

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université Ain Témouchent Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département des Mathématiques et de l'Informatique



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en Informatique
Option : Réseaux et Ingénierie des Données (RID)

Thème

Modèles de mobilité pour les VANETs dans les villes intelligentes.

Présenté par :

- 1) M. Abderrahmane ZEROUAL
- 2) M. Farouk Salah Eddine ZOUHIRI

Devant le jury composé de :

Dr M. MEDEDJEL	UAT.B.B (Ain Témouchent)	Président
Dr N. ABDERRAHIM	UAT.B.B (Ain Témouchent)	Examinateur
Dr A. BENZERBADJ	UAT.B.B (Ain Témouchent)	Encadreur

Année Universitaire 2022/2023

Dédicace

Je souhaite dédier humblement ce travail à mes chers et respectés parents, en particulier à ma mère qui a toujours veillé sur mes études. Je tiens également à inclure mon frère, ma soeur, ma grand-mère, mes tantes, mes oncles, ainsi que tous mes amis et cousins(es) dans cette dédicace.

Abderrahmane

Dédicace

Je souhaite dédier ce travail modeste à mes grands-parents et à mes chers parents, en signe de gratitude et de reconnaissance pour les sacrifices qu'ils ont consentis pour ma réussite et pour avoir illuminé mon chemin de leurs précieux conseils. J'espère sincèrement pouvoir un jour leur rendre ne serait-ce qu'une partie de ce qu'ils ont fait pour moi. Que Dieu leur accorde bonheur et longévité.

Je tiens également à dédier ce travail à mon frère Soufienne, ainsi qu'à mes chers cousins Zakaria, Abdel, Adel, Nadir, Akram, Rafik, Sidi Mohamed, Yasser, à mes cousines et à mes tantes.

Enfin, je souhaite dédier ce travail à tous mes enseignants et à tous mes amis qui m'ont soutenu et accompagné tout au long de ce parcours.

Farouk Salah Eddine

Remerciements

Avant tout, nous exprimons notre gratitude envers ALLAH, le tout puissant, pour nous avoir accordé la force et la volonté nécessaires pour mener à bien ce travail.

Nous souhaitons exprimer notre immense reconnaissance à notre encadrant, M. A. BENZERBADJ, pour avoir accepté de superviser ce travail. Ses précieuses contributions, son encouragement et ses conseils constructifs ont été d'une grande aide pour mener ce projet à bien.

Nous tenons à remercier chaleureusement M. M. MEDEDJEL, Maître de Conférences à l'Université d'Ain Témouchent BELHADJ Bouchaib, d'avoir accepté de présider le jury et d'évaluer notre projet de fin d'études.

Nos remerciements vont également à Mme N. ABDERRAHIM, Maître de Conférences à l'Université d'Ain Témouchent BELHADJ Bouchaib, d'avoir accepté d'examiner notre travail.

Nos remerciements vont également à l'ensemble de nos enseignants du département des Mathématiques et Informatique de l'Université d'Ain Témouchent BELHADJ Bouchaib.

Enfin, nous souhaitons exprimer nos sincères remerciements à nos familles et à nos proches qui nous ont soutenus et encouragés tout au long de ce projet.

Résumé

Les réseaux véhiculaires, communément appelés Vehicular Ad-Hoc NETWORKS (VANETs), jouent un rôle très important pour éviter les accidents, grâce à des messages d'alerte.

Pour comprendre comment fonctionnent ces réseaux et évaluer leur performance, nous avons utilisé la simulation. Nous avons évalué la performance de réseaux VANETs de différentes tailles en utilisant différents modèles de mobilité. D'autre part, nous avons évalué la performance de la communication inter-véhiculaire en utilisant 3 protocoles de routage, à savoir AODV, DSDV et OLSR.

Les résultats obtenus sont encourageants.

simulations avec des modèles de mobilité. Les réseaux VANETs ont besoin de certains paramètres pour se déplacer sur les routes, Les modèles de mobilité sont utilisés dans la simulation des réseaux VANETs. Ils permettent de représenter les mouvements des véhicules sur les routes et d'évaluer les performances du réseau dans différents scénarios.

L'objectif de ce travail est de :

- Étudier les paramètres impliqués dans les modèles de mobilité pour les réseaux VANETs.
- Étudier les outils de simulation pour les réseaux VANETs.
- Mettre en place les programmes de simulation NS-2, NS-3 et SUMO pour des exemples de scénarios.

Mots clés : Villes Intelligentes, VANETs, Modèles de Mobilité, Protocoles de Routage, SUMO, NS-3, AODV, DSDV, OLSR.

Abstract

The objective of this study is to understand the functioning of vehicular networks, commonly known as Vehicular Ad-Hoc Networks (VANETs), and evaluate their performance using simulation. We evaluated the performance of VANETs with different sizes using various mobility models. Additionally, we assessed the performance of inter-vehicle communication using three routing protocols : AODV, DSDV, and OLSR.

The obtained results are promising.

Simulations were conducted using mobility models. VANETs require specific parameters to navigate on the roads, and mobility models are utilized in VANET simulation. These models enable the representation of vehicle movements on the roads and facilitate the evaluation of network performance in different scenarios. The objective of this work is to :

- Study the parameters involved in mobility models for VANETs.
- Study simulation tools for VANET.
- Set up NS-2, NS-3, and SUMO simulation programs for example scenarios with mobile devices.

Keywords : Smart Cities, Mobility models, VANETs, Routing Protocols, SUMO, NS-3, AODV, DSDV, OLSR.

ملخص

ستلعب شبكات السيارات دورًا مهمًا جدًا في تجنب الحوادث. من خلال رسائل التحذير، ستتمكن هذه الشبكات من المساعدة في تجنب التصادمات. في دراستنا، قمنا بدراسة الشبكات السيارات، التي تثير اهتمامًا متزايدًا في مجال البحث والتطوير. قمنا بتقييم أداء الاتصال بين المركبات باستخدام ثلاثة بروتوكولات توجيه.

سننتقل في هذه الدراسة إلى الشبكات السيارات ودورها المتزايد في مجال البحث والتطوير. ستساعد هذه الشبكات في تجنب الحوادث من خلال إرسال رسائل تحذيرية.

تم الحصول على نتائج واعدة.

لفهم كيفية عمل هذه الشبكات وتقييم أدائها، قمنا باستخدام التجارب المحاكية. تحتاج شبكات السيارات إلى بعض المعلومات للانتقل على الطرق، بالإضافة إلى وجود أجهزة اتصال وتكرار حركة وبرامج قابلة للتكيف.

هدف هذا العمل هو :

- دراسة المعلومات المشاركة في نماذج الحركة لشبكات السيارات.

- دراسة أدوات المحاكاة لشبكات السيارات.

- إعداد برامج المحاكاة: برنامج محاكاة سومو، برنامج محاكاة نس-2 ، برنامج محاكاة نس-3 لأمثلة من السيناريوهات مع أجهزة متنقلة .

الكلمات المفتاحية: نماذج الحركة، شبكات السيارات، اسم برنامج محاكاة سومو، اسم برنامج محاكاة نس-2 ، اسم برنامج محاكاة نس-3

Table des matières

Introduction générale	14
1 GÉNÉRALITÉS SUR LES RÉSEAUX VEHICULAIRES	16
1.1 Introduction	16
1.2 Qu'est ce qu'un réseau ?	16
1.2.1 Réseau sans fil	17
1.2.2 Architecture d'un réseau sans fil	17
1.3 Réseaux MANETs	19
1.3.1 Définition d'un réseau MANET	19
1.3.2 Domaines d'utilisation d'un réseau MANET	19
1.4 Réseau VANET	20
1.4.1 Définition d'un réseau VANET	20
1.4.2 Nœud d'un réseau VANET	22
1.4.3 Composants d'un réseau VANET	22
1.4.4 Communications dans les réseaux VANETs	24
1.4.5 Caractéristiques des réseaux VANETs	25
1.4.6 Les projets de recherches sur les VANETs	25
1.5 Conclusion	27
2 LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE	28
2.1 Introduction	28
2.2 Les modèles de mobilité	29

2.2.1	Classification des modèles de mobilité	29
2.2.2	Exemple des modèles de mobilité	31
2.2.3	Le choix d'un type de modèle de mobilité	38
2.3	Les modèles de trafic	39
2.3.1	Modèle macroscopique	39
2.3.2	Modèle microscopique	39
2.3.3	Modèle mésoscopique	40
2.3.4	Modèle de trafic Random Trips	40
2.4	Conclusion	41
3	IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS	42
3.1	Introduction	42
3.2	Objectif de la simulation	42
3.3	Simulation de modèle de mobilité et les protocoles de routages . .	43
3.3.1	Simulation de modèle de mobilité	43
3.3.2	Simulation des protocoles de routages	43
3.3.3	Simulateurs de mobilité et du trafic pour les VANET . . .	47
3.3.4	Simulateurs de mobilité et du trafic utilisés	47
3.4	Les étapes de la simulation	48
3.4.1	Étape 1 : Simulation SUMO	48
3.4.2	Étape 2 : simulation sous NS-3	60
3.5	Conclusion	79
	Conclusion générale	80
	Annexe : Installation des outils	81
	Bibliographie	83

Liste des tableaux

3.1	Temps d'attente des véhicules.	58
3.2	Les paquets reçus avec le protocole AODV	62
3.3	Les paquets reçus avec le protocole OLSR.	64
3.4	Les paquets reçus avec le protocole DSDV.	66

Table des figures

1.1	Réseau sans fil : Mode avec infrastructure.	17
1.2	Réseau sans fil : Mode sans infrastructure.	18
1.3	Exemple de réseau VANET [1].	21
1.4	Véhicule intelligent [2].	22
1.5	Les RSUs [3].	24
1.6	Communications Véhicule à Infrastructure (V2I) [4].	24
1.7	Communications Véhicule à Véhicule (V2V) [4].	25
2.1	Modèle de freeway [5].	37
2.2	Modèle de Manhattan [5].	38
3.1	Classification des protocoles de routage dans les réseaux MANETs [4].	44
3.2	La classification des protocoles de routage dans les réseaux VANETs [4].	46
3.3	Exemple de création une Map manuelle [6].	51
3.4	Exemple de création d'une map manuelle [6].	51
3.5	Exemple de la création d'une map	52
3.6	Convertir les données en format SUMO	53
3.7	Les étapes de la simulation	55
3.8	Analyse des délais d'attente des véhicules sous forme d'un histogramme.	58
3.9	Code d'analyse de performance pour le programme vanet 1	61
3.10	Code d'analyse de performance pour le programme vanet 2. . . .	61
3.11	Code de la Configuration de 2ème scénario	62

3.12	Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole AODV à travers un histogramme.	63
3.13	Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole OLSR à travers un histogramme.	64
3.14	Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole DSDV à travers un histogramme.	66
3.15	Analyse graphique des performances de 20 véhicules avec le protocole AODV.	68
3.16	Analyse graphique des performances du 20 véhicules avec le protocole OLSR.	70
3.17	Analyse graphique des performances du 20 véhicules avec le protocole DSDV.	71
3.18	Analyse graphique des performances du 40 véhicules avec le protocole AODV.	72
3.19	Analyse graphique des performances du 40 véhicules avec le protocole OLSR.	73
3.20	Analyse graphique des performances du 40 véhicules avec le protocole DSDV.	74
3.21	Analyse graphique des performances du 60 véhicules avec le protocole AODV.	76
3.22	Analyse graphique des performances du 60 véhicules avec le protocole OLSR.	77
3.23	Analyse graphique des performances du 60 véhicules avec le protocole DSDV.	78

Liste des Abréviations

- AODV** : Ad-Hoc On-Demand Distance Vector
- AP** : Access Point
- CGSR** : Cluster Gateway Switch Routing
- DSDV** : Destination-Sequenced Distance-Vector Routing
- DSR** : Dynamic Source Routing
- DREAM** : Dynamic Resource Allocation with Mobility Prediction
- FSLs** : Full-Spectrum Link State
- FSR** : Flow State Routing
- GGAR** : Geographic Greedy Adaptive Routing
- GSR** : Global State Routing
- GRP** : Geographic Routing Protocol
- GyTAR** : Geographic and Temporal-Aware Routing
- HELLO** : Message utilisé pour détecter les voisins directs
- HSR** : Hierarchical State Routing
- IDM** : Intelligent Driver Model
- I2I** : Infrastructure-to-Infrastructure
- LAN** : Local Area Network
- LANMAR** : Landmark Ad-Hoc Routing
- MANET** : Mobile Ad-Hoc Network
- MORA** : Mobility and Route Optimization in Ad-Hoc Networks
- MOVE** : Mobility Model and Vehicle Environment

NS-3 : Network Simulator 3

NOW : Network On Wheels

OBU : On Board Unit

OLSR : Optimized Link State Routing

OMNET++ : Objective Modular Network Testbed in C++

OSM : OpenStreetMap

OSMWebWizard : OpenStreetMap Web Wizard

PDA : Personal Digital Assistant

RPGM : Reference Point Group Mobility Model

RSU : Road Side Unit

RREP : Route REPLY

RREQ : Route REQuest

STI : Systèmes et Technologies de l'Information

SUMO : Simulation of Urban Mobility

TC : Topology Control

TBRF : Time-Based Routing Function

TIC : Les technologies de l'information et de la communication

UTRA-TDD : UMTS Terrestrial Radio Access with Time Division Duplexing

V2I : Vehicle-to-Infrastructure

V2V : Vehicle-to-Vehicle

VANET : Vehicular Ad-Hoc Network

Veins : Vehicular Network Simulation Framework

WAN : Wide Area Network

WAVE : Wireless Access in Vehicular Environment

WLAN : Wireless Local Area Network

WiFi : Wireless Fidelity

ZRP : Zone Routing Protocol

Introduction générale

L'utilisation de la technologie sans fil transforme actuellement le domaine des réseaux de communication. Les progrès de cette technologie ont fait des réseaux de communication sans fil l'un des domaines les plus actifs de la recherche en informatique.

Les réseaux VANETs permettent aux véhicules de communiquer entre eux via des connexions sans fil pour faciliter les communications entre les véhicules, ainsi qu'entre les véhicules et les infrastructures routières. Malheureusement, VANET ne peut pas être utilisé directement sur le terrain pour la recherche. Il faut donc recourir à la simulation. La recherche dans ce domaine est actuellement très active. Les applications des réseaux de véhicules sans fil sont très diverses.

Les véhicules se déplacent au sein du réseau VANET en suivant des modèles de mobilité spécifiques. Afin de réaliser une simulation, plusieurs modèles de mobilité ont été développés pour représenter le mouvement des véhicules dans le réseau VANET.

Plan du mémoire

Le travail présenté dans ce mémoire se compose d'une introduction générale, de trois chapitres traitant de divers aspects des réseaux Ad-Hoc, en particulier les réseaux VANETs (Vehicle Ad-Hoc Networks), et d'une conclusion générale.

Le premier chapitre donne un aperçu général sur les réseaux Ad-Hoc. Il passe en revue les réseaux sans fil, leur classification et leur architecture, puis se concentre sur les réseaux MANETs pour discuter de leurs caractéristiques et domaines d'application. Enfin, il définit les réseaux VANETs.

Le chapitre 2 se concentre sur les modèles de mobilité et de trafic utilisés dans les VANETs. Il présente les différents modèles de mobilité, à savoir, les modèles basés sur les traces, sur les enquêtes, sur les simulateurs de trafic et les modèles synthétiques. D'autre part, il introduit les modèles de trafic en les classifiant en modèles macroscopiques, microscopiques et mésoscopiques. Ce chapitre met en effet l'accent sur l'importance de ces modèles dans la simulation des réseaux VANETs.

Le troisième chapitre est dédié à l'implémentation de notre modèle de simulation sous SUMO et NS3 ainsi que les différentes simulations réalisées. Nous rappelons que nous avons considéré des VANETs avec différentes tailles et qui utilisent trois modèles de mobilité et un modèle de trafic (RandomTrips). En matière de routage, nous avons testé trois protocoles de routage pour les réseaux Ad-Hoc (AODV, OLSR, DSDV). Les résultats des différentes simulations réalisées sur les différents scénarios considérés sont discutés en fin de ce chapitre.

Une conclusion générale avec précision de nos travaux futurs clôture le mémoire qui comporte une annexe décrivant toutes les commandes nécessaires à l'installation et l'exploitation des différents outils employés pour la réalisation de ce travail.

Chapitre 1

GÉNÉRALITÉS SUR LES RÉSEAUX VEHICULAIRES

1.1 Introduction

Les réseaux VANETs constituent un sous-ensemble des réseaux MANETs. Un VANET est un ensemble de véhicules équipés de dispositifs de communication sans fil pouvant créer un réseau Ad-Hoc pour communiquer les uns avec les autres et avec l'infrastructure routière. Ce genre de réseaux est utilisé pour fournir aux conducteurs des informations de sécurité spécifiques à la route, telles que la notification des accidents et des conditions de route dangereuses.

Dans ce chapitre, nous présentons les réseaux Ad-Hoc, les réseaux MANETs puis nous détaillons les réseaux VANETs.

1.2 Qu'est ce qu'un réseau ?

Un réseau peut être décrit comme un ensemble de noeuds connectés les uns aux autres par des liens de communication filaires ou radio. Le but des réseaux est le partage de données et de ressources entre les différents noeuds. Les réseaux peuvent comprendre des sous-réseaux et peuvent également être interconnectés avec d'autres réseaux. Les réseaux informatiques ont une grande importance dans divers domaines, notamment dans l'industrie et l'automobile [7].

1.2.1 Réseau sans fil

Un réseau sans fil est un type de réseau dans lequel les appareils participants tels que les ordinateurs portables, les PDAs et les téléphones portables ne sont pas connectés via des câbles les uns aux autres. Au lieu de cela, la transmission des données se fait via des ondes hertziennes, qu'elles soient de type radio ou infrarouge. Cette technologie permet aux utilisateurs de se déplacer dans une zone de couverture allant de quelques mètres à plusieurs kilomètres, tout en restant connectés au réseau et en échangeant des données avec les autres appareils du réseau [8].

1.2.2 Architecture d'un réseau sans fil

Mode avec infrastructure

Les stations d'un réseau sans fil avec infrastructure se connectent à un point d'accès (AP), également appelé station de base, qui partage la bande passante disponible entre eux [9]. Ce mode de fonctionnement est représenté dans la Figure 1.1.

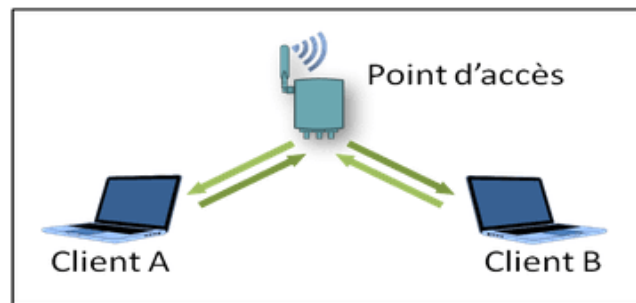


FIGURE 1.1 – Réseau sans fil : Mode avec infrastructure.

