

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingéniorat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Télécommunication
Spécialité : Réseaux et Télécommunication

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Implémentation du modèle SDN dans les réseaux VANET

Présenté par : *Bendif Nour el houda - Bensalem Rania*

Encadrant : *Hafs Toufik*

MCA

UBM Annaba

Jury de Soutenance :

Zermi Narima	MCA	UBM Annaba	Président
Hafs Toufik	MCA	UBM Annaba	Encadrant
Taibi Mahmoud	Professeur	UBM Annaba	Examineur

Année Universitaire : 2020/2021

Remerciements

Nous remercions dieu « *Allah* » le tout puissant de nous avoir donné la santé, le courage, la volonté et la patience pour achever notre projet de fin d'études.

Nos sincères remerciements et notre profonde gratitude s'adressent à notre encadreur Monsieur *Hafs Toufik*, pour sa disponibilité, son encouragement et sa patience a notre rencontre.

Nous tenons à remercier sincèrement les membres du jury qui nous font le grand honneur d'évaluer notre travail ainsi que *Mme ZERMI Narima* qui préside le jury et *Mr TAIBI Mahmoud* examinateur du projet.

Notre profonde reconnaissance envers *nos parents* pour leur soutien indéfectible aussi bien moral que financier durant toutes les années d'études

Nous tenons également à exprimer toute notre reconnaissance à *nos sœurs, nos frères* et tous *nos proches* et *nos amis* qui ont contribué à la réalisation de ce mémoire.

Enfin, nos remerciements vont à toute notre promo et a toute personne ayant contribué de loin ou de près à l'élaboration de ce travail.

Résumé

Propulsés par les longs embouteillages et les nombreux accidents de la route, les réseaux véhiculaires (Vehicular Ad-Hoc Network, VANET) ont émergés dans le but de rendre le voyage plus agréable, la route plus sûre et le système de transport plus efficace. De nos jours, les architectures des réseaux de véhicules souffrent de problèmes d'évolutivité, car il est difficile de déployer des services à une grande échelle. Ces architectures sont rigides, difficiles à gérer et souffrent d'un manque de flexibilité et d'adaptabilité à cause de l'hétérogénéité des technologies véhiculaires. Ces contraintes limitent les fonctionnalités du système, ralentissent la créativité et conduisent souvent à une sous-exploitation des ressources de réseau.

Au cours des dernières années, le paradigme émergent de l'architecture des réseaux de SoftwareDefined Networking (SDN), est devenu l'une des technologies les plus importantes pour la gestion des réseaux à grande échelle tels que les réseaux de véhicules. Le SDN est principalement basé sur une séparation physique entre le plan de contrôle et le plan de données et un contrôle et une intelligence logiquement centralisés dans un contrôleur logiciel. Récemment, plusieurs travaux ont montré que l'intégration du paradigme de SDN dans les VANETS permet d'apporter la flexibilité, la programmabilité et mieux supporter l'évolutivité.

L'objectif de ce travail est de :

- Mise en œuvre d'un réseau SDVANET simulé par Mininet-wifi.
- assurer l'accessibilité de réseau SDVANET avec un contrôleur SDN.

Mots clés : Réseaux véhiculaires VANET, SDN software defined network, réseau SDVANET, simulation, Mininet-wifi, contrôleur SDN.

Abstract

Driven by long traffic jams and numerous road accidents, Vehicular Ad-Hoc Networks (VANETs) have emerged with the aim of making the journey more pleasant, the road safer and the transport system more efficient. Nowadays, vehicular network architectures suffer from scalability problems, as it is difficult to deploy services on a large scale. These architectures are rigid, difficult to manage and suffer from a lack of flexibility and adaptability due to the heterogeneity of vehicle technologies. These constraints limit system functionality, slow down creativity and often lead to under-exploitation of network resources.

During the last years, the emerging network architecture paradigm of Software-Defined Networking (SDN) has become one of the most important technologies for managing large-scale networks such as vehicular networks. SDN is mainly based on a physical separation between the control plane and the data plane and a logically centralised control and intelligence in a software controller. Recently, several works have shown that the integration of the SDN paradigm in VANETS can bring flexibility, programmability and better support scalability.

The objective of this work is to:

- Implementing an SDVANET network simulated by Mininet-wifi.
- Ensure the accessibility of SDVANET network with a SDN controller.

Key words: VANET vehicular networks, SDN software defined network, SDVANET network, simulation, Mininet-wifi, SDN controller.

ملخص

ان الاختناقات المرورية الطويلة و العديد من حوادث الطرق ادت الى ظهور شبكات السيارات و ذلك بهدف جعل الرحلة اكثر متعة و الطريق آمنة و نظام النقل اكثر كفاءة . اليوم تعاني هياكل شبكة السيارات من مشكلة قابلية التوسع و ذلك لصعوبة نشر الخدمات على نطاق واسع كذلك يصعب ادارتها لانها بنى جامدة تعاني من نقص المرونة و عدم القدرة على التكيف بسبب غياب تجانس تقنيات السيارات و هذه القيود تحد من وظائف النظام و تبطيء الابداع و غالبا ما تؤدي الى نقص استغلال الموارد الشبكة. اليوم تعاني هياكل شبكة السيارات من مشكلة قابلية التوسع و ذلك لصعوبة نشر الخدمات على نطاق واسع كذلك يصعب ادارتها لانها بنى جامدة تعاني من نقص المرونة و عدم القدرة على التكيف بسبب غياب تجانس تقنيات السيارات و هذه القيود تحد من وظائف النظام و تبطيء الابداع و غالبا ما تؤدي الى نقص استغلال الموارد الشبكة

في هذه السنوات الاخيرة اصبح النموذج الناشئ لهندسة الشبكات المعرفة بالبرمجيات من اهم التقنيات المستعملة لادارة الشبكات و اسعه النطاق مثل شبكات السيارات. تعتمد هذه الأخيرة بشكل اساسي على الفصل بين مستوى التحكم ومستوى البيانات, اما التحكم المركزي والذكاء المنطقي ففي وحده التحكم البرمجية. في الاونة الاخيرة اثبتت العديد من الاعمال ان تكامل نموذج الشبكات المعرفة بالبرمجيات في شبكات السيارات يجعل من الممكن توفير المرونة ,قابلية البرمجة وايضا قابلية التوسع بشكل افضل .

الهدف من هذا العمل هو:

- تنفيذ شبكة SDVANET تمت محاكاتها بواسطة Mininet-wifi.

- ضمان إمكانية الوصول إلى شبكة SDVANET باستخدام وحدة تحكم SDN.

الكلمات الرئيسية:

شبكات المركبات VANET ، الشبكة المعرفة ببرمجيات SDN ، شبكة SDVANET ؛ محاكاة ، Mininet-

wifi؛ تحكم SDN

Liste des figures

Chapitre I :

Figure I.1:	Echange des données en mode infrastructure et en mode ad hoc.....	4
Figure I.2:	Exemple d'un réseau MANET	4
Figure I.3:	Exemple d'un réseau VANET.....	5
Figure I.4:	Mode de communication dans un réseau VANET.....	7
Figure I.5:	Exemple d'un message d'alerte.....	8
Figure I.6:	Exemple d'un véhicule intelligent	9
Figure I.7:	La classification des protocoles de routage dans les réseaux VANET...	10
Figure I.8:	Prévention d'une congestion	11
Figure I.9:	Exemple d'application de confort.....	12
Figure I.10:	l'architecture typique d'un réseau SDN.....	13
Figure I.11:	Le protocole Openflow.....	15
Figure I.12:	Comparaison entre les réseaux traditionnels et SDN.....	16

Chapitre II

Figure II.1:	Random Direction.....	23
--------------	-----------------------	----

Chapitre III

Figure III.1:	Exemple de mininet-wifi graph.....	29
Figure III.2:	les bibliothèques de mininet-wifi	29
Figure III.3:	La création du contrôleur	30
Figure III.4:	La création des nœuds.....	30
Figure III.5:	La création d'un point d'accès	30
Figure III.6:	La configuration des nœuds et du modèle de mobilité.....	31
Figure III.7:	les commandes qui permettent le démarrage du réseau et du contrôleur	31

Figure III.8:	Le lancement de la topologie	31
Figure III.9:	Visualisation du Graph à l'instant T1	32
Figure III.10:	Visualisation du Graph à l'instant T2	32
Figure III.11:	L'affichage des nœuds.....	33
Figure III.12:	L'affichage des liens du réseau.....	33
Figure III.13:	L'affichage des interfaces.....	33
Figure III.14:	l'affichage les adresses IP.....	33
Figure III.15:	Ping entre le point d'accès et car1.....	34
Figure III.16:	Ping entre car4 et car5.....	34
Figure III.17:	Le lancement du contrôleur POX.....	35
Figure III.18:	Le lancement de la topologie avec le contrôleur POX.....	35
Figure III.19:	Le contrôleur POX est connecté.....	36
Figure III.20:	Ping entre une station et point d'accès.....	36
Figure III.21:	Ping entre deux stations.....	36
Figure III.22:	Le contrôleur POX est déconnecté.	37
Figure III.23:	Ping entre deux stations après la déconnexion du POX.....	37
Figure III.24:	Le lancement du contrôleur Ryu.....	37
Figure III.25:	Le lancement de la topologie avec le contrôleur Ryu.....	38
Figure III.26:	Le résultat dans le 1er terminal après le lancement de la topologie.....	38
Figure III.27:	Ping entre deux stations avec le contrôleur Ryu.....	39
Figure III.28:	Les paquets sont transmis.....	39
Figure III.29:	la commande pour quitter le processus.....	40
Figure III.30:	Le contrôleur Ryu est déconnecté.....	40
Figure III.31:	Ping entre deux stations après la déconnexion de contrôleur Ryu.....	40

Liste des tableaux

Tableau I.1:	Tableau comparatif entre les réseaux traditionnels et SDN.....	17
Tableau II.2:	Les différents modèles de mobilité.....	22
Tableau III.3 :	Comparaison entre le contrôleur POX et Le contrôleur Ryu.....	41

Liste des équations :

Équation II.1: Taux de livraison de paquets PDR.....	25
Équation II.2: Le délai de bout en bout.....	26
Équation II.3: Les paquets perdus.....	26
Équation II.4: Le débit moyen.....	26

Liste des abréviations :

VANET : Vehicular Ad hoc Network

MANET : Mobile Ad hoc Network

SDN : Software Defined Networking

RSU : Road Side Unit

OBU : On Board Unit

CA : Certificate Authority

V2V : communications véhicule à véhicule

V2I : communications véhicule à infrastructure

GPS : Global Positioning System

AODV: Ad hoc On-Demand Distance Vector

QoS : Quality Of Service

API : Application Programming Interface

RD : Random Direction

PDR : Packet Delivery Ratio

Sommaire

Introduction Général.....	1
Chapitre I : Généralités	2
1 - Introduction	3
2 - Les réseaux Ad-hoc	3
3 - Les réseaux mobiles ad hoc (MANET).....	4
4 - Les réseaux ad hoc véhiculaires (VANET).....	5
4.1 - Architecture d'un réseau VANET	5
4.1.1 - Entités de communication.....	5
• Road Side Unit.....	6
• On Board Unit.....	6
• Certificate Authority.....	6
4.1.2 - Les modes de communication dans VANET.....	6
• Communications Véhicule à infrastructure (V2I).....	6
• Communications Véhicule à Véhicule (V2V).....	7
• Mode hybride.....	7
4.1.3 - Les types de messages.....	8
• Les messages de contrôle.....	8
• Messages d'alerte.....	8
4.1.4 - Nœud d'un réseau VANET.....	9
4.2 - Les caractéristiques d'un réseau VANET.....	9
4.3 - Les protocoles de routage.....	10
4.4 - Les applications.....	11
• Applications de gestion du trafic routier.....	11
• Applications de confort.....	11
• Application de sécurité du trafic routier.....	12
5 - Le Software Defined Networking	12
5.1 - Définition du SDN.....	12
5.2 - Architecture d'un réseau SDN:.....	13
5.2.1 - Les couches	14

• La couche plan de données.....	14
• La couche de contrôle.....	14
• La couche application	14
5.2.2 - Les interfaces de communication.....	14
• Interfaces Sud.....	14
• Interface Nord	14
• Interface Est/Ouest.....	14
5.3 - Les composants du SDN.....	15
5.3.1 - Le protocole Openflow.....	15
5.3.2 - Les contrôleurs SDN.....	15
5.3.3 - Quelques contrôleurs SDN.....	15
5.4 - Comparaison entre les réseaux traditionnels et SDN.....	16
5.5 - Les avantages du SDN	17
6 - Conclusion	19
Chapitre II : Architecture du système étudié	20
1 – Introduction.....	21
2 - Le modèle de mobilité	21
2.1 - les différents modèles de mobilité.....	21
• Random Direction (RD).....	23
3 - Les paramètres de travail dans le domaine des réseaux VANET.....	23
3.1 -Localisation de véhicule.....	23
3.2 -Problèmes de congestion :	24
3.3 -Dynamique du trafic véhiculaire dans les VANET.....	24
3.4 -Le routage dans les VANET.....	24
3.5 -La sécurité dans les VANET.....	25
3.6 -La qualité de service (QoS).....	25
4 - Les critères de performances.....	25
4.1 - Taux de livraison de paquets.....	25
4.2 - Délai de bout en bout.....	25
4.3 - Paquets perdus.....	26
4.4 - Débit moyen.....	26
5 – Conclusion.....	26

Chapitre III : Résultat et discussions	27
1 – Introduction.....	28
2 - Composants et outils logiciels.....	28
2.1 – Mininet.....	28
2.2 - Mininet-wifi.....	28
3- Les étapes de simulation.....	29
3.1 - La création de la topologie.....	29
3.2 -Le lancement de la topologie.....	31
3.3 - Les commandes principales de topologie Mininet-wifi.....	32
3.4 - SDVANET avec le contrôleur POX.....	35
3.5 - SDVANET avec le contrôleur Ryu.....	37
4- Discussion de simulation.....	41
4.1 - L’accessibilité du réseau :.....	41
4.2 - La comparaison entre le contrôleur POX et Le contrôleur Ryu.....	41
5 – Conclusion.....	42
Conclusion général.....	43

Introduction générale :

De nos jours, nous passons de plus en plus de temps dans les transports, que ce soit dans des véhicules personnels ou dans des transports en commun. De plus, nos usages des moyens de communication sont devenus de plus en plus nomades surtout avec les grandes avancées des technologies d'information et de communication (TIC).

