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

Université de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER RECHERCHE

En

Informatique

Option

Réseaux et Systèmes Distribués

Thème

Routage opportuniste pour le signalement d'infractions routières dans l'IoV

Présenté par :
M. HAMADACHE Smail
M. SIBY Sory Ibrahim

Soutenu le 6 Juillet 2022 devant le jury composé de :

Présidente	**Dr. M. Yaici**	**MCB**	**Université de Béjaia**
Promotrice	**Pr. L. Bouallouche**	**Professeur**	**Université de Béjaia**
Co-promotrice	**Dr. S. Yessad**	**MCB**	**Université de Béjaia**
Examinatrice	**Dr. S. Lahlah**	**MCB**	**Université de Béjaia**
Examinateur	**M. A. Beraza**	**Doctorant LMD**	**Université de Béjaia**

Béjaia, Juillet 2022.

⁂ *Remerciements* ⁂

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voulons témoigner toute notre reconnaissance. Nous voulons tout d'abord adresser toute notre gratitude à notre promotrice le professeur L. BOUALLOUCHE et à notre co-promotrice le docteur S. YESSAD, pour leur patience, leur disponibilité et surtout pour leurs judicieux conseils qui ont contribué à alimenter notre réflexion. Nous voulons exprimer notre reconnaissance envers nos amis et camarades qui nous ont apporté leur support moral et intellectuel tout au long de notre démarche.

⁂ *Dédicaces* ⁂

Nous dédions ce mémoire à nos parents qui nous ont soutenus et encouragés durant ces années d'études. Qu'ils trouvent ici le témoignage de notre profonde reconnaissance. A nos frères et sœurs, nos grands-parents et ceux qui ont partagé avec nous tous les moments d'émotion lors de la réalisation de ce travail. Ils nous ont chaleureusement supportés et encouragés tout au long de notre parcours. A nos familles, nos proches et à ceux qui nous donnent de l'amour et de la vivacité. A tous nos amis qui nous ont toujours encouragés, et à qui nous souhaitons plus de succès. A tous ceux qu'on aime.

M. SIBY Sory Ibrahim
M. HAMADACHE Smail

Table des matières

Table des figures

Liste des tableaux

Liste des acronymes

DTN	Delay Tolerant Networking.
IEEE	Institute of Electrical and Electronics Engineers.
IoT	Internet of Things.
IoV	Internet of Vehicles.
MANET	Mobile Ad hoc NETwork.
MOP	Mobility-aware Opportunistic routing Protocol.
OBU	On Board Unit.
ONE	Opportunistic Networking Environment.
POI	Point Of Interest.
PROPHET	Probabilistic ROuting Protocol using History of Encounters and Transitivity.
RSU	Road Side Unit.
SV	Summary Vector.
TDOR	Trajectory-Driven Opportunistic Routing protocol.
VANET	Vehicular Ad hoc NETwork.

Introduction générale

La sécurité routière dans notre ère est une problématique qui devient de plus en plus complexe à maîtriser. Mais avec les dernières tendances des techniques qui touchent le secteur automobile, et l'évolution exponentielle de l'intelligence artificielle cela est devenu possible.

Avec l'apparition de l'Internet des Objets (IdO ou en anglais IoT (Internet of things)), non seulement les objets de notre quotidien peuvent être connectés à Internet, mais aussi ces derniers peuvent agir comme des agents intelligents. Ainsi, ils pourront interagir, collaborer, et éventuellement rentrer en concurrence entre eux pour différentes finalités. Par exemple, dans les IoVs (Internet of Vehicle) qui est un sous-type de IoT, les véhicules interagissent et collaborent pour assurer la sécurité routière et le confort des conducteurs ou bien pour aider les différents services d'une ville dans leurs différentes tâches. Dans ce mémoire, nous nous intéressons à ce dernier type d'applications des IoVs en proposant un système de signalement d'infractions routières. Cette application a pour finalité le signalement de toutes les infractions routières qu'un conducteur pourrait commettre (excès de vitesses, non-respect de la signalisation routière, conduite en état d'ivresse, etc.) aux services de police de la ville, et cela grâce à des capteurs et récepteurs qui seront embarqués dans les véhicules. Cependant, cette application peut aider aussi à la sécurité routière puisque la plupart des accidents sont dus aux erreurs humaines et en violant le code de la route et si ces violations sont toutes rapportées et les commetteurs sont punis, les infractions seront réduites ainsi que les accidents de la route.

Cette application est destinée pour les réseaux véhiculaires tolérants les délais, et cela nécessite une communication entre les véhicules et pour que ces derniers puissent communiquer il faut un mécanisme d'acheminement des données d'où un protocole de routage. Ainsi, nous nous sommes intéressés à la couche 3 du modèle OSI, qui est la couche réseau, donc nous nous sommes focalisés sur la manière avec laquelle les messages d'infractions routières vont circuler dans notre réseau véhiculaire pour être délivrés aux services de police. Donc,

dans ce mémoire, nous allons essayer de répondre à la question suivante : quel est le protocole de routage le mieux adapté pour notre application de signalement d'infractions routières ? et quelles sont les améliorations que nous pouvons apporter à ce dernier pour augmenter les performances de notre application ?

Pour répondre à cette question, nous avons commencé par étudier les protocoles de routage proposés pour les réseaux tolérants les délais (DTNs pour Delay Tolerent Networks) et les réseaux véhiculaires. Puis, nous avons adapté les protocoles Epidemic et Prophet à notre application et nous avons évalué les performances de ces derniers avec le simulateur ONE. D'après les résultats de simulation, nous constatons que Prophet est très adapté pour notre application mais nécessite comme même une amélioration. Ainsi, nous avons proposé une amélioration de Prophet où les véhicules calculent une matrice de probabilités de rencontre avec le destinataire en fonction du jour de la semaine et l'heure dans la journée. Ceci va permettre de transmettre les messages à un véhicule dont la probabilité de rencontre avec la destination dont les heures à venir est plus importante.

Pour la réalisation de ce projet, nous avons structuré l'ensemble de notre travail en trois chapitres :

Le premier est consacré pour les définitions et la terminologies nécessaires afin de bien situer ce travail, et pour comprendre la majorité des concepts sur lesquels on va rebondir tous au long de ce mémoire, à savoir c'est quoi l'IOT, IoV, VANET, MANET ect.

Dans le deuxième chapitre, nous allons donner un état de l'art sur le routage opportuniste, afin d'avoir une idée sur ce qui a été proposé au paravent dans la littérature.

Dans le dernier chapitre, nous allons faire des tests et des simulations du routage des messages émit par l'application de signalement d'infractions routières sur le simulateur ONE, avec deux protocoles de routages qu'on va choisir comme candidats dans l'état de l'art, afin d'élire qu'un seul à la fin en fonction des performances et des résultats obtenus par ces derniers.

Enfin, nous allons achever ce travail avec une conclusion générale et quelques perspectives.

Chapitre 1

Généralités

1.1 Introduction

Avec l'accélération de l'innovation, il est facile d'imaginer un monde où les voitures sont autonomes ou volantes même. Mais des voitures qui parlent et qui pensent ? Vous avez beau ne pas y croire, mais c'est exactement vers cela que la mobilité se dirige. L'internet des véhicules (IoV), qui est un sous domaine de l'internet des objets (IoT), fait référence à la communication intelligente entre véhicules ou appareils électroniques, par l'échange d'informations stockées en cloud. De telles applications automatisées peuvent être très bénéfiques en termes de sécurité routière, de gestion de la circulation, de surveillance urbaine ou de signalement d'infractions routières.

Dans ce chapitre, nous allons présenter les concepts et les terminologies qui sont en relation avec ce domaine de recherche et notre projet.

1.2 Définitions de l'IoT

L'internet des objets (IdO), souvent écrit IoT (Internet of Things),le terme a été utilisé pour la première fois, en 1999 par le Britannique Kevin Ashton [6]. Son but était de décrire un système dans lequel les objets de notre monde physique pourraient être connectés à internet (comme illustré dans la figure 1.1) grâce à des capteurs. La littérature s'accorde à dire que l'IoT est l'internet du futur, ou le deviendra comme déclarent certains auteurs. Cependant, il n'y a aucune définition universelle.
Plusieurs définitions sont utilisées par des groupes différents dans le but de décrire une particularité du concept et de ses attributs les plus importants [38].

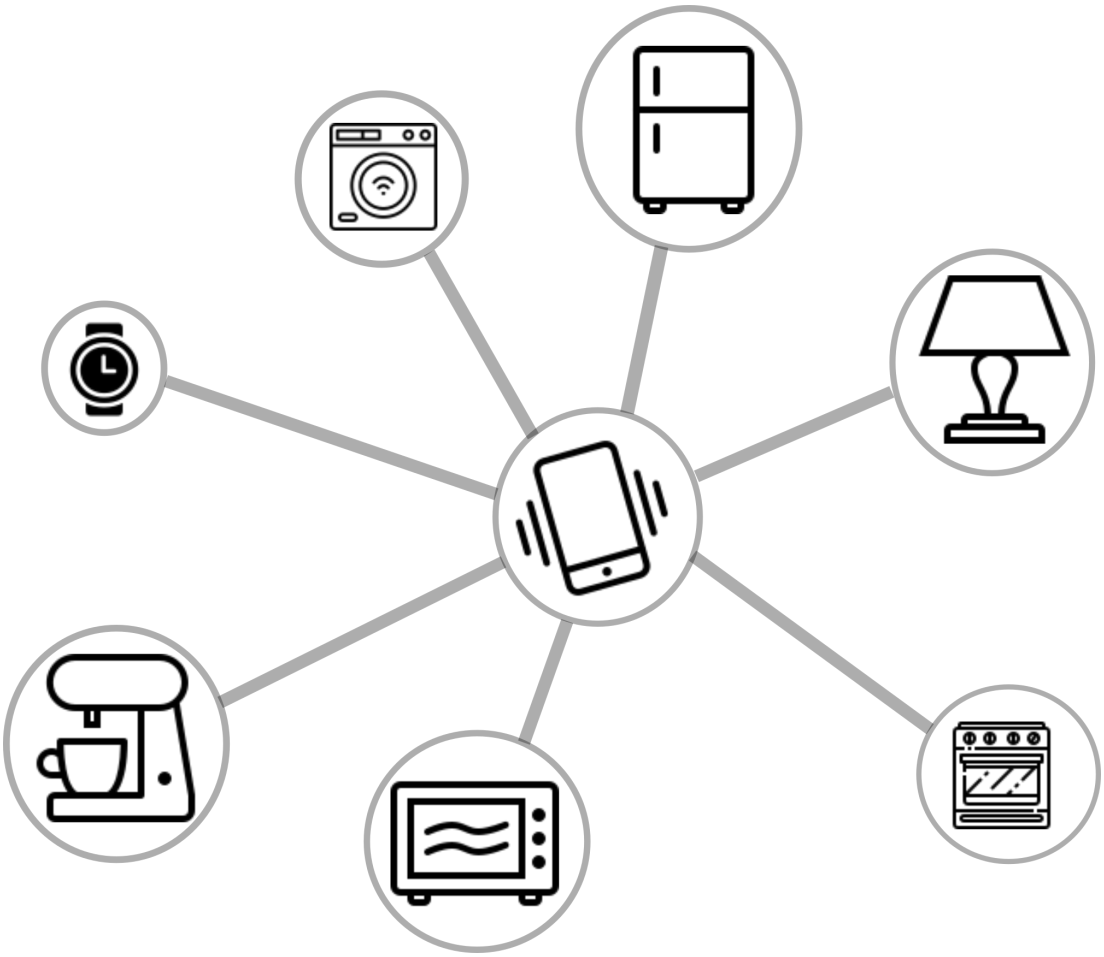

FIGURE 1.1 – Illustration de différents objets du quotidien connectés à Internet.

Reprenons quelques définitions du concept :

Celle des dictionnaires d'Oxford : Elle évoque que l'internet est un élément de l'IoT « L'interconnexion via Internet de dispositifs informatiques embarqués dans des objets du quotidien, leur permettant d'envoyer et de recevoir des données » [17].

Celle de l'union international des télécommunications [16] : L'union international des télécommunications propose une formule du concept qui discute de l'interconnectivité sans toutefois spécifier le lien entre IoT et internet : « infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en inter-connectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication (TIC) interopérables existantes ou en évolution ».

Celle de Kevin Ashton [6] : Le co-fondateur de l'Auto-ID Center du MIT (Massachusetts Institute of Technology) a employé le terme « Internet Of Things (Internet des Objets) » en 1999. IdO a été prononcé dans le cadre d'une présentation pour l'entreprise Procter & Gamble (P&G). Ce terme invoque le monde d'objets, d'appareils et de capteurs qui sont inter-connectés par internet.

Celle du CERP-IdO (Cluster des projets européens de recherche sur l'Internet des objets) : Il définit l'Internet des objets comme : « une infrastructure dynamique d'un réseau global. Ce réseau global a des capacités d'auto-configuration basées sur des standards et des protocoles de communication interopérables. Dans ce réseau, les objets physiques et virtuels ont des identités, des attributs physiques, des personnalités virtuelles et des interfaces intelligentes». [36]

Cette dernière définition montre les deux aspects de l'IdO : temporel et spatial qui permettent aux personnes de se connecter de n'importe où à n'importe quel moment à travers des objets connectés (Smartphones, tablettes, capteurs, caméras de vidéo-surveillance, etc). L'Internet des objets doit être pensé pour un usage facile et une manipulation sécurisée pour éviter des menaces et risques potentiels, tout en masquant la complexité technologique sous-jacente. L'évolution rapide de l'IoT bouleverse les frontières entre l'ordinateur et les produits du quotidien, cela est dû à deux facteurs : la généralisation des ressources informatiques et l'appropriation des services Web par les utilisateurs.

1.3 Qu'est ce qu'un MANET ?

1.3.1 Définition

Un réseau ad hoc mobile MANET (Mobile Ad hoc NETwork) ou un réseau ad hoc sans fil WANET(Wireless Ad hoc NETwork) est un type de réseau sans fil décentralisé. Un réseau ad hoc ne repose pas sur une infrastructure préexistante, telle que des routeurs dans les réseaux câblés ou des points d'accès dans les réseaux sans fil. Au lieu de cela, chaque nœud participe au routage en transférant des données pour d'autres nœuds, de sorte que la détermination des nœuds qui transmettent les données est effectuée dynamiquement sur la base de la connectivité réseau et de l'algorithme de routage utilisé [41].

Au début, le nom MANET a été attribué à un groupe de travail de l'IETF (Internet Engineering Task Force), créé entre 1998 et 1999, chargé de standardiser des protocoles de routage basés sur la technologie IP (Internet Protocol) pour les réseaux ad hoc sans fil [30].

Depuis la naissance de ce groupe de travail, le nom propre MANET est souvent utilisé comme nom commun pour désigner un réseau ad hoc, spécialement dans les pays anglophones.

1.3.2 Scénario de transmission d'un message dans un MANET

Dans la figure 1.2, nous avons donné un petit scénario de transmission d'un message où la station mobile S1 veut émettre un message à la station S5(la station s5 est hors de la porté de la station s1). La station S1 va exploiter la mobilité des autres nœuds et les utiliser comme des nœuds relais afin de livrer son message,et admettons que nous allons adopter la stratégie de relayage de message suivante : dès que deux nœuds rentrent en connexion, ils s'échangent impérativement entre eux les nouveaux messages qui ne figure pas dans leurs mémoires locaux.
A l'instant t0, la station mobile S1 transmet son message à la station S2, à l'instant t2, la station mobile S2 va relayer le message à la station mobile S3, et enfin en t3, la station mobile S3 va délivrer le message à la station mobile destinatrice S5.