Mode sans infrastructure ou mode Ad-Hoc

Les réseaux de ce type ne nécessitent pas de point d'accès pour fonctionner, car les stations communiquent directement entre elles. Chaque station du réseau agit comme un routeur pour la communication entre les différentes stations. Ce

CHAPITRE 1. GÉNÉRALITÉS SUR LES RÉSEAUX VEHICULAIRES

type de réseau est auto-organisé [9]. Ce mode de fonctionnement est représenté dans la Figure 1.2.

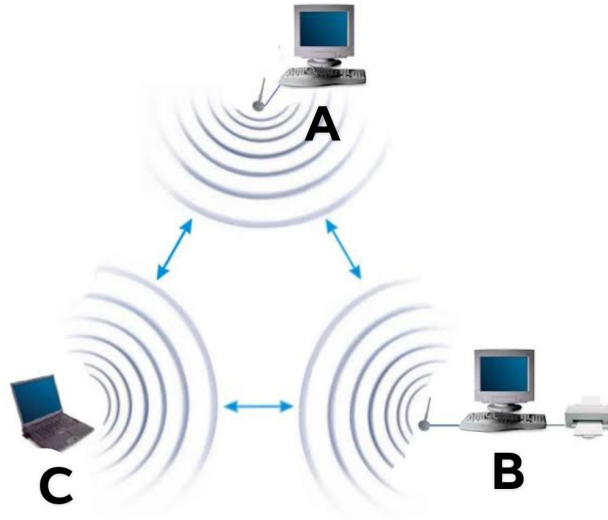


FIGURE 1.2 – Réseau sans fil : Mode sans infrastructure.

1.3 Réseaux MANETs

1.3.1 Définition d'un réseau MANET

Un réseau MANET (Mobile Ad-Hoc Network), est un réseau dynamiquement formé par un grand nombre de stations mobiles, qui se connectent sans utiliser une infrastructure existante et utilisent des interfaces sans fil telles que les ondes radio pour communiquer. Les stations interagissent et peuvent collaborer pour échanger des services, tout en étant libres de se déplacer et de s'organiser arbitrairement, ce qui implique une grande variabilité de la topologie du réseau. Chaque station est capable de communiquer directement avec ses voisins dans la zone de portée de leur antenne, et peut également servir de relais pour d'autres stations du réseau, leur permettant ainsi de communiquer avec des stations plus éloignées [10]. La gestion du réseau doit être distribuée à travers différentes stations, ce qui rend la détection et la gestion des pannes plus difficiles. En raison de la nature multi-sauts et de l'absence d'infrastructure fixe, ces réseaux sont confrontés à des défis tels que l'acheminement multi-sauts, les topologies de réseau changeantes et imprévisibles, la variation des capacités des liens et des noeuds, le fonctionnement énergétique contraint et la mise à l'échelle du réseau pour une grande quantité de noeuds [11].

1.3.2 Domaines d'utilisation d'un réseau MANET

Les réseaux MANETs sont utilisés dans plusieurs applications [12] :

Applications militaires : Les réseaux MANETs sont utilisées dans les applications militaires où la communication rapide et fiable entre les troupes est cruciale. Ils sont utilisés pour la coordination des opérations sur le terrain, la surveillance et la collecte de renseignements.

Applications de recherche scientifique : Les réseaux MANETs sont utilisés dans les applications de recherche scientifique pour collecter des données. Les données peuvent être utilisées pour l'étude de la météorologie, la surveillance de l'environnement, la recherche géologique, etc.

Applications de communication d'urgence : Les réseaux MANETs peuvent être utilisés dans les situations d'urgence pour permettre une communication rapide et fiable entre les différents intervenants. Par exemple, en cas de

catastrophe naturelle, les réseaux MANETs peuvent être utilisés pour coordonner les opérations de secours et fournir des informations sur l'état des victimes.

Applications de transport intelligent : Les réseaux MANETs peuvent être utilisés pour améliorer la sécurité et l'efficacité du transport intelligent, en permettant une communication directe et en temps réel entre les véhicules sur la route.

Applications commerciales et résidentielles : Les réseaux MANETs peuvent être utilisés dans les applications commerciales et résidentielles pour fournir une connectivité sans fil flexible et peu coûteuse. Par exemple, les réseaux MANETs peuvent être utilisés pour fournir une connectivité Internet dans les zones rurales ou dans les bâtiments où le câblage est difficile ou coûteux à installer.

Applications de loisirs : Les réseaux MANETs peuvent être utilisés dans les applications de loisirs pour permettre la communication entre les différents appareils et utilisateurs. Par exemple, les réseaux MANETs peuvent être utilisés pour permettre aux joueurs de jeux vidéo de jouer ensemble sans avoir besoin d'une connexion Internet ou d'un serveur centralisé.

1.4 Réseau VANET

1.4.1 Définition d'un réseau VANET

Les VANETs sont des réseaux de communication sans fil formés par des véhicules équipés de dispositifs de communication sans fil qui leur permettent de communiquer directement entre eux et avec les infrastructures routières. La Figure 1.3 montre un exemple d'un réseau VANET.

Les VANETs peuvent être utilisés pour fournir des services, tels que la diffusion d'alertes de sécurité routière, la surveillance de la circulation, les services de divertissement en voiture, les services d'information et de navigation, les services de paiement, les stationnements et les recharges électriques.

Cependant, les VANETs sont confrontés à de nombreux défis techniques, tels que la confidentialité des données, la tolérance aux pannes, la gestion des ressources du réseau, la conception de protocoles de routage efficaces et fiables, et la coordination de la communication entre les véhicules et les infrastructures

routières [1].

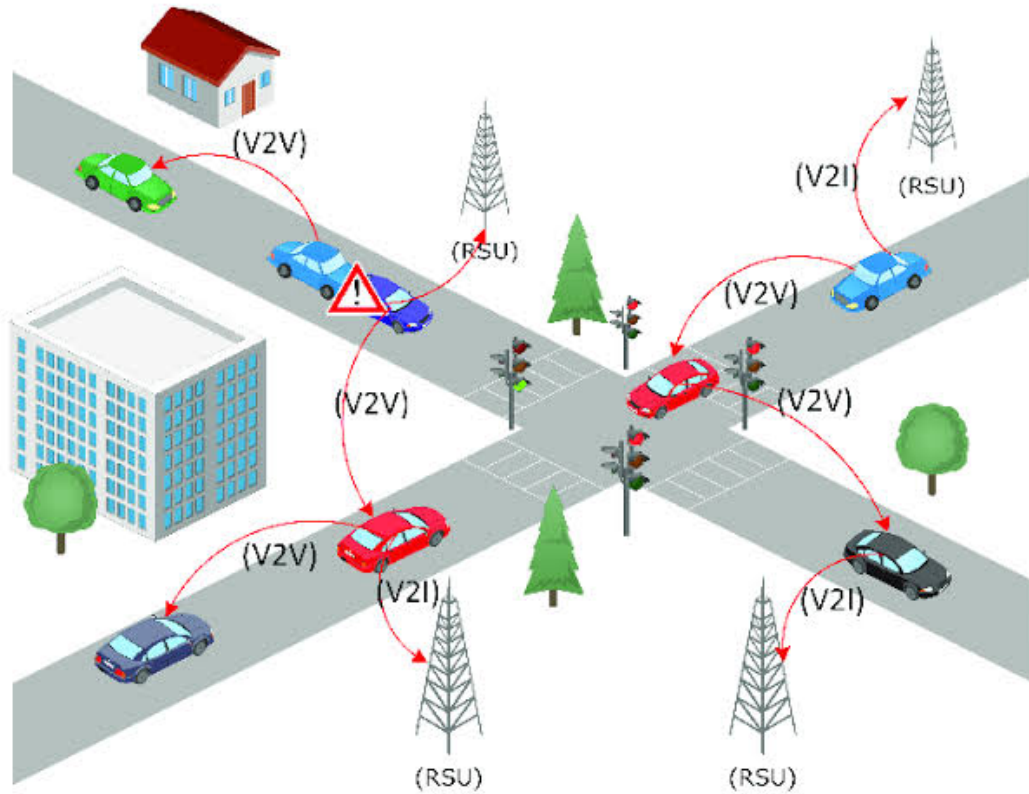


FIGURE 1.3 – Exemple de réseau VANET [1].

1.4.2 Nœud d'un réseau VANET

Dans un réseau de véhicules, un nœud est un élément du réseau qui est constitué d'un véhicule intelligent équipé de différents terminaux, capteurs et calculateurs pour collecter et traiter les informations [13].

Les calculateurs sont des ordinateurs embarqués dans le véhicule qui sont capables de traiter les données collectées par les capteurs. Ces calculateurs permettent de faire fonctionner les différentes applications qui sont utilisées par les conducteurs et les passagers.

Les capteurs sont des dispositifs qui permettent de collecter des informations sur l'environnement du véhicule, telles que la vitesse, la température, la pression, l'humidité, la luminosité, la présence d'autres véhicules ou de piétons. Ces capteurs peuvent être intégrés dans le véhicule ou être des dispositifs externes. La Figure 1.4 représentée un exemple de véhicule intelligent. La Figure 1.4 illustre un exemple d'un véhicule intelligent.

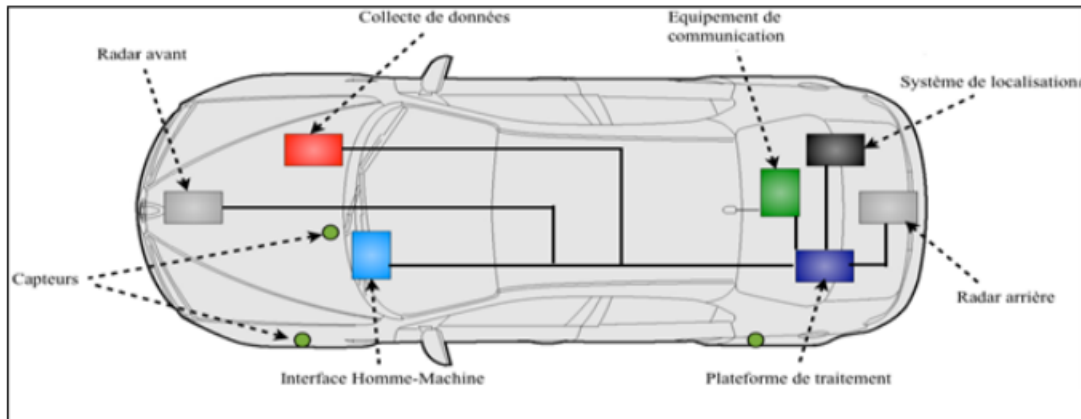


FIGURE 1.4 – Véhicule intelligent [2].

1.4.3 Composants d'un réseau VANET

Dans les VANETs, différents éléments permettent la communication et forment le dispositif sans fil appelé WAVE (Wireless Access in Vehicular Environment). Les principaux composants de ce système sont les AUs (Application Units), les OBU (On Board Units) et les RSUs (Road Side Units) [13, 14].

Les OBUs (On Board Unit)

Les OBUs sont des équipements mobiles qui permettent des communications en mode Ad-Hoc entre les différentes unités OBU, ainsi que des communications en mode infrastructure avec les unités RSU. Ils sont également équipés d'une gamme de capteurs et d'actionneurs, qui permettent de surveiller efficacement les véhicules et de collecter des informations telles que la vitesse du véhicule et son accélération.

Les AUs (Application Unit)

Les AUs sont des périphériques intégrés à l'intérieur du véhicule, connectés à l'OBU, et dédiés aux applications liées à la sécurité, au confort et aux loisirs. Ils permettent notamment de fournir des informations de navigation, de divertissement et de sécurité aux passagers du véhicule.

Les RSUs (Road Side Unit)

Les RSUs sont des périphériques WAVE installés le long des routes et sur des infrastructures existantes telles que les feux de circulation, les panneaux routiers ou encore les lampadaires. Leur rôle est de connecter le véhicule au réseau commun, qui ensuite les connecte au coeur central du réseau. Les RSUs sont essentiels pour assurer la communication entre les différents véhicules telle que l'illustre la Figure 1.5. Les RSUs sont essentiels pour améliorer la sécurité routière en permettant une coordination efficace entre les véhicules et les infrastructures [3].

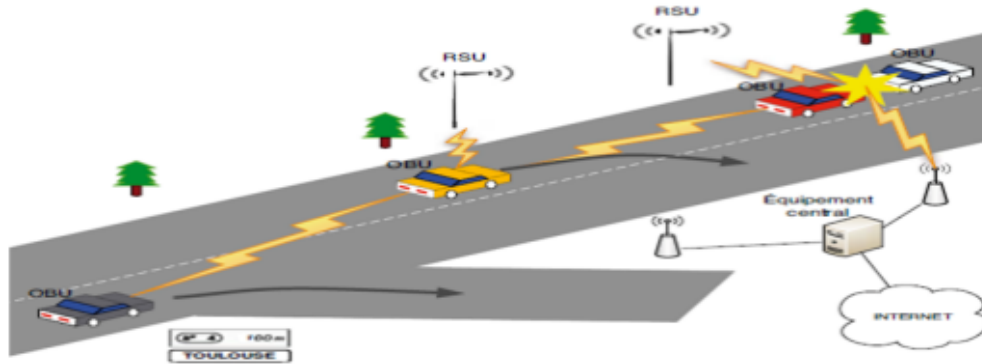


FIGURE 1.5 – Les RSUs [3].

1.4.4 Communications dans les réseaux VANETs

Communications Véhicule à Infrastructure (V2I)

Il s'agit d'une architecture centralisée basée sur des stations de base telle que l'illustre la Figure 1.6. Le noeud central peut être une station de base connectée à Internet, qui gère la distribution des messages aux véhicules et aux infrastructures de transport. Cette architecture est relativement simple à mettre en oeuvre, car elle ne nécessite pas de protocoles de routage sophistiqués. Cependant, elle peut entraîner des problèmes de latence et de fiabilité, car tous les messages doivent passer par le noeud central [15].

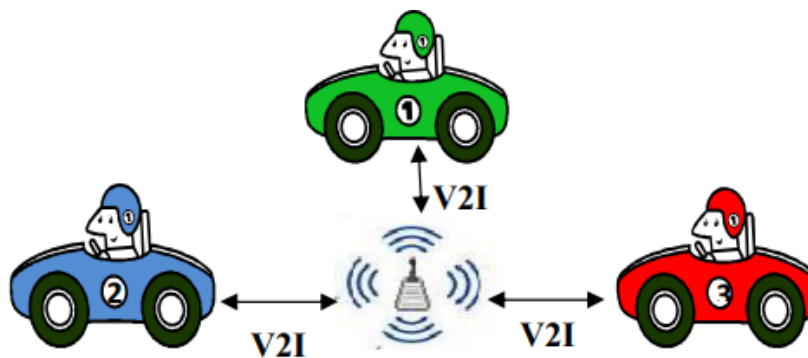


FIGURE 1.6 – Communications Véhicule à Infrastructure (V2I) [4].

Communications Véhicule à Véhicule (V2V)

Il s'agit d'une architecture décentralisée (voir Figure 1.7) où chaque noeud dans le réseau est responsable de la gestion de son propre routage et de la transmission des données. Les noeuds communiquent entre eux en utilisant des protocoles de routage Ad-Hoc. Cette architecture ne dépend pas d'un noeud central unique. Cependant, elle peut être plus complexe à mettre en oeuvre et peut nécessiter une gestion plus sophistiquée du réseau [15].



FIGURE 1.7 – Communications Véhicule à Véhicule (V2V) [4].

1.4.5 Caractéristiques des réseaux VANETs

Les réseaux VANETs sont des réseaux de communication avec des caractéristiques uniques qui les distinguent des autres formes de réseaux sans fil [5].

La forte mobilité : Les mouvements des véhicules peuvent être prévisibles, mais les changements topologiques sont fréquents en raison de l'entrée et de la sortie rapides des noeuds dans le réseau.

Energie : Contrairement à d'autres réseaux Ad-Hoc avec une énergie limitée, les VANETs ont une grande capacité pour l'énergie.

1.4.6 Les projets de recherches sur les VANETs

De nombreuses agences gouvernementales, entreprises commerciales et institutions de recherche académiques sont intéressées par la communication sans fil dans les transports, ce qui a entraîné la formation de nombreuses alliances et de grands projets. Voici quelques exemples remarquables dont nous avons jugé important de les citer :

Le projet AUTO21

Le projet a été Lancé en 2001, le projet AUTO21 a été achevé en 2015. Il comprenait plusieurs sous-projets, dont le projet VANET, qui se concentrait sur la communication sans fil entre les véhicules et sur les routes ainsi que sur la technologie de sécurité automobile.

L'objectif principal du projet était de créer un protocole de communication entre véhicules pour l'ajustement et le contrôle de la vitesse dans un système collaboratif [4].

Le projet FleetNet

L'objectif principal de ce projet est de fournir une plate-forme de communication "Internet sur la route" pour les réseaux de véhicules. Cette plate-forme utilise la méthode de communication UTRA-TDD (UMTS Terrestrial Radio Access with Time Division Duplexing) pour permettre la communication entre véhicules. Pour fournir aux véhicules une connectivité Internet, le protocole de routage utilisé tient compte de la topologie du réseau, de l'emplacement des véhicules et des communications V2I. La sécurité des conducteurs et des passagers est une préoccupation majeure pour le projet FleetNet, qui vise à accroître cette sécurité en fournissant des technologies de communication de pointe pour les réseaux de véhicules [16].

Le projet NOW(Network On Wheels)

Le gouvernement fédéral allemand chargé de la recherche et de l'éducation a remplacé le projet FleetNet par NOW (Network On Wheels), qui a été développé en partenariat avec des universités, des sociétés de télécommunications et des constructeurs automobiles. Il vise à mettre en oeuvre un protocole de communication et des algorithmes de sécurité des données pour un réseau mobile, en tenant compte des technologies sans fil telles que 802.11 et du routage basé sur la localisation dans les communications V2V ou V2I. L'objectif ultime est de travailler avec le consortium Car2Car pour mettre en place un système de référence et contribuer à la normalisation d'une telle solution en Europe [4].

1.5 Conclusion

Dans ce chapitre, nous avons présenté les réseaux MANETs qui sont des réseaux sans fil autonomes où chaque noeud peut communiquer avec d'autres noeuds sans l'aide d'une infrastructure. Nous avons discuté des domaines d'application des réseaux MANETs et ensuite, nous avons abordé les réseaux VANETs (Vehicular Ad-Hoc Networks), qui sont des réseaux Ad-Hoc créés spécifiquement pour les véhicules. Nous avons discuté de ses différentes parties et architectures. Nous avons également parlé des projets de recherche en cours dans ce domaine. Dans le prochain Chapitre, nous allons aborder les modèles de mobilité et de trafic pour les réseaux véhiculaires dans les villes intelligentes.