Le concept de réseaux ad hoc de véhicules a donc vu le jour suite à cette évolution dans le but d'offrir une large variété de services, allant de l'amélioration de la sécurité routière à l'optimisation du trafic, en passant par le divertissement du conducteur et des passagers. En effet, ce type de réseau se caractérise principalement par la haute mobilité des nœuds, la topologie extrêmement dynamique et la grande échelle de réseau. Les réseaux de véhicules représentent une projection des systèmes de transports intelligents (Intelligent transportation Systems. ITS), dans lesquels les véhicules sont capables de communiquer les uns avec les autres et avec aussi les infrastructures le long des routes.

Les réseaux de véhicules souffrent de plusieurs problèmes, notamment un problème d'encombrement, un délai de bout en bout accru en cas de messages de sécurité, etc. Dans cet article, nous avons tenté de fournir l'architecture Software Defined Network (SDN) contrôlée, dont il est open source et facilement programmable. La séparation du plan de contrôle et du plan de données dans SDN facilite la gestion et la programmation des réseaux VANETS. Le contrôleur SDN est le cerveau du système, il est responsable de la gestion globale et du contrôle efficace des ressources du réseau. En utilisant le contrôleur SDN, les performances globales du réseau sont améliorées et optimisées, il offre plusieurs avantages, notamment une sélection de chemin efficace, un contrôle de la congestion évitant et une communication rapide et fiable. Le plan de données dans SDN est uniquement responsable de la transmission des paquets, la connexion entre ces appareils est soit une connexion filaire, soit une connexion sans fil. Le modèle proposé de SDN VANET se compose de Contrôleur SDN logiquement centralisé qui contrôle les performances globales du réseau qui communique avec les périphériques de transfert à l'aide du protocole OpenFlow.

Ce mémoire étudie les manières permettant de profiter des points forts du SDN pour faciliter la gestion et améliorer les performances des prochaines générations des réseaux de véhicules. Il est organisé en trois chapitres : dans le premier chapitre, on va présenter des généralités liées aux réseaux VANET. Le deuxième chapitre montre l'architecture de notre système et le dernier chapitre est dédié à la présentation des résultats obtenus .

Chapitre I :

Généralités

1 - Introduction :

Les réseaux véhiculaires représentent aujourd'hui l'un des éléments de base sur lesquels vont se fonder les systèmes de transport intelligents. Les réseaux VANET ont reçu un grand intérêt de la majorité des projets des unités industrielles et les organisations de recherches. En raison de leurs caractéristiques comme la mobilité prévisible et forte, la topologie dynamique élevée. Ceci reflète une importante évolution dans le domaine des réseaux VANET due essentiellement d'un côté au nombre élevé des accidents recensés chaque année et de l'autre aux besoins actuels en termes de sécurité routière ou confort de conducteur.

Le paradigme Software-Defined Networking appliqué aux VANET, son principe est de sortir la partie intelligente des équipements d'interconnexions, et la placer vers un seul point de contrôle appelé contrôleur, ce dernier fournit une vue centrale de réseau ,ce qui simplifie d'une part, la gestion et la configuration de réseau ,il présente donc plusieurs avantages, tel que servir plusieurs domaines, et être intégré avec les nouvelles technologies, en offrant ainsi une programmabilité et une vue globale, centralisé du réseau.

Dans ce chapitre, nous présentons l'état de l'art sur les réseaux VANET et le paradigme SDN. Commençant par définir les réseaux ad hoc, MANET et VANET ainsi que l'architecture du VANET , les caractéristiques des réseaux VANET, les protocoles de routage dans les VANET et les applications. Puis on va définir le SDN, son architecture, ensuite nous traitons les différents composants d'un réseau SDN et ces avantages.

2 - Les réseaux Ad-hoc :

Les réseaux Ad-hoc sont des réseaux sans-fil capables de s'organiser spontanément et de manière autonome dans leur environnement. La tâche de la gestion du réseau est répartie sur l'ensemble d'entités communicantes par liaison sans-fil, ces entités sont souvent appelées «nœuds». Les nœuds proches effectuent une communication directe entre eux, tandis que les autres nœuds jouent le rôle d'intermédiaire pour transporter le message jusqu'à sa destination finale.

Dans un réseau Ad-hoc on rencontre plusieurs problèmes comme l'absence d'infrastructure, une bande passante limitée, beaucoup de perte de données, des erreurs de transmission, et plusieurs problèmes de sécurité [1].

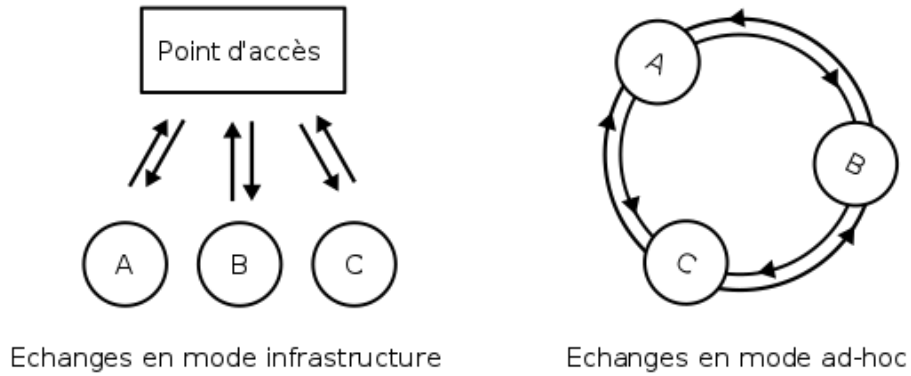


Figure I.1: Echange des données en mode infrastructure et en mode ad hoc [3].

3 - Les réseaux mobiles ad hoc (MANET):

Un réseau mobile ad hoc, appelé généralement Mobile Ad hoc Network (MANET), est un ensemble de nœuds mobiles qui se déplacent dans un territoire quelconque d'une manière autonome et coopérative, sans l'utilisation d'une infrastructure préexistante ou d'une administration centralisée. Les ondes radio qui se propagent entre les différents nœuds mobiles sont le seul moyen de communication. Dès qu'un ensemble de nœuds mobiles se trouve à portée radio les uns des autres, alors le réseau se forme spontanément, mais de manière provisoire [2].

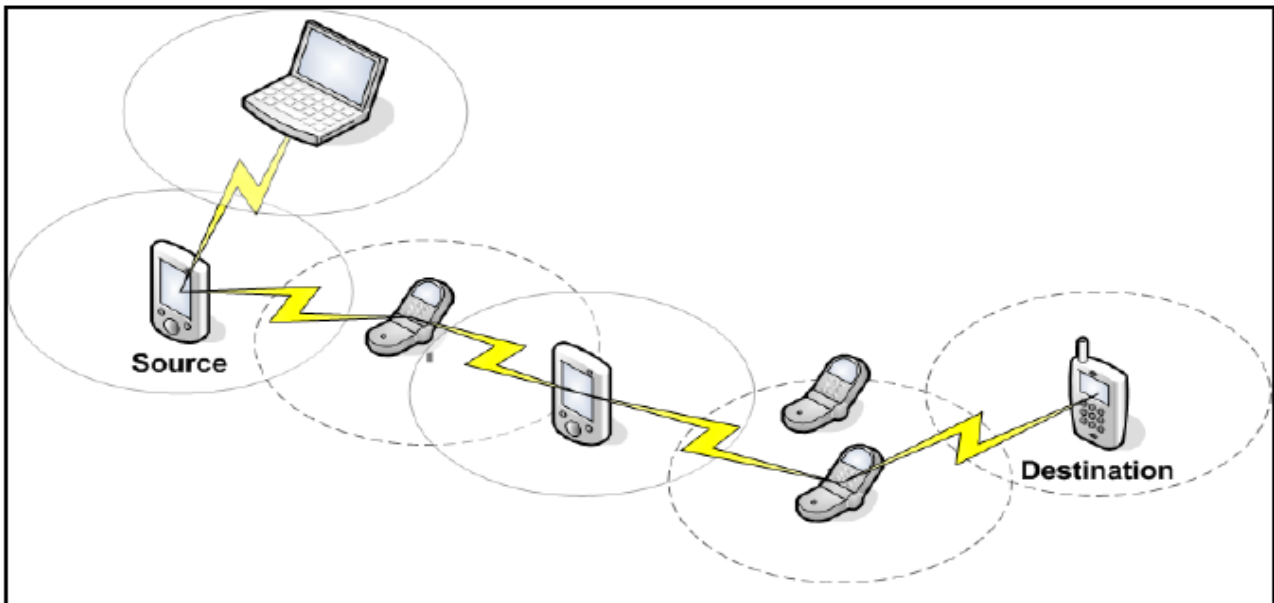


Figure I.2: Exemple d'un réseau MANET [4].

4 - Les réseaux ad hoc véhiculaires (VANET):

Les réseaux VANET constituent une nouvelle forme des réseaux MANET; ils sont caractérisés par une forte mobilité des nœuds rendant la topologie du réseau fortement dynamique par rapport à un réseau ad hoc classique où les nœuds mobiles sont des véhicules intelligents équipés de calculateurs, de cartes réseau et de capteurs.

Comme tout autre réseau ad hoc, les véhicules peuvent communiquer entre eux (pour échanger les informations sur le trafic par exemple) ou bien avec une infrastructure située aux bords de routes (pour demander des informations ou accéder à internet.) [1].

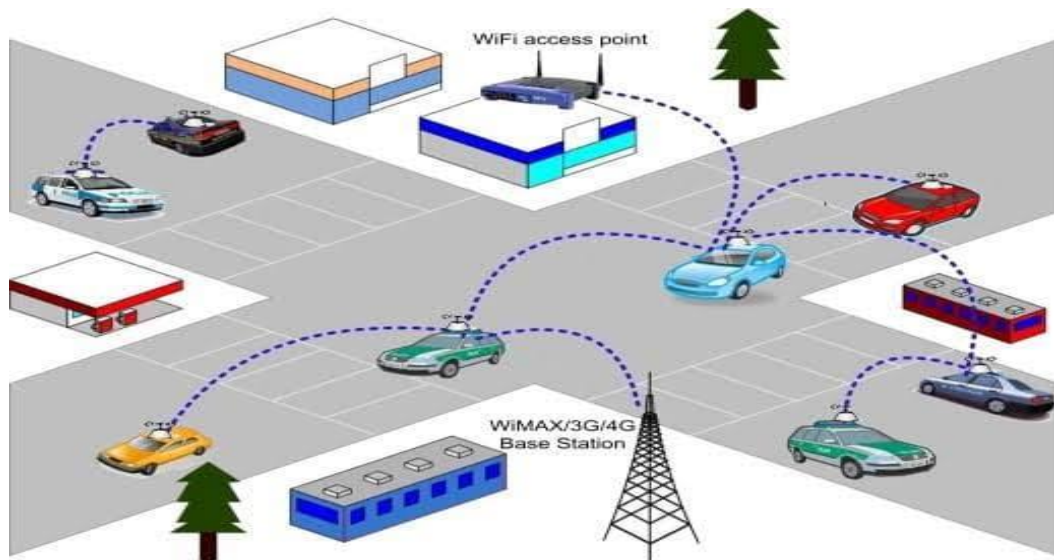


Figure I.3: Exemple d'un réseau VANET.

4.1 - Architecture d'un réseau VANET :

4.1.1 - Entités de communication:

- **Road Side Unit:**

Les Road Side Unit (RSU) sont des entités situées et installées au bord de la route. Ces entités présentent des points d'accès au réseau et sont déployées tout au long de la route. Elles peuvent être principalement, des feux de signalisation, des lampadaires ou autres. Chaque RSU a pour objectif de transmettre des messages aux véhicules qui se trouvent dans sa zone radio. Ces messages contiennent des informations sur les conditions météorologiques, ainsi que sur l'état de la route (vitesse maximale, autorisation de dépassement, etc.) [5] [6].

- **On Board Unit:**

Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres) .Leurs rôles sont de permettre aux véhicules de se localiser, recevoir, calculer, enregistrer et envoyer des messages sur une interface réseau à l'aide d'un ensemble de programmes.

Dans le réseau VANET, le conducteur ou l'utilisateur peut voir les pseudonymes des véhicules à proximité dans son OBU à l'aide des messages beacon. Ainsi, l'utilisateur peut choisir le véhicule avec lequel il veut communiquer [5] [6].

- **Certificate Authority:**

C'est une source d'authenticité de l'information. Elle contrôle l'ensemble du réseau, assure la gestion et la mise à jour de toutes les entités dans le réseau (RSU, véhicule). La CA devrait connaître toutes les vraies identités des véhicules et au besoin les divulguer pour les forces de l'ordre .De plus , dans certains ouvrages, CA est responsable de la délivrance et de l'attribution des certificats de communication et des pseudonymes [6] .

4.1.2 - Les modes de communication dans VANET :

Dans les réseaux VANET, on trouve principalement, les entités fixes qui constituent l'infrastructure (RSU et CA) et les entités mobiles (les véhicules). Pour pouvoir échanger les différentes informations et données liées à la sécurité et au confort des usagers de la route, ces différentes entités doivent établir des communications entre elles. Pour cette raison, on distingue les types de communications véhicule à véhicule (V2V) ; véhicule à infrastructure (V2I) et la communication hybride (V2V-V2I) [7].

- **Communications Véhicule à infrastructure (V2I):**

Dans ce mode, les véhicules communiquent avec l'infrastructure fixe du réseau, les RSUs, par l'intermédiaire d'une communication V2I.Ce mode de communication permet une meilleure utilisation des ressources partagées et il démultiplie les services fournis aux passagers concernant le trafic. L'inconvénient majeur de ce mode de communication est que le déploiement des RSUs le long des routes est une tâche coûteuse et prend beaucoup de temps, sans oublier les coûts relatifs à leur maintenance [7].

- **Communications Véhicule à Véhicule (V2V):**

Ce mode de communication représente une architecture ad hoc où les véhicules communiquent les uns avec les autres par l'intermédiaire d'une communication V2V. En effet, un véhicule peut communiquer directement avec un autre véhicule s'il se situe dans sa zone de couverture radio, ou bien par le biais d'un protocole multi sauts qui se charge de transmettre les messages en passant par les véhicules voisins comme des relais.

