



UNIVERSITÉ KASDI MERBAH OUARGLA



Faculté des Nouvelles Technologies de l'Informatique et

De la communication

Département d'Informatique et des Technologies de l'Informatique

## Mémoire

En vue de l'obtention du diplôme de

MASTER

**Domaine** : Informatique et Technologie de l'Information

**Filière** : Informatique

**Spécialité** : Administration et Sécurité Réseaux informatique

**Présenté par** : Fetni Ahmed Amine

**Djafour Fatma Zahra**

## Thème

**La dissémination des données basée sur  
SDN pour l'internet des véhiculés**

Soutenu publiquement le : .../06/2023

Devant le jury composé de :

Dr. Nadjat Azzaoui	Université de Ouargla	<b>Rapporteur</b>
M. Kelili khalida	Université de Ouargla	<b>Examineur</b>
M. Djedjai Hmida	Université de Ouargla	<b>Président</b>

Année universitaire : 2022/2023

## *Remerciements et Dédicaces*

Après un temps difficile, mon parcours universitaire s'est terminé. Et me voilà, en terminant mes recherches de fin d'études avec vigueur et activité.

- Je suis reconnaissant à tous ceux qui m'ont aidé de près et de loin dans ma carrière.
- À Mme **Azzaoui Nadjjet**, Nous tenons à remercier sincèrement, elle a eu le plus grand rôle en me soutenant et en me fournissant toutes les informations nécessaires long de la réalisation de ce mémoire, sans qui ce mémoire n'aurait jamais vu le jour.
- A celle qui a placé le Seigneur - Gloire à Lui - Paradis sous ses pieds, et l'a vénéré dans son livre sacré, à qui je préfère à moi-même, et pourquoi pas ; Elle s'est sacrifiée toute sa vie pour moi et elle n'a épargné aucun effort pour me rendre toujours heureuse.

(**Ma mère bien-aimée**) Que Dieu la bénisse

- Au propriétaire d'une biographie riche, d'une pensée éclairée, d'un bon esprit et de bonne humeur, car il a été le premier supporter de ma réussite à l'enseignement supérieur, et il n'a pas été avare de moi tout au long de sa vie

(**Mon père bien-aimé**), que Dieu bénisse sa vie.

- À **ma famille, mes amis, mes proches** et tous ceux qui m'ont soutenu...

Je vous les dédie ma recherche de fin d'étude.

**F. Ahmed Amine & DJ. Fatma Zahra**

## TABLES DES MATIERES

Tables des matières	i
Liste des figures	iii
Liste des Tables	iv
Résumé	1
Introduction générale	1

### Chapitre 1 : Généralités sur les réseaux véhiculaires et le SDN

1. INTRODUCTION :	2
2. Les réseaux Ad Hoc :	2
2.1. Les réseaux Ad Hoc :	2
2.2. Les réseaux MANET :	3
2.3. Les réseaux véhiculaires ad hoc :	3
3. les composants et les modes de communication :	3
3.1. les équipements :	3
3.1.1. Véhicule intelligent :	3
3.1.2. Les composants de la route :	5
3.2. les modes de communication dans les réseaux VANET :	6
3.3. Les types des messages :	7
3.4. Technologies et standards de communication :	8
3.4.1. Technologie DSRC :	8
3.4.2. Réseaux cellulaire :	9
3.5. Applications :	9
4. Evolution des réseaux véhiculaires :	9
4.1. Réseaux véhiculaire hétérogène :	10
4.2. Internet des objet (IOT) :	10
4.3. Internet des véhicules (IOV) :	11
5. Caractéristiques des réseaux Véhiculaires :	12
6. Le routage dans les réseaux véhiculaire :	13
7. réseau défini par logiciel (SDN) :	14
7.1. Définition :	14
7.2. Architecture :	14
7.2.1. Couche d'application :	14
7.2.2. Couche de contrôle :	14

7.2.3. Couche de données :	15
7.3. Le controleur SDN :	15
7.4. les interfaces de communication :	15
7.5. OpenFlow :	16
7.6. Apports de SDN pour les réseaux véhiculaire :	16
8. les travaux connexes :	17
9. conclusion :	22

## **Chapitre 2 : Revue de littérature sur SDN dans les réseaux IOV**

1. INTRODUCTION :	22
2. Définitions SDN :	22
3. Architectures SDN-IoV :	23
4. Les types de contrôleurs SDN :	29
5. Conclusion :	30

## **Chapitre 3 : Implémentation de SDN in IOV**

1. Introduction :	32
2. l'approche proposée :	32
3. Environnements de travail :	34
3.1. VIRTUELLE BOX :	34
3.2. UBUNTU :	34
3.3. Mininet-WiFi :	34
3.4. POX Controller :	34
3.5. SUMO :	35
3.6. Python :	35
4. Processus de simulation :	35
4.1. L'intégration des programmes et les version :	35
5. Les étapes d'installation :	36
5.1. Les commandes principales d'installation Mininet-wifi :	36
5.2. Les commandes principales pour d'installation du SUMO :	37
6. Les étapes du simulation :	38
6.1. La création de la topologie :	38
6.2. lancement de la topologie :	41
6.3. Evaluation de performance de réseau SDN vs le réseau de véhiculaire traditionnel :	43

7. Résultats des simulations :	47
8. CONCLUSION :	49
Conclusion générale :	50

## LISTE DES FIGURES

Figure 1: Les réseaux VANETs .....	3
Figure 2 : les composants d'un Véhicule intelligent .....	4
Figure 3: Road Side Unit.....	5
Figure 4 : Station de base (SB).....	5
Figure 5: Mode Véhicule à véhicule.....	6
Figure 6: Mode Véhicule à infrastructure .....	6
Figure 7: Mode hybride.....	7
Figure 8 : la FCC des bandes de fréquences DSRC pour VANET.....	8
Figure 9: Réseaux véhiculaire hétérogène .....	10
Figure 10: Internet des objet (IOT) .....	11
Figure 11: Réseau traditionnel et réseau défini par logiciel SDN .....	14
Figure 12: Architecture de SDN/open flow .....	16
Figure 13 : Architecture des réseaux SDN.....	16
Figure 14 : Architecture de logiciel. ....	23
Figure 15: Architecture de l'IoV 5G basée sur Fog-Cloud et SDN.....	20
Figure 16: Allocation des ressources .....	24
Figure 17: Architecture hiérarchique basée sur les SDN et les réseaux IoV .....	24
Figure 18: Architecture Générale .....	25
Figure 19: Fonctionnement de l'architecture proposée. ....	25
Figure 20: Architecture de réseau 5G proposée. ....	26
Figure 21: Architecture hiérarchique pour les IoV.....	26
Figure 22: Architecture de SDVN .....	27
Figure 23: Architecture de l'HÉMT.....	27
Figure 24: Architecture de réseau intégrée 5G compatible SDN-VANET. ....	28
Figure 25: System architecture. ....	18
Figure 26 : L'architecture proposée pour SD-IOV .....	18
Figure 27: Architecture du Système SD-IOV pour la Dissémination des Messages d'Alerte.....	19
Figure 28: opération de la sélection du réseau.....	20
Figure 29: L'approche proposée.....	33
Figure 30: installer Mininet wifi .....	37
Figure 31: exemple topologie dans Mininet-wifi.....	37
Figure 32: installation sumo. ....	37
Figure 33: les bibliothèques de mininet-wifi. ....	38
Figure 34: la définition les paramètres de topologie. ....	38
Figure 35 : la création des Nœuds (car & rsu). ....	38
Figure 36: main de POX n°1.....	39
Figure 37: main de POX n°2.....	39
Figure 38 : La création les Nœuds de contrôle. ....	39
Figure 39 : la configuration de modelé.....	39
Figure 40 : Création des liens.....	40
Figure 41 : Démarrage le sumo .....	40
Figure 42 : Démarrage le réseau .....	40

