les modifications apportées au script vanet-routing-compare.cc. En raison de la grande longueur de ce script qui contient 2453 lignes nous avons changé les paramètres suivants :

```
//-----Include module Flow Monitor-----
110 #include "ns3/flow-monitor-module.h"
123 #include "ns3/animation-interface.h"
...
1705 void
1706 VanetRoutingExperiment::Run ()
1707 {
1708   NS_LOG_INFO ("Run Simulation.");
1709
1710   CheckThroughput ();
1711
1712 //-----Flow monitor-----
1713 Ptr<FlowMonitor>flowMonitor;
1714 FlowMonitorHelperflowHelper;
1715 flowMonitor = flowHelper.InstallAll();
1716
1717 Simulator::Stop (Seconds (m_TotalSimTime));
1718 Simulator::Run ();
1719
1720 flowMonitor->SerializeToXmlFile("NAME_FILE.xml", true, true);
1721
1722 Simulator::Destroy ();
1723 }
...
2355 void
2356 VanetRoutingExperiment::SetupScenario ()
2357 {
2358   // member variable parameter use
2359   // defaults or command line overrides,
2360   // except where scenario={1,2,3,...}
2361   // have been specified, in which case
2362   // specify parameters are overwritten
2363   // here to setup for specific scenarios
2364
2365   // certain parameters may be further overridden
2366   // i.e. specify a scenario, override tx power.
2367
2368   if (m_scenario == 1)
2369   {
2370     // 40 nodes in RWP 1000 m x 1000 m synthetic highway, 10s
2371     m_traceFile = "";
```

```
2372     m_logFile = "";
2373     m_mobility = 2;
2374     if (m_nNodes == 156)
2375     {
2389 else if (m_scenario == 2)
2390 {
2391     // Realistic vehicular trace in SIDI HAMOU a aindefla
2392     // "low density, 23 total vehicles"
2393     m_traceFile = "/home/nezar/mobility.tcl";
2394     m_logFile = "mobnezar.log";
2395     m_mobility = 1;
2396     m_nNodes = 20;
2397     m_TotalSimTime = 60.01;
2398     m_nodeSpeed = 20;
2399     m_nodePause = 0;
2400     m_CSVfileName = "AODV30nodsc1.csv";
2401     m_CSVfileName = "AODV30nodsc12.csv";
2402 }
2403 }
```