Ce mode de communication fonctionne suivant une architecture décentralisée et garantit une communication moins coûteuse et plus flexible avec une petite latence. En effet, les communications V2V sont très efficaces pour le transfert des alertes relatives aux services liés à la sécurité routière. Toutefois, elles ne garantissent pas une connectivité permanente entre les véhicules et souffrent de fréquentes déconnexions dues à la forte mobilité des véhicules [7].

- **Mode hybride :**

Ce mode combine les deux premiers modes. Il utilise les communications V2V pour étendre la zone de couverture limite des infrastructures au bord de la route. Ce mode a un grand intérêt économique puisqu'il permet de diminuer les coûts de déploiement des unités tout au long des routes il permet de bénéficier des faibles délais des communications V2V par conséquent, ce type de communication est le plus utilisé, parce qu'il permet de bénéficier des avantages des deux modes précédents [7].

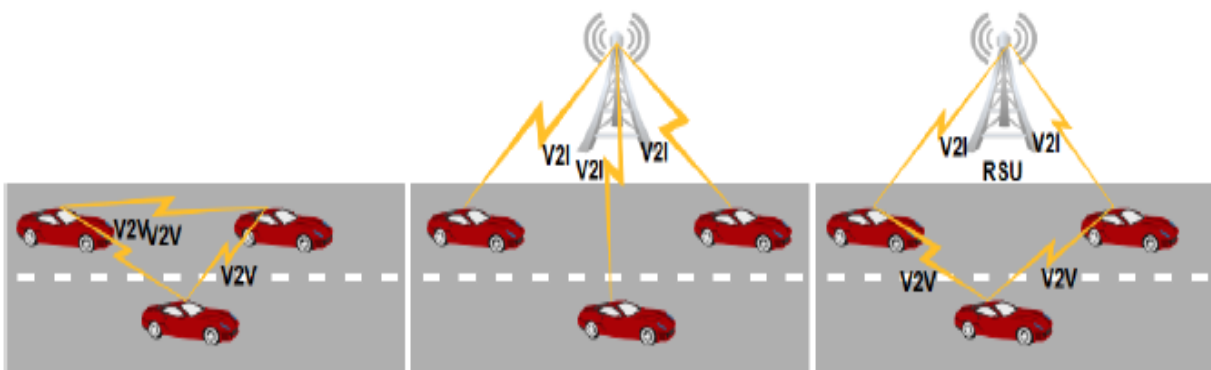


Figure I.4: Mode de communication dans un réseau VANET [7].

4.1.3 - Les types de messages:

Les différents messages échangés dans les réseaux VANET peuvent être facilement classés, selon leurs utilités et leurs contenus:

- **Les messages de contrôle:**

Ce sont les messages « Beacon ». Ce type de message contient souvent des informations relatives à l'identité et à l'état actuel du véhicule (Position, vitesse, direction et autres), il est diffusé périodiquement et est utilisé principalement pour permettre l'identification du voisinage. Ce type de message joue un rôle primordial dans la plupart des protocoles de routage et de sécurité [6].

- **Messages d'alerte:**

Les messages d'alerte ou les messages de sécurité sont envoyés lors d'une situation dangereuse ou lors d'un événement méritant l'attention du conducteur. Dans le cas d'un accident par exemple (**Figure I.5**), le message d'alerte sert à prévenir les véhicules qui se dirigent vers la zone de l'accident ou de congestion de ce fait.

Ces messages sont urgents et importants et leurs tailles sont petites afin de garantir leur transmission plus rapidement. Les véhicules désignés par la retransmission des messages d'alerte doivent les transmettre dès réception. Les messages d'alerte contiennent les coordonnées du lieu en question [5].

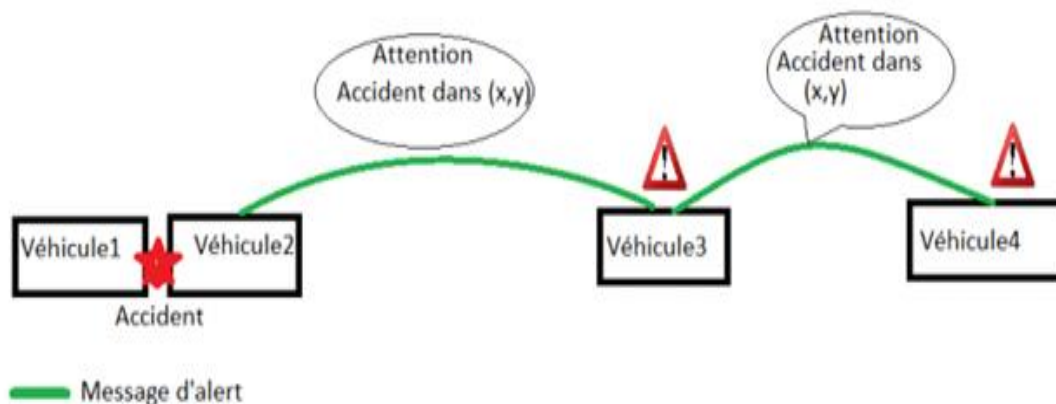


Figure I.5: Exemple d'un message d'alerte [5].

4.1.4 - Nœud d'un réseau VANET :

Un nœud d'un réseau VANET est un véhicule intelligent équipé d'appareils électroniques installés permettant des communications avec les autres véhicules ou avec l'infrastructure. Ils possèdent de nombreux capteurs et unités de calcul à bord permettant de gérer et traiter les informations reçues [3].

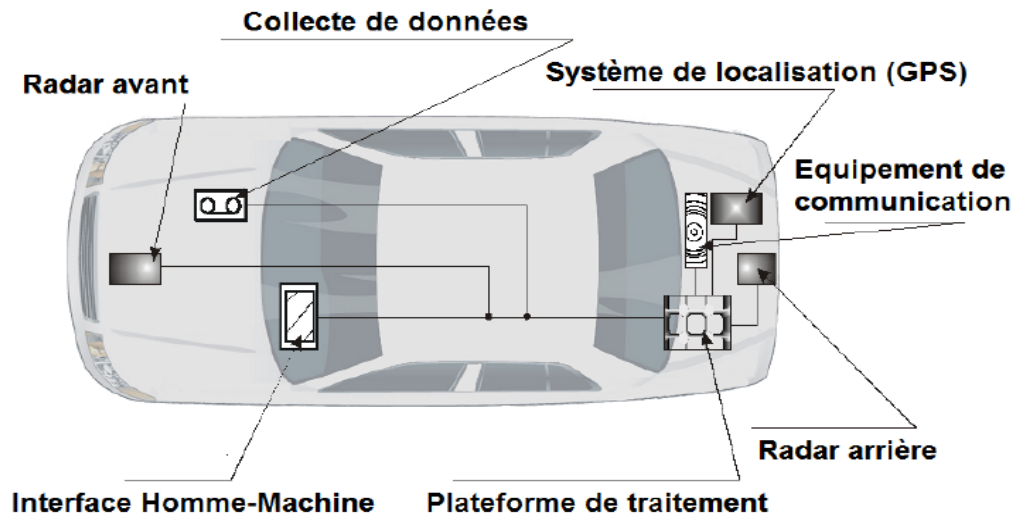


Figure I.6: Exemple d'un véhicule intelligent [1].

4.2 - Les caractéristiques d'un réseau VANET :

- **Forte mobilité et topologie du réseau:**

Les réseaux VANET se caractérisent par une mobilité extrêmement élevée. Cette dernière peut être affectée par plusieurs facteurs comme la vitesse des nœuds, le comportement des conducteurs sur les routes ainsi que les infrastructures routières. Cette forte mobilité des nœuds cause des changements rapides de la topologie du réseau [16] [17].

- **Connectivité:**

Comme les réseaux VANET se caractérisent par la forte topologie dynamique, alors, la connectivité est de courte durée surtout lorsque la densité des véhicules est très faible. Afin d'améliorer la connectivité, il faut un déploiement de plusieurs nœuds relais ou points d'accès le long de la route, ce qui permettrait la retransmission de l'information sur de longues distances [16].

- **localisation:**

Afin de localiser et de faciliter la communication entre les différentes entités du réseau, des systèmes de localisation par satellite comme les GPS sont utilisés dans les réseaux VANET [16].

- **Capacité et autonomie d'énergie:**

Parmi les contraintes les plus importantes lors d'un traitement dans les réseaux ad hoc mobiles sont la contrainte d'énergie, par contre dans un réseau VANET, les véhicules ne souffrent pas de cette contrainte vu qu'ils n'ont pas de limite en termes d'énergie et ils disposent suffisamment d'énergie qui peut alimenter les différents équipements électroniques d'une voiture intelligente. Donc, les nœuds sont censés avoir une grande capacité de traitement et de stockage de données [15] [17].

- **Modèle de communication:**

Les types de communication se basent sur la diffusion des messages de prévention ou d'alerte d'une source à une ou plusieurs destinations et on l'appelle communication broadcast. Aussi, une communication unicast peut être établie entre les entités [16].

4.3 - Les protocoles de routage :

L'objectif principal du protocole de routage est de fournir des chemins optimaux entre les nœuds du réseau avec un temps système minimal. Beaucoup des protocoles de routage ont été développés pour les réseaux VANET; nous distinguons deux classes de protocoles de routage: les protocoles basés sur la topologie qui est divisée en protocoles proactifs, réactifs et hybrides et les protocoles basés sur la localisation (géographique) qui utilisent la position physique des nœuds mobiles pour configurer le routage.

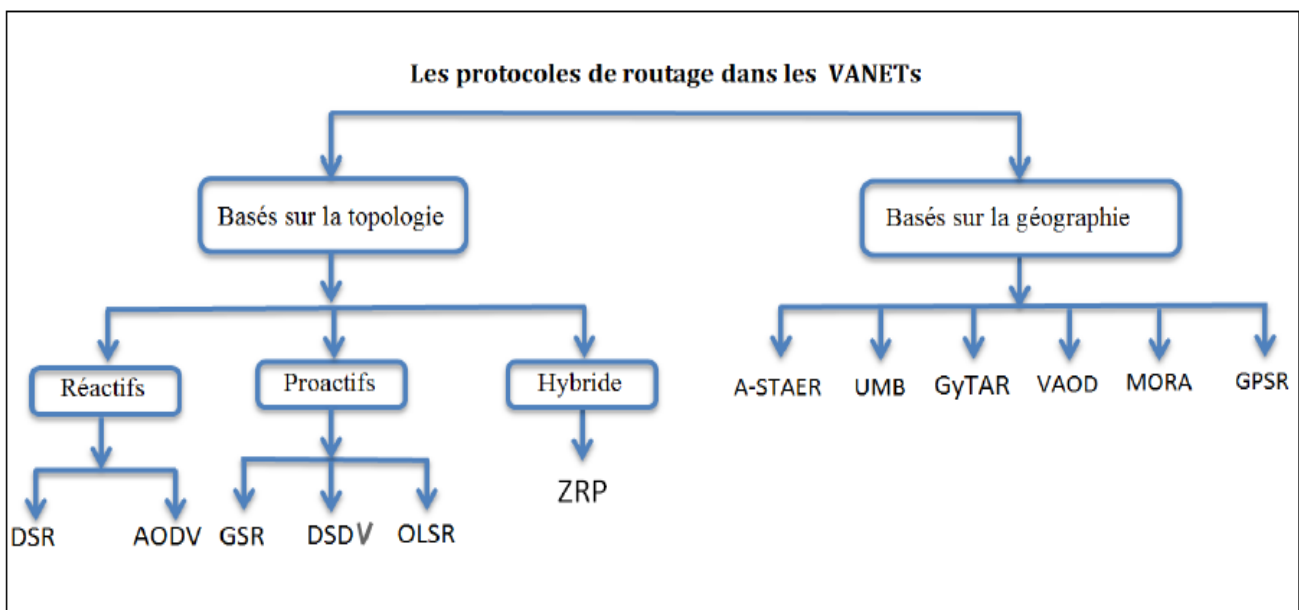


Figure I.7: La classification des protocoles de routage dans les réseaux VANET.

4.4 - Les applications :

On distingue trois types d'application dans les réseaux véhiculaires sans fil : des applications sur la gestion du trafic routier, des applications de confort et ceux du trafic routier :

- **Applications de gestion du trafic routier:**

Les applications de gestion du trafic routier permettent d'éclaircir au conducteur l'état de la route. Grâce aux messages échangés entre les véhicules, ces derniers deviennent des capteurs de trafic. En fournissant des informations sur l'état de la route, les véhicules collaborent afin de céder le passage à l'ambulance, d'éviter les congestions et les embouteillages (**Figure I.8**), et de proposer d'autres itinéraires aux véhicules qui se dirigent au même endroit [5].

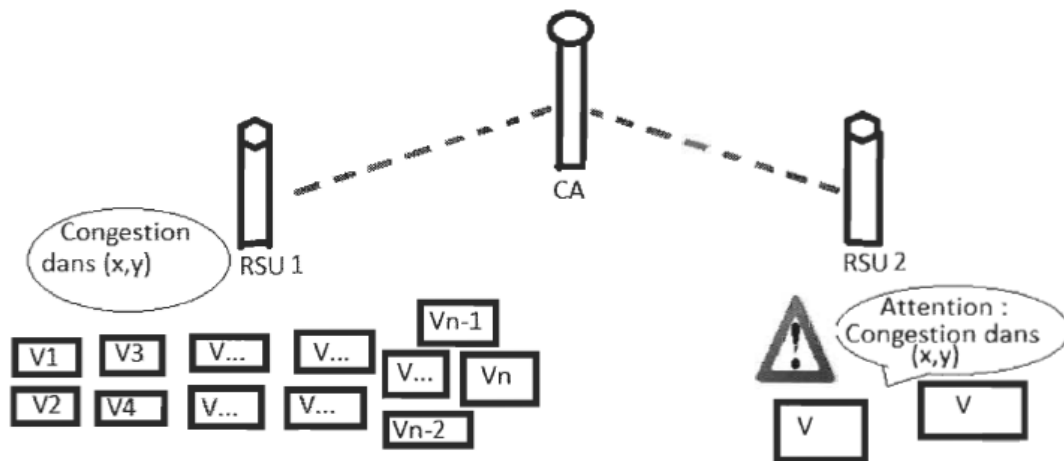


Figure I.8: Prévention d'une congestion [5].