Chapitre 2

LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

2.1 Introduction

Une ville intelligente, également appelée “smart city” en Anglais, est une ville qui utilise les technologies de l’information et de la communication (TIC) pour améliorer la qualité de vie de ses habitants. Dans cette étude, nous allons discuter des modèles de mobilité et de trafic pour les réseaux VANETs. Ces modèles sont utilisés pour simuler et prédire les mouvements des véhicules dans les villes intelligentes, ainsi que pour évaluer les performances des protocoles de communication qui permet d’améliorer la sécurité routière et la gestion du trafic. Ces modèles aident à éviter les accidents et les embouteillages dans une ville intelligente

Nous allons classer les modèles de mobilité en fonction de leur source de données et de leur niveau de réalisme en quatre catégories : les modèles basés sur les traces (trace-based models), les modèles basés sur des enquêtes (Survey-based models), les modèles basés sur des simulateurs de trafic (Simulator-based models) et les modèles synthétiques (Synthetic models).

Les modèles de trafic peuvent être classés en trois catégories : les modèles macroscopiques, microscopiques et mésoscopiques.

Les modèles de mobilité et de trafic jouent un rôle important. Ils nous permettent de mieux comprendre comment la circulation se passe, où se trouvent les

voitures et comment elles se déplacent d'un endroit à un autre.

2.2 Les modèles de mobilité

Un modèle de mobilité est un ensemble de règles et de paramètres qui décrivent le comportement des noeuds mobiles dans un réseau sans fil. Il permet de simuler les déplacements des noeuds dans le réseau.

Un modèle de mobilité doit inclure une carte réelle qui reflète les différentes densités de routes et les différentes catégories de rues avec leurs limites de vitesse [17]. Le modèle de mobilité détermine les mouvements des noeuds dans la zone simulée pendant la durée de la simulation [18].

2.2.1 Classification des modèles de mobilité

Dans ce qui suit, nous allons explorer les différents types de modèles de mobilité.

Les modèles basés sur les traces (trace-based models)

Ces modèles de mobilité sont basés sur des statistiques collectées à partir de traces d'utilisateurs en temps réel. Ces traces sont obtenues grâce à des services de localisation par ordinateur. Ces modèles sont similaires à ceux utilisés pour modéliser la mobilité, mais ont été spécialement développés pour les véhicules. Cependant, ces modèles nécessitent des quantités importantes de données de traces de véhicules qui ne sont pas facilement disponibles. C'est pourquoi l'utilisation de ces modèles peut être limitée dans certaines situations [19].

Les modèles basés sur des enquêtes (Survey-based models)

Les modèles basés sur des données d'enquête utilisent des données statistiques ou des données d'enquête pour simuler la mobilité. Similaires aux modèles basés sur les traces, cette catégorie a l'avantage de modéliser les caractéristiques de mobilité qui seraient trop difficiles à modéliser en utilisant des approches mathématiques. Cependant, un inconvénient majeur de l'approche basée sur l'enquête est qu'elle ne peut pas produire des modèles de mouvement précis [19].

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

Les modèles basés sur des simulateurs de trafic (Simulator-based models)

Les modèles basés sur des simulateurs de trafic extraient des modèles de mobilité à partir de simulateurs de trafic. Ces simulateurs sont basés sur ordinateur et offrent des solutions précieuses en donnant un aperçu clair du scénarios [19].

Il existe plusieurs logiciels de simulation dédiés à la mobilité des véhicules, on trouve SUMO, qui est conçu pour la gestion du trafic et la simulation des communications entre véhicules. MOVE est développé sur la couche supérieure de SUMO pour générer des traces de mobilité réalistes. VanetMobiSim est un simulateur de trafic qui permet de modéliser des caractéristiques telles que la topologie des routes, les panneaux de signalisation et les points d'intérêt [19] .

Les modèles synthétiques(Synthetic models)

Les modèles synthétiques sont créés en utilisant des méthodes mathématiques ou statistiques. Ils sont basés sur des hypothèses et des paramètres pré-définis et sont souvent utilisés pour modéliser des scénarios. Dans ce qui suit, nous allons énumérer quelques exemples de modèles de mobilité synthétique utilisés :

Les modèles aléatoires (random models) Les modèles de mobilité aléatoire sont depuis longtemps les favoris pour la modélisation des modèles de mobilité aléatoires pour les applications informatiques ou de télécommunications [20].

Dans ces modèles, Les noeuds peuvent se déplacer vers n'importe quelle destination et leurs vitesses et directions sont choisies au hasard. Ces modèles sont construits en se basant sur des hypothèses idéales plutôt que sur des situations réelles. Mais dans le monde réel, les véhicules ne se déplacent pas au hasard sans destination [21].

Les modèles avec une dépendance temporelle (The models with temporal dependency) Les modèles avec dépendance temporelle sont des modèles qui prennent en compte le temps comme variable importante. Ils sont utilisés pour modéliser des situations où les données changent dans le temps, plus précisément, le modèle prend en compte les heures de jour et de nuit et les variations de circulation qui se produisent au fil des saisons (les vacances scolaires, les festivals), et la distance minimale qui sera atteinte entre deux véhicules sur le parcours de

leur mouvement [22].

Les modèles de flux (flow models)

La modélisation de flux de trafic dont l'objectif est de décrire de manière mathématique précise et prendre en compte les interactions physiques entre les véhicules, ils ont abordé le problème en modélisant la mobilité des véhicules sous forme de flux.

Le trafic passe par trois étapes différentes, à savoir la phase libre, la phase capacité (synchronisée) et la phase congestionnée.

La phase libre : Est quand les véhicules peuvent se déplacer librement, ce qui peut permettre des dépassements sur les voies [23].

La phase capacité (synchronisé) : Se produit dans les trafics à plusieurs voies lorsque les vitesses des véhicules dans les voies deviennent égales [23].

La phase congestionnée : Ressemble à une position d'arrêt et de départ de véhicules individuels, avec une vitesse presque nulle. Le trafic dans cette phase a un débit presque égal à zéro [23].

Les modèles comportementaux de mobilité sont des modèles qui tentent de reproduire les comportements humains lors des déplacements. Ces modèles se basent sur des règles de comportement des conducteurs [19].

2.2.2 Exemple des modèles de mobilité

Modèle de Gauss-markov :

Le modèle de mobilité Gauss-markov est conçu pour s'adapter à différents degrés de randomité, également appelé indice de randomité, Le degré de randomité dans les mouvements des noeuds mobiles est contrôlé par l'indice de randomisation. Plus le système est aléatoire, plus les mouvements des noeuds sont imprévisibles et aléatoires. Initialement, Les noeuds sont positionnés de manière aléatoire sur une zone de terrain et se déplacent de manière autonome [24].

Un noeud N calcule la vitesse et la direction de son mouvement pour chaque période de temps spécifiée en se référant à la rapidité et à la direction au cours de la période précise Comme illustré par les équations 2.1 et 2.2, avec une certaine quantité de randomisation intégrée dans le calcul. Le noeud n est alors déplacé dans la direction suivante et à une vitesse calculée qui est prédit sur le passage

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

du temps [25].

$$s_n = \alpha s_{n-1} + (1 - \alpha)\hat{s} + s_{x,n-1} \quad (2.1)$$

$$d_n = \alpha d_{n-1} + (1 - \alpha)\hat{d} + d_{x,n-1} \quad (2.2)$$

ou :

- s_n et d_n sont la nouvelle vitesse et direction du nœud au temps n .
- α , avec $0 \leq \alpha \leq 1$, est le paramètre de réglage utilisé pour varier l'indice de randomité.
- $\alpha = 1$: les nœuds se déplacent linéairement.
- \hat{s} et \hat{d} sont des constantes représentant la valeur moyenne de la vitesse et de la direction à l'instant n .
- $s_{x,n-1}$ et $d_{x,n-1}$ sont des variables aléatoires issues d'une distribution gaussienne.

Le paramètre α ($0 \leq \alpha \leq 1$), permet de contrôler le niveau d'incertitude dans le mouvement des nœuds. Plus il est proche de zéro, plus les mouvements deviennent imprévisibles. D'autre part, à mesure que 1 se rapproche, les mouvements deviennent plus prévisibles et les nœuds se déplacent à une vitesse et une direction plus proches de ce qu'ils étaient avant [25].

Modèle de déplacement aléatoire (RANDOM WAYPOINT)

Le modèle "Random Waypoint" est l'un des modèles de mobilité les plus couramment utilisés pour simuler les réseaux VANETs. Il s'agit d'un modèle simple dans lequel chaque nœud se déplace de manière aléatoire à des vitesses aléatoires, s'arrête à des positions aléatoires et attend à des moments aléatoires avant de repartir. Ce modèle est facile à implémenter et à analyser, ce qui explique en grande partie sa popularité dans les simulations [26].

Le modèle de mobilité Random Waypoint inclut des temps d'arrêt entre les changements de direction et/ou de vitesse. Un nœud mobile commence par rester à un emplacement pendant une certaine période (temps d'arrêt). Une fois ce temps

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

écoulé, le nœud mobile choisit une destination aléatoire dans la zone de simulation et une vitesse uniformément distribuée entre $[\text{minspeed}, \text{maxspeed}]$ [27].

Le nœud mobile se déplace ensuite vers la destination nouvellement choisie à la vitesse sélectionnée.

A l'arrivée, le nœud mobile fait une pause pendant une période de temps avant de recommencer le processus. [27].

Modèle de conduite intelligent (IDM) :

est un modèle mathématique utilisé pour décrire le comportement de suivi des véhicules dans le trafic routier. Il vise à simuler la manière dont les conducteurs ajustent leur vitesse et leur position par rapport aux véhicules environnants afin de maintenir une distance de sécurité et d'assurer une conduite fluide. L'IDM est largement utilisé dans la modélisation et la simulation du trafic routier pour évaluer les performances des systèmes de transport et étudier les phénomènes de congestion [28].

Pour simuler le modèle IDM, des méthodes numériques sont utilisées pour résoudre les équations différentielles du modèle. Un intervalle de temps est défini, et les équations sont intégrées sur cet intervalle pour obtenir les valeurs de vitesse et de distance à chaque instant. Des règles spéciales sont également utilisées pour traiter les cas où les véhicules sont à l'arrêt ou lorsqu'il y a des obstacles sur la route [29].

En résumé, le modèle IDM est utilisé pour simuler le comportement des conducteurs et le suivi des véhicules dans la circulation routière en prenant en compte différents facteurs et paramètres pour représenter différents styles de conduite et conditions de circulation.

Modèle de Krauss :

Dans ce modèle, chaque véhicule maintient une certaine distance de sécurité par rapport au véhicule qui le précède. Cette distance de sécurité permet au véhicule de réagir et de s'adaptant à lui, prend en compte des variables telles que la densité du trafic et la distance entre les véhicules pour simuler le mouvement des véhicules [30].

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

- Le modèle de Krauss, utilisé par le logiciel de simulation de trafic SUMO, est basé sur la vitesse sécuritaire. La vitesse sécuritaire du modèle de Krauss est calculée comme indiqué dans l'équation 2.3 [31] :

$$v_{\text{safe}} = v_l(t) + \frac{g(t) - v_l \cdot \text{tr}}{\left(\frac{v_l(t) + v_f(t)}{2b}\right) + \text{tr}} \quad (2.3)$$

Explication : La vitesse sécuritaire est calculée en fonction de la vitesse du véhicule de tête ($v_l(t)$), de l'espace entre les véhicules ($g(t)$), du temps de réaction du conducteur (tr), de la décélération maximale du véhicule (b) et d'un facteur de correction.

- Calcul de l'espace entre les véhicules (équation 2.4) [31] :

$$g(t) = x_l(t) - x_f(t) - l \quad (2.4)$$

Explication : L'espace entre les véhicules est calculé en soustrayant la position du véhicule suiveur ($x_f(t)$) de la position du véhicule de tête ($x_l(t)$), en tenant compte de la vitesse du véhicule claculer en l'équation 2.3.

- Calcul de la vitesse désirée ((équation 2.5) [31] :

$$v_{\text{des}}(t) = \min[v_{\text{max}}, v(t) + a \cdot \Delta t, v_{\text{safe}}(t)] \quad (2.5)$$

Explication : La vitesse désirée est déterminée en prenant le minimum entre trois valeurs : la vitesse maximale autorisée sur la route (v_{max}), la vitesse actuelle du véhicule ($v(t)$) augmentée d'une accélération (a) sur un intervalle de temps (Δt), et la vitesse sécuritaire calculée précédemment ($v_{\text{safe}}(t)$).

- Mise à jour de la vitesse (équation 2.6) [31] :

$$v(t + \Delta t) = \max[0, v_{\text{des}}(t) - \eta] \quad (2.6)$$

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

Explication : La vitesse au prochain pas de simulation ($t + \Delta t$) est mise à jour en prenant la valeur maximale entre zéro et la vitesse désirée ($v_{des}(t)$) diminuée d'une perturbation aléatoire (η). Cela permet de modéliser les variations de vitesse dues à des facteurs imprévisibles.

— Mise à jour de la position(équation 2.7) [31] :

$$xf(t + \Delta t) = xf(t) + v(t + \Delta t) \cdot \Delta t \quad (2.7)$$

Explication : La position au prochain pas de simulation ($t + \Delta t$) est calculée en ajoutant à la position actuelle ($xf(t)$) le produit de la vitesse mise à jour ($v(t + \Delta t)$) par l'intervalle de temps (Δt). Cela permet de déterminer la nouvelle position du véhicule en fonction de sa vitesse.

CHAPITRE 2. LES MODELES DE MOBILITÉ ET DE TRAFIC POUR LES RÉSEAUX VÉHICULAIRES DANS UNE VILLE INTELLIGENTE

Modèle de freeway :

Le modèle de mobilité freeway est basé sur une carte et vise à reproduire le mouvement des véhicules sur les autoroutes. Dans ce modèle, il y a plusieurs autoroutes composées de chemins à sens unique comme La représentation est illustrée dans la figure 2.1. Au début de la simulation, les nœuds sont placés aléatoirement sur les pistes et ont des vitesses initiales dépendantes du temps, qui sont mises à jour selon l'équation 2.8 [5] :

$$V(t + 1) = V(t) + \text{random} \times a(t) \quad (2.8)$$

où $V(t)$ est la vitesse du nœud au moment t , random est une valeur aléatoire uniformément choisie dans l'intervalle $[-1, 1]$, et $a(t)$ est l'accélération du nœud au moment t .

Le modèle définit également une distance de sécurité entre les véhicules qui se suivent sur le même chemin. Ainsi, si le nœud A suit le nœud B, la vitesse de A doit être inférieure ou égale à la vitesse de B. Cette règle peut être formulée comme l'équation 2.9 [5] :

$$D_{ij}(t) = SD \Rightarrow V_i(t) = V_j(t) \quad (2.9)$$

où $D_{ij}(t)$ est la distance entre les nœuds i et j au moment t , SD est la distance de sécurité, $V_i(t)$ est la vitesse du nœud i au moment t , et $V_j(t)$ est la vitesse du nœud j au moment t .

Enfin, le modèle impose la contrainte selon laquelle un nœud ne peut pas changer de chemin. Si un nœud sort de la zone de simulation, un nouveau chemin est choisi aléatoirement parmi les chemins disponibles.

Il convient de noter que ce modèle présente certaines limitations, notamment l'absence de mécanismes de contrôle tels que l'arrêt des véhicules aux feux de circulation [5]

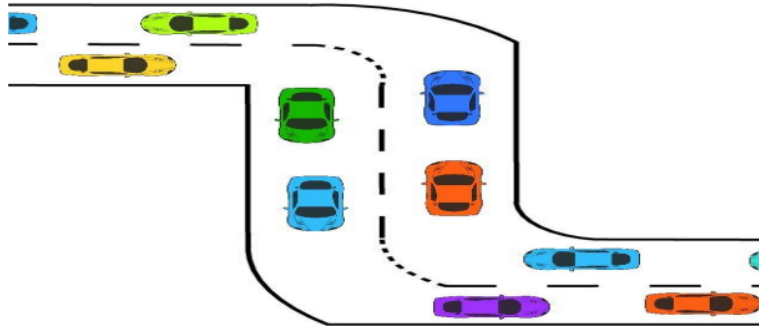


FIGURE 2.1 – Modèle de freeway [5].

Modèle de Manhattan :

Ce modèle de mobilité utilise une carte pour simuler la circulation des véhicules dans un environnement urbain. Avant de commencer la simulation, une carte est générée avec des rues horizontales et verticales. [5], La représentation de ce modèle est illustrée dans la figure 2.2.

Au début de la simulation, les véhicules sont placés aléatoirement sur la carte et se déplacent à des vitesses initiales (l'équation 2.9, tout comme dans le modèle freeway. Lorsqu'un véhicule atteint une intersection entre une rue horizontale et une rue verticale, il doit choisir une direction : continuer tout droit, tourner à droite ou à gauche [5].

La vitesse des véhicules est régie par une règle : un véhicule ne peut pas dépasser la vitesse de véhicule qui le précède pour garder une distance de sécurité [5].

Il est important de noter que ce modèle simplifié ne prend pas en compte certains aspects de la circulation urbaine, comme le modèle freeway.

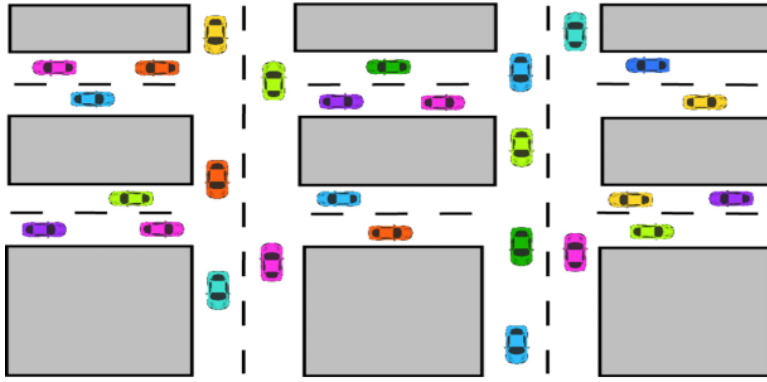


FIGURE 2.2 – Modèle de Manhattan [5].

2.2.3 Le choix d'un type de modèle de mobilité

Dans notre étude, nous avons utilisé trois modèles de mobilité différents : le modèle de mobilité Gauss-markov, le modèle de mobilité Krauss et le modèle IDM (Intelligent Driver Model). Ces modèles sont utilisés pour simuler le mouvement des véhicules dans notre scénario.

Par exemple, le modèle de Gauss-markov a été utilisé pour simuler le mouvement des noeuds mobiles en prenant en compte le degré de randomité contrôlé, tandis que le modèle de Krauss pour simuler le mouvement des véhicules en prenant en compte la distance de sécurité, la densité du trafic et la vitesse du véhicule. L'IDM est utilisé pour étudier comment les voitures interagissent entre elles en ajustant la vitesse et la position des conducteurs par rapport aux véhicules. En utilisant ces trois modèles, on peut mieux comprendre différents types de situations sur la route.