Figure 43 : Démarrage la télémétrie.....	40
Figure 44 : Le lancement du contrôleur POXI .....	41
Figure 45 : lancement la simulation.....	41
Figure 46 : interface SUMO.....	41
Figure 47 : Visualisation du Graph à l'instant t1 .....	42
Figure 48 : Visualisation du Graph à l'instant t2 .....	42
Figure 49 : fenêtre de contrôleur POX.....	42
Figure 50 : ligne interface de commande (CLI).....	43
Figure 51 : Test la connectivité.....	44
Figure 52 : les différents RTT de scenario SDN.....	44
Figure 53 : écoute du serveur sur UDP .....	45
Figure 54 : la connectivité client-serveur.....	45
Figure 55 : Exemple de résultat PDR.....	46
Figure 56 : comparaison de Latence entre les deux scenarios.....	47
Figure 57 : comparaison le PDR entre les deux scenarios .....	48
Figure 58 : comparaison le débit entre les deux scenarios .....	49

## LISTE DES TABLES

Tableau 1 : Les avantages/ Les inconvénients d'IOT.....	11
Tableau 2: des architectures SDN.....	28
Tableau 3: Comparaison entre les propriétés des contrôleurs SDN .....	30
Tableau 4: comparaison les travaux connexes.....	20
Tableau 5: LES PROGRAMMES ET LES VERSIONS .....	35
Tableau 6: Les PARAMETRES.....	36

## *Résumé*

*L'un des principaux objectifs de la recherche scientifique est la préservation de la vie humaine. Les véhicules sont devenus indispensables dans notre quotidien, mais ils représentent également un danger pour notre sécurité. Malgré ce risque, nous ne pouvons-nous passer des véhicules en raison de leur importance.*

*Pour faire face aux problèmes de congestion routière et d'accidents, les réseaux dédiés de véhicules (VANETs) ont été développés. Ces réseaux permettent aux véhicules de communiquer entre eux et avec les infrastructures, créant ainsi un réseau dynamique et autoorganisé. Les VANETs sont conçus pour prendre en charge diverses applications telles que la gestion du trafic, la prévention des collisions, les services d'urgence et les divertissements, dans le but d'améliorer les déplacements, la sécurité routière et l'efficacité du système de transport.*

*L'Internet des véhicules (IoV) étend le concept des VANETs en intégrant les véhicules dans l'écosystème plus large de l'Internet des objets (IoT). L'IoV utilise la connectivité Internet, le cloud computing et l'analyse de données pour permettre aux véhicules d'échanger des informations avec d'autres véhicules, les infrastructures et divers appareils IoT. Son objectif est d'améliorer l'efficacité des transports, la sécurité routière et l'expérience globale de conduite.*

*Cependant, les architectures actuelles des réseaux de véhicules rencontrent des problèmes d'évolutivité en raison des défis liés à la diffusion des données à grande échelle. Ces architectures sont rigides, difficiles à gérer et manquent de flexibilité et d'adaptabilité en raison de l'hétérogénéité des technologies utilisées dans les véhicules. Cela entraîne des problèmes tels que la congestion du réseau et des retards dans la transmission des messages de sécurité.*

*Pour remédier à ces problèmes, les réseaux définis par logiciel (SDN) ont émergé en tant que solution. Le SDN est une technologie open source facilement programmable qui sépare le plan de contrôle du plan de données. Cette séparation facilite la gestion et la programmation des réseaux véhiculaires. Le contrôleur SDN joue un rôle central en assurant la gestion globale et le contrôle efficace des ressources du réseau. En utilisant le contrôleur SDN et le protocole OpenFlow, les performances globales du réseau sont améliorées, offrant des avantages tels qu'une sélection efficace des itinéraires, une gestion de la congestion et une communication rapide et fiable. La couche de données du SDN est responsable uniquement de la transmission des paquets, que la connexion entre les appareils soit filaire ou sans fil.*

*Ainsi, dans cette étude, nous proposons une approche pour la dissémination des données basée sur SDN pour l'internet des véhicules. Cette approche permet de surmonter les limitations des architectures traditionnelles et offre une meilleure gestion et dissémination des données.*

**Mots-clés :** VANETs, IoV, IoT, Dissémination des données, SDN.

## *Abstract*

*One of the main goals of scientific research is the preservation of human life. Vehicles have become indispensable in our daily lives, but they also represent a danger to our safety. Despite this risk, we cannot do without vehicles because of their importance.*

*To deal with the problems of road congestion and accidents, vehicular ad-hoc networks (VANETs) have been developed. These networks allow the vehicles to communicate with each other and with the infrastructures, thus creating a dynamic and self-organized network. The VANETs are designed to support various applications such as traffic management, collision prevention, emergency services and entertainment, with the aim of improving travel, road safety and the efficiency of the transport system.*

*The Internet of Vehicles (IoV) extends the concept of VANETs by integrating vehicles into the wider ecosystem of the Internet of Things (IoT). IoV uses Internet connectivity, cloud computing and data analytics to enable vehicles to exchange information with other vehicles, infrastructure and various IoT devices. STI objective is to improve transport efficiency, road safety and the overall driving experience.*

*However, current vehicle network architectures encounter scalability problems due to the challenges associated with large-scale data dissemination. These architectures are rigid, difficult to manage and lack flexibility and adaptability due to the heterogeneity of the technologies used in vehicles. This leads to problems such as network congestion and delays in the transmission of security messages.*

*To remedy these problems, software-defined networks (SDN) have emerged as a solution. SDN is an easily programmable open-source technology that separates the control plane from the data plane. This separation facilitates the management and programming of vehicular networks. The SDN controller plays a central role in ensuring the overall management and effective control of network resources. By using the SDN controller and the OpenFlow protocol, the overall network performance is improved, offering advantages such as efficient route selection, congestion management and fast and reliable communication. The SDN data plan is responsible only for the transmission of packets, regardless of whether the connection between the devices is wired or wireless.*

*Thus, in this study, we propose an approach for the data dissemination based on software-defined networks for the Internet of vehicles. This approach makes it possible to overcome the limitations of traditional architectures and offers a better management and dissemination of data.*

**Keywords:** *VANETs, IoV, IoT, Data Dissemination, SDN.*



## ملخص

من الأهداف الرئيسية للبحث العلمي الحفاظ على حياة الإنسان. أصبحت المركبات لا غنى عنها في حياتنا اليومية، لكنها تمثل أيضا خطرا على سلامتنا. على الرغم من هذا الخطر، لا يمكننا الاستغناء عن المركبات بسبب أهميتها.