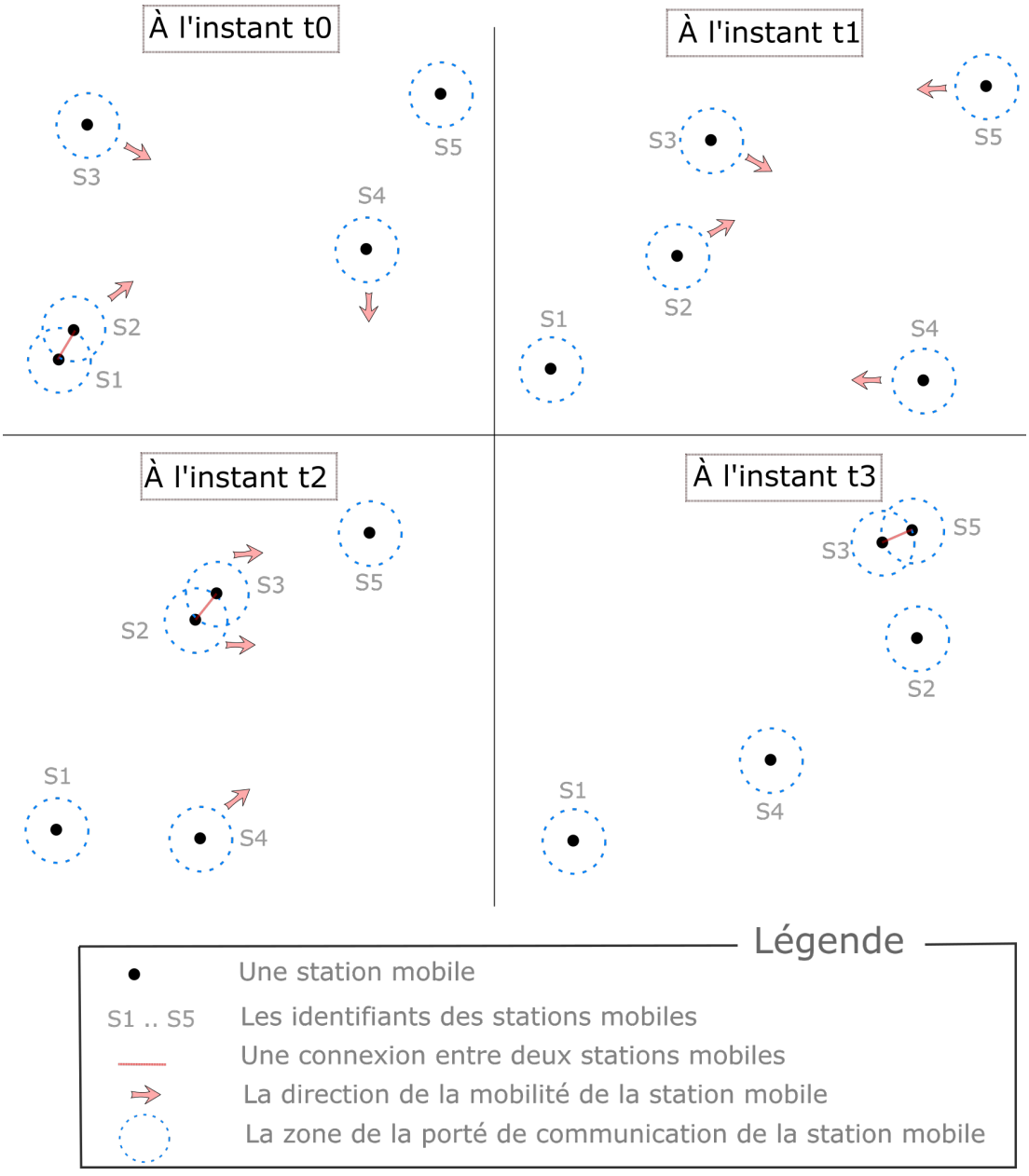

FIGURE 1.2 – Scénario de transmission d'un message dans un MANET.

1.4 Qu'est ce qu'un VANET ?

1.4.1 Définition

Un réseau véhiculaire VANET (Vehicular Ad hoc NETwork) est un sous type de réseau mobile MANET où les nœuds du réseau sont des véhicules. La recherche sur les réseaux

véhiculaires VANETs fait partie du grand domaine de recherche dans les systèmes de transport intelligents ITS (Intelligent Transportation Systems) qui affectent d'une manière directe l'amélioration de la sécurité routière en milieu urbain et en autoroute.

En plus de l'organisation du trafic routier, un VANET propose des technologies de communication qui facilitent l'accès aux services routiers. C'est une sorte de réseau mobile utilisé pour la communication entre les véhicules. Il se compose de trois éléments principaux : Les nœuds (les véhicules), l'unité de bord de route RSU (Road Side Unit) et l'autorité centrale CA(Central Authority).

Un nœud représente un véhicule équipé de l'unité embarqué OBU (On Board Unit) et de l'unité d'application (UA). L'OBU est utilisé pour calculer et afficher toutes les informations nécessaires à la localisation et pour partager et échanger des données. Le RSU est composé d'un ensemble de dispositifs installés sur le bord de la route, c'est un intermédiaire entre les véhicules et l'infrastructure. Enfin, La CA joue le rôle d'un serveur qui assure la sécurité des différents services tels que la délivrance des certificats, des clés de communication et le stockage de certaines données. [25].

1.4.2 Scénario de transmission d'un message dans un VANET

Dans la figure 1.3, nous avons donné un petit scénario de transmission d'un message dans un réseau VANET où le véhicule V1 veut émettre un message au véhicule V4(qui est hors de sa porté). Le véhicule V1 va exploiter la mobilité des autres véhicules en les utilisant comme des véhicules relayeurs afin de livrer le message à destination,et admettons que nous allons adopter la stratégie de relayage de message suivante : dès que deux véhicules rentrent en connexion, ils s'échangent impérativement entre eux les nouveaux messages qui ne figure pas dans leurs mémoires locaux.
A l'instant t0, le véhicule V1 transmet son message au véhicule V2, à l'instant t1 le véhicule V2 entre en connexion avec le véhicule V3 et il lui transmet le message de V1. A l'instant t2 le véhicule V2 va relayer le message au véhicule destinataire V4.

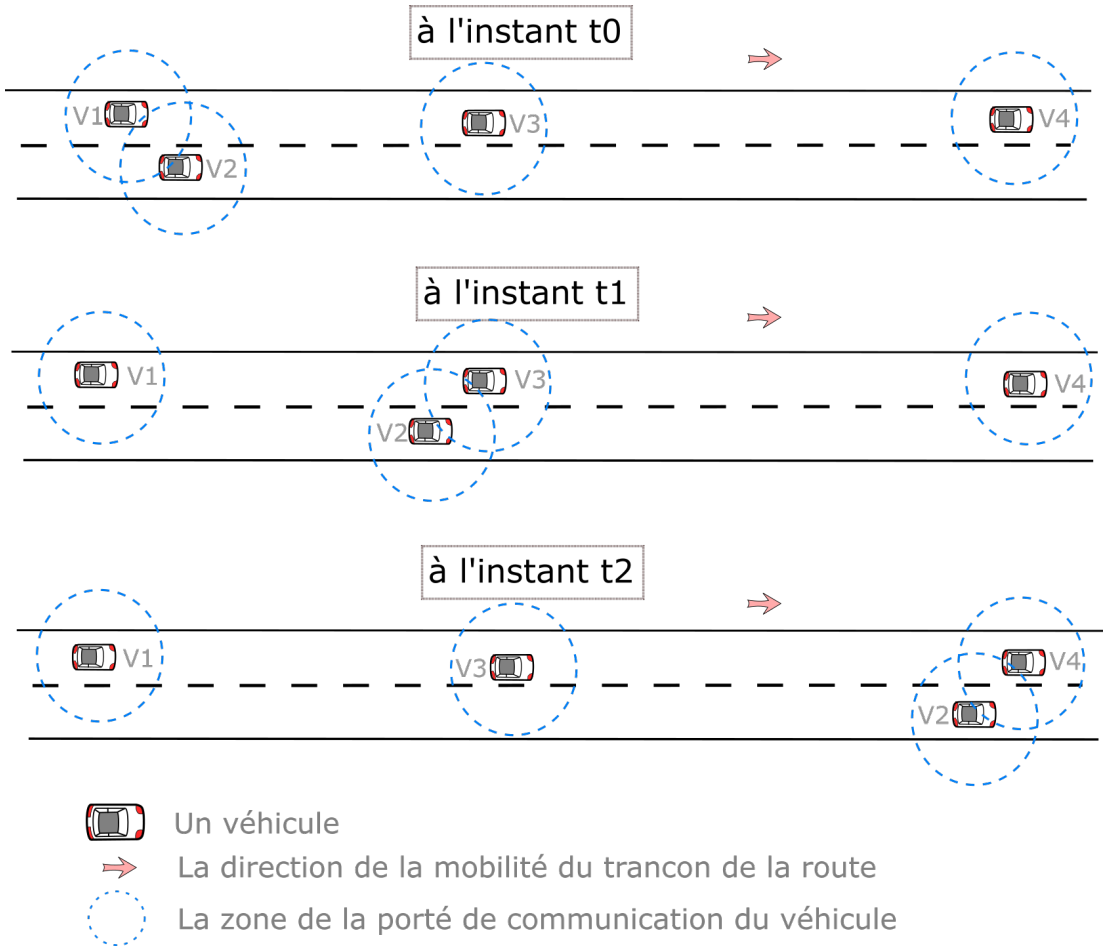

FIGURE 1.3 – Scénario de transmission d'un message dans un VANET.

1.4.3 Les modes de communication existant dans un réseau VANET

a) La communication dans le véhicule lui-même.

b) La communication ad hoc par les deux sous-modes : Véhicule à Véhicule V2V (Vehicle to Vehicle) et Véhicule à Infrastructure V2I (Vehicle to Infrastructure).

Un réseau VANET gère trois types de messages : Beacon, alerte et service. Le message Beacon est utilisé pour l'identification, la découverte et le contrôle des voisins. Les messages d'alerte sont utilisés pour la gestion du trafic routier, et les messages de service sont destinés aux sites de localisation et de découverte [25].

1.4.4 Les applications des réseaux VANETs

Les applications des VANETs sont classées en 4 grandes classes [21] :

a) Applications orientées vers la sécurité routière : trafic en temps réel, signalement d'accidents, signalement de dangers sur la route.

b) Applications orientées vers les commodités et le confort du conducteur : la planification d'itinéraire plus court et moins encombrant, connexion et échange d'informations avec d'autres chauffeurs, payement des frais d'accès à certaines autoroutes.

c) Applications commerciales : accès au réseau Internet, diagnostic à distance du véhicule, télé-chargement de carte numérique, publicité destinée aux prestataires de services, qui souhaitent attirer des clients dans ses magasins.

d) Applications productives : recueil de données de transport en temps réel pertinentes pour la recherche sur l'environnement, l'économie de carburant lorsqu'on utilise des applications via le réseau (payement d'accès aux autoroutes payantes), ainsi le carburant est économisé d'environ 3 pour cent, ce qui est consommé lorsqu'un véhicule attend en moyenne normalement pendant 2 à 5 minutes pour faire un payement.

1.5 Internet des véhicules (IoV)

1.5.1 Qu'est-ce que l'IoV ?

L'évolution de l'IoT a permis aux réseaux ad hoc véhiculaires classique (VANET) d'évoluer vers le paradigme de l'Internet des véhicules ou IoV (Internet of Vehicles), en d'autre termes on peut considérer qu'un IoV est un VANET connecté à internet.

1.5.2 Types de communication dans l'IoV

L'IoV est formé essentiellement par des unités installées dans les véhicules (On Board Unit ou OBU) et des bornes installées sur les routes (Road Side Unit ou RSU). Nous appelons les types de communication possibles dans les réseaux véhiculaires "V2X", où le X fait référence à un véhicule (V2V), une infrastructure (V2I), une personne (V2P) ou tout autre objet connecté (V2O). Grâce à ses caractéristiques, l'IoV est considéré comme une partie de l'IoT où les objets sont des véhicules qui permettent de créer une multitude de services dédiés à l'écosystème du transport intelligent. La communication V2V est la composante principale de l'IoV, car la grande quantité de données engendrée doit être transmise d'une source à une

ou plusieurs autres destinations. Pour cela les services offerts par l'IoV s'appuient beaucoup sur les véhicules en question, comme émetteur, relais et récepteur. Les véhicules s'occupent à la fois de leurs propres communications mais servent aussi de relais d'informations pour les communications entre d'autres véhicules [21], comme on peut aussi trouver une communication I2I (Infrastructure to Infrastructure). Nous avons résumé l'ensemble de ces types de communication dans la figure 1.4.

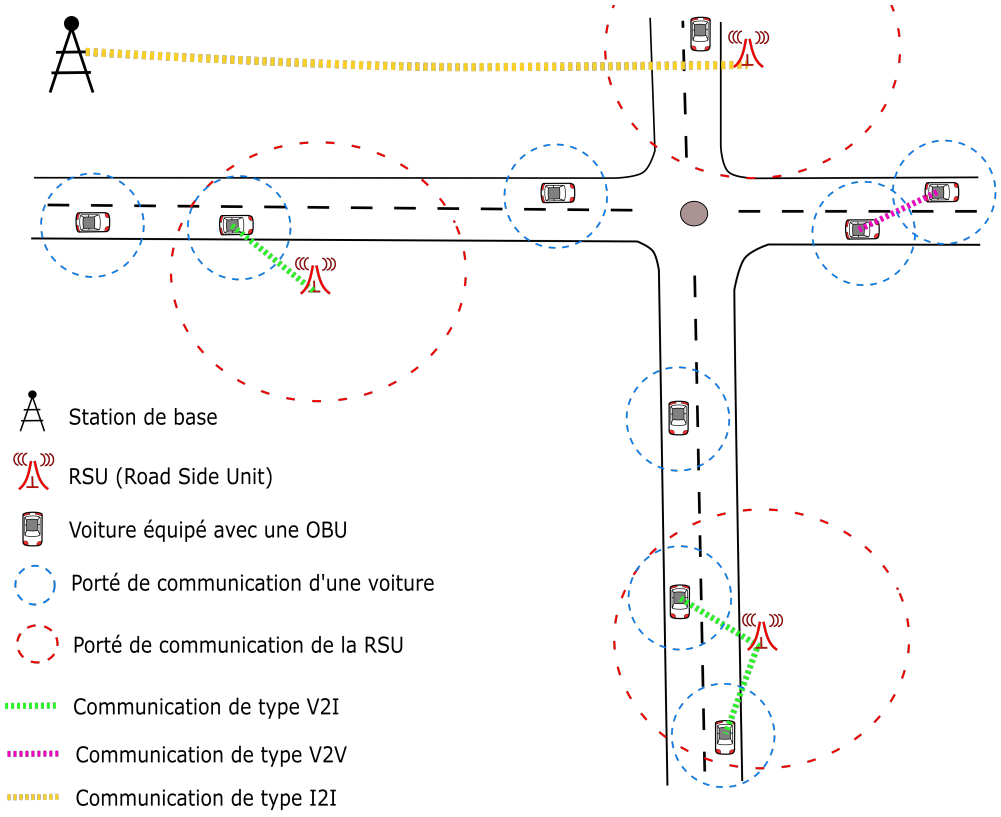

FIGURE 1.4 – Types de communication dans l'IoV.

1.5.3 La norme de communication IEEE 802.11p [24]

La norme IEEE 802.11p dérivée de la norme IEEE 802.11 est un amendement par le groupe de travail DSRC (Dedicated Short Range Communications) de l'IEEE pour l'accès sans fil dans le système de transport intelligent sera présentée dans cette section.

En effet, le processus de normalisation IEEE 802.11p Wireless Access for the Vehicular Environment (WAVE), qui est une sorte de Wi-Fi mobile, découle de l'attribution de la bande de fréquences DSRC et de l'effort de définition technologique.