Le choix du modèle de mobilité doit tenir compte des objectifs et des conditions spécifiques de la simulation du véhicule. En comparant les résultats de simulation de ces trois modèles, on peut voir comment ils se comportent et déterminer lequel est le meilleur en fonction des objectifs de la simulation. Cela permet de choisir le modèle qui donne les meilleurs résultats pour atteindre les objectifs spécifiques de la simulation.

2.3 Les modèles de trafic

Un modèle de trafic est une représentation mathématique ou informatique qui simule le comportement des véhicules dans un environnement routier. Il est utilisé pour étudier, prédire ou optimiser les performances du trafic.

Il existe différents niveaux de détail dans les modèles de trafic, On a les modèles macroscopiques, microscopiques et mésoscopiques

2.3.1 Modèle macroscopique

Un modèle macroscopique est une approche qui s'intéresse au trafic global sur un réseau routier à grande échelle. Les modèles macroscopiques utilisent des équations mathématiques et des lois de conservation pour décrire le trafic routière, en prenant en compte des facteurs tels que la densité des véhicules, la vitesse moyenne et le débit [32].

Les modèles macroscopiques sont des outils couramment utilisés dans les études de planification des transports. Ils aident à évaluer les performances d'un réseau routière existant et à prédire les effets de futurs changements de la demande de transport. Ces modèles peuvent également être utilisés pour simuler des événements de circulation tels que les embouteillages et les accidents de la route [32].

Les modèles macroscopiques permettent de déterminer des paramètres globaux tels que les temps de trajet moyens, la consommation de carburant [23].

2.3.2 Modèle microscopique

Les modèles microscopiques sont des simulations qui prennent en compte plusieurs facteurs pour décrire comment chaque véhicule se comporte sur la route, tels que sa vitesse, sa position, sa direction, sa taille et les interactions avec d'autres véhicules et l'environnement routier. Ils utilisent des équations de mouvement pour représenter le comportement individuel de chaque véhicule [32].

Ils utilisent plusieurs composants pour cela, comme un suiveur qui évalue comment un véhicule réagit à celui qui le précède. D'autres modules sont utilisés pour calculer les effets secondaires tels que la pollution, les coûts économiques et la perte de temps et meme il peut simuler des phénomènes tels que la création de

bouchons, les arrêts et les départs du véhicule.

Le modèle de changement de voie est également important car il détermine comment les véhicules changent de voie en fonction de leur environnement. Et le modèle de choix d'itinéraire aide les véhicules à trouver le chemin le plus court à travers un réseau de routes (infrastructures routières) [23].

2.3.3 Modèle mésoscopique

Le modèle mésoscopique est un moyen de comprendre comment se passe la circulation sur la route, sans être trop compliqué. Il tient compte des comportements humains des conducteurs, mais ne demande pas trop de paramètres à estimer, comme c'est le cas pour le modèle microscopique. Le modèle macroscopique, lui, ne prend pas assez en compte les détails du comportement des conducteurs. Le modèle mésoscopique, quant à lui, peut prendre en compte la variabilité des comportements des conducteurs, tout en regardant l'effet global de la circulation sur la route, avec des variables comme la densité du trafic et la vitesse moyenne du flux [23].

2.3.4 Modèle de trafic Random Trips

Le Random Trips permet de générer rapidement un ensemble de trajets aléatoires dans un intervalle de temps donné. Les paramètres d'entrée comprennent le réseau routier, les heures de début et de fin de la simulation, la définition des véhicules et le taux d'arrivée des véhicules. Par défaut, les points de départ et d'arrivée sont sélectionnés de manière aléatoire. Le fichier de sortie contient une liste de trajets avec les arêtes de départ et d'arrivée. Avec le Random Trips, les traces des véhicules sont calculées au moment du départ dans la simulation [33]. Il est possible d'ajuster les autres options selon les besoins de la simulation. Une fois les trajets générés, ils peuvent être utilisés avec duarouter [34].

Duarouter est un outil disponible dans SUMO qui permet de générer des itinéraires pour les véhicules dans une simulation de trafic. Il prend en entrée un réseau routier et un ensemble de trajets [34].

2.4 Conclusion

En conclusion, cette section a présenté différents modèles de mobilité et de trafic utilisés dans les réseaux VANETs. Les modèles de mobilité peuvent être classés en fonction de leur source de données et de leur réalisme, tels que les modèles basés sur les traces, les enquêtes, les simulateurs de trafic et les modèles synthétiques. Les modèles de trafic, quant à eux, peuvent être catégorisés en modèles macroscopiques, microscopiques et mésoscopiques

Le choix d'un modèle de mobilité et de trafic dépend des objectifs de la simulation, des conditions spécifiques du réseau et des niveaux de détail souhaités. Dans notre étude, nous avons utilisé les modèles de mobilité Gauss-markov, Krauss et IDM et le modèle de trafic RandomTrips sous les simulateurs SUMO et NS-3 pour simuler le mouvement des véhicules dans un environnement urbain. Dans le prochain Chapitre, nous allons décrire toutes les étapes de l'implémentation et la simulation d'un réseau VANETs à l'intérieur d'une ville intelligente. Nous présenterons aussi les résultats obtenus et nous les discutons.

Chapitre 3

IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

3.1 Introduction

Le chapitre de la simulation et évaluation des résultats a pour objectif de présenter les différentes étapes nécessaires pour mettre en place une simulation de réseau VANET. Cette simulation permet de tester différents protocoles de routage et de mobilité pour les véhicules connectés. Les prédictions de trafic sont également prises en compte en simulant des scénarios réalistes. La simulation est basée sur l'utilisation de différents outils, notamment SUMO (Simulation of Urban Mobility) pour la génération du trafic et la modélisation de la mobilité des véhicules, et NS3 (Network Simulator 3) pour la simulation des protocoles de routage dans les réseaux VANET. Dans ce chapitre, nous présentons les différentes étapes nécessaires à la mise en place de la simulation.

3.2 Objectif de la simulation

L'objectif de la simulation est de reproduire un environnement réaliste de communication sans fil entre les véhicules et les infrastructures routières, en utilisant des modèles de mobilité qui prennent en compte les mouvements des véhicules sur la route et les interactions entre eux. Les modèles de mobilité sont essentiels pour simuler le comportement des véhicules, leur vitesse, leur direction et leurs interactions avec les autres véhicules et les infrastructures routières.

3.3 Simulation de modèle de mobilité et les protocoles de routages

3.3.1 Simulation de modèle de mobilité

Pour développer ou configurer la simulation de modèles de mobilité et de trafic dans les VANETs, la première étape consiste à préparer une nouvelle carte routière (les nœuds, les routes, les voies, les jonctions et les véhicules) et les règles de circulation (la distance de sécurité et les priorités entre les véhicules), tandis que la deuxième étape consiste à utiliser un modèle de mobilité ou de trafic existant [35].

Lorsque nous exécutons et simulons un modèle de mobilité pour un réseau VANET, cela génère des fichiers TRACE qui fournissent des informations détaillées sur les déplacements des nœuds ainsi que les requêtes échangées entre eux.

3.3.2 Simulation des protocoles de routages

Dans les réseaux de véhicules, il est important de rappeler les principes du routage dans les réseaux Ad-Hoc mobiles (MANETs). Le routage consiste à acheminer des informations vers une destination [4]

Protocoles de routage pour dans les réseaux MANETs

Il existe deux familles principales de protocoles de routage : les protocoles à plat et les protocoles hiérarchiques ou géographiques. Les protocoles à plat comprennent les protocoles proactifs qui conservent toutes les routes possibles pour chaque destination dans le réseau et les protocoles réactifs qui attendent qu'une route soit demandée pour déterminer le chemin [4].

Les protocoles hiérarchiques ou géographiques sont conçus pour les réseaux à grande échelle. Le routage géographique utilise la position géographique des nœuds pour acheminer les données. Le routage hiérarchique consiste à diviser le réseau en groupes plus petits [4]. Le diagramme présenté dans la Figure 3.1 illustre les protocoles de routage utilisés dans les réseaux MANETs.

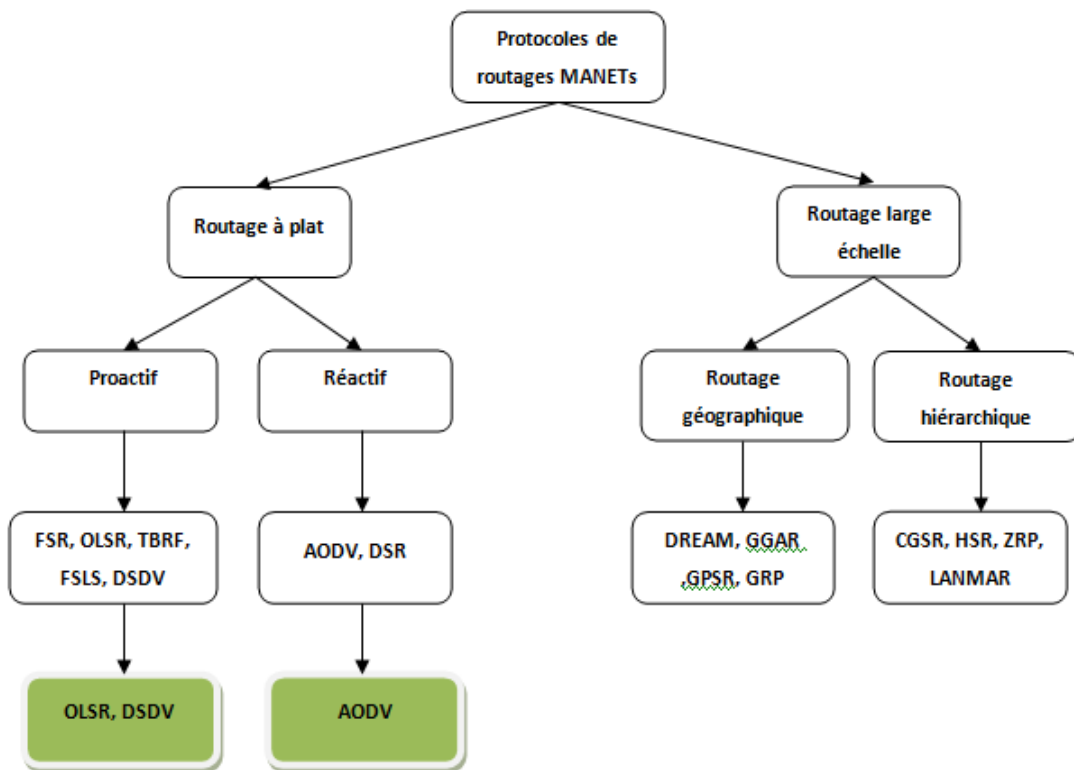


FIGURE 3.1 – Classification des protocoles de routage dans les réseaux MANETs [4].

Protocoles de routages dans les réseaux VANETs

Les protocoles de routage Ad-Hoc couramment utilisés pour les MANETs sont évalués pour être utilisés dans un environnement VANET [36].

Le diagramme présenté dans la Figure 3.2 illustre les protocoles de routage utilisés dans les réseaux VANETs.

En ce qui nous concerne, nous nous sommes concentrés uniquement sur les protocoles de routage : DSDV, AODV et OLSR.

DSDV (Destination Sequence Distance Vector) est un protocole de routage proactif où chaque nœud garde une liste des informations sur tous les autres nœuds du réseau. Cette liste est mise à jour régulièrement lorsqu'il y a des changements dans le réseau [36].

OLSR (Optimized Link State Routing) est un protocole de routage proactif où chaque nœud d'un réseau transmet des informations de routage à tous les autres nœuds. Le protocole OLSR est basé sur un algorithme universel d'état des liens. Pour mettre à jour les données, des messages d'état de liaison sont systématiquement échangés par les nœuds [37].

AODV (Ad-Hoc On-Demand Distance Vector) est un protocole de routage réactif. Il permet aux nœuds mobiles d'obtenir rapidement des routes vers de nouvelles destinations. Les protocoles réactifs tels AODV nécessitent un temps pour la recherche d'une route, ce qui peut être problématique pour les réseaux VANETs [4].

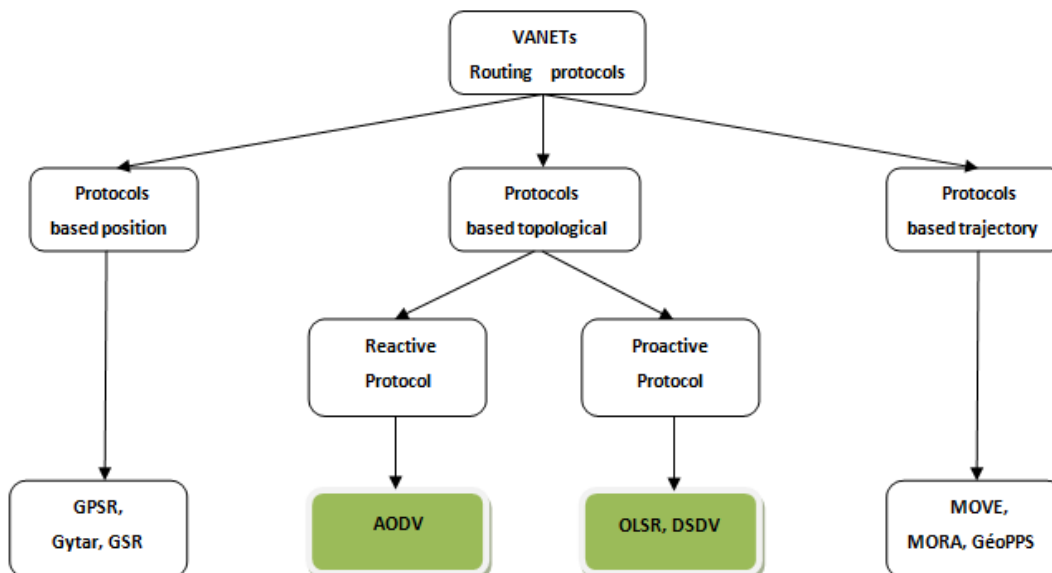


FIGURE 3.2 – La classification des protocoles de routage dans les réseaux VANETs [4].

3.3.3 Simulateurs de mobilité et du trafic pour les VANET

Les simulateurs de mobilité et de trafic pour les réseaux de véhicules Ad-Hoc (VANET) sont des outils logiciels qui permettent de modéliser le comportement et les déplacements des véhicules dans le réseau, ainsi que les échanges de données entre les nœuds.

Parmi les simulateurs de mobilité et de trafic les plus populaires pour les VANETs, on peut citer [35] :

SUMO (Simulation of Urban MObility) : un simulateur de trafic routier et de transport en commun qui peut être utilisé pour simuler les réseaux de véhicules Ad-Hoc.

NS-3 (Network Simulator version 3) : un simulateur de réseaux de communication qui peut être utilisé pour simuler les VANET.

Veins (Vehicular Network Simulation) : une extension de SUMO et d'OMNeT++ qui permet de simuler des réseaux de véhicules Ad-Hoc en utilisant les données de trafic générées par SUMO.

OMNeT++ (Objective Modular Network Testbed in C++) : un simulateur de réseaux de communication qui peut être utilisé pour simuler les VANET.

MOVE (Mobility model and Vehicle Environment) : un simulateur de mobilité pour les réseaux de véhicules Ad-Hoc.

3.3.4 Simulateurs de mobilité et du trafic utilisés

Pour notre simulation, nous avons utilisé les outils suivant :

SUMO (Simulation of Urban MObility)

SUMO est facilement utilisable et ne nécessite que l'installation des bibliothèques C++. Dans SUMO, chaque véhicule suit un itinéraire spécifique et son comportement est dynamique, incluant la possibilité de changer de voie. Les routes sont représentées par des voies multiples et des intersections qui obéissent à des règles de circulation ou de priorité [38] [39].

Lorsqu'on utilise SUMO pour simuler le trafic, il comprend un ensemble de fichiers tels que :

Le fichier "node" : Qui contient les coordonnées des nœuds.

Le fichier “edge” : Stocke les détails des voies routières. Chaque voie est identifiée par un numéro d’identification unique.

Le fichier “Net file” : Est un fichier de format XML portant l’extension .net.xml. Il est généré à partir de la commande NETCONVERT.

Le fichier “Flows” : Représente le flux de véhicules entre deux nœuds en spécifiant leur point de départ et d’arrivée, le nombre de véhicules, ainsi que les temps de début et de fin.

Le fichier “route” : Est généré à l’aide de la commande “DUAROUTER”, qui prend en entrée les fichiers “net.xml” et “flow.xml”.

Le fichier “sumocfg” : Il s’agit du fichier de configuration de SUMO. Il permet de configurer les paramètres de la simulation tels que le temps de simulation, le type de véhicules à utiliser, les itinéraires, etc.

NS-3 (Network Simulator-3)

NS-3 est un logiciel qui permet de modéliser et de simuler le comportement de réseaux de communication. Il utilise les scénarios de simulation et permet l’implémentation et l’évaluation de protocoles de communication dans diverses applications [35] [40].

3.4 Les étapes de la simulation

3.4.1 Étape 1 : Simulation SUMO

Générer et Importer la Map de Simulation

Les simulations réalistes permettent d’avoir une représentation des routes et de l’environnement dans lequel se déroulent les communications, les interactions, les rond points et Les feux rouges.

Dans notre recherche, nous devons d’abord trouver une zone de simulation, cette zone de simulation est une carte routière. Nous pouvons créer cette carte de deux façons différentes. La première façon : la création manuelle à partir des fichiers SUMO, tels que le fichier de réseau (.net.xml) et le fichier de configuration (.sumocfg), en utilisant les fichiers de SUMO. Ces fichiers peuvent être créés à

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

l'aide d'éditeurs de texte .

La deuxième méthode consiste à importer automatiquement une carte réelle à partir de bases de données en ligne OSM.

OSMWebWizard : OSMWebWizard est une application web open-source développée par le projet OpenStreetMap pour aider les utilisateurs à créer des cartes personnalisées. Il permet aux utilisateurs de sélectionner des régions géographiques et télécharger le résultat final sous forme de fichiers .osm.