للتعامل مع مشاكل ازدحام الطرق والحوادث، تم تطوير شبكات مخصصة للمركبات. تسمح هذه الشبكات للمركبات بالتواصل مع بعضها البعض ومع البنى التحتية، وبالتالي إنشاء شبكة ديناميكية ومنظمة ذاتيا. تم تصميم الشبكات المخصصة للمركبات لدعم التطبيقات المختلفة مثل إدارة حركة المرور ومنع الاصطدام وخدمات الطوارئ والترفيه، بهدف تحسين السفر والسلامة على الطرق وكفاءة نظام النقل.

يوسع إنترنت المركبات مفهوم الشبكات المخصصة للمركبات من خلال دمج المركبات في النظام البيئي الأوسع لإنترنت الأشياء. يستخدم إنترنت المركبات الاتصال بالإنترنت والحوسبة السحابية وتحليلات البيانات لتمكين المركبات من تبادل المعلومات مع المركبات الأخرى والبنية التحتية وأجهزة إنترنت الأشياء المختلفة. هدفها هو تحسين كفاءة النقل والسلامة على الطرق وتجربة القيادة الشاملة.

ومع ذلك، تواجه بنى شبكة المركبات الحالية مشاكل قابلية التوسع بسبب التحديات المرتبطة بنشر البيانات على نطاق واسع. هذه البنى جامدة ويصعب إدارتها وتفتقر إلى المرونة والقدرة على التكيف بسبب عدم تجانس التقنيات المستخدمة في المركبات. هذا يؤدي إلى مشاكل مثل ازدحام الشبكة والتأخير في إرسال رسائل الأمان.

لمعالجة هذه المشاكل، ظهرت الشبكات المعرفة بالبرمجيات كحل، وهي تقنية مفتوحة المصدر قابلة للبرمجة بسهولة تفصل مستوى التحكم عن مستوى البيانات ويسهل إدارة وبرمجة شبكات المركبات، حيث تلعب وحدة التحكم في الشبكات المعرفة بالبرمجيات دورا مركزيا في ضمان الإدارة العامة والتحكم الفعال في موارد الشبكة. باستخدام وحدة تحكم الشبكات المعرفة بالبرمجيات بواسطة البروتوكول أوبن فلو، يتم تحسين أداء الشبكة بشكل عام، مما يوفر مزايا مثل اختيار الطريق الفعال، وإدارة الازدحام والاتصالات السريعة والموثوقة. طبقة البيانات للشبكات المعرفة بالبرمجيات مسؤولة فقط عن إرسال الحزم، بغض النظر عما إذا كان الاتصال بين الأجهزة سلكيا أو لاسلكيا.

وبالتالي في هذه الدراسة، نقترح بنية لنشر البيانات على أساس الشبكات المعرفة بالبرمجيات لإنترنت المركبات، هذا النهج يجعل من الممكن التغلب على قيود البنى التقليدية ويوفر إدارة ونشر أفضل للبيانات.

**كلمات البحث:** الشبكات المخصصة للمركبات (VANETs)، إنترنت الأشياء (IoT)، إنترنت المركبات (IoV)، نشر البيانات (dissémination des données)، الشبكة المعرفة بالبرمجيات (SDN).

# *Introduction générale*

---

Récemment, les accidents de la circulation sont devenus un grave problème pour l'humanité en enregistrant des pourcentages élevés de décès, et études ont montré qu'il est possible d'éviter certains accidents en utilisant les nouvelles technologies pour alerter les conducteurs une fraction de seconde plus tôt.

Avec le développement des technologies informatiques et de communication et à Grâce à l'apparition d'un réseau de véhicules ad hoc (VANET), Les véhicules sont devenus des entités intelligentes. L'intégration de capteurs, de fonctions de communication et de réseau dans les véhicules leur a donné la possibilité d'interagir les uns avec les autres [1]. L'internet des véhicules est un réseau de véhicules utilisé pour échanger des informations via diverses technologies d'accès radio. Où IoV est considéré comme une pile complète de VANETS. L'un des principaux problèmes du Système de transport intelligent (STI) est l'augmentation de la congestion, en particulier dans les zones urbaines, ce qui a un impact significatif sur le temps de trajet et la sécurité. Il est donc nécessaire d'optimiser, de contrôler et de surveiller le trafic des véhicules dans l'intérêt des utilisateurs du monde entier [2].

Les réseaux véhiculaires souffrent de problèmes d'évolutivité et d'un manque de flexibilité et d'adaptabilité au contrôle car il est très difficile de dissémination des données à grande échelle. De telles structures sont rigides, difficiles à gérer.

Un nouveau paradigme d'architecture de réseau, la mise en réseau définie par logiciel (SDN), a été proposé comme l'une des technologies les plus prometteuses pour relever et relever les défis de la mise en réseau automobile. Le SDN est principalement basé sur la séparation physique du plan de contrôle, qui gère la gestion du réseau, et du plan de données, qui gère la distribution et la transmission des informations. Cette technologie est l'une des technologies qui contribuent de manière significative à la diffusion des données dans les réseaux véhiculaires avec une grande efficacité [3]. Les réseaux définis par logiciel (SDN) basés sur les réseaux véhiculaires peuvent jouer un rôle essentiel dans la résolution les problèmes de circulation.

Ce travail se concentre sur l'exploitation des avantages du SDN basé sur le réseau des véhicules pour améliorer la gestion et les performances des futurs véhicules.

L'organisation de la thèse : Il est organisé en trois chapitres. Le premier chapitre traite des concepts généraux des réseaux véhiculaires. Le deuxième chapitre fournit un examen approfondi des travaux de recherche pertinents. Enfin, dans le dernier chapitre, nous proposons une approche spécifique pour la dissémination des données basé sur SDN pour l'internet des véhicules, incluant l'implémentation une simulation de réseau et comparant les performances du réseau SDN à celles d'un réseau de véhicules traditionnel, Les résultats obtenus seront également affichés et analysés.

