



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : RTIC18/M2/2018

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Réseaux et technologies de l'information et de la
communication (RTIC)

La collecte de données dans l'internet des véhicules

Par :

HALIMI ROUFAIDA

Soutenu le : 24 juin 2018, devant le jury composé de :

Djerou Leila	MC-A	Président
Bitam Salim	MC-A	Rapporteur
Torki Fatima	MA-A	Examineur

Remerciements

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Ce travail de master n'est pas uniquement le fruit de mon travail, mais le résultat de la collaboration d'un grand nombre de personnes qui m'ont côtoyé de près - ma famille, mes collègues - ou de loin - simplement avec un bonjour un matin - durant quelques-uns de ces derniers mois de labeur, mais aussi durant tout mon cheminement.

Je voudrais exprimer mes sentiments les plus spontanés envers les personnes sans lesquelles ce travail de thèse n'aurait pas pu voir le jour. Leur aide, accompagnement et soutien m'ont été indispensables afin de pouvoir aboutir aux contributions de ma thèse.

Je suis très reconnaissante à Mr. Salim Bitam, mon directeur de thèse, qui m'a encadré et dirigé dans mes recherches tout au long de ces mois. Je lui dis ma gratitude pour l'aide compétente qu'il m'a apportée, pour ses encouragements et pour la confiance qu'il m'a toujours témoignée.

Mes plus chaleureux remerciements vont à la perle de mes yeux, ma mère Malika et à mon chère père Mohamed, qui ont toujours été là pour moi, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fier ».

Je souhaite poursuivre ces remerciements en exprimant ma gratitude envers mes collègues et les personnes que j'ai connu au cours de cette thèse de master et qui l'ont rendu plus agréable.

Merci à tous.

Dédicaces

A mes parents Malika et Mohamed

Affables, honorables, aimables : vous représentez pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Vos prières et vos bénédictions m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez pour tous les sacrifices que vous n'avez cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte. Vous avez fait plus que des parents puissent faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études. Aucun hommage ne pourrait être à la hauteur de l'amour Dont vous ne cessez de me combler. Je vous dédie ce modeste travail en témoignage de mon profond amour. Que dieu, le tout puissant vous procure bonne santé et longue vie et tout le bonheur.

A mon encadreur Mr. Salim Bitam

Je ne peux pas oublier vos efforts, disponibilité, gentillesse, vos conseils qui ont été des pièces d'or pour accomplir notre projet de master. Aucune dédicace ne saurait exprimer l'estime, le dévouement et le respect que j'ai toujours pour vous. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon frère Abdelhamid Abdelmouhaimen, mon cousin Takkie et ma cousine Khaoula leksouri

A mes chères ami(e)s et collègues

qui m'ont encouragé et m'ont aidé pour mener à bien ce projet de master, Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des frères, sœurs et des amis sur qui je peux compter. En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

Roufaida Halimi.

Résumé

De nos jours, le transport de la route est devenu une composante essentielle de la vie pour la majorité des gens. Mais malheureusement, les routes sont dangereuses et elles sont accompagnées d'un certain nombre de problèmes (accidents, pollution, embouteillages, etc.). Chaque jour, l'humanité perd des milliers de personnes sur les routes pendant qu'ils se rendent à travailler, à étudier ou même à se distraire. Cette horrible situation a poussé les chercheurs, les constructeurs des automobiles et les gouvernements publics à améliorer la sécurité de nos systèmes de transport et des technologies de communication dans le but d'offrir des routes plus sûres et une conduite plus paisible pour les usagers de la route.

La nouvelle ère de l'Internet des Objets conduit à l'évolution des réseaux ad hoc conventionnels de véhicules vers l'Internet des véhicules (IoV). Cette nouvelle ère de transport émerge comme une plateforme indispensable pour collecter efficacement des données véhiculaires afin d'avoir des données plus sûrs, complètes et efficaces qui servent non seulement à améliorer la sécurité routière en réduisant les risques de la route mais aussi à la gestion de trafic par le contrôle de congestion pour avoir un trafic fluide.

Dans ce projet de master, nous nous intéressons à la collecte de données dans l'internet des véhicules qui a pour objectif l'amélioration de la sécurité et la gestion du trafic routier. Nous proposons dans ce travail, un nouveau protocole de collecte et de dissémination de donnée nommé : Data Aggregation AODV (DAG_AODV) qui s'agit d'une amélioration du protocole AODV. DAG_AODV sert à collecter et disséminer les données dans l'internet des véhicules.

Après la simulation de notre protocole DAG_AODV en utilisant le simulateur NS-2, les résultats obtenus ont montré que DAG_AODV offre de bons résultats en termes de débit, taux de paquets reçus avec succès et le délai de bout en bout en le comparant avec AODV.

Mots clés : STI, IoT, IoV, collecte de données, DAG_AODV.

Sommaire :

Résumé.....	i
Sommaire	ii
Liste des figures.....	iv
Liste des tableaux.....	v
Introduction générale.....	1

Chapitre 01 : Introduction à la collecte de données dans l'internet des véhicules

1.1 Introduction.....	3
1.2 Définitions	3
1.2.1 Système de transport intelligent	3
1.2.2 Internet des objets	5
1.2.3 Internet des véhicules	8
1.2.4 La collecte de données.....	10
1.3 Problématique de la collecte de données dans l'internet des véhicules.....	11
1.4 Motivations	12
1.5 Conclusion	13

Chapitre 02 : La collecte de données dans l'internet des véhicules : Etat de l'art

2.1 Introduction.....	14
2.2 CarStream.....	14
2.2.1 Architecture générale de CarStream.....	15
2.2.2 Paradigme de mise en œuvre	17
2.2.3 Evaluation de performance de CarStream	18
2.2.4 Discussion.....	18
2.3 La collecte de données sur les véhicules via des capteurs Smartphone.....	19
2.3.1 Conception de collecte de données véhiculaires	19
2.3.2 Collecte de données véhiculaires : mise en œuvre du prototype.....	22
2.3.3 Discussion.....	22
2.4 Internet de véhicules pour les systèmes de transport intelligents.....	23
2.4.1 Défis de la mise en œuvre de l'IoV et la collecte de données	23
2.4.2 Importance de l'utilisation de la mise en œuvre basée sur le Cloud	23
2.4.3 Modèle proposé	24

2.4.4 Discussion.....	25
2.5 Conclusion	26

Chapitre 03 : Conception du DAG_AODV

3.1 Introduction.....	27
3.2 Conception générale de DAG_AODV.....	27
3.3 Conception détaillée de DAG_AODV	29
3.3.1 Débuter une collecte	30
3.3.2 Traitement et analyse de données	30
3.3.3 Le routage de données collectées	30
3.4 Conclusion	35

Chapitre 04 : Implémentation et résultats

4.1 Introduction.....	36
4.2 La simulation	36
4.3 Les outils de simulation de DAG_AODV	36
4.3.1 NS-2 (Network Simulator)	36
4.3.2 OpenStreetMap	38
4.3.3 SUMO (simulation of Urban Mobility).....	38
4.4 Evaluation de performance (Métriques considérées).....	38
4.4.1 Débit moyen.....	38
4.4.2 Taux de paquets reçus avec succès	39
4.4.3 Délai moyen de bout en bout	39
4.5 Implémentation du protocole DAG_AODV	39
4.5.1 Description et pseudo code de DAG_AODV	39
4.5.2 Configuration et paramètres de simulation.....	43
4.6 Résultats et discussion	45
4.7 Conclusion	48
<i>Conclusion générale</i>	49
<i>Bibliographie</i>	50

Table des figures

Figure 1.1 : Systèmes de transport intelligents.....	4
Figure 1.2 : les services dans les STI.....	5
Figure 1.3 : Les appareils connectant à l’IoT.....	6
Figure 1.4 : Un environnement d’internet des objets avec divers applications	6
Figure 1.5 : L’apparition de l’IoT	7
Figure 1.6 : Les communications dans l’IoV	8
Figure 1.7 : modèle d’interaction dans l’IoV	9
Figure 2.1 : Les données traitées dans CarStream	15
Figure 2.1 : L’architecture du système CarStream.....	16
Figure 2.3 : Un aperçu de CarStream.....	17
Figure 2.4 : Le processus de collecte, de stockage, de visualisation des données de capteur.	21
Figure 2.5 : Conception d’obdlib	21
Figure 2.6 : Modèle de l’IoV proposé pur la collecte et l’analyse de données	25
Figure 3.1 : Un environnement IoV	28
Figure 3.2 : Conception générale de DAG_AODV	29
Figure 3.3 : le processus de collecte de données par DAG_AODV	31
Figure 3.4 : File d’attente prioritaire	32
Figure 3.5 : File d’attente prioritaire pour les données collectées d’urgence	32
Figure 3.6 : Architecture de nœud mobile multi canal	34
Figure 4.1 : Architecture de base de NS-2	37
Figure 4.2 : Visualisation SUMO ville de Biskra Rue.....	44
Figure 4.3 : Visualisation OpenStreetMap.....	44
Figure 4.4 : Visualisation NAM.....	45
Figure 4.5 : Débit moyen	46
Figure 4.6 : Taux de paquets reçus avec succès.....	46
Figure 4.7 : délai de bout en bout.....	47

Liste des tableaux

Tableau 4.1 : Les paramètres de simulation..... 43

Introduction générale

La révolution de l'« Internet des objets » apportera de nombreuses améliorations à tous les secteurs de l'industrie, mais s'il y a un domaine dans lequel un bon nombre des avantages de cette nouvelle technologie se concrétiseront dans les prochaines années, c'est bien celui des transports.

Le vingtième siècle a connu un très fort développement dans les transports, en particulier dans les transports terrestres et aériens. L'acquisition de l'automobile dans les foyers et chez les professionnels, l'augmentation des transports de marchandises par poids lourd ainsi que le développement des transports en commun engendrent un trafic fort dans tous les pays.

De plus, l'urbanisation, la concentration des populations dans les villes et l'accroissement de la mobilité des personnes mènent souvent vers une congestion du trafic. Il y a plusieurs années de cela, les autorités -à l'origine principalement anglaises et américaines- ont commencé à chercher des solutions à ce problème, ce qui a poussé vers la naissance du concept des « systèmes de transport intelligents ».

L'Internet des objets (IoT) accélère considérablement l'allure à laquelle apparaissent les innovations dans le secteur des transports, et en particulier en ce qui concerne les véhicules que nous conduisons tous les jours. Aujourd'hui, de nombreux véhicules sont équipés de systèmes connectés qui permettent aux conducteurs d'écouter la radio par satellite, de regarder des vidéos en streaming, de consulter les applications de leurs Smartphones, de définir leurs itinéraires, d'utiliser des services d'assistance routière, d'ouvrir les portières à distance et de repérer les places de parking libres. Ces véhicules connectés forment ce qu'on appelle l'Internet des Véhicules (Internet of Vehicles -IoV-)

Selon le Cabinet de conseil Gartner, à l'horizon 2020, près de 250 millions de voitures seront connectées à l'Internet, pour un marché de l'automobile que PricewaterhouseCoopers évalue à USD 149 milliards. Il est donc clair que l'Internet des véhicules exigera de très nombreuses normes, et que l'ISO/TC 204, Systèmes intelligents de transport, y jouera un rôle de premier plan [1].

Avec l'augmentation continue de la population urbaine, la gestion du trafic est devenue un grand problème dans notre vie d'aujourd'hui. Le concept de l'IoV offre de nombreuses nouvelles opportunités et applications et met à disposition des conducteurs, des sociétés et des entreprises divers avantages. Puisque les véhicules sont de plus en plus connectés à l'Internet, une quantité de masse de données se génèrent rapidement. Alors que ces données provenant de divers capteurs d'un même véhicule fournissent souvent une connaissance imparfaite, tronquée ou bien incomplète de l'environnement relatif à la route, il est nécessaire de confronter les informations issues de plusieurs véhicules. En d'autres termes, en collectant, échangeant et en traitant les informations émanant de diverses automobiles, l'information devient plus précise et plus pertinente ce qui permet d'avoir un degré de confiance beaucoup plus élevé. Dans ce contexte, il est donc nécessaire de combiner l'envoi des données avec un

processus de collecte car elle est indispensable pour compléter la perception d'un seul véhicule sur son environnement. La collecte de données permet de réduire les risques de la route, remédier les problèmes de congestion routière, offrir un confort aux conducteurs et aux passagers lors de leurs déplacements, et elle sert d'améliorer la sécurité de tous.

Dans ce projet de master, nous étudions le problème de la collecte de données au sein de l'internet des véhicules afin de pouvoir améliorer la sécurité de la route et satisfaire les exigences des usagers de la route. Pour ce faire nous proposons une amélioration du protocole AODV par la proposition d'un nouveau protocole appelé : Data Agregation AODV (DAG_AODV) ; c'est un protocole destiné à collecter et disséminer les données dans un environnement IoV pour renforcer les services routiers.

Notre mémoire est organisé en quatre chapitres avec une introduction générale et une conclusion. Dans le premier chapitre, nous présentons une vue générale concernant notre thème, les problématiques de la collecte de données dans l'IoV ainsi les motivations. A travers le deuxième chapitre, nous traitons quelques travaux connexes. Le chapitre trois et quatre expliquent la conception et l'implémentation de notre solution.

Chapitre 01 : Introduction à la collecte de données dans l'internet des véhicules.

L'Internet des objets transforme intelligemment divers domaines de recherche existants en de nouveaux thèmes. S'appuyant sur le «transport intelligent», Internet of Vehicles (IoV) évolue comme un nouveau thème de recherche et de développement à partir des réseaux ad-hoc véhiculaires (VANET). L'IoV renforce la prévention routière, et permet de munir nos voitures et nos routes de capacités permettant de rendre la route plus sûre (les informations sur le trafic, les accidents, les dangers, les déviations possibles etc.). En agrégeant, traitant et exploitant différents types de données qui proviennent des véhicules connectés. Dans ce chapitre, nous allons initier notre projet qui est intitulé « La collecte de données dans l'internet des véhicules », commençant par des définitions et passant par les problématiques et les motivations de la collecte de données dans l'IoV.

1.2. Définitions

1.2.1. Système de transport intelligent

Les systèmes de transport intelligents sont apparus au début des années 1980 dans l'exploitation des transports, mais le terme lui-même n'est apparu que dans les années 1990, dans le contexte des transports américains dominé par l'automobile. Les laboratoires de recherche qui avaient travaillé au programme « guerre des étoiles » (lancé par Reagan en 1983 et abandonné par Clinton en 1993) ont pensé que les technologies mises au point pour explorer l'espace permettraient de remplacer des voitures par des robots. Le concept de « route automatique » qui a fait l'objet de démonstrations en Californie (San Diego 1997) et au Japon n'a pas pu décoller du fait notamment des questions de responsabilité qui n'avaient pas été anticipées. Il a cependant traversé l'Atlantique sous l'appellation « route intelligente » et « voitures communicantes » puis « transport intelligent » en France [2].

Les systèmes de transport intelligents (en anglais Intelligent Transportation Systems (ITS)) désignent les applications des nouvelles technologies de l'information et de la communication au domaine des transports. On les dit "Intelligents" parce que leur développement repose sur des fonctions généralement associées à l'intelligence : capacités sensorielles, mémoire, communication, collecte et traitement de l'information et comportement adaptatif [3] [4]. La figure 1.1 présente un système de transport intelligent.

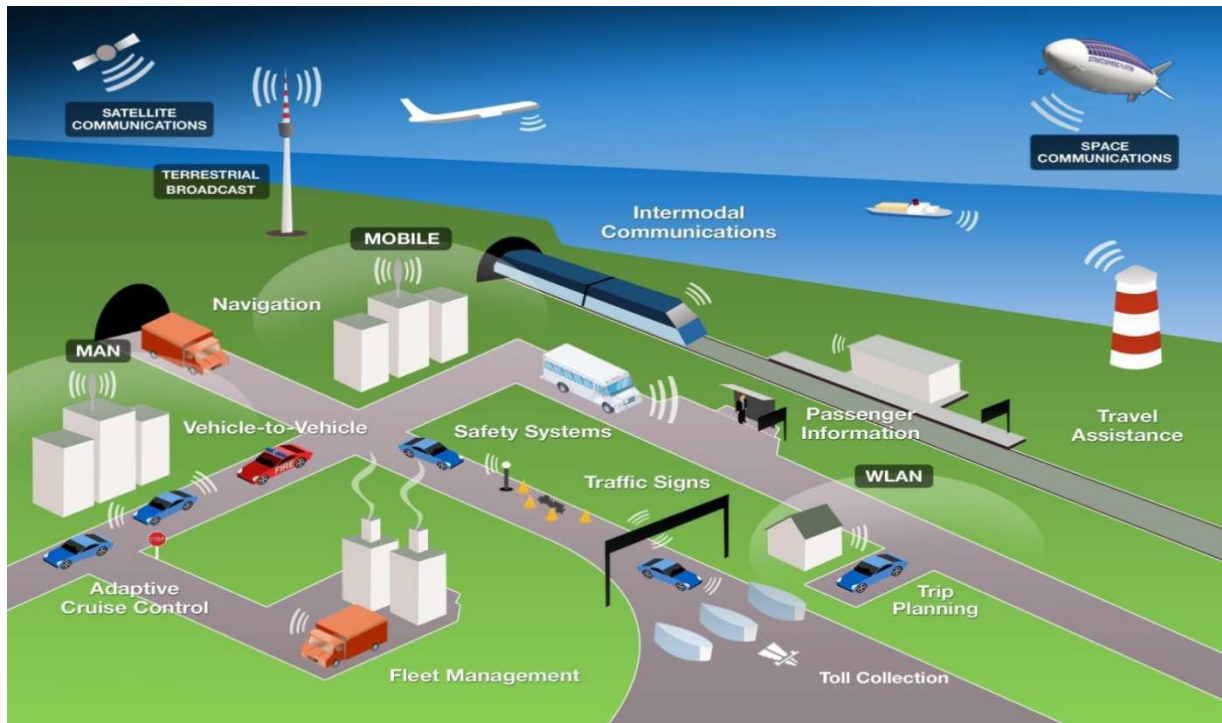


Figure 1.1 Systèmes de transports intelligents.

Ce nouvel ère de transport (STI) qui est une technologie, une application ou une plateforme comprend une vaste gamme de services et de solutions technologiques qui fournissent et gèrent l'information pour améliorer la sécurité, l'efficacité et la performance de notre réseau de transport. Parmi ces services, on trouve : des capteurs pour détecter la congestion et surveiller le flux de trafic, un ensemble de caméras de vidéosurveillance qui détectent les stalles, les collisions de véhicules et encore des lieux de congestion. Alors la qualité de transport est améliorée grâce à la combinaison de l'informatique et les télécommunications.

Les STI favorisent, pour l'organisation des transports du quotidien, la qualité de service et de confort pour l'utilisateur, une meilleure coordination pour la gestion de la circulation ainsi qu'une rationalisation de l'utilisation des réseaux. Ils peuvent également faciliter le report vers des moyens de transport plus économiques en temps, en coût ou en énergie [5]. La figure 1.2 montre les différents services offerts par les STI.