Étapes de La méthode manuelle pour créer manuellement la carte représentée dans les figures 3.3 et 3.4 en suivant les étapes ci-dessous :

- Étape 1 : Création d'un fichier de nœuds contenant les informations sur les jonctions. L'extension du fichier serait .nod.xml [6].

```
<nodes> <!-- Balise d'ouverture -->
  <node id="0" x="400.0" y="700.0" type="priority"/>
  <!-- Définition du noe{}ud "0" -->
  <node id="1" x="700.0" y="700.0" type="priority"/>
  <!-- Définition du noe{}ud "1" -->
  <node id="2" x="900.0" y="500.0" type="traffic_light"/>
  <node id="3" x="700.0" y="300.0" type="priority"/>
  <node id="4" x="400.0" y="300.0" type="priority"/>
  <!-- Définition du noe{}ud "4" -->
  <node id="5" x="200.0" y="500.0" type="traffic_light"/>
</nodes> <!-- Balise de fermeture -->
```

- Étape 2 : Création des arêtes (routes reliant les jonctions). L'extension est .edg.xml [6].

```
<edges>
  <edge id="l01" from="0" to="1" priority="2" numLanes="2"
  speed="11.11"/>
  <edge id="r01" from="1" to="0" priority="3" numLanes="2"
  speed="13.89"/>
```

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

```
<edge id="u12" from="1" to="2" priority="1" numLanes="2"  
speed="11.11"/>  
<edge id="d12" from="2" to="1" priority="2" numLanes="2"  
speed="11.11"/>
```

```
<edge id="d23" from="2" to="3" priority="3" numLanes="2"  
speed="13.89"/>  
<edge id="u23" from="3" to="2" priority="1" numLanes="2"  
speed="11.11"/>
```

```
<edge id="l43" from="3" to="4" priority="2" numLanes="2"  
speed="11.11"/>  
<edge id="r43" from="4" to="3" priority="3" numLanes="2"  
speed="13.89"/>
```

```
<edge id="d54" from="5" to="4" priority="1" numLanes="2"  
speed="11.11"/>  
<edge id="u54" from="4" to="5" priority="2" numLanes="2"  
speed="11.11"/>
```

```
<edge id="d05" from="0" to="5" priority="3" numLanes="2"  
speed="13.89"/>  
<edge id="u05" from="5" to="0" priority="1" numLanes="2"  
speed="11.11"/>
```

```
<edge id="l52" from="5" to="2" priority="3" numLanes="2"  
speed="13.89"/>  
<edge id="r52" from="2" to="5" priority="1" numLanes="2"  
speed="11.11"/>
```

```
</edges>
```

- Étape 3 : Créer un fichier net.xml en utilisant les fichiers de noeuds et d'arêtes [6].

```
$ netconvert -n file.nod.xml -e file.edg.xml -o file.net.xml
```

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

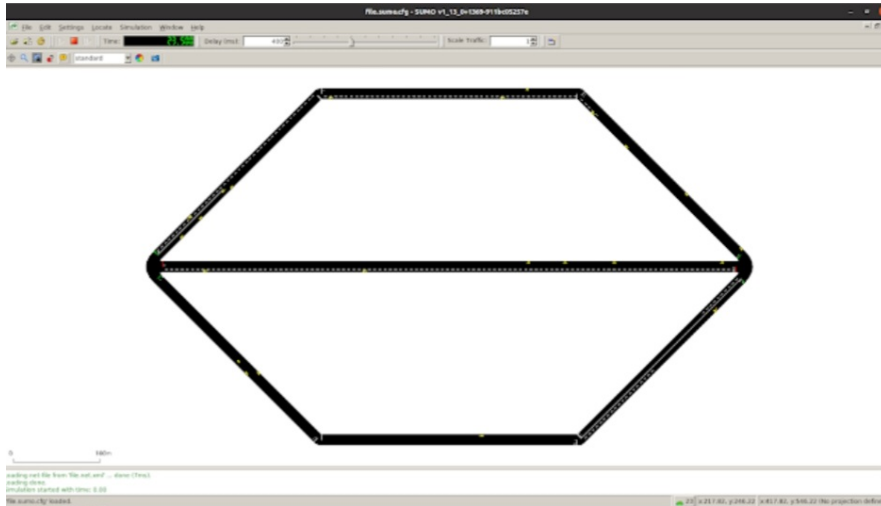


FIGURE 3.3 – Exemple de création une Map manuelle [6].

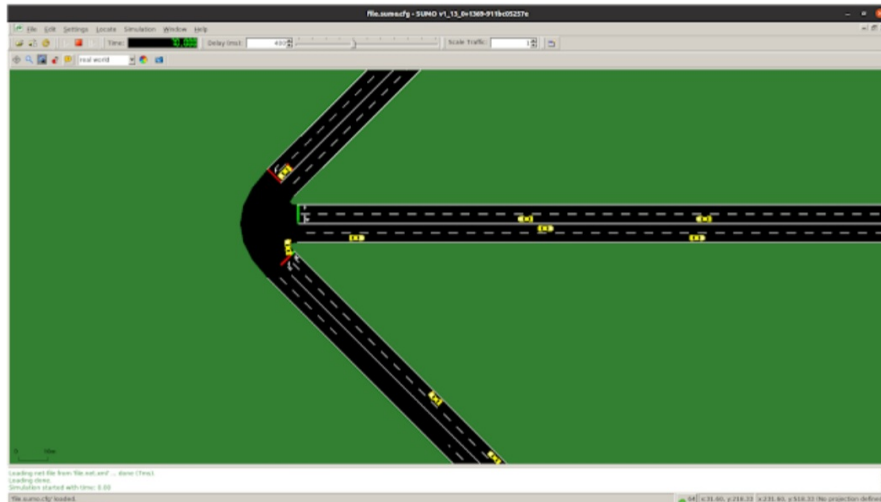


FIGURE 3.4 – Exemple de création d'une map manuelle [6].

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

Étapes de la méthode automatique(OSM)

Lorsque nous travaillons avec la méthode automatique, nous pouvons trouver le siège de la radio d’Ain Témouchent et le siège de la wilaya d’Ain Témouchent sur le site Web d’OpenStreetMap en suivant les étapes ci-dessous :

- Accédez à la page d’accueil d’OpenStreetMap à l’adresse <https://www.openstreetmap.org/>
- Dans la barre de recherche en haut de la page, tapez “Ain Témouchent” et appuyez sur “Entrée”.
- Après avoir sélectionné la zone que nous souhaitons(Le siège de la radio d’Ain Témouchent et le siège de la wilaya d’Ain Témouchent), nous l’exportons et avons maintenant un fichier appelé ”map.net.xml”.
- Le fichier généré par OSM contient des informations , tels que les nœuds, les arcs et les relations entre eux. Il contient également des informations sur les caractéristiques des routes, comme le nombre de voies, les limitations de vitesse, les feux de signalisation, etc.
- Avec ce fichier, il est possible de générer un scénario de simulation classique en définissant les positions et les mouvements des véhicules dans la zone sélectionnée.

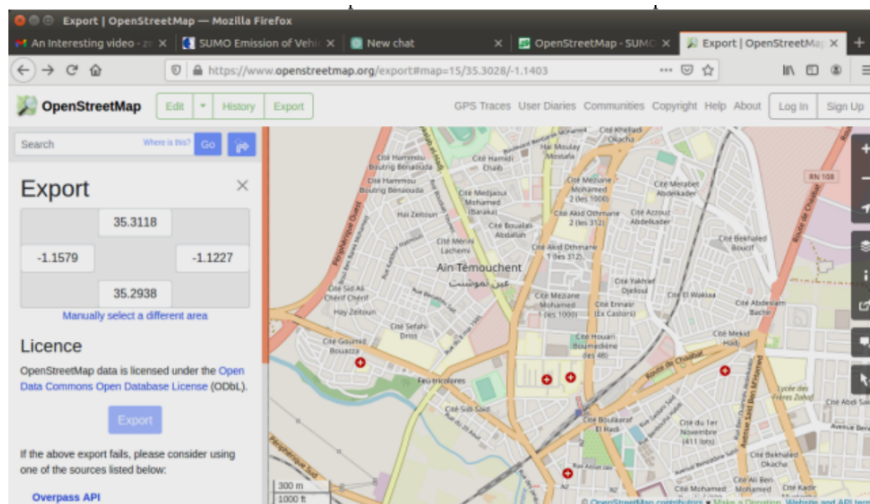


FIGURE 3.5 – Exemple de la création d’une map

Convertir les données en SUMO

”Nous allons dans le terminal. Nous accédons au répertoire où nous avons téléchargé les données OpenStreetMap. Dans notre exemple comme indiqué dans la figure 3.6, nous faisons : `cd Download/` et nous exécutons la commande suivante pour convertir les données en format SUMO :

```
netconvert --osm-files map.osm -o map.net.xml
```

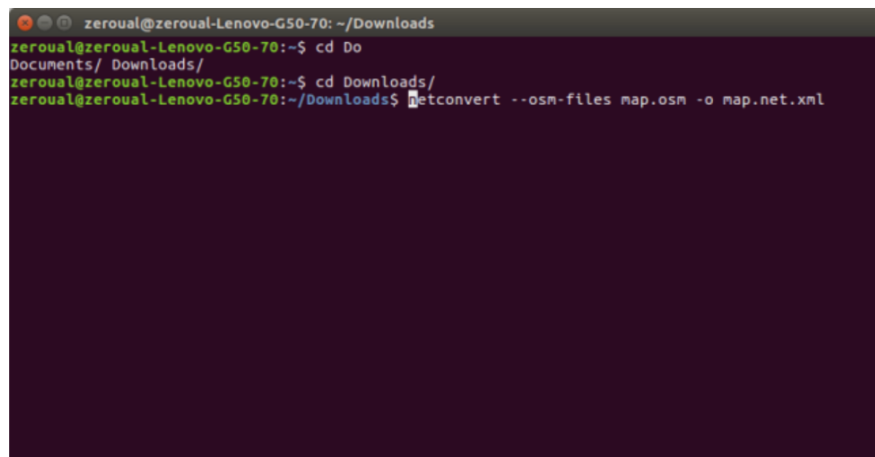


FIGURE 3.6 – Convertir les données en format SUMO

Création d’un nouveau dossier dans le répertoire principal

- Créez un nouveau dossier appelé ”zeroual” dans le répertoire principal.
- Mettez le fichier `map.net.xml` dans ce dossier.

Génération de trajets aléatoires

Il existe un fichier appelé `tools/randomTrips.py` qui peut générer des véhicules sur le réseau. Le nombre de véhicules par défaut est de 1 véhicule par seconde.

Notre simulation dure 3600 secondes et nous voulons faire rouler 80 véhicules.

Nous modifions donc la ligne 740 dans `randomtrips` comme suit :

```
—————departureTime += period+44—————
```

Puis exécuter la commande suivante :

```
#!/bin/bash
```

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

```
# Generate 3 sets of random trips with different seed
values and output files
for i in 10 20 30
do
    python3 sumo/tools/randomTrips.py -n

    "Desktop/pfe/map.net.xml" -s $i -o "Desktop/pfe/roudef_$i.xml"
done
```

Création d'un fichier nommé "file.sumo.cfg"

```
<configuration>
<input>
<net-file value="map.net.xml"/>
<route-files value="zeroual.rou.xml"/>
</input>
<report>
<verbose value="true"/>
<duration-log.statistics value="true"/>
<no-step-log value="true"/>
</report>
</configuration>
```

On a résumé les étapes de la simulation dans Le diagramme présenté dans la figure 3.7 :

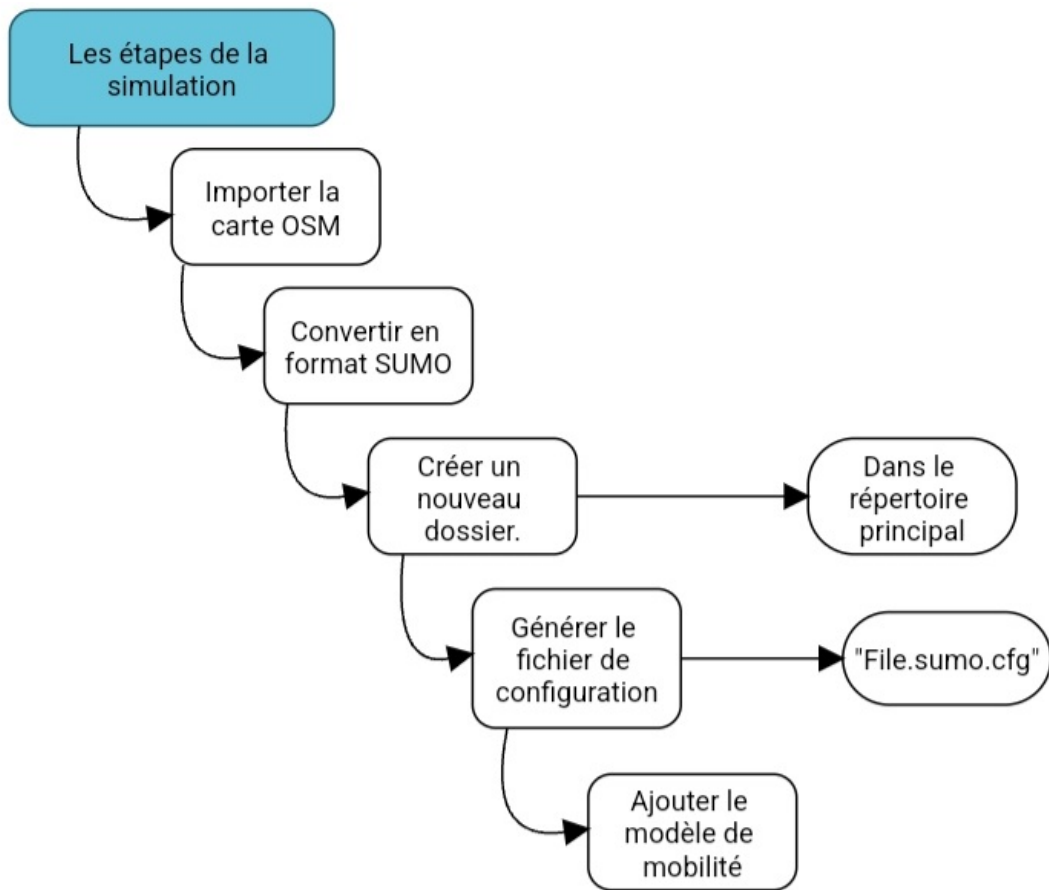


FIGURE 3.7 – Les étapes de la simulation

Ajouter un modèle de mobilité

Pour ajouter et spécifier un modèle de mobilité dans SUMO :

1. Ajouter les fichiers "Gaussmarkov.cpp" et "Gaussmarkov.h" dans le dossier "sumo/src/microsim/cfmodels".
2. Aller dans le fichier "sumo/src/microsim/cfmodels/CMakeLists.txt" et ajouter les noms des fichiers "Gaussmarkov.cpp" et "Gaussmarkov.h".
3. Nous ajoutons maintenant les éléments XML qui nous permettent de définir et d'analyser les paramètres du modèle.
 - Aller dans le fichier `src/Utils/xml/sumoxmldefinitions.cpp`.
Le fichier `sumoxmldefinitions.cpp` définit les éléments XML utilisés dans les fichiers de configuration de SUMO. Nous apportons les modifications suivantes :
 - La première section `"StringBijection<int>::Entry SUMOXMLDefinitions:tags[]"`, définit le mappage entre les balises XML et leurs ID correspondants. Dans cette section, nous pouvons ajouter une nouvelle entrée pour définir l'élément XML pour le modèle de suivi de voiture Gaussmarkov. :

```
{ "carFollowing-Gaussmarkov", SUMO_TAG_CF_Gaussmarkov },
```
 - La section `"carFollowModelValues[]"` définit le mappage entre les noms de modèles de suivi de voiture et leurs balises XML correspondantes. Dans cette section, nous pouvons ajouter une entrée pour le modèle de Gaussmarkov. :

```
{ "Gaussmarkov", SUMO_TAG_CF_Gaussmarkov },
```

 pour définir l'élément XML du modèle Gaussmarkov :
4. Aller dans `/src/microsim/MSVehicleType.cpp`
Le fichier `MSVehicleType.cpp` est responsable de l'analyse des définitions de type de véhicule. Nous apportons les modifications suivantes :
 - Ajouter :

```
#include "cfmodels/MSCFModel_Gaussmarkov.h"
```

Nous incluons le fichier d'en-tête contient la déclaration de la classe `MSCFModel_Gaussmarkov`.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

- Dans l'instruction switch pour analyser le type de modèle suivant la voiture, Ajouter :

```
case SUMO_TAG_CF_Gaussmarkov:
    vtype->myCarFollowModel = new MSCFModel_Gaussmarkov(vtype);
    break;
```

5. Aller dans `src/utils/vehicle/SUMOVehicleParserHelper.cpp`

- Ajouter :

```
std::set<SumoXMLAttr> GaussmarkovParams;
GaussmarkovParams.insert(SUMO_ATTR_ACCEL);
GaussmarkovParams.insert(SUMO_ATTR_DECEL);
GaussmarkovParams.insert(SUMO_ATTR_SIGMA);
GaussmarkovParams.insert(SUMO_ATTR_TAU);
allowedCFModelAttrs[SUMO_TAG_CF_Gaussmarkov] = GaussmarkovParams;
```

6. Dans le terminal, accéder à `sumo/build/cmake-cache` et exécuter `make -j8` pour compiler.
7. Dans les fichiers de simulation générés dans le dossier `zeroual/zeroual.sumo.cfg`, ajouter la commande suivante pour choisir le modèle : Krauss, IDM ou Gaussmarkov.

```
<processing>
<default.carfollowmodel value="Gaussmarkov"/>
</processing>
```

8. Lancer la simulation avec la commande :

```
$ sumo-gui zeroual.sumo.cfg
```

pour visualiser la simulation en temps réel, ou pour voir le résultat de la simulation.

```
$ sumo -c "zeroual.sumo.cfg"
```

pour voir le résultat de la simulation.

Voici le code par défaut des paramètres de type de véhicule du modèle CF : `src/utils/vehicle/SUMOVTypeParameter.cpp`

Résultats de la simulation sous SUMO

Le tableau 3.1 présente le temps d'attente des véhicules (20, 40, 60), avec les modèles Gaussmarkov, IDM, Krauss. Comme illustré dans la Figure 3.8

	veh20	veh40	veh60
Gaussmarkov	0,8	2,30	4,2
IDM	0,8	0,35	0,7
Krauss	1,49	1,58	2,9

TABLE 3.1 – Temps d'attente des véhicules.

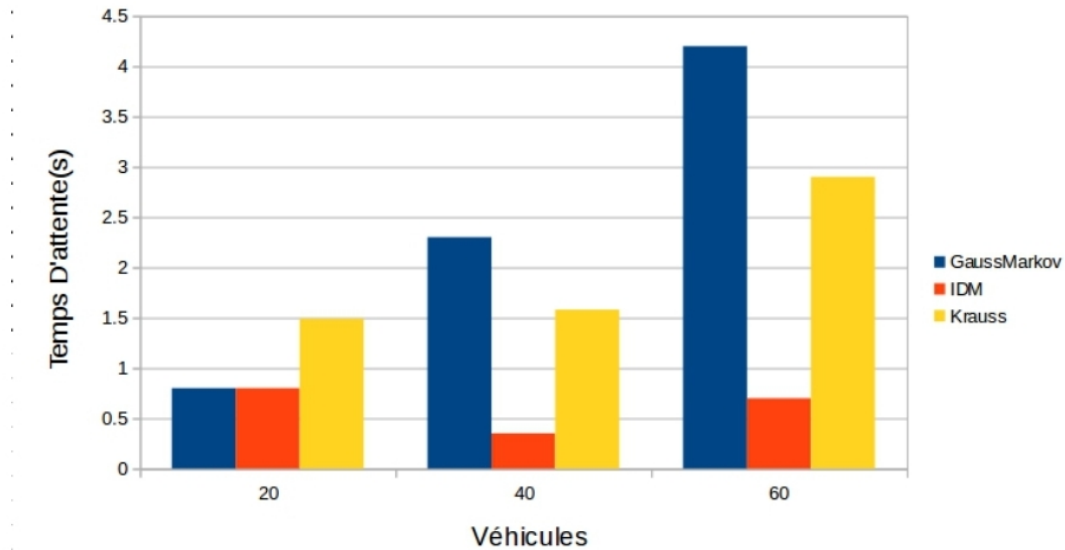


FIGURE 3.8 – Analyse des délais d'attente des véhicules sous forme d'un histogramme.

