

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira de Béjaïa



Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Master Professionnel

Thème

**Conception et Réalisation d'un simulateur  
dédié pour les réseaux véhiculaires**

Présenté par :

BENZOUAOUA Abdellah KEBBI Menad

Soutenu devant le jury composé de :

Président	Mme. MITIDJI
Encadreur	Mme. TASSOULT Nadia
Examinatrice	Mme ZIDANI Faroudja

Promotion 2015 – 2016

## *Remerciements*

Nos remerciements vont en premier lieu notre encadreur Mme. **TASSOULT Nadia** pour la préciosité de ses conseils, son infinie disponibilité et son orientation qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.

Nous tenons à remercier chacun des membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent à tous nos enseignants et les membres du département d'informatique de l'université **ABDERRAHMANE MIRA**.

Ainsi qu'à tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail.

## *Dédicaces*

*A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de notre profonde affection et de notre attachement indéfectible ; nulle dédicace ne puisse exprimer ce qu'on leur doit.*

*A nos frères et sœurs et tous nos amis pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.*

# Table des Matières

<b>Table des Matières</b>	<b>i</b>
<b>Table des Figures</b>	<b>v</b>
<b>Liste des Tableaux</b>	<b>vii</b>
<b>Liste des Abréviations</b>	<b>viii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Généralité sur les réseaux véhiculaires</b>	<b>3</b>
Introduction . . . . .	4
1.1 les réseaux véhiculaire VANET . . . . .	4
1.1.1 Définition d'un Réseau VANET . . . . .	4
1.1.2 Composants de réseaux VANET . . . . .	5
1.1.2.1 TA (Trusted Authority) . . . . .	5
1.1.2.2 RSUs (Road Side Unit) . . . . .	5
1.1.2.3 OBU (On-Board Unit) . . . . .	5
1.1.3 Caractéristique Des Réseaux VANET . . . . .	6
1.1.3.1 Le potentiel énergétique . . . . .	6
1.1.3.2 L'environnement de communication et le modèle de mobilité .	6
1.1.3.3 Le modèle de communication . . . . .	7
1.1.3.4 La taille du réseau . . . . .	7
1.1.4 Les applications des réseaux VANET . . . . .	7

1.1.4.1	Applications de gestion du trafic routier . . . . .	7
1.1.4.2	Applications de confort . . . . .	8
1.1.4.3	Applications de sécurité du trafic routier . . . . .	8
1.1.5	Architectures des réseaux véhiculaires . . . . .	9
1.1.5.1	Communications de Véhicule à Véhicule (V2V) . . . . .	9
1.1.5.2	Communications de Véhicule à Infrastructure (V2I) . . . . .	9
1.1.5.3	Communication hybride . . . . .	10
1.1.6	Les techniques de communications . . . . .	11
1.1.6.1	Les communications radio . . . . .	11
1.1.6.2	Les communications sans fil . . . . .	12
1.1.7	Types de message . . . . .	12
1.1.7.1	Message de contrôle . . . . .	13
1.1.7.2	Message d’alerte . . . . .	13
1.1.7.3	Autres messages . . . . .	13
1.2	Normes et standardisation dans les VANET . . . . .	13
1.2.1	DSRC (Dedicated Short Range Communications) . . . . .	14
1.2.2	La norme IEEE 802.11p . . . . .	15
1.2.3	WAVE (Wireless Access in Vehicular Environments) . . . . .	16
	Conclusion . . . . .	19
<b>2</b>	<b>Protocoles de routage dans les VANETs</b>	<b>20</b>
	Introduction . . . . .	21
2.1	Routage dans ad hoc . . . . .	21
2.2	Routage dans les VANETs . . . . .	22
2.3	Classification des protocoles de routage dans les réseaux VANET . . . . .	22
2.3.1	Les protocoles de routage basés sur la topologie . . . . .	23
2.3.1.1	Les protocoles réactifs . . . . .	23
2.3.1.2	Les protocoles proactifs . . . . .	27
2.3.1.3	Protocoles hybrides . . . . .	29
2.3.1.4	Comparaison entre les protocoles de routage proactive et réactive	30
2.3.2	Les protocoles de routage basés sur la géographique . . . . .	32

Conclusion . . . . .	35
<b>3 Les simulateurs des VANETs</b>	<b>36</b>
Introduction . . . . .	37
3.1 La Simulation dans un Réseau VANET . . . . .	37
3.2 Les Simulateurs de Réseaux Ad Hoc . . . . .	37
3.2.1 Le simulateur NS-2 . . . . .	38
3.2.1.1 Les composants de NS2 . . . . .	39
3.2.1.2 Modèle de mobilité . . . . .	39
3.2.1.3 Modèle de propagation dans NS2 . . . . .	40
3.2.2 Le simulateur NS-3 . . . . .	40
3.2.3 Le simulateur OMNET ++ . . . . .	42
3.2.3.1 Architecture d'OMNeT++ . . . . .	42
3.2.3.2 Composants d'OMNET++ . . . . .	43
3.2.4 Comparaison entre les simulateurs NS-2 et OMNET++ . . . . .	44
3.3 Les Simulateurs de mobilité et de trafic routier . . . . .	45
3.3.1 Les simulateurs indépendants . . . . .	45
3.3.1.1 Le simulateur VanetMobisim . . . . .	46
3.3.1.2 Le simulateur SUMO . . . . .	47
3.3.1.3 Le simulateur CORSIM . . . . .	47
3.3.1.4 Le simulateur PARAMICS . . . . .	48
3.3.1.5 Le simulateur CityMob . . . . .	48
3.3.2 Les simulateurs intégrés . . . . .	49
3.3.2.1 Le simulateur NCTU . . . . .	49
3.3.2.2 Le simulateur TraNs . . . . .	50
Conclusion . . . . .	50
<b>4 Conception et Réalisation du Simulateur</b>	<b>51</b>
PART I : conception . . . . .	52
Introduction . . . . .	53
4.1 La problématique . . . . .	53
4.2 Processus de la simulation . . . . .	54

4.3	Besoins fonctionnels . . . . .	54
4.3.1	Description générale . . . . .	54
4.3.2	Cas d'utilisation . . . . .	55
4.3.2.1	Cas d'utilisation du simulateur . . . . .	55
4.3.2.2	Cas d'utilisation au niveau applicatif . . . . .	56
4.3.2.3	Cas d'utilisation au niveau réseau . . . . .	57
4.3.2.4	Diagrammes de séquences . . . . .	57
4.3.2.5	Diagramme d'état transition . . . . .	59
4.3.2.6	Diagramme de classe . . . . .	60
	Conclusion . . . . .	60
	PART II : Réalisation . . . . .	61
	Introduction . . . . .	62
4.4	Environnement de travail . . . . .	62
4.4.1	Environnement matériel . . . . .	62
4.4.2	Environnement logiciel . . . . .	62
4.4.2.1	Éclipse Luna . . . . .	62
4.4.2.2	Java . . . . .	63
4.5	Présentation de simulateur . . . . .	63
	Conclusion . . . . .	69
	<b>Conclusion Générale et Perspectives</b>	<b>70</b>
	<b>A Annexe</b>	<b>72</b>
A.1	Annexe A . . . . .	73
A.1.1	téléchargement d'une carte map sur le site openstreetmap.com . . . . .	73

**Bibliographie**

# Table des figures

1.1	Nœuds représentant les réseaux VANET. . . . .	4
1.2	Exemple d'un réseau véhiculaire . . . . .	6
1.3	Types de communication dans un réseau de véhicules . . . . .	11
1.4	Exemple d'architecture réseau de DSRC . . . . .	15
1.5	WAVE On Board Unit. . . . .	17
1.6	WAVE RoadSide Unit. . . . .	17
1.7	Architecture protocolaire du système WAVE. . . . .	18
1.8	Modules du standard IEEE 1609.1. . . . .	18
2.1	Les protocoles de routage dans les réseaux VANETs. . . . .	23
2.2	Construction de l'enregistrement de route dans DSR. . . . .	25
2.3	L'envoi du chemin ou de la route replay (RREP). . . . .	26
2.4	Erreur dans DSR : envoi de Route Error (RRER). . . . .	27
3.1	Synoptique de l'interpréteur de script Otcl avec les bibliothèques C++ de simulation de réseau. . . . .	38
3.2	Architecture en couches du simulateur NS-3 . . . . .	41
3.3	Architecture modulaire du simulateur OMNET++ . . . . .	43
3.4	interface OMNET++ . . . . .	44
3.5	Simulateur de VANET basée sur le couplage de VanetMobisim et NS2 . . . . .	46
4.1	processus de la simulation de notre simulateur. . . . .	54
4.2	Diagramme de cas d'utilisation pour la création de scénario. . . . .	55

4.3	diagramme de cas d'utilisation au niveau applicatif. . . . .	56
4.4	diagramme de cas d'utilisation au niveau réseau. . . . .	57
4.5	Diagramme de séquence de simulateur. . . . .	58
4.6	Diagramme d'état de transition de notre simulateur . . . . .	59
4.7	Diagramme de classe du simulateur . . . . .	60

# Liste des tableaux

2.1	Comparaison entre protocoles proactifs et protocoles réactifs . . . . .	31
3.1	Liste des principaux composants disponible dans NS2 . . . . .	39
3.2	Composent de OMNET++ . . . . .	44
3.3	Comparaison entre NS2 et OMNET ++ . . . . .	45

# Liste des Abréviations

A-STAR Anchor-based Street and Traffic Aware Routing

ABS système anti-blocage

AODV Ad hoc On-demand Distance Vector

BRP Bordercast Resolution Protocol

CA autorité de confiance

DAB Digital Audio Broadcasting

DSDV Destination-Sequenced Distance-Vector

DSR Dynamic Source Routing

DSRC Dedicated Short Range Communications

DVB Digital Vidéo Broadcasting

EDCA Enhanced Distributed Channel Access

EIRP Effective Isotropic Radiated Power

ESP Electronic Stability Control

ETSI European Télécommunications Standards Institute

GLS Grid Location Service

GPS Global Position System

GPSR	Greedy Perimeter Stateless Routing
GSM	Global Système for Mobile communications
GSR	Global State Routing
IARP	IntrAzone Routing Protocol
IERP	IntrEzone Routing Protocol
IETF	Internet Engineering Task Force
MANET	Mobile ad hoc network
MORA	MOvement-based Routing Algorithm
MPR	Multi-Point Relaying
NCTU	Nation Chiao Tang University Network Simulator
NDP	Neighbour Discovery Protocol
NPAD	No Program Associated Data
OBU	On-Board Unit
OLSR	Optimized Link State Routing
OTcl	Object Tools Command Language
PAD	Program Associated Data
RDS	Radio Data System
RERR	Route Error Message
RFC	Request For Comment
RM	Resource Manager
RREQ	Route Request Message
RSU	Road Side Unit

STI	systèmes de transports intelligents
TA	Trusted Authority
TMC	Traffic Message Channel
TraNs	Traffic et Network Simulation Environment
UMB	Urban Multi hop Broadcast Protocol
UML	Universel Modeling Language
VADD	Vehicle-Assisted Data Delivery
VSN	Vehicular Sensor Network
WAVE	Wireless Access in Vehicular Environments
WSMP	WAVE Short Messages Protocol
ZRP	Zone Routing Protocol

# Introduction Générale

Depuis quelques années, le besoin d'être connecté est devenu fondamental pour l'homme. Ce besoin pose de plus en plus de défis pour la technologie moderne l'obligeant à plus d'innovation et de créativité. Ainsi, de nouvelles technologies sont apparues comme les réseaux sans fil, et les réseaux Ad Hoc.

Ces réseaux se sont vite répandus depuis les lieux de travail aux domiciles en passant par les aires de distraction. Ainsi, l'introduction de ces réseaux dans les voitures est devenu une chose indispensable, d'où l'apparition des réseaux Ad Hoc véhiculaires ou VANET (Vehicular Ad Hoc Networks). Les VANETs visent à déployer la communication et l'échange d'informations entre les usagers de la route.

Afin d'étudier les VANETs, le déploiement sur terrain n'est malheureusement pas envisageable à ce jour, d'où le recours à la simulation. Plusieurs simulateurs ont été mis à la disposition des chercheurs dans ce but (OMNET++, NS2, ...). Lors d'une simulation, la mobilité est un paramètre à ne pas négliger, car les unités dans un VANET peuvent se déplacer à grande vitesse suivant un schéma de mobilité particulier. Pour modéliser la mobilité des VANETs, plusieurs modèles ont été conçus. Ces modèles de mobilité doivent prendre en considération les contraintes de la mobilité véhiculaire, pour que la simulation soit proche de la réalité.

Notre projet consiste à implémenter notre propre simulateur dédié aux réseaux véhiculaires : VANETs. La communication entre véhicules est assurée par le protocole de routage A-Star. Ainsi, et pour modéliser la mobilité des véhicules, nous avons exploité des cartes *.osm*. directement téléchargeable à partir des sites.

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre, présente un état de l'art sur les réseaux VANETs : définition, modèle de système, architecture, et en fin les normes et les standardisations des VANETs.

Le deuxième chapitre est consacré à la description des différents protocoles de routage dans Ad Hoc et VANET qui sont basés sur la topologie et des protocoles basés sur la géographie, selon la topologie il existe des protocoles réactifs comme AODV, DSR et des protocoles proactifs comme OLSR, GSR, DSDV et des protocoles hybrides comme ZRP, selon la géographie il existe des protocoles comme A-STAR, UMB, GyTAR, VADD, MORA, GPSR.

Dans le troisième chapitre, nous décrivons les différents simulateurs dédiés pour les VANETs. Ils en existe plusieurs. Généralement pour simuler les réseaux vanets deux simulateurs sont utilisés conjointement : un simulateur pour simuler le réseau tels que : NS3, OMNET ++, et un deuxième pour simuler le trafic tel que : SUMO, MOVE et VanetMobisim. Ainsi, des simulateurs intégrés qui ont été, par la suite, conçus pour simuler les VANET tel que : VEINS, TraNs.

Le dernier chapitre est composé de deux parties : conception et réalisation. Nous discutons, dans la première partie, de la conception du simulateur en exposant notre problématique, les besoins fonctionnels et les différents diagrammes (cas d'utilisation, séquences, classes ...etc). La deuxième partie détaille les étapes de réalisation de notre simulateur à savoir les outils matériels et logiciels utilisés. Ainsi nous donnons une vue générale sur ses fonctionnalités.

Nous terminons par une conclusion et quelques perspectives

# 1

Généralité sur les réseaux véhiculaires

## Introduction

Les réseaux véhiculaires dérivent de l'exploitation des technologies conçues pour les réseaux Ad Hoc mobile (MANETs) leur élaboration accentue l'émergence des systèmes de transports intelligents (STI) qui ont pour but d'améliorer la sécurité routière et de rendre plus efficace, voire plus convivial, le temps passé sur la route. À l'aide des systèmes embarqués aussi bien dans la voiture qu'installés au bord des routes et des alertes sur des accidents de la route. Aussi le passage des voitures peuvent s'échanger des données (musique, vidéo) et d'autres informations utiles ; ceci pour rendre le temps passé sur la route, moins ennuyeux.

### 1.1 les réseaux véhiculaire VANET

#### 1.1.1 Définition d'un Réseau VANET

Un réseau VANET (véhiculaire Ad Hoc) est un réseau de communication entre véhicules intelligents équipés de calculateurs, de périphériques réseau et de différents types de capteurs.

Les VANET font parti de la famille des réseaux MANET (mobile Ad Hoc Network) qui fonctionnent dans des réseaux à liaison point à point sans infrastructure, c'est-à-dire que tout nœud constituant le réseau est un point d'accès. Dans un réseau VANET les nœuds sont les véhicules intelligents appartenant au réseau. [1]

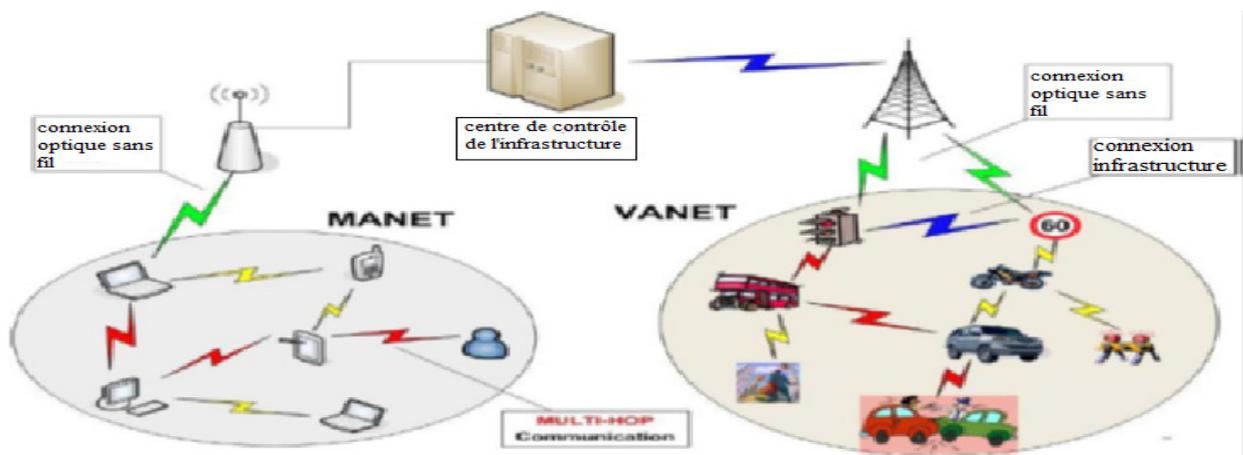


FIGURE 1.1 – Nœuds représentant les réseaux VANET. [1]

## 1.1.2 Composants de réseaux VANET

Un réseau VANET est constitué principalement de trois entités :

### 1.1.2.1 TA (Trusted Authority)

Dites CA (*autorité de confiance*). C'est une source d'authenticité de l'information. Elle assure la gestion et l'enregistrement de toutes les entités sur le réseau (RSU et OBU). La TA est sensée connaître toutes les vraies identités des véhicules et au besoin les divulguer pour les forces de l'ordre. Aussi, la TA dans certains travaux se charge de la délivrance et l'attribution des certificats et des pseudonymes de communications. [2]

### 1.1.2.2 RSUs (Road Side Unit)

Ces entités sont les subordonnés des TA. Elles sont installées au bord des routes. Elles peuvent être principalement, des feux de signalisation, des lampadaires ou autres. Leur principale responsabilité est de soutenir la TA dans la gestion du trafic et des véhicules. Elles représentent des points d'accès au réseau et aux différentes informations sur la circulation.

### 1.1.2.3 OBU (On-Board Unit)

Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres). Leurs rôles sont d'assurer la localisation, la réception, le calcul, le stockage et l'envoi des données sur le réseau. Ce sont des émetteurs-récepteurs qui assurent la connexion du véhicule au réseau.

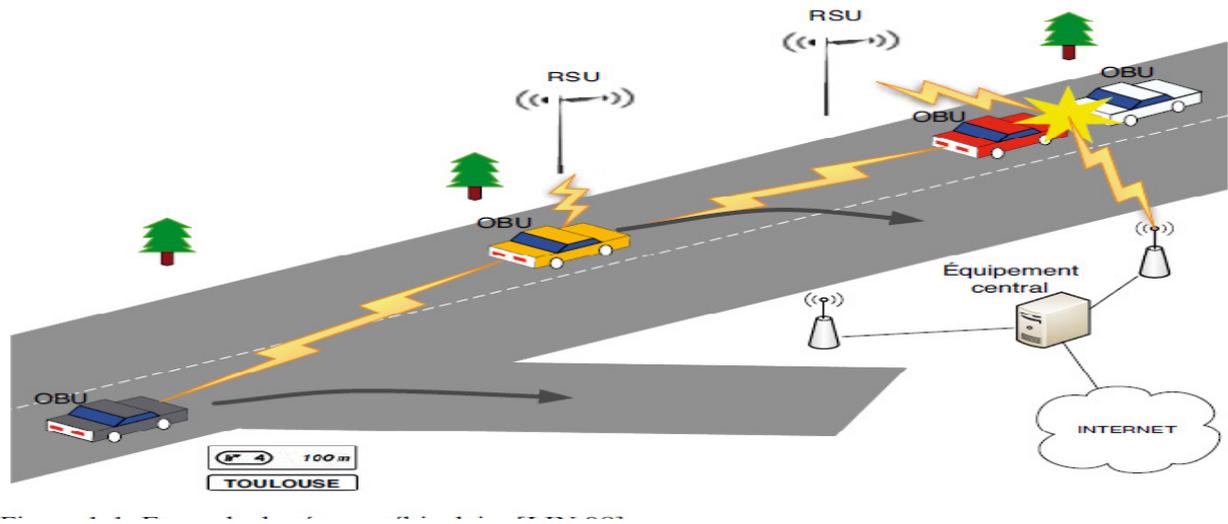


FIGURE 1.2 – Exemple de réseau véhiculaire [3]

### 1.1.3 Caractéristique Des Réseaux VANET

Les réseaux véhiculaires (VANET) se distinguent des réseaux sans fils traditionnels par un certain nombre de caractéristiques spécifiques dont on peut citer : [4]

#### 1.1.3.1 Le potentiel énergétique

La différence des réseaux sans fils traditionnels où la contrainte d'énergie représente un facteur limitant important, les entités des réseaux véhiculaires disposent de grandes capacités énergétiques qu'elles tirent du système d'alimentation des véhicules.