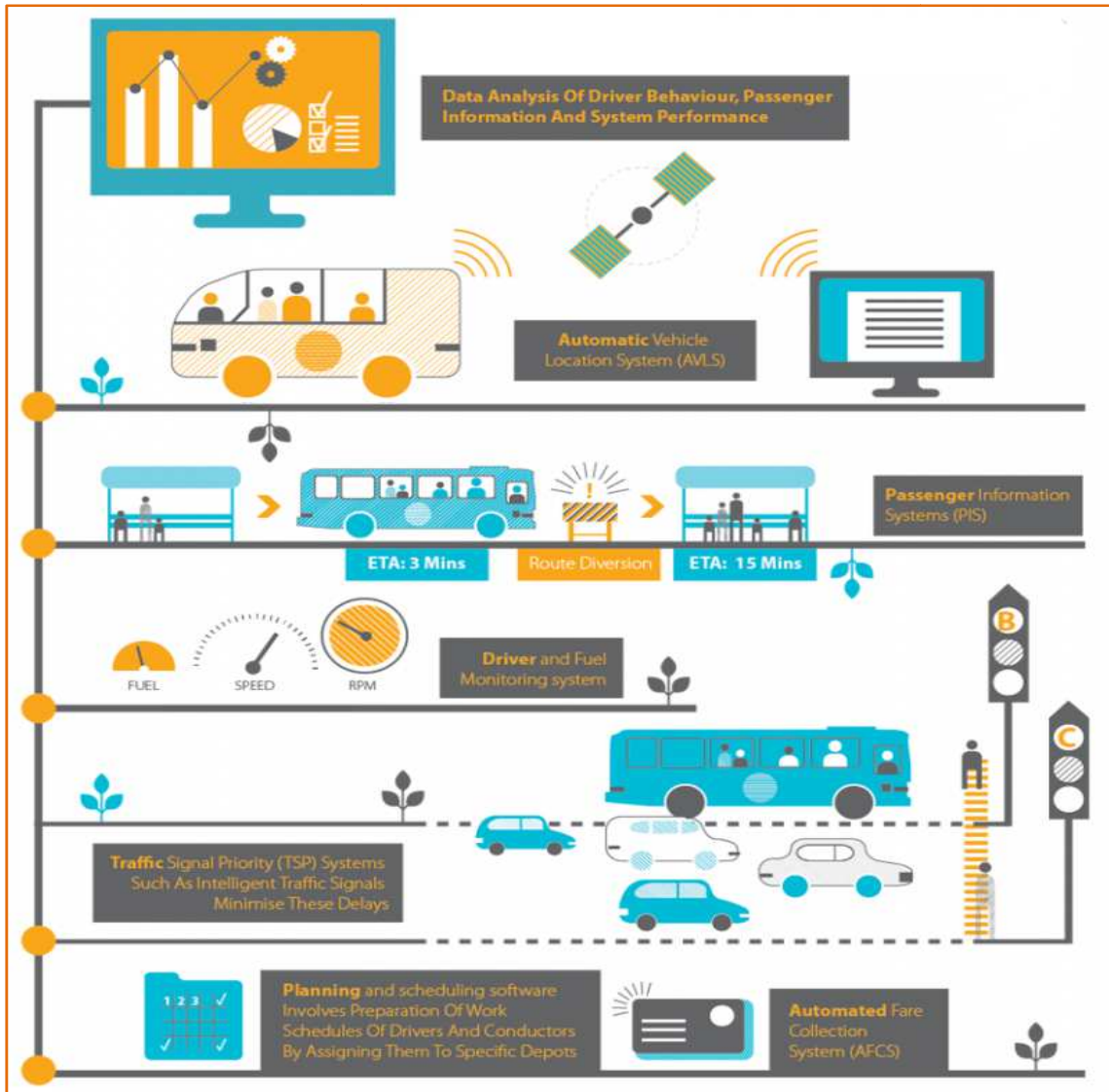


Figure 1.2 Les services dans les STI.

1.2.2. Internet des objets

L'Internet des choses, abrégé IoT vue sa formulation anglaise (Internet of Things), est un néologisme utilisé pour la première fois en 1999 par Kevin Ashton, cofondateur et directeur de l'Auto-ID Center qui référence tous ces appareils (en dehors des ordinateurs et des Smartphones) connectés à Internet. Qu'il s'agisse de voitures, de bracelets de fitness, de réfrigérateurs, de systèmes électriques ou d'autres choses encore, tous sont connectés à Internet et vous permettent d'échanger des données grâce aux capteurs [6]. Au fur et à mesure que l'Internet des Objets augmentera au cours des prochaines années, d'autres appareils rejoindront cette liste. La figure 1.3 donne une vue sur les appareils connectant à l'internet.

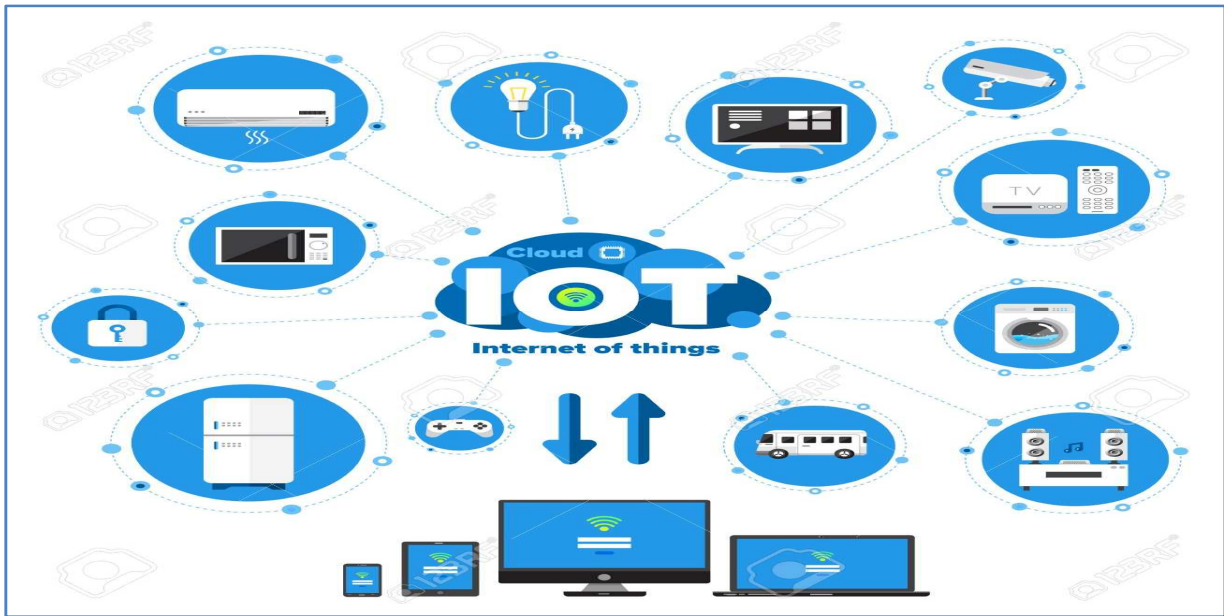


Figure 1.3 Les appareils connectant à l'IoT.

Cette nouvelle technologie a pour objectif de rendre le monde réel plus intelligent grâce à la connexion des objets qui sont capables de communiquer sans l'intervention d'un être humain. À ce jour, des milliards de dispositifs physiques à travers le monde sont connectés à Internet, collectant et partageant de divers types de données. Alors, l'IoT est entrain de fusionner le monde numérique avec le physique. La figure 1.4 présente les différentes applications qui favorisent notre vie quotidienne grâce à l'IoT.

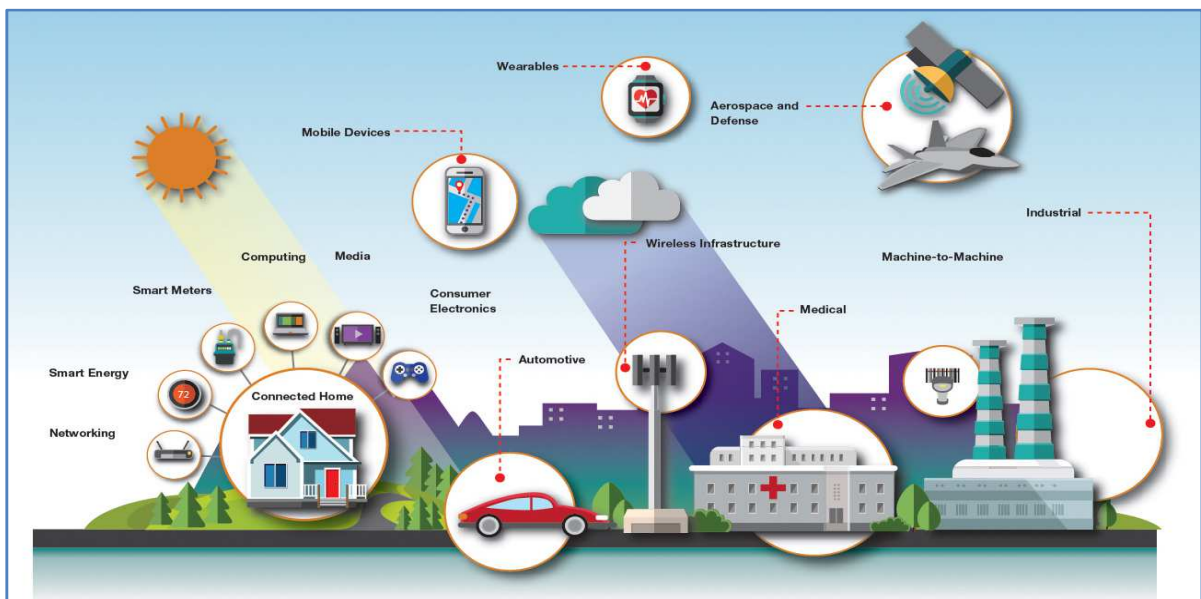
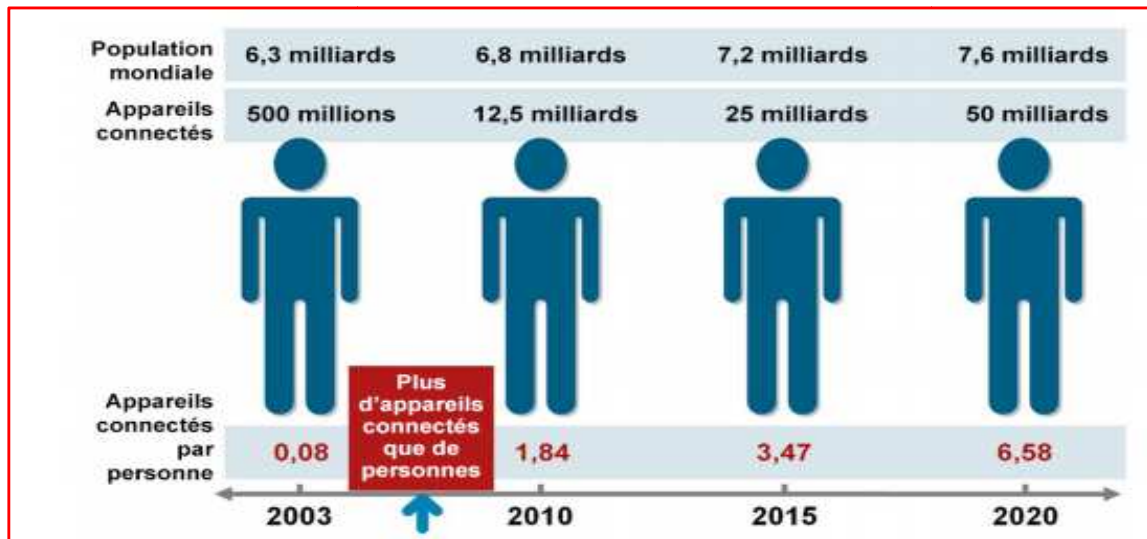


Figure 1.4 un environnement d'Internet des objets avec divers applications.

Pour reprendre les termes de CISCO, entreprise américaine spécialisée dans les réseaux : "l'IoT c'est le moment où le nombre "de choses et d'objets" connectés à Internet est supérieur au nombre de personnes (en 2010, il y avait 12,5 milliards d'appareils connectés à Internet alors qu'il y avait 6,8 millions d'habitants sur Terre). Nous pouvons donc considérer l'année 2010 comme l'année de naissance de l'IoT [6]. La figure 1.5 suivante donne une vue concernant l'apparition de l'IoT.



La figure 1.5L'apparition de l'IoT [8].

Le groupe de travail Internet of Things Global Standards Initiative (IoT- GSI), piloté par l'International Telecommunication Union (ITU), considère l'IoT comme « une infrastructure mondiale au service de la société de l'information » permettant « d'offrir des services évolués en interconnectant des objets (physiques et virtuels) grâce à l'interopérabilité de technologies de l'information et de la communication existantes ou en évolution »[7].

On considère un objet connecté n'importe quel équipement possédant les sept attributs suivants :

- Capteurs
- Connectivité à Internet
- Processeurs
- Efficacité énergétique
- Coût optimisé
- Fiabilité
- Sécurité

1.2.3 Internet des véhicules

L'Internet des objets (IoT) est un réseau mondial qui connecte des objets intelligents et leur permet de communiquer les uns avec les autres. Chaque fois que ces objets intelligents connectés sur Internet sont exclusivement des véhicules, alors l'IoT devient Internet of Vehicles (IoV). Ainsi, IoV est une application étendue de l'IoT dans le transport intelligent [9].

L'Internet des véhicules (IoV) se compose de véhicules qu'on peut les considérer comme une plateforme de capteurs, capables d'absorber des informations provenant de l'environnement, des autres véhicules ou bien leurs conducteurs. Ces informations générées sont utilisées pour la sécurité de la navigation, le contrôle de la pollution et la gestion du trafic.

Dans un environnement IoV, Il existe certains types de communications véhiculaires qui sont les suivants : véhicule à véhicule (V2V), de véhicule à route (V2R), de véhicule à infrastructure, de véhicules à piétons (V2P) et de véhicules à capteurs (V2S), ce qui forme alors un système de communication mobile et dynamique. Ces différentes communications sont illustrées dans la figure 1.6 ci-joint et la figure 1.7 présente le modèle d'interaction pour l'IoV.

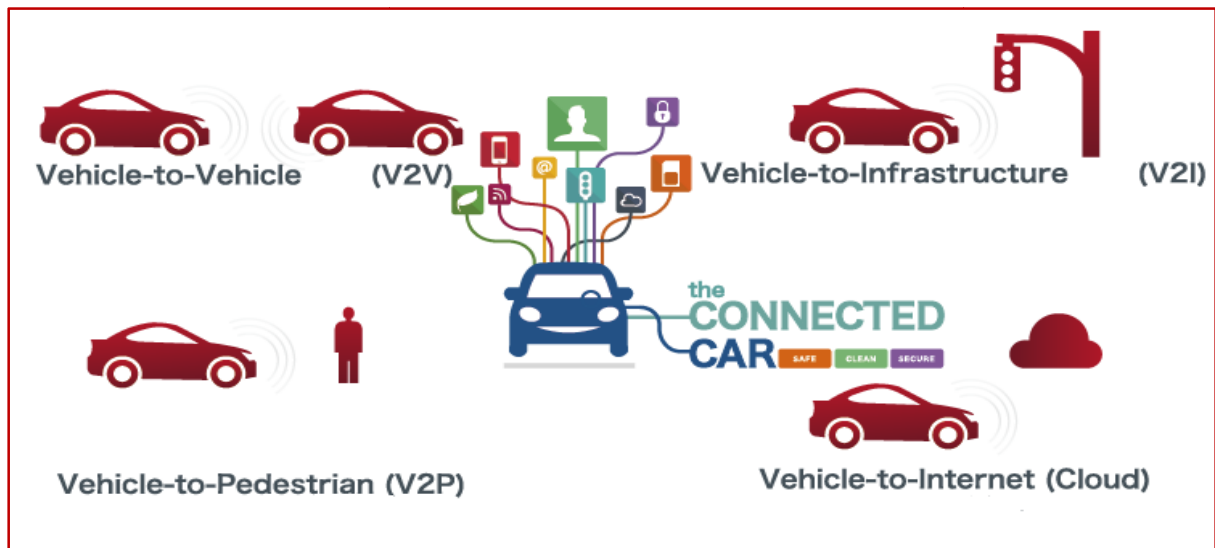


Figure 1.6 Les communications dans l'IoV.

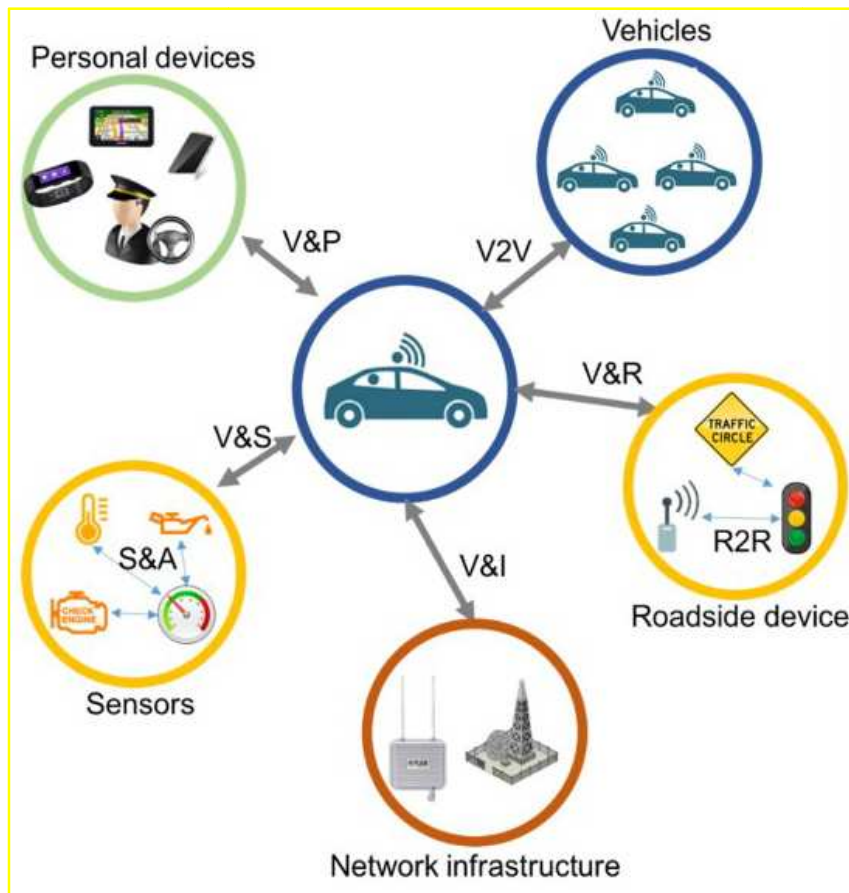


Figure 1.7 modèle d'interaction dans l'IoV [10].