- **Applications de confort:**

Les applications de confort sont des applications qui offrent le confort au conducteur tout au long de la route, surtout pendant les longs trajets (**Figure I.9**). Grâce à ces applications et à l'accès à internet, le conducteur et les passagers ont la possibilité d'écouter de la musique, de voir des vidéos, de jouer à des jeux en ligne, de localiser les restaurants et les stations à proximité, d'avoir des informations touristiques, ainsi que de vérifier à distance les papiers du conducteur et d'effectuer le paiement à distance sur l'autoroute [5].

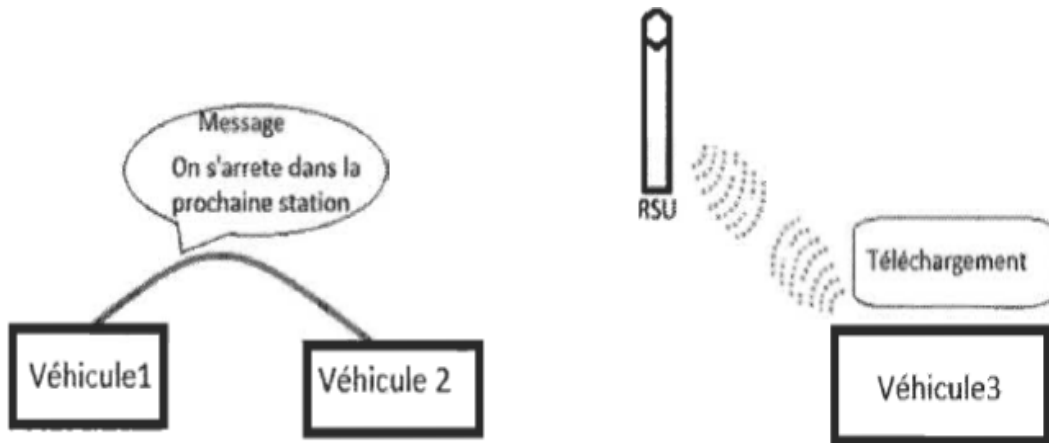


Figure I.9: Exemple d'application de confort [5].

- **Application de sécurité du trafic routier:**

Les applications de sécurité qui visent à améliorer la sécurité des passagers sur les routes en avisant les véhicules de toute situation dangereuse. Ces applications se basent en général sur une diffusion, périodique ou non, de messages informatifs permettant aux conducteurs d'avoir une connaissance de l'état de la route et des véhicules voisins [5].

5 - Le Software Defined Networking :

5.1 - Définition de la SDN :

SDN signifie littéralement Software Defined Networking, c'est-à-dire le réseau défini par logiciel. On comprend donc immédiatement que le sujet est vaste et qu'il va être difficile d'avoir une définition unique.

La définition académique consistait à voir le SDN comme une architecture qui découplait les fonctions de contrôle et de transfert des données du réseau afin d'avoir une infrastructure physique complètement exempte de tout service réseau.

Dans ce modèle, les équipements réseau se contentent d'implémenter des règles, injectées par les applications, de traitement des flux de données. Plus besoin d'avoir sur ces équipements de protocoles de routage : une entité intelligente, appelé « contrôleur » voit le réseau dans sa globalité et injecte directement les règles de traitement des données sur chaque équipement.

Le SDN est donc reconnue aujourd'hui comme une architecture permettant d'ouvrir le réseau aux applications. Cela intègre les deux volets suivants :

- Permettre aux applications de programmer le réseau afin d'en accélérer le déploiement ;
- Permettre au réseau de mieux identifier les applications transportées pour mieux les gérer (qualité de service, sécurité, ingénierie de trafic...) [8] [9].

5.2 - Architecture d'un réseau SDN:

Un réseau traditionnel est composé généralement des équipements d'interconnexions tels que des switches et des routeurs. Ces équipements incorporent à la fois la partie transmission et la partie de contrôle de réseau. Dans ce modèle d'architecture, il est difficile de développer de nouveaux services, en raison du fort couplage qui existe entre le plan de contrôle et le plan de transmission.

Afin d'ouvrir les équipements réseau aux innovations, l'architecture SDN, a vu le jour. Elle permet de découpler la partie de contrôle de la partie transmission des équipements d'interconnexions. Le SDN est composée principalement de trois couches et d'interfaces de communication (**Figure I.10**), nous décrivons dans ce qui suit ces couches, ainsi que les interfaces de communications :

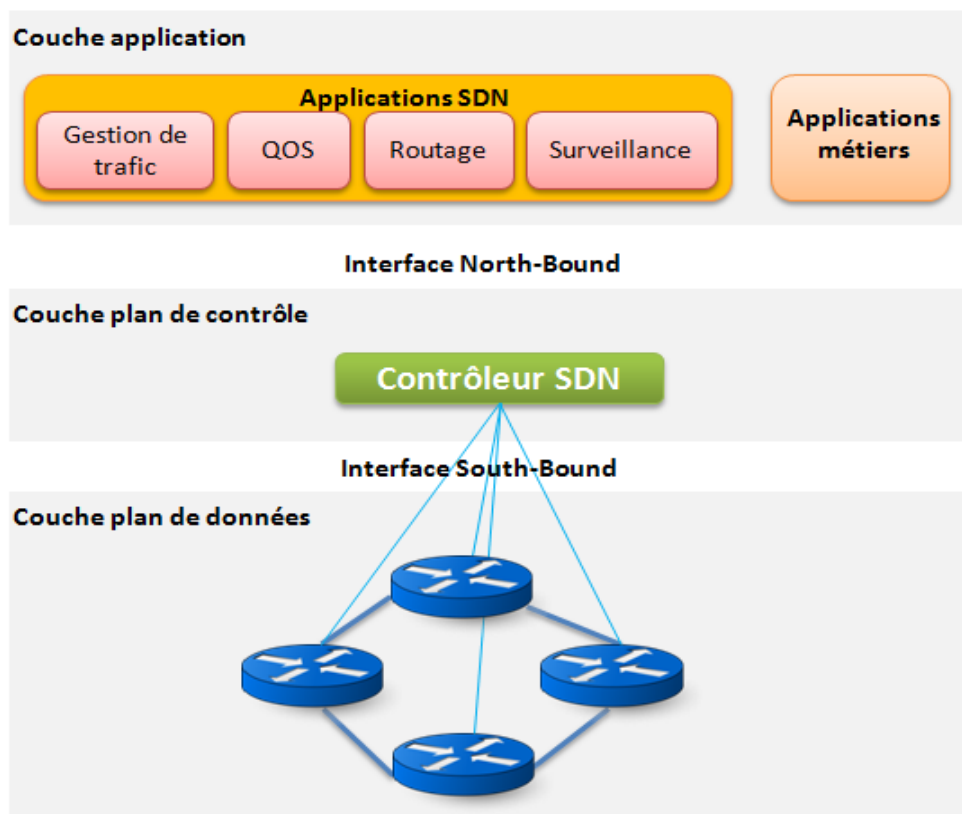


Figure I.10: l'architecture typique d'un réseau SDN [14].

5.2.1 - Les couches :

- **La couche plan de données:**

Elle est composée des équipements d'acheminement tels que les switches ou les routeurs, son rôle principal est de transmettre les données, et collecter les statistiques [10].

- **La couche de contrôle :**

Elle est constituée principalement d'un ou plusieurs contrôleurs SDN, son rôle est de contrôler et de gérer les équipements de l'infrastructure à travers une interface appelée "south-bound API" [10].

- **La couche application :**

Elle représente les applications qui permettent de déployer de nouvelles fonctionnalités réseau, comme l'ingénierie de trafic, QoS, la sécurité, etc. Ces applications sont construites moyennant une interface de programmation appelée "north-bound API" [10].

5.2.2 - Les interfaces de communication:

Il existe principalement trois types d'interfaces permettent aux contrôleurs de communiquer avec leur environnement : interface Sud, Nord et Est/Ouest.

- **Interfaces Sud :**

Les interfaces Sud ou (Southbound APIs) représentent les interfaces de communication, qui permettent au contrôleur SDN d'interagir avec les équipements de la couche d'infrastructure, telle que les switches, et les routeurs.

Le protocole le plus utilisé, et le plus déployé comme interface Sud est le protocole OpenFlow, qui est largement accepté et répandu dans les réseaux SDN [10].

- **Interface Nord :**

Les interfaces Nord (North-bound) servent à programmer les éléments de la transmission en exploitant l'abstraction du réseau fourni par le plan de contrôle. En d'autres termes elles permettent la communication entre le contrôleur la couche applicative [11].

- **Interface Est/Ouest :**

Ce sont des interfaces inter-contrôleurs, on les trouve dans les architectures distribuées (Multi-contrôleurs). Ils permettent la communication entre contrôleurs pour synchroniser les états du réseau. Aucun standard n'est encore disponible pour ce type d'interfaces [11].

5.3 - Les composants du SDN :

5.3.1 - Le protocole Openflow :

Openflow est le protocole utilisé pour la communication entre la couche transmission et la couche de contrôle . Il permet la réalisation des architectures Software Defined Networking. Il permet d'accéder directement au plan de données des périphériques réseau (routeurs, switch), il s'agit d'une façon dynamique, centralisée et programmable d'interagir avec les différents équipements de l'infrastructure (Coker & Azodolmolky, 2017 ; Pujolle).



Figure I.11 : Le protocole Openflow.

5.3.2 - Les contrôleurs SDN :

Toute l'intelligence du réseau est externalisée dans un point logiquement centralisé appelé contrôleur SDN. Ce dernier offre une connaissance globale de l'infrastructure physique et des abstractions pour la configurer. Le contrôleur SDN est un composant programmable. Il peut être vu comme un système d'exploitation qui expose une application programming interface (API) « Northbound API » pour spécifier des applications de contrôle pour les réseaux programmables. Le contrôleur communique avec les équipements via une ou plusieurs API dites «Southbound» ou sud [8] [12].

5.3.3 - Quelques contrôleurs SDN :

Il existe plusieurs contrôleurs SDN, tel que :

- **NOX :**

NOX est le premier contrôleur SDN .Il a été initialement développé chez Nicira Networks et a été le premier à soutenir le protocole OpenFlow. C'est Open-source et écrit en C ++. Il est actuellement à la baisse: il n'y a pas eu de changement majeur depuis mi 2012 [12].

- **POX :**

POX est la version la plus récente. C'est un contrôleur open-source écrit en Python et comme NOX, fournit un cadre pour le développement et le test d'un contrôleur OpenFlow, mais les performances POX sont nettement inférieures à celles des autres contrôleurs et ne convient donc pas au déploiement d'entreprise [12].

- **Ryu :**

Le contrôleur Ryu est un contrôleur de réseau défini par logiciel (SDN) ouvert conçu pour augmenter l'agilité du réseau en le rendant facile à gérer et à adapter la façon dont le trafic est traité. En général, le contrôleur SDN est le cerveau de l'environnement SDN, communiquant des informations jusqu'aux commutateurs et routeurs avec les API sud, et jusqu'aux applications et à la logique métier avec les API nord [23].

- **Floodlight :**

Floodlight est un contrôleur open-source OpenFlow basé sur Java, pris en charge par BigSwitch Networks. Il est sous licence Apache. Il est facile à configurer et à montrer aussi de grandes performances. Avec toutes ses fonctionnalités, Floodlight est plus une solution complète [8].

- **OpenDaylight :**

OpenDaylight est un projet de la Fondation Linux pris en charge par l'industrie. C'est Un framework open source pour faciliter l'accès au logiciel de définition de réseau (SDN). Comme Floodlight, il peut également être considéré comme une solution complète [8].

5.4 - Comparaison entre les réseaux traditionnels et SDN :

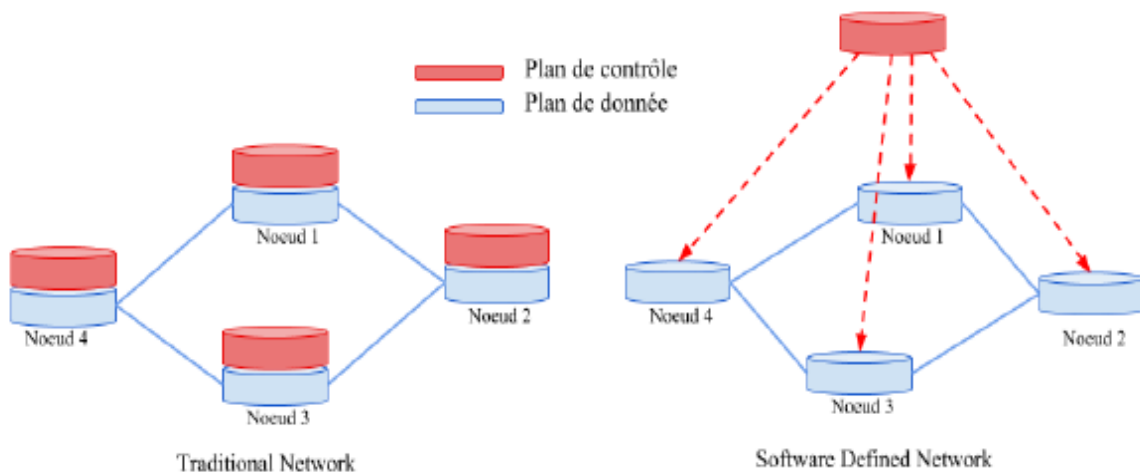


Figure I.12: Comparaison entre les réseaux traditionnels et SDN [13].

	Réseau traditionnel	Réseau SDN
Fonctionnalité	-Le contrôle du réseau est complexe.	-Découple le plan de contrôle de celui du plan de données. -Offre un meilleur contrôle du réseau et la possibilité de le programmer.
Configuration	-Une configuration manuelle et la possibilité de faire des erreurs qui vont entraîner un comportement erroné du réseau.	-Configuration automatique à travers une centralisation du contrôle du réseau. -Optimisation de la configuration.
Performances	-Le problème de configuration statique	-Contrôle global de l'information.
Innovation	-Difficultés d'implémentation de logiciels et des mises à jour dans le réseau. -Environnements de tests limités.	-Implémentation facile de logiciels et des mises à jour dans le réseau. - Environnements de tests suffisants.

Tableau 1.1 : Tableau comparatif entre les réseaux traditionnels et SDN [12].