Selon les résultats de la comparaison des modèles de mobilité pour les véhicules 20, 40 et 60, les conclusions suivantes peuvent être tirées :

1. Les modèles IDM et Gaussmarkov ont des temps d'attente similaires pour le véhicule 20, ce qui indique que les deux modèles ont une efficacité dans la gestion de trafic. Cependant, le modèle Krauss affiche un temps d'attente légèrement plus élevé, indiquant une efficacité légèrement inférieure de trafic par rapport aux autres modèles.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

2. Pour un nombre de véhicules égal à 40, le modèle IDM présente le temps d'attente le plus faible, le modèle Gaussmarkov montre un temps d'attente plus élevé, à cause de la possibilité des retards par rapport au modèle IDM. Le modèle Krauss affiche un temps d'attente de 1.58 seconde, indiquant une performance moyenne de trafic pour ce véhicule.
3. En ce qui concerne un nombre de véhicules égal à 60, le modèle de Gaussmarkov présente le temps d'attente le plus élevé, Le modèle IDM démontre un temps d'attente de 0.7 seconde, suggérant une meilleure efficacité de trafic par rapport au modèle de Gaussmarkov. Le modèle Krauss présente un temps d'attente de 2.9 secondes, indiquant une performance moyenne de trafic pour ce véhicule .

Dans l'ensemble, Le modèle IDM présente des temps d'attente plus faibles, indiquant une meilleure efficacité de la gestion de trafic, tandis que le modèle de Gaussmarkov montre des temps d'attente qui ne sont pas constants et peuvent changer. et le modèle Krauss présente des temps d'attente qui indiquent une performance moyenne pour différents nombres de véhicules.

3.4.2 Etape 2 : simulation sous NS-3

Pour réaliser une simulation avec le simulateur NS-3, il est nécessaire de créer des fichiers traces à partir de SUMO (Simulation of Urban MObility), un outil de simulation de trafic routier.

Génération des fichiers traces

La création des fichiers traces est une étape très importante pour la création d'un scénario réseau pour le simulateur NS-3. Il existe différentes façons de générer ces fichiers traces, telles que l'utilisation de `SUMO trace exporter`, `MOVES` ou `iTETRIS`. Dans notre cas, nous allons utiliser `SUMO trace exporter`, qui est un utilitaire disponible dans le dossier `tools` du package SUMO.

Pour générer les fichiers traces, nous allons d'abord créer le fichier trace en utilisant la commande suivante :

```
$ sumo -c zeroual.sumo.cfg --fcd-output trace.xml
```

Ensuite, nous allons convertir le fichier trace créé vers un autre format, dans notre cas `.tcl` qui correspond à NS2, en utilisant la commande suivante :

```
$ python3 sumo/tools/traceExporter.py -i "zeroual/trace.xml" --ns2mobility-output="zeroual/mobility.tcl"
```

Exécution d'un script dans NS-3

Pour exécuter un script dans NS-3, nous allons d'abord copier le fichier `vanet-routing-compare.c` du dossier `ns3/src/wave/example` vers le dossier `ns3/scratch`.

Ensuite, nous allons ajouter les éléments suivants dans le fichier `vanet-routing-compare.cc` :

1. Ajouter le module `NetAnim` :

```
#include "ns3/netanim-module.h"  
AnimationInterface anim("Vanetanim.xml"); //Add  
before Simulator::Run();
```

2. Ajouter le code d'analyse de performance pour le programme `vanet` qui va générer les résultats. Nous allons ajouter le code suivant à la ligne 1729 comme illustré dans la figure 3.9 et 3.10 :

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

```
1729 ////////////////////////////////////////////////////////////////// Network Perfomance Calculation ////////////////////////////////////////
1730 uint32_t SentPackets = 0;
1731 uint32_t ReceivedPackets = 0;
1732 uint32_t LostPackets = 0;
1733 int j=0;
1734 float AvgThroughput = 0;
1735 Time Jitter;
1736 Time Delay;
1737
1738 Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (flowmon.GetClassifier ());
1739 std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();
1740
1741 for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator iter = stats.begin (); iter !=
stats.end (); ++iter)
1742 {
1743     Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (iter->first);
1744
1745 NS_LOG_UNCOND("----Flow ID:" <<iter->first);
1746 NS_LOG_UNCOND("Src Addr" <<t.sourceAddress << "Dst Addr " << t.destinationAddress);
1747 NS_LOG_UNCOND("Sent Packets=" <<iter->second.txPackets);
1748 NS_LOG_UNCOND("Received Packets =" <<iter->second.rxPackets);
1749 NS_LOG_UNCOND("Lost Packets =" <<iter->second.txPackets-iter->second.rxPackets);
1750 NS_LOG_UNCOND("Packet delivery ratio =" <<iter->second.rxPackets*100/iter->second.txPackets << "%");
1751 NS_LOG_UNCOND("Packet loss ratio =" << (iter->second.txPackets-iter->second.rxPackets)*100/iter-
>second.txPackets << "%");
1752 NS_LOG_UNCOND("Delay =" <<iter->second.delaySum);
```

FIGURE 3.9 – Code d’analyse de performance pour le programme vanet 1 .

```
1752 NS_LOG_UNCOND("Delay =" <<iter->second.delaySum);
1753 NS_LOG_UNCOND("Jitter =" <<iter->second.jitterSum);
1754 NS_LOG_UNCOND("Throughput =" <<iter->second.rxBytes * 8.0/(iter->second.timeLastRxPacket.GetSeconds
()-iter->second.timeFirstTxPacket.GetSeconds())/1024<<"Kbps");
1755
1756 SentPackets = SentPackets +(iter->second.txPackets);
1757 ReceivedPackets = ReceivedPackets + (iter->second.rxPackets);
1758 LostPackets = LostPackets + (iter->second.txPackets-iter->second.rxPackets);
1759 AvgThroughput = AvgThroughput + (iter->second.rxBytes * 8.0/(iter->second.timeLastRxPacket.GetSeconds
()-iter->second.timeFirstTxPacket.GetSeconds())/1024);
1760 Delay = Delay + (iter->second.delaySum);
1761 Jitter = Jitter + (iter->second.jitterSum);
1762
1763 j = j + 1;
1764
1765 }
1766
1767 AvgThroughput = AvgThroughput/j;
1768 NS_LOG_UNCOND("-----Total Results of the simulation-----"<<std::endl);
1769 NS_LOG_UNCOND("Total sent packets =" << SentPackets);
1770 NS_LOG_UNCOND("Total Received Packets =" << ReceivedPackets);
1771 NS_LOG_UNCOND("Total Lost Packets =" << LostPackets);
1772 NS_LOG_UNCOND("Packet Loss ratio =" << ((LostPackets*100)/SentPackets)<< "%");
1773 NS_LOG_UNCOND("Packet delivery ratio =" << ((ReceivedPackets*100)/SentPackets)<< "%");
1774 NS_LOG_UNCOND("Average Throughput =" << AvgThroughput<< "Kbps");
1775 NS_LOG_UNCOND("End to End Delay =" << Delay);
1776 NS_LOG_UNCOND("End to End Jitter delay =" << Jitter);
1777 NS_LOG_UNCOND("Total Flod id " << j);
1778 monitor->SerializeToXmlFile("manet-routing.flowmon", true, true);
1779
```

FIGURE 3.10 – Code d’analyse de performance pour le programme vanet 2.

3. Configurer le scénario 2 à la ligne 2441 comme illustré dans la figure 3.11 :

```
}  
else if (m_scenario == 2)  
{  
  
    m_traceFile = "/home/zeroual/Desktop/pfe/mobility.tcl";  
    m_logFile = "/home/zeroual/Desktop/pfe/vanet.log";  
    m_mobility = 1;  
    m_nNodes = 20;  
    m_TotalSimTime = 116.01;  
  
    m_csvfileName = "/home/zeroual/Desktop/pfe/vanet.csv";  
}
```

FIGURE 3.11 – Code de la Configuration de 2ème scénario .

4. Ouvrez NetAnim à l'aide des commandes suivantes :

```
$ cd ns-allinone-3.29/netanim  
$ ./NetAnim
```

Chargez le fichier Vanetanim.xml pour voir l'animation du réseau.

Résultats de simulation des protocoles de routage AODV, OLSR et DSDV sous NS-3

Dans cette section, nous discutons des protocoles de routage existants pour les réseaux VANETs, nous avons choisi de travailler avec les protocoles AODV, OLSR et DSDV.

Résultats de simulation des protocoles AODV : Le tableau 3.2 présente les paquets reçus en utilisant le protocole AODV, comme illustré dans la Figure 3.12

	vehicle20	vehicle40	veh60
Gaussmarkov	82,00%	91,00%	90,00%
IDM	89,00%	90,00%	90,00%
Krauss	87,00%	91,00%	90,00%

TABLE 3.2 – Les paquets reçus avec le protocole AODV

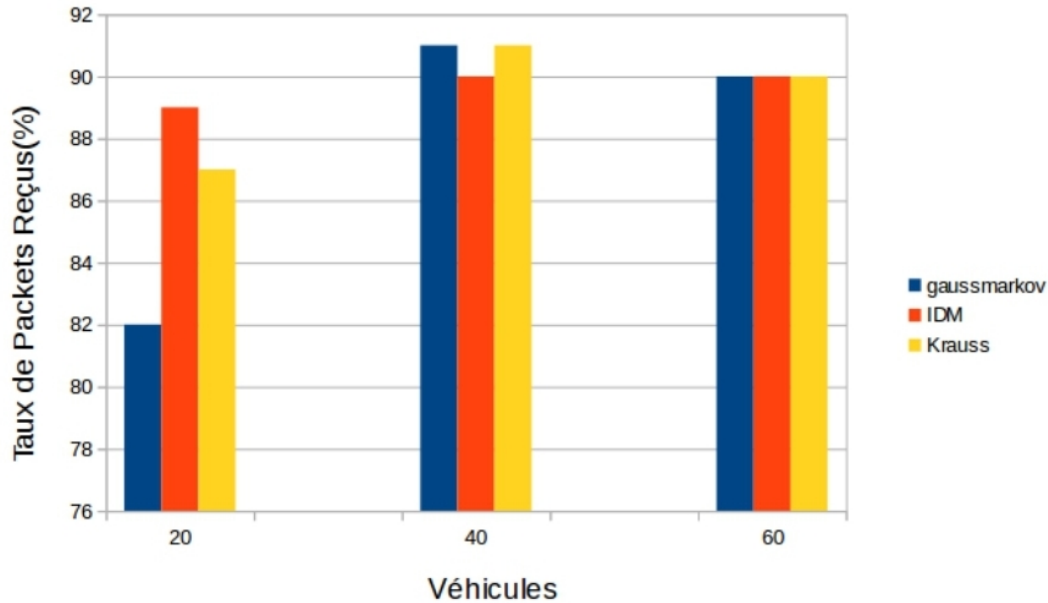


FIGURE 3.12 – Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole AODV à travers un histogramme.

En comparant le taux de livraison de paquets de différents modèles de mobilité et configurations de véhicules utilisant le protocole AODV, nous pouvons tirer les remarques suivants :

le modèle IDM se distingue comme étant le plus performant en termes de taux de livraison des paquets. Le modèle de Gaussmarkov fonctionne bien lorsque le nombre de véhicules augmente, tandis que le modèle de Krauss obtient des résultats compétitifs dans différentes configurations. Le protocole AODV fonctionne bien avec le modèle IDM.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

Résultats de simulation avec protocoles OLSR : Le tableau 3.3 présente les paquets reçus en utilisant le protocole OLSR, comme illustré dans la Figure 3.13

	vehicle20	vehicle40	vehicle60
Gaussmarkov	92,00%	95,00%	96,00%
IDM	94,00%	93,00%	95,00%
Krauss	95,00%	94,00%	95,00%

TABLE 3.3 – Les paquets reçus avec le protocole OLSR.

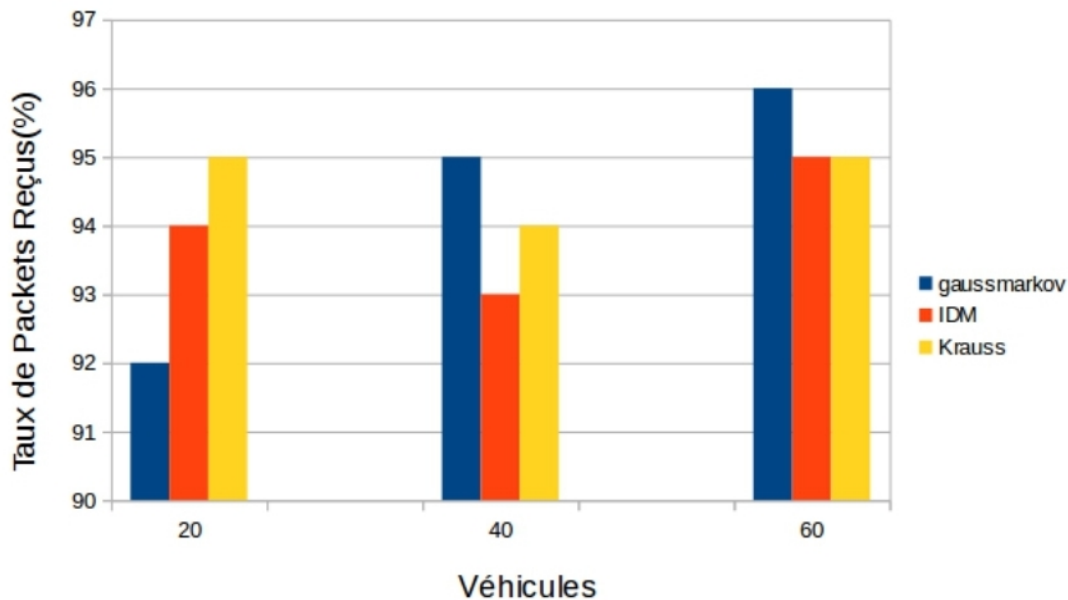


FIGURE 3.13 – Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole OLSR à travers un histogramme.

1. Modèle de mobilité de Gaussmarkov :
 - Avec le protocole OLSR, le modèle de Gaussmarkov démontre des taux de livraison de paquets améliorés par rapport au protocole AODV.
 - Dans toutes les configurations de véhicules, le taux de livraison des paquets varie de 92,00 % à 96,00 %.
2. Modèle de mobilité IDM (Intelligent Driver Model) :

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

- Le modèle IDM bénéficie également du protocole OLSR, les taux de livraison de paquets améliorés par rapport au protocole AODV.
 - Ce modèle fonctionne toujours bien.
3. Modèle de mobilité de Krauss :
- Comme le modèle IDM, le modèle Krauss démontre des taux de livraison de paquets améliorés avec le protocole OLSR.
 - Le taux de livraison des paquets varie de 94,00 % à 95,00 % sur différentes configurations de véhicules.

Avec le protocole OLSR, le modèle de Gaussmarkov montre des performances constantes, tandis que le modèle IDM et le modèle Krauss bénéficient également d'une amélioration des taux de livraison de paquets. Gaussmarkov se révèle plus fonctionnel lorsque le nombre de véhicules augmente.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

Résultats de simulation avec le protocole DSDV : Le tableau 3.4 présente les paquets reçus en utilisant le protocole DSDV, comme illustré dans la Figure 3.14

	Vehicule20	Vehicule40	Vehicule60
Gaussmarkov	89%	86%	85%
IDM	84%	81%	82%
Krauss	88%	83%	83%

TABLE 3.4 – Les paquets reçus avec le protocole DSDV.

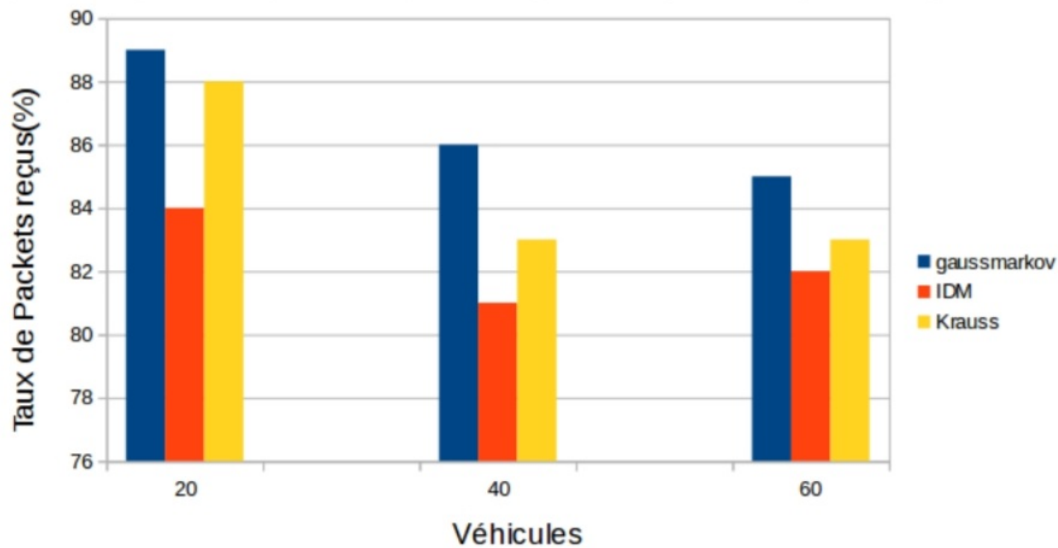


FIGURE 3.14 – Visualisation de la distribution des taux de réception de paquets des véhicules utilisant le protocole DSDV à travers un histogramme.

1. Modèle de mobilité de Gaussmarkov : Avec le protocole DSDV, le modèle de Gaussmarkov montre des taux de livraison de paquets moyen. Dans toutes les configurations de véhicules, le taux de livraison des paquets varie de 85 % à 89%. Bien qu'il soit encore efficace dans une certaine mesure, ce modèle présente des performances inférieures à celles des autres protocoles.
2. Modèle de mobilité IDM (Intelligent Driver Model) : Le modèle IDM présente également des taux de livraison de paquets moyen avec le protocole DSDV. Le taux de livraison des paquets varie de 81% à 84% sur

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

différentes configurations de véhicules. Semblable au modèle Gaussmarkov, il présente des performances inférieures à celles des autres protocoles.

3. Modèle de mobilité de Krauss : Le modèle Krauss ne parvient toujours pas à atteindre des taux de livraison de paquets élevés.