A. L'architecture de l'IoV

Actuellement, il n'existe pas d'architecture spécifique de l'IoV en raison de la compréhension différente existante de l'IoV [11]. Basé sur l'interaction de différentes technologies dans l'environnement IoV, des chercheurs dans [12] ont identifié une architecture à trois niveaux. Cette architecture comprend les couches suivantes :

i. Couche de perception

C'est le premier niveau qui contient tous les capteurs du véhicule qui sont capables de recueillir des données provenant de l'environnement et détectent des événements spécifiques tels que les habitudes de conduite et la situation des véhicules, les conditions environnementales.

ii. Couche réseau

Cette couche intervient au deuxième niveau, elle est la couche de communication. Alors, elle prend en charge les différents modes de communication sans fil tels que véhicule à véhicule (V2V), véhicule à infrastructure (V2I), véhicule à piéton (V2P) et véhicule à capteur

(V2S). La couche de communication permet d'assurer une connectivité transparente aux réseaux existants et émergents (on site GSM, Wi-Fi, LTE, Bluetooth, 802.15.4).

iii. Couche Application

La Couche application (troisième couche) comprend des outils de statistiques, de stockage et de traitement constituant l'intelligence de l'IoV. A ce niveau, L'analyse, le traitement et la prise de décision sur différentes situations de risque (telles que la congestion de la circulation, les conditions routières dangereuses) sont prises en charge. Cette couche a eu comme objectif de pouvoir prendre une décision unifiée basée sur la fusion d'informations obtenues à partir de différents systèmes et technologies (tels que Big Data, Wireless Sensor Network (WSN), Cloud computing, etc.).

B. Les protocoles de l'IoV

La pile de protocole vise à accomplir les exigences fonctionnelles de chaque couche identifiée dans l'architecture [13]. La pile de protocoles possède trois plans qui comprennent la sécurité, l'opération et la gestion. Les protocoles appropriés sont identifiés pour différentes couches et différents plans de l'architecture de l'IoV.

L'un des aspects de recherche le plus important dans l'Internet de Véhicule est ses protocoles de routage. Plusieurs protocoles de routage classiques tels que le vecteur de distance à séquence de destination (DSDV), le routage dynamique de source (DSR) et le vecteur de distance ad hoc sur demande (AODV) proviennent de l'étude MANET.

1.2.4 La collecte de données

Il s'agit de recueillir un maximum de données en temps réel (circulation, accidents, parking, travaux, ligne de bus...) et de faire parvenir cette information aux utilisateurs pour les aider à choisir leur mode de transport et aussi leur parcours.

La collecte de données débute à l'aide de capteurs, de boucles électromagnétiques enfouies dans la chaussée, et de caméras disposées sur le réseau routier. Ces derniers surveillent et d'analysent l'état du trafic en flux continu. Les informations peuvent également provenir de la police, la gendarmerie, les pompiers, la voirie ou encore des automobilistes et des routiers qui avertissent spontanément les secours en cas d'incident.

Les systèmes de transport intelligents récoltent quotidiennement une énorme quantité d'informations sur le comportement des usagers et les conditions opératoires du réseau de transport. Ces informations peuvent être regroupées en des classes :

- a. **Sécurité** : sécurité publique des voyages, amélioration de la sécurité pour les usagers, jonctions intelligentes, etc.

- b. **Gestion du trafic** : contrôle du trafic, gestion des accidents, gestion des flux, gestion de maintenance des infrastructures, planification des infrastructures, etc.

- c. **Véhicule** : amélioration de la couverture visuelle, fonctionnement automatisé du véhicule, évitement des collisions (latérales et longitudinales), sécurité, maintenance du véhicule, etc.
- d. **Gestion des urgences** : avis d'urgence et sécurité personnelle des voyageurs, gestion des véhicules d'urgence, gestion du transport des matériaux dangereux et notification d'incidents, etc.
- e. **Paiement électronique** : opérations financières électroniques (péages).
- f. **Information pour les voyageurs** : informations de préparation au voyage, informations durant le voyage pour le conducteur, information durant le voyage dans les transports publics, informations routières, localisations, etc.

1.3. Problématique de la collecte de données dans l'IoV

Le nombre de transports ne cesse d'augmenter ces dernières années, conduisant ainsi à une saturation du réseau routier. Cette saturation représente la cause principale d'encombrement, d'accidents, et de pollution. Ces problèmes ont des conséquences majeures sur l'économie du pays et la vie quotidienne du citoyen.

Une solution possible à ce problème consiste à gérer d'une manière efficace le trafic routier, en se basant sur les systèmes de transport intelligents(STI). En tant qu'extension de l'Internet des Objets (IoT), Internet of Vehicles (IoV) permet une gestion unifiée dans le domaine des transports intelligents. L'IoV à grande échelle permet de recueillir des données à partir de différents lieux et de divers attributs. Cette collecte d'un grand nombre d'informations sur le trafic contribue à la gestion intelligente et l'optimisation de la route.

La collecte de données et par le biais des STI a pour objectif de répondre à des problèmes de société ciblés sur l'utilisation des transports. Les STI sont donc ancrés dans un contexte d'amélioration des systèmes autant pour les usagers, que pour les conducteurs et les gestionnaires. Ainsi, les données partagées via l'IoV sont présents principalement dans la gestion de congestion du trafic routier et dans le développement de nouvelles technologies de l'information embarquées dans les véhicules. Alors, les données collectées à partir des véhicules connectés à internet ne participent pas seulement au gain de temps, mais aussi à la minimisation des coûts principalement sur les énergies à utiliser, et à la croissance de productivité mais aussi de sauver des vies. La mise en place d'un système de collecte de donnée dans l'Internet des véhicules est encouragée, entre autres, pour réduire les risques de fautes humaines et ainsi améliorer la sécurité de tous.

L'utilisation d'un tel système permet aux organismes publics, constructeurs et aux sociétés de transport d'améliorer :

- La sécurité routière en réduisant nombre d'accidents.

- L'efficacité des transports routiers en régulant le volume de trafic routier ou les ralentissements aux péages.
- L'organisation et de gérer la circulation d'une manière sûre et efficace.

La quantité de données qui seront générées par les véhicules connectés paraît tout simplement stupéfiante. Avec des millions de véhicules offrant seulement quelques octets de données toutes les quelques minutes, la collecte sera de plusieurs péta-octets de données chaque jour. L'augmentation sera spectaculaire quand ils commenceront à recueillir des données à partir de plusieurs capteurs par véhicule et quand ils commenceront la collecte en continu. Le problème n'est pas de savoir comment recueillir suffisamment de données mais de savoir comment gérer et donner un sens à toutes ces données [14].

D'après notre projet nous cherchons à concevoir puis développer un système de collecte de données dans le contexte des IoV afin de pouvoir :

- Collecter et fusionner de données.
- Diffuser ces données d'une manière efficace et sûre.
- Saisir et à exploiter des informations.
- Augmenter la sécurité routière.
- Améliorer le flux du trafic.
- Préserver notre environnement.

1.4 Motivations

La révolution numérique, avec le développement de l'Internet mobile, des Smartphones, de l'open data, du big data, et l'évolution des technologies de communication comme le LTE et la 4G, modifie profondément la chaîne de valeur des métiers traditionnels du transport [15].

Imaginons un monde des transports où la circulation serait redirigée en quelques secondes après un accident, où les limites de vitesse seraient modulées automatiquement en fonction de la densité de la circulation, où les automobilistes auraient accès à des voies supplémentaires avant que les bouchons ne se forment, imaginons qu'un véhicule puisse informer si un autre conducteur est sur le point de refuser la priorité à droite... De nombreuses collisions seraient ainsi évitées, grâce à des dispositifs permettant aux véhicules de communiquer et où les voitures pourraient non seulement « parler », mais également suggérer des solutions sur la base d'une donnée collectée et partagée.

Il ne s'agit pas de fantasmes futuristes, mais bien de situations concrètes et de possibilités réalistes grâce à l'évolution des systèmes de transport intelligents (STI), qui transforment nos réseaux de transport. Les données trafic permettent d'afficher les bouchons, les accidents de circulation ou les travaux dans n'importe quelle ville ou rue, ainsi que les ralentissements automobiles ou les conditions météo (neige, pluie, verglas).

Cette évolution de transport fait émerger de nouveaux usages et de nouvelles solutions pour gérer les flux urbains, la mobilité des voyageurs et des marchandises, le traitement de l'information en situation normale et perturbée ainsi que les dispositifs de paiement. Le développement de l'internet des véhicules et de la géo-localisation fait évoluer en profondeur tous les modes de transport et permet l'apparition de nouveaux services de mobilité comme le covoiturage, l'auto-partage, la gestion de flotte de véhicules, impactant fortement les modèles économiques traditionnels des acteurs. La donnée numérique collectée transforme en profondeur la ville de demain, laboratoire d'innovation pour expérimenter de nouveaux dispositifs (transports multimodaux, carrefours intelligents, parkings dynamiques, voitures connectées et autonomes...) au service d'une mobilité plus harmonieuse pour le citoyen et l'environnement.

Par conséquent, ce projet contribue à améliorer la sécurité routière et à servir les usagers de la route en se basant sur la collecte de données issues des différents nœuds de l'internet des véhicules et des systèmes de transport intelligents.

Ce projet montre pourquoi les systèmes de collecte de données sont importants en pratique dans la sécurité routière et durant la conduite. À titre d'exemple, les données relatives à la sécurité routière sont utilisées par divers acteurs tels que la police, les services de transport, les établissements de santé, les sociétés d'assurances, ainsi que par les décideurs et les intervenants. Les données sur les accidents de la circulation sont essentielles pour cerner les risques, définir des stratégies et des interventions pour y remédier, et évaluer l'incidence des interventions.

Le but principal derrière notre projet est la conception et le développement d'un système de collecte de donnée dans l'internet des véhicules. Il part du principe qu'une bonne gestion de la sécurité routière repose sur une approche systématique qui comprend la collecte, l'analyse, l'interprétation et l'application de bonnes données.

1.5 Conclusion

A travers ce premier chapitre, nous avons défini les concepts de base concernant l'internet des véhicules et la collecte de données. Nous avons aussi expliqué l'importance de la collecte de donnée dans l'IoV qui est apparue comme un nouveau paradigme inspiré de l'IoT et qui sert à améliorer la vie d'aujourd'hui en offrant de nouvelles opportunités à l'urbanisation.

Dans le prochain chapitre, Nous allons présenter et discuter quelques travaux connexes qui traitent le problème de la collecte de données dans l'IoV.

Chapitre 02

2. La collecte de données dans l'IoV : Etat de l'art

Ces dernières années, la technologie de l'Internet des objets (IoT) est apparue comme un important domaine de recherche et d'application. En tant que branche majeure de l'IoT, Internet des véhicules (IoV) a attiré l'attention de l'industrie et de la recherche.

IoV a été pendant des années l'une des applications les plus populaires dans le paradigme IoT. Mais peu de grands déploiements IoV existent, en raison du coût élevé de déploiement et de maintenance des voitures individuelles.

Pour mettre en œuvre une plateforme IoV, les exigences fonctionnelles de base devraient au moins inclure la collecte de données sur les véhicules, le stockage, l'analyse et les services. A travers ce chapitre nous allons discuter certaines technologies et des travaux connexes suggérés dans le domaine de l'internet des véhicules dans l'objectif de la collecte de données.

2.2 CarStream

Récemment, l'IoV basée sur le Cloud a bénéficié du développement rapide des technologies de réseaux mobiles et de Big Data. Une exigence de base dans ce scénario IoV basé sur le Cloud est de collecter et traiter le flux de données de gros véhicules en temps opportun.

Afin de satisfaire cette exigence de base, la conception d'un système IoV industriel à grande échelle qui prend en charge la collecte de données présente trois principaux défis. Premièrement, le système doit être hautement évolutif pour gérer la grande échelle des données. Contrairement à d'autres systèmes, les données de l'IoV sont des données massives donc une collecte de données suivie d'une agrégation et un traitement sont des processus indispensables. Deuxièmement, le système doit faire un compromis souhaitable entre les exigences de temps réel et de précision des services centrés sur les données recueillies lors du traitement de données à grande échelle de faible qualité et de faible densité de valeur. Troisièmement, le système doit être très fiable pour prendre en charge les services essentiels à la sécurité, tels que l'aide d'urgence. Une fois qu'un service est déployé, il doit fonctionner de manière continue et fiable tout en collectant, analysant et conservant les données capturées à partir des véhicules connectés. Une panne de service inattendue ou une perte d'une information collectée peuvent causer de graves dommages matériels ou même la perte de vie.

Il peut y avoir plusieurs solutions pour concevoir un système IoV qui permet de collecter les données provenant de différents véhicules.

Zhang et al. dans [16] ont proposé CarStream, un système de traitement de données industriel pour les services de voiture avec chauffeur dans l'IoV. CarStream a connecté plus de 30 000 véhicules dans plus de 60 villes en chine et dispose de multiples sources de données telles que la vitesse, les trajectoires, le nombre de tours par minute, l'essence restante, les

activités du conducteur (comme le démarrage d'un service, ramasser un passager), et les commandes de passagers. CarStream fournit plusieurs services en temps réel basés sur ces données collectées. En particulier, ils ont abordé le problème de l'évolutivité en équipant CarStream de la capacité de traitement distribué et de stockage de données.

CarStream est équipé de composants suivants :

- *Un sous-système de traitement de flux* : pour prétraiter les données, de sorte que les problèmes de faible qualité de données et de faible densité de valeur peuvent être résolus. Cette solution atteint des performances supérieures avec le coût de plus d'espace de stockage.
- *Un sous-système de mise en mémoire cache* : pour accélérer également les performances.
- *Un sous-système de surveillance* : afin de fournir une fiabilité et une facilité de gestion élevées. Il fournit une vue complète de l'exécution du système global, ce qui aide à détecter l'état de santé de CarStream en temps réel.

CarStream a été déployé et maintenu pendant trois ans dans l'utilisation industrielle. En général, il collecte, stocke et analyse les données téléchargées en termes de gestion des véhicules et des conducteurs.

La Figure 2.1 illustre les multiples données collectées et traitées dans CarStream au cours d'une journée. Ces données peuvent être grossièrement classées en quatre types : l'état du véhicule, l'ordre des passagers, l'activité du conducteur et les données de trajectoire.. Les données dans CarStream sont non seulement volumineuses mais aussi complètes [16].



Figure 2.1 Les données traitées dans CarStream [16].

2.2.1 Architecture générale de CarStream

L'architecture générale de CarStream comprend les différentes couches suivantes, comme décrit La figure 2.2.

i. Couche de bus de données

Cette couche fournit un concentrateur d'échange de données pour que les modules du système puissent connecter différentes tâches de calcul. Elle est implémentée avec une

plateforme de messagerie distribuée avec une capacité tolérante aux pannes, haute performance et évolutive horizontalement.

ii. Couche de traitement

Elle comprend deux parties :

- Le sous-système de traitement de flux en ligne est conçu pour fournir une fonctionnalité de prétraitement aux flux de manière à réduire la pression de requête de la base de données.
- Le sous-système de traitement par lots hors ligne est utilisé pour satisfaire les exigences d'analyse complexes qui impliquent davantage de données historiques.

iii. Couche de gestion des données

Cette dernière adopte une structure hétérogène construite avec plusieurs bases de données. Dans CarStream, les bases de données NOSQL et SQL sont toutes deux adoptées pour atteindre des performances élevées. Plus spécifiquement, les données les plus fréquemment utilisées sont stockées dans le sous-système de mise en cache en mémoire de sorte que les données peuvent être consultées et mises à jour fréquemment. Les choix de mise en cache en mémoire peuvent être multiples, comme Memcached [17] et Redis [18].

iv. Surveillance du système

La surveillance du système implique la surveillance des couches : infrastructure, plateforme informatique et application. La surveillance de l'infrastructure est utilisée pour afficher des statistiques en direct ou archivées couvrant des métriques telles que la charge moyenne du processeur. Pour la couche de plateforme informatique, diverses plateformes sont surveillées telles que le sous-système de traitement de flux et par lots, et les bases de données pour suivre l'état de fonctionnement du logiciel de plateforme. La surveillance de la couche d'application peut fournir des informations utiles pour la prévention ou la récupération des défaillances de service.

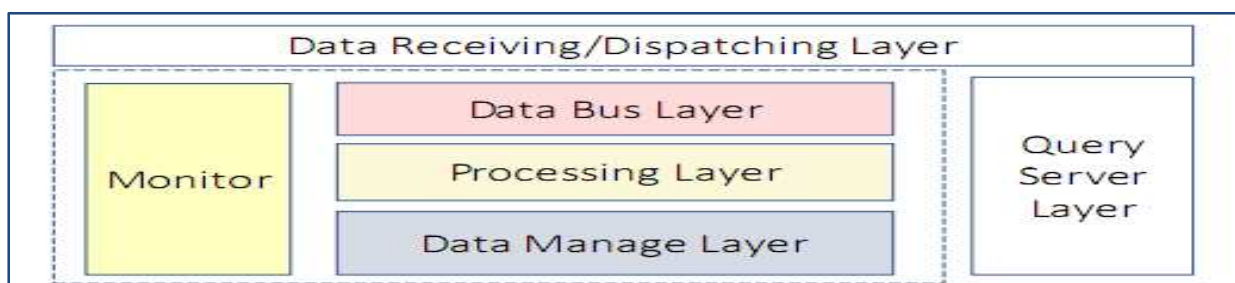


Figure 2.2 l'architecture du système de CarStream [16].

2.2.2 Paradigme de mise en œuvre

La structure globale de CarStream et les connexions entre différents modules sont illustrées par la Figure 2.3.