#### 1.1.3.2 L'environnement de communication et le modèle de mobilité

Les réseaux véhiculaires imposent la prise en compte d'une plus grande diversité environnementale. Du fait de la mobilité des véhicules, il est en effet possible de passer d'un environnement urbain caractérisé par de nombreux obstacles à la propagation des signaux, à un environnement périurbain ou autoroutier présentant des caractéristiques différentes. En plus de cette diversité environnementale, les réseaux véhiculaires se distinguent également des réseaux sans fils ordinaires par un modèle de mobilité dont une des traductions les plus évidentes est l'importante vitesse des nœuds qui réduit considérablement les durées de temps

pendant lesquelles les nœuds peuvent communiquer.

### 1.1.3.3 Le modèle de communication

Les réseaux véhiculaires ont été imaginés principalement pour les applications liées à la sécurité routière (ex. diffusion de messages d'alerte). Dans ce type d'application, les communications se font presque exclusivement par reliage successifs d'une source vers une multiplicité de destinataires. Le modèle de transmission en Broadcast ou en Multicast est donc appelé à dominer largement dans les réseaux véhiculaires, ce qui n'est par exemple pas sans conséquence sur la charge du réseau et le modèle de sécurité à mettre en œuvre.

### 1.1.3.4 La taille du réseau

Étant donné les avancées importantes réalisées dans le domaine des communications sans fil et les bas coûts des équipements associés, les véhicules qui intègrent déjà massivement des systèmes GPS et des équipements Bluetooth, seront très probablement équipés et ce, tout aussi massivement, de plateformes de communication leur permettant de constituer de véritables réseaux. Ce faisant, et compte tenu de l'importance sans cesse grandissante de la densité et du parc des véhicules, on peut s'attendre à ce que la taille des réseaux véhiculaires dont les déploiements restent encore très confidentiels, soit d'une tout autre ampleur. L'importance potentielle de la taille des réseaux véhiculaires constitue donc une caractéristique majeure à prendre en compte dans la conception de ces réseaux.

## 1.1.4 Les applications des réseaux VANET

Les principales applications des réseaux VANET peuvent être classées en trois catégories [5]

### 1.1.4.1 Applications de gestion du trafic routier

Les applications de gestion de trafic sont axées sur l'amélioration des conditions de circulation dans le but de réduire les embouteillages et les risques d'accidents. Elles fournissent aux conducteurs un support technique leur permettant d'adapter leur parcours à la situation du trafic routier. Ces applications visent à équilibrer la circulation des véhicules sur les routes pour

une utilisation efficace de la capacité des routes et des carrefours et à réduire par conséquent les pertes humaines, la durée des voyages et la consommation d'énergie...etc.

#### 1.1.4.2 Applications de confort

Cette catégorie comporte toutes les applications qui participent au confort du conducteur et qui ne relèvent pas de la gestion du trafic ni de la sécurité routière. Ces applications se présentent donc autant que services fournis au conducteur. Parmi ces applications, citons les panneaux d'annonces locales : d'ordre commercial comme les offres de restaurants, la présence de stations-service proximité, ou culturel comme des informations touristiques relatives à la localisation du véhicule.

Il ya aussi des services télématiques comme le péage à distance sur autoroute, le paiement automatique dans les stations-service (ce qui peut faciliter la vie des handicapés). Un autre type d'application de confort est la communication à vocation de divertissement. Une offre de connexion internet à bord avec vidéo à la demande en est un parfait exemple.

À toutes ces applications s'ajoutent aussi les communications point à point entre deux conducteurs qui voyagent ensemble. Ils peuvent ainsi s'échanger des messages ou partager des données (vidéo, musique, itinéraires, jeux en réseau). La vie des usagers pourra aussi être facilitée par le contrôle à distance de véhicule de manière électronique (vérification du permis de conduire, contrôle technique, plaque d'immatriculation) pour les services compétents (police, douane, gendarmerie).

#### 1.1.4.3 Applications de sécurité du trafic routier

Ils visent à améliorer la sécurité des passagers sur les routes en avisant les véhicules de toute situation dangereuse. Ces applications se basent en général sur une diffusion, périodique ou non, de messages informatifs permettant aux conducteurs d'avoir une connaissance de l'état de la route et des véhicules voisins. Des exemples répandus de services dans cette catégorie d'applications sont, l'avertissement des collisions, les avertissements sur les conditions de la route, l'assistance dépassement et changement de voie, etc.

### 1.1.5 Architectures des réseaux véhiculaires

Dans les réseaux VANETs, on trouve principalement, les entités fixes qui constituent l'infrastructure (RSU et TA) et les entités mobiles (les véhicules). Pour pouvoir échanger les différentes informations et données liées à la sécurité et au confort des usagers de la route, ces différentes entités doivent établir des communications entre elles. Pour cette raison, on distingue trois types de communications véhicule à véhicule (V2V) et véhicule à infrastructure (V2I) et Hybride [4].

#### 1.1.5.1 Communications de Véhicule à Véhicule (V2V)

Dans cette approche, un réseau de véhicules est vu comme un cas particulier de réseaux mobiles ad hoc MANET (Mobile Ad hoc Network) où les contraintes d'énergie, de mémoire et de capacité de calcul sont relaxées, et où le modèle de mobilité n'est pas aléatoire, mais prévisible (sous-couche du réseau routier) avec une très forte mobilité. Cette architecture peut être utilisée dans les scénarios de diffusion d'alerte (freinage d'urgence, collision, ralentissement, etc...) ou pour la conduite coopérative. En effet, dans le cadre de ces applications de sécurité routière, les réseaux à infrastructure montrent leurs limites, surtout en termes de délais. Et il est clair qu'une communication ad hoc multi-sauts est plus performante qu'une communication passant par un réseau d'opérateur [4].

#### 1.1.5.2 Communications de Véhicule à Infrastructure (V2I)

L'architecture (V2I) est composée de RSU, auxquels les véhicules accèdent pour les applications de sécurité, de gestion et de confort. Les RSU sont administrés par un ou plusieurs organismes publics ou bien par des opérateurs autoroutiers. Un véhicule qui informe le service de voirie au sujet d'un obstacle est un exemple de communication V2I. Dans cet exemple, la communication est unidirectionnelle, d'OBV vers le RSU.

Nous parlons d'I2V dans le cas de communication Infrastructure-vers-Véhicule. Un panneau de signalisation équipé d'un RSU qui envoie une information aux véhicules passant à proximité est un exemple de communication I2V. Dans la suite, par V2I, nous englobons toutes les communications Véhicule-Infrastructure, quelle que soit la direction du trafic de données [4].

### 1.1.5.3 Communication hybride

La combinaison de ces deux types d'architecture de communication permet d'obtenir une architecture hybride intéressante. En effet, les portées des infrastructures étant limitées, l'utilisation de véhicules comme relais permet d'étendre cette distance. Dans un but économique et en évitant de multiplier les bornes à chaque coin de rue, l'utilisation de sauts par véhicules intermédiaires prend toute son importance. Néanmoins, les communications inter-véhiculaires souffrent de problèmes de routage lors de transmission longue distance. Dans de telles situations, l'accès à une infrastructure peut améliorer les performances réseau. Nous comprenons donc la complémentarité des deux types de communication et l'intérêt d'une architecture hybride [6].

Un cas particulier de l'architecture hybride est le réseau VSN (*Vehicular Sensor Network*). En effet, ce type de réseau émerge en tant que nouvelle architecture de réseaux de véhicules. Le VSN a pour objectif la collecte et la diffusion proactive en temps réel des données relatives à l'environnement dans lequel évoluent les véhicules, et ce, plus particulièrement en zone urbaine. En effet, les voitures sont munies de plus en plus de capteurs de toutes catégories (cameras, capteurs de pollution, capteurs de pluie, capteurs d'état des pneumatiques, ESP, ABS, géolocalisation satellite, etc.). Les informations délivrées par ces équipements peuvent être utiles pour l'obtention d'états sur le trafic routier (embouteillages, ralentissements, vitesse moyenne du trafic, etc.), sur les places de parking disponibles, pour des informations plus générales telles que la consommation moyenne de carburant et le taux de pollution, ou encore pour des applications de surveillance (grâce aux caméras embarquées sur des véhicules).

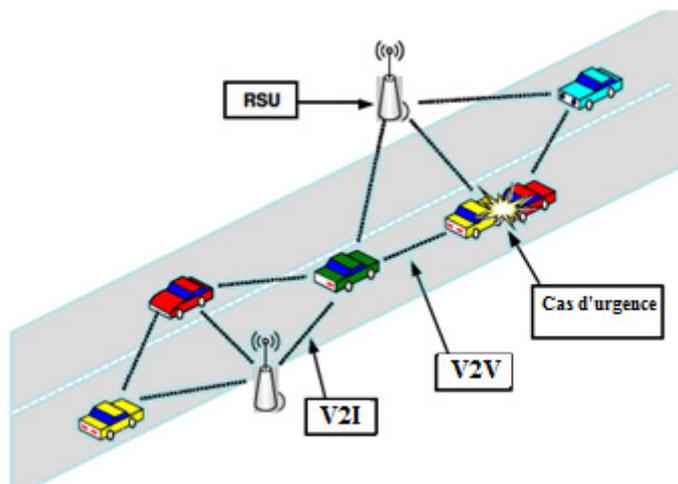


FIGURE 1.3 – Types de communication dans un réseau de véhicules. [7]

## 1.1.6 Les techniques de communications

En distingue deux types de communications :

### 1.1.6.1 Les communications radio

Un système de communication comprend tous les éléments capables de véhiculer de l'information (son, données informatiques, vidéo..) d'une source vers une ou plusieurs destinations. Depuis la naissance des réseaux sans fil, les communications sont passées de la communication analogique filaire vers une communication numérique sans fil. De nombreuses technologies de communication peuvent être utilisées pour assurer l'échange de données entre véhicules, ces techniques en liaison décrites avec les communications radio. [8]

**Le système RDS (*Radio Data System*) :** cette technique est basée sur un système de diffusion de données par la radio, elle permet de diffuser en parallèle un flux permanent d'informations numériques avec les émissions des radios FM.

Nous pouvons citer comme exemple : TA (*Traffic Announcement*) diffuse des informations routières. TP (*Traffic Program*) désigne une station offrant le radioguidage. Le média RDS-TMC (*Radio Data System - Traffic Message Channel*). Les véhicules équipés de récepteurs RDS-TMC peuvent recevoir des messages à raison de 20 par minute. Ce mode de communication est adapté pour les liaisons à moyenne distance (de 10 à 100 km) [9].

**Le système DAB (*Digital Audio Broadcasting*) :** c'est un système de transmission du son numérique, Deux bandes de fréquence sont allouées au DAB terrestre : la bande III dans l'intervalle de fréquence 174 - 230 MHz et la bande utilisée en France qui se situe entre 1452 MHz et 1467.5 MHz. En plus du son à qualité numérique, on peut diffuser par le média DAB des services de données textuelles ou graphiques. Il s'agit soit de données insérées dans la composante audio elle-même et qui sont appelés PAD (*Program Associated Data*) soit de données diffusées par un canal indépendant de l'audio et ce sont les NPAD (*No Program Associated Data*) [6].

**Le système DVB(*Digital Vidéo Broadcasting*) :** Les spécifications de la norme du DVB définissent un ensemble de moyens permettant la diffusion de tous types de données, accompagnées d'informations les concernant sur tous types de médias [6].

#### 1.1.6.2 Les communications sans fil [6]

Les technologies de communications sans fil sont en plein développement. Parmi les technologies utilisées dans les communications inter-véhicules, nous pouvons citer :

- Les systèmes qui réutilisent des infrastructures existantes, systèmes cellulaires de 2G vers 3G et d'autres évolutions (GSM, GPRS, 3GPP...).
- Les réseaux locaux sans fil (WLAN) qui sont principalement constitués des normes IEEE 802.11 (Wifi), IEEE 802.16 (Wi-MAX) et 802.11p (DSRC).

La communication sans fil dans l'environnement véhiculaire est basée sur deux entités d'équipements : le premier est un dispositif installé sur des sections critiques autour de l'infrastructure (stop, intersections...), et le deuxième est celui qu'on trouve embarqué sur le véhicule. [10].

#### 1.1.7 Types de message

Les entités formant un réseau sans fil véhiculaire vont générer et s'échanger des messages. En fonction de l'application et du contexte environnemental, un véhicule peut envoyer (ou recevoir) un message de contrôle, d'alerte ou d'autre. [5]

### 1.1.7.1 Message de contrôle

Le message de contrôle est généré à intervalle régulier. Conventionnellement, chaque véhicule émet un message de contrôle toutes les 100 ms. Ce message, appelé aussi < beacon >, contient la position, la vitesse, la direction et l'itinéraire du véhicule émetteur. Grâce aux messages de contrôle, chaque véhicule se crée une vue locale de son voisinage. Le véhicule peut aussi prédire et anticiper des situations accidentogènes ou de congestion. Le message de contrôle est l'équivalent du message HELLO des protocoles de routage. Chaque véhicule se fait donc connaître de son voisinage direct. Bien entendu, les messages de contrôle ne sont pas transfères et utilisent une diffusion a un saut.

### 1.1.7.2 Message d'alerte

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

### 1.1.7.3 Autres messages

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

## 1.2 Normes et standardisation dans les VANET

L'utilisation des normes et des standards permet de simplifier le développement des produits, réduire les coûts, et permet aux utilisateurs de comparer les produits de concurrence. On

trouve une multitude de norme qui se rapporte à l'accès sans fil dans les environnements véhiculaires. Ces normes s'étendent des protocoles qui s'appliquent à l'équipement de transmission et protocoles de transmission en passant par les spécifications de sécurité, l'acheminement des paquets, et les protocoles d'interopérabilité.

### 1.2.1 DSRC (Dedicated Short Range Communications) [11]

Les premiers standards définis pour les communications sans fil dans les STI utilisent la bande de fréquence de 915MHz essentiellement pour assurer des services tels que, le péage électronique, l'accréditation et la surveillance des opérations des véhicules commerciaux. Cette bande de fréquence étant trop étroite et polluée pour supporter l'évolution envisagée pour les applications dans les réseaux véhiculaires, l'ITSA a sollicité la FCC (Federal Communications Commission) pour l'allocation d'une bande passante de 75MHz dans la gamme de fréquences 5,850-5,925GHz pour les communications à courte portée dédiées aux STI aux USA. Cette demande a été accordée par la FCC en 1999 et a donné naissance à la technologie DSRC.

En Europe, l'ETSI (*European Telecommunications Standards Institute*) a créé un comité technique pour les STI en 2007 dont la mission est la création de normes et de spécifications pour l'utilisation des technologies de l'information et de la communication dans les futurs systèmes de transport en Europe. Le processus d'attribution des fréquences en Europe est beaucoup plus complexe et plus fastidieux qu'aux USA étant donné que tous les pays européens et leurs autorités nationales sont concernés. Ce n'est qu'en août 2008 que l'ETSI a affecté un spectre de fréquences dans la bande des 5,9GHz pour l'accès sans fil dans les réseaux véhiculaires.

Du point de vue technique, l'approche européenne pour les communications des véhicules a de nombreux points communs avec le système américain. Les deux se fondent sur la variante 802.11p de la norme IEEE 802.11 avec une bande de fréquences similaires et utilisent principalement des messages périodiques envoyés par chaque véhicule et infrastructure pour les services de sécurité et IPv6 pour les services de données, la bande passante est divisée en sept canaux de 10MHz. Les canaux se répartissent fonctionnellement en un canal de contrôle et six canaux de service. Le canal de contrôle est réservé à la transmission des messages de gestion

du réseau tel que les messages d'annonce de services et les messages de très haute priorité comme certains messages critiques liés à la sécurité routière. Les six autres canaux sont dédiés à la transmission des données des différents services annoncés sur le canal de contrôle.

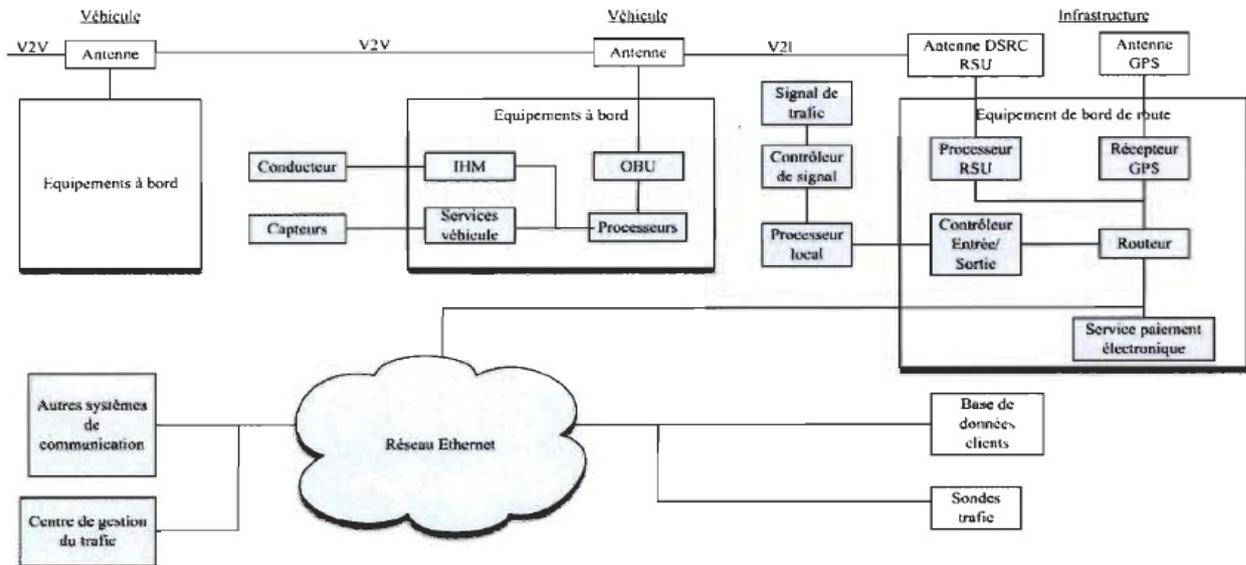


FIGURE 1.4 – Exemple d'architecture réseau de DSRC. [12]

### 1.2.2 La norme IEEE 802.11p

La norme IEEE 802.11p est un amendement du standard IEEE802.11 que le groupe de travail IEEE (TGP, task group protocol) a commencé à développer en 2004 pour l'accès sans fil dans les systèmes de transport intelligents. Il définit les spécifications des couches MAC et PHY dans le cadre des réseaux véhiculaires. [11]

**La couche physique** du 802.11p utilise les mêmes mécanismes de traitement de signal et les mêmes spécifications que dans le standard 802.11a avec cependant quelques modifications pour l'adapter aux environnements véhiculaires. Pour offrir des communications à grandes portées, quatre classes de puissance maximale EIRP (*Effective Isotropic Radiated Power*) sont autorisées. La plus grande valeur,  $44,8dBm$  ( $30w$ ), est réservée pour les véhicules d'urgence (*approching emergency vehicles*). La valeur typique des messages de sécurité pertinents est de  $33dBm$ . Pour augmenter la tolérance à l'effet de propagation des signaux par trajets multiples, une bande passante de  $10MHz$  est utilisée au lieu de  $20MHz$  comme dans la norme 802.11a.

En réduisant la bande passante, tous les paramètres du domaine temporel sont doublés. Ceci permet de réduire d'une part l'effet Doppler grâce à l'utilisation d'une bande passante plus petite et d'autre part, les interférences inter symboles en doublant la valeur des intervalles de garde. Ces modifications permettent à la norme 802.11p d'offrir des débits allant de 3 à 27Mbit/s sur des portées de communications de 300m à 1000m.

**La couche MAC** de la norme 802.11p est équivalente à la technique EDCA (*Enhanced Distributed Channel Access*) de la norme 802.11e. Dans EDCA, les messages sont classifiés en quatre catégories d'accès (AC, Access Category) avec, AC0 la catégorie de messages ayant la plus faible priorité et AC3 la catégorie de ceux ayant la plus grande priorité. A chaque catégorie est associée une file d'attente où sont gardés les paquets en attente d'envoi. La priorité est assurée en affectant différents paramètres d'accès à chaque catégorie.

### 1.2.3 WAVE (Wireless Access in Vehicular Environments) [11]

L'IEEE a développé une architecture connue sous le nom de WAVE (*Wireless Access in Vehicular Environments*), pour fournir l'accès sans fil dans les environnements véhiculaires. Deux modes de communication sont possibles dans l'architecture WAVE, véhicule-à-véhicule et véhicule-à-infrastructure. Les véhicules communiquent via un dispositif installé à leur bord dit, OBU montré sur la figure 1.5. Les infrastructures, sont équipées de dispositif dit RSU montré sur la figure 1.6. Les RSU sont des entités fixes qui permettent de connecter les véhicules aux réseaux communs qui par la suite les connectent au cœur central du réseau. Les RSU sont habituellement installés sur des infrastructures existantes tels que les feux de circulation, les panneaux routiers ou encore les lampadaires. En plus des capteurs, les RSU disposent d'un émetteur-récepteur DSRC et d'un processeur d'applications qui offre des services de sécurité et des services non liés à la sécurité pour les multiples OBU qui sont dans la zone de transmission. Un OBU est un équipement WAVE mobile qui permet des communications OBU à OBU en mode ad hoc, et des communications OBU à RSU en mode infrastructure. Les OBU sont également reliés à une gamme de capteurs et d'actionneurs au sein du véhicule. Ceci facilite la surveillance efficace des véhicules pour rassembler des informations comme la vitesse du véhicule et son accélération.