Cette technologie permet d'éviter les collisions dans lesquelles les voitures sont mutuellement attentives aux conditions changeantes et peuvent améliorer considérablement la sécurité routière.

De plus, cette technologie supporte plus de 200km/h de vitesse avec une portée de transmission allant jusqu'à 1000m. Le spectre DSRC est structuré en sept canaux d'une largeur de 10 MHz. Le canal 178 est le canal de contrôle (CCH), qui est limité aux communications de sécurité. Les deux canaux situés aux extrémités de la bande de fréquences sont réservés à des usages particuliers. Les autres sont des voies de service (PPB) disponibles à la fois pour une utilisation sécuritaire et non sécuritaire.

De plus, le DSRC est considéré comme le meilleur candidat pour les communications véhiculaires dans de nombreux systèmes coopératifs de STI en raison de ses avantages par rapport à d'autres technologies potentielles, telles que Bluetooth, infrarouge, Zigbee et les communications mobiles 3G.

1.5.4 Définition d'un OBU

L'unité embarquée OBU (On Board Unit) est un petit dispositif véhiculaire doté de capteurs qui mesures plusieurs métriques (distances parcourus, positions géographique, etc). L'appareil est fixé en générale sur le pare-brise et il est en contact avec un satellite de navigation ou bien avec des RSUs dans le cas des réseaux IoVs. Dans la Figure 1.5 nous avons illustré un exemple d'un dispositif OBU.

FIGURE 1.5 – Image d'une OBU [2].

1.5.5 Définition d'une RSU

L'unité routière RSU (Road Side Unit) est un émetteur-récepteur monté le long d'une route ou d'un passage piéton. Une RSU peut également être montée sur un véhicule ou être portée à la main, mais elle ne peut fonctionner que lorsque le véhicule ou l'unité portée à la main est à l'arrêt. En outre, une RSU opérant en vertu de cette partie est limitée à l'emplacement où elle est autorisée à opérer. Cependant, les RSU portables ou portatives sont autorisées à fonctionner lorsqu'elles n'interfèrent pas avec une opération sous licence de site. Une RSU diffuse des données vers des OBUs ou échange des données avec des OBUs dans sa zone de communication. Une RSU fournit également des attributions de canaux et des instructions de fonctionnement aux OBUs dans sa zone de communication, si nécessaire [4]. Dans la Figure 1.6 nous avons illustré un exemple d'un dispositif RSU.

FIGURE 1.6 – Image d'une RSU [15].

1.5.6 La relation de l'IoV avec le Cloud

La nouvelle ère de l'IoT entraîne l'évolution du VANET conventionnel vers la nouvelle vision IoV. Ainsi, IoV intègre : VANET, IoT et le Cloud Computing mobile. Le paradigme Cloud fournit des installations de stockage et d'analyse de données, ainsi que la fourniture de tous types de services, qu'ils soient liés aux logiciels ou aux matériels virtualisés. Dans la communication Vehicle-to-Cloud (V2C), n'importe quel véhicule dans un VANET peut communiquer directement avec l'environnement Cloud et fournir des services basés sur le cloud.[40].

1.5.7 Objectifs de l'IoV

L'un des principaux objectifs de l'IoV est de permettre au véhicule de communiquer en temps réel avec le conducteur, les piétons, les autres véhicules et les infrastructures routières.

1.5.8 Les difficultés rencontrées par l'IoV

L'IoV est un système de réseau très complexe principalement pour les raisons suivantes : [40] :

1) **Le flux de noeuds :** le nombre de nœuds dans le réseau peut augmenter et diminuer en temps réel.
2) **La mobilité :** les nœuds sont mobiles par nature et ont des vitesses variables et des directions changeantes en temps réel, ainsi les distances entre les nœuds changent en temps réel aussi.
3) **La confidentialité :** comme dans tout réseau, il existe également des problèmes liés à la confiance, à la vie privée et à la confidentialité des données.
4) **L'aspect sécurité :** les cyber-menaces qui guette le réseau d'une manières constante.

1.5.9 IoV et DTN

La conception de systèmes distribués pour le stockage de données, qui permettent un accès omniprésent aux données, est essentielle dans la conception des systèmes VANETs contemporains (IoV). Les réseaux de véhicules doivent avoir accès à un magasin de données central lorsqu'ils traversent l'environnement pour prendre des décisions intelligentes en cas de besoin. Bien qu'il existe de nombreuses exigences pour la conception d'un tel système, un système distribué tolérant aux pannes est essentiel pour les environnements à forte intensité d'informations. Ainsi, il est important de concevoir le système distribué basé sur des réseaux tolérants aux perturbations (tel qu'un réseau tolérant les délais - DTN) intégrés dans un réseau maillé soigneusement conçu où la perturbation de ce dernier ne se traduit pas par une perturbation globale du réseau [40].

1.6 Le paradigme des DTNs

1.6.1 Définition

Les réseaux tolérants aux délais (RDTs) souvent écrit (DTNs) (Delay Tolerant Networks) sont des types de réseaux conçus pour supporter des latences de plusieurs minutes. Ils sont utiles pour fonctionner sur de très longues distances comme pour des télécommunications spatiales pour lesquels la latence atteint parfois des heures, voire des jours. De tels réseaux

peuvent également s'avérer utiles lorsque l'interférence est extrême ou bien les ressources sont surchargées.

1.6.2 Histoire chronologique des DTNs

L'histoire du paradigme DTN, qui est la mise en réseau à tolérance de retard, remonte aux années 1970 lorsque l'informatique a connu un succès fulgurant à travers le monde, les chercheurs ont commencé à développer une technique de routage pour les ordinateurs non fixes. Durant les années 1980, le domaine du routage ad hoc n'était pas sollicité, mais au début des années 1990 avec l'utilisation massive des protocoles sans fil, cela a donné une bouffée d'oxygène à ce domaine en le reboostant grâce à des chercheurs qui se sont intéressés là-dessus à travers les réseaux mobile ad hoc.

Le terme DTN était utilisé pour la première fois en 2002 par Kevin Fall [7] en s'inspirant des bases de l'IPN (Inter Planetary Network), des principes qui font face à la corruption de paquets et aux retards de messages dans les communications spéciales, et en 2003 une première publication sur les DTNs a été faite et à partir de là les chercheurs ont commencé à s'intéresser à ce paradigme en essayant d'étudier et d'améliorer l'aspect sécurité et fiabilité de ce dernier.

1.6.3 Utilité et principes des DTNs

Les DTNs permettent le transfert de données lorsque les nœuds mobiles ne sont connectés que par intermittence. Les applications des DTNs comprennent les réseaux de capteurs pour la surveillance écologique, les réseaux de capteurs océaniques, les réseaux de véhicules, etc [23].

Les principes de conception de l'architecture DTN peuvent être résumés comme suit :

Premièrement : Des messages de longueur variable existeront en tant qu'abstraction de communication pour faciliter la capacité du réseau à prendre des décisions d'ordonnancement ou de sélection de chemin.

Deuxièmement : Le stockage au sein du réseau est utilisé pour prendre en charge l'opération de stockage et de retransmission sur plusieurs chemins et sur des échelles de temps potentiellement longues.

1.6.4 DTN et routage

Pour chaque protocole fonctionnant dans un DTN, il y a toujours de nombreuses caractéristiques qui doivent être prises en compte :

1) Les informations sur les contacts futurs (s'ils sont facilement disponible) : Il est possible de prédire l'avenir en termes d'opportunités (comme nous allons le voir avec le protocole Prophet dans le chapitre 2), lorsque les contacts sont disponibles et éventuellement leur durée. Ces contacts peuvent être prévisibles ou programmés, ou encore intermittents ou opportunistes.

2) La mobilité : il y a des cas où la mobilité peut être exploitée, en occurrence on exploite les nœuds mobiles. Il existe trois cas de mobilité dans un réseau donné :

a) seul une partie des nœuds sont mobiles. Ces derniers, sont exploités pour leur mobilité (on les appelle parfois mules de données). Ils assurent une communication dite transitive entre des nœuds non voisins.

b) aucune entité n'est mobile : dans ce genre de cas, les nœuds apparaissent et disparaissent en fonction uniquement de la qualité du canal de communication entre ces derniers.

c) il est possible que la grande majorité, sinon la totalité des nœuds du réseau soient mobiles. Dans ce cas, un protocole de routage est plus que nécessaire afin de garantir une transition correct de donné dans le réseau. Un des exemples qui illustre le plus ce genre de réseau est le réseau véhiculaire où des véhicules agissent comme entités communicantes.

3) La disponibilité des ressources du réseau : la durée de vie de la batterie, l'espace de stockage, la vitesse de transmission, etc, des nœuds mobiles (par exemple des téléphones cellulaires), sont des ressources limitées et précieuses qui nécessitent une gestion optimale et intelligente afin de garantir une disponibilité continue qui assure le bon déroulement de transition de l'information dans le réseau. D'autres part, il existe des OBUs qu'on peut trouver à bord des automobiles, des bus, des camions, et qui peuvent avoir une capacité illimité des ressources cité ci-dessus. C'est pour cela que des protocoles de routage sont mis en place afin de fixé la manière dans lequel l'information doit transiter et être stocker pour éviter le sur-chargement des ressources.

1.7 Conclusion

Dans ce premier chapitre, nous avons donné quelques définitions et terminologies réseau que nous allons utiliser tout au long de ce mémoire à savoir : l'IoT, VANET, l'IoV, DTN, etc. Nous avons présenté quelques domaines d'application des réseaux véhiculaires ainsi que les différents types de communications qu'il est possible de trouver dans ce genre de réseau. Ensuite, nous avons détaillé le concept des réseaux véhiculaires connectés à internet (IoV) ainsi que le lien entre eux et les réseaux tolérants les délais (DTNs).

Dans le prochain chapitre, nous allons présenter un état de l'art sur les protocoles de routages opportunistes jugés plus appropriés pour notre réseau, qui est de type DTN, et nous choisirons celui qui convient le mieux à notre application de signalement d'infractions routières.

Chapitre 2

État de l'art sur les protocoles de routage opportunistes

2.1 Introduction

Lors de la mise en œuvre d'un réseau à tolérance de délai (DTN) nous faisons face à des difficultés de connectivité réseau continue souvent causées par des nœuds qui sont en constante mobilité (réseau Manet, Vanet, etc) ou dû au manque de nœuds qui ne garantit pas une constante connectivité (endroits désertiques, milieux hostiles, etc), c'est pour cela que le routage opportuniste est plus que nécessaire dans ce genre d'architecture car il assure des résultats satisfaisants malgré les difficultés citées ci-dessus, d'où notre intérêt pour ce type de routage, qui nous permettra, d'une part, de contrôler la circulation du flux émit par notre application de signalement d'infractions routières dans notre réseau véhiculaire, et d'autre part, de nous garantir des résultats prometteurs.

Dans ce chapitre, nous allons donner une classification des protocoles de routage dans les réseaux véhiculaires : tolérants les délais (DTNs), non tolérants les délais (Non-DTNs) et hybrides. Nous allons, par la suite, nous focaliser sur l'étude des protocoles de routage dédiés aux réseaux tolérants les délais, c'est à dire aux protocoles de routage opportunistes, et nous choisirons celui que nous jugerons le plus adapté à notre application de signalement d'infractions routières.

2.2 Classification des protocoles de routages dans les réseaux véhiculaires

Certaines recherches [26] [22] ont classé les protocoles de routage VANETs selon l'application où ils sont les plus adaptés : DTNs, Non-DTNs ou hybride :

2.2.1 Les protocoles de routage DTNs

Les protocoles de routages dédies aux réseaux tolérants aux délais (DTNs), appelés aussi protocoles de routage opportunistes, ne nécessitent pas de connexion stable entre le noeud émetteur et le noeud destinataire pour transmettre un message d'une part, d'autre part ils utilisent des stratégies opportunistes tels que carry-and-forward (lorsqu'un nœud ne peut pas contacter d'autres nœuds, il stocke le paquet, et le transfert est effectué sur la base de certaines métriques des nœuds voisins) [19] et cela pour surmonter les déconnexions fréquentes du réseau (les noeuds intermédiaires, qui relayent les messages, stockent des messages et attendent un saut approprié). Par conséquent, les protocoles de cette classe sont bien adaptés aux déconnexions et aux perturbations du réseau. Parmi ces protocoles, on trouve : VADD [42], EPEDIMIC [39], Prophet [33].

2.2.2 Les protocoles de routage Non-DTNs

Les protocoles de cette classe ne tolèrent pas les retards de livraison de données car il sont souvent sous la contrainte du temps. Ils utilisent généralement des acquittements de réception de messages afin de s'assurer que les messages sont bien reçu par l'expéditeur. Parmi les protocoles de cette classe : GPSR [31], LOUVRE [32].

2.2.3 Les protocoles de routage hybrides

Les types hybrides combinent les principes des DTNs et des Non-DTNs. Lorsque le réseau est dense, une stratégie de routage est utilisée pour acheminer les paquets de données et lorsqu'une déconnexion se produit, la mobilité du véhicule est exploitée (en transportant le paquet jusqu'à ce qu'un voisin éligible apparaisse ou qu'il atteigne lui-même la destination) afin d'atténuer l'impact de la connectivité intermittente. Parmi ces protocoles, nous citons : ROAMER [35], GéoDTN+NAV [29].

Dans ce qui suit, nous allons nous focaliser sur l'étude de quelques protocoles de routage de la classe DTN (c'est à dire les protocoles de routage opportunistes susceptible d'être conforme à notre application) car cette dernière est jugée plus approprié pour l'application de signalement d'infractions routières que nous proposons.

2.3 Étude de quelques protocoles de routage opportunistes

Dans cette section, nous commençons par présenter deux protocoles de routage opportunistes de référence pour les réseaux ad hoc, il s'agit du protocole Epidemic [39] et du protocole Prophet [33]. Par la suite, nous présenterons trois autres protocoles de routages opportunistes récents pour les VANETs afin d'avoir une idée sur les critères et les métriques sur lesquels ces protocoles se basent.

2.3.1 Le protocole de routage Epidemic

2.3.1.1 Présentation du protocole

C'est l'un des premiers protocoles de routages [39] proposé pour les réseaux ad hocs, il occupe une place très importante dans la littérature car il a été cité plus de 3000 fois par d'autre articles scientifiques. Son nom épidémique est inspiré du fait que les messages dans le réseau se propagent comme une épidémie dès que les nœuds entrent en connexion.

L'objectif du routage Epidemic est de minimiser la latence de livraison des messages, tout en minimisant les ressources système globales consommées lors de la livraison des messages (bande passante, mémoire, énergie). Il suppose que la mémoire d'un nœud est suffisante pour stocker entre 10 à 25 pour cent des messages provenant d'un scénario donné.

2.3.1.2 Fonctionnement du routage Epidemic

Chaque hôte possède un buffer (structure de stockage, tampon) où il stocke les messages qu'il a émis et les messages au compte d'autres utilisateurs. Un vecteur appelé SV (Summary Vector) index les Ids (identifiants) des messages de cette liste. Lorsque les deux hôtes se rencontrent ils échangent leurs SV afin de déterminer quels messages stockés à distance n'ont pas été vus par l'hôte local. À son tour, chaque hôte demande alors des copies des messages qu'il n'a pas encore vus. L'hôte récepteur conserve sa pleine autonomie dans la décision d'accepter ou non un message.

Le champ de nombre de sauts associé au message détermine le nombre maximum d'échanges épidémiques auxquels un message particulier est soumis. Chaque hôte définit une taille de mémoire tampon maximale qu'il est prêt à allouer pour la distribution des messages épidémiques. L'utilisation des ressources de ce schéma est réglée par le nombre de sauts

défini dans les messages et l'espace tampon disponible au niveau des nœuds. Si ceux-ci sont suffisamment grands.