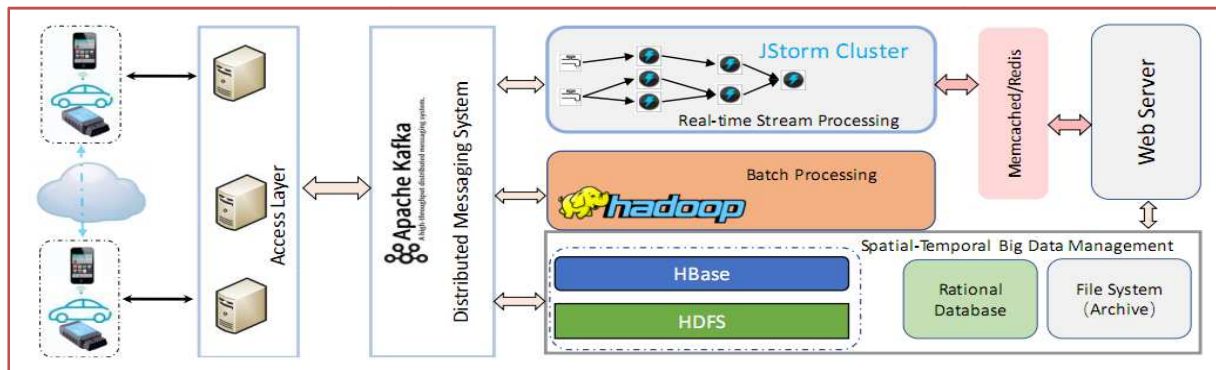


Figure 2.3 Un aperçu de CarStream [16].

a) *La mise en œuvre d'un sous-système de traitement*

Les développeurs de [16] ont adopté JStorm comme plateforme de traitement de flux de base dans CarStream. La raison principale est que, outre les caractéristiques de base de haute performance, distribuées, tolérantes aux pannes du traitement de flux, JStorm utilise également la topologie comme modèle de tâche de calcul, ce qui permet d'intégrer plusieurs applications par la fusion des tâches similaires ou diviser une tâche énorme en des fonctions plus petites.. CarStream utilise également Hadoop comme un sous-système de traitement hors ligne. En outre, la couche de bus de données de CarStream est implémentée avec KAFKA, l'une des plates-formes de messagerie distribuée les plus puissantes, grâce à sa facilité d'intégration.

b) *La mise en œuvre d'un sous-système de gestion de données en continu*

Pour construire un sous-système de gestion de données, Zhang et al. dans [16] ont adopté trois plateformes de stockage différentes dans CarStream, y compris la mise en cache en mémoire, la base de données relationnelle et le stockage de gros volumes de données NOSQL. En pratique, ils ont utilisé HBase comme plateforme de stockage d'archives pour les données brutes (telles que l'état et les trajectoires du véhicule, les données relatives aux activités du conducteur et les données relatives à l'ordre d'utilisation).

La surveillance par trois couches

Dans CarStream, l'infrastructure est surveillée avec Ganglia [19] qui fournit des fonctions puissantes afin de surveiller les performances du cluster à partir d'une vue complète. Pour la surveillance de plateforme informatique, le sous-système de surveillance intègre simplement les API de supervision fournies par chaque plateforme. Puisque La surveillance de la couche application est aussi importante, les développeurs de CarStream dans [16] ont autorisé la surveillance de la couche application. En pratique, Ces développeurs ont intégré

dans chaque application un module pour surveiller les paramètres qui peuvent refléter les performances et l'état de santé des applications. De plus, ils utilisent des algorithmes de détection d'anomalies sur ces paramètres collectés pour prédire les défaillances potentielles du système.

2.2.3 Évaluation de la performance de CarStream

Zhang et al. dans [16] effectuent une évaluation de performance de bout en bout de la procédure de traitement de flux entier de la réception des données à la requête de données. Le sous-système de traitement est déployé sur un cluster virtualisé avec 25 serveurs. HBase et HDFS sont déployés dans un environnement non virtualisé avec trois clusters. Les performances de CarStream ont été testées le à partir des aspects suivants :

Débit : CarStream peut facilement gérer le flux de données téléchargées par 30 000 véhicules. Le débit de traitement total de CarStream a atteint 240 000 instances de données par seconde [16].

Délai de bout en bout : Les concepteurs de [16] utilisent le suivi des véhicules en temps réel comme application de test. Dans cette application, les données sont collectées à partir des véhicules et traversent une série de plateformes (telle que la plateforme de tampon, de traitement de flux, de mémoire cache), puis finalement interrogées par le serveur web. Pour le délai de bout en bout, ils comparent les temps-tampons entre les données générées et les données en cours de traitement. Les résultats ont montré que le retard moyen est inférieur à 2 secondes.

Traitement de l'évolutivité : l'évolutivité du traitement de CarStream est testée en augmentant l'échelle de déploiement des sous-systèmes de mise en mémoire tampon et de traitement. Dans ce test, une clôture électronique est utilisée comme application de base. Les résultats de l'évaluation ont montré une évolutivité quasi linéaire du sous-système de traitement en fonction de nombre de serveurs.

2.2.4 Discussion

Nous avons décrit la conception de CarStream, un système industriel de traitement de données volumineuses pour l'IoV. CarStream offre une garantie de fiabilité élevée pour les services critiques en matière de sécurité dans le cadre de l'IoV.

Cependant, CarStream a été déployé et maintenu que dans l'utilisation industrielle. En outre, CarStream est basé sur une société de service de voitures avec chauffeur en Chine, nommée UCAR, cette dernière nécessite que la flotte appartienne à l'entreprise et les chauffeurs sont des employés de l'entreprise.

2.3 La collecte de données sur les véhicules via des capteurs Smartphone

Ces dernières décennies, les chercheurs ont mis la concentration sur plusieurs techniques de la collecte de données concernant les autoroutes et les zones urbaines. L'une des approches proposées par Reininger et al. dans [20] est l'utilisation des Smartphone pour la collecte de données à travers différentes véhicules connectées.

Les Smartphones et les tablettes modernes disposent de puissantes capacités de calcul, de communication et de détection [21]. En plus d'être capables de réaliser des tâches informatiques complexes et de communiquer sans fil les uns avec les autres, les Smartphones et les tablettes disposent d'un riche ensemble de capteurs embarqués, tels que des accéléromètres, des gyroscopes, des GPS et des caméras. Ces capteurs fournissent des informations précieuses lors de l'étude des besoins et des comportements des utilisateurs [21]. La puissance de communication des Smartphones pour fournir des données véhiculaires à grande échelle offre des opportunités uniques pour améliorer la qualité de vie dans les routes et les villes modernes intelligentes d'aujourd'hui.

Les auteurs dans [20] ont présenté un prototype de plateforme informatique mobile qui utilise des Smartphones et des tablettes pour accéder aux capteurs des véhicules sans compromettre la sécurité des appareils dans le but de collecter les données. Grâce à cette plateforme, des données sur les véhicules peuvent être collectées en temps réel, telles que la vitesse, le régime moteur, la consommation de carburant, la localisation GPS, etc. Ces données collectées peuvent ensuite être retournées et affichées sur un serveur distant pour l'analyse.

2.3.1 Conception de collecte de données véhiculaire

Cette collecte de données sur les véhicules se compose d'un backend considéré comme une unité centrale intelligente qui collecte, analyse et stock les données collectées par les véhicules et qui communique directement avec les capteurs embarqués et recueille les données des capteurs. Ce backend fait partie d'un banc d'essai distribué qui permet d'accéder aux données des capteurs du Smartphone de manière sécurisée et isolée de la performance [21]. En outre, les données peuvent être retournées à un serveur centralisé pour le stockage permanent des données.

i. Infrastructure de la collecte de données

➤ Banc de test de capteur de Smartphone distribué :

Cette plateforme est déployée sur un banc de test distribué [21] qui fournit un environnement de programmation efficace pour telle collecte de données. Ce banc d'essai utilise un sandbox léger [22] pour limiter la quantité de ressources de stockage, réseau, mémoire, batterie et processeur utilisées sur un périphérique mobile. Pour protéger les périphériques des utilisateurs finaux contre les attaques malveillantes, ces programmes en mode sandbox sont isolés des autres programmes sur le même périphérique.

➤ Vue d'ensemble de la collecte de données :

Le processus de déploiement d'une expérience sur le banc d'essai, puis la collecte de données à partir de capteurs de véhicules sur les dispositifs de l'utilisateur final est la suivante :

- Les propriétaires de périphériques installent une application testbed [23] sur leurs appareils. Et puisque les concepteurs dans [20] ont été intéressés par les données véhiculaires, alors le groupe cible des propriétaires d'appareils sont également propriétaires de véhicules.
- Ces propriétaires insèrent simplement leur capteur de diagnostic embarqué WiFi (OBD) [24] dans les ports OBD de leurs voitures (situés sous le volant),
- après, ils connectent leur Smartphone ou tablette au capteur, qui fonctionne également comme un point d'accès WiFi.

Un expérimentateur intéressé à obtenir des données sur les véhicules à partir des Smartphone et des tablettes des propriétaires d'appareils devra :

1) S'enregistrer auprès de cette infrastructure de banc d'essai et acquérir des dispositifs d'utilisateur final à des fins d'expérimentation.

2) Écrire un code de test en utilisant l'environnement de programmation sandbox [22].

3) Télécharger le code à un ensemble d'appareils Smartphone auxquels il ou elle a accès (obtenu à l'étape 1), en utilisant l'outil de gestion d'expérience de banc d'essai.

4) Démarrer ou arrêter l'expérience à tout moment ; l'expérimentateur peut également collecter des données de capteur en utilisant la fonction de journalisation du banc d'essai.

Le code de test s'exécute en arrière-plan de ces périphériques mobiles. Par conséquent, les propriétaires d'appareils n'ont pas à s'inquiéter de devoir interagir avec l'application via une interface ou d'être interrompus lorsqu'ils utilisent leur Smartphone. Parallèlement et discrètement, l'expérience en arrière-plan enregistre les données de capteur et peut transmettre les données collectées à un serveur distant. Sur le serveur distant, les données de capteur de véhicule collectées à partir de plusieurs périphériques d'utilisateur final sont stockées dans une base de données non relationnelle. La figure 2.4 résume le processus proposée par les chercheurs de [20]), pour collecter stocker et visualiser des données de capteur en utilisant des Smartphone et des tablettes.

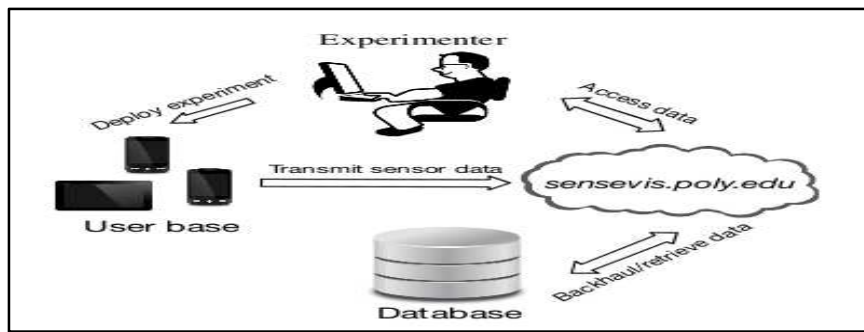


Figure 2.4 Le processus de collecte, de stockage, de visualisation des données de capteur [20].

ii. Collecte de données sur les véhicules

Les développeurs de [20] ont conçu une bibliothèque pour le sandbox qui fournit une interface pour communiquer avec un capteur OBD embarqué, qui sert également de point d'accès WiFi. Elle est nommée la bibliothèque de communication OBD obdlib [25]. Dans l'implémentation de prototype, le code de test peut se connecter au capteur OBD, similaire à la connexion à un point d'accès WiFi, et collecter les données du véhicule à partir de ce capteur via l'interface obdlib. Cette obdlib permet d'accéder aux données du capteur de véhicule telles que la vitesse, le régime moteur et la consommation de carburant. Ils ont mis en œuvre l'expérience de sorte que les données collectées soient mises en cache périodiquement sur le téléphone et transmises à un serveur distant chaque fois qu'une connexion Internet est disponible via le réseau cellulaire. La conception du protocole de communication entre le véhicule, le Smartphone et le serveur distant est illustrée à travers la figure 2.5.

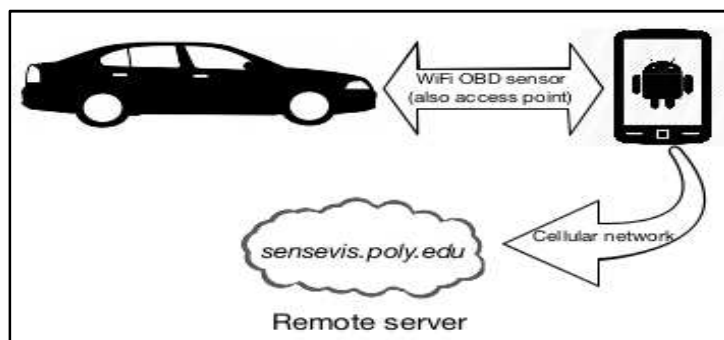


Figure 2.5 Conception d'obdlib [20].

iii. Backhaul de données et visualisation

Pour gérer le stockage des données de capteurs véhiculaires de plusieurs appareils Android de manière conviviale, les concepteurs de [20] ont déployé un site Web qui accepte les requêtes POST contenant les données de capteur collectées et souhaitées (et cryptées) et les stocke dans des comptes d'expérimentation privés. Les expérimentateurs, cependant, n'ont pas besoin de construire les requêtes ni de chiffrer eux-mêmes les données des capteurs, car ils fournissent une bibliothèque, appelée 'storesense', pour exécuter ces deux fonctions [26].

Par la suite, les expérimentateurs peuvent se connecter à leurs comptes sur ce site Web pour analyser, visualiser et interagir avec les données qu'ils collectent à partir des Smartphones des utilisateurs finaux distants.

2.3.2 Collecte de données véhiculaires : mise en œuvre du prototype

Afin de collecter les données sur les véhicules en utilisant cette plateforme prototype de [20], le capteur OBD-II ELM327 WiFi a été utilisé. Pour faciliter les communications entre les programmes en sandbox sur un Smartphone et le capteur OBD, ils ont développé la bibliothèque obdlib [25], qui implémente le protocole de communication OBD standard. Une connexion TCP entre le Smartphone et le capteur OBD via l'adresse IP par défaut de la passerelle WiFi attribuée au capteur OBD a été utilisée.

Compte tenu de la variété des capteurs sur un Smartphone, les mesures des capteurs se présentent sous différentes formes. En conséquence, il peut être très difficile de stocker des données de capteur dans une base de données relationnelle, qui a une structure de colonne fixe.

Pour ces raisons, les développeurs de [20] ont décidé d'utiliser une base de données non relationnelle, MongoDB [27], pour stocker les données des capteurs. MongoDB a une structure de type document binaire JSON (BSON). Cela fournit l'évolutivité qui est cruciale pour ce système et qui permet le stockage dynamique de nouveaux capteurs. Dans la base de données, les dispositifs individuels peuvent être distingués par un expérimentateur en utilisant un identifiant unique stocké. Ils ont choisi d'utiliser le numéro IMEI de l'appareil car il s'agit d'un identifiant à 15 chiffres unique au matériel de chaque appareil [28].

2.3.3 Discussion

Comme avec la plupart des technologies, les systèmes véhiculaires basés sur les Smartphones pourraient être une arme à double tranchant s'ils ne sont pas correctement utilisés. Par exemple, si les données collectées ne sont pas anonymisées, elles pourraient être liées à des utilisateurs individuels, ce qui poserait de sérieuses menaces de surveillance si les Smartphones étaient connectés à un serveur accessible par des tiers. Par conséquent, des mesures de protection doivent être prises pour assurer la sécurité des Smartphones embarqués et des systèmes de capteurs. Le système doit garantir que même si un pirate contourne l'environnement du capteur, aucun mal ne peut être causé aux Smartphones des utilisateurs et, plus important encore, aux systèmes de leurs véhicules. Ces facteurs doivent être pris en compte lors de la conception des systèmes télématiques véhiculaires pour Smartphone. Dans Ce prototype que nous avons décrit, le mécanisme de sandboxing prévient les attaques malveillantes mais reste moins sûrs.

Ce travail ne présente qu'un premier aperçu de la collecte de données sur les véhicules au moyen de capteurs de Smartphones. La sécurité et les problèmes de confidentialité qui entourent les réseaux de véhicules reste un défi à prendre en considération dans les futures applications.

2.4 Internet de véhicules pour un système transport intelligent

Dans [29] les chercheurs ont montré comment les véhicules avec capteurs et actionneurs peuvent absorber (recueillir) une grande quantité d'informations de l'environnement et fournir cette information utile pour aider à la sécurité de la navigation et la gestion efficace du trafic. En concentrant principalement sur la mise en œuvre de l'IoV, de la collecte de données en passant par des capteurs jusqu'à l'analyse et, à la fin, la prise des décisions basées sur ces derniers.

2.4.1 Défis de la mise en œuvre de l'IoV et la collecte de données

i. Énorme besoin de stockage

Une grande quantité de données stockées en péta-octets est nécessaire, car une grande quantité d'informations est collectée à l'aide de capteurs provenant d'un grand nombre de véhicules commerciaux. Avec l'avènement de l'implémentation de l'Internet des objets basée sur le Cloud, chaque périphérique sur Internet envoie et reçoit des informations depuis et vers le Cloud. Le défi de l'énorme besoin de stockage peut être satisfait par la catégorisation et l'archivage des données car il est nécessaire d'assurer une utilisation optimale des ressources en ne les chargeant pas gratuitement.

ii. L'analyse et le traitement efficace des données

La collecte de données à l'aide de divers capteurs, suivie d'un traitement et d'un stockage puis d'une analyse avec précision dans un environnement en temps réel constituera un grand défi pour la mise en œuvre d'IoV. Pour cette activité, l'analyse de données volumineuses est l'élément important qui a émergé avec divers logiciels de calcul. Des logiciels computationnels comme Apache Hadoop et Spark offrent des solutions rentables, tolérantes aux pannes et rapides aux problèmes impliquant de grands ensembles de données.

iii. La sécurité

Les données collectées doivent être suffisamment sécurisées. Toute modification ou perte de données peut entraîner des dommages sur toutes les choses connectées et / ou impliquées dans le traitement et l'analyse des données.

2.4.2 Importance de l'utilisation de la mise en œuvre basée sur un Cloud

Le Cloud dépend du partage de ressources au lieu d'utiliser des serveurs locaux ou des périphériques propres pour le calcul. Le Cloud est standardisé avec divers attributs tels qu'un large accès au réseau, une flexibilité rapide, un service calculé et un service à la demande. Ces attributs différencient l'utilisation du Cloud des autres services. Un grand nombre de services de Cloud peuvent être classés principalement en trois catégories. L'infrastructure en tant que service (IaaS), la plate-forme en tant que service (PaaS) et le logiciel en tant que service (SaaS).

2.4.3 Modèle proposé

i. Le rôle des capteurs :

Les capteurs joueront un rôle important dans la collecte de l'information. Différents capteurs ont été utilisés par Munjal dans [20], tels que des capteurs pour détecter les nids-de-poule (les trous de la route), la vitesse, les embouteillages et autres informations de détection. Chaque véhicule participera au transfert de l'information et toutes les informations recueillies seront transférées dans le Cloud.

ii. Traitement et analyse des données :

Les informations collectées ont été structurées avec certains mots-clés. Ce qui aide les développeurs dans [29] pour identifier le type de ces données recueillies.

iii. Comparaison des valeurs de seuil et envoi au département concerné :

Une valeur 'Seuil' représentant la valeur acceptée pour un objet envisagé (tel que le nombre des nids de poule détecté) est définie pour chaque objet sous supervision qui est donné par le service concerné. La valeur de seuil est ensuite comparée à la valeur réelle calculée après analyse des données. Si la valeur moyenne calculée est égale ou supérieure au seuil, cette information sera considérée pour l'envoi au département concerné.

iv. Mise en œuvre et résultats :

Deux firmwares ont été établis sur la base de l'OBU (On Board Unit) et RSU (Road Side Unit) sont développés en langage 'C' en utilisant les fonctions de la bibliothèque Contiki avec certaines bibliothèques 'C' pour envoyer des rapports concernant la route et diffuser les données routières. Ces firmwares ont été compilés avec Contiki sur des notes, qui sont utilisés en simulation pour démontrer le comportement et la fonctionnalité de l'OBU et de l'RSU.