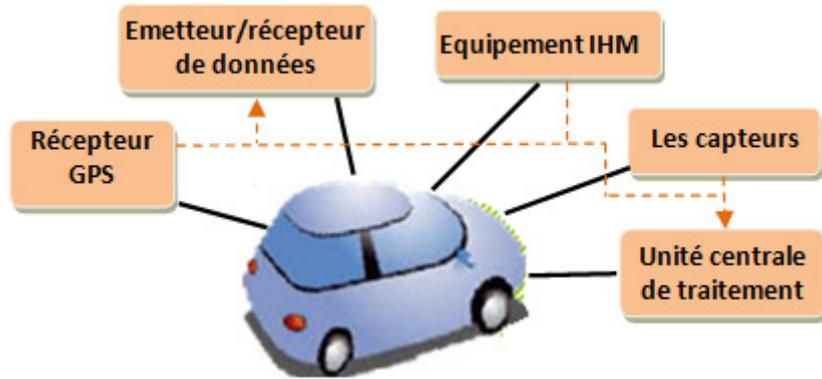


FIGURE 1.5 – WAVE On Board Unit. [12]



FIGURE 1.6 – WAVE RoadSide Unit. [12]

La figure 1.7 montre l'architecture WAVE qui est une association de l'amendement IEEE 802.11p et de quatre standards 1609.1, 1609.2, 1609.3, et 1609.4 définis par le groupe de travail IEEE 1609 pour décrire les spécifications des couches hautes pour les communications WAVE [11] :

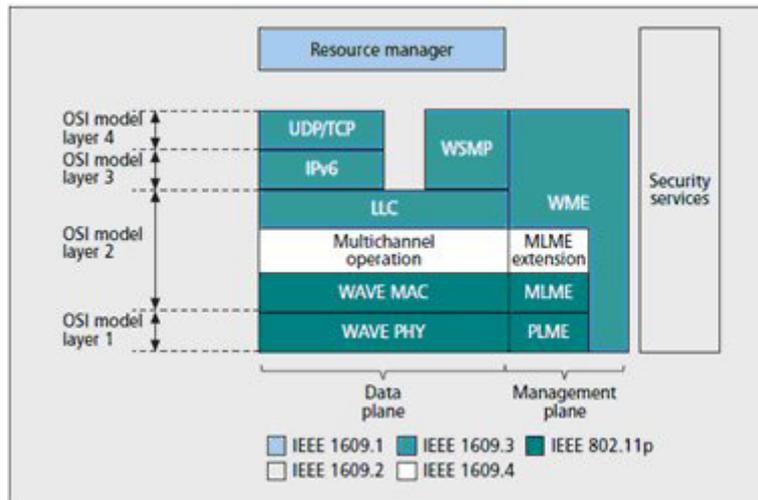


FIGURE 1.7 – Architecture protocolaire du système WAVE. [13]

**IEEE 1609.1** (*Gestionnaire des ressources*) ce standard décrit le service de gestion de ressources dit RM (Resource Manager) conçu pour permettre aux applications distantes de communiquer avec les OBU via les RSU. Le RM a pour objectif de répondre aux exigences des applications distantes en leur fournissant un accès en temps opportun aux ressources du OBU telles que la mémoire et l'interface utilisateur de manière cohérente et en garantissant l'interopérabilité .

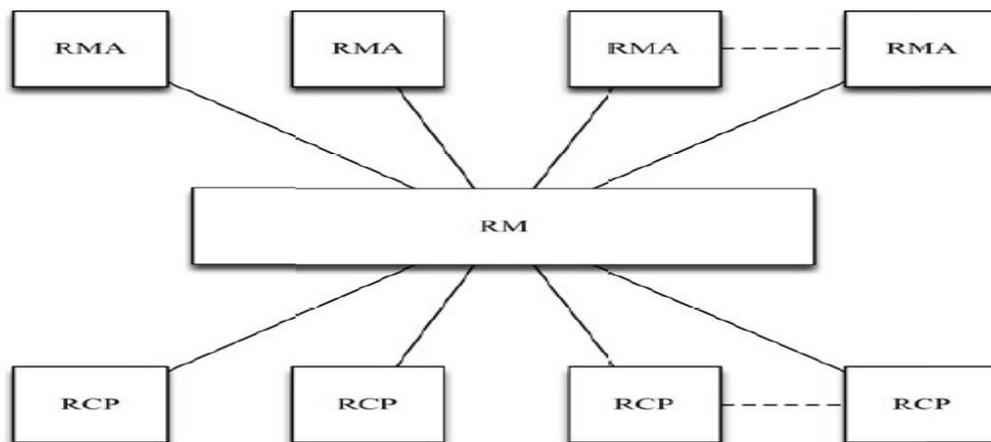


FIGURE 1.8 – Modules du standard IEEE 1609.1. [14]

**IEEE 1609.2** (*Services de sécurité pour les applications et les messages de gestion*) décrit les services de sécurité dans le système WAVE. Les principales applications du système étant des

applications de sécurité critiques, il est vital de définir des services de sécurité afin de protéger les messages contre les attaques telles que l'écoute clandestine, l'usurpation d'identité, et aussi de préserver la vie privée des conducteurs. Ce standard définit le format des paquets et les fonctions de sécurité, de chiffrement et d'authentification, pour les trois types de messages, de sécurité, de données et de gestion.

**IEEE 1609.3** (*Services de la couche réseau*) ce standard décrit les fonctions des couches réseau et transport pour les communications dans un système WAVE dont l'adressage et le routage. Il définit un nouveau type de messages dits WSM (*WAVE Short Messages*) et un nouveau protocole WSMP (*WAVE Short Messages Protocol*) pour la transmission des WSM. Le WSMP est une alternative à IPv6 qui fournit aux applications un échange de données efficace en leur permettant d'envoyer les WSM directement sur n'importe quel canal de DSRC.

**IEEE 1609.4** (*Opération multi-canal*) étant basés sur le DSRC, les dispositifs WAVE doivent fournir un accès multi-canal et permettre des communications sur le canal de contrôle et les canaux de service. C'est le rôle du standard 1609.4 qui définit tous les mécanismes nécessaires pour l'accès avec priorité aux canaux, la coordination et le routage des données vers les canaux et la transmission des données.

## Conclusion

Dans ce chapitre, nous avons présenté les réseaux véhiculaires VANETs qui ne sont qu'une particularité des réseaux MANETs. Nous avons signalé également leur caractéristique, leur application. Ainsi la communication dans les réseaux véhiculaires avec chacun de ces modes, Le chapitre suivant sera consacré pour le routage dans les réseaux véhiculaire VANET.

# 2

Protocoles de routage dans les VANETs

## Introduction

Le routage dans les réseaux VANET est un problème très difficile qui pose des défis pour de nombreux chercheurs. Alors pour que les véhicules puissent communiquer entre eux, nous devons définir le protocole de routage. En fait, quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules.

Dans ce chapitre, nous présentons la problématique du routage dans les réseaux de véhicules et nous décrivons les différentes solutions pour le routage dans VANET et nous illustrons les problèmes liés à ces protocoles.

### 2.1 Routage dans ad hoc

Les réseaux ad hoc se caractérisent par une absence d'infrastructure et de gestion centralisée. Dans ce type de réseaux, chaque élément peut bien évidemment émettre et recevoir des messages, mais assure également un rôle de relais de l'information afin que les messages circulent dans le réseau de proche en proche. Chaque nœud du réseau doit donc posséder des capacités de routage, c'est le routage dit ad hoc. Grâce à ce routage, la portée radio d'un nœud peut être virtuellement étendue en utilisant ses voisins comme relais de l'information.

Le principal défi des réseaux ad hoc est la conception de protocoles de routage à la fois robustes (tolérants aux pannes, adaptatifs à la mobilité des nœuds) et permettant le passage à l'échelle. Cette propriété permet au protocole de conserver des performances raisonnables lorsque la taille du réseau ou le nombre de nœuds augmente : limite de *l'over-head* généré, délai d'acheminement, longueur des chemins, rupture des chemins. [15]

**Le routage** est en quelque sorte le mécanisme clé des réseaux ad hoc. C'est grâce au mécanisme de routage que les stations formant le réseau vont pouvoir communiquer, même si elles ne sont pas à portée directe de communication. Il est donc très important d'avoir un protocole de routage efficace si on veut pouvoir tirer parti du potentiel des réseaux ad hoc. [16]

## **2.2 Routage dans les VANETs**

Le routage joue un rôle très important dans les VANET puisque tous les services supportés, unicast ou multicast, se basent sur des communications multi-saut pour l'acheminement des données. Les transferts de fichiers et les jeux. Les communications multicast sont utilisées dans les applications de sécurité et de gestion de trafic telles que l'avertissement de collision et le platooning. Pour réaliser les échanges, les protocoles de routage utilisent des informations locales, sur le voisinage immédiat, ou globales, concernant tout le réseau, an de déterminer les nœuds relais qui participent à l'acheminement des données communications unicast sont généralement utilisées dans les applications de confort. [11]

## **2.3 Classification des protocoles de routage dans les réseaux VANET**

Les réseaux véhiculaires ont comme caractéristique principale une forte mobilité qui entraîne une topologie très dynamique. Cette caractéristique fait que les protocoles de routage traditionnels des MANETS sont pour la plupart adaptée aux VANETS. En effet, dans les VANETS, la vitesse peut être beaucoup plus élevée que les MANETS dans certains environnements de communication comme les autoroutes. Dans [17] [18] Différentes solutions pour le routage dans les réseaux VANET qui ont été proposées, nous distinguons deux classes de protocoles de routage : (voire figure 2.1) les protocoles basés sur la topologie qui sont divisés en protocoles proactifs, réactifs et hybrides et les protocoles basés sur la localisation (géographique) qui utilisent la position physique des nœuds mobiles pour configurer le routage.

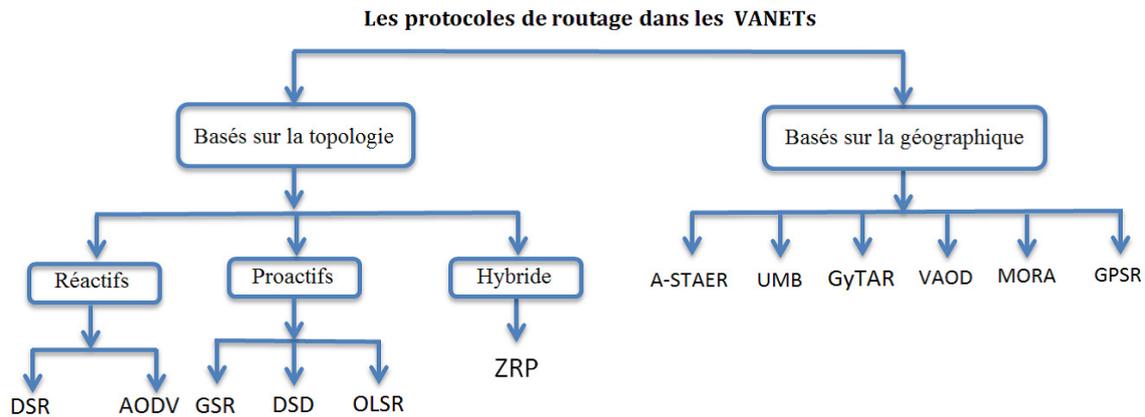


FIGURE 2.1 – Les protocoles de routage dans les réseaux VANETS.

## 2.3.1 Les protocoles de routage basés sur la topologie

### 2.3.1.1 Les protocoles réactifs

Les protocoles réactifs adoptent des algorithmes classiques tels que le routage par vecteur de distance. Les routes sont établies uniquement sur demande et seules les routes en cours d'utilisation sont maintenues. Dans ce cas, un délai supplémentaire est nécessaire au début de chaque session pour la recherche du chemin. Lorsqu'un nœud veut envoyer des paquets, une étape de découverte de route est initiée par la diffusion d'un message de recherche de route. Tout nœud qui reçoit ce message et qui ne dispose pas d'informations à propos de la destination diffuse à son tour le message. Ce mécanisme est appelé mécanisme d'inondation. [17]

#### A. Le protocole AODV

Le protocole de routage AODV (*Ad hoc On-demand Distance Vector*) [19] est un protocole décrit dans la [20]. Ce protocole crée les routes au besoin et utilise le principe de numéro de séquence afin d'utiliser les routes les plus nouvelles, dites encore les plus fraîches. En plus, il utilise le nombre de sauts comme métrique pour choisir entre plusieurs routes disponibles. Trois types de paquets sont utilisés par AODV : les paquets de requête de route RREQ (*Route Request Message*), les paquets de réponse de route RREP (*Route Reply Message*) et les paquets d'erreur de route RERR (*Route Error Message*). En plus de ces paquets, AODV invoque des

paquets de contrôle HELLO qui permettent de vérifier la connectivité des routes. AODV repose sur deux mécanismes : découverte de route et maintenance de route. La découverte de route permet de trouver une route pour atteindre une destination et la maintenance de route permet de détecter et signaler les coupures de routes provoquées éventuellement par la mobilité des nœuds.

## B. Le protocole DSR

Le protocole de routage DSR (*Dynamic Source Routing*) [19] est un protocole qui est normalisé dans la [21]. Ce protocole crée les routes à la demande comme le protocole AODV. Il utilise la technique "routage à la source" dans laquelle la source inclut dans l'entête du paquet la route complète par laquelle un paquet doit passer pour atteindre sa destination. Les nœuds intermédiaires entre la source et la destination n'ont pas besoin de maintenir à jour les informations sur la route traversée puisque la route complète est insérée dans l'entête du paquet. DSR est composé de deux mécanismes : la découverte de route et la maintenance de route. Le premier permet de chercher les routes nécessaires à la demande, tandis que le second permet de s'assurer de la maintenance des routes tout au long de leur utilisation.

- **Découverte de la route** [21] [22]

La procédure est la diffusion d'une requête de route et l'émission d'une réponse. Lorsqu'un nœud source S souhaite envoyer des données à un nœud destination D et n'a pas de route vers ce nœud D, Le nœud S envoie un paquet *Route Request* (*requête de recherche de route*) à destination du nœud D, ce paquet se propage dans le réseau (voir figure 2.2), cette propagation se termine lorsque le nœud D ou un nœud possédant un chemin vers celui-ci dans son cache est atteint. Le paquet contient l'adresse source, l'adresse de destination, un numéro d'identification et un champ *Route Record* dans lequel sera enregistrée la séquence des nœuds visités durant l'inondation du paquet *Route Request* dans le réseau.

Quand un nœud reçoit un paquet *Route Request*, il vérifie s'il connaît un chemin vers la destination, si ce n'est pas le cas, il ajoute son adresse dans le champ *Route Record* du paquet *Route Request* et transmet ce paquet à ses voisins. Pour éviter les boucles et les multiplications des paquets *Route Request*, ce transfert ne se fait que si l'adresse du nœud n'apparaît pas déjà

dans le champ Route Record.

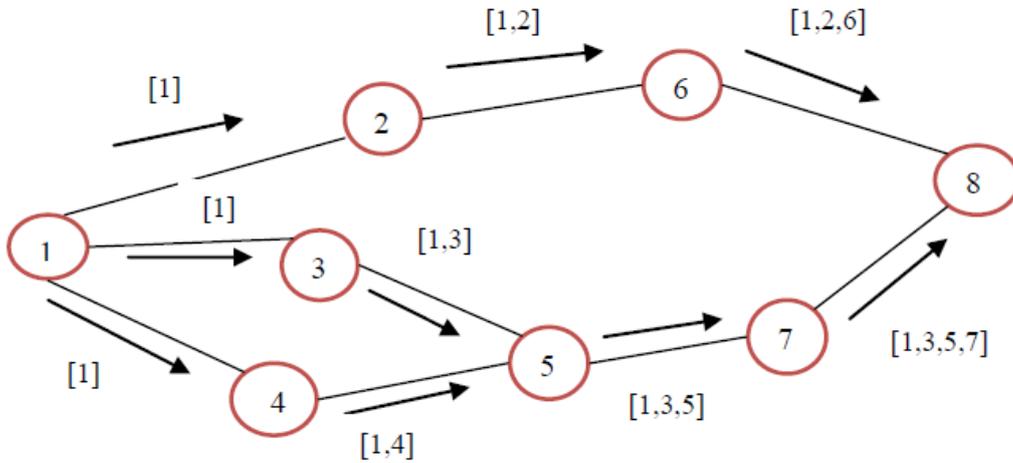


FIGURE 2.2 – Construction de l’enregistrement de route dans DSR.

Lorsque le paquet *Route request* atteint la destination (ou bien une station intermédiaire ayant une route vers la destination), la destination (ou le nœud intermédiaire) envoie un paquet de réponse *Route Reply* (voir figure 2.3) via le chemin donné dans le Route Record si les liaisons sont bidirectionnelles (symétriques) ou via un autre chemin (utilisant éventuellement une découverte de la route). Afin de diminuer le coût de la recherche de route, chaque nœud peut garder en mémoire les routes qu’il a apprises dans le cache de route.

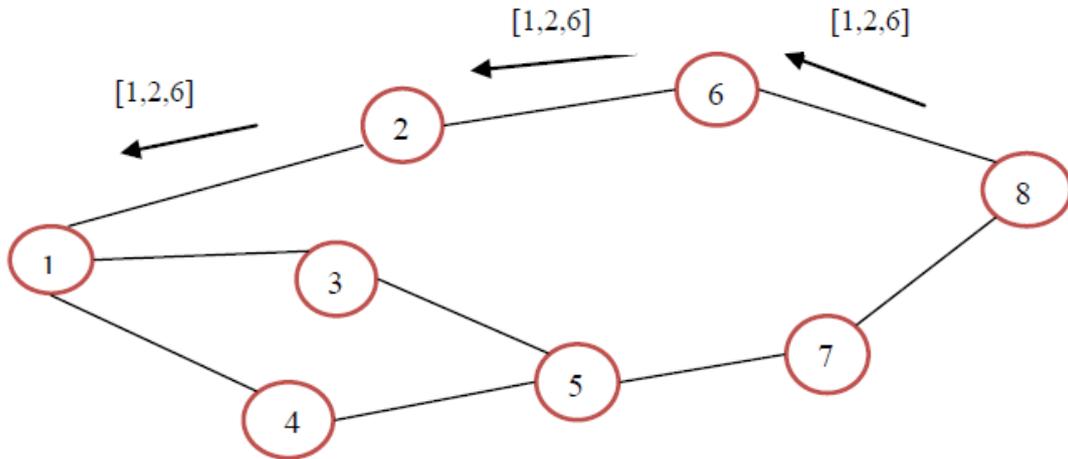


FIGURE 2.3 – L’envoi du chemin ou de la route replay (RREP).

- **Maintenance de route**

Lorsqu’un nœud transmet un paquet, il est responsable de confirmer sa bonne réception par son prochain saut vers la destination tout au long de la route source. Par exemple, dans la figure 2.4 le nœud (A) a transmis un paquet au nœud (F) en utilisant les nœuds (B), (C), (D) et (E) comme des nœuds intermédiaires. Dans ce cas le nœud (A) est responsable de la réception du paquet par (B) qui est lui-même responsable de sa réception par (C). La mobilité des nœuds dans les réseaux Ad Hoc nécessite de vérifier, après l’envoi d’une donnée, que la topologie est toujours la même et que la source peut utiliser une source pour atteindre la destination en utilisant une procédure de maintenance de route.

Quand un nœud détecte une erreur de transmission, un paquet route error (erreur de route) contenant l’adresse du nœud qui a détecté l’erreur et celle du nœud qui le suit dans le chemin, est envoyé à l’émetteur original du paquet. Lors de la réception de ce paquet par la source, le nœud concerné par l’erreur est supprimé du chemin sauvegardé, et tous les chemins qui contiennent ce nœud sont tronqués à ce point-là. Par la suite une nouvelle opération de découverte de routes vers la destination est initiée par l’émetteur.

Reprenons la figure 2.4 le nœud (D) est incapable d'envoyer le paquet au prochain saut (E) (à cause de la coupure des liens entre D et E), alors il retourne un paquet erreur de route (*Route Error*) à (A) en signalant une coupure de lien entre lui et (E).

Quand le nœud source (A) et les nœuds intermédiaires (B) et (C) reçoivent ce paquet, ils suppriment la route de leur cache. Si (A) à une autre route vers (E) dans son cache, il peut l'utiliser pour envoyer le paquet immédiatement, sinon il initie une nouvelle requête de route.

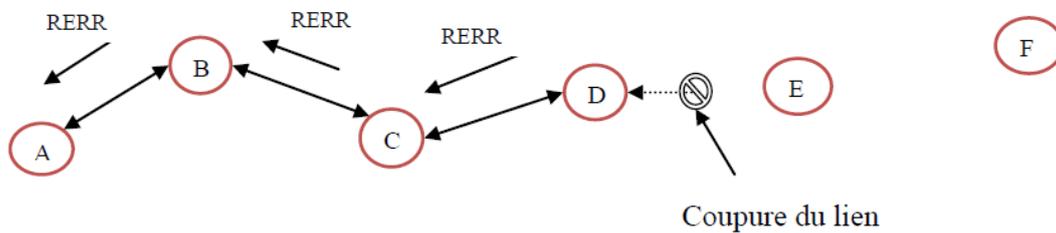


FIGURE 2.4 – Erreur dans DSR : envoi de Route Error (RERR).