2.3.1.3 Les problèmes rencontrés par ce protocole

routage sous l'incertitude : l'expéditeur ne connais pas forcément les emplacements des autres nœuds car ces derniers sont éventuellement hors de portés et en mobilités.

allocation de ressources : le système doit équilibrer entre le taux de livraison de message et la consommation de ressource car augmenter l'un fait augmenter l'autre.

performance : tenir compte des métriques suivantes : latence de livraison d'un message, consommation moyenne d'espace et de la bande passante.

fiabilité : utiliser des accusés de réception pour informer l'expéditeur et libérer les ressources associer au message dans le système.

sécurité : le message qui se déplace arbitrairement peut être exposé à des hôtes non approuvés, ceci dit les récepteurs doivent s'assurer de l'authenticité de ce dernier [39].

2.3.2 Le protocole de routage probabiliste PROPHET

2.3.2.1 Présentation du protocole

PROPHET (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [33] est une amélioration du protocole Epidemic [39], qui s'inspire de la situation suivante : les utilisateurs réels (des nœuds) ne se déplacent probablement pas au hasard, mais plutôt de manière prévisible en fonction de modèles de comportement répétitifs tels que si un nœud a visité un endroit plusieurs fois auparavant, il est probable qu'il reviendra à cet endroit.

2.3.2.2 Fonctionnement du protocole

Les observations cité ci-dessus ont induit à l'amélioration des performances du routage épidémique en faisant du routage probabiliste utilisant l'historique des rencontres et la transitivité, et en ajoutant une métrique de prévisibilité de livraison locale à chaque nœud P(a, b) qui indique la probabilité que chaque nœud "a" émet vers une destination connue "b". Ainsi chaque nœud aura une visibilité sur les nœuds qui sont les plus sollicités par rapport à la livraison d'un message vers la destination voulus. Ce protocole consomme moins d'espace mémoire par rapport à l'épidémique car, lorsque deux nœud se rencontre, un message n'est

transmis que si la prévisibilité de la destination du message est plus élevée à l'autre nœud. Lorsqu'un échange de message est décidé, il applique la même procédure de transmission que l'épidémique.

2.3.3 Le protocole de routage guidé par la trajectoire TDOR

2.3.3.1 Présentation du protocole

TDOR (Trajectory-Driven Opportunistic Routing protocol) [27] est un protocole de routage qui assure de bons résultats en termes de taux de livraison de paquets et de surcharge du réseau. Il choisit les nœuds intermédiaires qui vont relayer les messages en fonction de la proximité de la trajectoire de ces derniers.

2.3.3.2 Fonctionnement du protocole

Dans ce qui suit, nous allons décrire les phases suivies et appliquées par le protocole :

1) **Phase de calcul de la trajectoire :** Uniquement déclenchée une fois que le nœud source envoie un message. Il calcule une trajectoire et embarque les informations de la trajectoire dans le message. La politique du chemin le plus court est appliquée à un nœud dit L (éléments de liaison routière pour le calcul de la trajectoire).
2) **Phase de relais de messages :** Elle est exécutée par le nœud porteur du message (ou d'une copie du message) lorsqu'il rencontre d'autres relais possibles. Le défit est de décider si oui ou non un nœud intermédiaire serait préférable d'aide à faire passer le message. Il définit les nœuds u et v comme le message transporteur et nœud rencontré (un nœud relais possible), tandis que le nœud d est la destination du message. Le but est de trouver les nœuds associés à la trajectoire. À chaque rencontre entre les nœuds u et v, ils calculeront leur association de trajectoire liée à L. Lorsque le nœud u commence à relayer le message M. La livraison du message est découpée en trois cas suivants :

   Aucune association : cela se produit lorsque les deux nœuds u et v ne sont pas associés à L, la politique appliquée est la file d'attente à faible priorité LPQ (Low Priority Queue).

   Association unique : ne se produit que lorsque le nœud v est associé à L, alors que le nœud u ne l'est pas, la politique appliquée est la file d'attente de priorité moyenne MPQ (Medium Priority Queue).

Double association : se produit lorsque les deux nœuds u et v sont associés à L, la politique appliquée est la file d'attente à haute priorité HPQ (High Priority Queue).

3) Phase de gestion des messages : Tous les messages ne peuvent pas être transmis avec succès. Il est donc pratique de classer les messages afin d'assurer celui qui a plus de potentiel de livraison à transmettre. Les messages sont prioritairement classés en séquence. Ensuite, en suivant trois cas d'association (pas d'association, association simple et association double), les messages mis en file d'attente sont transmis. Les messages sont également hiérarchisés se référant aux trois cas d'association.

2.3.4 Le protocole de routage basé sur la mobilité MOP

2.3.4.1 Présentation du protocole

MOP (Mobility-aware Opportunistic routing Protocol) [28] est un protocole de routage opportuniste qui traite de la mobilité individuelle des véhicules. Il est utilisé dans les réseaux où le(s) nœud(s) de destination sont mobiles, et vise à réduire la surcharge du réseau, caractérisé par le rayon de giration et la métrique d'entropie pour vérifier la variabilité de la mobilité des nœuds du réseaux (des véhicules), et il utilise le routage unicast (dans lequel une seule source envoie un message à une seule destination) ainsi que le surcoût, tout en conservant le taux de livraison et une latence meilleure dans le réseau.

Le rayon de giration : consiste à quantifier la mobilité du véhicule par rapport à son centre de masse de mouvement et permet de déterminer si un nœud parcourt une longue distance ou pas.

Mobilité d'entropie : est utilisée pour quantifier la dynamique spatiale d'un véhicule.
Ces deux métriques sont utilisées pour décider si un transfert de message doit se faire ou pas.

2.3.4.2 Fonctionnement du protocole

Quand la portée de communication des véhicules se chevauchent, ils s'identifient et vérifient s'ils contiennent un message dans le tampon indiquant que la destination est le véhicule en contact ; si c'est le cas, ils échangent le message. Une autre situation pertinente est la réplication des messages contenus dans le tampon, mais le véhicule en contact n'est pas la destination des messages, ils peuvent échanger les informations suivantes : un résumé des informations utiles sous forme de vecteur appelé (SV) avec des métadonnées sur les messages qu'ils contiennent dans leurs buffers et la valeur de leur métrique de mobilité (MM). Les

métriques de mobilité utilisées peuvent être le rayon de giration ou l'entropie de mobilité. Si le véhicule i identifie que MM j est supérieur à MM i, il envoie alors au véhicule j les messages contenus dans son buffer que le véhicule j n'a pas (il est représenté par MS). Sinon, ils ne reçoivent que des messages du véhicule j.

Ce protocole autorise plusieurs copies du message émis par le véhicule source pour être transmis sur le réseau pour augmenter les chances de le livrer au véhicule de destination.

Il existe deux versions du protocole MOP :
MOP-RG basé sur le rayon de giration.
MOP-Entropy basé sur entropie de mobilité.

2.3.5 Le protocole de routage multicritères dans les IoVs

2.3.5.1 Présentation du protocole

Les auteurs de [21] ont proposé un protocole de routage géographique qui se base sur l'estimation des durées de contact entre les véhicules, les charges de données à transmettre et en fin les logs d'anomalies de communication.
L'objectif est d'assurer la disponibilité, la fiabilité et la robustesse des communications inter-véhiculaires en prenant en compte ces trois différents critères dans un algorithme d'acheminement des paquets de données. Ce protocole a été proposé pour détecter et contourner les zones de déconnexions.

2.3.5.2 Fonctionnement du protocole

Ce protocole multicritères procède en 3 phases :

La première phase : découverte des voisins
Un message de type becon (x,y,v,dir,c,id) est envoyé par un véhicule à ses voisins où :
x,y : correspondent au données géographiques du véhicule.
v : la vitesse du véhicule.
dir : la direction du véhicule.
c : espace mémoire vide du véhicule.
id : l'identifiant du véhicule (son adresse ip).

La deuxième phase : mise à jour des logs d'anomalie
Les logs d'anomalie (fichiers locaux qui contiennent une liste des évènements de déconnexions avec les détails et éventuellement leurs causes pour un véhicule donné). Dans Cette phase on va détecter les zones noirs, et cela va aider les véhicule lors du relayage des messages.

La troisième phase : déterminer le routage géographique
Habituellement les métriques qui déterminent le routage sont : la distance, la densité de communication, le débit, l'état des liens, l'appartenance à un cluster, la rupture prématurée des communications. Dans cet article, les auteurs proposent deux métriques supplémentaires : le calcul de la durée de contact, et le taux de référencement dans le log d'anomalie de communication qui seront définis à partir des données des deux phases précédentes.

2.4 Critique des protocoles Epidemic et Prophet

L'un des inconvénients du protocole Epidemic est la surcharge des routeurs ainsi qu'un très grand nombre de saut de messages dans le réseau. Par contre, le nombre de livraison de message accompli est remarquable (à condition que les nœuds aient suffisamment d'espace mémoire).

Par rapport au protocole probabiliste Prophet, il y a moins de surcharge des nœuds par rapport à Epidemic. Cependant, le temps et le nombre de messages transmis sont moins bons comparativement à Epidemic.

2.5 Motivations et choix des protocoles de routage pour notre simulation

Après une longue réflexion, nous avons décidé de tester les deux protocoles de routage les plus emblématiques et les plus cités dans la littérature qui sont Epidemic et Prophet (voir la simulation dans le chapitre 3) pour les raisons suivantes :
pour Epidemic, dans le passé les chercheurs ont essayé de trouver des alternatives à ce dernier car malgré ces performances remarquables il rencontre des problèmes de surcharge du réseau, en d'autre terme il possède un problème d'espace de stockage au niveau des nœuds, mais actuellement, nous n'avons plus ce genre de limitation de mémoire, car avec l'évolution des OBUs ces dernières sont équipés d'une capacité de stockage vraiment importante, mais aussi nous pouvons stocker dans le Cloud une fois la mémoire de l'OBU est surchargé par exemple,

c'est pour cela que ce protocole nous semble très intéressent de renouer avec lui dans notre ère et de faire une simulation avec.
Pour le protocole Prophet, durant ce travail nous voulions un protocole de routage qui prend en considération l'historique des rencontres des véhicules notamment d'exploiter la probabilité de fréquentation de certain endroit (placement des RSUs en occurrence les commissariat par exemple) par certain véhicules afin de favoriser ces derniers dans le relayage des messages d'infractions vers ses RSUs, et d'autre part le protocole Prophet nous permet de réalisé ça mais aussi ils nous permet de mettre au point des futures probabilités de rencontre entre les nœuds et en plus de ça nous avons pu constater qu'il a assuré sa place en terme de performance dans la littérature, c'est pour ça ce protocole aussi nous semble être un bon candidat pour la simulation en répondant aux critères que nous avons fixé au dessu

2.6 Conclusion

Durant ce chapitre, nous avons présenté une classification des protocoles de routage dans les réseaux véhiculaires (protocoles DTNs, Non-DTNs, hybrides). Ensuite nous nous sommes focalisés sur la classe des protocoles DTNs appelés aussi protocoles de routage opportunistes car nous avons jugé qui'ils sont les plus adaptés à notre application.

Le prochain chapitre sera consacré à la présentation de notre application ainsi qu'à la discussion des résultats obtenu après simulation avec les deux protocoles que nous avons choisi comme candidat dans ce deuxième chapitre afin d'élire celui qui est le plus performant et l'améliorer.

Chapitre 3

Simulations et résultats

3.1 Introduction

Dans le présent chapitre, nous commençons par une présentation de l'application que nous avons proposé et on enchaîne par une brève description du simulateur avec lequel nous avons choisi de travailler et nous évoquerons les raisons qui ont motivé notre choix. Nous décrirons, par la suite, les étapes de l'implémentation des itinéraires et des routes de la cartographie de la ville de Bejaıa dans le simulateur, après, nous détaillerons l'ensemble de la configuration des simulations que nous avons réalisé. Enfin, nous présenterons les résultats de la simulation de notre application et une proposition d'une amélioration du protocole de routage Prophet.

3.2 Etat de l'art sur les applications de signal d'infractions routières

Dans [18], un système visant à aider les policiers à identifier et sanctionner les contrevenants à la conduite à grande vitesse est proposé. Dans ce modèle proposé, des dispositifs RSU sont utilisés sur les routes ou les autoroutes de la ville intelligente pour détecter si un conducteur dépasse la limite de vitesse afin de lui envoyer un message d'avertissement. Si le conducteur ne réduit pas sa vitesse à la limite requise dans un délai donné, le système lui infligera automatiquement une amende. L'application proposée ne prend en compte que les violations de vitesse limite et non toutes les infractions routières comme nous le proposons dans notre apllication.
Dans [37] et [34], les auteurs ont considéré le problème de sécurité pour l'application de notification des infractions routières dans un environnement de cloud automobile. Cette question n'est pas abordée dans notre présent travail.

Les auteurs de [20] ont proposé un système intelligent pour l'établissement et l'envoi automatique de PV d'infraction routière pour les réseaux véhiculaires centrés sur le contenu. Le système prend en compte la problématique des zones non surveillées en milieu urbain ou autoroutier. Si un véhicule civil détecte une infraction au code de la route dans cette zone, il stocke cette information et lorsqu'il passe dans une zone surveillée, il reçoit un message d'intérêt du véhicule de police auquel il répond en lui envoyant les informations sur l'infraction au code de la route. Ainsi, le véhicule de police générera un PV d'infraction pour le contrevenant et l'enverra au véhicule correspondant.

3.3 Généralités et spécifications de notre application de signalement d'infractions routières

Pour aider les services de la police à surveiller toutes les infractions routières et infliger une amende à leurs commetteurs, nous proposons un système de signalement d'infractions routières dans les IoVs. Cette application a pour finalité le signalement de toutes les infractions routières qu'un conducteur pourrait commettre (excès de vitesses, non-respect de la signalisation routière, conduite en état d'ivresse, etc.) aux services de police de la ville, Cette application peut aider à réduire les accidents de la route en réduisant les infractions qui sont les causes principales des accidents.

3.3.1 Présentation générale de l'application

Notre application de signalement d'infractions routières sera composée de deux modules :

Un module de signalement d'infraction routière : Ce module sera distribué sur l'ensemble des nœuds du réseau, il sera implémenté dans un dispositif qui sera embarqué à bord des véhicules et il sera connecté à différents capteur (GPS, détecteurs d'obstacle, compteur de vitesse, etc.), qui seront primordiales dans la détection d'infractions que le conducteur pourrait commettre. Ses unités embarquées(OBUs) dans les véhicules peuvent être aussi assistées par des unités de routes (RSUs) qui peuvent être placées devant des feux rouges, des radars, etc, pour détecter des infractions ou afin de participer à l'acheminement des messages.

Concernant le signalement des infractions au niveau des RSUs se fait de deux manière, soit en envoyant directement l'infraction aux commissariats via internet(en utilisant par la station de base), soit en participant au relayage des infractions dans le réseau véhiculaire (en cas de perturbation lié au réseau Internet) en appliquant le protocole

de routage admis. Du coté destinataires, ce module pourras être aussi implémenté sur des dispositifs embarqués dans les véhicules de police ou dans les sièges des services de police de la ville ou même au niveau des barrages de la police, où il va envoyer des requêtes à des intervalles réguliers pour récolter les messages de signalement d'infractions qui sont peut être enregistrés sur les véhicules passants. Ces requêtes seront aussi utilisées pour calculer les probabilités de délivrances pour le protocole de routage. Ainsi, chaque véhicule recevant cette requête envoie les messages de signalement d'infractions s'il en a dans sa file d'attente et met la probabilité de rencontre avec la destination de ces messages.
Remarque : C'est le module que nous allons traiter dans ce mémoire.