5.5 - Les avantages du SDN :

- **Réseaux programmables :**

Avec SDN, il est plus facile de modifier les stratégies réseau, car il suffit de changer une politique de haut niveau et non de multiples règles dans divers équipements de réseau. De plus, la facilité de la conception et du contrôle du réseau, et surtout l'existence d'une structure de contrôle centralisée du trafic dans le réseau avec des connaissances globales et une puissance de calcul élevée, simplifient le développement de fonctions plus sophistiquées (centralisation et programmabilité) [12].

- **Flexibilité :**

SDN apporte également une grande flexibilité dans la gestion du réseau. Il devient facile de rediriger le trafic, d'inspecter des flux particuliers, de tester de nouvelles stratégies ou de découvrir des flux inattendus [12].

- **Politique unifiée :**

Avec son contrôleur, SDN garantit également un politique réseau unifié et à jour, puisque le contrôleur est responsable de l'ajout de règles dans les commutateurs, il n'y a aucun risque qu'un administrateur de réseau ait oublié un commutateur ou installé des règles incohérentes entre les dispositifs [8].

- **Optimisation/Évolutivité :**

Technologie à faible coût à comparer avec les équipements réseaux actuels que nous achetons fermés qu'il est impossible de développer ou de fusionner avec d'autres produits ou d'ajouter de nouveaux périphériques (on devient indépendant du fournisseur), ainsi que de réduire le nombre d'ingénieurs, et gagner le temps de travail de plus de 50% c'est-à-dire réduire le temps de travail d'un mois à moins de 15 jours [8].

- **Routage :**

SDN peut également être utilisée pour gérer les informations de routage de manière centralisée en déléguant le routage et en utilisant une interface pour le contrôleur [8].

- **Simplification matérielle :**

SDN a tendance d'utiliser des technologies standard et de base pour contrôler les équipements du réseau, tandis que la puissance de calcul n'est requise qu'au niveau du contrôleur. Ainsi, les équipements de réseau deviendront des produits à bas prix offrant des interfaces standard.

Avec ce type de matériel, il serait également simple d'ajouter de nouveaux périphériques, puisqu'ils ne sont pas spécialisés, de les connecter au réseau et de laisser le contrôleur les gérer conformément à la politique définie. Ainsi, le réseau devient facilement évolutif dès que le contrôleur est évolutif [8].

6 - Conclusion :

Au cours de cette première partie, nous avons fourni une base théorique sur les réseaux VANET, nous l'avons définie puis on a présenté son architecture, ses caractéristiques et ses applications. Ainsi que l'étude bibliographique du paradigme SDN, nous a démontré la nécessité du déploiement de celui-ci, en présentant la définition et l'architecture de ce dernier, puis nous avons traité les composants essentiels de cette solution et ses avantages, afin d'appliquer ses concepts à notre contexte.

Chapitre II :

Architecture du système étudié

1 - Introduction

Dans ce chapitre nous allons définir la mobilité de manière générale qui est un paramètre primordial dans les VANET, ainsi que ses différents modèles de mobilité puis nous allons définir le modèle Random Direction, après on a déterminé les paramètres de travail dans le domaine des réseaux VANET, enfin nous allons évaluer les critères de performances et la qualité de service dans les réseaux SDN.

2 - Le modèle de mobilité :

La mobilité a un rôle essentiel dans la simulation de réseaux de véhicules et pour déterminer si deux nœuds sont proches et peuvent communiquer et une des spécificités des réseaux de véhicules est que le déplacement des véhicules est caractérisé par les infrastructures (routes, feux, tricolores, carrefour...). La prise en compte des dépassements, des bouchons sont très importants pour la représentation de la réalité [21].

2.1 - les différents modèles de mobilité :

Plusieurs modèles sont définis dans la littérature afin de simuler le comportement des nœuds. Ces modèles sont répartis en deux classes, selon le mode de déplacement des nœuds. Dans la première classe (modèles de mobilité par entité), les nœuds se déplacent indépendamment les uns des autres. Tandis que dans la deuxième (modèles de mobilité par groupe), les nœuds se déplacent en groupe.

Nous allons donc étudier les différents modèles de mobilité afin d'avoir une vue sur les modèles existants.

les différents modèles de mobilité	Définition	Exemples
Les modèles individuels	Dans ces modèles, chaque nœud se déplace indépendamment des autres. Ces modèles peuvent être classés selon l'aspect aléatoire de leur mouvement, soit un mouvement absolument aléatoire sans aucune mémoire du passé, soit un mouvement souple où les variations de la vitesse, direction et position à chaque instant sont fonction de l'état précédent [20].	Les modèles sans mémoire : <ul style="list-style-type: none"> • Random Walk(RW) • Random Waypoint(RWP) • Random Direction(RD) • Restricted Random Waypoint
		Les modèles avec mémoire : <ul style="list-style-type: none"> • Boundless • Gauss Markov • Markovian Random Path • City Section (CS) • Le modèle de mobilité avec obstacles
Les modèles de groupe	Dans un modèle de groupe, les nœuds se déplacent ensemble, on remarque bien dans les modèles par entité déjà cités, qu'un nœud se déplace indépendamment des autres nœuds. Sa vitesse, position et direction de mouvement ne sont pas affectées par d'autres nœuds [20].	<ul style="list-style-type: none"> • Le modèle exponentiel aléatoire corrélé • Modèle de mobilité de colonne • Le modèle de mobilité de communauté nomade (NCMM) • Le modèle de mobilité de poursuite • Le modèle de mobilité d'un groupe avec point de référence(RPGM)

Tableau II.2 : Les différents modèles de mobilité.

- **Random Direction (RD) :**

Le Random Direction a été créé pour éviter l'effet de concentration des nœuds au centre, dont chaque nœud choisit aléatoirement une direction qui est un angle entre 0 et 2π et une vitesse entre V_{min} et V_{max} . Dans ce modèle le nœud ne voyage pas pendant un certain temps ou d'une certaine distance, mais se déplace suivant la direction choisie jusqu'à atteindre le bord de la surface de simulation où il prend un temps de repos. Une fois le temps de pause terminé, le nœud choisit de nouveau et aléatoirement une nouvelle direction et une nouvelle vitesse et répète le même processus. La **Figure II.1** montre un nœud utilisant le Random Direction comme modèle de mobilité. La position initiale du nœud est au centre de la surface de simulation. Le nœud commence à se déplacer et chaque fois il se déplace jusqu'au bord où il prend un temps de repos avant de changer sa direction et sa vitesse [20].

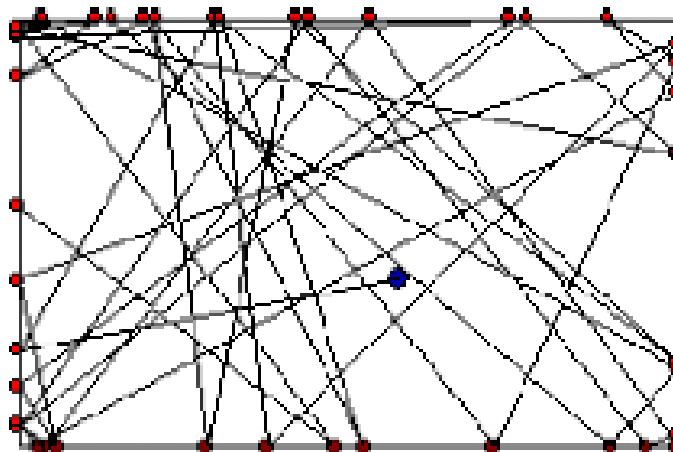


Figure II.1 : Random Direction [20].

3 - Les paramètres de travail dans le domaine des réseaux VANET :

Les réseaux VANET s'étendent sur plusieurs domaines de recherches et développements, nous allons citer les principaux critères tenus en compte pour caractériser ces réseaux dans chaque paramètre:

3.1 -Localisation de véhicules :

Tous les véhicules possédant le système GPS (par satellite) sont informés de la localisation d'un véhicule quelconque (cas d'un accident par exemple), les autres véhicules qui ne sont pas dotés de ce système GPS ne pourront pas repérer ce véhicule c'est pourquoi un autre mécanisme de localisation doit être mis en œuvre.

De plus, le système GPS peut ne pas être performant dans le cas d'insuffisance de satellites (à cause du blocage par des bâtiments par exemple), la position donnée n'est pas toujours précise. Le problème de localisation peut être amélioré si les nœuds peuvent collaborer avec les supports de localisation de l'infrastructure routière [19].

3.2 -Problèmes de congestion :

L'augmentation des communications établies dans une même zone de couverture entre les véhicules peut engendrer une dégradation dans la qualité de service (QOS) et cela est en relation proportionnelle avec le nombre de véhicules présents dans le réseau. La congestion fait l'objet de plusieurs recherches et études approfondies pour améliorer l'état des communications dans le réseau [19].

3.3 -Dynamique du trafic véhiculaire dans les VANET :

Les VANET sont des réseaux extrêmement dynamiques, la mobilité des nœuds perturbe la structure hiérarchique et provoque de fréquentes réorganisations des groupes. Cette instabilité se traduit par un plus grand nombre de messages échangés et de mauvaise performance. La notion de stabilité est primordiale pour la dynamique de ces réseaux, son amélioration se base sur des dépendances spatiales (position du véhicule et sa distance vis-à-vis d'un autre point, sa direction de déplacement ou la qualité de son lien radio avec un autre véhicule) et des dépendances temporelles (vitesse de déplacement du véhicule et surtout la vitesse relative entre deux véhicules) [19].

3.4 -Le routage dans les VANET :

Pour l'établissement de communications entre véhicules un protocole de routage doit être défini, dans le cas où les terminaux ne sont pas à une portée de transmission radio directe le routage est exigé pour acheminer l'information à destination. Les protocoles de routage dans les VANET sont multiples, la plupart des protocoles proposés ont en commun l'utilisation de l'information géographique où les informations indiquent des distances géographiques entre les nœuds. L'adaptation des protocoles de routage topologiques sur les VANET (AODV, DSR, OLSR, DSDV) était un point de départ pour définir un protocole de routage tout en ajoutant des extensions à ces protocoles [19].

3.5 -La sécurité dans les VANET :

La sécurisation des réseaux VANET est un facteur indispensable pour assurer l'intégrité des informations échangées lors des communications entre véhicules ou bien entre véhicules et infrastructures. Les problèmes liés aux intrusions de véhicules malicieux ont des conséquences graves sur l'ensemble des véhicules interconnectés. Cependant, des mécanismes de sécurité (la cryptographie, le certificat numérique, système de détection d'intrusions... etc.) sont toujours en voie d'amélioration pour minimiser les pertes d'informations et garantir l'intégralité des données de deux véhicules) [19].

3.6 -La qualité de service (QoS) :

Dans un réseau véhiculaire, le bon acheminement des données est un rôle déterministe pour définir la qualité de service du réseau. Plusieurs protocoles ont été établis afin d'améliorer cette qualité, ce paramètre sera étudié par la suite [19].

4 - Les critères de performances :

4.1 - Taux de livraison de paquets :

C'est un facteur très important pour évaluer les performances d'un protocole de routage dans n'importe quel type de réseau. Ces performances dépendent des différents paramètres choisis pour la simulation. Les facteurs les plus importants sont la taille du paquet, le nombre de nœuds, la portée de communication et la structure du réseau. On peut obtenir le taux de livraison de paquet 'PDR' (Packet Delivery Ratio) à partir de la somme de nombre de paquets reçus par le destinataire, ce dernier divisé par la somme de paquets émis par tous les nœuds émetteurs [22].

$$\text{PDR} = \frac{\sum \text{nbr de paquets recus par la destination}}{\sum \text{nbr de paquets envoyés par tous les noeuds source}} \quad (\text{II.1})$$

4.2 - Délai de bout en bout :

Représente l'intervalle de temps qui s'écoule entre l'instant d'envoi du paquet, par la source, et l'instante de réception de ce paquet par la destination [21].

$$\text{D} = \text{Temps de réception} - \text{le temps de transmission} \quad (\text{II.2})$$

4.3 - Paquets perdus :

Ce sont les paquets qui n'ont pas pu atteindre leur destination. Cela est traduit mathématiquement par l'équation suivante[22]:

$$D = \frac{\textit{nbr paquets envoyés} - \textit{nbr paquets recus}}{\textit{nbr paquet env oyés}} \quad (\text{II.3})$$

4.4 - Débit moyen :

Parfois appelée bande passante, elle détermine la quantité maximale d'informations (bits) par unité de temps (b / s) [3].

$$\textit{debit}_{\textit{moy en}} = \frac{\textit{taille du paquet reçu}}{(\textit{temps reception} - \textit{temps emission})} \quad (\text{II.4})$$

5 - Conclusion

Dans ce chapitre, nous avons décrit la mobilité et ses différents types ainsi que le modèle Random Direction, après nous avons déterminé les paramètres de travail dans le domaine des réseaux VANET, ensuite on a évalué les critères de performances de réseau véhiculaire sans fil et enfin la qualité de service dans les réseaux SDN.

Chapitre III:

Résultat et discussions

1 - Introduction :

Dans ce chapitre nous allons définir le simulateur utilisé « Mininet-Wifi » ensuite nous allons présenter les étapes qu'on a fait pour finaliser notre travail commençant par aborder les points essentiels de l'implémentation du SDN dans un réseau VANET et de son étude d'une façon plus détailler et nous allons donner les résultats obtenus .

2 - Composants et outils logiciels :

2.1 - Mininet :

L'avancement de la technologie SDN dépend fortement des transformations réussies des idées de recherche internes en produit réel. Pour permettre de telles transformations, un banc de test offrant un environnement de mise en réseau évolutif et haute-fidélité pour tester et évaluer des conceptions nouvelles, existantes et extrêmement précieuses. Mininet est un logiciel open source disponible gratuitement qui émule les périphériques OpenFlow et les contrôleurs SDN, il est de loin l'émulateur SDN le plus populaire, il est conçu pour atteindre à la fois précision et évolutivité en exécutant du code non modifié d'application réseaux dans des conteneurs Linux légers.

Cet émulateur réseau permet de créer un réseau d'hôtes virtuels, de commutateurs, de contrôleurs et de liens. Les hôtes Mininet exécutent un logiciel de réseau Linux standard et ses commutateurs prennent en charge OpenFlow pour un routage personnalisé très flexible et une mise en réseau définie par logiciel. Il dispose d'outils de ligne de commande et d'API très simples. Mininet permet à l'utilisateur de créer, personnaliser, partager et tester facilement des réseaux SDN.