- Au niveau de l'OBU : Après l'exécution de l'OBU, des rapports tels que Accident, état de la route, etc. ont été diffusés à tous les utilisateurs à un intervalle de temps aléatoire.
- Au niveau de l'RSU : La connexion de diffusion a été établie en utilisant Contiki OS fourni par la bibliothèque et la fonction callback () est appelée après la réception des données de la connexion. Lorsque RSU reçoit un message de l'OBU, il traite le message puis augmente la valeur du compteur du message qu'il reçoit, puis RSU décide de disséminer le message ou non en fonction de la valeur correspondante de son seuil et réinitialise le compteur. Fondamentalement, quand RSU reçoit un rapport de divers OBU, il vérifiera que "Comment les OBU ont-ils envoyé le même rapport ?" Et alors le compteur augmentera sa valeur. Cette valeur de compteur est utilisée pour calculer une valeur de seuil particulière qui sera ensuite comparée en utilisant la fonction définie par l'utilisateur threshold ()).

La figure 2.6 suivante présente le modèle de l'IoV proposé par les concepteurs de [29] pour les systèmes de transports intelligents dans l'objectif de collecter des données.

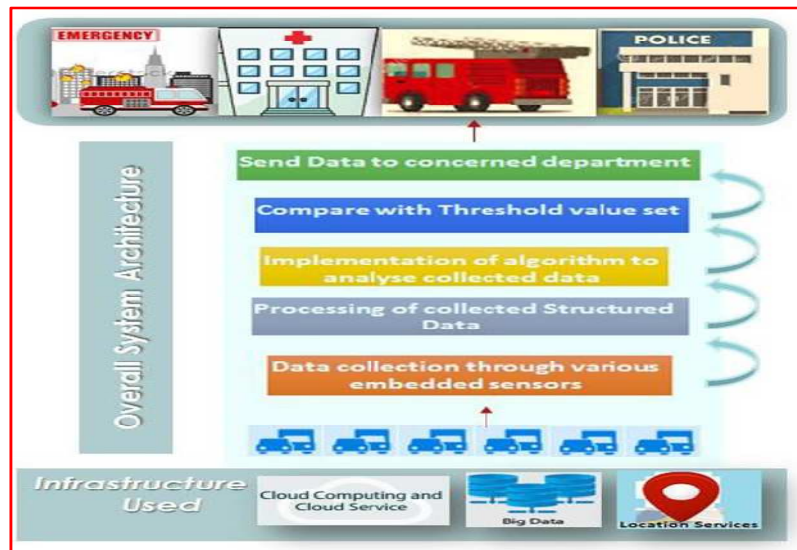


Figure 2.6 modèle de l'IoV proposé pour la collecte et l'analyse des données.

2.4.4 Discussion

Cette approche a eu comme objectif d'identifier les types d'informations collectées par les véhicules, aussi de concevoir un algorithme efficace qui analyse les données structurées collectées et de fournir les détails de l'analyse au canal approprié afin que les actions nécessaires peuvent être prises à temps.

Cependant, cette dernière n'implique pas l'inclusion d'un grand nombre d'objets connectés donc les types d'informations énormes et différents ne puissent pas être collectés, traités et analysés. En outre, ce modèle proposé repose sur l'utilisation de Cloud qui pose les problèmes de sécurité et la confidentialité des données.

2.5 Conclusion

Dans ce chapitre, nous avons vu quelques travaux connexes concernant notre domaine de recherche (la collecte de données dans l'IoV) tel que CarStream, un système de traitement de données évolutif et performant centré sur le Cloud. Par ailleurs, comment on peut collecter des données importantes concernant les véhicules connectées en utilisant des Smartphones pour accéder aux capteurs de ces véhicules. Une autre technologie basée sur la mise en œuvre de l'IoV et de la collecte de données à partir des capteurs jusqu'à l'analyse et, à la fin, la prise des décisions basées sur ces derniers est étudiée.

D'après notre étude, nous avons trouvé que ces travaux posent certaines limites et inconvénients malgré les avantages qui les offrent. Ce qui nous a poussés à proposer notre contribution qui va être décrite dans le chapitre suivant.

Chapitre 03 : Conception du DAG_AODV

La nouvelle ère de l'Internet des Objets (IoT : Internet of Things) a suscité l'évolution des réseaux ad hoc véhiculaires classiques (VANET) vers le paradigme de l'Internet des véhicules (IOV : Internet of Vehicles). Dans cette partie émergente de l'IoT, les objets sont des véhicules intelligents interconnectés, qui accèdent au réseau internet pour l'échange et le traitement des données.

Comme mentionné au début, nous nous sommes intéressés par ce projet de la conception et le développement d'un système de collecte de données dans l'internet des véhicules (IoV). Cette collecte de données peut être bénéfique aux conducteurs et aux passages en améliorant la sécurité routière et en satisfaisant les exigences des usagers de la route.

Dans le chapitre précédent, nous avons présenté quelques travaux connexes. Et comme tout un nouveau et un récent domaine de recherche, peu de travaux ont été publiés sur les protocoles et les techniques de la collecte de données dans l'IOV. Cependant, à notre avis, ces travaux connexes existants ne sont pas satisfaisants et nous estimons qu'il est possible d'obtenir de meilleurs résultats en termes de débit, taux de paquets reçus avec succès et délai de bout en bout.

Pour combler le problème de la collecte de données dans l'internet des objets, nous proposons dans ce chapitre une amélioration du protocole AODV dénommée DAG_AODV, désignée dans le but de rendre l'opération de la collecte plus réactive à la mobilité des nœuds.

3.2 Conception générale de DAG_AODV

L'IOV permet l'échange d'informations entre les véhicules, les infrastructures routières, les passagers, les conducteurs, les capteurs et les actionneurs électriques et Internet [30].

La collecte de données suivie d'un traitement et d'un stockage puis d'une analyse avec précision dans un environnement en temps réel constituera un grand défi pour la mise en œuvre d'IOV [31].

La plupart des collectes de données dans les réseaux véhiculaires sont réalisées par l'intermédiaire d'un mécanisme de dissémination par lequel chaque véhicule diffuse périodiquement ses propres informations à travers le réseau [32]. Cependant, toutes les collectes de données reposant sur ce mécanisme, ne résultent pas d'une demande spécifique correspondant à un besoin momentané. Conséquemment, les informations sont diffusées régulièrement et de manière récurrente même si elles ne sont pas utilisées. Et dans le pire des cas lorsque les véhicules ne savent pas quelles sont les données qui pourraient servir, alors ils auront souvent tendance à propager plus de données que nécessaire, ce qui engendre un gaspillage de la bande passante.

Pour contourner ce problème, il serait donc plus raisonnable que toute collecte de données découle d'une requête correspondant à un besoin précis et momentané émanant d'un véhicule spécifique.

Pour ce faire, nous présentons notre contribution qui est une collecte de données pour les réseaux de véhicules basée sur la priorité de la donnée transmise. Dans notre proposition, toute collecte de données émane préalablement d'une demande incluant, de manière non exhaustive, le type de donnée visé par cette dernière telle qu'un accident, état de la route, embouteillage, congestion routière, etc. et le plus important est la priorité de cette donnée qui exprime également son niveau de gravité. Par conséquence, la donnée sera traitée et exploitée selon ce niveau de gravité.

Afin d'accélérer le transfert de données, nous proposons l'intégration de la technique multi canaux, multi interfaces et une file d'attente prioritaire pour assurer que les données avec une priorité élevée ne se retardent jamais. La troisième idée proposée dans ce projet afin de garantir un aspect important de la qualité de service(QoS), le processus de diffusion stochastique (stochastic broadcast) expliqué dans ce chapitre.

La figure 3.1 présente un environnement IoV, et la figure 3.2 donne un vue générale concernant notre conception du protocole DAG_AODV. Notre application prend comme des entrées de diverses informations provenant de la route, donc à ce moment là où le processus de la collecte de donnée se débute. Une fois ces informations sont récoltées, elles doivent être traitées et analysées en caractérisant chacune par une priorité spécifique. Finalement, ces données collectées et traitées seront envoyées au destinataire approprié.

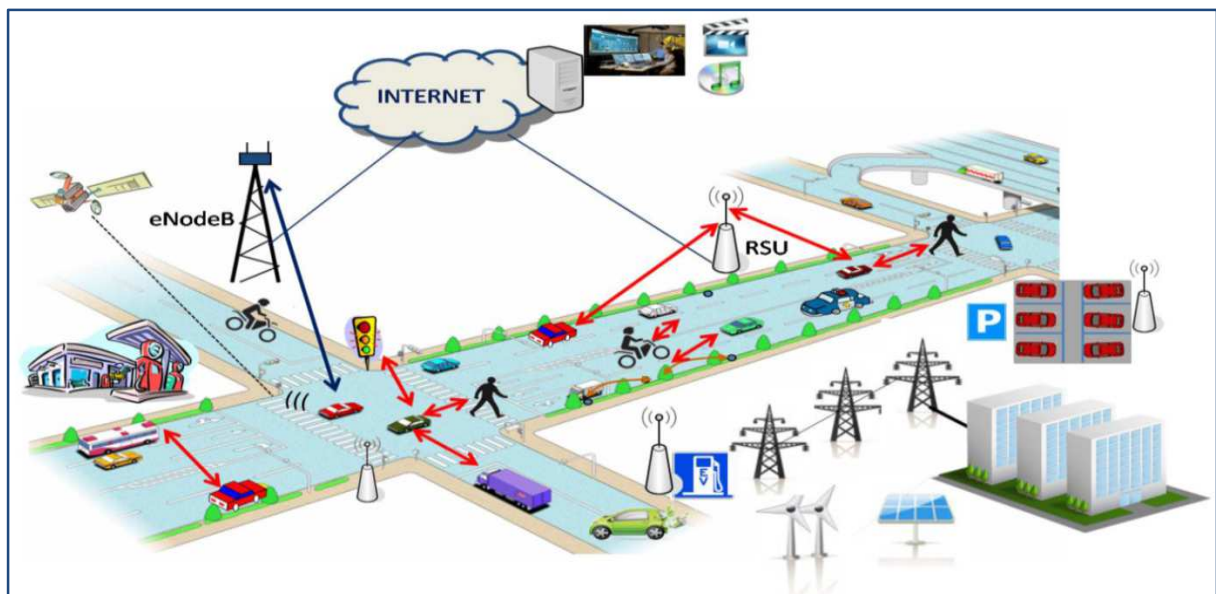


Figure 3.1 un environnement IoV.

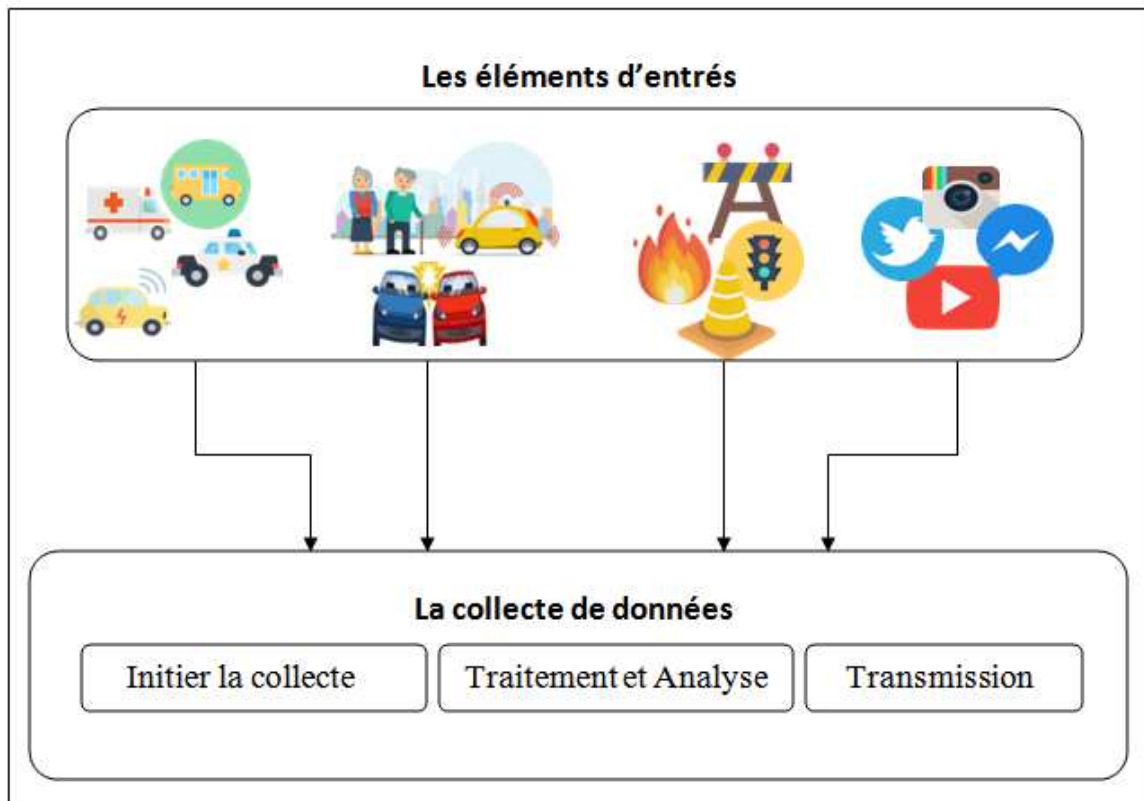


Figure 3.2 Conception générale de DAG_AODV.

3.3 Conception détaillée de DAG_AODV

La collecte de données s'agit de recueillir le maximum des informations routières qui ont pour but d'améliorer la sécurité routière, en évitant les accidents et en réduisant les situations à risques, de faciliter la gestion des situations exceptionnelles et des crises par les autorités. En outre, la collecte de données permet de réduire les coûts économiques et environnementaux pour les usagers de la route, par fluidification du trafic ou par la proposition d'itinéraires alternatifs ou de périodes de départ conseillé, d'améliorer le confort de l'utilisateur en lui permettant de connaître les conditions de conduite qu'il rencontrera, et l'inciter à adapter sa conduite ou son itinéraire.

Ces données routières peuvent être produites par les gestionnaires routiers, par les forces de police et de gendarmerie ou par les services de secours, puis validées et intégrées par les centres de gestion de trafic en un ensemble agrégé et exploitable. L'information ainsi agrégée est ensuite diffusée au public [33].

Une application de collecte de données est divisée en quatre phases à savoir :

- 1) une phase de préparation,
- 2) une phase de collecte,

- 3) une phase d'agrégation et
- 4) une phase d'envoi.

Dans notre travail, les quatre phases sont effectuées comme le suivant :

3.3.1 Débuter une collecte (phase de préparation et de collecte)

Dès l'instant qu'un véhicule décide de débiter une collecte de données, il envoie un message composé de la donnée et d'un champ à destination de son voisinage. Ce champ correspond à la priorité de la collecte (dag_P). Avec plus de détail ce champ nous permet de décider si l'information recueillie s'agit d'une donnée utile et doit être servit rapidement telle qu'un accident ou bien inutile donc on peut la retarder s'il y a une autre donnée utile en cours d'envoi. Par conséquent, il devient possible d'envoyer uniquement des informations utiles pour économiser la bande passante.

3.3.2 Traitement et analyse des données (phase d'agrégation)

Les informations collectées seront structurées et s'appuieront sur certains mots-clés. Le traitement des données implique l'identification du type en fonction de ces mots-clés. Après l'identification du type de donnée collectée, une étape intéressante est l'affectation de la priorité. Si cette donnée est importante, sensible ou bien correspond à une situation de risque alors la donnée doit avoir une priorité élevée afin d'être servi exclusivement.

Le processus de la collecte utilisé dans notre application, ainsi la façon de transmission de la donnée collectée est illustré dans la figure 3.3.

3.3.3 Le routage de données (phase d'envoi)

Lors de la transmission d'un paquet d'une source vers une destination, il est nécessaire de faire appel à un protocole de routage qui acheminera correctement le paquet par le «meilleur» chemin. Notre choix s'est porté sur le protocole réactif AODV avec quelques modifications. Notre amélioration DAG_AODV utilise un mécanisme de diffusion dans le réseau pour découvrir les routes valides.

Dans notre travail, l'acheminement de données collectées se base essentiellement sur la priorité de ces dernières. Les informations avec une priorité élevée c'est-à-dire un niveau de gravité important doivent être transmit en premier lieu et dans un temps opportun.

Pour plus de détail, le routage de donnée collectée s'appuie sur le champ dag_P . Donc, si cette dernière est une donnée utile le champ dag_P doit avoir la valeur 1 ($dag_P = 1$). Dans ce cas la donnée est acheminée à sa destination en premier lieu. Sinon, la valeur de ce champ vaut 0 et elle sera retarder s'il y a d'autres données utiles en cours d'envoi.

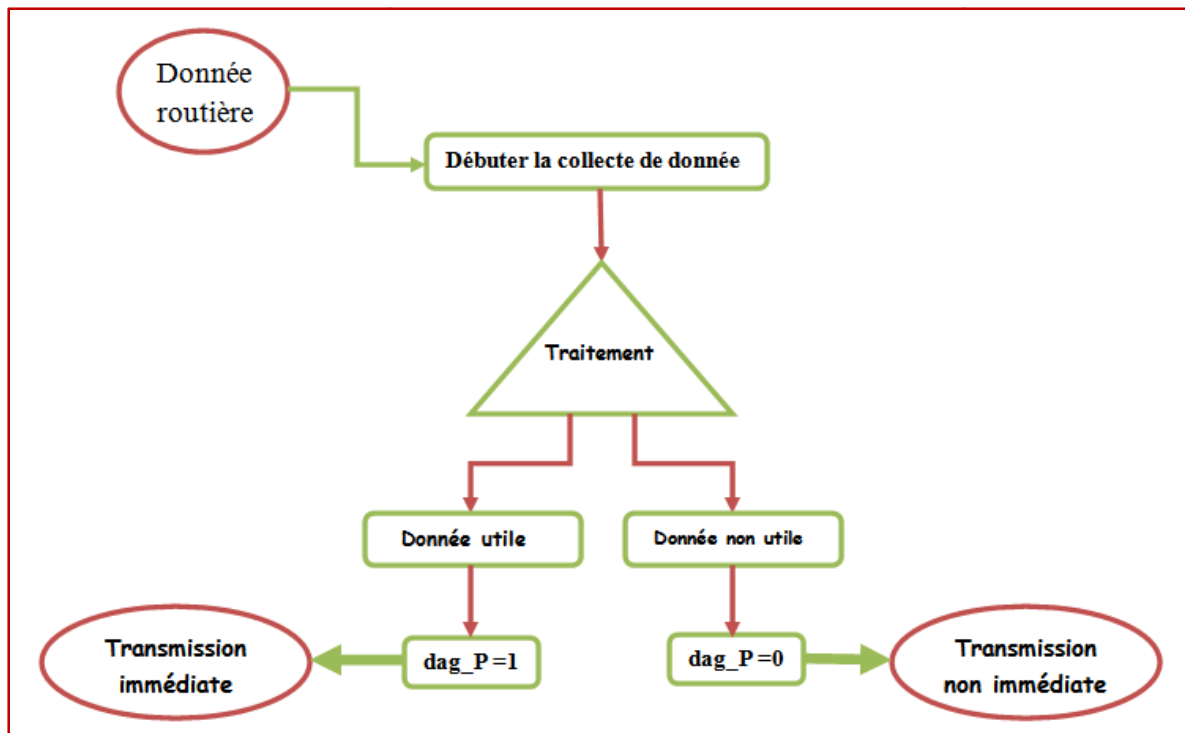


Figure 3.3 Le processus de la collecte de donnée par DAG_AODV.