# Chapitre

## 1

# *Généralités sur Les Réseaux Véhiculaires et Le SDN*

---

## 1. INTRODUCTION :

Au cours des dernières années les réseaux véhiculaires sont devenus un domaine de recherche actif, ils permettent de réaliser une communication sans fil rapide entre les véhicules ce qui a motivé de nombreux chercheurs à développer les paradigmes de communication qui répondent aux exigences émergentes des véhicules. Parallèlement, le véhicule est devenue le moyen de transport le plus utilisé, mais malheureusement, il est accompagné d'un certain nombre de problèmes (accidents, violation, embouteillages, etc.). Ces problèmes ont incité les fabricants et les chercheurs à améliorer la sécurité des systèmes de transport, à développer des technologies de communication et à stocker nos informations afin d'offrir des routes et une conduite plus sûre[13].

Le paradigme SDN (Software-Defined Networking) est une approche de réseau qui vise à séparer le plan de contrôle du plan de données dans les réseaux véhiculaires, Cela permet une gestion centralisée et programmable du réseau, offrant ainsi une plus grande flexibilité et une meilleure agilité.

L'objectif de ce chapitre est de prendre une vue générale sur les réseaux véhiculaire et de présenter un bref survol sur le paradigme de software-defined networking dans les réseaux véhiculaires.

## 2. LES RESEAUX AD HOC :

### 2.1 LES RESEAUX AD HOC :

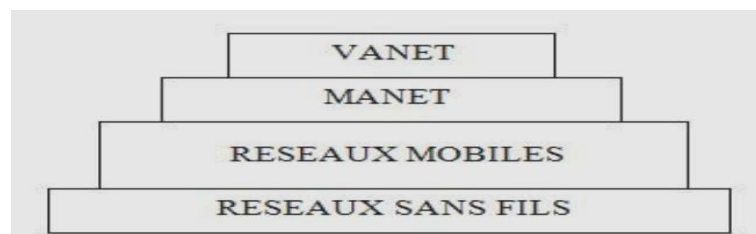
Les réseaux ad hoc (en latin : « qui va vers ce vers quoi il doit aller », c'est-à-dire « formé dans un but précis »), sont des réseaux sans-fil capables de s'organiser spontanément et de manière autonome dans l'environnement dans lequel ils sont déployés sans infrastructure définie préalablement, criés à la demande pour répondre à un besoin spécifique. 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 » [14].

## 2.2 LES RESEAUX MANET :

Le réseau mobile ad hoc, appelé généralement MANET (Mobile Ad hoc Network) est un système autonome qui se compose d'un nœud mobile dynamique interconnecté par des liens sans fil sans l'utilisation de l'infrastructure fixe et sans gestion centralisée [15]. Les nœuds sont libres à se déplacer de façon aléatoire et, par conséquent, peuvent changer la structure du réseau rapidement et de manière imprévisible.

## 2.3 LES RESEAUX VEHICULAIRES AD HOC :

Un réseau véhiculaire ou VANET (Vehicular Ad-hoc Network) est une classe particulière des réseaux mobiles Ad-hoc (MANET) ou les nœuds mobiles de réseau véhiculaire sont des véhicules (intelligents) équipés de calculateurs, de carte réseau et capteurs qui permettent la communication sans fil et qui permettent aussi l'échange d'informations entre les véhicules et avec les infrastructures routières environnantes. Ces réseaux sont conçus pour améliorer la sécurité, l'efficacité et le confort des déplacements en permettant aux véhicules de communiquer entre eux et avec leur environnement .voir la **figure 1** ,Dans un réseau de véhicules, les véhicules peuvent établir des connexions sans fil directes ou indirectes pour échanger des informations telles que leur position, leur vitesse, leur direction et d'autres données liées à la sécurité routière [4].



**Figure 1:** Les réseaux VANETs.

## 3. LES COMPOSANTS ET LES MODES DE COMMUNICATION :

### 3.1. LES EQUIPMENTS :

#### 3.1.1. VEHICULE INTELIGENT :

Dans le contexte des réseaux véhiculaires, Une véhicule intelligente, fait référence à un véhicule équipé de technologies avancées qui lui permettent de communiquer et de collaborer avec d'autres véhicules et infrastructures routières, voir la **figure 2**. Ces véhicules intelligents utilisent des capteurs, des systèmes de communication sans fil et des logiciels embarqués pour collecter et échanger des informations en temps réel [8].

- ❖ **Enregistreur de données d'événement (EDR)** : utilisé pour enregistrer des paramètres importants : vitesse, accélération, événements importants (accidents...). ces données peuvent être utilisées pour les reconstructions des événements.
- ❖ **Système de positionnement** : localisation des véhicules.
- ❖ **Radar arrière** : est un système qui donne une visibilité complète de ce qui se passe à l'arrière d'un véhicule. Sur certains modèles, il aide le conducteur à stationner le véhicule.
- ❖ **Plateforme informatique** : procession des entrées provenant des différentes composantes utilisées, générer des informations utiles à échanger avec d'autres véhicules ou avec l'infrastructure.
- ❖ **Afficher (display)** : ensemble d'équipements proposés en option par le constructeur à l'acheteur d'un véhicule, et dont composition ne peut être modifiée.
- ❖ **Radar avant** : détection des obstacles (<200m). Facilité de communication [9].

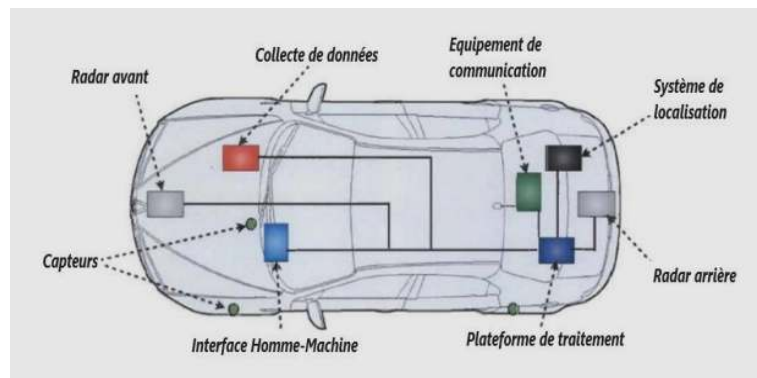


Figure 2 : les composants d'un Véhicule intelligent.

❖ **Unité Embarquée (OBU) :**

L'unité embarquée (OBU) est une unité embarquée dans un véhicule intelligent. Son but est que le véhicule utilise une série de programmes pour localiser, calculer, enregistrer et envoyer des messages via une interface réseau. Dans un réseau VANET [23].

❖ **Central Authority (CA) :**

Le CA (Central Authority) représente l'autorité de confiance dans le réseau véhiculaire VANET. Le CA joue le rôle d'un serveur qui assure la sécurité des différents services tels que la délivrance des certificats, des clés de communication et le stockage de certaines données. L'AU est un dispositif électronique installé dans les véhicules pour assurer les communications [25].

### 3.1.2. LES COMPOSANTS DE LA ROUTE :

#### ❖ Road Side Unit (RSU) :

une unité routière (RSU) est une entité localisée et installée en bordure d'une route. Ces fonctions fournissent des points d'accès au réseau et sont déployées le long des routes. Chaque RSU est conçu pour transmettre des messages aux véhicules dans sa portée radio. Ces messages contiennent des informations sur les conditions météorologiques ainsi que sur les conditions routières (vitesse maximale, autorisation de dépassement, etc.) voir la **figur3** [23].