### 2.3.1.2 Les protocoles proactifs

Dans les protocoles proactifs, chaque nœud garde une image de la topologie de tout le réseau. Cette image est mise à jour, périodiquement ou à chaque modification topologique, par un échange de messages de contrôle. Les routes sont déterminées sur la base de cette image. [11]

#### A. Le protocole OLSR

Le protocole de routage OLSR (*Optimized Link State Routing*) [23] [11] [24] est un protocole de routage proactif développé dans le cadre du projet Hypercom de l'Institut National de la Recherche en Informatique et Automatique (INRIA) de France et proposé en tant que RFC (Request For Comment) expérimentale à l'IETF (*Internet Engineering Task Force*). Il est considéré comme une optimisation du protocole à état des liens filaires pour les réseaux mobiles Ad Hoc. Il a pour objectif de fournir des routes de plus court chemin vers une destination en termes de nombre de sauts en utilisant l'algorithme de Dijkstra. Son innovation réside dans sa façon d'économiser les ressources radio lors des diffusions, ceci est réalisé grâce

à l'utilisation de la technique des relais multipoints (MPR : *Multi-Point Relaying*), donc le principe est que chaque nœud construit un sous-ensemble appelé MPR, parmi ses voisins, qui permet d'atteindre tous ses voisins à deux sauts, les nœuds de cet ensemble servent à acheminer et retransmettre les messages qu'ils reçoivent. Les voisins d'un nœud qui ne sont pas MPRs, lisent et traitent les paquets mais ne les retransmettent pas.

## B. Le protocole DSDV

Le protocole de routage DSDV (*Destination-Sequenced Distance-Vector*) [19] [25] est un protocole de routage de type vecteur de distance. Chaque nœud maintient une table de routage contenant des informations sur les destinations accessibles dans le réseau. Ces informations comprennent le nœud suivant utilisé pour atteindre la destination, le nombre de sauts qui sépare le nœud de la destination et le numéro de séquence estampillé par la destinataire. Ce numéro de séquence permet de distinguer les nouvelles routes des anciennes. Chaque nœud envoie périodiquement à ses voisins la totalité de sa table de routage. D'autres paquets de mise à jour sont aussi envoyés à la suite d'un changement dans la topologie du réseau. Ces paquets n'incluent que les entrées de la table affectées par le changement et ont pour objectif de propager les informations de routage aussi rapidement que possible. Quand un nœud reçoit un paquet de mise à jour, il le compare avec les informations existantes dans sa table de routage. Toute entrée dans la table est mise à jour si l'information reçue est plus récente (ayant un numéro de séquence plus grand), ou si elles ont le même numéro de séquence mais avec une distance plus courte.

Dans le protocole DSDV, une unité mobile doit attendre jusqu'à ce qu'elle reçoive la prochaine mise à jour initiée par la destination afin de mettre à jour l'entrée associée à cette destination dans la table de distance. De ce fait, la réaction de DSDV aux changements de la topologie est considérée lente. D'autre part, ce protocole cause une charge de contrôle importante dans le réseau à cause des paquets de mise à jour envoyé périodiquement ou à la suite des événements.

## C. Le protocole GSR

Le protocole GSR (*Global State Routing*) [19] est un protocole proactif à état de liens où chaque nœud connaît la topologie globale du réseau ce qui lui permet de calculer les routes pour atteindre chaque destination. GSR diffère des protocoles à état de liens dans le fait que les nœuds ne diffusent pas leurs états de liens à tout le réseau, mais ils se limitent à l'envoyer aux voisins uniquement. Ainsi, GSR réduit le trafic des paquets de contrôle. Le problème de GSR est la taille de ses paquets de mise à jour (table de topologie) qui peuvent devenir considérable si le réseau contient un grand nombre de nœuds. En plus, il a une lenteur dans la détection des changements de la topologie.

### 2.3.1.3 Protocoles hybrides

#### A. Le protocole ZRP

Le protocole de routage ZRP (*Zone Routing Protocol*) [19] [26] est un protocole hybride qui combine les deux approches proactives et réactive. Le protocole ZRP divise le réseau en différentes zones. Pour chaque nœud, il définit une zone de routage exprimée en nombre de sauts maximal  $\alpha$ . Ainsi, la zone de routage d'un nœud inclut tous les nœuds qui sont à une distance au maximum de  $\alpha$  sauts. Les nœuds qui sont exactement à  $\alpha$  sauts sont appelés nœuds périphériques.

À l'intérieur de cette zone, ZRP utilise un protocole proactif et à l'extérieur de cette zone de routage, il fait appel à un protocole réactif.

Le protocole proactif est IARP (*IntraZone Routing Protocol*) [27] et celui réactif est IERP (*IntErzone Routing Protocol*) [28] et Chaque nœud doit tout d'abord connaître ses voisins. Pour cela, ZRP utilise soit le protocole de contrôle d'accès au support (MAC) pour connaître les voisins immédiats ou le protocole NDP (*Neighbour Discovery Protocol*) pour la transmission et la gestion des échanges de messages HELLO. Par la suite, chaque nœud invoque le protocole IARP pour découvrir les routes vers tous les autres nœuds qui se trouvent dans sa zone de routage. Cependant, le protocole IERP est utilisé à la demande pour chercher les routes entre un nœud et une destination qui se trouvent à l'extérieur de sa zone de routage. Un troisième protocole BRP (*Bordercast Resolution Protocol*) [29] est inclus avec IERP pour guider la

propagation des requêtes de recherche de route dans le réseau. BRP utilise les données de la topologie fournies par le protocole IARP afin de construire sa liste des nœuds de périphérie et la façon de les atteindre.

#### 2.3.1.4 Comparaison entre les protocoles de routage proactive et réactive

Une comparaison entre les deux classes proactive et réactive est présentée dans le Tableau (2.1) : [30]

Dans un protocole de routage réactif, les mobiles ne conservent pratiquement aucune information sur la topologie globale du réseau. Seules sont stockées les informations sur les routes actives. Les routes sont construites à la demande et sont détruites lorsqu'elles ne sont plus utilisées.

Dans un protocole de routage proactif, la topologie du réseau est connue de tous les mobiles. Les routes sont disponibles immédiatement mais, en contrepartie, il faut diffuser régulièrement des informations sur les changements de topologie du réseau.

Les protocoles réactifs génèrent a priori un volume plus faible de signalisation mais en contrepartie engendrent un délai lors de la construction (ou de la reconstruction) des routes et produisent plus difficilement des routes optimales (quel que soit le critère).

Les protocoles proactifs disposent en permanence d'une route pour chaque destination dans le réseau mais génèrent en contrepartie un volume de signalisation important. De nombreux débats ont lieu sur la performance des deux approches.

Routage proactif		Routage réactif	
Avantages	inconvénients	Avantages	inconvénients
<p>La topologie du réseau est connue de tous les mobiles.</p> <p>Les routes sont disponibles immédiatement.</p>	<p>Il faut diffuser régulièrement des informations sur les changements de topologie du réseau.</p>	<p>Les mobiles ne conservent pratiquement aucune information sur la topologie globale du réseau : seules les informations sur les routes actives sont stockées.</p>	
<p>Les protocoles proactifs disposent en permanence d'une route pour chaque destination dans le réseau.</p>	<p>Un volume de signalisation important.</p>	<p>Les protocoles réactifs génèrent a priori un volume plus faible de signalisation.</p>	<p>Les protocoles réactifs engendrent un délai lors de la construction (ou de la reconstruction) des routes et produisent plus difficilement des routes optimales.</p>

TABLE 2.1 – Comparaison entre protocoles proactifs et protocoles réactifs [30]

L'avantage des protocoles hybrides est le fait qu'ils s'adaptent mieux aux réseaux de grandes tailles. Cependant, ce type de protocole cumule les inconvénients des protocoles proactifs et ceux des protocoles réactifs, tels que l'échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné.

### 2.3.2 Les protocoles de routage basés sur la géographique

Les protocoles de routage géographiques sont les plus adaptés pour les réseaux ad hoc de véhicules, puisque le mécanisme de routage se base sur les données géographiques des nœuds.

#### A. Le protocole A-STAR

Le protocole de routage A-STAR (*Anchor-based Street and Traffic Aware Routing*) [31] [24] est un protocole de routage basé sur la localisation (position) pour un environnement de communication véhiculaire métropolitain. Il utilise particulièrement les informations sur les itinéraires d'autobus de ville pour identifier une route d'ancre (anchor route) avec une connectivité élevée pour l'acheminement des paquets. A-STAR est semblable au protocole GSR, qui adopte une approche de routage basée sur l'ancrage (*anchor based*) qui tient compte des caractéristiques des rues.

Cependant, contrairement à GSR, il calcule les "*Anchor paths*" chemins d'ancrages en fonction du trafic (trafics de bus, véhicules, etc.).

Un point est associé à chaque rue en fonction de sa capacité (grande ou petite rue qui est desservie par un nombre de bus différent).

Les informations de routes fournies par les bus donnent une idée sur la charge du réseau véhiculaire dans chaque rue. Ce qui donne une image de la ville a des moments différents.

Le routage à base d'ancrage, le nœud source dans chaque paquet comprend un vecteur de voie constitué d'une liste d'ancres ou des points géographiques fixes, à travers lesquels les paquets doivent passer.

#### B. Le protocole UMB

Le protocole de routage UMB (*Urban Multi hop Broadcast Protocol*) [24] C'est un protocole efficace de la norme 802.11, basé sur l'algorithme de diffusion multi saut pour les réseaux inter-véhiculaires avec support d'infrastructure, dans le but de réduire les collisions et d'utiliser

efficacement la bande passante. Contrairement aux protocoles de diffusion par inondation, UMB confie les opérations d'envoi et de reconnaissance des paquets aux nœuds les plus éloignés sans connaître à priori des informations sur la topologie du réseau.

UMB est décomposé en deux phases : la première appelée diffusion directionnelle, où le véhicule source sélectionne un nœud dans la direction de diffusion pour faire un reliaje de données sans aucune information sur la topologie. La deuxième diffusion aux intersections pour disséminer les paquets dans toutes les directions, pour cela UMB utilise des répéteurs installés dans les intersections pour l'envoi des paquets vers tous les segments. On suppose que chaque véhicule est équipé par un récepteur GPS (*Global Position System*) et une carte routière électronique. Le principal avantage du protocole UMB est la fiabilité de diffusion multi-saut dans les canaux urbains [32].

### C. Le protocole GyTAR

GyTAR (*improved Greedy Traffic-Aware Routing protocol*) [33] [34] [35] est un protocole de routage géographique basé sur la localisation (position) et adapté aux réseaux véhiculaires capables de trouver des chemins robustes dans un environnement urbain. L'objectif de ce protocole est de router les données de proche en proche en considérant les différents facteurs spécifiques à ce genre d'environnements/réseaux. Ce protocole suppose que chaque véhicule connaît sa position courante et ceci grâce au GPS. De plus un nœud source est censé connaître la position du destinataire pour pouvoir prendre des décisions de routage, cette information est donnée par un service de localisation tel que GLS (*Grid Location Service*) et peut déterminer la position des intersections voisines à travers des cartes numériques.

### D. Le protocole VADD

Le protocole de routage VADD (*Vehicle-Assisted Data Delivery*) [31] [36] est un protocole de routage qui prend en considération le contexte des réseaux de véhicules et exploite le mouvement prévisible des véhicules pour décider de retransmettre ou non le message. Il utilise particulièrement les informations sur le trafic routier au niveau d'une route pour estimer le délai mis par un paquet pour parcourir un tel segment. Par conséquent, les paquets seront

achemines le long d'un chemin ayant le plus faible délai de bout en bout.

### E. Le protocole MORA

Le protocole de routage MORA (*MOvement-based Routing Algorithm*) propose dans [36] [37] exploite la position et la direction de mouvement de véhicules pour adapter les décisions de retransmission au contexte des véhicules et faire face ainsi à la forte mobilité des nœuds et au changement assez fréquent de la topologie.

### F. Le protocole GPSR

GPSR (*Greedy Perimeter Stateless Routing*) [38] est un protocole de routage réactif et efficace qui a été conçu et adapté pour les réseaux ad hoc mobiles et les réseaux de capteurs. Son modèle de fonctionnement suppose que tous les nœuds se trouvent au niveau d'un même plan. Du fait de la mobilité des nœuds, certains algorithmes de routage qui se basent sur la topologie du réseau, ou lance une phase de découverte de routes pour acheminer des données ne sont pas adaptés à GPSR. De ce fait, il utilise la position géographique des nœuds pour l'acheminement des paquets de données ou de contrôle.

Dans un réseau mobile, les nœuds sont susceptibles de se déplacer. Il faut ainsi un mécanisme permettant à chaque nœud de savoir la position de ses voisins. Afin de signaler leur présence et leur localisation, les nœuds inondent le réseau en envoyant un paquet de signalement (messages « beacon ») contenant la position et un identifiant (par exemple, son adresse IP). Nous proposons d'utiliser les messages « beacon » de contrôle pour renseigner les nœuds voisins sur les directions que peuvent assumer un nœud.

L'échange périodique de ces paquets de contrôle permet aux nœuds de construire leur table de position. La période d'émission des messages « beacon » dépend du taux de mobilité dans le réseau ainsi que de la portée radio des nœuds. En effet, lorsqu'un nœud ne reçoit pas de message « beacon » d'un voisin après un temps  $T$ , il considère que le voisin en question n'est plus dans sa zone de couverture et l'efface de sa table de position. Il faut donc adapter le temps d'émission des paquets de contrôle. Un des avantages du BP (*Beaconing Protocol*) est

que chaque nœud n'a besoin que des informations sur ses voisins directs, ce qui nécessite peu de mémoire.

Alternativement, le protocole GPSR permet au nœud d'en-capsuler sur quelques bits leur position dans les paquets de données qu'il envoie. Dans ce cas, toutes les interfaces des nœuds doivent être en mode promiscuité afin de recevoir les paquets s'ils se trouvent dans la zone de couverture de l'émetteur.

L'acheminement des paquets par GPSR se fait selon deux modes suivant la densité du réseau : le « *Greedy Forwarding* » et le « *Perimeter Forwarding* » (appelés respectivement GF et PF dans la suite).

## Conclusion

Dans ce chapitre, nous avons comparé différents protocoles pour montrer que la performance de chacune dépend énormément des conditions d'utilisation. Les protocoles construisant des routes par diffusion sont plus performants que leurs concurrents lorsque le nombre de routes actives est faible ou lorsque la mobilité dans le réseau est élevée. Ce pendant, l'environnement et la topologie complexe des réseaux véhiculaires, une mise en œuvre réelle peut être difficile aussi bien sur le plan économique que logistique. Pour contourner ce problème, la simulation est le moyen le plus largement utilisé.

Dans le chapitre qui suit nous allons présenter les différents simulateurs utilisés pour la simulation des VANETS.

# 3

Les simulateurs des VANETs

## Introduction

La mise en œuvre sur le terrain des réseaux VANETs est extrêmement coûteuse et difficile. Aussi, la simulation est une étape nécessaire dans cette thématique. Il existe plusieurs simulateurs de réseaux Ad Hoc capables d'effectuer des simulations répondant aux critères des VANETs. Nous présentons dans cette partie quelques simulateurs pouvant répondre à ceux-ci.

### 3.1 La Simulation dans un Réseau VANET

L'objectif de la simulation est de reproduire les mêmes résultats que ceux que l'on obtiendrait en conditions réelles. Un simulateur de réseau VANET réaliste se compose des éléments indispensables : un simulateur de réseau regroupant les bases <réseaux> d'une simulation VANET et un simulateur de mobilité.

Le simulateur de mobilité génère des fichiers qui décrivent la mobilité des nœuds dans un environnement, et qui sont transmis au simulateur de réseau Ad-Hoc. Bien que les deux simulateurs soient indispensables à la simulation des réseaux VANETs, il n'y a aucun lien direct entre eux. L'un des problèmes que l'on rencontre alors est l'interconnexion entre les simulateurs. Il existe cependant des simulateurs intégrés qui regroupent les deux simulateurs au sein d'une même entité pour répondre au mieux à ce problème.

Dans ce qui suit nous montrons les principes de ces deux éléments nécessaires à la simulation des VANETs, à savoir les simulateurs de réseau Ad Hoc et les simulateurs de mobilité [39].

### 3.2 Les Simulateurs de Réseaux Ad Hoc

Il n'y a pas de principe de fonctionnement standard pour les simulateurs de réseau Ad Hoc ou les simulateurs de mobilité. Il existe plusieurs stratégies et architectures qui permettent de simuler un réseau Ad Hoc.

Chaque simulateur répond aux besoins de la simulation de réseau via des architectures variées, en fonction des objectifs que se sont fixés leurs développeurs. Cependant, ils ont tous le même but : reproduire le plus fidèlement possible le fonctionnement d'une pile protocolaire.

Les simulateurs payants tels qu'Opnet [40] ou Qualnet [41] avec des licences commerciales ne sont pas abordés dans ce chapitre. Cette partie aborde les principaux simulateurs issus du logiciel libre : NS-2, NS-3 et Omnet++... etc.

### 3.2.1 Le simulateur NS-2

Le plus célèbre et le plus répandu des simulateurs de réseaux est sans conteste *Network Simulator 2* (NS-2) [42]. Il s'agit d'un simulateur à événements discrets disponibles gratuitement et open source. L'ouverture du code source à la communauté a contribué à l'enrichir de nouveaux protocoles et de nouvelles fonctions au fil du temps. Il permet à l'utilisateur de définir un réseau et de simuler des communications entre les nœuds de ce réseau. La simulation doit d'abord être saisie sous forme de fichier texte que NS-2 utilise pour produire un fichier trace contenant les résultats. [39]

NS-2 nécessite deux langages : OTcl (*Object Tools Command Language*) et C++ [43]. À travers le langage OTcl, l'utilisateur décrit les conditions de la simulation : topologie du réseau, caractéristiques des liens physiques, protocoles utilisés, etc. Bien que les scripts de simulation soient écrits en OTcl, la base du simulateur est en C++ ce qui permet à chacun de modifier à sa guise les différents protocoles. Ces éléments sont résumés sur la figure (3.1).

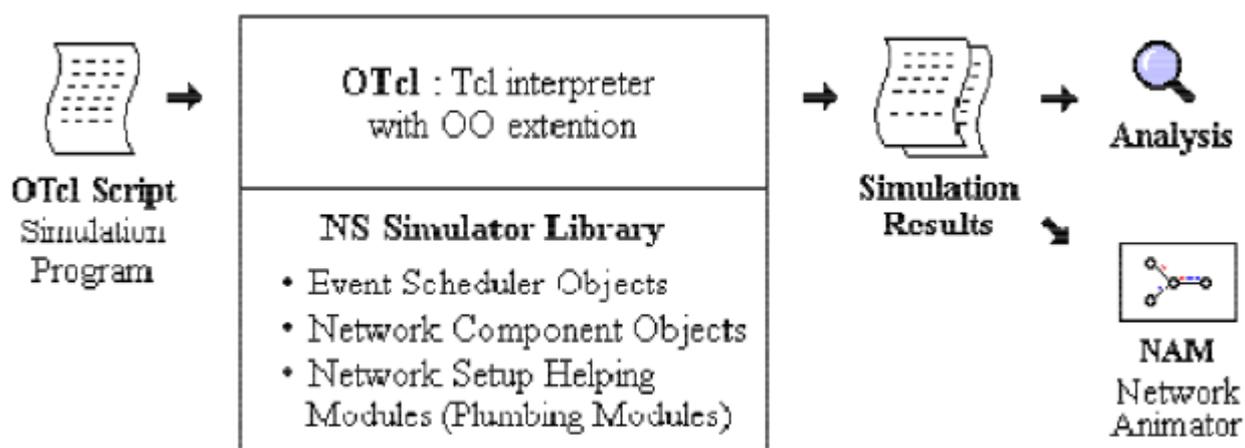


FIGURE 3.1 – Synoptique de l'interpréteur de script Otcl avec les bibliothèques C++ de simulation de réseau. [42]

Comme le montre la figure 3.1, le script Otcl qui définit la simulation est interprété et associé aux bibliothèques C++ du simulateur NS-2. L'ensemble permet d'obtenir des fichiers de résultats autorisant l'analyse ou la visualisation de la simulation.

### 3.2.1.1 Les composants de NS2

Le simulateur NS2 est adapté aux réseaux à commutation de paquets et à la réalisation de simulations de grande taille. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unicast ou multicast, des protocoles de transport, de session, de réservation, des services intégrés et d'application. La liste des principaux composants disponibles dans le simulateur NS2 est présentée dans le tableau (3.1) [44].

Application	Web, Ftp, Telnet, générateur de trafic (CBR...)
Transport	TCP, UDP, RTP, SRM
Routage	Statique, Dynamique (vecteur distance) et routage multipoint (DVMRP, PIM)
Gestion des files d'attente	RED, DROP Tail, Token bucket
Discipline de service	CBQ, SFQ, DRR, Fair queueing
Système de transmission	CSMA/CD, CSMA/CA, Lien point à point

TABLE 3.1 – Liste des principaux composants disponible dans NS2 [45]

### 3.2.1.2 Modèle de mobilité

NS2 implémente deux modèles de mobilité

#### *Le Random Waypoint Mobility Model*

Ce modèle génère un mouvement aléatoire du nœud qui choisit ensuite, aléatoirement, sa prochaine destination. Aussi dans ce modèle, le nœud se déplace avec une vitesse aléatoire constante.