Afin d'accélérer le routage de données d'urgence collectées, et garantir un transmission rapide de telles données nous proposons dans notre travail les techniques suivantes :

A. File d'attente prioritaire (PQ)

Ils existent différentes disciplines de mise en file d'attente qui peuvent être utilisées pour contrôler quels paquets sont transmis (allocation de bande passante) et quels paquets sont supprimés (espace tampon). La discipline de mise en file d'attente affecte la latence subie par un paquet, en déterminant combien de temps un paquet attend d'être transmis. A titre d'exemples de disciplines de mise en file d'attente communes on peut distinguer la mise en file d'attente FIFO (first-in-first-out), la mise en file d'attente prioritaire (PQ) et la mise en file d'attente pondérée-équitable (WFQ) [34].

Afin de garantir que pendant l'encombrement les données ayant la priorité la plus élevée ne sont pas retardées par le trafic de priorité inférieure, la mise en file d'attente prioritaire est utilisée dans notre application comme il est présenté dans la figure 3.4. PQ est conçu pour les environnements qui se concentrent sur les données essentielles à la mission, en excluant ou en retardant le trafic moins critique pendant les périodes de congestion.

La mise en file d'attente prioritaire affecte plusieurs files d'attente à une interface réseau avec un niveau de priorité pour chacune de ces files d'attente. Une file d'attente avec une priorité plus élevée est traitée plus tôt qu'une file d'attente avec une priorité inférieure. La mise en file d'attente prioritaire peut avoir quatre files d'attente préconfigurées : haute, moyenne, normale et basse priorité.

Si les paquets arrivent dans la file d'attente haute, la file d'attente prioritaire supprime tout ce qu'elle fait pour transmettre ces paquets. La figure 3.5 montre l'aspect de ce type de la file d'attente.

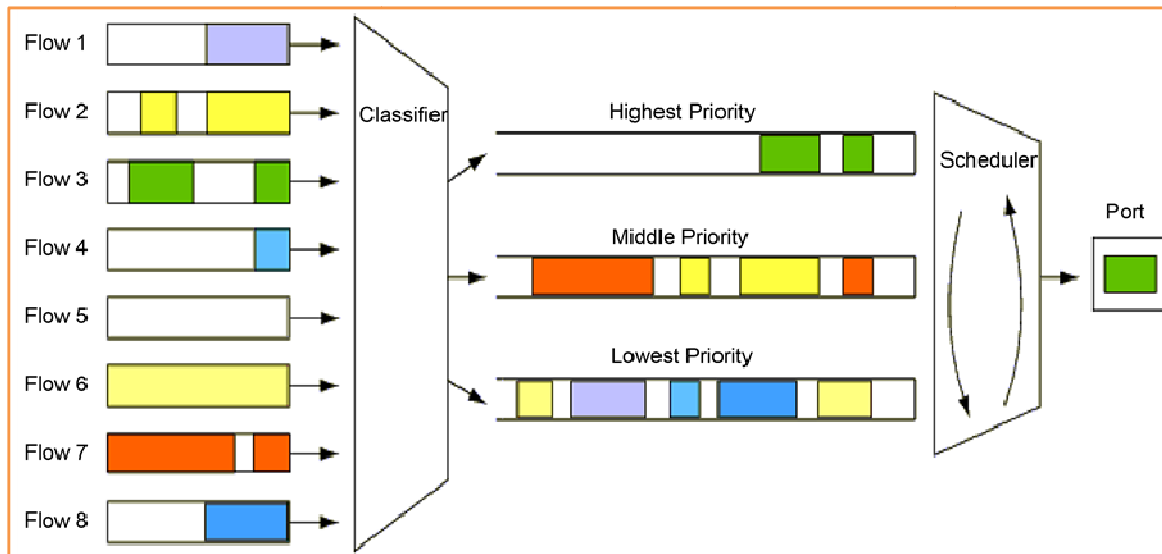


Figure 3.4 File d'attente prioritaire.

La figure 3.5 présente le mécanisme de mise en file d'attente utilisée par le protocole DAG_AODV.

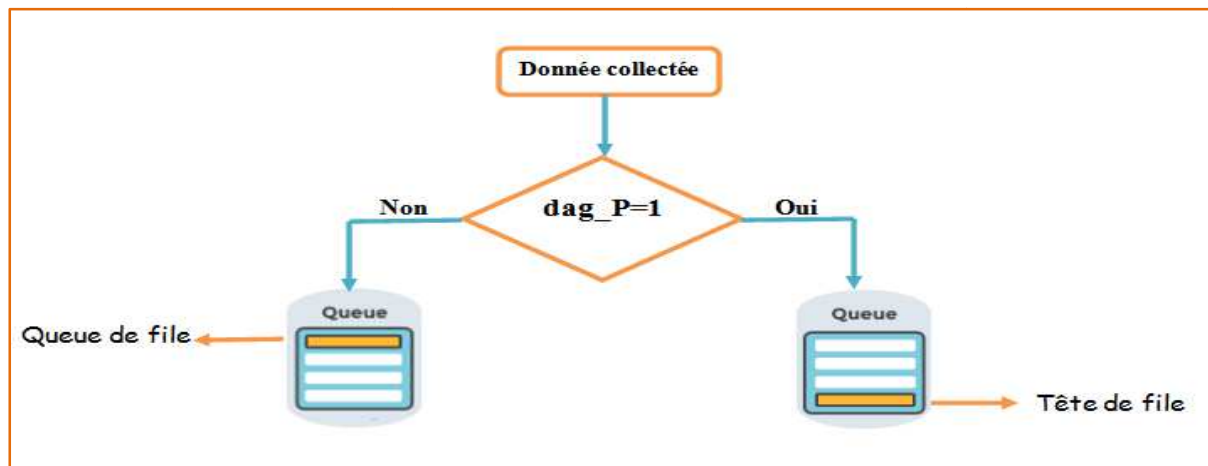


Figure 3.5 File d'attente prioritaire pour les données collectées d'urgence.

B. Diffusion stochastique (Stochastic broadcast)

La diffusion est une opération d'amélioration de la découverte d'itinéraire dans le réseau véhiculaire. Bien que la diffusion par inondation soit simple mais inefficace et entraîne

des relais de messages redondants. La diffusion allégée est nécessaire pour la découverte d'itinéraire efficace et pour améliorer la durée de vie du réseau [39]. Dans notre protocole DAG_AODV, la technique de diffusion basée sur les probabilités est proposée pour réduire les problèmes d'inondation.

La diffusion stochastique est une technique proposée pour améliorer les performances de l'AODV en réduisant les frais généraux des paquets RREQ pendant l'opération de découverte de l'itinéraire [35].

En utilisant cette technique pendant le processus de découverte d'itinéraire, tous les nœuds intermédiaires ou de routage entre les nœuds source et de destination prennent la décision de diffuser / transmettre le paquet RREQ vers la destination ou de le supprimer. Avant de transmettre un paquet RREQ, chaque nœud calcule le facteur de baisse qui est une fonction de l'inverse du nombre de comptes de bonds traversés par le paquet RREQ. Ce facteur de baisse est compris entre 0 et 1. De plus, le nœud génère un nombre aléatoire de 0 à 1. Si ce nombre aléatoire est supérieur au facteur de baisse, le nœud transmet le paquet RREQ. Sinon, le paquet RREQ est supprimé. La suppression de tels paquets n'entraîne pas nécessairement un nouveau processus de découverte d'itinéraire par le nœud source. Puisque la diffusion originale produit plusieurs paquets RREQ via les voisins et cette onde diffusante résulte rapidement en un grand nombre de paquets RREQ traversant le réseau à la recherche de la destination. Une grande partie de ces paquets est redondante car, dans le cas idéal, un seul paquet RREQ peut trouver le meilleur chemin. En outre, un certain nombre de ces paquets diffusant dans des directions éloignées de la destination finiront par expirer.

Nous avons utilisé cette diffusion puisque elle sert à minimiser les paquets RREQ redondants, ou bien, de laisser tomber le plus possible de ces paquets RREQ redondants. La politique de baisse est conservatrice et sa valeur diminue avec le nombre de sauts. Lorsque les paquets RREQ s'approchent du nœud de destination, les chances de survie des paquets RREQ sont plus élevées. Par conséquent, la première phase du processus de découverte d'itinéraire est achevée dès que possible et un paquet RREP peut être transmis du nœud de destination au nœud source.

Cette technique permet de réduire une proportion de paquets RREQ qui n'atteindront jamais le nœud de destination grâce à l'abandon de paquets RREQ redondants, ce qui entraîne une diminution de l'encombrement du réseau.

L'algorithme suivant est utilisé dans le processus de prise de décision : abandonner les paquets RREQ par l'intermédiaire ou les nœuds de routage [36].

Stochastic broadcast: Etape1: calculer $drop_factor$ $drop_factor = (1/(Hop_count_of_RREQ_packet + 1))$ Etape 2: calculer une valeur aléatoire de l'ordre de 0 à 1 si $(random_value > drop_factor)$ alors Broadcast/forward RREQ_packet ; sinon drop RREQ_packet ;
--

C. Multi canal multi interface (MCMI)

Dans la plupart des réseaux multi-sauts, une seule interface est suffisante car un seul canal est utilisé. Bien que, lorsque plusieurs canaux sont disponibles, avoir plus d'une interface est bénéfique [37].

MCMI est une technique qui assigne dynamiquement des canaux à chaque interface des nœuds et réserve les temps de transmission des nœuds sur ces canaux sélectionnés. Cette technique peut fournir la transmission simultanée dans différents canaux pour chaque nœud avec plusieurs interfaces.

Si les interfaces de deux nœuds sont sur des canaux distincts et qu'une seule interface est utilisée, ils ne peuvent pas communiquer entre eux. Pour réduire les besoins de synchronisation et les surcoûts, chaque interface doit rester sur un canal pendant de nombreuses durées de transmission de paquets [37].

DAG_AODV utilise la technique de MCMI grâce à ses avantages tel que la transmission simultanée et le problème des stations exposées est ainsi atténué. Lorsque plusieurs canaux non superposés sont utilisés, les nœuds voisins peuvent utiliser des canaux différents en même temps. En outre, un nœud équipé de plusieurs interfaces peut utiliser des canaux différents en même temps. Ainsi, il accélère le transfert de nos données collectées et traitées en transmettant et en recevant des paquets en parallèle sur des canaux qui ne se chevauchent pas ce qui permet de réduire les interférences de canal et améliorer la capacité du réseau.

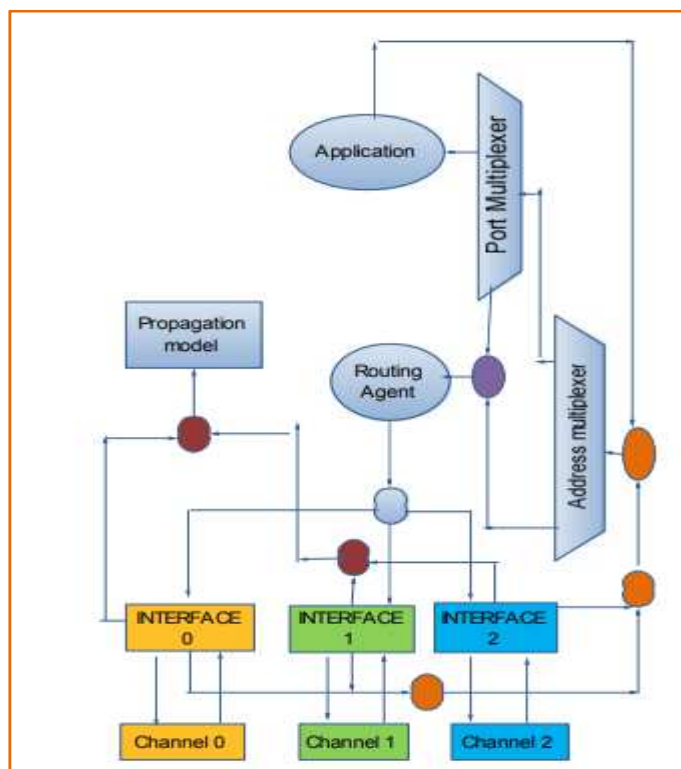


Figure 3.6 Architecture de nœud mobile multi canal [38].

3.4 Conclusion

Dans ce troisième chapitre, nous avons présenté notre conception générale et détaillée prise en charge dans le but d'avoir un système de collecte de donnée dans l'IoV. A travers ce chapitre nous avons pu savoir les différentes approches et techniques utilisées dans notre amélioration DAG_AODV pour avoir une meilleure collecte de donnée.

Nous allons présenter notre implémentation du protocole DAG_AODV dans le chapitre suivant ainsi nos résultats de la simulation.

Chapitre 04 : Implémentation et résultats

Nous avons présenté précédemment les concepts théoriques relatifs à l'amélioration du protocole AODV afin de collecter les données dans un environnement IoV et acheminer ces dernières au destinataire approprié. À travers ce chapitre, nous nous intéressons à l'évaluation de notre amélioration DAG_AODV par l'intermédiaire de mesures de performance sur différents scénarios de mobilité, de mouvement et de trafic. Dans ce qui suit, nous présentons le choix de l'environnement de simulation. Puis, nous détaillons l'implémentation de la solution.

Enfin, après avoir décrit les différents paramètres à prendre en considération pour mettre en place la simulation, nous montrons l'efficacité de notre solution à travers les résultats obtenus de la simulation.

4.2 La simulation

La simulation des réseaux est une technique qui sert à modéliser le comportement d'un réseau, soit par le calcul de l'interaction entre les entités du réseau en utilisant des formules mathématiques, ou en capturant et reproduisant des observations à partir d'un réseau réel. L'outil de mise en œuvre de la simulation est le simulateur, Il présente l'évolution du modèle du phénomène sous des conditions contrôlables et observables.

L'une des raisons les plus motivantes d'utiliser la simulation est la difficulté de créer une implémentation réelle. Afin de montrer l'efficacité et les performances d'un système, il n'est pas toujours possible d'accéder aux infrastructures nécessaires en raison de leurs coûts élevés. Pour préserver ce problème, on a du faire recours à la simulation qui met à la disposition des utilisateurs un environnement de simulation assez complet.

4. 3 Les outils de simulation de DAG_AODV

4.3.1 NS2 (Network Simulator)

a) Définition

Network Simulator (Version 2), largement connu sous le nom de NS2, est simplement un outil de simulation événementiel qui s'est révélé utile pour étudier la nature dynamique des réseaux de communication. La simulation de fonctions et de protocoles de réseau câblé ainsi que sans fil (par exemple, des algorithmes de routage, TCP, UDP) peut être réalisée en utilisant NS2. En général, NS2 fournit aux utilisateurs un moyen de spécifier de tels protocoles réseau et de simuler leurs comportements correspondants [40].

NS2 utilise le langage **Tcl**(Tool Command Language) pour créer un fichier de scénario de simulation qui définit La topologie du réseau, le temps de transmission, l'utilisation du protocole, etc. l'exécution de fichier de scénario génèrent les fichiers out.tr et out.nam.

- **out.tr** (fichier de trace) contient toutes les informations nécessaires sur la communication. Ce fichier décrit la façon dont un paquet a été transmis.
- **out.nam** contient les données pour l'animation du résultat de l'expérience. Ce fichier peut être exécuté par Nam, un logiciel d'animation.

b) Architecture de base :

Cet outil de simulation combine deux langages de script OTcl et le langage C++.C ++ est rapide à exécuter mais plus lent à changer, ce qui le rend adapté à l'implémentation détaillée du protocole. OTcl fonctionne beaucoup plus lentement mais peut être changé très rapidement (et interactivement), ce qui le rend idéal pour la configuration de simulation [41].

Nous montrons l'architecture de NS2 à travers la figure 4.1 suivantes.

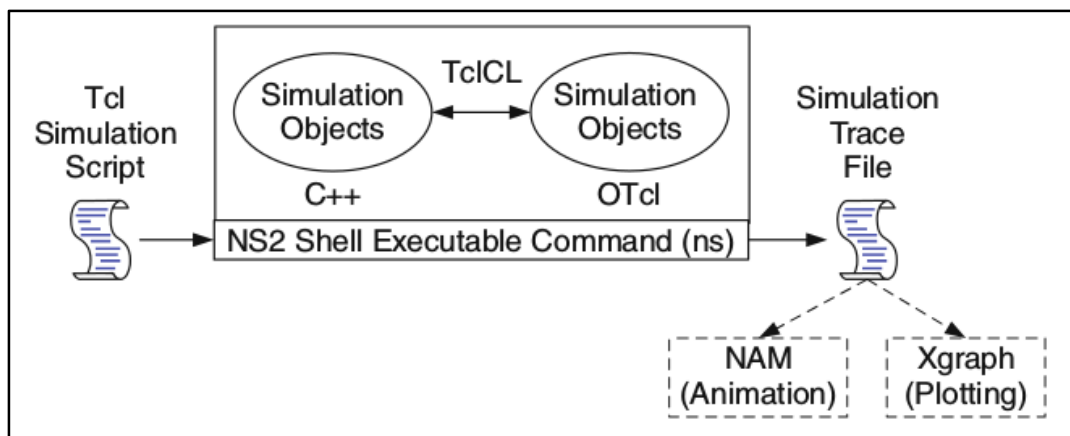


Figure 4.1 Architecture de base de NS2 [40].

c) Motivation du choix deNS2

En outre de ces avantages tels que la flexibilité, la réutilisabilité, l'extensibilité, la disponibilité, NS2 présente les avantages suivant :

- ✓ Un simulateur de type Open source,
- ✓ Ne nécessite pas d'équipement coûteux,
- ✓ Inclut une vaste bibliothèque de modèles de composants réseau,
- ✓ Les résultats peuvent être rapidement obtenus,
- ✓ Plus d'idées peuvent être testées dans un délai plus court,
- ✓ Conditions expérimentales contrôlées,
- ✓ Support multiplateforme (GNU/Linux, MAC OSX, Solaris, Windows)

On cite que NS-2 offre un compromis entre performance et facilité d'utilisation. Ainsi, nous avons adopté cet outil pour l'implémentation et la simulation de notre proposition.