**Figure 3:** Road Side Unit.

#### ❖ Station de base (SB) :

Les Stations de Base sont des équipements installées sur un site et muni d'une antenne émettrice-réceptrice avec lequel communiquent les appareils mobiles, pour avoir accès à un réseau de télécommunications, les stations de base permettent la communication bidirectionnelle entre les véhicules et les infrastructures routières. Elles collectent des données, diffusent des informations, contribuent à la gestion du trafic et prennent en charge diverses applications avancées[26], voir la **figure 4**.

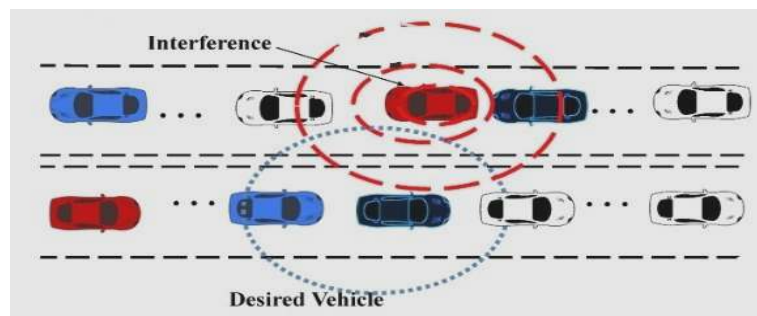


**Figure 4 :** Station de base (SB).

### 3.2. LES MODES DE COMMUNICATION DANS LES RESEAUX VANET :

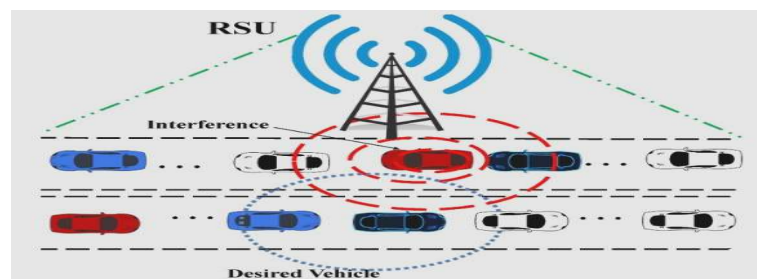
Dans les réseaux véhiculaires, on peut distinguer trois modes de communication, les communications Véhicule-à-Véhicule (V2V), les communications Véhicule-à-Infrastructure (V2I) et hybride(V2X).

- I. **Mode Véhicule à véhicule : V2V (Vehicle-to-Vehicle) :** Il s'agit d'un mode de communication où les véhicules échangent des informations directement entre eux. Les véhicules utilisent des technologies sans fil, telles que le Wi-Fi ou le Bluetooth, pour communiquer. Les informations échangées peuvent inclure la position, la vitesse, la direction, les intentions de changement de voie, les avertissements de sécurité, etc. Le V2V permet aux véhicules de détecter et de réagir aux situations de trafic en temps réel, améliorant ainsi la sécurité et l'efficacité de la circulation, voir la **figure 4** [29][30].



**Figure 5:** Mode Véhicule à véhicule.

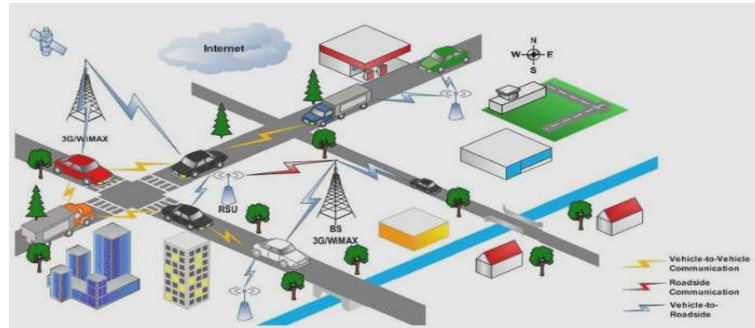
- II. **Mode Véhicule à infrastructure : V2I (Vehicle-to-Infrastructure) :** Dans ce mode de communication, les véhicules échangent des informations avec les infrastructures routières environnantes. Les infrastructures peuvent inclure des stations de base, des feux de signalisation, des panneaux de signalisation et des capteurs routiers, etc. Les informations échangées peuvent comprendre des données de circulation en temps réel, des indications sur les conditions de la route, des alertes de sécurité, des recommandations d'itinéraire, etc. Le V2I permet aux conducteurs d'obtenir des informations précieuses sur l'environnement routier pour une conduite plus sûre et plus efficace, voir la **figure 5** [36].



**Figure 6:** Mode Véhicule à infrastructure.

- III. **Mode hybride (V2X) : V2X (Vehicle-to-Everything) :** Le mode de communication V2X englobe à la fois le V2V et le V2I, mais également d'autres formes de communication. Il permet aux véhicules de communiquer avec une variété d'entités, notamment les piétons

(V2P), les réseaux de communication (V2N), les serveurs cloud (V2C), etc. Le V2X étend les possibilités de communication des véhicules et facilite l'échange d'informations avec différents acteurs de l'écosystème routier. Cela permet d'améliorer la sécurité, l'efficacité et le confort des déplacements. voir la **figure 7** [29][30].



**Figure 7:** Mode hybride.

### 3.3 LES TYPES DES MESSAGES :

Les nœuds qui forment un réseau sans fil véhiculaire généreront et échangeront des messages. Selon l'application et le contexte environnemental, un véhicule a la capacité de transmettre et de recevoir des messages de contrôle, d'avertissement et autres.

#### ❖ Les Messages de contrôle :

Les messages de contrôle sont envoyés correctement. Traditionnellement, chaque véhicule envoie un message de contrôle dans les 100 millisecondes. Le rapport vous demande également de déterminer l'emplacement, la direction, l'état et l'itinéraire du véhicule. Chaque véhicule peut voir l'emplacement des zones où les informations peuvent être contrôlées. Les accidents ou incidents peuvent également entraîner des dommages et des dommages au véhicule. Le message de contrôle correspond au message HELLO dans le protocole de routage. Chaque véhicule doit être directement associé à une véhicule. Bien entendu, les messages de contrôle ne sont pas transférés et utilisant une émission de saut.

#### ❖ Messages d'alerte :

Lorsqu'un événement est identifié, un message d'alerte est créé. Cela peut être dû à la découverte d'un accident, d'un obstacle ou à la réception d'un autre message d'alerte. Pour assurer la viabilité à long terme des avertissements, des messages d'alerte doivent être envoyés régulièrement. En conséquence, les véhicules désignés pour la retransmission des messages déclencheront périodiquement l'alarme. Par conséquent, les messages d'alerte doivent être de petite taille afin qu'ils puissent être transmis le plus rapidement possible. Le message contient notamment les coordonnées du lieu de l'accident et le réglage de la zone de rediffusion.