#### *Le Trajectory based Mobility Model*

Ce modèle est défini par un scénario dans lequel l'utilisateur précise une destination et la vitesse de déplacement des nœuds. La vitesse de déplacement des nœuds est constante. [44]

### 3.2.1.3 Modèle de propagation dans NS2

Les modèles de propagation implémentés dans NS2 ont pour objectif de prédire la puissance de réception du signal des paquets. À la couche physique de chaque nœud, il existe un seuil de réception du signal de paquets. Quand la puissance de réception du signal du paquet reçu par un nœud est inférieure au seuil, le paquet est marqué comme erroné et il est abandonné dans la couche MAC [46] [47].

## 3.2.2 Le simulateur NS-3

Comme son nom l'indique, *Network Simulator 3* [48] est le successeur de NS-2. Son développement a commencé en 2006 et la première version stable est disponible depuis 2008. NS-3 a été programmé par la Georgia Institute of Technology, l'ICSI Center for Internet Research et le groupe de travail "Planète" de l'INRIA Sophia-Antipolis. NS-3 est développé en C++ et peut être relié au langage de script Python. Tout comme NS-2, il est gratuit et open source. De plus, il privilégie l'utilisation de logiciels libres, comme "Waf" (outils de compilation et distributions), "Doxygene" (génération automatique de documentation) ou encore le standard "posix" pour les fonctions liées à la gestion des sockets. Le but de NS-3 est de proposer un simulateur de réseaux capable de correspondre aux attentes des chercheurs. [39]

L'architecture de NS-3 a été modifiée par rapport à celle de NS-2, et la rétro-compatibilité de NS-3 avec NS-2 n'est pas possible. Ainsi le double langage utilisé par NS-2 n'existe plus dans NS-3. Le langage OTcl a complètement disparu de son architecture. NS-3, ainsi que tous ses modèles associés, sont écrits en C++. Il reste cependant possible d'écrire du code dans d'autres langages de programmation, comme le Python ou le Perl, en utilisant des générateurs intégrés à NS-3 qui vont transcrire ce code en C++ in fine. L'objectif est de garder la possibilité de créer ces scripts flexibles sans besoin de compilation, et d'avoir accès en même temps aux performances d'exécution du C++.

NS-3 utilise principalement les avancées technologiques des compilateurs C++ et des nouvelles bibliothèques associées qui n'existaient pas lors de la création de NS-2. Le simulateur contient un système d'objets hiérarchiques, et un système intégré de production de résultat de simulation. Cette base est associée à un système de gestion de paquets efficace en termes de mé-

moire, ainsi que des algorithmes de simulation d'évènements discrets qui gèrent la progression de chaque simulation. L'un des objectifs de cette architecture est de parvenir à simuler des réseaux de grande envergure, le tout avec un niveau de détails satisfaisant dans la simulation.

L'architecture réseau de NS-3 est basée sur les logiciels et matériels existants dans le monde réel. Cela facilite l'utilisation du simulateur et permet l'intégration aisée de nouveaux modèles. Par exemple, les paquets échangés dans la simulation sont équivalents au bit près, au même format que des paquets qui seraient échangés en expérimentation. Le passage de l'expérimentation à la simulation (et vice versa) est donc extrêmement commode avec NS-3. Et cela offre également la possibilité d'émuler l'un ou l'autre des éléments de la simulation sans avoir aucun problème de compatibilité entre l'existant et la simulation.

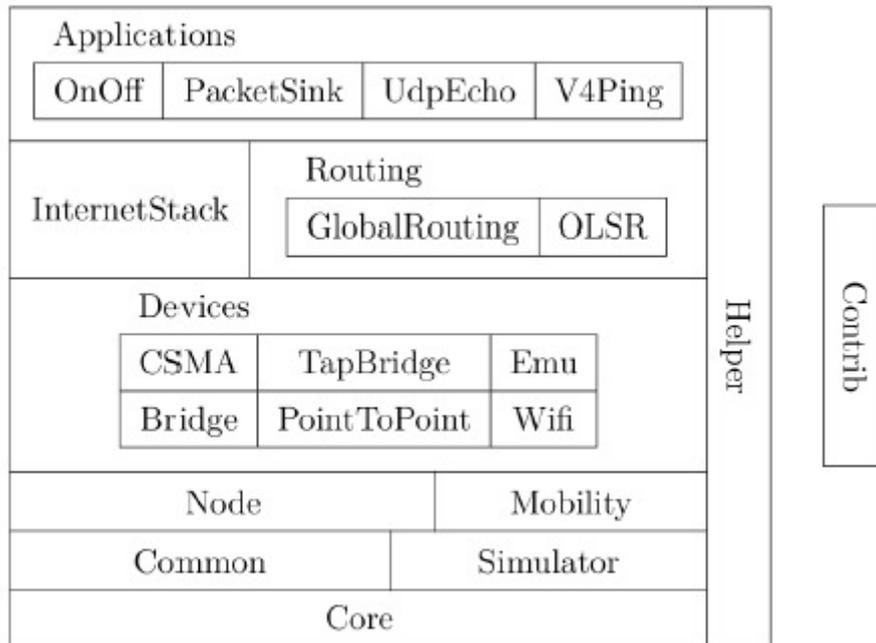


FIGURE 3.2 – Architecture en couches du simulateur NS-3 [39]

La figure 3.2 présente les différents éléments de base de l'architecture en couche de NS-3. Le principe est le suivant : chacun des composants repose sur le programme de la couche immédiatement inférieure. Le composant <core> forme la base de chaque programme NS-3 en créant une structure de classe hiérarchique C++. Le composant <Common> s'occupe de la simulation des paquets dans la simulation. Comme son nom l'indique, le composant <simu-

lator> gère la simulation d'évènements discrets et donc la gestion du temps. <node>est bien entendu consacre aux nœuds de la simulation. Comme précise précédemment, ce module copie les systèmes réels dans son fonctionnement. <mobility> est le composant dédiée à la mobilité et <device> celui aux interfaces des nœuds. Les couches 3 et 4 du modelé OSI sont gérées par le composant <internet stack>. La couche routage est gérée par le composant <routing> et la couche application par le composant <applications>. Enfin, le but du composant <helper> est de simplifier le codage des scénarios de simulation complexe, et le composant <contrib> contient tous les codes contributifs qui ne sont pas encore incorporés a l'architecture globale du simulateur NS-3.

### 3.2.3 Le simulateur OMNET ++

L'environnement de simulation a évènements discrets OMNeT++ est disponible gratuitement depuis 1997. OMNeT++ n'est pas un simulateur spécialisé pour un domaine particulier. Il a pour vocation d'être aussi généraliste que possible. Ainsi, OMNeT++ est utilisé dans de nombreux domaines, depuis les réseaux <peer-to-peer> jusqu'aux réseaux Ad Hoc en passant par la simulation de réseaux de stockage d'information. Ce logiciel a été développé pour fournir un simulateur puissant a évènements discrets open source. L'objectif était à l'époque de créer une alternative entre le simulateur dédiée à la recherche NS-2 et les alternatives commerciales coûteuses tel que le simulateur OPNET. [39]

Le simulateur OMNeT++ est dédié à la modélisation des communications dans différents types de réseaux. Il utilise GCC ou Microsoft Visual++ comme compilateur en fonction de la plate-forme de simulation. Contrairement aux autres simulateurs, OMNeT++ ne propose pas de composants prédéfinis pour la simulation de réseau mais plutôt les outils permettant de créer ces composants. Il existe bien sur de nombreux modules déjà implémentés, mais ils sont indépendants du simulateur de base et se développent (ou non) indépendamment de OMNeT++.

#### 3.2.3.1 Architecture d'OMNeT++

L'architecture du modèle OMNeT++ se compose de plusieurs modules hiérarchiquement imbriqués qui sont [49] :

- Le module système
- Modules simples (les feuilles) : programmés en C++ encapsulant le comportement d'un réel d'un système. Pour chaque module simple correspond un fichier *.cc* et un fichier *.h*.
- Modules composés : constitués d'un ou de plusieurs modules simples ou des modules composés reliés entre eux. Les paramètres, les ports et les modules de chaque module sont spécifiés dans un fichier *.ned*.

L'architecture d'OMNET++ est visualisée dans la figure qui suit :

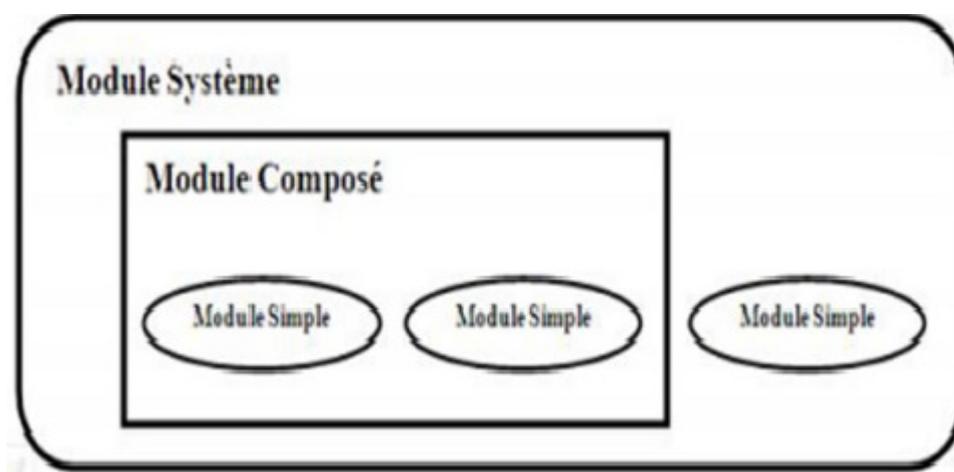


FIGURE 3.3 – Architecture modulaire du simulateur OMNET++ [49]

Dans la construction d'un nouveau programme de simulation, il y aura à chaque fois des informations chargées dynamiquement telles que la topologie du réseau à partir des fichiers *.ned*, et les configurations sont disponibles dans les fichiers *.ini*.

Lors de la simulation, différents fichiers trace seront remplis. On a aussi le « Plove » qui est un outil pouvant visualiser les données enregistrées. Les deux fichiers *omnet.vec* et *omnet.sca* seront utiles lors du traçage de la courbe et du calcul des statistiques.

### 3.2.3.2 Composants d'OMNET++

Les composants les plus pertinents dans le simulateur OMNET++ sont représentés dans le Tableau suivant [49] :

Application	FTP, Telnet, generateur de trafic (IP Trf Gen..), Ethernet, Ping App, UDP App, TCP App
Transport	TCP, UDP, RTP
Réseau	IPv4, IPv6, ARP, OSPF, LDP, MPLS, ICMP, TED
Liaison	Mgmt, MAC, Radio
Node	Ad Hoc, Wireless, MPLS

TABLE 3.2 – Composent de OMNET++ [49]

la figure suivante représentai l’interface OMNET++

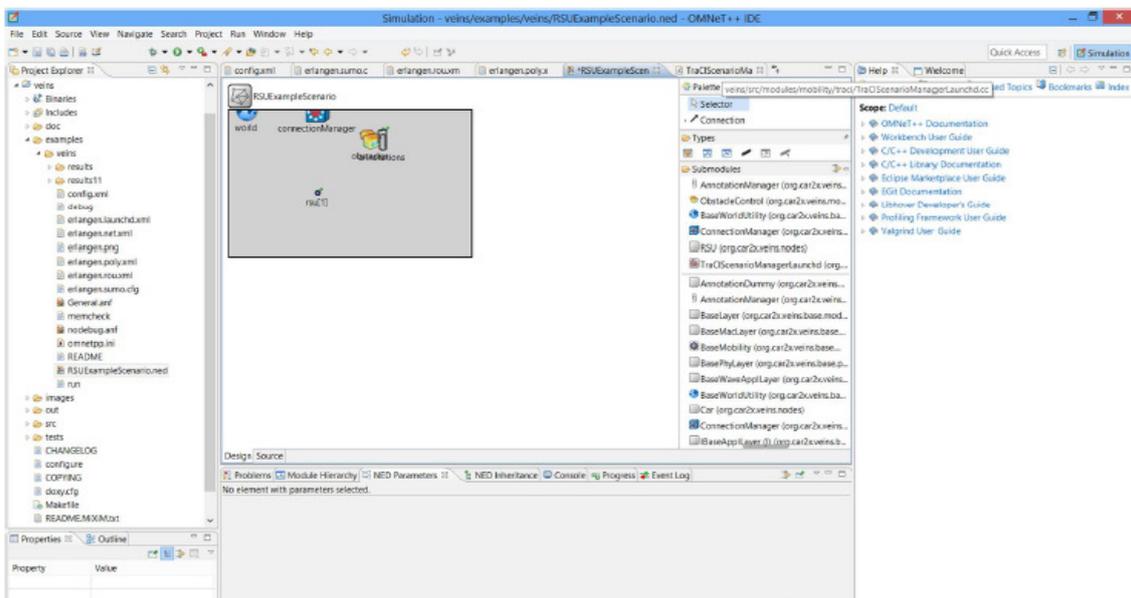


FIGURE 3.4 – interface OMNET++ [49]

### 3.2.4 Comparaison entre les simulateurs NS-2 et OMNET++

Après avoir brièvement décrit les deux simulateurs réseaux (NS2 et OMNET ++), nous allons faire une étude comparative des deux simulateurs dans cette section. Les résultats de cette étude sont présentés dans le tableau 3.3.

Propriétés	OMNET++	NS2
Flexibilité	Très flexible et générique. Il peut simuler n'importe quel type de réseau.	Il est difficile de simuler autre choses que les réseaux de commutations de paquets et protocoles
Mobilité	Plusieurs mode de mobilités sont fournis (Random Waypoint Mobility Model, Constant Speed Mobility Model, . . . )	Supporte que Random Waypoint Mobility Model et le Trajectory Based Mobility Model
La gestion de modèle	Les modèles sont indépendants du noyau de simulation	La limite entre les modèles et le noyau de simulation n'est pas très considérable
Support de traçage	Visibilité de la transmission de paquet lors de la simulation	Pas de traçage
Habilité à couvrir les grands réseaux	Peut simuler une grande topologie de réseaux	Beaucoup de problèmes dans la simulation des grandes topologies de réseaux

TABLE 3.3 – Comparaison entre NS2 et OMNET ++

### 3.3 Les Simulateurs de mobilité et de trafic routier

#### 3.3.1 Les simulateurs indépendants

On trouve deux types de simulateurs de mobilité : les simulateurs indépendants et les simulateurs intégrés. Les simulateurs indépendants génèrent uniquement la mobilité des réseaux VANETS. Il faut en général leur associer des outils pour pouvoir exploiter cette mobilité dans un simulateur de réseau Ad Hoc. Les simulateurs intégrés regroupent à la fois un simulateur de mobilité et un simulateur de réseau Ad Hoc, donc il n'y a pas de problème de compatibilité.

### 3.3.1.1 Le simulateur VanetMobiSim

VanetMobiSim [50] est une extension de CanuMobiSim [51]. En effet, CanuMobiSim est une plateforme flexible pour la modélisation de la mobilité. Cette plateforme est mise en œuvre comme une application JAVA autonome et peut être utilisée avec presque n'importe quel outil de simulation pour des réseaux mobiles (format NS-2 et QualNet).

La modélisation des VANET inclut des relations entre véhicules mais aussi entre véhicules et infrastructures. Au niveau mobilité, elle doit également inclure les panneaux stop, les feux de circulation et une mobilité basée sur l'activité humaine. VanetMobiSim regroupe tous ces éléments, i.e. Il utilise les deux aspects macroscopiques et microscopiques, il permet aussi d'extraire des cartes à partir des bases de données tel que TIGER, GDF ou OSM. Ainsi que de créer des cartes manuellement ou aléatoirement.

VanetMobiSim offre autant de possibilités et de fonctionnalités pour créer des scénarios réalistes. En outre, des scénarios de simulation pour VanetMobiSim sont définis au format XML à l'aide des balises, facilitant la configuration de scénario et d'une manière plus pratique [40].

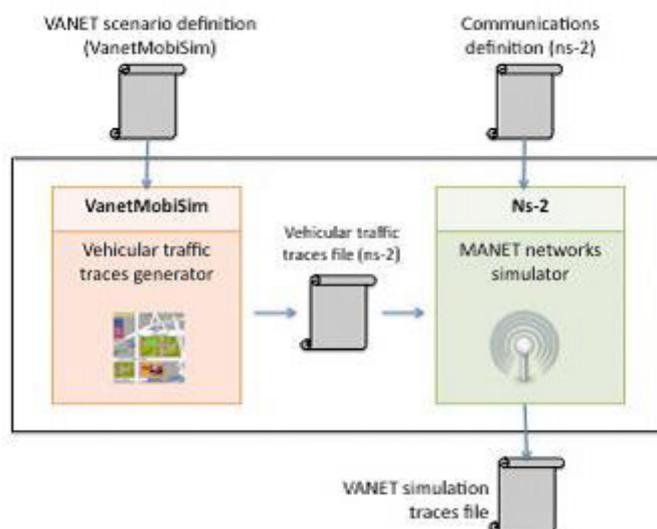


FIGURE 3.5 – Simulateur de VANET basée sur le couplage de VanetMobiSim et NS2 [52]

### 3.3.1.2 Le simulateur SUMO

SUMO (*Simulation of Urban Mobility*) [53] [54] est un simulateur de mobilité open source. Il fait partie des logiciels précurseurs du domaine et a ouvert la voie sur cette thématique. Il est complexe d'utilisation mais permet de générer de la mobilité sur une zone géographique de la taille d'une ville.

Le but premier de SUMO est de pouvoir générer une mobilité basée sur des critères de décision qui ne sont pas totalement aléatoires et qui imitent le comportement humain. Pour cela, SUMO intègre plusieurs éléments dont notamment : un modèle de conducteur (*Car-Driver Model*), un modèle de feux de croisement (*Traffic Lights*), et un format de fichier de sortie pour sauvegarder le résultat de la simulation de mobilité.

Le Car-Driver Model implémenté dans SUMO permet à chaque véhicule d'adapter sa vitesse en fonction des véhicules le précédant, évitant ainsi toute collision. Le Trafic *Lights* donne des règles d'arrêt au carrefour qui évolue dans le temps. Le fichier de résultats inclut les points de départ avec les vitesses et destinations initiales pour chaque nœud, ainsi que toutes les modifications de comportement (changement de vitesse ou de destination). Ces modifications sont sauvegardées chronologiquement par rapport au temps simulé. [39]

### 3.3.1.3 Le simulateur CORSIM

CORSIM (*Corridor Simulation*) : est un simulateur commercial très puissant et largement utilisé dans les systèmes de transport intelligent (STI), CORSIM est développé par le département Américain des transport et l'USFHWA (*Federal Highway American*) pour la simulation et la génération de trafic, il est basé sur un modèle de simulation macroscopique pour simuler le modèle de mobilité dans les autoroutes et les routes urbaines, il intègre plusieurs mécanismes dont le changement de voies, l'accélération et la décélération, car following, le contrôle aux intersections, etc. Le simulateur CORSIM est une combinaison de deux programmes distincts : le simulateur de route urbaine NETSIM, et le simulateur de routes expresse et d'autoroute FRESIM [6].

#### 3.3.1.4 Le simulateur PARAMICS

Ce simulateur est développé et commercialisé par une société britannique « Qudstone Limited », PARAMICS est constitué d'une suite de modules de simulation de trafic microscopique qui sont : un modeleur, un estimateur, un processeur, un analyseur, un programmeur et un moniteur. Le modeleur est considéré comme le cœur de simulateur, il fournit les fonctionnalités fondamentales pour la configuration de modèle de mobilité, la simulation, la visualisation du trafic et la génération des statistiques. Il utilise trois modèles pour définir le comportement des véhicules. Le premier modélise les déplacements en file des véhicules (*car following*), le second concerne la distance inter-véhicule et le dernier s'occupe de mécanisme de changement de voies.

#### 3.3.1.5 Le simulateur CityMob

Est un simulateur développé à l'Université Polytechnique de Valence, il est conçu pour être utilisé avec le simulateur NS-2 dans la génération de modèle de mobilité pour les réseaux VANET. Pour une représentation réelle des traces de scénario dans VANET, CityMob propose trois modèles de mobilité différents [6] :

- (i) Modèle simple (MS) : Modèle de mobilité vertical et horizontal sans changements de direction ou feux de circulation.
- (ii) Le Modèle Manhattan (MM) : dans ce modèle, les nœuds se déplacent dans le sens horizontale et verticale sur une carte de type Manhattan, les routes sont à double sens avec une seule voie pour chaque direction, le mouvement de nœuds est choisi aléatoirement à chaque moment et ne peut pas être répétés dans deux mouvements consécutifs, ce modèle simule les feux de circulation à des positions aléatoire et des délais différents.
- (iii) Le modèle de Centre-ville (Downtown) réel (MD) : Ce modèle améliore le modèle de Manhattan avec la densité du trafic pas uniformément distribuée ; il y a des zones avec une densité plus élevée agissant en tant que centre-ville. Ces secteurs les véhicules doivent se déplacer plus lentement que la périphérie. Le centre-ville peut être choisi et peut ne jamais couvrir plus de 90 pour-cent de tout le domaine de carte. Il y a un paramètre,  $p$ , qui peut être employé pour établir la probabilité d'un nœud étant au commencement

situé à l'intérieur du centre-ville et également la probabilité que les nœuds sur les périphéries entrent dans le centre-ville. Les dispositifs restants sont les mêmes que pour le modèle de mobilité de Manhattan.