2.2 - Mininet-wifi :

Mininet-WiFi est un fork de l'émulateur de réseau Mininet SDN . Les développeurs de Mininet-wifi ont étendu les fonctionnalités de Mininet en ajoutant des stations wifi virtualisées et des points d'accès basés sur les pilotes sans fil Linux standard. Ils ont également ajouté des classes pour prendre en charge l'ajout de ces dispositifs sans fil dans un scénario de réseau Mininet et pour émuler les attributs d'une station mobile tels que la position et le mouvement par rapport aux points d'accès.

Le Mininet-WiFi a étendu le code Mininet de base en ajoutant ou en modifiant des classes et des scripts. Ainsi, Mininet-wifi ajoute de nouvelles fonctionnalités et prend toujours en charge toutes les capacités d'émulation SDN normale de l'émulateur de réseau Mininet standard.

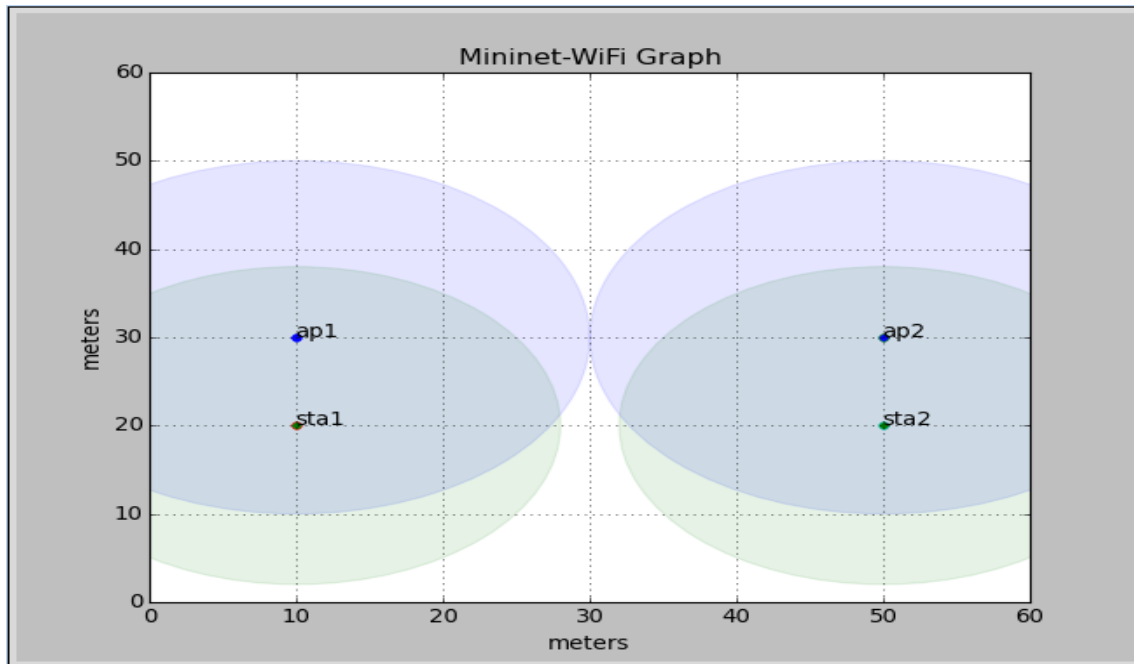


Figure III.1: Exemple de mininet-wifi graph.

3 - Les étapes de simulation :

3.1 - La création de la topologie :

La topologie utilisée dans notre projet se compose de un point d'accès, un contrôleur et 5 stations (cars). Cette topologie a été créée grâce au simulateur Mininet-wifi. Après avoir lancé ce dernier, nous avons créé une topologie sous forme de fichier python. Il comporte les parties suivantes:

1. Importation de quelques bibliothèques de mininet-wifi :

```
import sys
from mininet.node import RemoteController
from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
```

Figure III.2: les bibliothèques de mininet-wifi.

2. Création de contrôleurs : notre contrôleur est relié avec cette topologie à l'adresse IP 127.0.0.1 et le port =6633 avec le protocole tcp comme se présente dans la figure :

```
info("*** Creating controller\n")
c1=net.addController('c1',controller=RemoteController,ip='127.0.0.1',
                    protocol='tcp',port=6633)
```

Figure III.3: La création du contrôleur.

3. Création des nœuds : dans notre topologie on a 5 véhicules, au qu'elle nous avons attribué les adresses IP de 10.0.0.2 à 10.0.0.6 et les coordonnées de mobilité avec la vitesse de chaque véhicule.

```
info("*** Creating nodes\n")
sta1=net.addStation('car1', mac='00:00:00:00:00:02', ip='10.0.0.2/8',
                    min_x=1, max_x=40, min_y=10, max_y=50, min_v=5, max_v=10)
sta2=net.addStation('car2', mac='00:00:00:00:00:03', ip='10.0.0.3/8',
                    min_x=45, max_x=90, min_y=20, max_y=60, min_v=5, max_v=10)
sta3=net.addStation('car3', mac='00:00:00:00:00:04', ip='10.0.0.4/8',
                    min_x=1, max_x=40, min_y=30, max_y=70, min_v=5, max_v=10)
sta4=net.addStation('car4', mac='00:00:00:00:00:05', ip='10.0.0.5/8',
                    min_x=45, max_x=90, min_y=40, max_y=80, min_v=5, max_v=10)
sta5=net.addStation('car5', mac='00:00:00:00:00:03', ip='10.0.0.6/8',
                    min_x=45, max_x=90, min_y=50, max_y=90, min_v=5, max_v=10)
```

Figure III.4: La création des nœuds.

4. Création de points d'accès : dans notre topologie on a un seul point d'accès (ap1).

```
if '-m' in args:
    ap1 = net.addAccessPoint('ap1', wlans=2, ssid='ssid1,ssid2', mode='g',
                            channel='1', failMode="standalone",
                            position='50,50,0')
else:
    ap1 = net.addAccessPoint('ap1', ssid='new-ssid', mode='g', channel='1',
                            failMode="standalone", position='50,50,0')
```

Figure III.5: La création d'un point d'accès.

- Après avoir créé ces derniers, nous allons maintenant configurer les nœuds et le modèle de mobilité « RandomDirection ».

```
info("*** Configuring wifi nodes\n")
net.configureWifiNodes()

if '-p' not in args:
    net.plotGraph(max_x=300, max_y=300)

net.setMobilityModel(time=0, model='RandomDirection',
                    max_x=100, max_y=100, seed=20)
```

Figure III.6: La configuration des nœuds et du modèle de mobilité.

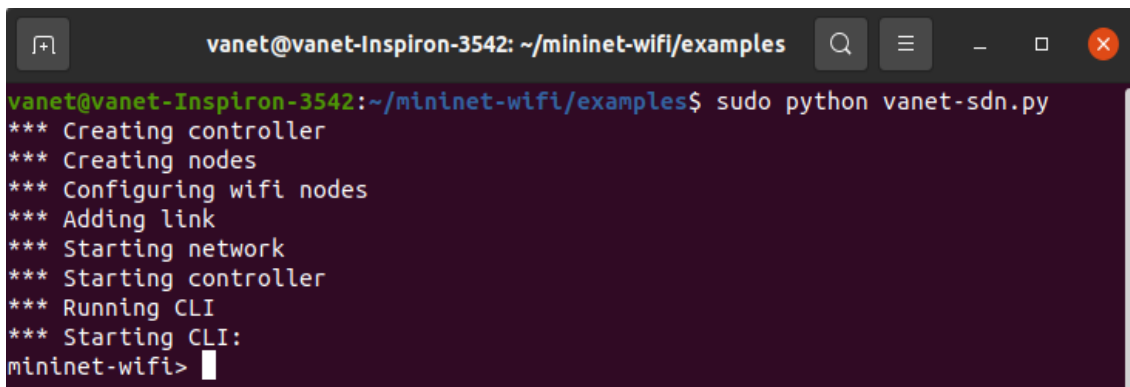
- Pour démarrer le réseau et le contrôleur on a utilisé les commandes ci-dessous :

```
info("*** Starting network\n")
net.build()
ap1.start([])
info("*** Starting controller\n")
for controller in net.controllers:
    controller.start()
```

Figure III.7: les commandes qui permettent le démarrage du réseau et du contrôleur.

3.2 - Le lancement de la topologie :

Après avoir créé la topologie, on ouvre le terminal on entre dans le fichier nommé mininet-wifi puis exemples après on lance notre fichier python « vanet-sdn.py » grâce à la commande présentée dans la figure :



```
vanet@vanet-Inspiron-3542: ~/mininet-wifi/examples
vanet@vanet-Inspiron-3542:~/mininet-wifi/examples$ sudo python vanet-sdn.py
*** Creating controller
*** Creating nodes
*** Configuring wifi nodes
*** Adding link
*** Starting network
*** Starting controller
*** Running CLI
*** Starting CLI:
mininet-wifi> |
```

Figure III.8:Le lancement de la topologie.

Ensuite nous obtenons le mininet-wifi Graph dont les cars sont en mouvement, nous avons pris des figures à deux moments différents comme se présente ci-dessous :

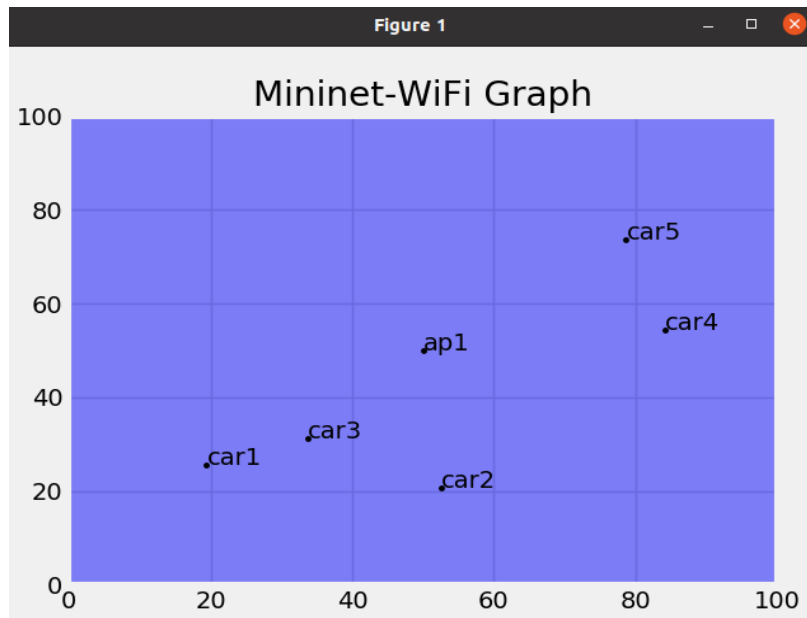


Figure III.9: Visualisation du Graph à l'instant T1.

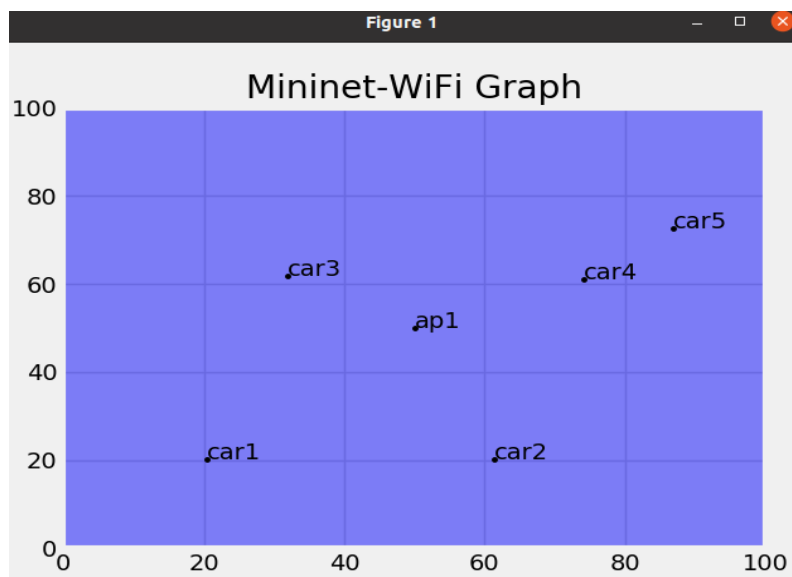


Figure III.10: Visualisation du Graph à l'instant T2.

3.3 - Les commandes principales de topologie Mininet-wifi :

Mininet-wifi fournit une interface en ligne de commande qui peut être utilisée pour voir l'état du réseau et faire des tests dessus.