Dans dsdv nous voyons le taux de livraison de paquets diminuer à chaque fois que nous augmentons le nombre de véhicules nous avons Gaussmarkov ont les meilleures performances dans chaque scénario nous pouvons dire que gauss markov est plus fonctionnel lors de l'utilisation du protocole dsdv

Étude des performances de 20 Véhicules avec différents protocoles de routage

A- Protocole AODV : La Figure 3.15 présente une analyse graphique des performances du protocole AODV pour un ensemble de 20 véhicules.

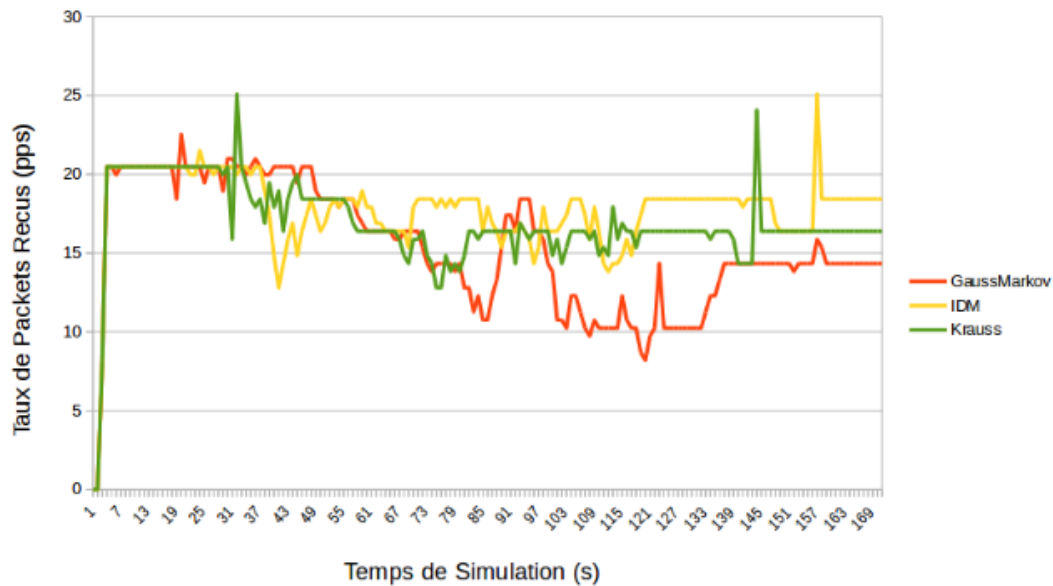


FIGURE 3.15 – Analyse graphique des performances de 20 véhicules avec le protocole AODV.

1. Gaussmarkov :

- Au début (pas de temps 0 et 1), les valeurs de Gaussmarkov sont toutes nulles.
- Les valeurs commencent à augmenter à partir du pas de temps 2 et atteignent une valeur maximale de 22,528 au pas de temps 19.
- Après cela, les valeurs fluctuent entre environ 14 et 22.

2. IDM :

- Le modèle IDM commence initialement par des zéros pour les pas de temps 0 et 1.
- Les valeurs augmentent progressivement et atteignent un maximum de 21,504 au pas de temps 23.
- Les valeurs fluctuent entre environ 14 et 21,5, similaire à Gaussmarkov.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

3. Kraus :

- Les valeurs augmentent progressivement et atteignent un maximum de 25,088 au pas de temps 31.
- Après cela, les valeurs fluctuent entre environ 12,8 et 25.

Sur la base de cette analyse, nous pouvons faire les observations suivantes :

- Les trois modèles commencent par des zéros, indiquant aucun mouvement initial.
- Gaussmarkov et IDM ont un comportement similaire et leurs valeurs fluctuent dans une plage similaire.
- Kraus montre des fluctuations plus importantes de ses valeurs, avec une plage plus large par rapport à Gaussmarkov et IDM.

B- Protocole OLSR : La Figure 3.16 présente une analyse graphique des performances du protocole OLSR pour un ensemble de 20 véhicules.

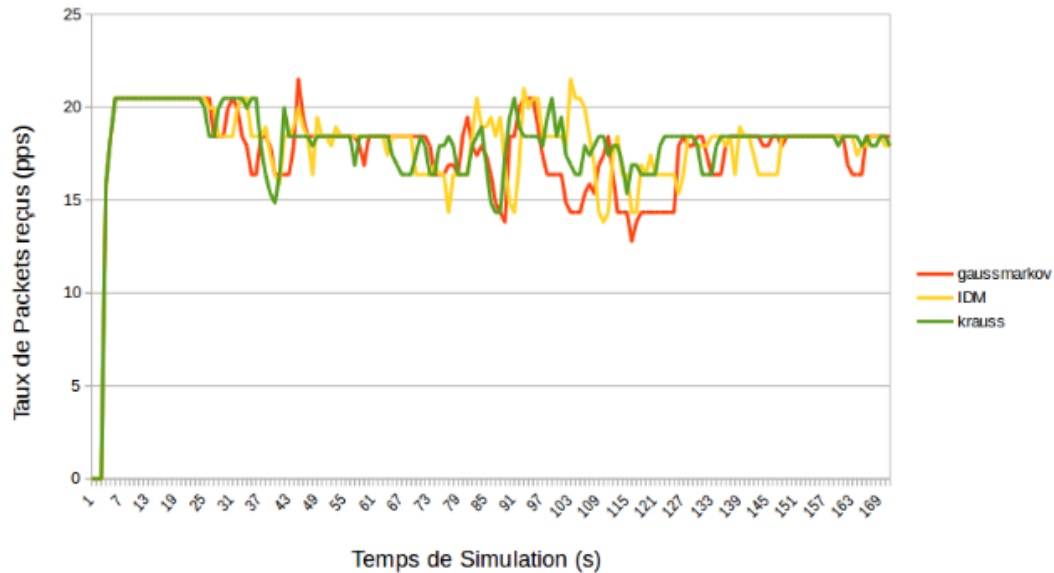


FIGURE 3.16 – Analyse graphique des performances du 20 véhicules avec le protocole OLSR.

1. Le modèle Gaussmarkov montre une valeur constante de 20,48 pendant la majeure partie de la simulation, le modèle Gaussmarkov maintient un état relativement stable avec des changements minimes au fil du temps.
2. Le modèle IDM (Intelligent Driver Model) présente également un comportement similaire, avec des valeurs qui restent principalement à 20,48. Cependant, il y a quelques cas où ce descendant à 19,968 ou 19,456, avant de revenir à 20,48.
3. En revanche, le modèle Krauss montre plus de variabilité par rapport aux deux autres modèles. Bien qu'il commence avec une valeur de 15,872, similaire aux modèles Gaussmarkov et IDM, le modèle Krauss atteint une valeur maximale de 21,504 et descend aussi bas que 13,824. Vers la fin de la simulation, le modèle Krauss se stabilise à une valeur de 17,92.

C- Protocole DSDV : La Figure 3.17 présente une analyse graphique des performances du protocole DSDV pour un ensemble de 20 véhicules.

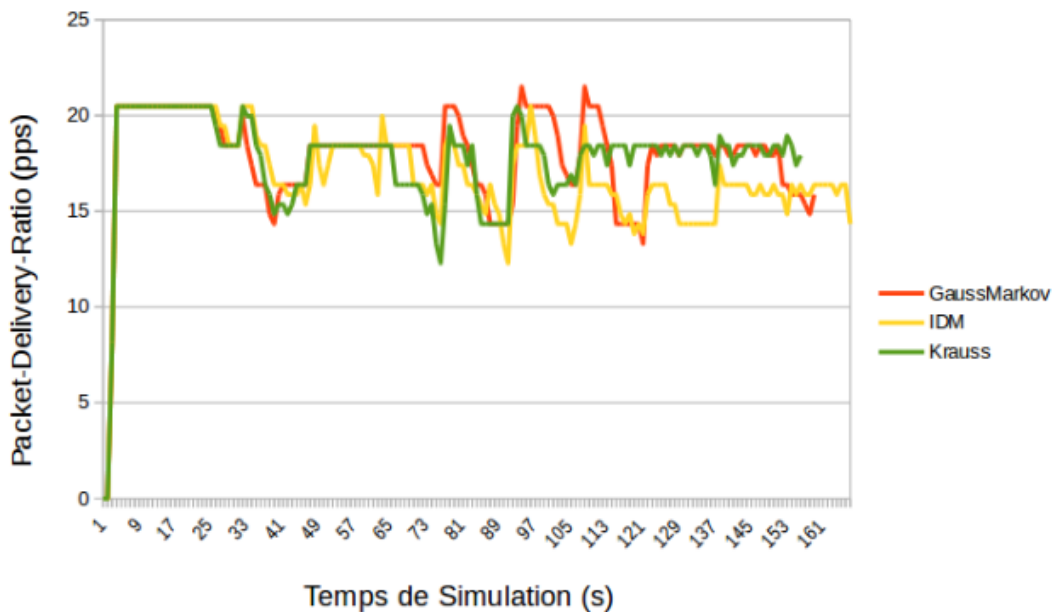


FIGURE 3.17 – Analyse graphique des performances du 20 véhicules avec le protocole DSDV.

1. Le modèle Gaussmarkov reste constante à 20,48 pour la plupart des points de données, avec quelques variations vers la fin.
2. Le modèle IDM suit un schéma similaire au modèle Gaussmarkov, restant également constant à 20,48 pour la plupart des points de données, avec de légères variations vers la fin.
3. Le modèle de Krauss commence à 8,192 et reste constante à 20,48 pour la majorité des points de données, avec quelques fluctuations vers la fin.

En regardant les données, il apparaît que les modèles Gaussmarkov et Krauss ont des valeurs sont exactement les mêmes pour toutes les entrées, tandis que le modèle IDM a des valeurs légèrement différentes.

Étude des performances de 40 Véhicules avec différents protocoles de routage

A- Protocole AODV : La Figure 3.18 présente une analyse graphique des performances du protocole AODV pour un ensemble de 40 véhicules.

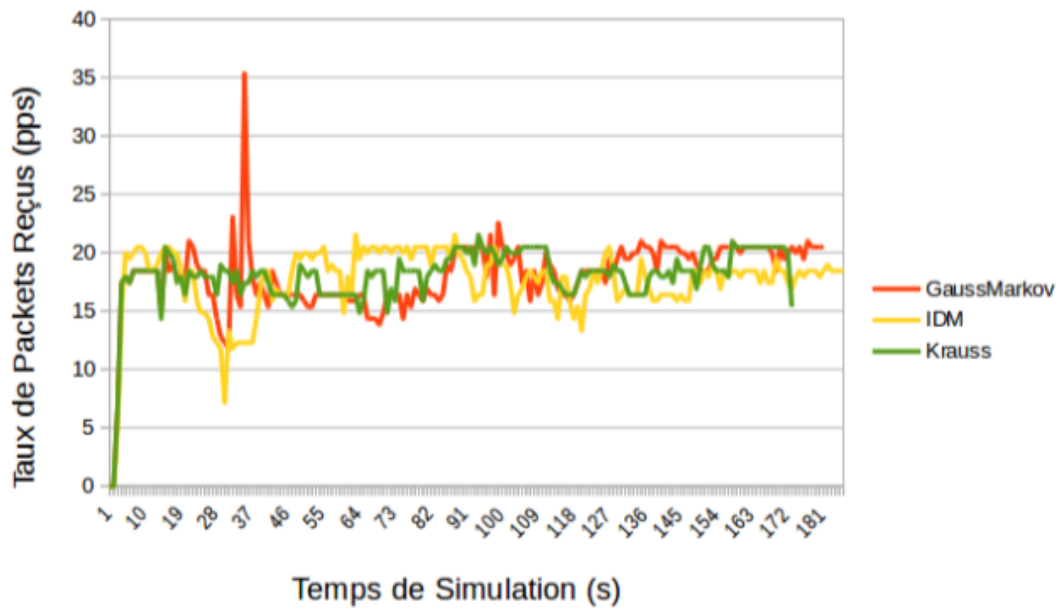


FIGURE 3.18 – Analyse graphique des performances du 40 véhicules avec le protocole AODV.

1. Modèle de Gauss Markov : Les valeurs du modèle de Gauss semblent varier au fil du temps, allant de 11,776 à 35,328. Les valeurs semblent suivre un schéma aléatoire.
2. IDM (Modèle du Conducteur Intelligent) : Les valeurs du modèle IDM vont de 5,12 à 21,504. Les valeurs du modèle IDM montrent un schéma relativement plus régulier que celui du modèle de Gauss.
3. Modèle de Krauss : Les valeurs du modèle de Krauss varient entre 14,336 et 23,04. De manière similaire au modèle IDM.

B- Protocole OLSR : La Figure ?? présente une analyse graphique des performances du protocole OLSR pour un ensemble de 40 véhicules.

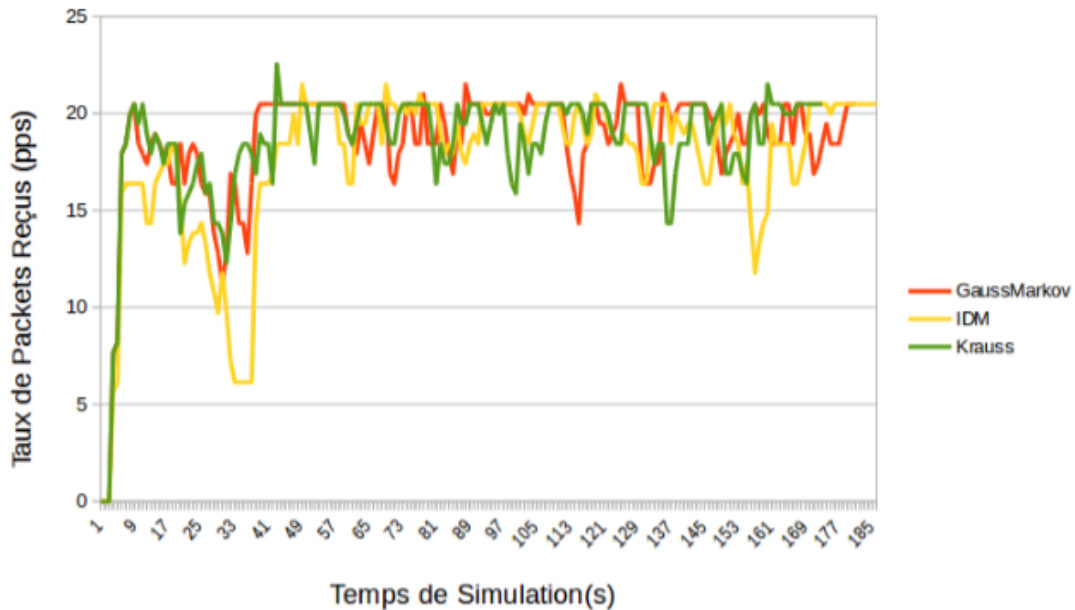


FIGURE 3.19 – Analyse graphique des performances du 40 véhicules avec le protocole OLSR.

1. Gaussmarkov : Le modèle Gaussmarkov semble avoir des valeurs relativement stables. Les valeurs commencent à 0 et augmentent progressivement, atteignant un maximum de 21,504, les variables restent constantes.
2. IDM (Intelligent Driver Model) : Le modèle IDM montre plus de variabilité dans sa variable par rapport à Gaussmarkov. Les valeurs commencent à 0 et augmentent avec le temps, la valeur maximale atteinte par la variable IDM est de 21,504, similaire à Gaussmarkov. Cependant, Le modèle IDM affiche des changements dans ses valeurs.
3. Krauss : Le modèle Krauss présente un schéma différent par rapport à Gaussmarkov et IDM. Sa variable commence à 0 et atteignant une valeur maximale de 22,528. Cependant, après ce pic, les valeurs diminuent à 20,48.

C- Protocole DSDV : La Figure 3.20 présente une analyse graphique des performances du protocole DSDV pour un ensemble de 40 véhicules.

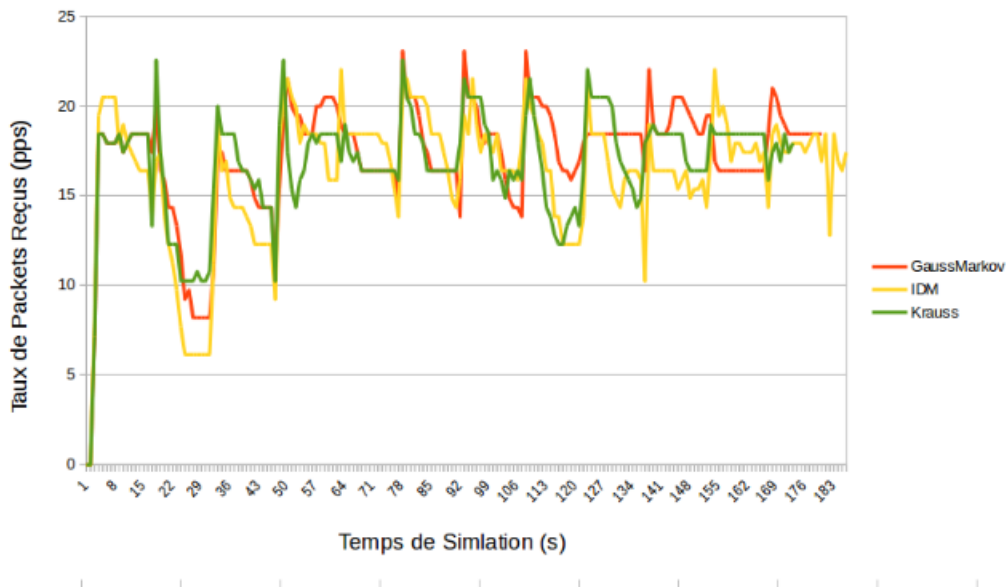


FIGURE 3.20 – Analyse graphique des performances du 40 véhicules avec le protocole DSDV.

1. Gaussmarkov :

- Le modèle de Gaussmarkov présente un taux de livraison de paquets qui changent régulièrement.
- Il commence avec 0% de livraison de paquets dans les deux premières secondes et augmente jusqu'à un maximum de 18,432%

2. IDM :

- Le modèle IDM dépasse le modèle de Gaussmarkov en termes de livraison de paquets.
- IDM présente des taux de livraison de paquets allant de 8,192% à 21,504%, avec des fluctuations occasionnelles.

3. Krauss :

- Le modèle de Krauss présente également des caractéristiques similaires aux modèles Gaussmarkov et IDM.
- Le taux de livraison de paquets allant de 7,168% à 22,528%.

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

— Alors que le modèle de Krauss fonctionne mieux que Gaussmarkov.

En comparant la fonctionnalité de ces modèles, IDM apparaît comme le modèle le plus fonctionnel et le plus fiable pour la livraison de paquets. Les taux de livraison plus élevés par rapport à Gaussmarkov et Krauss.

Étude des performances de 60 Véhicules avec différents protocoles de routage

A- Protocole AODV : La Figure 3.21 présente une analyse graphique des performances du protocole AODV pour un ensemble de 60 véhicules.