### 3.3.2 Les simulateurs intégrés

Plusieurs solutions au problème d'interconnexion entre les simulateurs de mobilité et les simulateurs de réseau Ad Hoc existent. La première consiste à rendre le format des résultats produits par le générateur de mobilité lisible par le simulateur de réseau. Cela n'est malheureusement pas toujours possible et il faut en général adapter les résultats de la simulation de mobilité spécialement pour le simulateur de réseau via des scripts spécifiques. Une autre solution consiste à utiliser un simulateur dédié VANETs qui intègre à la fois le simulateur de mobilité et de réseau Ad Hoc, et qui donc, par nature, ne souffre pas de soucis d'interconnexion.

Le grand changement apporté par ces simulateurs par rapport aux autres simulateurs de réseau est le fait qu'ils associent les simulateurs de mobilité et de réseau Ad Hoc au sein d'une même structure. Ainsi, ces simulateurs peuvent être caractérisés de simulateur VANETs à proprement parler car ils intègrent chacun des éléments nécessaires à la simulation. [39]

#### 3.3.2.1 Le simulateur NCTU

NCTU (*Nation Chiao Tang University Network Simulator*) [55] : C'est un logiciel open-source sous Linux, Fedora 11, avec un environnement graphique intégré. Ce simulateur est développé comme un émulateur réseau, mais les versions les plus récentes (5.0) intègre des fonctionnalités non seulement pour les réseaux véhiculaires en général comme l'implémentation complète de la norme émergente IEEE 802.11p et les normes 1609 pour les réseaux de véhicules sans fil, mais aussi pour la modélisation et la simulation des réseaux de véhicules tels que la conception et le contrôle de la mobilité des véhicules i.e. Auparavant, pour étudier la mobilité des véhicules nous utilisons des simulateurs de mouvements externes, l'avantage de NCTU par rapport aux simulateurs réseaux traditionnels est qu'il offre un module interne pour la simulation de la mobilité.

- Génère des résultats de simulation proche à la réalité en se basant sur des applications réelles dans la génération du réseau de trafic.

- La performance de toute application réelle peut être facilement évaluée sur NCTU dans différentes conditions de simulation réseau.
- Tout programme d'application de réseau développé pour le dispositif NCTU peut être exécuté directement sur un périphérique réel Linux sans aucune modification.

### 3.3.2.2 Le simulateur TraNs

TraNs (*Traffic et Network Simulation Environment*) [56] : Est un simulateur du trafic et réseaux en même temps, TraNs combine entre le simulateur réseau Ns-2 et le simulateur de mobilité et du trafic SUMO, il est considéré quasiment le premier simulateur des VANET, comme il peut fonctionner selon les deux modes centré réseau et centré-applications. En mode centré réseau, il n'y a pas de retour d'informations de simulateur TraNs vers SUMO, les traces de mobilité obtenu de simulateur SUMO sont transmises à Ns-2. Dans le mode centré application, la rétroaction entre Ns-2 et SUMO par l'interface TraCI pour assurer le fonctionnement simultané des deux simulateurs (SUMO et Ns-2). Ce simulateur garanti un lien entre les deux en convertissant les commandes en provenance de ns-2 en des primitives de mobilité telles que l'arrêt, changement de voies, changement de vitesse qui sont envoyées à SUMO. Cependant, l'inconvénient de TraNs réside dans les modèles de propagation radio irréal utilisés par ns-2 et qui ne sont pas spécifiques aux réseaux. Véhiculaires, aussi SUMO qu'est incapable de modifier les trajectoires des véhicules lorsque des imprévus apparaissent.

## Conclusion

Dans ce chapitre nous avons montré l'objectif de la simulation et les différents simulateurs dédiés pour les réseaux Ad hoc et spécialement les VANETs. Ainsi, nous avons présenté quelques simulateurs de mobilité, utilisés conjointement avec des simulateurs réseaux pour simuler les vanets.

Dans ce qui suit nous allons présenter la partie conception et réalisation de notre simulateur.

# 4

## Conception et Réalisation du Simulateur

# PART I : conception

## Introduction

Avant la réalisation de toute application, celle-ci doit être passer par une conception, afin de tracer les besoins et de mieux comprendre son fonctionnement.

De ce fait, ce chapitre est dédié à la conception de notre simulateur, dans lequel on a présenté une description de simulateur ainsi que les objectifs à atteindre. Après cela on passera à la modélisation UML de celle-ci.

### 4.1 La problématique

Le but de ce projet est d'implémenter un simulateur dédié pour les VANETs, et d'utiliser le protocole A-STAR.

Ce protocole de routage permet d'acheminer l'information d'un véhicule à un autre tout en suivant l'évolution du réseau.

## 4.2 Processus de la simulation

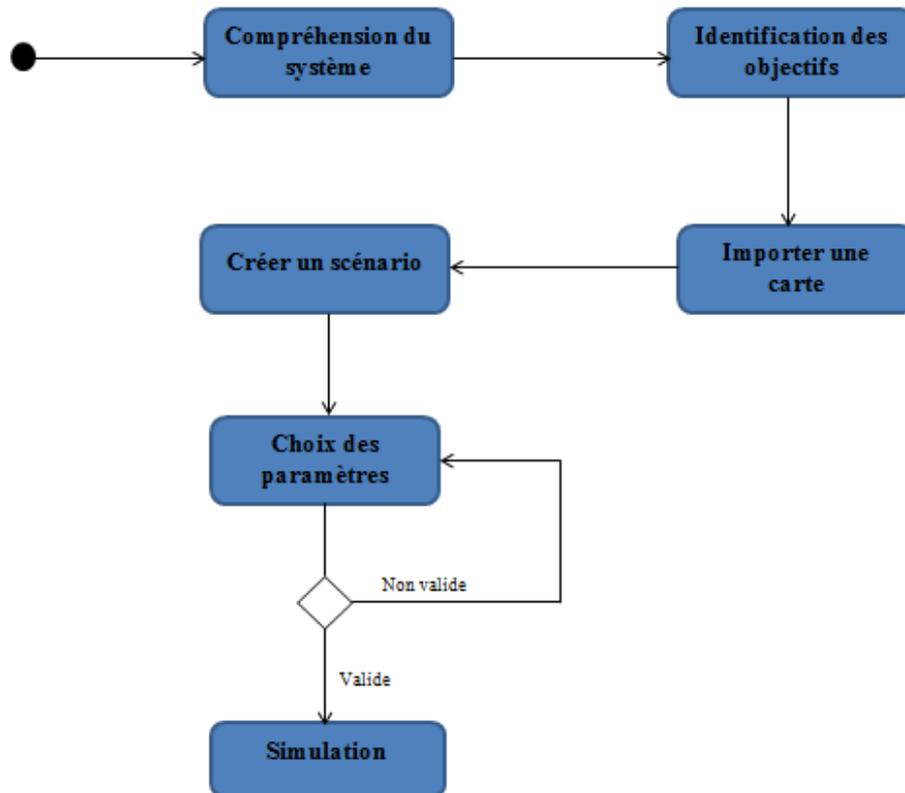


FIGURE 4.1 – processus de la simulation de notre simulateur.

## 4.3 Besoins fonctionnels

Notre travail consiste à implémenter un simulateur dédié pour les VANETs avec l'utilisation de protocoles A-STAR et les fonctionnalités de ce simulateur.

Nous allons d'abord donner une description générale de ces besoins.

### 4.3.1 Description générale

1. Implémentation d'un module représentant un nœud mobile qui :

- Se déplace selon un modèle de mobilité précis.

- Utilisation du protocole A-STAR.
2. Construction d'un réseau à partir des nœuds modélisés.
  3. Simulation de l'échange de paquets entre les différents nœuds mobiles suivant des scénarios.

### 4.3.2 Cas d'utilisation

Le but de cette partie est de décrire les requis fonctionnels du programme selon le formalisme UML (Universel Modeling Language) qui fournit un moyen visuel standard pour spécifier, concevoir et documenter les applications orientées objets, à l'aide d'éléments appelés diagrammes UML.

#### 4.3.2.1 Cas d'utilisation du simulateur

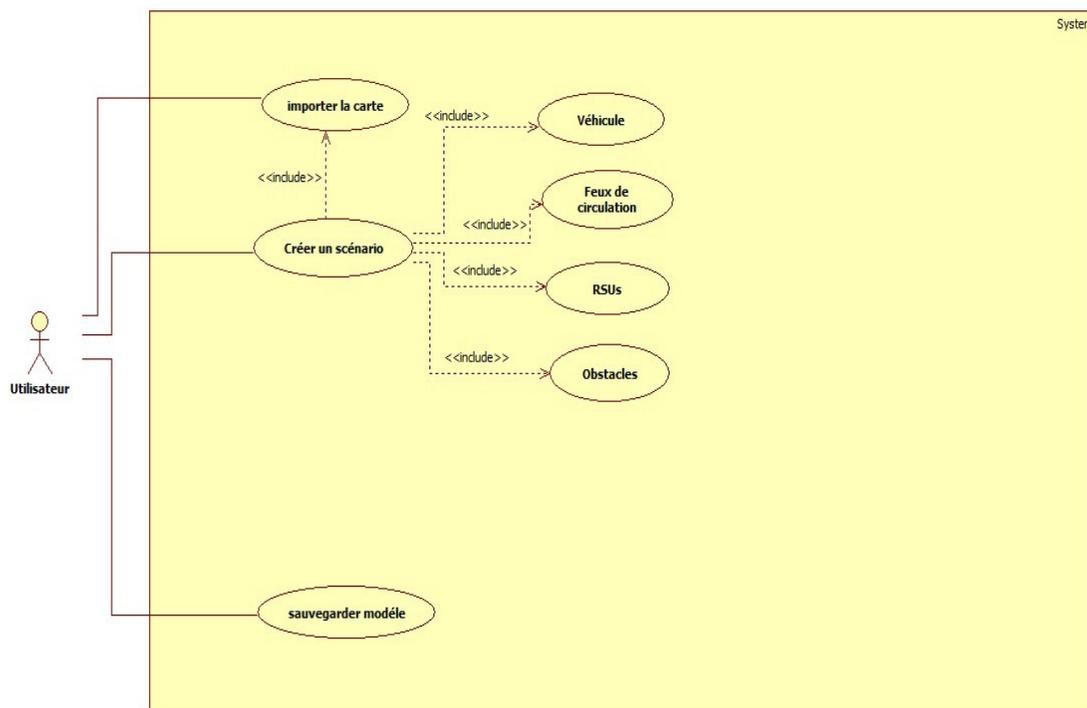


FIGURE 4.2 – Diagramme de cas d'utilisation pour la création de scénario.

### Scénario

1. Définir un modèle de mobilité : importer une carte.
2. Créer un scénario qui regroupe les éléments nécessaire pour créer un scénario et voire l'échange de données.
3. Utiliser le scénario crée pour simuler la communication selon le protocole A-STAR.

#### 4.3.2.2 Cas d'utilisation au niveau applicatif

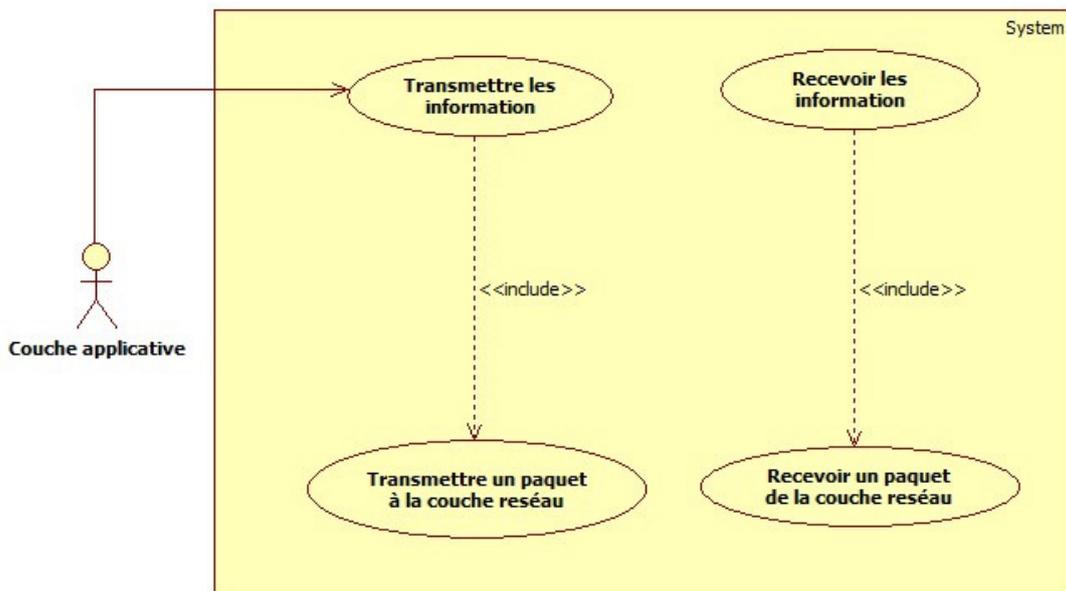


FIGURE 4.3 – diagramme de cas d'utilisation au niveau applicatif.

#### Scénario :

- La couche applicative de nœud 1 communique avec une application de nœud 2.
- La couche applicative de nœud 1 transmet des informations à la couche applicative de nœud 2 en passant par sa propre couche réseau.
- La couche applicative de nœud 2 reçoit les informations qui sont passées par sa couche réseau sous forme de paquet.

### 4.3.2.3 Cas d'utilisation au niveau réseau

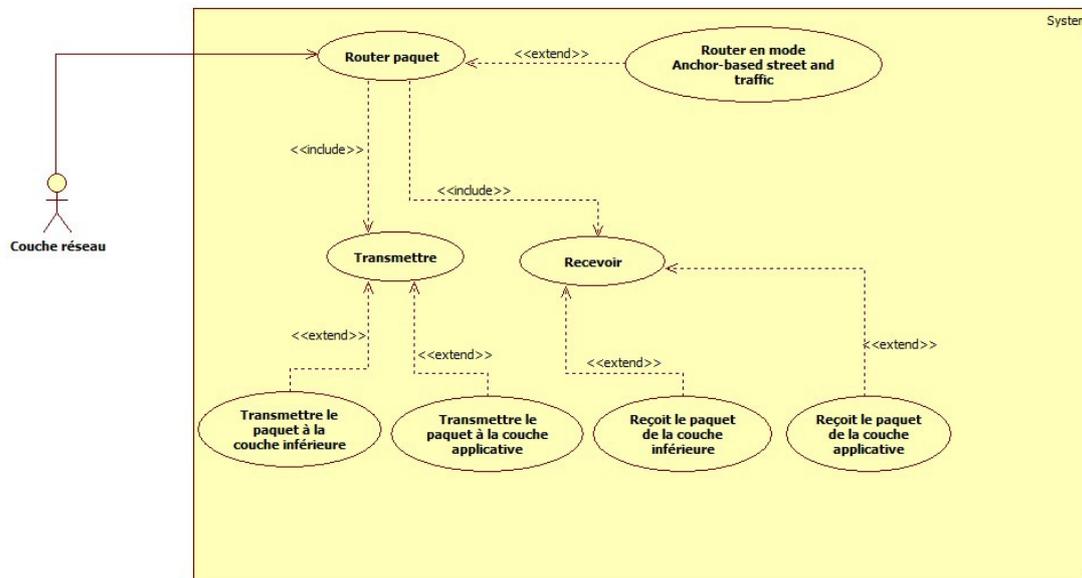


FIGURE 4.4 – diagramme de cas d'utilisation au niveau réseau.

#### Scénario :

- La couche réseau du premier nœud reçoit un paquet de sa couche applicative.
- La couche réseau de premier nœud route le paquet en mode (*Anchor based street and traffic*) tout en passant à sa couche inférieure.
- La couche réseau du deuxième nœud reçoit le paquet et le passe à sa couche réseau qui le délivre à sa couche applicative.

### 4.3.2.4 Diagrammes de séquences

Diagramme d'interaction qui représente les objets participant à une interaction particulière et les messages qu'ils échangent organisé en séquences horaires. Axé sur ce que fait un système et non sur la manière dont il le fait, un diagramme de séquence définit la Logique d'une instance particulière d'un CAS d'utilisation. En général, dans un diagramme de séquence, la Dimension verticale représente les heures (de haut en bas) et la Dimension horizontale représente les différents objets.

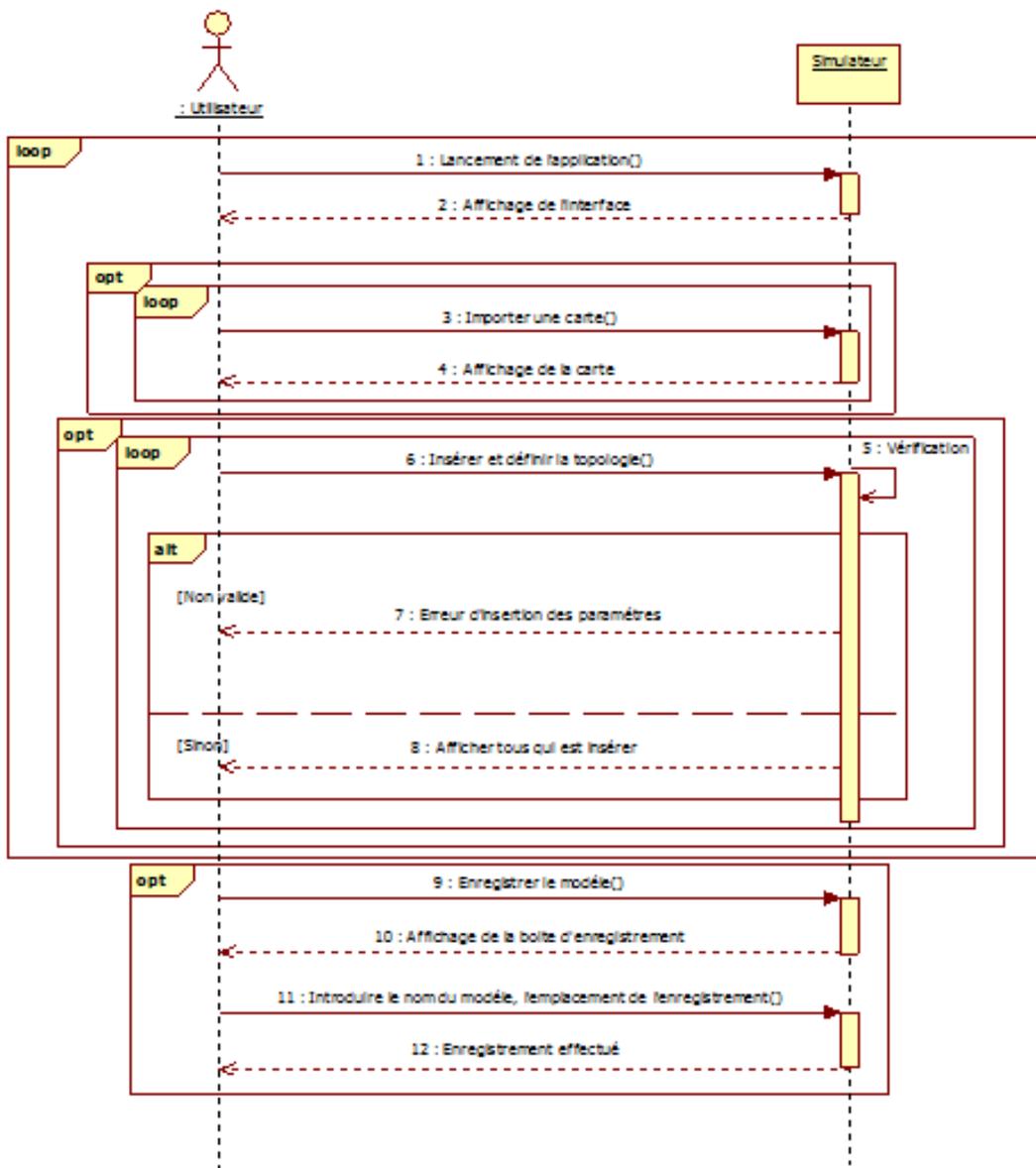


FIGURE 4.5 – Diagramme de séquence de simulateur.