Un module de gestion d'infractions routière : Ce module sera sous forme d'une API hébergé dans le Cloud, qui a pour but la gestion des infractions routières qui va recevoir du premier module cité ci-dessus, à travers ce module les policiers pourrons visionner, traiter ,ajouter , supprimé ou altéré une infraction récolté, et ce module continueras à marcher en locale (hors ligne) dans les commissariats (par exemple en cas de coupures d'Internet) pour continuer de recevoir et de traité les infractions récolté à partir des véhicules qui passé à côté de ces postes de polices.
Remarque : Ce deuxième module nous allons le traiter dans de futurs travaux.

3.3.2 Alimentation de la base de données de l'application

Avant la mise en route de notre application, on doit d'abord initialiser un certain nombre d'informations qui vont aider à détecter les différentes infractions routières :

1) Traçage de toutes les routes automobiles nationales, autoroutes, urbaines, pistes cyclables, de la ville sur laquelle on va déployer l'application.

2) Indication des directions des routes.

3) Indexation des limitations de vitesse via les tronçons de toutes les routes spécifiées au-dessus.

4) Identification des coordonnées géographiques des emplacements des forces de l'ordre (commissariat, gendarmerie, brigades, casernes, barrages fixes, etc).

5) Identification des zones de stationnement interdites.

6) Implémentions de l'ensemble de la signalisation routière de la ville.

3.3.3 Spécifications de notre réseau IoV sur lequel notre application sera déployée

Tout d'abord, notre réseau véhiculaire va contenir les nœuds mobiles suivants : véhicules des civils, véhicules de police, et des nœuds fixes sous forme de RSU (unités sur les bords de la route) qu'on va éparpiller un peu partout dans notre réseau (dans les sièges des forces de l'ordre, dans les barrages de police, dans les radars, devant les feux tricolores, ...) comme indiqué dans la figure 3.1.

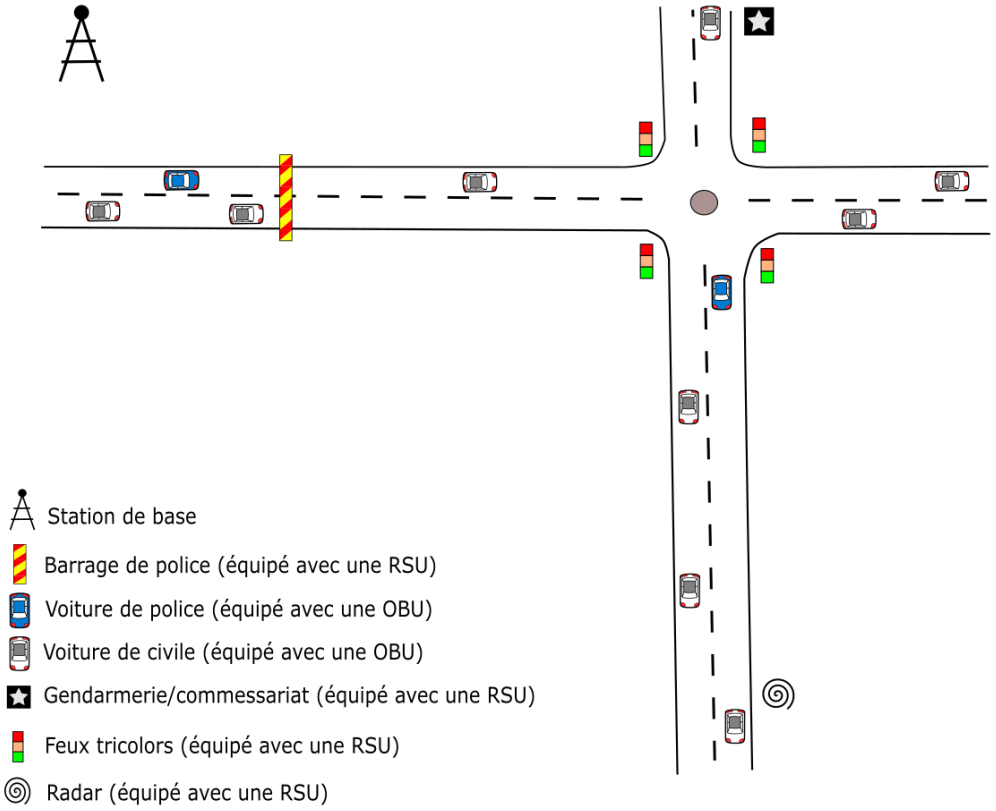

FIGURE 3.1 – Les différents éléments de notre réseau IoV

Les véhicules civils vont servir de relais afin de transiter les message d'infractions qu'ils produisent ou qu'ils reçoivent pour qu'ils les dirigent vers le destinataire final. Notre réseau IoV sera partiellement connecté à internet via les RSU afin de transférer les infractions routières recueillies vers les serveurs de notre application via internet.

3.3.4 Les infractions signalées par notre application

Dans le tableau 3.1, nous avons présenté quelques unes des infractions qui seront signalées par l'application ainsi que le(s) équipement(s) concerné(s).

Infractions	Équipements utilisés
Excès de vitesse	Compteur de vitesse, GPS, Radar
Circulation en sens interdit	GPS
Marche arrière et demi-tour en autoroute	GPS
Non-respect de la distance de sécurité entre deux véhicules	Détecteur d'obstacle, Compteur de vitesse
Stationnement interdit	GPS
Non-respect des feux tricolores	Feux tricolores

Tableau 3.1 – Les infractions signalées par l'application

3.4 Le simulateur ONE

Il existe des dizaines de simulateurs qui supportent les protocoles de routages DTNs, tels que : NS3[9], OPNET Modeler[11], NeSsim[8], ADYTON[1], etc. Dans ce projet, nous avons choisi de travailler avec le simulateur ONE.
Dans ce qui suit nous allons présenter ce dernier toute en justifiant le choix de ce dernier.

3.4.1 Présentation du simulateur

Le simulateur ONE (Opportunistic Network Environment simulator) [12], est un environnement de simulation qui est capable de générer des mouvements des nœuds d'un réseau donné tout en utilisant une large gamme de modèles de mouvements. Il permet également de gérer le routage de messages entre les nœuds du réseau en appliquant différents protocoles de routage tolérants aux délais et offre une visualisation en temps réel de la mobilité des nœuds et des messages qui transitent dans le réseau. Il donne aussi la possibilité d'importer des données dans le monde réel comme des cartes géographiques en format WKT (Well-Known Text) afin d'appliquer une simulation réelle sur ses routes, et enfin il génère au choix une large gamme de rapports détaillés qui donne un compte rendu sur différentes métriques et statistiques qui concernent le déroulement et les résultats d'un scénario de simulation (les messages livrés, les messages supprimés, la latence, le nombre de saut, le nombre de messages générés, etc). Nous avons utilisé la version 1.6.0 du simulateur ONE.

3.4.2 Choix du simulateur

Nous avons fait le choix de travailler sous le simulateur ONE, premièrement, pour son haut niveau de réalisme, car il offre la possibilité d'utiliser et de configurer n'importe quel interface de communication(bluetooth, HighSpeedInterface, etc), il nous permet de créer des groupes de noeuds et de faire une configuration très précises de ces derniers (vitesse de mouvement, modèle de mouvement, taille de la mémoire locale, les interfaces de communications utilisées et les routes sur lesquels ils se déplacent).

Deuxièmement, pour ses performances en simulations à savoir, il est capable de supporter la simulation de mouvement de milliers de noeuds, ce simulateur propose une large gamme de protocoles de routage opportuniste près à utiliser et il offre la possibilité qu'on lui intègre de nouveau protocoles de routage (à conditions que le code source de ces derniers soit écrit en JAVA).

Enfin, c'est l'un des rares simulateurs qui supportent les réseaux conçus à la base du paradigme DTN.

3.5 Implémentation de la carte géographique des routes de Béjaïa dans le simulateur

Par défaut, le simulateur ONE propose le plan de la ville d'Helsinki en Finlande, mais nous avons préféré utiliser notre propre carte qui est celle de notre ville de Béjaıa. Nous avons donc récupéré les données brutes de notre carte dans le site d'Openstreetmap [13], qui offre le code source cartographique des villes du monde entier en open source. Nous avons sélectionné le centre ville ainsi que l'ancienne ville de Béjaia, les bordures de notre sélection correspondent au coordonnées géographiques suivantes (5.1052° E, 5.0420° W, 36.7723° N, 36.7372° S), elle couvre 20 kilomètres carrés de surface. Par la suite, nous avons utilisé deux logiciels, l'Osm2Wkt [14] afin de convertir l'extension OSM (Open Street Map) en WKT(Well-Known Text) qui est le format vectoriel de nos routes supporté par le simulateur, et OpenJump [10] afin d'éditer (identifier les emplacements des commissariats) et d'ajuster notre carte représentée dans la Figure 3.2 pour l'implémenter dans le simulateur comme illustré dans la Figure 3.3.

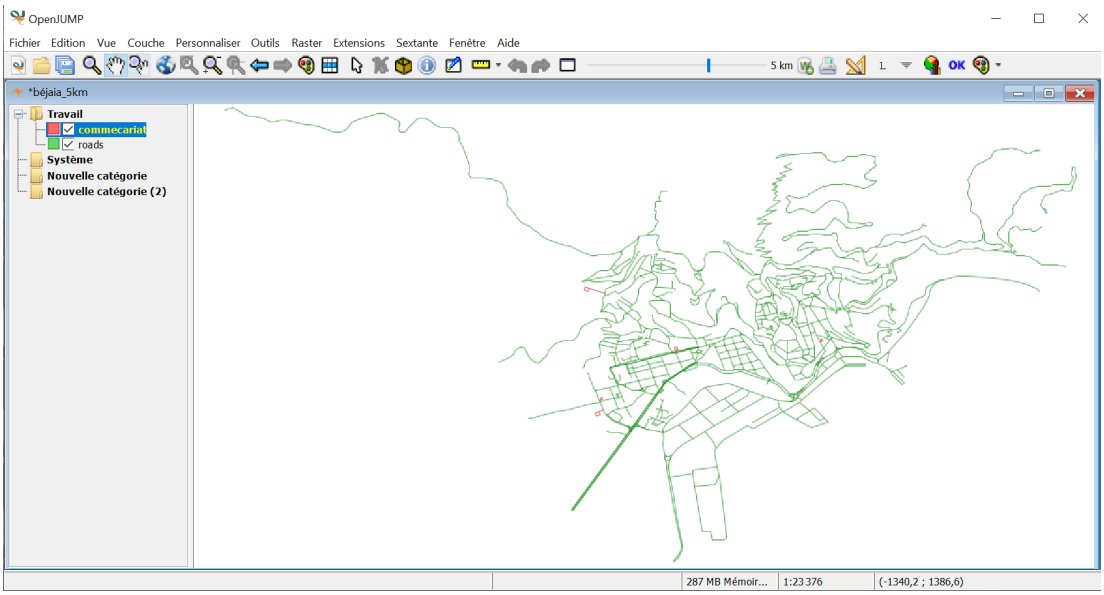

FIGURE 3.2 – Capture de l'édition de la carte géographique de Bejaia dans OpenJump.

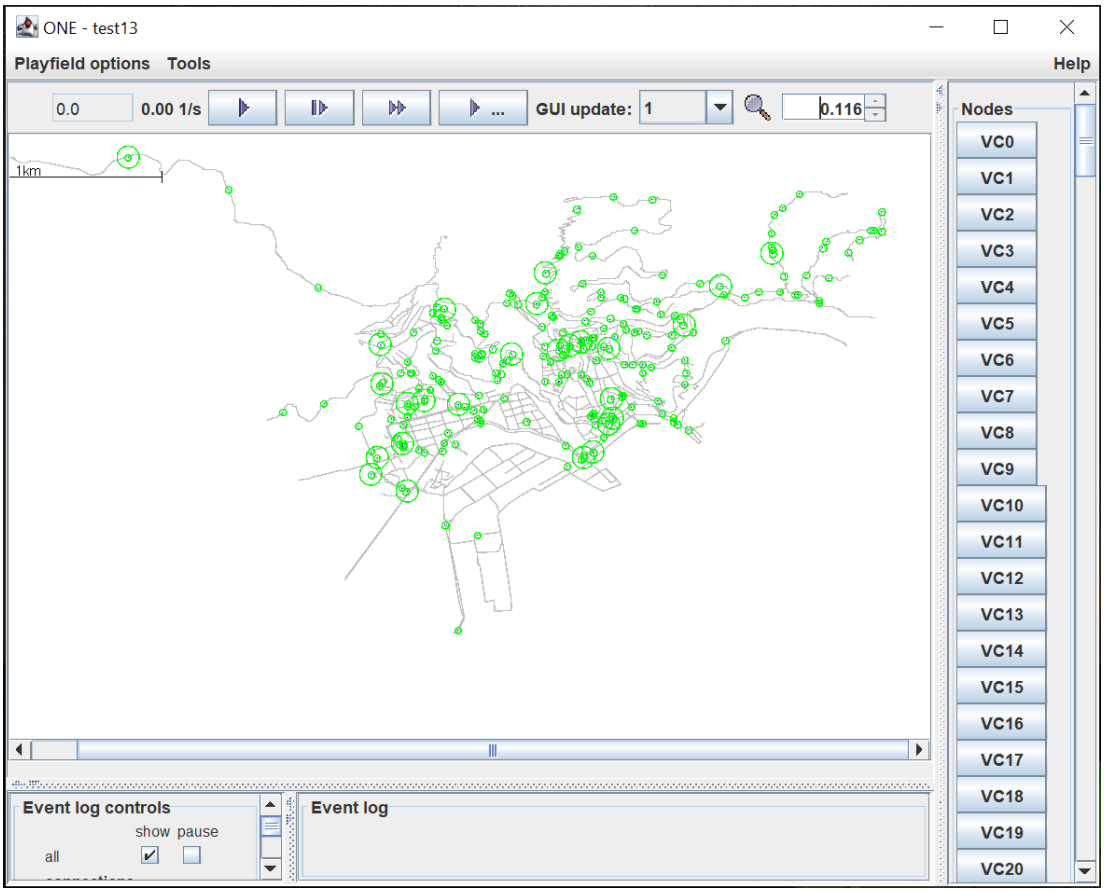

FIGURE 3.3 – Capture de l'interface graphique du simulateur ONE

3.6 Configuration du simulateur ONE

3.6.1 Mise en route du simulateur ONE

Après avoir téléchargé le code source du simulateur ONE sur GITHUB [3], nous avons utilisé l'environnement de développement ECLIPSE [5] (Nous avons utilisé la version 2022-03 '4.23.0'). Ensuite, nous avons intégré le code source du simulateur ONE dans eclipse sous forme d'un projet sous le nom de 'ONE'. Après cela, il faut faire les configurations suivantes :

1) **Intégration de 'DTNConsoleConnection.jar' et 'ECLA.jar' :** Faites un clique droit sur le projet 'ONE' dans ECLIPSE, Sélectionnez 'Build Path', puis 'Configure Build Path'. Ensuite dans la section 'Java Build Path' et dans l'onglet 'libraries' selectionez 'Classpath' puis cliquez sur le boutton 'AddjaRs', ensuite sélectionnez les

deux librairies qui sont dans le dossier 'lib' du projet 'ONE', et enfin cliquez sur 'Apply' et 'Apply and Close'.

2) **Ajout de la 'junit-4.12-beta-3.jar' :** Faites un clique droit sur le projet 'ONE' dans ECLIPSE, Sélectionnez 'Build Path', puis 'Add External Archives' puis sélectionnez la 'junit-4.12-beta-3.jar' (que vous auriez téléchargé sur internet).