### ❖ Autres messages :

Tous les messages qui ne sont pas des alertes ou des avertissements entrent dans cette catégorie. Ces messages ne sont généralement pas répétés régulièrement. Par exemple, l'envoi de messages de transaction financière ou d'e-mails. Tous les messages reçus seront stockés dans un cache des messages récemment reçus. Chaque message sera associé à une durée de vie dans le cache [37].

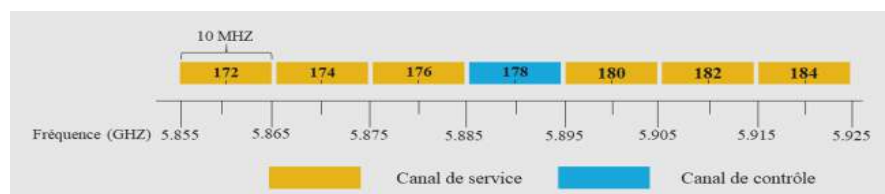
## 3.4 TECHNOLOGIES ET STANDARDS DE COMMUNICATION :

Pour satisfaire les exigences des applications véhiculaires, de nombreuses normes ont été développées telles que la technologie DSRC (Dedicated Short-Range Communication) et le standard WAVE (Wireless Access in Vehicular Environment).

### 3.4.1 TECHNOLOGIE DSRC :

Dedicated Short Range Communication » en Anglais ou bien la communication dédiées à courte portée, est un radiocommunication à sens unique ou double sens, et à courte ou moyenne portées, spécialement conçues pour les systèmes de transport intelligent c'est-à-dire il est utilisé pour décrire la communication entre deux véhicules dans un réseau véhiculaires a équipé d'une unité intérieure (OBU) avec prise en charge des normes ce qui lui permet de partager des informations et des données avec d'autres véhicules. Le sigle DSRC désigne aussi l'ensemble de protocoles et de normes mis en jeu dans ce type de communication .a été proposée comme norme pour la communication à courte portée afin de prendre en charge la communication à faible latence (V2V et V2I). DSRC permet la communication dans une plage de 300 m à 1 km et prend en charge une vitesse de véhicule allant jusqu'à 200 km/h. De plus, l'utilisation du spectre DSRC est gratuite [7][27].

Conformément à la technologie DSRC, l'American Society for Testing and Materials standardization company (ASTM) ont proposé la première norme ASTM-DSRC, La couche physique est basée sur IEEE 802.11 a, tandis que la couche MAC est basée sur IEEE 802.11. Par la suite, IEEE spécifie la famille de protocoles IEEE 802.11 p, qui est basée sur IEEE 802.11 et ASTM-DSRC. La couche physique et la couche MAC de la norme IEEE 802.11 p ont été modifiées pour permettre la communication sans fil dans les réseaux *véhiculaires*. L'IEEE est définie ensuite par l'accès sans fil dans les environnements véhiculaires (WAVE), qui spécifie les protocoles de communication sans fil véhiculaire à chaque niveau de couche du modèle OSI. La **figure 8** illustre l'attribution par la FCC des bandes de fréquences DSRC pour VANET [33] :



**Figure 8** : la FCC des bandes de fréquences DSRC pour VANET.

### 3.4.2. RESEAUX CELLULAIRE :

Dans un environnement réseaux véhiculaires, les réseaux cellulaires peuvent jouer un rôle important en fournissant une connectivité aux véhicules, Les réseaux cellulaires tels que la 4G (LTE) et la 5G peuvent être utilisés pour la communication entre les véhicules et l'infrastructure, ainsi que pour permettre l'accès à Internet à bord des véhicules. Cependant, il est important de prendre en compte les défis techniques et d'optimiser les architectures de réseau et les protocoles de communication pour répondre aux exigences spécifiques des réseaux véhiculaires [38].

### 3.5. APPLICATIONS :

Les applications dans les réseaux véhiculaires sont divisées en trois catégories en fonction de leur étendue de service .

#### ❖ Application de gestion de la circulation routière :

Les applications de gestion du trafic routier Peuvent optimiser les itinéraires des véhicules sur la route en échangeant des informations entre les conducteurs, par exemple B. sur d'éventuels embouteillages, et ajuster les itinéraires en fonction des conditions de circulation routière [5].

#### ❖ Application d'information et de divertissant :

Les applications d'information et de divertissement sont conçues pour améliorer le confort des conducteurs et des passagers grâce aux informations, à la publicité et au divertissement fournis par ces applications. Par exemple, des applications qui fournissent des informations sur les installations routières, l'accès à Internet, les demandes de renseignements et d'autres services en temps réel.

#### ❖ Demande de sécurité routière :

Ces applications aident à minimiser le nombre d'accidents de la route et à assurer la sécurité des conducteurs et des passagers. Ils fournissent des informations qui alertent les conducteurs des obstacles, des véhicules collusoires ou des véhicules en approche. Le moment des avertissements pour ces applications est essentiel car il affecte la vie du conducteur [6].

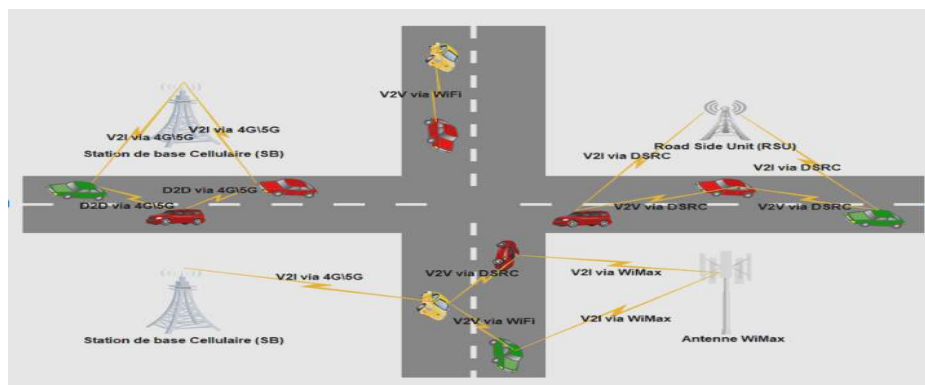
## 4. EVOLUTION DES RESEAUX VEHICULAIRES :

À cause de la haute mobilité des véhicules et la topologie hautement dynamique des VANETs, il est difficile de fournir une qualité de service (Quality of Service, QoS) satisfaisante pour les services STIs en s'appuyant uniquement sur un seul réseau d'accès sans fil à l'instar de DSRC. D'un côté, les communications V2V et V2I souffrent des problèmes d'évolutivité à cause de la couverture radio limitée de DSRC, le coûteux déploiement de l'infrastructure (RSUs) ainsi que les longs délais de bout en bout face au grand nombre de véhicules. De l'autre côté, bien que l'intégration des réseaux cellulaires, déjà existants et largement déployés dans les VANETs peut assurer une large couverture aux utilisateurs de réseau de véhicules avec un débit élevé. Cependant,

ils ne peuvent pas toujours garantir les besoins des services de sécurités exigeant des communications en temps réels stricts. Au fil du temps, les réseaux véhiculaires ont connu plusieurs évolutions et avancées technologiques [7][24].