4.3.2.5 Diagramme d'état transition

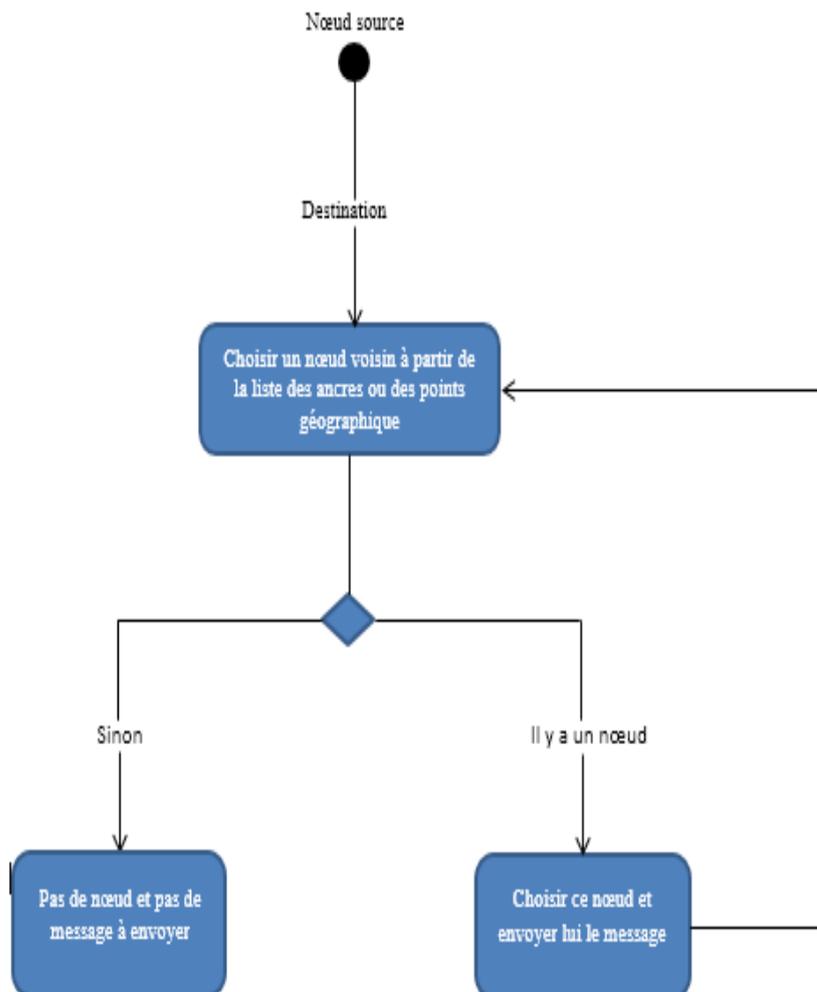


FIGURE 4.6 – Diagramme d'état de transition de notre simulateur

### 4.3.2.6 Diagramme de classe

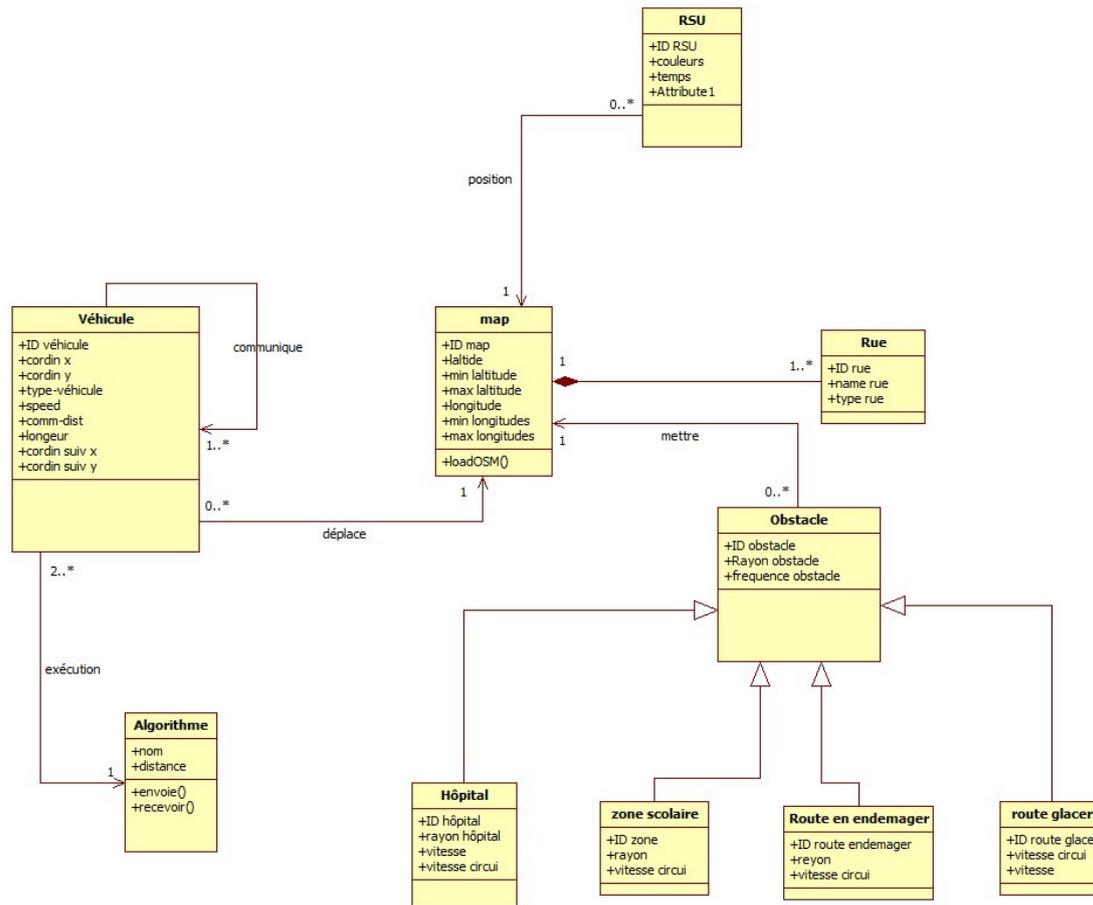


FIGURE 4.7 – Diagramme de classe du simulateur

## Conclusion

Dans ce chapitre, on a présenté la conception de notre simulateur, en montrant ses principales fonctionnalités a travers un diagramme d'utilisation, les deux diagrammes d'utilisation au niveau réseau et applicatif, de séquence, d'état de transition et de classe.

Dans le prochain chapitre on va mettre en œuvre cette conception.

## PART II : Réalisation

## Introduction

Après avoir terminé l'étape de la conception nous passons à la dernière phase qui est la réalisation. Nous allons présenter les environnements matériels et logiciels. Ensuite, nous décrirons la simulation et les étapes de son déroulement.

### 4.4 Environnement de travail

Nous allons détailler les outils utilisés dans la réalisation de notre simulation :

#### 4.4.1 Environnement matériel

La simulation a été réalisée sur un ordinateur *hprobook4540s* dont la configuration est :

- Système D'exploitation : Linux
- Distribution : Debian GNU/Linux 8 (jessie) 64bits
- Mémoire RAM : 3,8Gio
- Processeur : *Intel®Core™i3 – 3110M CPU@2.40GHz X4*
- Carte graphique : *Intel®Ivybridge Mobile*

#### 4.4.2 Environnement logiciel

##### 4.4.2.1 Éclipse Luna

Éclipse est un environnement de développement intègre (IDE) libre extensible, universel et multi langage, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

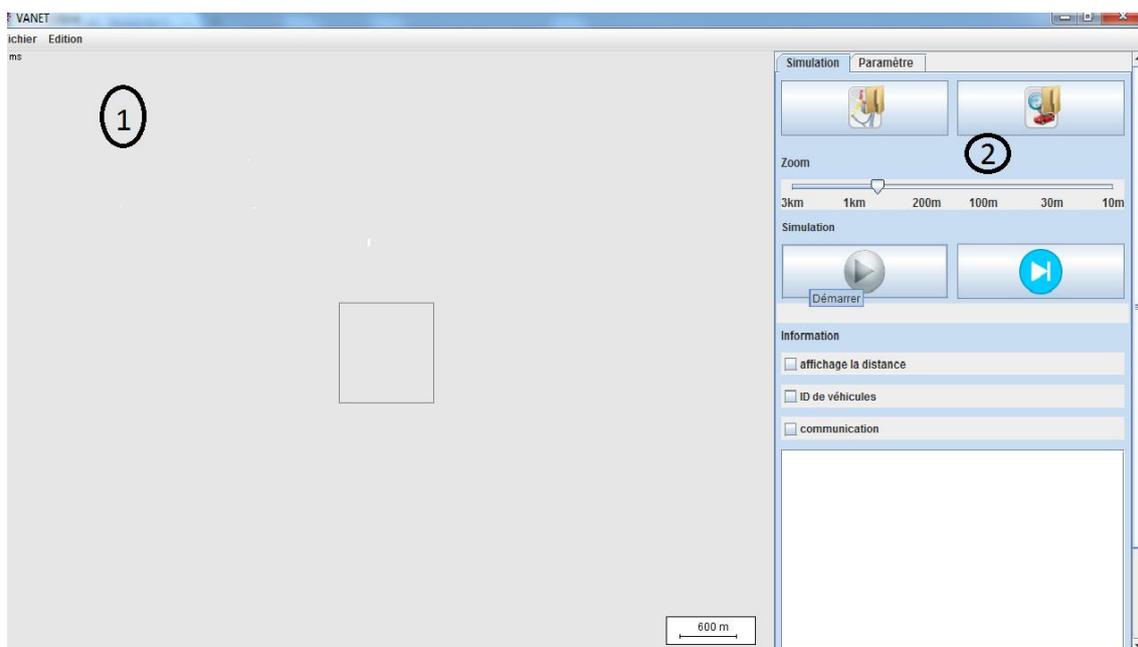
La spécification d'éclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (extension), toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plugin.

#### 4.4.2.2 Java

C'est un langage de programmation orienté objet développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh). Java donne aussi la possibilité de développer des programmes pour téléphones portables. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

### 4.5 Présentation de simulateur

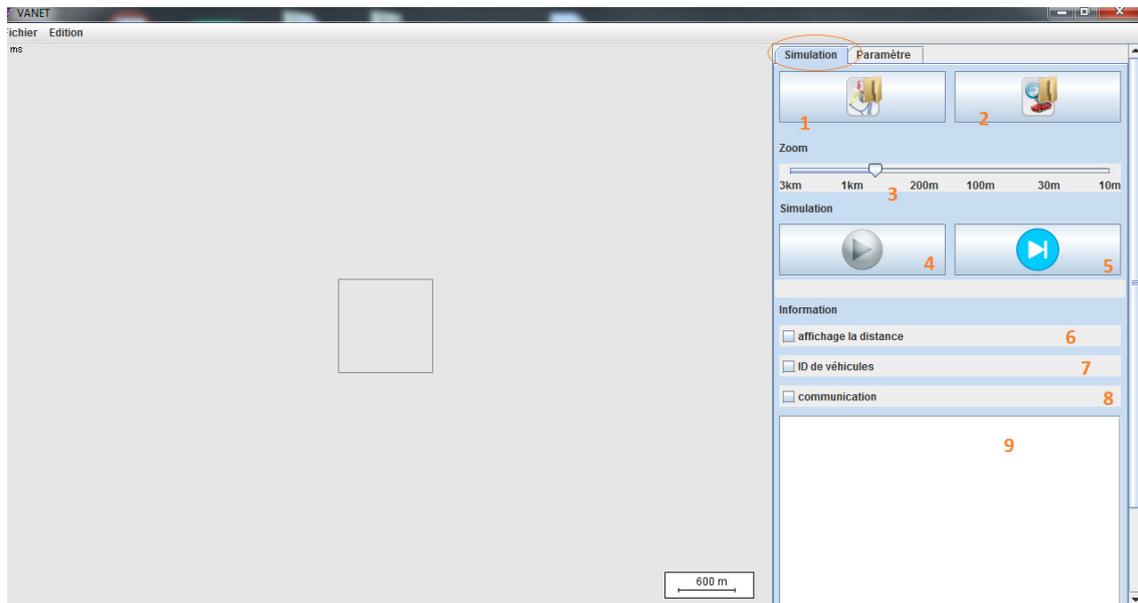
Le volet technique de ce chapitre étant terminé, nous allons désormais consacrer cette partie du chapitre à la présentation des principales zones de simulateur



**Zone 1 :** C'est la zone où se déroule la simulation, elle comporte les cartes de type '.xml' ou '.osm' qu'on peut obtenir des différents sites map comme «openstreetmap.org », et le scénario créé par l'utilisateur.

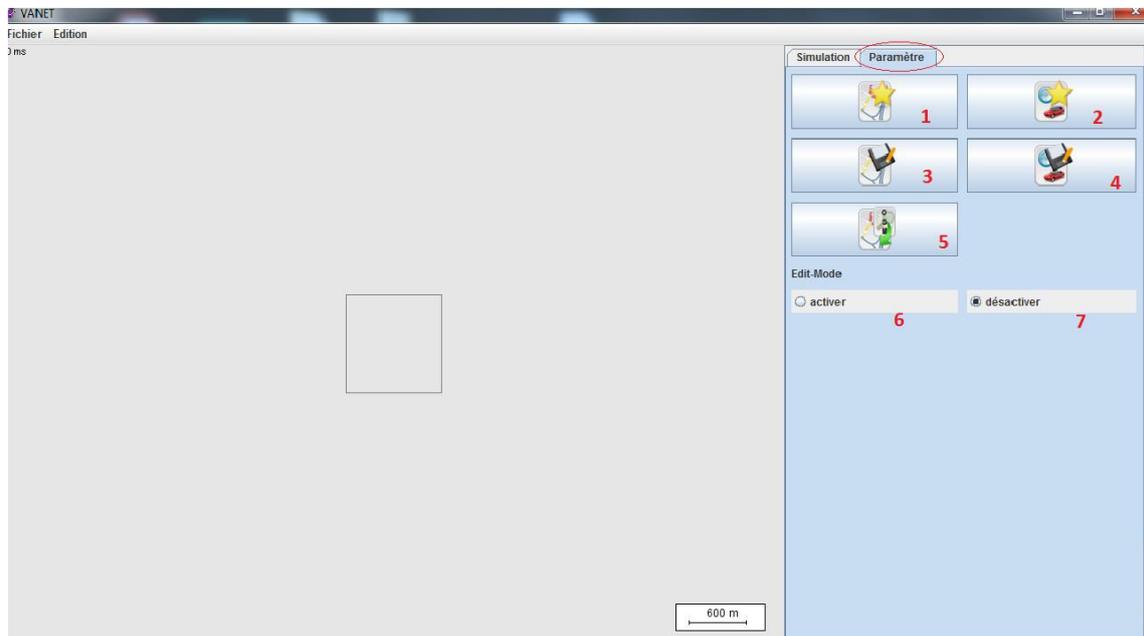
**Zone 2 :** Elle comporte les différents paramètres.

## angle simulation



1. Charge une carte de type *.xml*.
2. Charge un scénario déjà sauvegarde.
3. Zoom sur la carte.
4. Lance/arrête la simulation.
5. Accélère la simulation.
6. Affiche la zone de couverture de véhicule.
7. Affiche l'identifiant de véhicule.
8. Affiche la communication entre le véhicule.

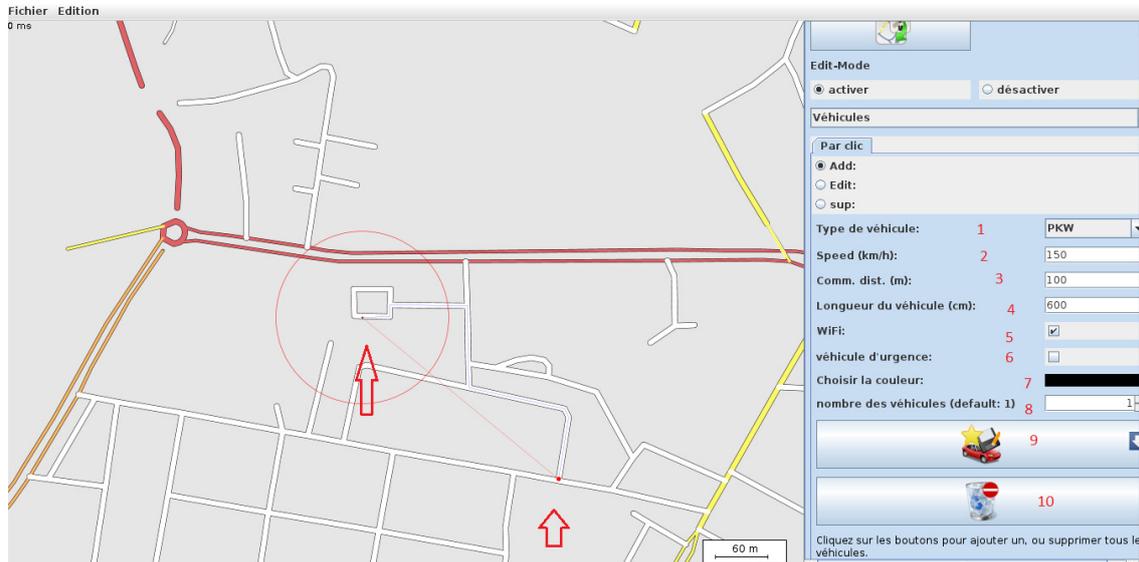
## angle paramètre



1. Nouvelle carte.
2. Nouveau scénario.
3. Enregistre la carte.
4. Enregistre le scénario.
5. Charge une carte de type '.osm'
6. Active l'ajout des paramètres.
7. Désactive l'ajout des paramètres.

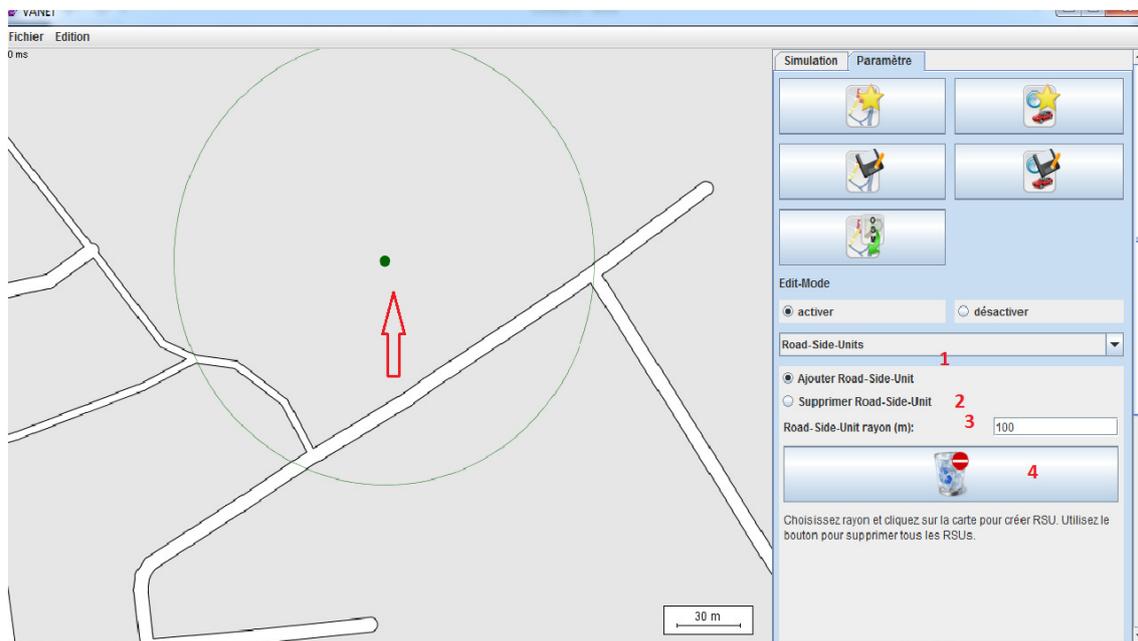
## Les différents paramètres

### Ajout d'un véhicule



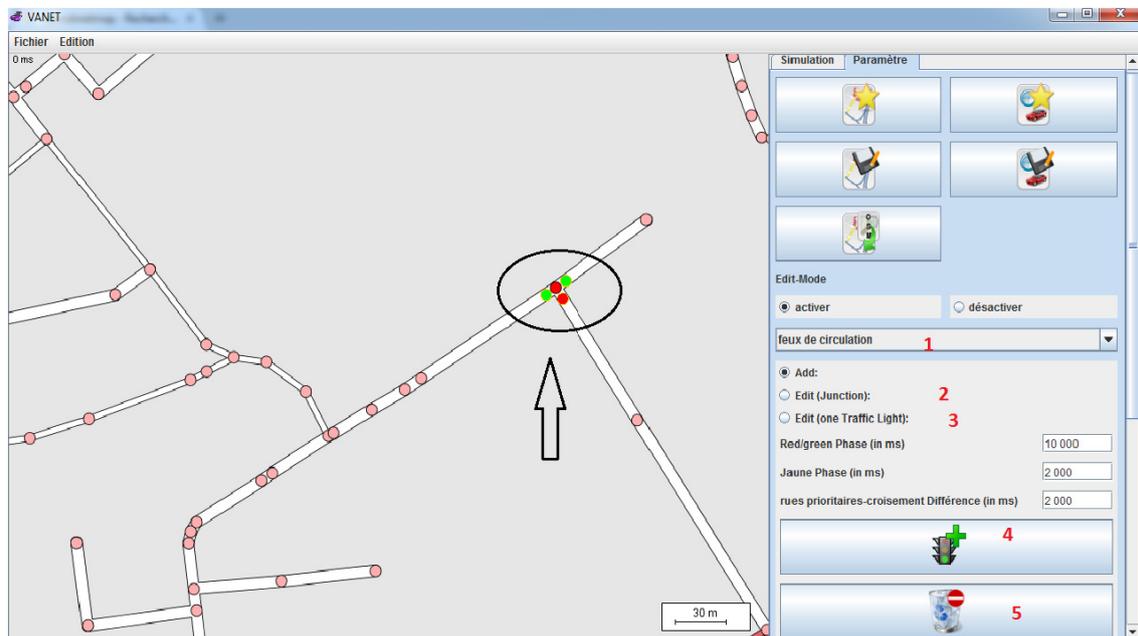
1. Type de véhicule.
2. Vitesse de véhicule.
3. Distance de communication.
4. Longueur de véhicule.
5. Véhicule équipé d'un wifi
6. Véhicule en état d'urgence.
7. Choisir la couleur de véhicule.
8. Nombre de véhicule.
9. Ajoute le véhicule sur la carte en choisissons le point de départ et point d'arrivée.
10. Suppression du véhicule qui est dans la carte.