Mininet-wifi> nodes: on a utilisé cette commande pour afficher les nœuds de notre topologie réseau, ap1 correspond a un point d'accès et c1 correspond a un contrôleur.

```
mininet-wifi> nodes
available nodes are:
ap1 c1 car1 car2 car3 car4 car5
mininet-wifi> █
```

Figure III.11: L'affichage des nœuds.

mininet-wifi>links : cette commande est pour afficher les liens du réseau.

```
mininet-wifi> links
car1-wlan0<->wifi (OK wifi)
car2-wlan0<->wifi (OK wifi)
car3-wlan0<->wifi (OK wifi)
car4-wlan0<->wifi (OK wifi)
car5-wlan0<->wifi (OK wifi)
mininet-wifi> █
```

Figure III.12: L'affichage des liens du réseau.

mininet-wifi>intfs : afin d'afficher les interfaces du réseau.

```
mininet-wifi> intfs
c1:
car1: car1-wlan0
car2: car2-wlan0
car3: car3-wlan0
car4: car4-wlan0
car5: car5-wlan0
ap1: lo,ap1-wlan1
mininet-wifi> █
```

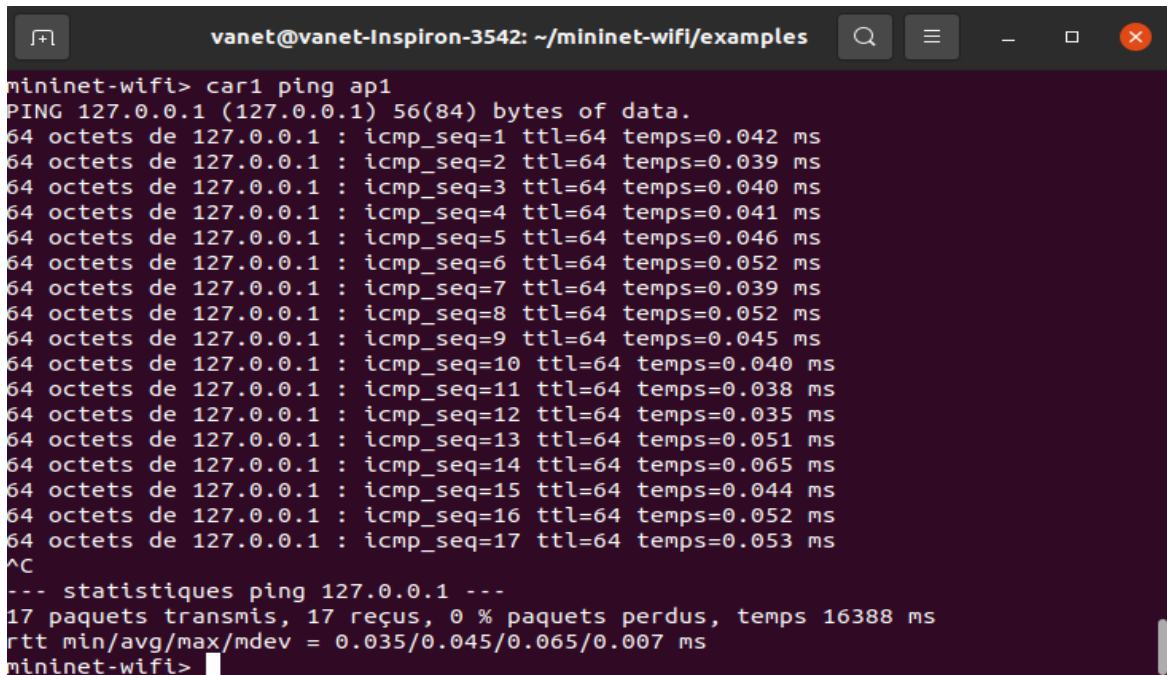
Figure III.13:L'affichage des interfaces.

mininet-wifi>dump : affiche les adresses IP de chaque machine ainsi que le nom de la carte réseau.

```
mininet-wifi> dump
<Controller c1: 127.0.0.1:6653 pid=358074>
<Station car1: car1-wlan0:10.0.0.2 pid=358081>
<Station car2: car2-wlan0:10.0.0.3 pid=358083>
<Station car3: car3-wlan0:10.0.0.4 pid=358085>
<Station car4: car4-wlan0:10.0.0.5 pid=358087>
<Station car5: car5-wlan0:10.0.0.6 pid=358089>
<OVSAP ap1: lo:127.0.0.1,ap1-wlan1:None pid=358095>
mininet-wifi> █
```

Figure III.14: l'affichage les adresses IP.

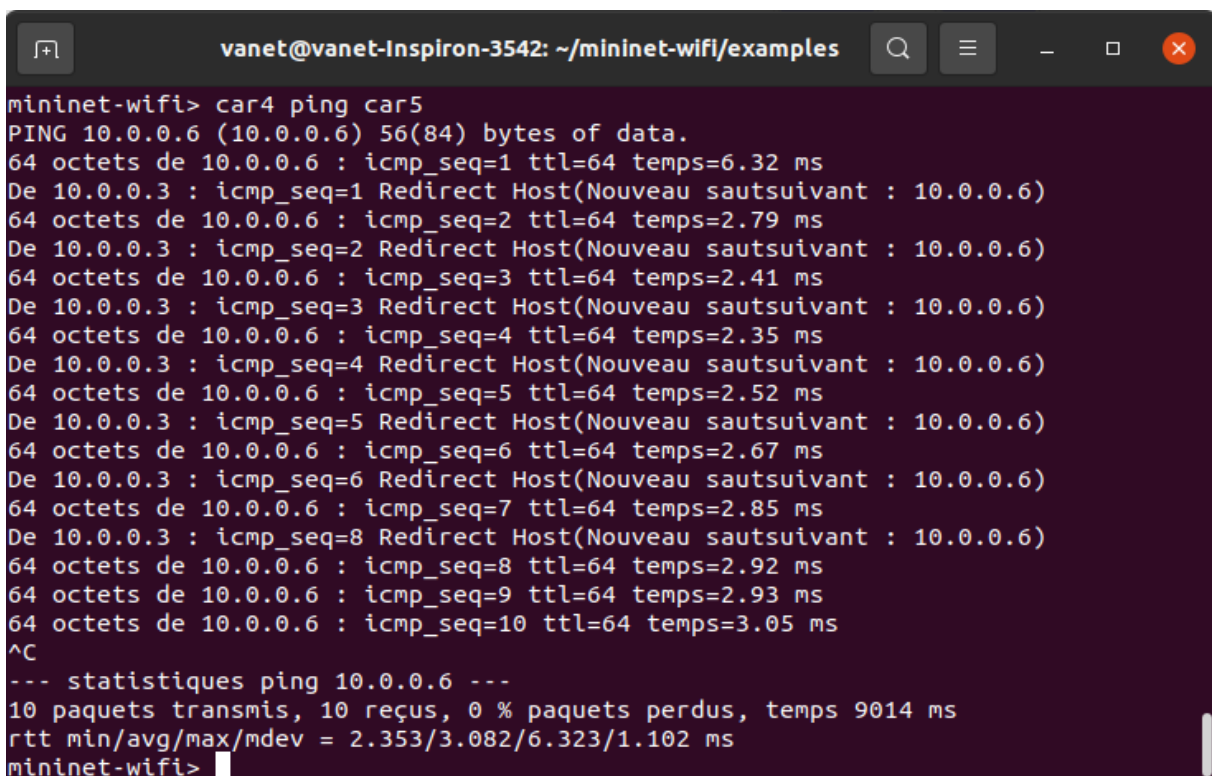
mininet-wifi> car ping ap : Demande à car d'effectuer un ping sur le point d'accès, on a pris l'exemple de car1 ping ap1.



```
vanet@vanet-Inspiron-3542: ~/mininet-wifi/examples
mininet-wifi> car1 ping ap1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 octets de 127.0.0.1 : icmp_seq=1 ttl=64 temps=0.042 ms
64 octets de 127.0.0.1 : icmp_seq=2 ttl=64 temps=0.039 ms
64 octets de 127.0.0.1 : icmp_seq=3 ttl=64 temps=0.040 ms
64 octets de 127.0.0.1 : icmp_seq=4 ttl=64 temps=0.041 ms
64 octets de 127.0.0.1 : icmp_seq=5 ttl=64 temps=0.046 ms
64 octets de 127.0.0.1 : icmp_seq=6 ttl=64 temps=0.052 ms
64 octets de 127.0.0.1 : icmp_seq=7 ttl=64 temps=0.039 ms
64 octets de 127.0.0.1 : icmp_seq=8 ttl=64 temps=0.052 ms
64 octets de 127.0.0.1 : icmp_seq=9 ttl=64 temps=0.045 ms
64 octets de 127.0.0.1 : icmp_seq=10 ttl=64 temps=0.040 ms
64 octets de 127.0.0.1 : icmp_seq=11 ttl=64 temps=0.038 ms
64 octets de 127.0.0.1 : icmp_seq=12 ttl=64 temps=0.035 ms
64 octets de 127.0.0.1 : icmp_seq=13 ttl=64 temps=0.051 ms
64 octets de 127.0.0.1 : icmp_seq=14 ttl=64 temps=0.065 ms
64 octets de 127.0.0.1 : icmp_seq=15 ttl=64 temps=0.044 ms
64 octets de 127.0.0.1 : icmp_seq=16 ttl=64 temps=0.052 ms
64 octets de 127.0.0.1 : icmp_seq=17 ttl=64 temps=0.053 ms
^C
--- statistiques ping 127.0.0.1 ---
17 paquets transmis, 17 reçus, 0 % paquets perdus, temps 16388 ms
rtt min/avg/max/mdev = 0.035/0.045/0.065/0.007 ms
mininet-wifi>
```

Figure III.15: Ping entre le point d'accès et car1.

-mininet-wifi> car x ping car y: Demande à car x d'effectuer un ping sur car y, on a pris l'exemple de car4 ping car5.



```
vanet@vanet-Inspiron-3542: ~/mininet-wifi/examples
mininet-wifi> car4 ping car5
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 octets de 10.0.0.6 : icmp_seq=1 ttl=64 temps=6.32 ms
De 10.0.0.3 : icmp_seq=1 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=2 ttl=64 temps=2.79 ms
De 10.0.0.3 : icmp_seq=2 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=3 ttl=64 temps=2.41 ms
De 10.0.0.3 : icmp_seq=3 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=4 ttl=64 temps=2.35 ms
De 10.0.0.3 : icmp_seq=4 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=5 ttl=64 temps=2.52 ms
De 10.0.0.3 : icmp_seq=5 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=6 ttl=64 temps=2.67 ms
De 10.0.0.3 : icmp_seq=6 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=7 ttl=64 temps=2.85 ms
De 10.0.0.3 : icmp_seq=8 Redirect Host(Nouveau sautsuivant : 10.0.0.6)
64 octets de 10.0.0.6 : icmp_seq=8 ttl=64 temps=2.92 ms
64 octets de 10.0.0.6 : icmp_seq=9 ttl=64 temps=2.93 ms
64 octets de 10.0.0.6 : icmp_seq=10 ttl=64 temps=3.05 ms
^C
--- statistiques ping 10.0.0.6 ---
10 paquets transmis, 10 reçus, 0 % paquets perdus, temps 9014 ms
rtt min/avg/max/mdev = 2.353/3.082/6.323/1.102 ms
mininet-wifi>
```

Figure III.16: Ping entre car4 et car5.

3.4 - SDVANET avec le contrôleur POX :

Pour lancer le contrôleur POX On tape la commande «**pox/pox.py forwarding.l2_learning** » dans le terminal.

```
vanet@vanet-Inspiron-3542:~$ pox/pox.py forwarding.l2_learning
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
INFO:core:POX 0.5.0 (eel) is up.
█
```

Figure III.17: Le lancement du contrôleur POX.

Dans un autre terminal on lance la topologie en la reliant avec le contrôleur POX, en tapant la commande :

« **sudo mn --wifi --custom vanet-sdn.py --controller=remote ,ip=10.0.2.15, port=6633** ».

-Sudo pour exécuter la commande en mode administrateur.

-Custom vanet-sdn.py : notre topologie customisée qui se trouve dans le fichier vanet-sdn.py.

- Le contrôleur à relier avec cette topologie à l'adresse IP '10.0.2.15'.

- Le port au qu'elle le contrôleur se relie a la topologie est le port 6633.

```
vanet@vanet-Inspiron-3542:~/mininet-wifi/examples$ sudo mn --wifi
--custom vanet-sdn.py --controller=remote ,ip=10.0.2.15,port=6633
*** Adding stations:
sta1 sta2
*** Adding access points:
ap1
*** Configuring wifi nodes...
*** Creating network
*** Adding controller
*** Adding hosts:

*** Adding switches:

*** Adding links:
(sta1, ap1) (sta2, ap1)
*** Starting controller(s)
c0
*** Starting L2 nodes
ap1 ...
*** Starting CLI:
mininet-wifi> █
```

Figure III.18: Le lancement de la topologie avec le contrôleur POX.

Une fois la topologie lancée et que le contrôleur marche bien on aura le résultat présenté dans la figure suivante :

```
vanet@vanet-Inspiron-3542:~$ pox/pox.py forwarding.l2_learning
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
INFO:core:POX 0.5.0 (eel) is up.
INFO:openflow.of_01:[00-00-00-00-00-01|4096 2] connected
█
```

Figure III.19: Le contrôleur POX est connecté.

Pour tester l'accessibilité « entre deux stations » et « entre station et un point d'accès », on utilise la commande *ping*, comme se présente dans les figures ci-dessous :

```
mininet-wifi> sta1 ping ap1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 octets de 127.0.0.1 : icmp_seq=1 ttl=64 temps=0.077 ms
64 octets de 127.0.0.1 : icmp_seq=2 ttl=64 temps=0.073 ms
64 octets de 127.0.0.1 : icmp_seq=3 ttl=64 temps=0.075 ms
64 octets de 127.0.0.1 : icmp_seq=4 ttl=64 temps=0.073 ms
^C
--- statistiques ping 127.0.0.1 ---
4 paquets transmis, 4 reçus, 0 % paquets perdus, temps 3056 ms
rtt min/avg/max/mdev = 0.073/0.074/0.077/0.001 ms
mininet-wifi> █
```

Figure III.20: Ping entre une station et point d'accès.

```
mininet-wifi> sta1 ping sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 octets de 10.0.0.2 : icmp_seq=1 ttl=64 temps=190 ms
64 octets de 10.0.0.2 : icmp_seq=2 ttl=64 temps=1.21 ms
64 octets de 10.0.0.2 : icmp_seq=3 ttl=64 temps=0.283 ms
^C
--- statistiques ping 10.0.0.2 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2004 ms
rtt min/avg/max/mdev = 0.283/63.754/189.767/89.105 ms
mininet-wifi> █
```

Figure III.21: Ping entre deux stations.

Lorsqu'on quitte la CLI de mininet-wifi avec la commande « **mininet-wifi>exit** » le contrôleur POX se déconnecte automatiquement.

```

vanet@vanet-Inspiron-3542:~$ pox/pox.py forwarding.l2_learning
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
INFO:core:POX 0.5.0 (eel) is up.
INFO:openflow.of_01:[00-00-00-00-00-01|4096 2] connected
^CINFO:core:Going down...
INFO:openflow.of_01:[00-00-00-00-00-01|4096 2] disconnected
INFO:core:Down.
vanet@vanet-Inspiron-3542:~$ █

```

Figure III.22 : Le contrôleur POX est déconnecté.

```

mininet-wifi> sta1 ping sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- statistiques ping 10.0.0.2 ---
3 paquets transmis, 0 reçus, 100 % paquets perdus, temps 2054 ms
mininet-wifi> █

```

Figure III.23 : Ping entre deux stations après la déconnexion du POX.