#### 4.1 RESEAUX VEHICULAIRE HETEROGENE :

Les réseaux véhiculaires hétérogènes sont des systèmes de communication sans fil qui permettent la connectivité entre les véhicules et les infrastructures routières, ainsi qu'entre les véhicules eux-mêmes. Ces réseaux jouent un rôle crucial dans le développement des véhicules autonomes, de la sécurité routière et des services de transport intelligents et ont une technologie qui enrichit les VANETs traditionnels avec différentes technologies sans fil (DSRC, WiFi, WiMax, etc.) et principalement avec la technologie cellulaire (4G/5G) .Au fil du temps, les réseaux véhiculaires hétérogènes ont connu plusieurs évolutions et avancées technologiques ,voir la **figure 9** [7][24].

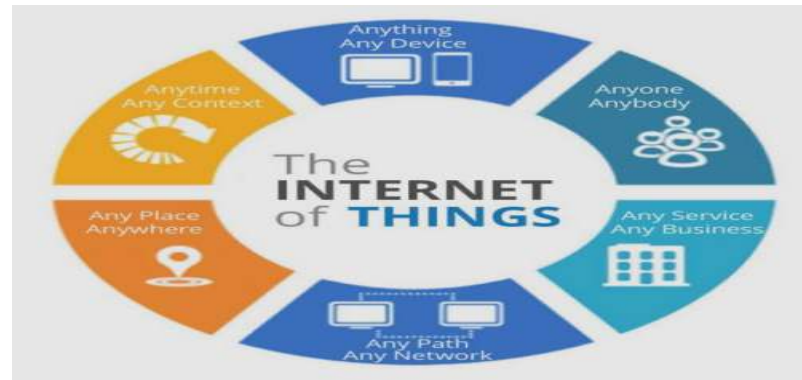


**Figure 9:** Réseaux véhiculaire hétérogène.

#### 4.2 INTERNET DES OBJET (IOT) :

L'internet des objets est une technologie de pointe dans le domaine des technologies de l'information, également appelée IOE (Internet of Everything). C'est le résultat du développement d'internet d'une manière qui n'a pas été prise en compte en raison de sa lente progression au début.

L'internet est simplement un réseau de réseaux qui sont reliés par des fils et des liaisons sans fil, et sous lesquels se trouvent un groupe de liens qui forment l'épine dorsale qui rend le monde entier à notre portée. L'émergence du terme internet des objets permettra aux choses qui nous entourent d'être connectées à internet, telles que les maisons et leurs appareils, les rues, les champs agricoles, les feux de signalisation, etc. [32].



**Figure 10:** Internet des objet(IOT) .

Dans le tableau suivant, nous mentionnons certains des avantages et des inconvénients de L'internet des objets.

**Tableau 1 :** Les avantages/ Les inconvénients d'IOT.

<b>Les avantages</b>	La possibilité d'accéder aux informations de n'importe où, à tout moment et sur n'importe quel appareil.
	Améliorer la communication entre les appareils électroniques connectés.
	Transférer des paquets de données sur un réseau connecté, ce qui permet d'économiser du temps et de l'argent.
<b>Les inconvénients</b>	Transférer des paquets de données sur un réseau connecté, ce qui permet d'économiser du temps et de l'argent. Perte ou vol possible d'informations.
	La possibilité de traiter avec un grand nombre d'appareils et cela constitue un grand défi.
	S'il y a une erreur dans le système, il est possible que tous les appareils qui y sont connectés soient endommagés.

### 4.3 INTERNET DES VEHICULE (IOV) :

IOV est une intégration de trois réseaux : un réseau inter-véhicules, un réseau interne, et réseau internet mobile pour les véhicule. Basé sur le concept de trois réseaux combinés en un seul, nous définissons l'internet des véhicules comme largement distribué système de communication sans fil et d'échange d'informations entre le véhicule V2X (X : véhicule, route, humain, internet ...) selon les protocoles de communication convenus et normes d'interaction de données (par exemple, Les avantages La possibilité d'accéder aux informations de n'importe où, à tout moment et sur n'importe quel appareil. Améliorer la communication entre les appareils électroniques connectés. Transférer des paquets de données sur un réseau connecté, ce qui permet d'économiser du temps et de l'argent. Les inconvénients Perte ou vol possible d'informations. La possibilité de traiter avec un grand nombre d'appareils et cela constitue un grand défi. S'il y a une erreur dans le

système, il est possible que tous les appareils qui y sont connectés soient endommagés. WAVE et standard technologies cellulaires potentielles). C'est un réseau intégré pour soutenir l'intelligence gestion du trafic, service d'information dynamique intelligent et véhicule intelligent control, qui est une application typique de la technologie Internet des Objets (IOT)[31].

## 5. CARACTERISTIQUES DES RESEAUX VEHICULAIRE :

Dans la référence [5] la présente section, nous vous proposons les éléments clés qui distinguent les réseaux véhiculaires :

- ❖ **Forte mobilité** : c'est le premier facteur qui distingue les réseaux véhiculaires des autres classes de réseaux sans fil. La vitesse des véhicules varie selon l'environnement, elle est en moyenne de 50km/h en zones urbaines et peut atteindre 130km/h sur autoroute. Bien que les mouvements des véhicules soient relativement prédictibles, l'impact de la mobilité sur la connectivité du réseau reste l'une des difficultés majeures des réseaux véhiculaires.
- ❖ **Caractéristiques inhérentes au canal radio** : dans les réseaux sans fil traditionnels, les échanges de données s'effectuent généralement dans des espaces ouverts sans obstacle ou dans des espaces clos en intérieur. Les communications dans les réseaux véhiculaires se font en environnement externe défavorable pour l'établissement des liens radio en raison de la multitude d'obstacles (forêts, montagnes, bâtiments . . .) notamment en zones urbaines. Ces obstacles causent une sévère dégradation de la qualité et de la puissance des signaux émis.
- ❖ **Connectivité intermittente** : une conséquence directe de la forte mobilité et des obstacles de l'environnement est une connectivité intermittente. Un lien établi entre deux entités du réseau peut rapidement disparaître en raison soit de la mobilité qui éloigne les deux entités communicantes, soit des obstacles qui empêchent la propagation du signal.
- ❖ **Diversité de la densité** : la densité des nœuds dans un réseau véhiculaire n'est pas uniforme mais à variation spatio-temporelle. La densité en milieu urbain est par exemple beaucoup plus élevée qu'en milieu rural. Le nombre de véhicules dans une intersection ou dans un embouteillage est plus important que sur des routes extra-urbaines où le trafic est souvent fluide. D'un point de vue temporel, la densité est par exemple différente selon qu'on considère la nuit ou la journée, les heures de pointe ou les heures creuses. Cette diversité de la densité rend difficile la conception de solutions génériques étant donné que les problèmes rencontrés à forte densité sont différents de ceux causés par la faible densité.
- ❖ **Partitionnement du réseau** : le problème du partitionnement du réseau survient essentiellement lorsque la densité des nœuds est faible. Les véhicules se déplacent alors en groupes isolés non connectés. Il devient dans ce cas difficile d'assurer les communications de bout-en-bout.
- ❖ **Broadcast Storm** : le problème du broadcast Storm a été considéré depuis longtemps dans les réseaux MANET et diverses solutions ont été proposées [35]. Ce problème se pose en particulier dans les protocoles de routage qui inondent le réseau avec les paquets de