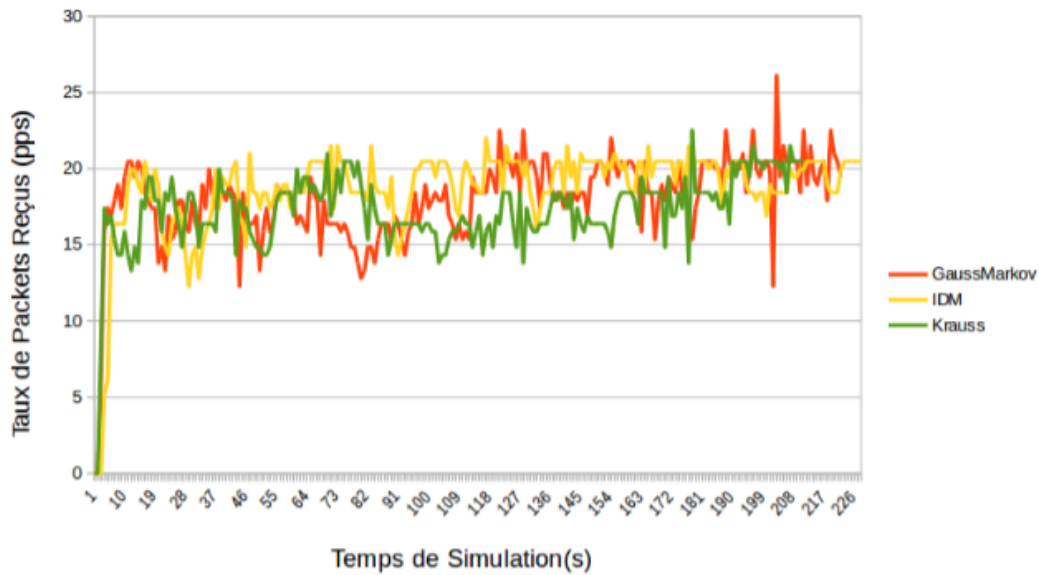


FIGURE 3.21 – Analyse graphique des performances du 60 véhicules avec le protocole AODV.

1. Gaussmarkov : Les valeurs de Gaussmarkov semblent augmenter progressivement au fil du temps. Les valeurs sont relativement constantes.
2. IDM : Les valeurs augmentent généralement.
3. Kraus : Les valeurs restent relativement constante.

B- Protocole OLSR : La Figure 3.22 présente une analyse graphique des performances du protocole OLSR pour un ensemble de 60 véhicules.

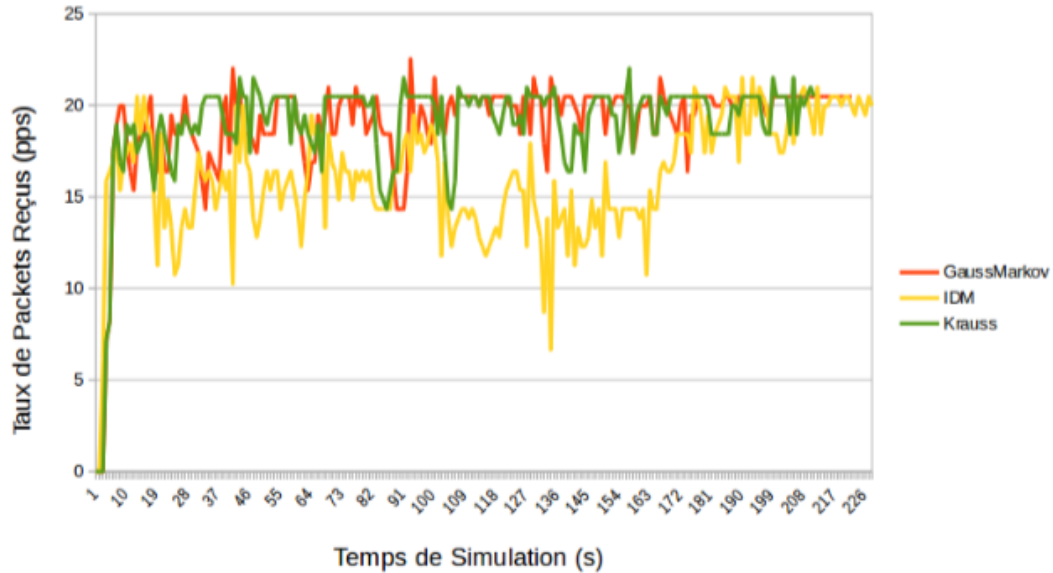


FIGURE 3.22 – Analyse graphique des performances du 60 véhicules avec le protocole OLSR.

- Le modèle Gaussmarkov montre des valeurs cohérentes à travers toutes les itérations de la simulation. Les valeurs restent relativement stables tout au long de la simulation.
- Le modèle IDM présente des changements dans ses valeurs tout au long de la simulation.
- Le modèle Krauss montre un changements de comportement.

C- Protocole DSDV : La Figure 3.23 présente une analyse graphique des performances du protocole DSDV pour un ensemble de 60 véhicules.

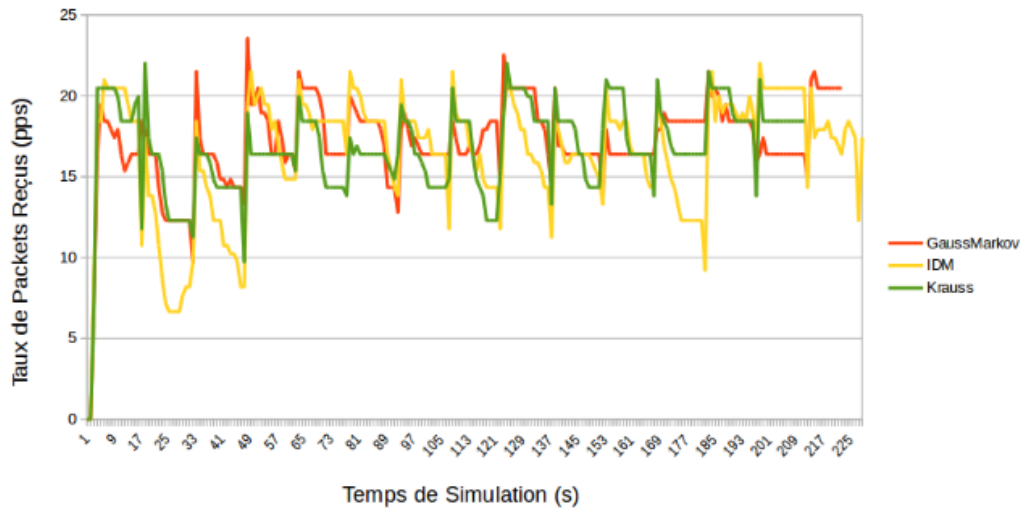


FIGURE 3.23 – Analyse graphique des performances du 60 véhicules avec le protocole DSDV.

- Le modèle Gaussmarkov présentait le taux de livraison de paquets le plus élevé parmi les trois modèles. Cela indique que le modèle Gaussmarkov a démontré le taux de réussite le plus élevé dans la livraison de paquets par rapport à IDM et Krauss.
 - Le modèle Krauss a affiché un niveau modéré de livraison de paquets, indiquant une performance décente en termes de fiabilité.
 - IDM a montré le taux de livraison de paquets le plus bas de notre analyse.
- Sur la base de nos résultats, il est évident que le modèle de Gaussmarkov a surpassé les modèles de Krauss et IDM en termes de taux de livraison de paquets.

3.5 Conclusion

Le nombre de véhicules dans un scénario peut également avoir un impact sur le choix du modèle de mobilité. Dans les scénarios avec un petit nombre de véhicules, les interactions entre les véhicules peuvent avoir un impact plus important. Les scénarios avec un grand nombre de véhicules peuvent nécessiter des modèles de mobilité basée sur le comportement collectif

En conclusion, la mise en place d'une simulation de réseau VANET nécessite la réalisation de plusieurs étapes. La génération de la carte de simulation, la modélisation de la mobilité des véhicules, la génération du trafic, la simulation des protocoles de routage et l'utilisation de l'outil NS-3 pour la simulation du réseau VANET.

Dans ce Chapitre, nous avons montré décrit toutes étapes des simulations que nous avons conduites et qui nous permis de mesurer l'impact de différents paramètres sur les performances du réseau et de fournir des résultats précis pour aider à la prise de décision.

Conclusion générale

Les réseaux VANETs représentent une avancée. De nombreux travaux théoriques ont été réalisés dans le domaine des réseaux et plusieurs expériences ont vérifié l'hypothèse selon laquelle la mise en place de cette structure est coûteuse, mais que des efforts supplémentaires sont attendus dans un avenir proche. Cela permettrait non seulement d'apporter du plaisir, mais aussi de sauver des vies.

La communication sans fil n'est pas fiable. Le taux d'erreur des réseaux locaux sans fil est beaucoup plus élevé. Tous ces problèmes rendent la mise en oeuvre des réseaux difficile. Dans le cadre de ce projet, nous avons produit et étudié différents modèles de mobilité et différents protocoles de routage pour les VANETs. La simulation est essentielle, car des facteurs réalistes tels que la mobilité des véhicules et la communication sans fil influencent fortement les résultats de l'évaluation.

Nous avons utilisé les simulateurs SUMO et NS-3. Trois modèles de mobilité ont été utilisés dans le cadre du projet - Gaussmarkov, IDM (Intelligent Driver Model) et Krauss.

En conclusion, notre étude souligne l'importance des modèles de mobilité pour simuler avec précision les performances des protocoles de routage dans les réseaux VANETs. Les modèles de mobilité proposés dans cette étude capturent les détails spécifiques des intersections et des mouvements de véhicules qui ont un impact significatif sur les performances de routage. Nous avons aussi évalué la communication inter-véhiculaires en utilisant 3 protocoles de routage. D'autres recherches sont nécessaires pour améliorer ces modèles et mieux comprendre la dynamique de la mobilité dans les réseaux VANETs et surtout faire de la prédiction.

Annexe : Installation des outils

Introduction

Cette annexe fournit les instructions pour l'installation de plusieurs outils nécessaires à la simulation, à savoir le simulateur NS-3 et le simulateur de trafic SUMO.

Installation du simulateur NS-3

Téléchargement et décompression de NS-3

Il existe deux façons différentes de télécharger NS-3 :

- Méthode 1 : Téléchargement direct depuis le site <https://www.nsnam.org>.
- Méthode 2 : Utilisation du terminal avec les commandes suivantes :
 - Créez un dossier en entrant la commande : `mkdir repos` (ce n'est pas obligatoire).
 - Accédez au dossier en entrant la commande : `cd repos`.
 - Téléchargez NS-3 en entrant la commande : `wget http://www.nsnam.org/releases/ns-allinone-3.31.tar.bz2`.

Pour décompresser et convertir le fichier tar en fichier bz2, utilisez la commande suivante :

```
tar jxvf ns-allinone-3.31.tar.bz2
```

Ensuite, compilez NS-3 en utilisant les commandes suivantes :

- Accédez au dossier NS-3 avec la commande : `cd ns-allinone-3.31/`

- Lancez la compilation en entrant les commandes :
`./build.py --enable-examples --enable-tests`

Installation du simulateur SUMO

Étapes d'installation

Suivez les étapes ci-dessous pour installer SUMO :

Étape 1 :

Mettez à jour votre système et installez les dépendances nécessaires avec les commandes suivantes :

- `sudo apt update`
- `sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev libgl2ps-dev swig`

Étape 2 :

Installez Git pour cloner le référentiel SUMO en entrant la commande suivante :

- `sudo apt install git`
- `git clone --recursive https://github.com/eclipse/sumo`

Étape 3 :

Définissez la variable d'environnement `SUMO_HOME` en entrant la commande suivante :

```
export SUMO_HOME="home/zeroual/sumo"
```

Étape 4 :

Créez un dossier de construction et générez les fichiers de configuration pour la compilation en utilisant les commandes suivantes :

- `mkdir sumo/build/cmake-build`

CHAPITRE 3. IMPLEMENTATION, SIMULATION ET ÉVALUATION DES RÉSULTATS

- `cd sumo/build/cmake-build`
- `cmake ../..`
- `make -j8`

Bibliographie

- [1] Kahina Ait Ali. *Modélisation et étude de performances dans les réseaux VANET*. PhD thesis, Université de Technologie de Belfort-Montbéliard, 2012.
- [2] JP Hubaux. Vehicular networks how to secure them. *MiNeMa Summer School*, 2005.
- [3] Xiaodong Lin, Rongxing Lu, Chenxi Zhang, Haojin Zhu, Pin-Han Ho, and Xuemin Shen. Security in vehicular ad hoc networks. *IEEE communications magazine*, 46(4) :88–95, 2008.
- [4] Djamel Bektache, Cherif TOLBA, and Nassira GHOUALMI. *Application et Modélisation d'un Protocole de Communication pour la Sécurité Routière*. PhD thesis, Université Badji Mokhtar Annaba, 2014.
- [5] Melle MEHARZA Wafa. *Etude de la mobilité dans les réseaux Ad hoc véhiculaires VANET*. PhD thesis, UNIVERSITE BADJI MOKHTAR ANNABA, 2019.
- [6] Sumo emission of véhicules. <https://www.nsnam.com/2023/04/sumo-emission-of-vehicles-vanets.html?m=1>, last accessed on 2023-05-24.
- [7] Muriel Aubert. Les réseaux informatiques. *Université de Nice Sophia Antipolis*, 2014.
- [8] Paul Muhlethaler and Olivier Salvatori. *cours 802.11 et les réseaux sans fil*. Eyrolles Paris, 2002.

- [9] Fatma LAIDOUI. Approche bas'ee sur la confiance pour l'etablissement des routes dans les r'eseaux ad hoc mobiles.
- [10] J Van der Meerschen. Hybridation entre les modes adhoc et infrastructure dans les r'eseaux de type wi-fi, 2006.
- [11] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. Mobile ad hoc networking imperatives and challenges. *Ad hoc networks*, 1(1) :13–64, 2003.
- [12] KHADIDJA AYAD. *S'ecurit'e du routage dans les r'eseaux ad hoc mobile*. PhD thesis, Ecole Nationale Sup'erieure d'Informatique, 2012.
- [13] Vinita Jindal and Punam Bedi. Vehicular ad-hoc networks introduction, standards, routing protocols and challenges. *International Journal of Computer Science Issues (IJCSI)*, 13(2) :44, 2016.
- [14] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37 :380–392, 2014.
- [15] Hassnaa Moustafa and Gilles Bourdon. Vehicular networks deployment view applications, deployment architectures and security means. *Ubiquitous Computing and Communication Journal, special issue on Ubiquitous Roads*, 3, 2008.
- [16] Andreas Festag, H Fussler, Hannes Hartenstein, Amardeo Sarma, and Ralf Schmitz. Fleetnet bringing car-to-car communication into the real world. *Computer*, 4(L15) :16, 2004.
- [17] Feliz Kristianto Karnadi, Zhi Hai Mo, and Kun-chan Lan. Rapid generation of realistic mobility models for vanet. In *2007 IEEE wireless communications and networking conference*, pages 2506–2511. IEEE, 2007.
- [18] Djamel Djenouri, Wassim Soualhi, and Elmalik Nekka. Vanet's mobility models and overtaking an overview. In *2008 3rd International Conference on Information and Communication Technologies From Theory to Applications*, pages 1–6. IEEE, 2008.

- [19] Oumaima El Joubari. *Mobility and traffic models for VANETs*. PhD thesis, Universit'e Paris-Saclay, 2022.
- [20] Kenneth Laberteaux and Hannes Hartenstein. *VANET vehicular applications and inter-networking technologies*. John Wiley Sons, 2009.
- [21] Qunwei Zheng, Xiaoyan Hong, and Sibabrata Ray. Recent advances in mobility modeling for mobile ad hoc network research. In *Proceedings of the 42nd annual Southeast regional conference*, pages 70–75, 2004.
- [22] Nikoletta Sofra, Athanasios Gkelias, and Kin K Leung. Route construction for long lifetime in vanets. *IEEE Transactions on Vehicular Technology*, 60(7) :3450–3461, 2011.
- [23] Adel DEGAICHIA. *Une extension de la plateforme Omnet++ pour la gestion de la mobilit'e dans les VANET-CLOUD*. PhD thesis, Universite laarbi tebessi tebessa, 2019.
- [24] Robbi C Manurung, Doan Perdana, and Rendy Munadi. Performance evaluation gauss-markov mobility model in vehicular ad-hoc network with spearman correlation coefficient. In *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 351–356. IEEE, 2016.
- [25] PH Gordon, Doan Perdana, and Rendy Munadi. Performance evaluation of vanet with edca access scheme using gauss-markov mobility model. *International Journal of Simulation Systems, Science Technology*, pages 1–7, 2017.
- [26] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. Important a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 2, pages 825–835. IEEE, 2003.
- [27] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5) :483–502, 2002.

- [28] Dominik Salles, Stefan Kaufmann, and Hans-Christian Reuss. Extending the intelligent driver model in sumo and verifying the drive off trajectories with aerial measurements. In *SUMO Conference Proceedings*, volume 1, pages 1–25, 2020.
- [29] traffic simulation. https://traffic-simulation.de/info/info_IDM.html, last accessed on 2023-05-26.
- [30] Xue Yuan, Bai Yun, and Chunming Li. Simulation study on the vehicle mobility model based on the sumo. In *3rd International Conference on Mechatronics, Robotics and Automation*, pages 934–938. Atlantis Press, 2015.
- [31] Mohd Shareef. Analysis and development of a car-following model with xgcd. 2020.
- [32] Abdel Mehsen Ahmad. *Techniques de transmission et d'accès sans fil dans les réseaux ad-hoc véhiculaires (VANETS)*. PhD thesis, Evry, Institut national des télécommunications, 2012.
- [33] Pablo Barbecho Bautista, Luis Urquiza Aguiar, and Mónica Aguilar Igartua. How does the traffic behavior change by using sumo traffic generation tools. *Computer Communications*, 181 :1–13, 2022.
- [34] Davide Andrea Guastella and Gianluca Bontempi. Traffic modeling with sumo : a tutorial. *arXiv preprint arXiv :2304.05982*, 2023.
- [35] Imane Karima BOUABDALLAH and Yasmina BOUDERBANE. *Etude de la norme 802.11 v dans les réseaux WLAN mobile sous l'environnement de simulation NS-3*. PhD thesis.
- [36] Namita Chandel and Mr Vishal Gupta. Comparative analysis of aodv, dsr and dsdv routing protocols for vanet city scenario. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(6) :1380–1384, 2014.
- [37] Radhika Chauhan, Nitin Goyal, and Rakesh Kumar. A review on tuning of olsr routing protocol in vanet. *International Journal of Advanced Research and Innovative Ideas in Education*, 2(2) :508–512, 2016.

- [38] Jerome Harri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks a survey and taxonomy. *IEEE communications surveys tutorials*, 11(4) :19–41, 2009.
- [39] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.
- [40] Christos Profentzas. Studying routing issues in vanets by using ns-3. *Alexander Technological Educational Institute of Thessaloniki in November*, 2012.