3.5 - SDVANET avec le contrôleur Ryu :

Avec cette commande « **ryu-manager ryu.app.simple_switch_13** » on peut lancer le contrôleur Ryu :

```

vanet@vanet-Inspiron-3542:~$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
█

```

Figure III.24 : Le lancement du contrôleur Ryu.

Dans un autre terminal on lance la topologie en la reliant avec le contrôleur Ryu, en tapant la commande :

« **sudo mn --wifi --custom vanet-sdn.py --controller=remote,ip=10.0.2.15,port=6633** ».

```

vanet@vanet-Inspiron-3542:~/mininet-wifi/examples$ sudo mn --wifi
--custom vanet-sdn.py --controller=remote,ip=10.0.2.15,port=6633
*** Adding stations:
sta1 sta2
*** Adding access points:
ap1
*** Configuring wifi nodes...
*** Creating network
*** Adding controller
*** Adding hosts:

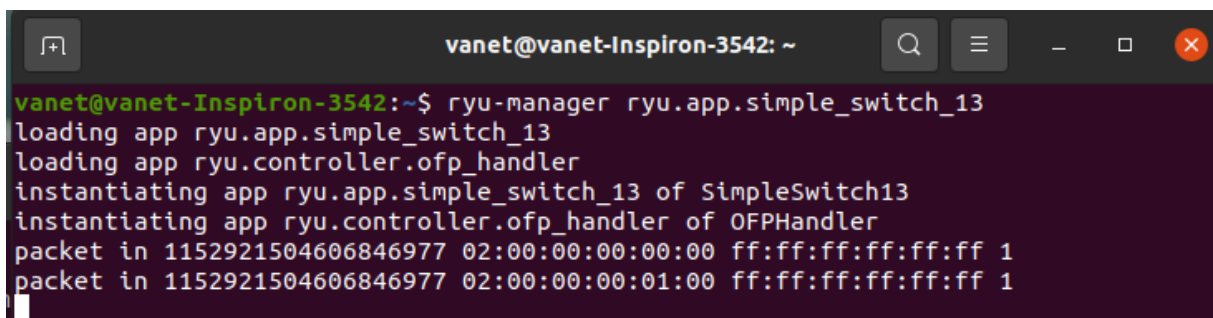
*** Adding switches:

*** Adding links:
(sta1, ap1) (sta2, ap1)
*** Starting controller(s)
c0
*** Starting L2 nodes
ap1 ...
*** Starting CLI:
mininet-wifi>

```

Figure III.25: Le lancement de la topologie avec le contrôleur Ryu.

Une fois la topologie lancer et que le contrôleur marche bien on aura le résultat présenter dans la figure suivante :



```

vanet@vanet-Inspiron-3542:~$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 1152921504606846977 02:00:00:00:00:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 ff:ff:ff:ff:ff:ff 1

```

Figure III.26: Le résultat dans le 1^{er} terminal après le lancement de la topologie.

Pour tester l'accessibilité dans le réseau on utilise la commande *ping* comme on a fait déjà avec le contrôleur POX :

```

mininet-wifi> sta1 ping sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 octets de 10.0.0.2 : icmp_seq=1 ttl=64 temps=1.17 ms
64 octets de 10.0.0.2 : icmp_seq=2 ttl=64 temps=0.209 ms
64 octets de 10.0.0.2 : icmp_seq=3 ttl=64 temps=1.28 ms
64 octets de 10.0.0.2 : icmp_seq=4 ttl=64 temps=0.406 ms
64 octets de 10.0.0.2 : icmp_seq=5 ttl=64 temps=0.761 ms
64 octets de 10.0.0.2 : icmp_seq=6 ttl=64 temps=0.365 ms
^C
--- statistiques ping 10.0.0.2 ---
6 paquets transmis, 6 reçus, 0 % paquets perdus, temps 5047 ms
rtt min/avg/max/mdev = 0.209/0.697/1.279/0.407 ms
mininet-wifi>

```

Figure III.27 : Ping entre deux stations avec le contrôleur Ryu.

Dans le 1^{er} terminal où on a lancé le contrôleur Ryu on va voir que les paquets sont transmis comme se présente dans la figure ci-dessous :

```

vanet@vanet-Inspiron-3542:~$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 1152921504606846977 02:00:00:00:00:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:ff:00:01:00 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:ff:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:00:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1

```

Figure III.28 : Les paquets sont transmis.

Pour déconnecter le contrôleur Ryu on utilise la commande « **sudo fuser -k 6653/tcp** » dans le 2^{eme} terminal après on va voir que le processus est arrêté.

```
vanet@vanet-Inspiron-3542:~/mininet-wifi/examples$ sudo fuser -k 6653/tcp
6653/tcp:          3302
```

Figure III.29 : la commande pour quitter le processus.

```
vanet@vanet-Inspiron-3542:~/ryu$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 1152921504606846977 02:00:00:00:00:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 ff:ff:ff:ff:ff:ff 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:ff:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:ff:00:01:00 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:16 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:01:00 33:33:00:00:00:02 1
packet in 1152921504606846977 02:00:00:00:00:00 ff:ff:ff:ff:ff:ff 1
Processus arrêté
vanet@vanet-Inspiron-3542:~/ryu$
```

Figure III.30 : Le contrôleur Ryu est déconnecté.

Après on va tester le *ping* lorsque le contrôleur est déconnecté on aura ce résultat.

```
mininet-wifi> sta1 ping sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
De 10.0.0.1 icmp_seq=1 Hôte de destination injoignable
De 10.0.0.1 icmp_seq=2 Hôte de destination injoignable
De 10.0.0.1 icmp_seq=3 Hôte de destination injoignable
De 10.0.0.1 icmp_seq=4 Hôte de destination injoignable
De 10.0.0.1 icmp_seq=5 Hôte de destination injoignable
De 10.0.0.1 icmp_seq=6 Hôte de destination injoignable
^C
--- statistiques ping 10.0.0.2 ---
8 paquets transmis, 0 reçus, +6 erreurs, 100 % paquets perdus, temps
 7164 ms
 tuyau 4
mininet-wifi>
```

Figure III.31 : Ping entre deux stations après la déconnexion de contrôleur Ryu.

4 - Discussion de simulation :

4.1 - L'accessibilité du réseau :

Lorsqu'on a testé l'accessibilité du réseau avec la commande « *ping* » entre deux stations et entre une station et un point d'accès sachant que le contrôleur est relié à la topologie est connecté, les paquets transmis sont bien reçus comme on a vu dans le cas du contrôleur POX (figure III.20 figure III.21) et pour le cas du contrôleur Ryu figure III.27.

Si le contrôleur relié à la topologie est déconnectée, les paquets transmises sont perdus comme on a vu dans le cas du contrôleur POX (figure III.23) et le cas du contrôleur Ryu (figure III.30).

Donc, la perte de paquets se produit lorsqu'un ou plusieurs paquets de données circulant sur un réseau informatique ne parviennent pas à atteindre leurs destinations. La perte de paquets est causée par l'absence ou le mal fonctionnement du contrôleur SDN. La perte de paquets est mesurée comme un pourcentage de paquets perdus par rapport aux paquets envoyés.

Les contrôleurs disposent donc d'une maîtrise globale de réseau et sont capables de s'informer en temps réel sur l'état et l'activité des équipements qu'ils contrôlent.

4.2 - La comparaison entre le contrôleur POX et le contrôleur Ryu :

Contrôleur	Organisation	Langage	Open source	gui	Plateforme	Fonctionnalités
POX	Nicira	Python	Yes	Yes	Linux, Mac, Windows	Améliorer les performances du 1 ^{er} contrôleur NOX
RYU	NTT, OSRG group	Python	Yes	Yes	Linux	Supporte l'Open Stack

Tableau III3 : Comparaison entre le contrôleur POX et le contrôleur Ryu.

Le contrôleur POX peut principalement fonctionner sur Python v2.7 et peut officiellement prendre en charge Windows, Mac OS et Linux. De plus, le contrôleur POX est capable de convertir des périphériques OpenFlow inhabituels pour fonctionner comme des commutateurs, des routeurs, des équilibreur de charge, des périphériques de pare-feu, etc., et offre de meilleures performances à l'architecture SDN.

Le contrôleur Ryu peut être utilisé avec OpenFlow pour collecter des informations statistiques à partir des commutateurs. Il peut donc être configuré en tant que moniteur de trafic, pare-feu, routeur et commutateur. De plus, Ryu a un meilleur débit TCP que POX.

5 - Conclusion

Dans ce chapitre nous avons exploité l'émulateur Mininet-Wifi et le contrôleur POX ,qui est défini comme un contrôleur OpenFlow. Notre contribution principale est en place le modèle SDN dans un réseau VANET, notre émulateur nous a aidés à créer, gérer, visualiser notre réseau et tester son accessibilité.

Conclusion générale :

Les réseaux ad hoc de véhicules (VANETs) ont émergé comme une nouvelle technologie pour améliorer la sécurité sur les routes, les conditions de trafic et le confort de voyage. De nos jours, les architectures VANETs souffrent de problèmes d'évolutivité vu qu'il est très difficile de déployer des services à grande échelle. Ces architectures sont rigides, difficiles à gérer et souffrent d'un manque de flexibilité et d'adaptabilité au contrôle.

Le SDN – Software Defined Networking – est certainement le sujet chaud qui agite le monde du réseau depuis ces dernières années, la technologie SDN a déclenché un changement radical à long terme dans la conception des réseaux, et le marché s'est rapidement approprié le SDN comme ensemble de solutions/architectures permettant de supprimer les frontières existantes entre les mondes des applications et du réseau. Alors que le déploiement d'applications est toujours plus aisé et dynamique.

Récemment, plusieurs travaux ont montré que l'implémentation de SDN dans les réseaux de véhicules peut apporter de la flexibilité, de la programmabilité et la mise à disposition d'interfaces de programmation d'applications permet de programmer les équipements du réseau en utilisant différents langages et de mieux supporter l'évolutivité.

En effet, notre travail vise à implémenter l'architecture SDN dans les réseaux VANET, on a utilisé le simulateur Mininet-wifi pour réaliser ce projet. Nous avons créé une topologie avec un certain nombre de nœuds avec un point d'accès et un contrôleur. Dans ce travail, on a focalisé à assurer que le contrôleur SDN est bien installé et fonctionne bien pour que les paquets sont bien reçus dans tout le réseau. Cependant, le développement d'un tel projet n'est jamais totalement achevé et certaines idées n'ont pas pu être réalisées à cause de contraintes de temps.

Bibliographies

- [1] CHAIB Noureddine, La sécurité des communications dans les réseaux VANET, UNIVERSITE ELHADJ LAKHDER - BATNA.
- [2] NAIMI Sabrine, Gestion de la mobilité dans les réseaux Ad Hoc par anticipation des métriques de routage, UNIVERSITE PARIS-SUD ,22 Juillet 2015.
- [3] MEHARZA Wafa, Etude de la mobilité dans les réseaux Ad hoc véhiculaires VANET, UNIVERSITE BADJI MOKHTAR-ANNABA ,2019 .
- [4] TAHAR ABBES Mounir, Proposition d'un protocole de routage a économie d'énergie dans un réseau hybride GSM+Ad Hoc.
- [5] LARROUNI Youssef, Détection d'intrusions dans les réseaux véhiculaires sans fil, UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES ,JUN 2017.
- [6] MOGHRAOUI Kahina, Gestion de l'anonymat des communications dans les réseaux véhiculaires Ad Hoc sans fil (VANETs), UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, JUILLET 2015.
- [7] ALIOUA Ahmed, Intégration du Software-Defined Networking (SDN) dans les réseaux de véhicules (VANETs), UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE, le 21 Février 2019 .
- [8] BOUIDA Hafida, Étude et mise en œuvre d'une solution SDN: Application de Gestion de VLANs, UNIVERSITE ABOU BAKR BELKAID-TELEMCEN, 2017.
- [9] JEROME Durand, «Le SDN pour les nuls», Montpellier, JRES 2015.
- [10] IHSSANE Choukri, Mohammed Ouzzif, Khalid Bouragba, Software Defined Networking (SDN): Etat de L'art, Université Hassan II, Casablanca, Maroc, 2019.
- [11] AICHAOUI Anis, AITBELKACEM Yanis, Etude et implémentation d'une architecture SDN LAN, UNIVERSITE Mouloud MAMMERI DE TIZI- OUZOU, 2018.
- [12] MERIDJI Rania, Etude exploratoire et mise en œuvre des solutions basées sur SDN pour Groupe Sonelgaz, université BLIDA 1.
- [13] SOUFIAN TOUFGA, Vers des réseaux véhiculaires programmables via la technologie SDN, 2019.
- [14] BEN CHAHED Seifeddine, Mise en œuvre des aspects de gestion des réseaux définis par logiciels (réseaux SDN), UNIVERSITÉ DE MONTRÉAL, OCTOBRE 2015.

- [15] RIF Karim, Les Modèles de propagation pour la communication V2V, UNIVERSITE ABDELHAMID IBN BADIS – MOSTAGANEM, 2015/2016.
- [16] CHIHI Ines, Etude de l'attaque (Black Hole) sur le protocole de routage VADD, UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES ,2017 .
- [17] BEKTACHE Djamel, Application et Modélisation d'un protocole de communication pour la sécurité routière, UNIVERSITE BADJI MOKHTAR ANNABA, 2013/2014.
- [18] MOUSAOUI Djilali, Réseaux véhiculaires en Cloud : gestion de la sécurité, UNIVERSITE ABOU BAKR BELKAID-TELEMCEN ,2019.
- [19] BOUCEFAR Slimane, BOUCEFAR Walid, La qualité de service dans les réseaux véhiculaires (VANET), UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, le 04/07/ 2017.
- [20] BECAYE Dioum, Effets de la mobilité sur les protocoles de routage dans les réseaux ad hoc, Université MOULOUD MAMMERI de TIZI OUZOU.
- [21] ABBACI Mohamed Lamine et SBAHI Mohamed Ali, Etude de l'impact du modèle de Nkagami sur les performances des protocoles de routage dans les VANET, UNIVERSITE BADJI MOKHTAR ANNABA, 2020
- [22] MESSAOUDI Yamina, Simulation et évaluation des performances des protocoles de routage AODV, OLSR et GPCR pour les réseaux VANETs sous NS-3 et SUMO, Université A/Mira de Béjaia , 2016-2017.
- [23] www.sdxcentral.com.