## Ajout d'un RSU



1. Ajoute les RSU.
2. Sélection pour supprimer un RSU particulier.
3. Rayon de zone de couverture d'un RSU.
4. Supprime tout le RSU qui se situe sur la carte.

## Ajout d'un feu de circulation



1. Ajoute des feux de circulation sur des Junction « intersection ».
2. Modifier le placement de feux de circulation dans une Junction.
3. Modifier un feu de circulation.
4. Mettre un feu de circulation sur la carte.
5. Supprime les feux de circulation.

## Exemple d'une simulation



## Conclusion

Dans ce chapitre, nous avons présenté la conception et la réalisation de notre simulateur dédié pour les VANETs qui permet de créer les différents scénarios de circulation et de consulter les informations relatives pendant la simulation.

# Conclusion Générale et Perspectives

Le développement des nouvelles technologies a favorisé une formidable évolution des réseaux véhiculaires. Cette évolution vise à rendre les réseaux plus sûrs, plus efficaces, plus fiables et plus écologiques aussi bien du point de vue de l'industrie automobile que des opérateurs de réseaux et services. Les réseaux véhiculaires sont en effet une classe émergente des réseaux mobiles Ad Hoc, permettant des échanges de données entre véhicules ou encore véhicules et infrastructure. Ils suscitent un intérêt certain dans le but d'améliorer la sécurité et l'efficacité des transports routiers et éviter la circulation ainsi que l'amélioration du confort de l'utilisateur en lui fournissant une multitude des services d'information, d'aide à la décision, de guidage et d'accès à internet.

L'objectif de notre travail est de concevoir et réaliser un simulateur pour les réseaux VANETS, pour cela on a présentée dans ce mémoire une étude générale sur les réseaux VANETS.

Nous avons aussi présenté quelque simulateur de ce réseau à savoir OMNET++, NS2, NS3 avec leurs composants et leur architecture. À travers cette étude nous avons pris de chacun de ces simulateurs étudiés quelques points et des idées pour la conception et la réalisation de notre simulateur.

Afin de concevoir et de réaliser notre simulateur on a opté à utiliser l'un des protocoles de réseau Ad Hoc "A-STAR" par rapport à sa simplicité.

Notre simulateur réalisé permet de :

- Importer une carte sur laquelle elle va se dérouler la simulation.
- Créer un scénario d'où il comporte les différents éléments : les véhicules, les RSU, les feux de circulation et les événements comme route glacé, hôpital...etc.
- Visualiser la simulation.
- Enregistrement de scénario qu'on a créer.

Dans les perspectives, nous proposons d'améliorer notre simulateur en suivant ces points :

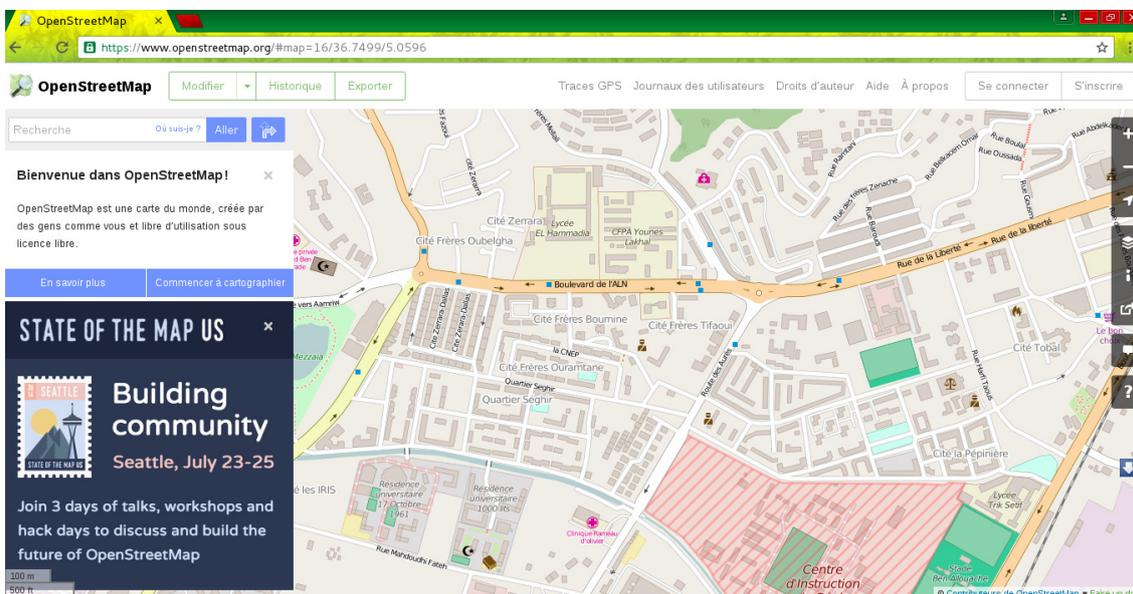
- amélioré la visualisation de la simulation
- ajout d'autre fonctionnalité (Intégration des panneaux de signalisation, création des routes, lancer plusieurs simulations en même temps, relier le simulateur a un serveur map, ...etc).
- intégrer des autorités de certificat CA.
- permettre d'autre type de communication (véhicule-à-infrastructure, infrastructure-à-CA, CA-à-véhicule)
- implémenter d'autres protocoles de routage.
- visualisation des résultats de la simulation graphiquement

A

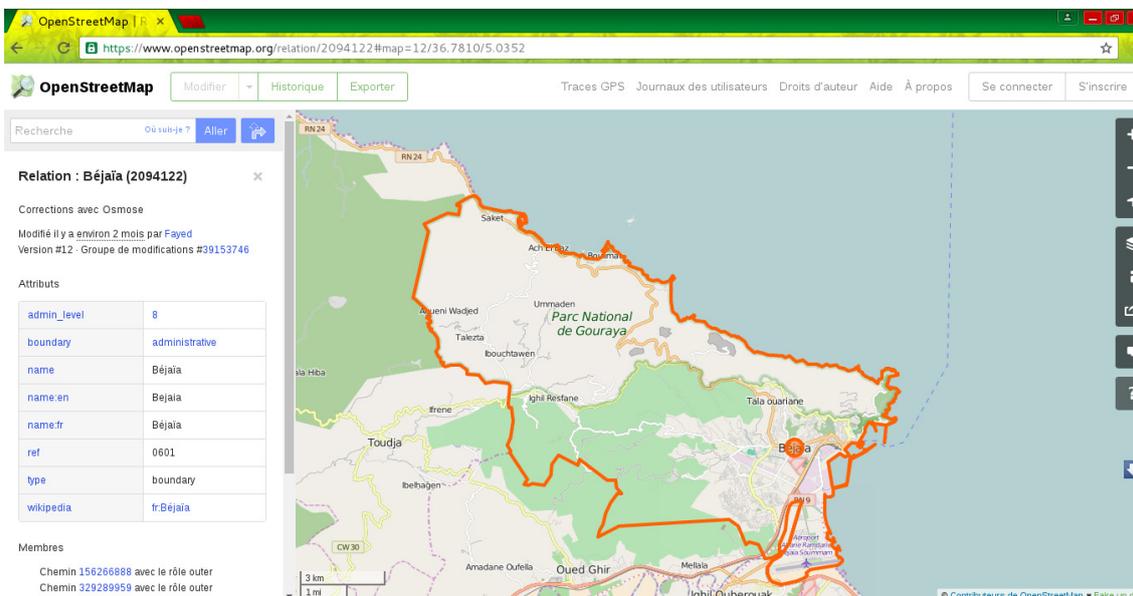
Annexe

# A.1 Annexe A

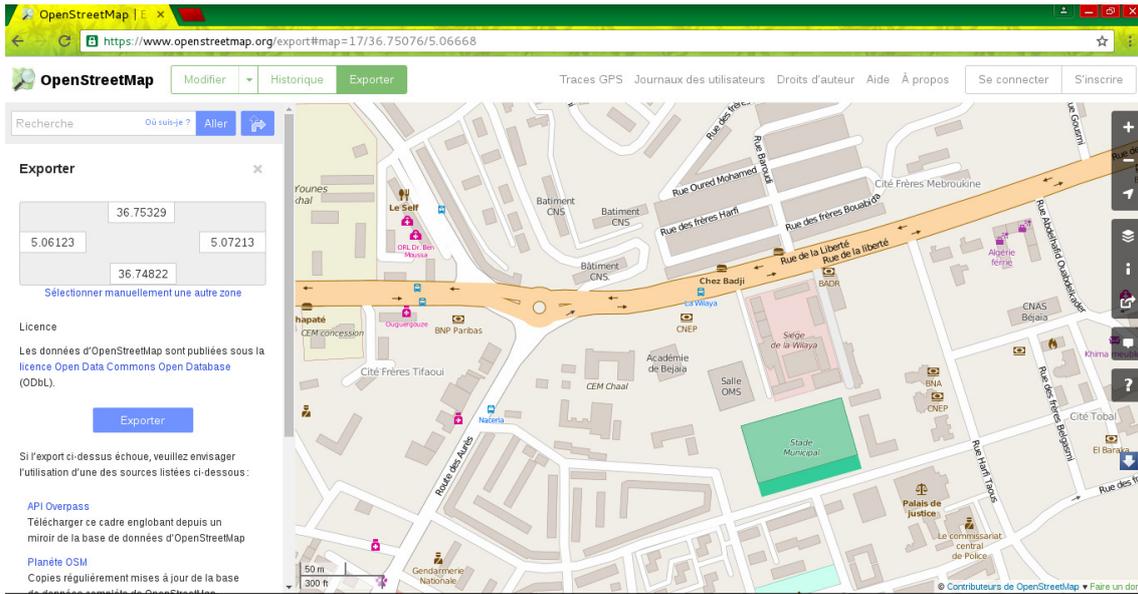
## A.1.1 téléchargement d'une carte map sur le site openstreetmap.com



Page accueil du site



On fait une recherche de l'endroit qu'on veut choisir (dans notre cas en fait une recherche sur la ville de béjaïa).



ensuite on clique sur export pour sauvegarder la carte.

# Bibliographie

- [1] R. Mazot, W. Meslem, M. Layouni, and A. Tran, *Communication inter véhiculaire*. PhD thesis, Arles Avignon, 2013.
- [2] S. Busanelli, G. Ferrari, and L. Veltri, “Conférence,” in *Short-lived Key Management for Secure Communications in VANETs*, 2011.
- [3] X. Lin, R. Lu, C. Zhang, H. Zhu, P. Ho, and X. Shen, *Security in vehicular ad hoc networks*. IEEE Communications Magazine, 2008.
- [4] K. Moghraoui, *Gestion de l’anonymat des communications dans les réseaux véhiculaires AD HOC sans fil (VANETs)*. PhD thesis, l’université du Québec à trois-rivieres, 2015.
- [5] J. Petit, *Surcoût de l’authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires*. PhD thesis, l’université de Toulouse, 2011.
- [6] D. Bektache, *Application et Modélisation d’un protocole de communication pour la sécurité routière*. PhD thesis, l’université de Badji Mokhtar Annaba, 2014.
- [7] M. JERBI, *Protocoles pour les communications dans les réseaux de véhicules en environnement urbain : Routage et GeoCast basés sur les intersections*. PhD thesis, l’université d’Evry Val d’Essonne, 2008.
- [8] P.Plainchault, *sécurisation de la conduite par communication véhicule infrastructure a base de transpondeurs*. PhD thesis, institut national polytechnique de toulouse, 2005.

- [9] F. Belarbi, *les systemes de communication entre les vehicules et l'infrastructure : leur contribution aux pratiques d'exploitation de la route*. PhD thesis, l'Ecole Nationale des Ponts et Chaussées, 2004.
- [10] Y-L. Morgan, *Novel Issues in DSRC vehicular communication radios*. PhD thesis, University of Regina, 2010.
- [11] A. A. Kahina, *Modélisation et étude de performances dans les réseaux VANET*. PhD thesis, l'université de Technologie de Belfort-Montbéliard, 2012.
- [12] X. Zeng, C. Tao, and Z. Chen, "Conférence," in *The application of DSRC Technology in Intelligent Transportation System*, 2009.
- [13] R. Uzcategui and G. Acosta-Marum, *Wave : A tutorial*. *IEEE Communications Magazine*. IEEE Communications Magazine, 2009.
- [14] A. Rao, *Performance Evaluation of Secure Communication in Vehicular Networks*. PhD thesis, Indian Institute of Technology Delhi, 2009.
- [15] F. Kaisser, C. Johnen, and V. Vèque, *Vers une modélisation robuste des protocoles de routage pour réseaux de véhicules* Manuscrit auteur. CFIP'2009 inria-00419460, 2009.
- [16] H. Michaël, *Contributions à l'étude des gestionnaires de services distribués dans les réseaux ad hoc*. PhD thesis, l'Université des Sciences et Technologies de Lille, 2005.
- [17] A. A. Ba, *Protocole de routage basé sur des passerelles mobiles pour un accès Internet dans les réseaux véhiculaires*. PhD thesis, l'université de Montréal, 2011.
- [18] L. Qabajeh, L. Kiah, and M. Qabajeh, *A qualitative comparison of position-based routing protocols for ad-Hoc networks*. *International Journal of Computer Science and Network Security*, 2009.
- [19] G. Badreddine, *Algorithme De Clusterisation Et Protocoles De Routage Dans Les Réseaux Ad Hoc*. PhD thesis, l'université de Technologie de Belfort-Montbéliard Tunisie, 2012.
- [20] C. Thomas and J. Philippe, *Optimized Link State Routing(OLSR)*. RFC 3626, 2003.

- [21] J. B. David, A. M. David, and B. Josh., *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728 (Experimental), 2007.
- [22] L. Barrere, *Etude et proposition de service dans les réseaux mobiles*. PhD thesis, l'université Bordeaux, 2009.
- [23] B. Hakim, G. Ignacy, and A. A. Khaldoun, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728 (Experimental), 2007.
- [24] M. Yassine, *Routage Dans Les Réseaux Vehiculaires (Vanet) Cas D'un Environnement Type Ville*. PhD thesis, l'université M'Hamed Bougara – Boumerdes, 2011.
- [25] C. E. Perkins and P. Bhagwat, “Conférence,” in *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*, 1994.
- [26] J. Z. Haas, R. M. Pearlman, and P. Samar, *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*. Internet-draft, IETF MANET Working Group, 2002.
- [27] J. Z. Haas, R. M. Pearlman, and P. Samar, *The Intrazone Routing Protocol (IARP) for Ad Hoc Networks*. Internet-draft, IETF MANET Working Group, 2002.
- [28] J. Z. Haas, R. M. Pearlman, and P. Samar, *The Interzone Routing Protocol (IERP) for Ad Hoc Networks*. Internet-draft, IETF MANET Working Group, 2002.
- [29] J. Z. Haas, R. M. Pearlman, and P. Samar, *The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks*. Internet-draft, IETF MANET Working Group, 2002.
- [30] E. C. Perkins, M. E. Belding-Royer, and S. R. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC k3561 (Experimental), 2003.
- [31] Z. Jing and C. Guohong, “Conférence,” in *VADD : Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks*, 2006.
- [32] W. Hao, F. Richard, G. Randall, and H. Michael, *MDDV : A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks*. ACM Workshop, 2004.
- [33] A. K. Saha and D. B. Johnson, “Conférence,” in *Modeling mobility for vehicular ad hoc Networks . Proceedings of the 1st ACM international workshop on vehicular ad hoc networks*, 2004.

- 
- [34] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. A. Wang, *Urban mobility models for vanets.*, 2006.
- [35] L. Wischhof and H. Rohling, “Conférence,” in *Congestion control in vehicular ad hoc networks*, 2005.
- [36] J. Moez, *Protocoles pour les communications dans les reseaux de vehicules en environnement urbain : Routage et GeoCast bases sur les intersections.* PhD thesis, L’universite D’evry Val D’essonne, 2008.
- [37] F. Granelli, G. Boato, and D. Kliazovich, *MORA : a Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks.* AutoNet, 2006.
- [38] M. Hamid, *Prédiction de Mouvement pour le Routage et le Contrôle d’Accès au Canal dans des Réseaux Sans-Fil Véhiculaires.* PhD thesis, l’Ecole Nationale Supérieur des Télécommunications de Paris, 2008.
- [39] J. Ledy, *Stratégie d’adaptation de liens sur canaux radios dynamiques pour les communications entre véhicules - Optimisation de la qualité de service.* PhD thesis, École doctorale de Sciences et ingénierie pour l’information, mathématiques, 2012.
- [40] opnet, “Généralités.” <http://www.opnet.com/>. Connecté.
- [41] qualnet, “Généralités.” <http://www.qualnet.fr/>. Connecté.
- [42] isi, “Généralités.” <http://www.isi.edu/nsnam/ns/>. Connecté.
- [43] openclassrooms, “Généralités.” <https://openclassrooms.com/courses/programmez-avec-le-langage-c>. Connecté.
- [44] A. Adigun, *Gestion de l’anonymat et de la traçabilité dans les réseaux véhiculaires sans fil.* PhD thesis, l’université du Québec à trois-rivieres, 2014.
- [45] H. Lu, J. Li, and M. Guizani, “Conférence,” in *A Novel ID-based Authentication Framework with Adaptive Privacy Preservation for VANETs*, 2012.
- [46] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, and M. Zitterbart, “Conférence,” in *Introducing Probabilistic Radio Propagation Models in OMNeT ++ Mobility Framework and Cross Validation Check with NS-2*, 2008.

- [47] kom, “Généralités.” <http://kom.aau.dk/group/05gr1120/ref/Channel.pdf>. Connecté.
- [48] nsnam, “Généralités.” <http://www.nsnam.org/>. Connecté.
- [49] G. sofiane, *Contribution à la qualité de service dans les réseaux vanet*. PhD thesis, l’université d’Oran, 2015.
- [50] vanet.eurecom, “Généralités.” <http://vanet.eurecom.fr/>. Connecté.
- [51] canu.informatik, “Généralités.” <http://canu.informatik.uni-stuttgart.de/mobisim/>. Connecté.
- [52] neo, “Généralités.” <http://neo.lcc.uma.es/staff/jamal/vanet/?q=node/9>. Connecté.
- [53] sumo.sourceforge, “Généralités.” <http://sumo.sourceforge.net/>. Connecté.
- [54] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rossel, *SUMO (Simulation of urban mobility)*. Middle East Symposium on Simulation and Modelling, 2002.
- [55] nsl.csie.nctu, “Généralités.” <http://nsl.csie.nctu.edu.tw/nctuns.html>. Connecté.
- [56] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, *TraNS : realistic joint traffic and network simulator for VANETs*. SIGMOBILE, 2008.

## Résumé

Les réseaux véhiculaires représentent aujourd'hui un intérêt certain pour l'industrie automobile, les opérateurs des réseaux, les organisations et même les particuliers. Ils représentent une des briques de base sur lesquelles vont se fonder les systèmes de transport intelligents.

Le but des systèmes de transport intelligents, de nos jours, est d'améliorer la sécurité et l'efficacité des transports routiers afin de diminuer les accidents et fournir un environnement confortable aux conducteurs et à leurs passagers.

Afin de choisir le protocole de routage qui convient le mieux aux VANETs parmi d'autres protocoles Ad hoc, ces derniers doivent être appliqués, un par un, sur le réseau.

Mais vu le coût de réalisation concrète d'un VANET, et afin de pouvoir produire différents scénarios de tests, la simulation devient nécessaire avant de mettre ce système réellement en œuvre.

Notre travail consiste à créer un simulateur pour les VANETs qui prend en considération les contraintes de la mobilité véhiculaire, pour cela on a utilisé quelques fonctionnalités de protocole de routage A-STAR qui fait cette mobilité des véhicules dans un environnement spécifique (carte géographique).

Mots clés : VANET, Ad hoc, la mobilité véhiculaire, simulation, réseau, routage.

## Abstract

Vehicular networks today are of interest to the automotive industry, network operators, organizations and even individuals. They represent one of the building blocks on which will be based intelligent transport systems.

The goal of intelligent transport systems, nowadays, is to improve the safety and efficiency of road transport in order to reduce accidents and provide a comfortable environment for drivers and their passengers.

To choose the routing protocol that is best suited VANETs among other ad hoc protocols, these must be applied one by one, on the network.

But considering the cost of concrete realization of a VANET, and in order to produce different test scenarios, simulation becomes necessary before putting the system actually implemented.

Our work is to create a simulator for VANETs that takes into account the constraints of vehicular mobility for which was used a routing protocol features A-STAR makes this vehicle mobility in an environment specific (map). . . .

Keywords : VANET, Ad hoc, vehicular mobility, simulation, network routing.