4.3.2 OpenStreetMap

OpenStreetMap (**OSM**) est un projet fondé en 2004 qui a pour but de constituer une base de données géographique libre du monde (permettant par exemple de créer des cartes sous licence libre), en utilisant le système GPS et d'autres données libres. Il nous permet de collecter des données dans le monde entier sur les routes, voies ferrées, les rivières, les forêts, les bâtiments et bien plus encore.

4.3.3 SUMO (Simulation of Urban Mobility)

SUMO est un logiciel de simulation de circulation routière open source, hautement portable, microscopique et continu conçu pour gérer les grands réseaux routiers [42]. SUMO capable de simuler villes et autoroutes. Il prend en charge le changement de voies ainsi que la fonction de suivi des voitures [43].

SUMO a beaucoup de caractéristiques qui lui rendent un logiciel de simulation très puissant et utilisé.

Parmi ces caractéristiques on peut citer [44] :

- Simulation microscopique : les véhicules, les piétons et les transports en commun sont modélisés explicitement,
- Interaction en ligne : contrôle de la simulation avec TraCI (un protocole de communication basé sur socket)
- Simulation du trafic multimodal, c'est-à-dire les véhicules, les transports en commun et les piétons
- Les horaires des feux de circulation peuvent être importés ou générés automatiquement par SUMO.
- Aucune limitation artificielle de la taille du réseau et du nombre de véhicules simulés
- Formats d'importation pris en charge : OpenStreetMap, VISUM, VISSIM, ICI
- SUMO est implémenté en C++ et utilise uniquement des bibliothèques portables

Prise en charge multiplateforme.

- SUMO est testé sur Linux, macOS et Windows.

4.4 Évaluation de performance (Métriques considérées)

L'évaluation de notre système nécessite l'utilisation de certaines métriques qui ont pour but de mesurer les performances de cette dernière. Nous avons choisi trois métriques pour notre évaluation qui sont les suivantes :

4.4.1 Débit moyen (throughput)

C'est une métrique importante et très utilisée dans l'analyse des protocoles réseaux. Le débit est le nombre de paquets reçus avec succès dans une unité de temps.

$$\text{Débit} = \frac{\text{nombre total de paquets de données livrés}}{\text{temps de transmission des paquets}}$$

4.4.2 Taux de paquets reçus avec succès

Ce taux noté PDR (pour Packet Delivery Ratio) est le nombre de paquets de données reçus avec succès par la destination par rapport au nombre de paquets de données émis par la source. Cette métrique nous permet de vérifier si l'extension du protocole a un impact sur le transfert de paquets de données avec succès.

$$PDR = \frac{\text{nombre de paquets de données recus par la destination}}{\text{nombre de paquets de données émis par la source}} \times 100$$

4.4.3 Délai moyen de bout-en-bout

Il s'agit du temps moyen qu'un paquet de données envoyé avec succès prend pour atteindre la destination. Parfois il est dénoté RAL pour Route Acquisition Latency ou encore End-to-End Delay.

Ce temps inclut le délai de traitement ainsi que le délai d'attente dans les files d'attentes dans chaque nœud intermédiaire. Il est calculé selon la formule suivante :

$$\text{Délai} = \frac{\text{délai de livraison total des paquets de données}}{\text{nombre de paquets reçus}}$$

4.5 Implémentation du protocole DAG_AODV

4.5.1 Description et pseudo code de DAG_AODV

A. La collecte de données

L'internet des véhicules est capable de capturer une gamme d'informations sur la route. Les données collectées doivent être traitées et exploitées d'une manière efficace et sûre.

Afin de répondre au problème de la collecte de donnée dans l'internet des véhicules (IoV), nous avons bien expliqué notre méthode à suivre dans le chapitre précédent. Dans cette section un pseudo code qui sert à collecter les données concernant le trafic routier et se base surtout sur les situations d'urgences et sensibles provenant de diverses véhicules avec les différentes étapes prises en charge sont présentés et expliqués. Après cette collecte, un traitement et transfert basé sur la priorité de ces données est nécessaire pour les transformer en des informations utilisées afin de déterminer les actions futures.

Etape 1 : déclaration et initialisation des mots clés :

On a choisi de déclarer des variables (mots clés) sous forme d'un booléen qui présentent le type de donnée à collecté de la route. Cette déclaration doit être présente dans le fichier dag_aodv.h.

```
bool accident ;  
bool trafficCongestion ;
```

Les variables déjà déclarées doivent être initialisées avec la valeur "false" dans le fichier dag_aodv.cc. Cette valeur va nous indiquer s'il existe une situation d'urgence dans la route.

```
accident = false;
trafficCongestion = false;
```

Etape 2 : la collecte de données.

A partir des paquets circulant dans le réseau on peut capturer une quantité massive de données, certaines sont utiles mais autres on peut les abandonner. Notre collecte s'appuie sur certains mots clés (accident, traffic_Congestion etc.) existant dans le fichier TCL et correspond à une situation d'urgence ou dérangeante.

```
if(strcmp(argv[1], "accident") == 0) {
    accident = true;
    return TCL_OK;
}
else if(strcmp(argv[1], "congestionRoutier") == 0)
{
    congestionRoutier = true;
    return TCL_OK;
}
```

Etape 4 : Traitement de données

L'une des étapes qui doit suivre la collecte de données et le traitement de ces dernières. La tâche qui caractérise notre traitement est l'affectation de certain niveau de priorité à chaque donnée collectée. Pour ce faire, nous avons ajouté le champ dag_P de type entier dans la structure de paquet (déclaré dans le fichier DAG_AODV_packet.h).

```
if(accident==true || congestionRoutier == true)
{ch->dag_P = 1;
}
else {
ch->dag_P = 0;}

```

Pour suivre le parcours de l'un des parquets d'urgence, nous avons ajouté le pseudo code suivant. La situation d'urgence choisie est un accident.

```
if(ih->saddr() == 0 &&ih->daddr() == 9)
{FILE *fp;
fp = fopen("Path.txt", "a");
fprintf(fp,"hops : %d \t next_hop: %d \t \n",index,rt->rt_nexthop);
fclose(fp);
}
```


B. File d'attente prioritaire

Dans le but d'assurer que les paquets ayant une priorité élevée par rapport au d'autres paquets doivent être passés rapidement, nous avons pris en considération la file d'attente prioritaire qui se base sur la valeur du champ dag_P. Le code suivant montre les différentes modifications ajoutées à la méthode **enque** en modifiant le fichier dag_aodv_rqueue.cc. Alors, un paquet d'urgence se place au sommet de la queue.

```
.....
if(head_ == 0) {
head_ = tail_ = p;
}
else {
if(ch->dag_P == 1)
{
p->next_ = head_;
head_ = p;
printf("Emergency packet \n");
printf ("srcaddress : %d \t dst address :%d \n",ih->src_.addr_,ih->dst_.addr_);
}
else {
tail_->next_ = p;
tail_ = p;
printf ("Not emergency_packet \n " );
printf ("srcaddress : %d \t dst address :%d \n",ih->src_.addr_,ih->dst_.addr_);
}
}
}
```

C. Multicanaux, multi-interface :

Comme déjà expliqué dans le chapitre précédent, les données collectées seront envoyées à travers plusieurs canaux simultanément. L'outil de simulation NS-2 permet seulement l'envoi des données d'une manière séquentielle via un seul canal. Et puisque nous voulons que notre protocole envoie les données sur plusieurs canaux et interfaces alors des changements dans l'environnement de notre simulation sont requis.

Nous avons choisi à montrer quelques modifications, qui semblent plus importantes, appliquées dans l'environnement de NS-2 afin qu'il puisse supporter cette technique. Pour plus de détaille le lecteur peut s'orienter vers les références [45] [46].

Parmi ces modifications, nous avons pris en considération l'ajout de quatre nouvelles procédures. Elles sont :

1. Procédure pour changer le nombre d'interfaces

```
Simulator instproc change-numifsnewnumifs {
$self instvarnumifs_
Set numifs $newnumifs}

```

2. Procédure pour ajouter une interface (canal)

```
Simulator instproc add-channel {indexch} {  
  $self instvarchan  
  setchan($indexch) $ch  
}
```

3. Pour obtenir le nombre d'interfaces de l'architecture TCL

```
Simulator instproc get-numifs { } {  
  $self instvarnumifs  
  if (info exists numifs) {  
    return $numifs  
  }else {  
    return ""  
  }  
}
```

4. Pour ajouter plusieurs interfaces en tant qu'argument

```
Simulator instprocifNum {val} {$self setnumifs $val}
```

D. Diffusion stochastique

Afin d'acheminer les données collectées et faire passer ces données vers la destination requis, DAG_AODV devrait déterminer tous d'abord un itinéraire vers cette destination. Lorsqu'un itinéraire n'est pas disponible pour la destination, un paquet de demande de route (RREQ) est inondé sur l'ensemble du réseau. L'inondation des paquets RREQ engendrera une consommation de la bande passante ce qui dégrade les performances de réseau.

Dans le but de minimiser le nombre de paquets RREQ pendant le processus de découverte d'itinéraire, nous avons utilisé une technique de diffusion stochastique (Stochastic broadcast). Le code suivant illustre les modifications apportées au fichier DAG_AODV.cc, spécifiquement nous avons modifié la méthode `recvRequest` pour qu'elle puisse éliminer une partie des paquets RREQ de façon aléatoire.

```

void
VAODV::recvRequest(Packet *p) {
-----
floatdrop_factor;
floathopsc;
floatrandom_value;
floatnum;
srand(rand());
num=(float)rand();
random_value = (num/RAND_MAX);
hopsc = (float)rq->rq_hop_count;
drop_factor = (1/(hopsc + 1.0));
if(random_value<drop_factor) {
Packet::free(p);
return;
}
}

```

4.5.2 Configuration et paramètres de simulation

Après l'explication détaillées de notre protocole DAG_AODV et la présentation de différentes parties des pseudos codes utilisées pour avoir un protocole de collecte de données, alors une évaluation de ce protocole est une tâche indispensable pour accomplir notre travail. Pour cela on a fait recours vers la simulation qui se déroule dans 100s avec des connexions CBR et FTP dans des scénarios différents où à chaque scénario la densité des véhicules se change (50, 75, 100). Différentes types de véhicules ont été pris en considération comme ambulance, la police, protection civile.

Cette section décrit les différents paramètres requis à prendre en considération dans notre simulation.

Paramètre de simulation	La valeur
Simulateur	NS2.35
Zone de simulation	3390×22140 : ville de Biskra
Temps de transmission	100 s
Modèle d'antenne	Antenna /Omni Antenna
Modèle de propagation radio	Propagation/TwoRayGround
TYPE MAC	IEEE 802.11
Type de file d'attente	Queue/DropTail/PriQueue
Protocoles de routage	AODV, DAG_AODV
Nombre de véhicules	50, 75, 100
Type de trafic	FTP, CBR

Tableau 4.1 Les paramètres de simulation.

Pour exploiter la mobilité véhiculaire qui est une nécessité dans les réseaux véhiculaire, nous avons utilisé SUMO couplé avec OpenStreetMap. Alors, sumo nous a permet de créer une mobilité des véhicules dans la topologie exploité par OpenStreetMap.

Les figures 4.2 et 4.3 nous donnent un aperçu concernant ces deux outils.

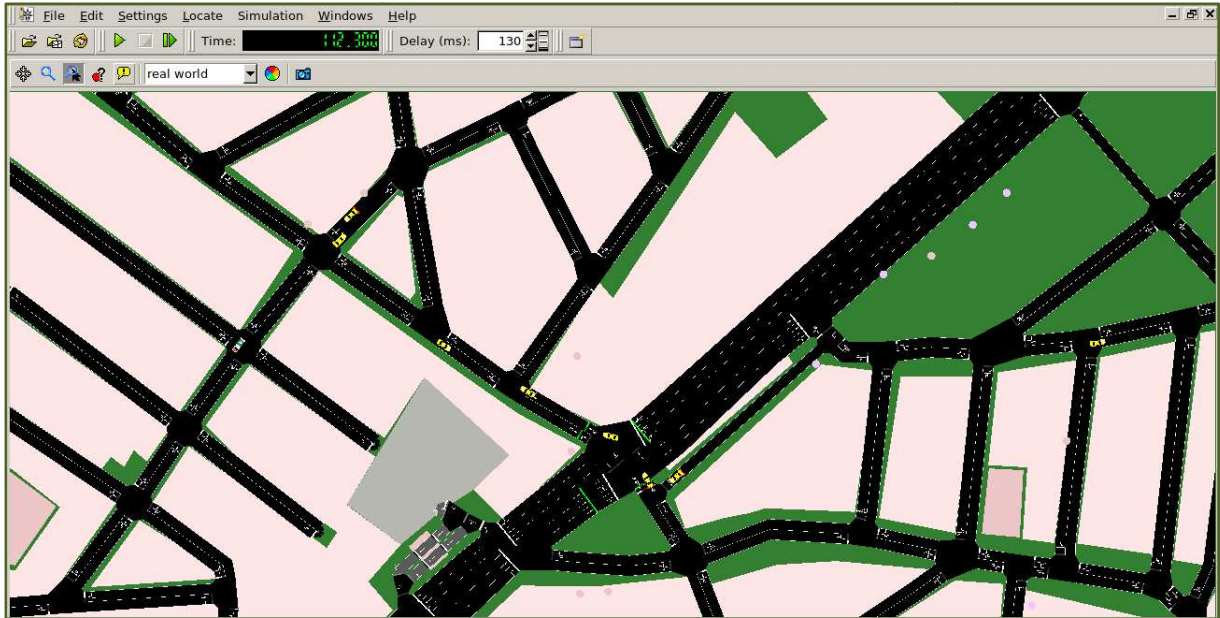


Figure 4.2 visualisation Sumo (ville de Biskra).



Figure 4.3 visualisation OpenStreetMap.

La figure 4.4 montre la représentation de notre topologie réseau décrit à l'aide de NS-2 avec l'outil de visualisation NAM.

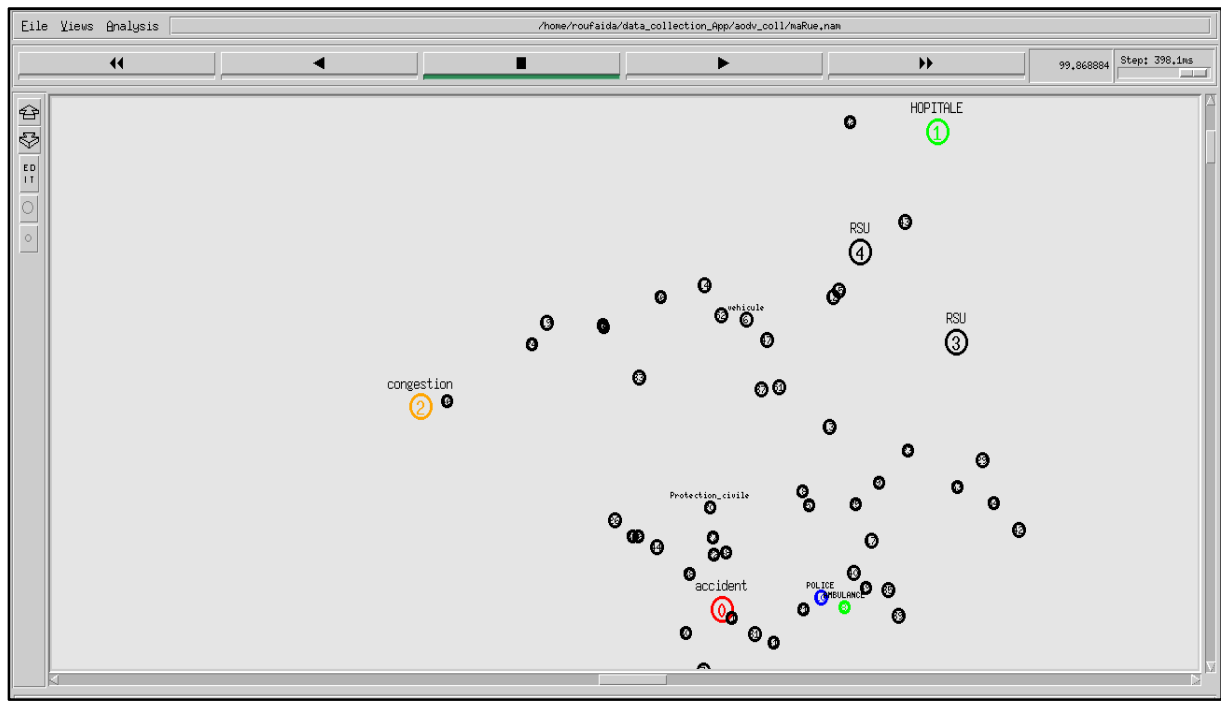


Figure 4.4 Visualisation NAM.

4.6 Résultats et discussion

Afin d'évaluer et présenter les performances de notre protocole DAG_AODV qui est un protocole destiné à la collecte de donnée, nous mettons en œuvre à travers cette partie de thèse les résultats obtenus de notre simulation. Nous avons choisi le protocole AODV pour faire les comparaisons avec DAG_AODV en utilisant les métriques : débit, délai de bout en bout et taux de paquets reçus avec succès. Ces métriques sont calculées à partir du fichier de trace en utilisant le programme AWK.

Alors, les différentes comparaisons et les résultats obtenus sont présentés dans les graphes suivant.