3) **Configuration de la compilation du simulateur :** Faites un clic droit sur le projet 'ONE' dans ECLIPSE, Sélectionnez 'Run As', puis 'Run Configurations', ensuite, faites un double clic sur la section Java Application, vous allez voir une nouvelle sous section 'New Configuration' apparaître, cliquez sur cette dernière et renommez la en 'ONE', après dans l'onglet 'Main' cliquez sur le bouton 'Search' ensuite cliquez sur 'DTNSim-core' puis sur le bouton 'OK',après entrez dans l'onglet Arguments et selectionnez le Bouton radio 'Others' puis le boutton 'WorkSpaces'. une fois c'est fait, vous choisissez le dossier 'src' dans le projet ONE. Vous cliquez sur le bouton 'Apply' puis sur 'Run' et vous verrez l'interface graphique sur le simulateur apparaître.

3.6.2 Configurations des interfaces de communications

Nous avons utilisé deux types d'interfaces de communications que nous avons configuré comme illustré dans la Figure 3.4 :

3.6.2.1 L'interface Bluetooth

C'est une interface de communication qui hérite de la classe java suivante "SimpleBroadcastInterface". Nous avons étendu sa portée à 20 mètres et fixé sa vitesse de transmission à un méga byte par seconde.

3.6.2.2 La HighSpeedInterface

C'est une interface de communication qui hérite de la classe "SimpleBroadcastInterface". Nous avons étendu sa portée à 70 mètres et fixé sa vitesse de transmission à 10 mégas byte par seconde.

```
18
19  # "Bluetooth" interface pour tous les noeuds
20  btInterface.type = SimpleBroadcastInterface
21  # vitesse de transsmission de 1 Mbps
22  btInterface.transmitSpeed = 1M
23  btInterface.transmitRange = 20
24
25  # High speed, porté plus lente
26  highspeedInterface.type = SimpleBroadcastInterface
27  highspeedInterface.transmitSpeed = 10M
28  highspeedInterface.transmitRange = 70
29
```

FIGURE 3.4 – Configurations des interfaces de communications sur le simulateur ONE

3.6.3 Configurations des nœuds de notre réseau

Nous avons utilisé un modèle de mouvement qui permet à nos nœuds une fois qu'ils déterminent leurs destinations (dans l'ensemble des routes que nous avons pu intégré) de prendre le plus court chemin en utilisant la classe JAVA (ShortestPathMapBasedMovement), et nous avons fixé le temps d'attente de ces nœuds une fois ils sont arrivés à destination à un intervalle qui varie entre 0 et 120 secondes afin qu'ils puissent ensuite repartir de nouveau vers une nouvelle destination.

```
52
53  Group.movementModel = ShortestPathMapBasedMovement
54  Group.router = EpidemicRouter
55  Group.waitTime = 0, 120
56  # Tous les noeuds ont une interface bluetooth
57  Group.nrofInterfaces = 1
58  Group.interface1 = btInterface
59
```

FIGURE 3.5 – Configurations des nœuds de notre réseau sur le simulateur ONE

Nous avons utilisé quatre types de nœuds répartis en 4 groupes :

3.6.3.1 Véhicules de civiles

Les véhicules des civiles peuvent être de type : voitures, mobylettes, fourgons, camionnettes, camions, etc. Ils sont la colonne vertébrale de notre réseau et par conséquent de notre simulation, car ce sont eux qui vont générer le trafic routier, ce qui va garantir un dynamisme et une mobilité dans le réseau. Nous avons fixé la vitesse des nœuds de ce groupe dans un intervalle qui varie d'une manière aléatoire entre 38.7 et 56.9 kilomètres par heure. Nous les avions équipés d'une interface Bluetooth qui sera utilisée pour transmettre les infractions (détectées à partir des OBU à bord) qui sont commises par les conducteurs, et aussi afin de relayer les infractions des autres automobilistes. Le nombre de nœuds de ce groupe varie entre 20 et 200 véhicules selon le type de scénario de simulation effectué. Nous avons équipé chaque nœud d'un buffer, qui a une capacité de stockage d'un giga byte. Les infractions engendrées et émises dans le réseau IoV ont une durée de vie de 480 minutes (8 heures). Le préfixe des identifiants des nœuds de ce groupe est VC (Véhicule Civile), ils se déplacent d'une manière aléatoire sur l'ensemble des routes de notre réseau. Ces configurations sont illustrées sur la Figure 3.6

```
72
73  # (obu) group de voiture de civile
74  Group1.groupID = VC
75  Group1.speed = 38.7, 50.9
76  Group1.nrofHosts = 200
77  Group1.bufferSize = 1G
78
```

FIGURE 3.6 – Configurations des voitures de civiles sur le simulateur ONE

3.6.3.2 Emplacements des forces de l'ordre

Nous avons effectué une recherche sur les différents emplacements des forces de l'ordre (commissariats, brigades de gendarmerie, douanes, casernes militaires) dans la carte que nous avons sélectionnée, et nous avons trouvé six emplacements. Nous les avons représentés sous forme de nœuds fixes(immobiles) et nous les avons équipés d'un buffer avec une capacité de mémoire de 2 gigas bytes (nous aurions aimé équiper les commissariats avec un buffer d'une capacité de mémoire plus grandes mais malheureusement le simulateur ONE est limité à une mémoire de deux gigas bytes). Nous avons équipé les nœuds de ce groupe avec deux interfaces de communications : une interface Bluetooth pour interagir avec les véhicules de civiles et une HighSpeedInterface pour interagir avec les véhicules de police. L'utilité principale de ces

nœuds est de recevoir les infractions émises par les véhicules de civiles et les RSUs. Le préfixe des identifiants des nœuds de ce groupe est FO (Force de l'Ordre) et leurs configurations sont présentées dans la Figure 3.7

```
89
90  # poste police 1
91  Group3.groupID = PP
92  Group3.MovementModel = StationaryMovement
93  Group3.okMaps = 1
94  Group3.speed = 0, 0
95  Group3.nrofHosts = 1
96  Group3.nrofInterfaces = 2
97  Group3.interface1 = btInterface
98  Group3.interface2 = highspeedInterface
99  Group3.bufferSize = 2G
100
```

FIGURE 3.7 – Configurations des emplacements de forces de l'ordre sur le simulateur ONE

3.6.3.3 Véhicules de polices

Nous avons mis en place 10 nœuds qui joueront le rôle de véhicules de police. Nous les avons équipés de deux types d'interfaces de communications Bluetooth et HighSpeedInterface. Ils interagissent avec les véhicules de civiles via l'interface Bluetooth, et avec les postes de police et les RSUs via la HighSpeedInterface.

Ces véhicules circuleront de manière aléatoire dans le réseau à des vitesses variant entre 35,5 et 55,5 kilomètres à l'heure. Nous avons configuré la mémoire de leur buffer à deux gigas bytes. Les nœuds de ce groupe apparaissent dans notre réseau avec le préfixe VP (Véhicule de Police). La Figure 3.8 montre ces différentes configurations.

Le rôle principal des véhicules de police consiste à recueillir des messages d'infractions routières qui transitent dans le réseau.

```
160
161   # (obu) group2 voiture de polices
162   Group9.groupID = VP
163   Group9.speed = 35.5, 55.5
164   Group9.nrofHosts = 10
165   Group9.nrofInterfaces = 2
166   Group9.interface1 = btInterface
167   Group9.interface2 = highspeedInterface
168   Group9.bufferSize = 2G
169
170
```

FIGURE 3.8 – Configurations des voitures de polices sur le simulateur ONE

3.6.3.4 Les unités de bord de route (RSUs)

C'est un type très particulier de nœuds fixes que nous avons éparpillés un peu partout le long des routes de notre carte routière. Ils sont présents dans les radars, les feux de signalisation, etc. Ils servent à signaler toutes les infractions au code de la route qu'ils détectent. Ils participent aussi au processus de routage des messages dans le réseau. Nous les avons dotés d'un buffer avec une mémoire de stockage de 2 gigas octets. Comme pour les nœuds des commissariats, nous aurions aimé équiper ces RSU avec une taille de mémoire un peu plus grande, mais malheureusement le simulateur ne le permet pas. Les nœuds de ce groupe apparaissent dans notre réseau avec le préfixe « RSU ». Ils possèdent également 2 interfaces de communication (Bleutooth et HighSpeedInterface) comme le montre la Figure 3.9.

Nous avons préféré mettre ces RSUs à bord des feux tricolores et des radars afin qu'ils puissent bénéficier du même niveau de sécurité que ces derniers reçoivent de la part des autorités concernées, car ce sont des équipements qui coûtent extrêmement chers.

```
78
79  # (RSU) barage, feu tricolore, radar ...
80  Group2.groupID = RSU
81  Group2.MovementModel = StationaryMovement
82  Group2.speed = 0, 0
83  Group2.nrofHosts = 10
84  Group2.nrofInterfaces = 2
85  Group2.interface1 = btInterface
86  Group2.interface2 = highspeedInterface
87  Group2.bufferSize = 2G
88
89
```

FIGURE 3.9 – Configurations des RSUs sur le simulateur ONE

3.6.4 Configurations des points d'intérêts

Les POIs (Points Of Intrests) en français points d'intérêts, sont des points géographiques bien précis dans la carte du simulateur, on les utilise uniquement avec la configuration du protocole PROPHET. On utilise ses points afin de fixer des probabilités de fréquentations de certains endroits (POIs) avec un ou des groupes de noeuds donnés.

Dans notre application, nous avons mis les 6 commissariats cités ci-dessus comme points d'intérêt, et nous avons fixé la probabilité initiale de fréquentation de ces commissariats avec les véhicules de civiles à 0.3, comme illustré sue la Figure 3.10. Le choix de cette valeur est justifié par le fait que notre application est essayé pour l'instant dans la ville de Bejaia, alors en général si un véhicule passe pour la première fois devant un commissariat il y a une forte chance qu'il repasse.

```
242
243  # definir les fichiers des POIs
244  PointsOfInterest.poiFile1 = data/Central_police_POIs.wkt
245
246
247  # définir les probabilités de fréquentation des noeuds des POIs
248  Group1.pois = 1,0.3
249
250
```

FIGURE 3.10 – Configurations des POIs sur le simulateur ONE

3.6.5 Interactions entre les nœuds

Dans le Tableau 3.2, nous avons montré les différentes possibilités de connexion en relation avec les différents types de groupes de nœuds :

	véhicules de civiles	véhicules de polices	RSU	emplacement de force de l'ordre
véhicules de civiles	/	+ (blt)	+(blt)	+(blt)
véhicules de polices	+ (blt)	/	+ (hsi)	+ (hsi)
RSU	+ (blt)	+ (hsi)	/	/
emplacement des forces de l'ordre	+ (blt)	+ (hsi)	/	/

\+ : possibilité de connexion / : pas de possibilité de connexion
blt : Bluetooth hsi : HighSpeedInterface

Tableau 3.2 – Tableau des interactions entre les nœuds sur le simulateur ONE

3.6.6 Configuration des messages d'infractions

Les messages d'infractions routières sont générés à partir des OBUs à bord des véhicules de civiles et aussi à partir des feux tricolores et des radars qui sont équipés de RSU. Nous avons fixé leurs tailles entre cinq et dix kilos bytes chacun, et configuré leur intervalle de génération entre 95 et 105 secondes. A chaque fois, le simulateur prend une valeur aléatoire dans cette intervalle pour définir le temps qu'il faut attendre afin qu'une infractions soit commit par un des véhicules de civiles. Nous avons également mis un TTL (Time To Live) avec une durée de 8heurs, ce qui signifie que les messages ont une durée de vie de 8 heures.

Sur notre réseau il n'y auras que des messages d'infractions routières qui vont circuler sur les routes de notre carte géographique. Tout ces paramètres sont configurés comme montré sur la Figure 3.11

```
180
181  ## Parametres de création de messages
182  # Nombre d'evenment généré
183  Events.nrof = 1
184  # la classe de la génération du premier evenement
185  Events1.class = MessageEventGenerator
186  # L'interval de création de message en secondes
187  Events1.interval = 95, 105
188  # tailles des Messages (50kB - 60kB)
189  Events1.size = 5k,10k
190  # Les champs d'addresses des message source/destination
191  Events1.hosts = 0, 199
192  Events1.tohosts = 210, 215
193
194  # L'id des préfixes des messages
195  Events1.prefix = M
196
```

FIGURE 3.11 – Configurations des messages d'infractions sur le simulateur ONE

3.6.7 Configuration des protocoles de routage utilisés

Nous avons choisi pour le routage des messages d'infractions routières deux protocoles de routage existants pour analyser les performances de notre application sur le simulateur ONE. Il s'agit de Epidemic et Prophet.

Pour utiliser le protocole Epidemic, ça ne nécessite pas une configuration particulière, il faut juste déclarer la classe java 'EpidemicRouter' dans la configuration de routage des nœuds du réseau.

Pour utiliser le protocole Prophet, nous avons fixé 6 points d'intérêt POIs (Point of interest), au niveau de nos 6 commissariats, et nous avons fixé la probabilité initiale de fréquentation des véhicules de civiles de ses 6 POIs à 0,3.

3.6.8 Scénarios de simulation

Pour chaque protocole de routage, nous avons fait 5 simulations en variant le nombre de véhicules de civiles qui circulent dans notre réseau (nous avons simulé avec 20, 50, 100, 150 et enfin 200 véhicules).

Nous avons fixé le temps de simulation à 2.5 heures pour chaque scénario de simulation (voir la Figure 3.12).

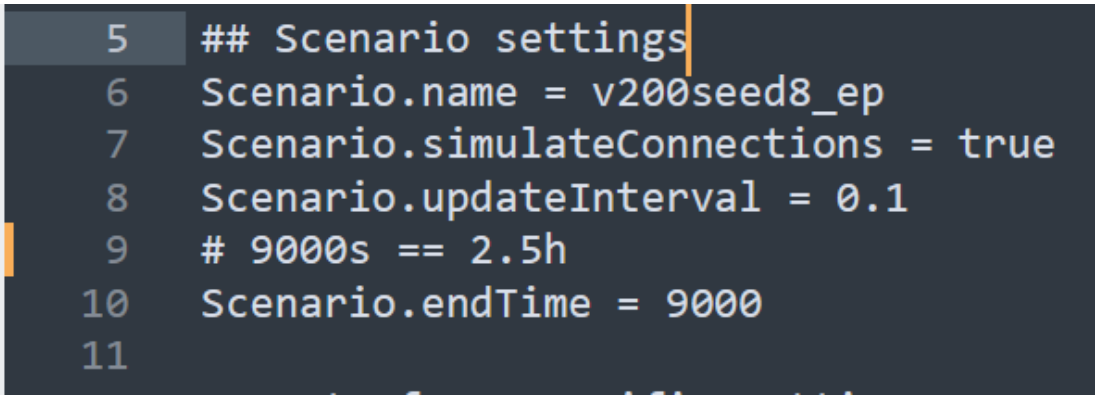

FIGURE 3.12 – Configuration d'un scénario de simulation

3.6.8.1 Les métriques de performance

Pour l'évaluation de performances de notre application avec les deux protocoles de routage Epidemic et Prophet, nous avons choisi de varier le nombre de véhicules de civiles en fonctions duquel nous mesurons les métriques suivantes :

a) le taux de livraison (delivery rate) : c'est un paramètre qui définit le rapport entre le nombres de messages créés (les infractions émises dans le réseau) avec le nombre de messages délivrés.

b) le taux de surcharge du réseau (overhead ratio) : c'est le nombre de copies des messages émis dans le réseau qui sont stockés et relayés par les nœuds intermédiaires dans le réseau.

c) la latence moyenne (latency average) : c'est le temps moyen que les messages prennent pour atteindre leurs destination.

d) la moyenne du nombre de sauts (hopcount medium) : c'est le nombre de sauts moyen nécessaire pour que les messages atteignent leur destination.

3.7 Test et résultats

Nous présentons les résultats de simulation dans des graphes comme suit.

3.7.1 Résultats du taux de livraison en fonction du nombre de véhicules

Dans la Figure 3.13, nous avons présenté le taux de livraison en fonction du nombre de véhicules de civiles pour les deux scénarios de simulation avec Epidemic et Prophet.

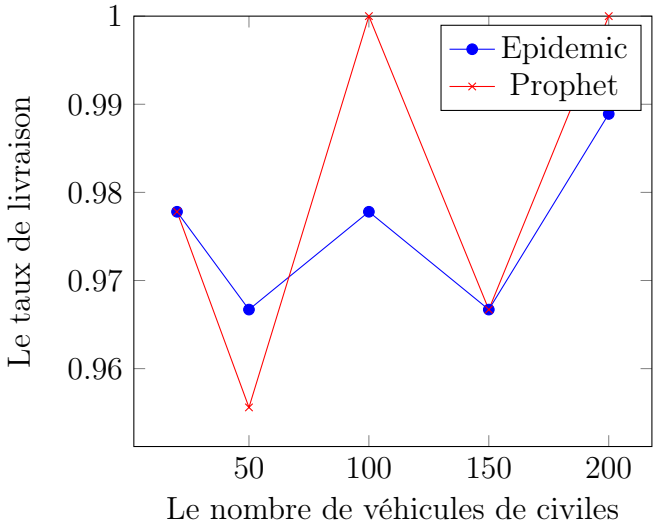

FIGURE 3.13 – Le taux de livraison en fonction du nombre de véhicules de civiles

Avec l'Epidemic, nous avons eu des taux de livraison qui varie entre 97 et 99 pour cent tandis qu'avec le protocole Prophet, nous avons pu atteindre une probabilité de livraison de 100 pour cent avec 100 et 200 véhicules.

3.7.2 Résultats de la latence en fonction du nombre de véhicules

Dans la Figure 3.14, nous avons présenté la latence en fonction du nombre de véhicules de civiles pour les deux scénarios de simulation avec Epidemic et Prophet.

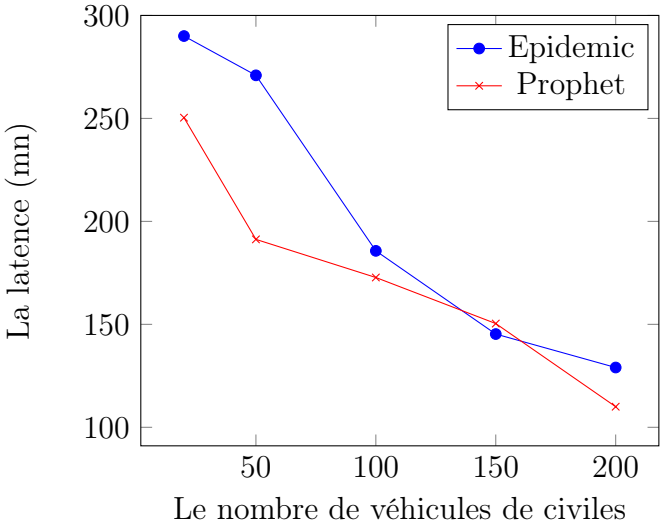

FIGURE 3.14 – La latence en fonction du nombre de véhicules de civiles

En ce qui concerne la latence, nous avons pu remarquer une légère amélioration du protocole Prophet par rapport à Epidemic.

3.7.3 Résultats du taux de surcharge du réseau en fonction du nombre de véhicules

Dans la Figure 3.15, nous avons présenté le taux de surcharge du réseau en fonction du nombre de véhicules de civiles pour les deux scénarios de simulation avec Epidemic et Prophet.

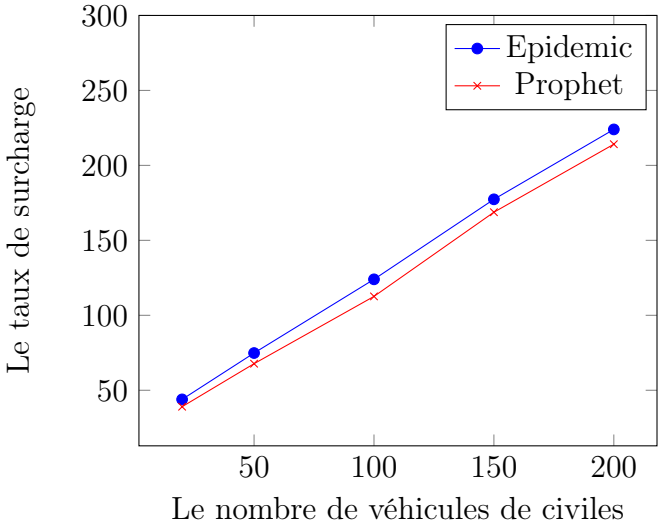

FIGURE 3.15 – Le taux de surcharge du réseau en fonction du nombre de véhicules de civiles

Concernant le taux de surcharge du réseau, nous avons constaté une très légère amélioration du taux de surcharge du Prophet par rapport à Epidemic.

3.7.4 Résultats du nombre de sauts en fonction du nombre de véhicules

Dans la Figure 3.16, nous avons présenté le nombre de saut en fonction du nombre de véhicules de civiles pour les deux scénarios de simulation avec Epidemic et Prophet.

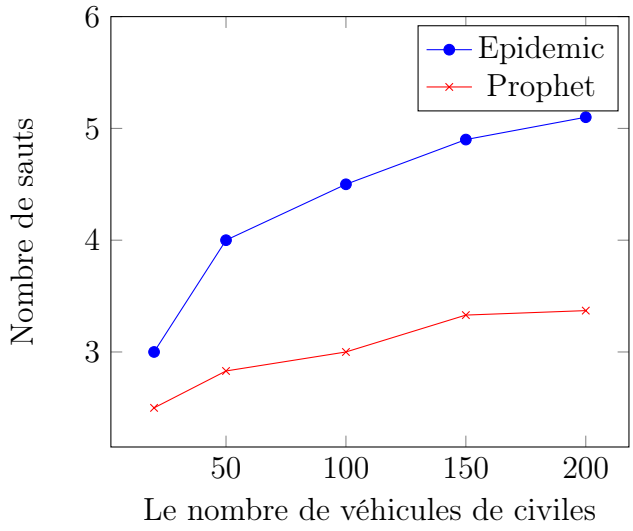

FIGURE 3.16 – Le nombre de sauts en fonction du nombre de véhicules de civiles

Nous constatons dan ce dernier graphe que Prophet délivre les messages avec un nombre de saut minimal.

3.8 Discussion des résultat obtenus

D'après les résultats obtenus, nous constatons que le protocole Prophet est le mieux adapté et le plus adéquat à notre application de signalement d'infractions routières car il assure moins de latence, de nombres de sauts et de surcharge du réseau par rapport au protocole Epidemic. Cependant, pour avoir de meilleurs résultats dans des scénarios plus réalistes, nous avons besoin de proposer une amélioration de Prophet. Cette dernière est le sujet de la section suivante.

3.9 Contribution à l'amélioration de Prophet

En analysant le trafic dans les villes et la mobilité des véhicules, nous constatons que nous pouvons encore mieux utiliser cette mobilité pour le routage de données vers la destination dans les réseaux véhiculaires. En effet, dans la réalité un véhicule i peut passer quotidiennement par un siège de la police si ce dernier est sur la route qui mène le propriétaire du véhicule de son domicile vers son lieu de travail. Alors, la rencontre du véhicule i avec ce siège de police dépend en fait de la journée de la semaine et de l'heure de passage. Prophet ne prend pas en considération cette information dans le calcul des probabilités de livraison. Alors, nous proposons que chaque véhicule détient une matrice de probabilité de livraison selon la journée de rencontre dans la semaine et l'heure de rencontre dans la journée. Notre matrice sera de 7 lignes (qui représentent les jours de la semaine avec la première ligne est dimanche) et 24 colonnes (qui représentent les heures de la journée avec la première colonne est 00 :00))

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,24} \\ P_{2,1} & d_{2,2} & \cdots & d_{2,24} \\ \vdots & \vdots & & \vdots \\ P_{7,1} & P_{7,2} & \cdots & P_{7,24} \end{pmatrix}$$

$P_{i,j}$: désigne la probabilité de rencontre du véhicule avec un point d'intérêt le jour i à l'heure j.

Pour désigner un relayeurs parmi plusieurs véhicules, Prophet choisit le véhicule qui a une probabilité qui représente la fréquence des rencontres du véhicule avec la destination supérieur à un seuil prédéfini P_{seuil}. Mais dans notre proposition, on choisit selon l'heure et

la date. Alors si le message d'infraction est généré à $T_{i,j}$, on choisit les véhicules qui ont une probabilité $P_{k,l}$ supérieur au seuil défini comme pour Prophet, avec

$$\begin{cases} k = i \; et \; j < l < ((j+\sigma) \mod 24) & \text{si } j < (24-\sigma) \\ k = (i \mod 7) + 1 \; et \; j < l < ((j+\sigma) \mod 24) & \text{sinon.} \end{cases}$$

Ceci permet de ne considérer que les probabilités de rencontre dans les σ heures qui viennent. Avec $\sigma \in [1, 24]$ et c'est un paramètre à ajuster en fonction de l'application dans le but de limiter les délais et la surcharge du réseau.

Dans ce qui suis nous allons présenter l'algorithme de notre contribution :

Algorithm 1: RelayageMessageProphetAmeliorer

```
/* Éxécuté par chaque véhicule 'a'                                          */
Input:
Pinit : réel ; /* Pinit ∈ [0,1]                                             */
1 Ta, Tb : tableau[i,j] de réels ; /* la matrice des probabilités de rencontre par
    jour et heure, avec i ∈ [0,6] et j ∈ [0,23]                            */
2 Véhicule : enregistrement
3 nom : chaine,
4 type : chaine,
5 Fin enregistrement ;
6 BEGIN
/* Initialisation                                                           */
7 if (Véhicule.nom = 'a') et (Véhicule.type = 'VéhiculeDePolice') then
8     for i allant de 0 à 6 do
9         for j allant de 0 à 23 do
10            Ta[i,j] = 1 ;
11 else if (Véhicule.nom = 'a') et (Véhicule.type = 'VéhiculeDeCivile') then
12    for i allant de 0 à 6 do
13        for j allant de 0 à 23 do
14            Ta[i,j] = Pinit ;
15 if RécéptionRequete() then
      /* Récéption d'une requête de la part de l'application, ie. rencontre de 'a'
         avec les stations, barrage ou véhicule de police                   */
16    J = JourDeLaRencontre ; H = HeureDeLaRencontre ; Ta(J,H) = Ta(J,H) + (1-
      Ta(J,H))*Pinit ;
17    DélivrerMessages() ; /* Délivrer des messages de signalement des infractions
         routières s'ils existent dans son buffer                           */
18 if CroisementAvecVéhicule(Véhicule.nom = 'b') then
19    i= JourDuCroisement() ;
20    j=HeureDuCroisement()+1 ;
21    nb=0 ;
22    while nb < TTL do
23        while (j < 24) et (nb < TTL) do
24            if Tb(i,j) >=Ta(i,j) then
25                EnvoyerMessages() ; /* Envoyer messages de signalement des
                     infractions routières                                  */
26                nb=TTL ;
27                j=24 ;
28            nb=nb+1 ;
29            j=j+1 ;
30        j=0 ;
31        i=(i+1) mod 7 ;
32 END
```

Prenons un exemple dans lequel nous allons voir la lacune de Prophet et comment notre proposition va y remédier.

Soit un véhicule i qui passe tout les jours de la semaine à part le week-end devant un siège de la police à 7h15 et 17h00. On suppose que le propriétaire habite dans le lieu A et travaille dans un lieu B et qu'un siège de la police se trouve au lieu c sur la route qui mène de A vers B.

Soit un véhicule k qui passe tout les jours de la semaine à part le week-end devant un siège de la police à 9h00 et 16h15. On suppose que le propriétaire habite dans le lieu B et travaille dans un lieu A et qu'un siège de la police se trouve au lieu c sur la route qui mène de B vers A.

Soit, un véhicule l qui fait une infraction au lieu F (voir la Figure 3.17 pour voir les lieux de l'exemple) le lundi à 8h00 et rencontre les deux véhicules i et k.

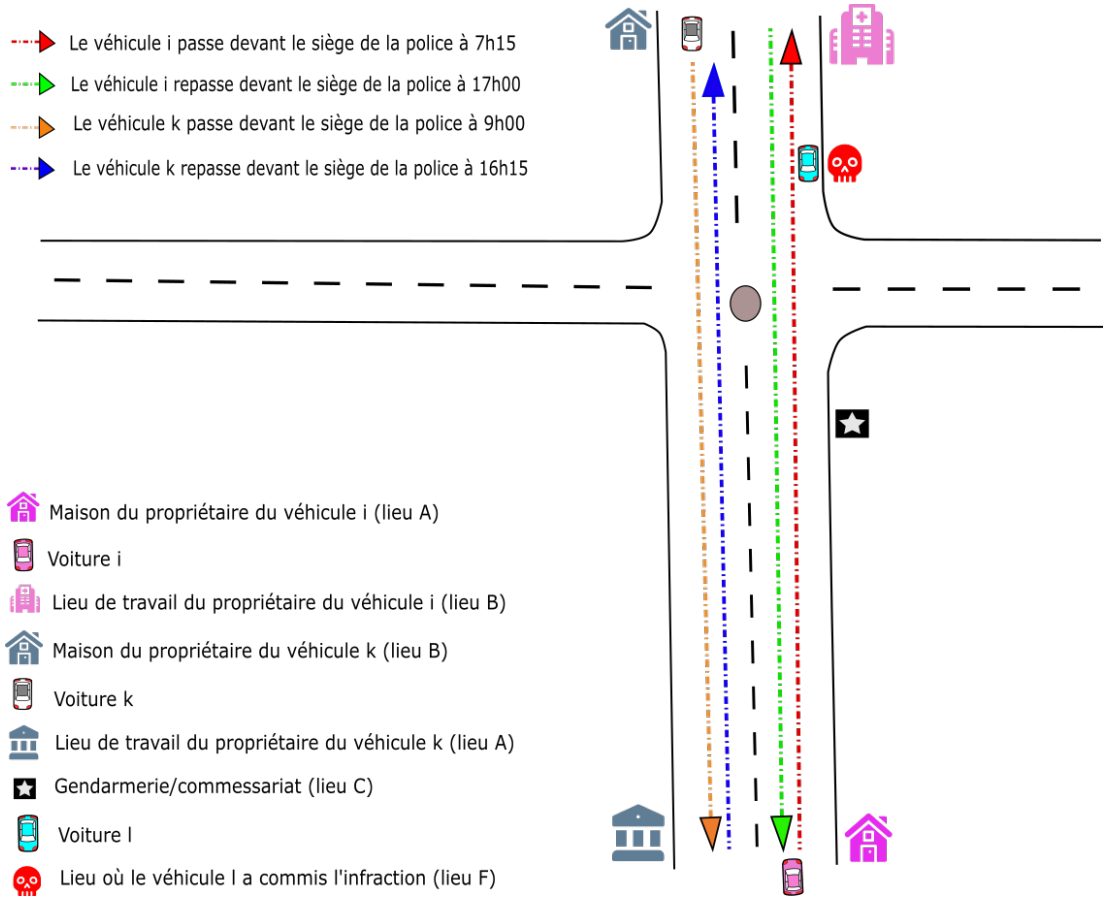

FIGURE 3.17 – Scénario du réseau qui illustre notre contribution

Suivant ce scénario et en initialisant pour la journée du dimanche les probabilités pour les véhicules i et k, nous avons calculé les probabilité dans le cas de Prophet et dans le cas de notre proposition avec $P_{init} = 0.3$ comme présenté dans le tableau 3.3.
Et on met à jour les probabilités de rencontre avec la formule du protocole prophet suivante : $P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) * P_{init}$ [33]

	Prophet		Notre proposition	
Date et heure des événements	Véhicule i	Véhicule k	Véhicule i	Véhicule k
Dimanche à 15h30	0.6	= 0.6	$P_{1,8} = P_{1,18}$=0.3 (le reste des éléments de la matrice sont à 0)	$P_{1,10} = P_{1,17}$=0.3 (le reste des éléments de la matrice sont à 0)
Dimanche à 16h15 (k rencontre POI)	0.6	0.72	$P_{1,8} = P_{1,18}$=0.3 (le reste des éléments de la matrice sont à 0)	$P_{1,10}$ = 0.3 et $P_{1,17}$=0.51 (le reste des éléments de la matrice sont à 0)
Dimanche à 17h15 (i rencontre POI)	0.72	0.72	$P_{1,8} = 0.3 et P_{1,18}$=0.51 (le reste des éléments de la matrice sont à 0)	$P_{1,10}$ = 0.3 et $P_{1,17}$=0.51 (le reste des éléments de la matrice sont à 0)
Lundi à 7h30 (i rencontre POI)	0.804	0.72	$P_{2,8} = 0.51 et P_{2,18}$=0.51 (le reste des éléments de la matrice sont à 0)	$P_{2,10}$ = 0.3 et $P_{2,17}$=0.51 (le reste des éléments de la matrice sont à 0)
Lundi à 8h00 (i et k rencontre l qui fait une infraction)	0.804	0.72	$P_{2,8} = 0.51 et P_{2,18}$=0.51 (le reste des éléments de la matrice sont à 0)	$P_{2,10}$ = 0.3 et $P_{2,17}$=0.51 (le reste des éléments de la matrice sont à 0)

Tableau 3.3: Tableau des probabilités de l'exemple illustratif

Maintenant, en utilisant les données du tableau, à 8h00 quand l'infraction est commise par le véhicule l et le message est généré, il faut choisir des relayeurs pour le message.