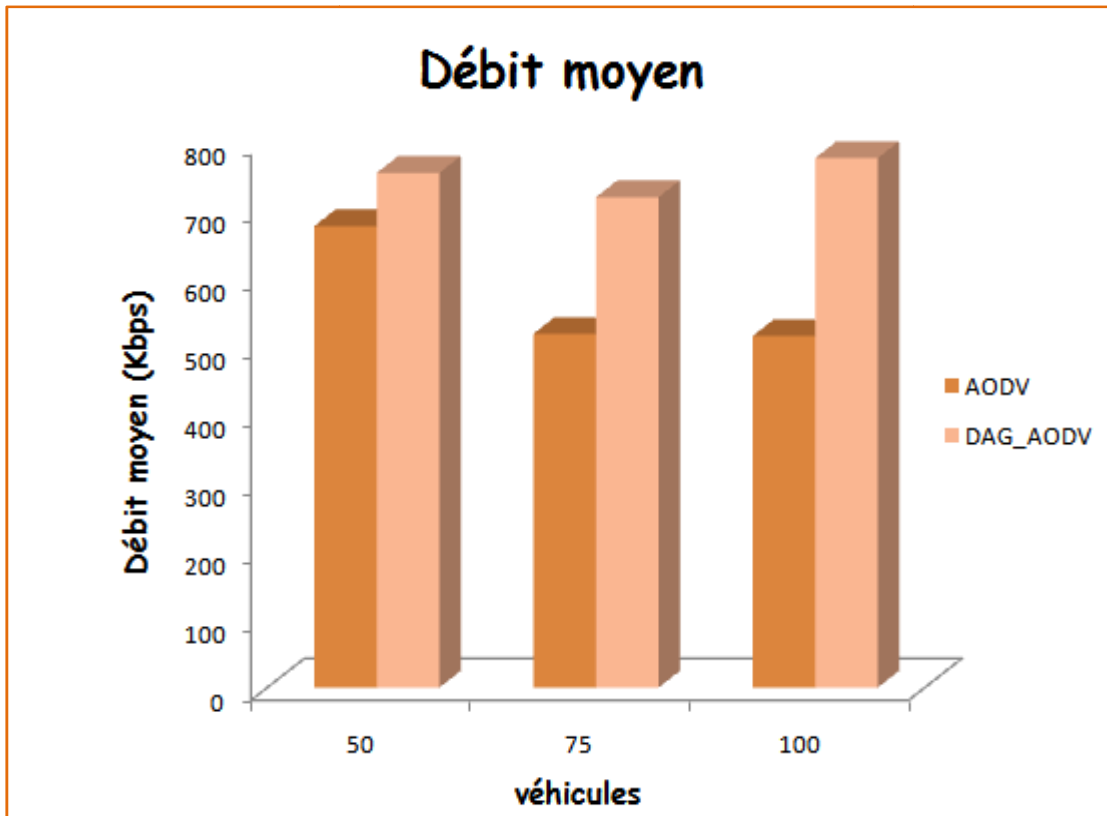


Figure 4.5 Débit moyen.

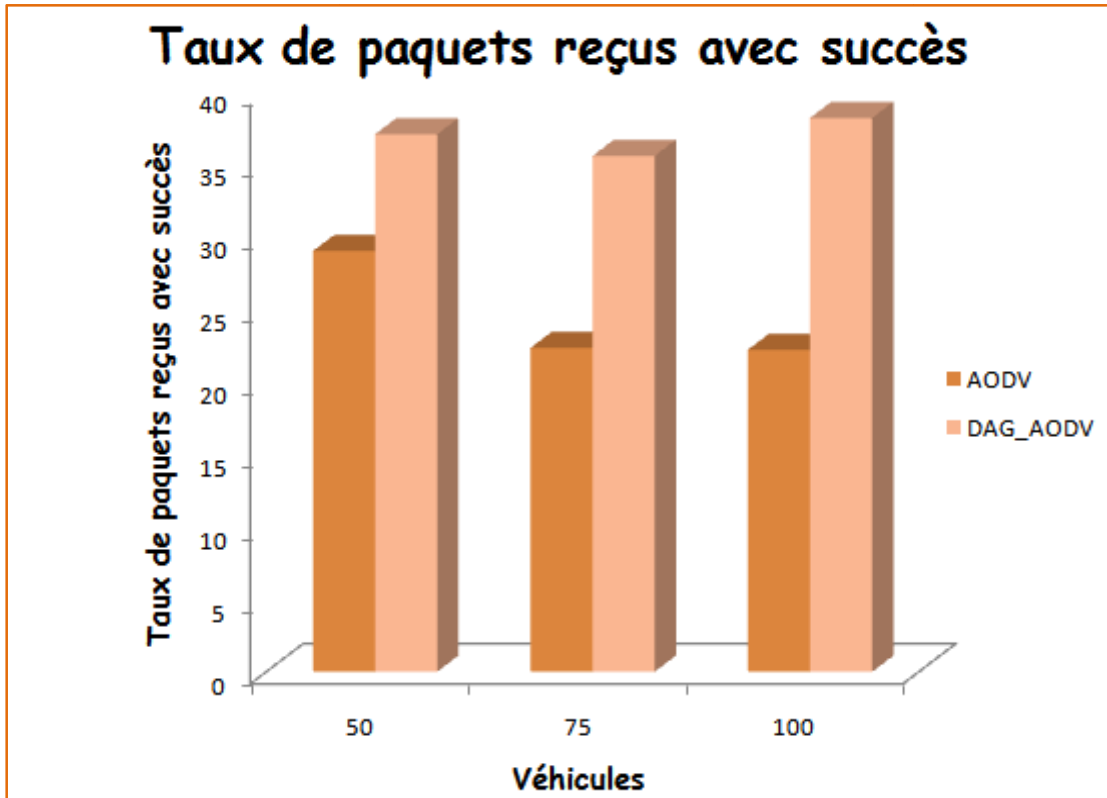


Figure 4.6 Taux de paquets reçus avec succès.

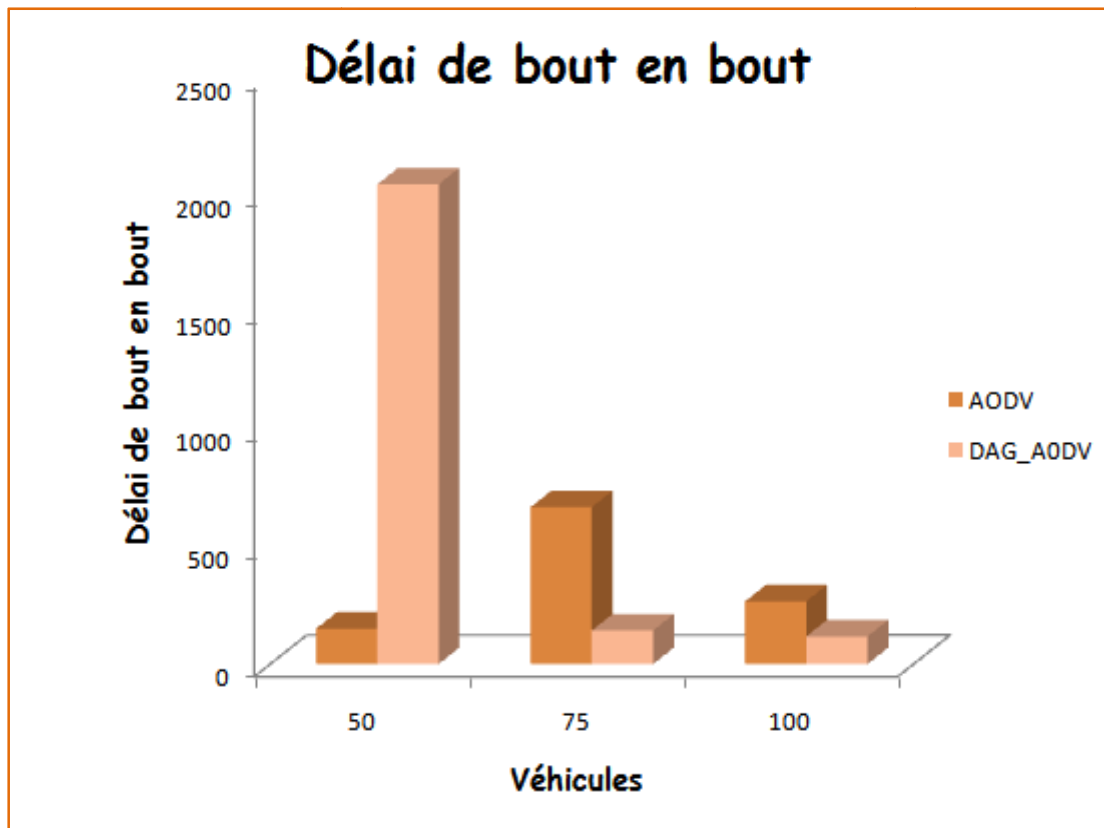


Figure 4.7 délai de bout-en-bout.

✓ Discussion

Nous avons présenté trois graphes chacun associé à une métrique, avant de finaliser cette section nous allons discuter les résultats que nous avons obtenus.

En commençant par le débit montré dans la figure 4.5, nous remarquons que notre protocole DAG_AODV offre de bons résultats par rapport à AODV traditionnel. Ces meilleurs résultats sont obtenus grâce à l'utilisation de la technique multi canaux et multi interface. Cette technique a aussi un impact positif sur le taux de paquets reçus avec succès comme nous voyons dans la figure 4.6. Alors, en comparant AODV avec DAG_AODV en termes de taux de paquets reçus avec succès, nous trouvons que notre protocole surpasse le protocole AODV.

La comparaison entre ces deux protocoles en termes de délai de bout en bout, comme présenté dans la figure 4.7, a montré qu'AODV généralement surmonte notre protocole. DAG_AODV utilise la diffusion stochastique, cette approche à mener d'avoir ce résultat. Cependant, lorsque le réseau est moins dense tel que le cas avec 50 véhicules le protocole AODV fournit un bon résultat par rapport notre protocole DAG_AODV. DAG_AODV a besoin de temps pour transmettre la donnée ce qui engendre une situation de perte de paquets.

4.7 Conclusion

A travers ce chapitre, nous avons présenté l'implémentation de notre amélioration du protocole AODV qui a été dénommée DAG_AODV. Au début, nous avons commencé par la définition de la simulation, et nous avons passé à définir nos outils pris en charge. Comme une phase essentielle nous avons expliqué tous les étapes réalisées pour implémenter notre protocole avec des pseudos codes. Après nous avons fait une comparaison en utilisant certain métriques d'évaluation de performance.

Finalement, la simulation de DAG_AODV avec les tests réalisés nous permet de déduire que notre protocole est plus performant qu'AODV traditionnel en termes de débit moyen, taux de paquets reçus avec succès et le délai de bout-en-bout. Donc on peut le considérer en tant qu'un protocole destiné à la collecte de données dans un environnement IoV.

Conclusion générale

L'internet des véhicules (IoV) représente aujourd'hui un intérêt certain pour l'industrie automobile, les opérateurs des réseaux, les organisations et même les particuliers. Le but de l'IoV, est d'améliorer la sécurité et l'efficacité des transports routiers afin de diminuer les accidents et fournir un environnement confortable aux conducteurs et à leurs passagers.

De plus en plus, les véhicules sont devenus « connectés », alors ils communiquent avec des applications mobiles, d'autres véhicules, des piétons etc. Qui dit véhicule « connectés » dit une quantité importante de données concernant le domaine de transport se génère d'une manière évolutive. Donc, l'une des exigences de base de l'IoV doit inclure la collecte de données.

Ce mémoire de master a eu pour objectif de proposer une solution adaptée à la collecte de données dans l'internet de véhicules afin de rendre ce domaine plus sûr, plus efficace et plus confortable.

La collecte de données repose sur l'échange de données entre les véhicules, ce qui permet de couvrir l'insuffisance de perception d'un seul véhicule sur son itinéraire. Grâce à l'agrégation de données, ces dernières sont devenues plus précises et plus pertinentes. Pour cela, l'objectif principale des protocoles de collecte de données est de définir comment les données collectées seront échangées entre les véhicules, afin de fournir des informations, en temps réel ou non, plus fiables et plus pertinentes.

Dans ce contexte, nous avons proposé un protocole de collecte et de dissémination de données appelé Data_Aggregation_AODV (DAG_AODV) capable de collecter et acheminer les données vers la destination requis, ce protocole est une amélioration du protocole AODV. Puisque il n'y a pas de références qui traitent jusqu'à ce jour, la collecte de données dans l'IoV, nous avons choisi d'améliorer d'AODV afin qu'on puisse collecter des données et avoir un mode de transport plus efficace et des routes plus sûres.

Les résultats de simulation ont montré que DAG_AODV a offert de bons résultats en termes de débit, délai de bout en bout et taux de paquets reçus avec succès en le comparant avec le protocole AODV traditionnel.

Bibliographie

- [1] Knut Evensen, Young-Jun Moon, *En route pour des transports connectés*, Organisation internationale de normalisation., 5 septembre 2016
- [2] Jean-françois JANIN *Systèmes de transports intelligents- Risques et opportunités* 10 mai 2013.
- [3] <http://www.matricis.com/fr/iot-analytique-avancee/systemes-de-transport-intelligents/>visité le 20/01/2018.
- [4] https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_transport_intelligent visité le 20/01/2018.
- [5] Ministère de la Transition écologique et solidaire *les systèmes de transport intelligents- L'expertise française-version française*, collection expertise française, 12 juillet 2017.
- [6] Jessica Murgia *Qu'est-ce que l'Internet des objets ?*, <https://www.androidpit.fr/>, 28 février 2017.
- [7] Jean Claude Tapia, Bruno Grossi, *La sécurité de l'internet des objets*.
- [8] Dave Evans, *L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?*, <https://www.cisco.com>, avril 2011.
- [9] Matthew N. O. Sadiku, Mahamadou Tembely, Sarhan M. Musa *Internet of Vehicles: An Introduction*, International Journals of Advanced Research in Computer Science and Software Engineering ISSN: 2277-128X (Volume-8, Issue-1), janvier 2018.
- [10] Juan Contreras-Castillo, Sherali Zeadally, Juan Antonio Guerrero Ibañez *A seven-layered model architecture for Internet of Vehicles*, Journal of Information and Telecommunication, 2 mars 2017.
- [11] Fangchun Yang, Jinglin Li, Tao Lei, Shangguang Wang, *Architecture and key technologies for Internet of Vehicles: a survey*, Journal of Communications and Information Networks, Posts & Telecom Press and Springer Singapore 2017
- [12] Juan Contreras-Castillo, Sherali Zeadally, and Juan Guerrero-Ibañez *Internet of Vehicles: Architecture, Protocols, and Security*, IEEE Internet of Things Journal
- [13] OMPRAKASH KAIWARTYA, ABDUL HANAN ABDULLAH, YUE CAO², AYMAN ALTAMEEM, MUKESH PRASAD⁴, CHIN-TENG LIN, XIULEI LIU *Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects*, IEEE ACCESS 7 September 2016.

- [14] https://www.cio-online.com/partner_zones/hp-bvex/article-la-voiture-connectee-implique-de-gros-traitements-de-donnees-66.html visité le 28/01/2018.
- [15] <http://www.telecom-evolution.fr/fr/actualites/sti-et-mobilite-des-nouvelles-competences-pour-de-nouveaux-metiers> visité le 15/02/2018.
- [16] Mingming Zhang, Tianyu Wo, Tao Xie, Xuelian Lin, Yaxiao Liu *CarStream: An Industrial System of Big Data Processing for Internet of Vehicles*, VLDB Endowment Homepage archive Volume 10 Issue 12, August 2017 Pages 1766-1777.
- [17] B. Fitzpatrick, *Distributed caching with Memcached*. Linux journal, 2004(124):5, 2004.
- [18] J. L. Carlson., *Redis in Action*, Manning Publications Co., 2013.
- [19] *Ganglia monitor system*. <http://ganglia.info/>.
- [20] Michael Reininger_, Seth Miller_, Yanyan Zhuang_y, Justin Cappos *A First Look at Vehicle Data Collection via Smartphone Sensors* 2015 IEEE Sensors Application Symposium (SAS).
- [21] *Sensibility Testbed*, <https://sensibilitytestbed.com/>.
- [22] J. Cappos, A. Dadgar, J. Rasley, J. Samuel, I. Beschastnikh, C. Barsan, A. Krishnamurthy, and T. Anderson, *Retaining sandbox containment despite bugs in privileged memory-safe code* in Proceedings of the 17th ACM conference on Computer and communications security, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 212–223. [Online]. Disponible: <http://doi.acm.org/10.1145/1866307.1866332>
- [23] *Sensibility Testbed*, Google PlayStore, <https://play.google.com/store/apps/details?id=com.sensibilitytestbed>.
- [24] *On-board diagnostics* <http://en.wikipedia.org/wiki/Onboard> diagnostics.
- [25] *obdlib*, <https://github.com/CyberAdmin/obdlib>.
- [26] *storesense*, <https://github.com/smmiller/sensevis/blob/master/storesense.r2py>.
- [27] *MongoDB Architecture*, <http://www.mongodb.com/mongodbarchitecture>.
- [28] *IMEI.info*, <http://www.imei.info/>.
- [29] Kundan Munja, and Shilpa Verma *INTERNET OF VEHICLES FOR INTELLIGENT TRANSPORTATION SYSTEM*, Springer International Publishing AG 2016
J.M. Corchado Rodriguez et al. (eds.), Intelligent Systems Technologies and Applications 2016, Advances in Intelligent Systems and Computing 530,
- [30] Juan Contreras-Castillo, Sherali Zeadally, and Juan Guerrero-Ibañez *Internet of Vehicles: Architecture, Protocols, and Security*, IEEE Internet of Things Journal 2017.
- [31] KundanMunjaj, Shilpa Verma *Internet of vehicles for intelligent transportation system*, Intelligent Systems Technologies and Applications 2016. ISTA2016. Springer, Cham.
- [32] Yoann Dieudonné, Bertrand Ducourthial, Sidi-Mohammed Senouci *Conception et expérimentation d'un protocole de collecte de données pour réseaux de véhicules*, CFIP

2011 - Colloque Francophone sur l'Ingénierie des Protocoles, May 2011, Sainte Maxime, France. 2011.

[33] Rapport STI – article 171 de la directive 2010/40/UE -
Août 2011 https://www.mobiliteintelligente.com/system/files/documents/2017/09/Rapport_STI_art_17-1_final-2-2.pdf

[34] Payal, RichaGupta *Enhancement in Priority Queuing using NS2*, International Journal for Advance Research in Engineering and Technology 2014.

[35] Salim Bitam, Mohamed Batouche, Abdelhamid Mellouk *QoS BeeManet: A new QoS multipath routing protocol for mobile ad-hoc networks* GLOBCOM Workshops (GC wkshps), 2010 IEEE.

[36] <https://github.com/softvar/ns2-roadv> visité le 12/05/2018.

[37] AditiMandap, DeeptiTheng *Multi interface multi channel and improved AODV routing protocol*, Information communication and Embedded Systems (ICICES), 2014conference on

[38] sapanaGarde, PrashantPurohit *Multi-interface Ad-hoc Wireless Network with AODV*, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 9, Septembre2013.

[39] Parma Nand, S.C.sharma *Probability Based Improved Broadcasting for AODV Routing Protocol* computational Intelligence and Communication Networks (CICN), 2011 International Conference on, IEE Xplor.

[40] TeerawatIssariyakul, EkramHossain *Introduction to Network Simulator NS2*, second edition, Springer.

[41] Kevin Fall, KannanVaradhan *The ns Manual (formerly ns Notes and Documentation)*, November 4, 2011.

[42] site web "<http://sumo.dlr.de/index.html>" visité le 20/05/2018.

[43] Olivier Rivaton *Le routage de l'information dans les réseaux véhiculaires mobiles*, 2016.

[44] GregorLaemmel *Simulation of Urban Mobility-SUMO*, EclipseNewsletter, Août 2017.

[45] <http://itantenna.blogspot.com/2012/01/add-multiple-interfaces-and-multiple.html> visité le 20/04/2018.

[46] Ramon agueroCalvo, Jesus Pérez Campo *Adding Multiple Interface Support in NS-2*, janvier 2017.