Pour Prophet, si nous fixons $P_{seuil} = 0.75$, le nœud l va transférer le message d'infraction au véhicule i et non au véhicule k puisque à cet instant $P_i = 0.804 > 0.75$ mais $P_k = 0.72 < 0.75$.

Pour notre proposition, si nous fixons $P_{seuil} = 0.4$ et $\sigma = 8$, le nœud l va transférer le message d'infraction au véhicule k et non au véhicule i puisque à cet instant $Pi_{2,n} = 0 < 0.4$ pour $n \in [10, 17]$ mais $Pk_{2,17} = 0.51 > 0.4$.

Cette analyse à travers l'exemple illustratif n'est pas vraiment une évaluation de performance qui prouve que notre approche est meilleure que Prophet mais elle est prometteuse en attendant d'avoir des résultats de simulation qui vont la compléter.

3.10 Conclusion

Dans ce dernier chapitre, nous avons présenté notre application de signalement d'infraction routières. Ensuite, nous avons enchaîné avec une petite présentation du simulateur ONE et nous avons évoqué ce qui nous a motivé à utiliser ce simulateur. Après ça, nous avons cité les étapes de l'implémentation de la carte géographique du centre de la ville de Béjaia dans le simulateur, et nous avons donné et expliqué toutes les configurations que nous avons pu faire dans le simulateur afin d'implémenter notre application dans ce dernier.

Par la suite, nous avons donné et commenté les résultats que nous avons obtenus après la simulation avec deux protocoles de routage (Epidemic et Prophet), et nous avons déduit que le protocole Prophet est plus adapté à notre application mais nécessite une amélioration. C'est pour cela que nous avons à la fin présenté notre contribution à l'amélioration de Prophet et nous l'avons expliqué avec un exemple illustratif.

Conclusion et perspectives

Ce travail a été réalisé dans le cadre de notre projet de fin de cycle master en réseaux et systèmes distribués. Dans ce projet, nous avons proposé une application de signalement d'infractions routières. Cette application est destinée à tous les conducteurs civiles qui circule sur les routes algérienne et sa fonctionnalité principale est le signalement d'infractions routières commises par ces derniers aux autorités concernées (forces de l'ordre). Dans ce mémoire, nous nous sommes intéressés à l'aspect routage des infractions émises par notre application dans un réseau IoV. Dans notre approche, nous avons initialement adapté le protocole Prophet en utilisant une métrique sous forme d'un facteur qui indique le taux de fréquentations d'un véhicule des lieux de forces d'ordres (commissariats, brigade , casernes, etc), et plus ce facteur est grands plus on favorise ces véhicules pour relayer les infractions routières qui circulent dans le réseau jusqu'aux destinataires finaux. Après, nous avons proposé une amélioration du protocole Prophet.

La réalisation de ce projet a débuté par des généralité et des définitions qui ont une relation de près ou de loin avec notre projet à savoir c'est quoi l'IoT, IoV, MANET, VANET, etc. Après avoir bien situé notre travail en donnant la terminologie nécessaire qui concerne notre projet, nous avons donné un état de l'art sur les protocoles de routage opportunistes en résumant 5 articles scientifique qui concerne des protocoles de routages (Epidemic, Prophet, TDOR, MOP et le protocole multicritères) que nous avons trouvé intéressant pour assurer le routage dans l'application que nous avons proposé. Ensuite, nous avons porté comme candidat deux protocoles de routage (Epidemic et Prophet) pour faire la simulation du routage lorsque nous avons implémenté notre applications sur le simulateur ONE.

Après avoir obtenus les résultats de simulations dans ONE (taux de livraison, taux de surcharge, nombre de saut moyen et la latence), nous avons choisi le protocole Prophet pour ses performances et ses bons résultats afin de l'améliorer pour assurer le routage des infractions émises par notre application. Ainsi, à la fin de ce projet, nous avons proposé une

amélioration du protocole de routage opportuniste Prophet qui calcule des probabilités de livraisons en fonction du temps de rencontre entre les véhicules et les points d'intérêt. Nous estimons que cette contribution va améliorer considérablement les performances de l'application et le routage opportuniste en général. Cependant, nous n'avions pas eu le temps d'évaluer ses performances par simulation puisque nous avons besoin d'un ensemble de données réelle qui est difficile à trouver.

Ce projet représente une occasion pour mettre en pratique nos connaissances que nous avons acquis lors de notre cursus universitaire, notamment dans le domaine des réseaux informatiques.

A la fin de ce travail, nous avons conscience qu'il y'a toujours une possibilité d'améliorer et de continuer le développement de de projet :

* Concevoir une api de gestion d'infraction routière et l'héberger sur le Cloud et qui va continuer à marcher en hors ligne dans les commissariats de police en cas de coupure d'Internet.

* Recolter des données du monde réel et construire un data set qui donne les fréquences de passage des véhicules devant les commissariats de Béjaia afin d'alimenter les probabilités des POIS initial avec des données plus précises et évaluer les performances de notre proposition et les comparer avec ceux de Prophet et MOP.

* Nous ambitionnons de faire à l'avenir des simulations avec un nombre important de véhicules afin de bien souligner les différences entre les performances de notre proposition, Prophet et MOP (cela va nécessiter l'utilisation d'un super ordinateur car le simulateur ONE est gourmand en terme de consommation de ressource).

Bibliographie

[1] Adyton. https://github.com/npapanik/Adyton. Dernier accès le 2022-11-06.

[2] Certified obu for hu-go toll payment in hungary. https://shop.obu1.eu/en/product/obu-standard/. Dernier accès le 2022-06-7.

[3] Code source du simulateur one v1.6.0. https://github.com/akeranen/the-one. Dernier accès le 2022-01-6.

[4] Cornell law school, legal information institute, 47 cfr § 90.7 - definitions. https://www.law.cornell.edu/cfr/text/47/90.7. Dernier accès le 2022-05-7.

[5] Eclips. https://www.eclipse.org. Dernier accès le 2022-05-7.

[6] Kevin ashton. https://en.wikipedia.org/wiki/Kevin_Ashton. Dernier accès le 2022-09-30.

[7] Kevin fall. https://www.linkedin.com/in/kfall. Dernier accès le 2022-09-30.

[8] Netsim network simulator. https://www.boson.com/netsim-cisco-network-simulator. Dernier accès le 2022-11-06.

[9] Ns3 network simulator. https://www.nsnam.org/. Dernier accès le 2022-11-06.

[10] Openjump. http://www.openjump.org. Dernier accès le 2022-05-16.

[11] Opnet network simulator. https://opnetprojects.com/opnet-network-simulator/. Dernier accès le 2022-11-06.

[12] The opportunistic network environment simulator. https://www.netlab.tkk.fi/tutkimus/dtn/theone/. Dernier accès le 2022-01-11.

[13] osm. https://www.openstreetmap.org. Dernier accès le 2022-05-13.

[14] osm2wkt. `https://github.com/julianofischer/osm2wkt`. Dernier accès le 2022-05-7.

[15] Siemens mobility, inc.'s roadside unit is first to receive omniair certification. `https://www.businesswire.com/news/home/20181008005481/en/Siemens-Mobility-Inc.s-Roadside-Unit-is-First-to-Receive-OmniAir-Certification`. Dernier accès le 2022-06-7.

[16] Union internationale des télécommunications. `https://fr.wikipedia.org/wiki/Union_internationale_des_tÃľlÃľcommunications`. Dernier accès le 2022-11-24.

[17] *Oxford Dictionary*. Oxford University Press, 2018.

[18] Maythem K Abbas, Tan Jung Low, and Raed Abdulla. Automated fining system for high speed driving offences via VANET. In *2019 International Conference on Green and Human Information Technology (ICGHIT)*, pages 36–38. IEEE, 2019.

[19] Ahmed Mohamed Abdalla and Salem H Salamah. Performance comparison between delay-tolerant and non-delay-tolerant position-based routing protocols in vanets. *International Journal of Communications, Network and System Sciences*, 15(1) :1–14, 2022.

[20] Syed Hassan Ahmed, Muhammad Azfar Yaqub, Safdar H Bouk, and Dongkyun Kim. Towards content-centric traffic ticketing in VANETs : An application perspective. In *2015 Seventh International Conference on Ubiquitous and Future Networks*, pages 237–239. IEEE, 2015.

[21] Lylia Alouache, Nga Nguyen, Makhlouf Aliouat, and Rachid Chelouah. New robust protocol for iov communications. *Challenges of the Internet of Things : Technology, Use, Ethics*, 7 :137–163, 2018.

[22] Alvaro Torres Amaya, Mauro Sergio Fonseca, Alexandre Pohl, and Ricardo Lüders. Performance assessment of dtn and vanet protocols for transmitting periodic warning messages in high vehicular density networks. *Journal of Communication and Information Systems*, 37(1) :91–103, 2022.

[23] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. Dtn routing as a resource allocation problem. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 373–384, 2007.

[24] MA Baovola, PA Randriamitantsoa, and AN Andriamanalina. Évaluation de la performance du protocole ieee 802.11 p sur la couche phy avec la communication véhicule-véhicule dans un réseau véhiculaire ad hoc.

[25] W. BOUKSANI. Gestion de la protection de la vie privée dans les reseaux vehiculaires (vanet), 2017.

[26] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune. Geographic routing protocols for vehicular ad hoc networks (vanets) : A survey. *Vehicular Communications*, 11 :20–31, 2018.

[27] Yue Cao, Omprakash Kaiwartya, Nauman Aslam, Chong Han, Xu Zhang, Yuan Zhuang, and Mehrdad Dianati. A trajectory-driven opportunistic routing protocol for vcps. *IEEE Transactions on Aerospace and Electronic Systems*, 54(6) :2628–2642, 2018.

[28] Clayson Celes, Azzedine Boukerche, and Antonio AF Loureiro. Mop : A novel mobility-aware opportunistic routing protocol for connected vehicles. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2020.

[29] J Harri, PC Cheng, JT Weng, LC Tung, M Gerla, and K Lee. Geodtn+ nav : a hybrid geographic and dtn routing with navigation assistance in urban vehicular networks. In *Proceedings of the First Annual International Symposium on Vehicular Computing Systems, Dublin, Ireland*, 2008.

[30] David B Johnson. Routing in ad hoc networks of mobile hosts. In *1994 First Workshop on Mobile Computing Systems and Applications*, pages 158–163. IEEE, 1994.

[31] Brad Karp and Hsiang-Tsung Kung. Gpsr : Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, 2000.

[32] Kevin C Lee, Michael Le, Jerome Harri, and Mario Gerla. Louvre : Landmark overlays for urban vehicular routing environments. In *2008 IEEE 68th Vehicular Technology Conference*, pages 1–5. IEEE, 2008.

[33] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3) :19–20, 2003.

[34] Sanoop Mallissery, MM Manohara Pai, Nabil Ajam, Radhika M Pai, and Joseph Mouzna. Transport and traffic rule violation monitoring service in ITS : A secured VANET cloud application. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 213–218. IEEE, 2015.

[35] Khaleel Mershad, Hassan Artail, and Mario Gerla. Roamer : Roadside units as message routers in vanets. *Ad Hoc Networks*, 10(3) :479–496, 2012.

[36] Rida MEZGHACHE. Reconfiguration dynamique des architectures logicielles. 2018.

[37] Lewis Nkenyereye and Kyung-Hyune Rhee. Secure and privacy preserving protocol for traffic violation reporting in vehicular cloud environment. *Journal of Korea Multimedia Society*, 19(7) :1159–1165, 2016.

[38] Karen Rose, Scott Eldridge, and Lyman Chapin. The internet of things : An overview. *The internet society (ISOC)*, 80 :1–50, 2015.

[39] Amin Vahdat, David Becker, et al. Epidemic routing for partially connected ad hoc networks, 2000.

[40] M. Zaigham. *Connected vehicles in the Internet of things*. Springer, 2020.

[41] Morteza M Zanjireh, Ali Shahrabi, and Hadi Larijani. Anch : A new clustering algorithm for wireless sensor networks. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, pages 450–455. IEEE, 2013.

[42] Jing Zhao and Guohong Cao. Vadd : Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE transactions on vehicular technology*, 57(3) :1910–1922, 2008.

RÉSUMÉ

Ce document est rédigé en vue de l'obtention du diplôme de master recherche réseaux et systèmes distribués et il présente le projet que nous avons réalisé. L'objectif de ce dernier est d'étudier le routage opportuniste dans l'internet des véhicules (IoV) et sa relation avec la couche application. Pour atteindre cet objectif, nous avons proposé un système de signalement d'infractions routières qui a pour but de rapporter des infractions routières qu'un conducteur pourrait commettre à la police. Puis, nous avons étudié et évalué le routage opportuniste des infractions rapportées par des unités embarquées dans les véhicules ou des unités routière à travers un réseaux de véhicules connectés. Pour ce faire, nous avons pris notre ville Béjaia comme cas d'étude pour notre application et nous avons utilisé le simulateur ONE pour évaluer les performances des deux protocoles opportunistes de référence à savoir Epidemic et Prophet après leur adaptation à notre application. Nous avons par la suite, proposé une amélioration de Prophet qui va permettre de réduire la latence et la surcharge du réseau en augmentant le taux de livraison.

Mots clés : IoV ; DTN ; Signalement d'infractions routières, Routage opportuniste ; Simulateur ONE ;

ABSTRACT

This document is written with a view to obtaining the master's degree in networks and distributed systems research and it presents the project that we have carried out. The objective of the latter is to study opportunistic routing in the Internet of Vehicles (IoV) and its relationship with the application layer. To achieve this goal, we have proposed a traffic violation reporting system that aims to report traffic violations that a driver might commit to the police. Then, we studied and evaluated the opportunistic routing of violations reported by on-board vehicle units or roadside units through a network of connected vehicles. To do this, we took our city Béjaia as a case study for our application and we used the ONE simulator to evaluate the performance of the two reference opportunistic protocols, namely Epidemic and Prophet, after their adaptation to our application. We have then proposed an enhancement to Prophet that will reduce latency and network overhead by increasing the delivery rate.

Keys words : IoV ; DTN ; Traffic violation reporting ; Opportunistic routing ; ONE Simulator ;