contrôle à la recherche de routes. Les retransmissions successives des paquets causent de sérieuses redondances qui saturent le réseau. Dans le cas des réseaux véhiculaires, le problème du broadcast Storm se pose également au niveau application. En effet, les principaux services proposés pour les réseaux véhiculaires sont des services de sécurité qui se basent presque exclusivement sur les retransmissions de proche en proche des données. De plus, le problème du broadcast Storm est aggravé dans les réseaux à forte densité notamment dans des scénarios tels les embouteillages et les files d'attente aux intersections.

## 6. LE ROUTAGE DANS LES RESEAUX VEHICULAIRE :

Dans les réseaux véhiculaires, il existe différents types de routage qui peuvent être utilisés pour acheminer les communications entre les véhicules et l'infrastructure , Ces routages sont basés sur plusieurs techniques et sont divisés en 4 catégories : networking (par exemple SDN, IoT, 5G), intelligence (Deep Learning, Machine Learning, Réseaux de neurones...), traditionnel (clustering, pull, push, priorité...) et hybride (concaténation de techniques ou plusieurs).

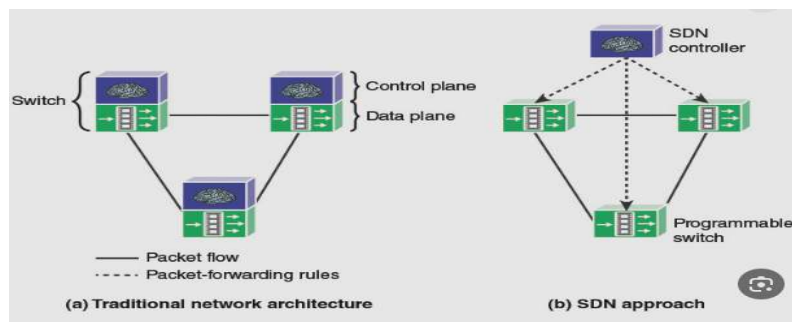
Voici trois types de routage couramment utilisés dans les réseaux véhiculaires :

- ❖ **Routage géographique** : Le routage géographique est largement utilisé dans les réseaux véhiculaires en raison de la nature de la mobilité des véhicules. Dans ce type de routage, les messages sont acheminés en fonction des coordonnées géographiques des véhicules. Les véhicules transmettent les messages aux véhicules voisins qui se trouvent plus près de la destination finale. Le routage géographique peut être efficace pour réduire les délais de transmission et minimiser la surcharge du réseau.
- ❖ **Routage basé sur le contenu** : Dans le routage basé sur le contenu, également appelé routage basé sur le nom, les messages sont acheminés en fonction du contenu ou des informations contenues dans les messages. Chaque message est associé à un identifiant unique, et les nœuds du réseau utilisent cet identifiant pour prendre des décisions de routage. Ce type de routage peut faciliter la recherche et l'accès à des informations spécifiques dans les réseaux véhiculaires.
- ❖ **Routage basé sur l'infrastructure** : Le routage basé sur l'infrastructure est utilisé lorsque les communications entre les véhicules dépendent principalement des points d'accès ou des relais d'infrastructure. Dans ce type de routage, les messages sont acheminés vers les points d'accès ou les relais qui relaient ensuite les messages vers les destinations appropriées. Le routage basé sur l'infrastructure peut être utilisé pour optimiser la connectivité dans les zones densément peuplées ou les zones où la couverture sans fil est limitée[34].

## 7. RESEAU DEFINI PAR LOGICIEL (SDN) :

### 7.1 DEFINITION :

(Software-Defined Networking) est une technologie de réseau qui permet de centraliser la gestion et le contrôle des réseaux informatiques en séparant la couche de contrôle (qui prend les décisions de routage et de gestion des flux de données) de la couche de données (qui transfère effectivement les données). Dans le contexte des réseaux véhiculaires, SDN peut être utilisé pour améliorer la gestion et la sécurité des communications entre les véhicules et l'infrastructure de transport. Le principe fondamental de SDN est de séparer la logique de contrôle du réseau des équipements de commutation et de routage, et de la centraliser dans un contrôleur SDN, voir la **figure 11** [10].



**Figure 11:** Réseau traditionnel et réseau défini par logiciel SDN.

### 7.2 ARCHITECTURE :

Effectivement, l'architecture SDN (Software-Defined Networking) peut être décomposée en trois couches pour les réseaux véhiculaires. Voici un aperçu de ces trois couches **figure 12** :

#### 7.2.1 COUCHE D'APPLICATION :

Cette couche comprend toutes les applications utilisées dans les réseaux véhiculaires, telles que la gestion de la sécurité, la gestion de la congestion, la signalisation des feux de circulation, les services de divertissement et bien d'autres. Ces applications sont déployées sur les serveurs distants, dans le Cloud ou directement sur les RSUs (Unités de Bord de Réseau) des véhicules.

#### 7.2.2 COUCHE DE CONTROLE :

Cette couche comprend le contrôleur SDN responsable de la gestion des périphériques réseau qui composent le plan de données. Le contrôleur SDN traduit les requêtes des applications SDN en règles de flux qu'il déploie au niveau des équipements du réseau : sélection des chemins de communication, allocation de la bande passante, redirection des paquets, etc. Les contrôleurs doivent également être en mesure de fournir aux applications SDN une vue abstraite du réseau. Pour ce faire, il récupère des informations sur l'état de tous les équipements du réseau (actif ou non, niveau de charge du port, temps de traitement des paquets, etc.) [11].



### 7.2.3 COUCHE DE DONNEES :

Cette couche contient tous les périphériques réseau (routeurs, commutateurs, RSU, véhicules, etc.) responsables du traitement et de la transmission des données ; ces périphériques doivent prendre en charge le protocole Open Flow pour la communication des données avec le contrôleur. Dans des environnements tels que l'environnement véhiculaire, cette couche peut être décomposée en deux sous-couches. La première est la couche supérieure, responsable des connexions filaires. Ensuite, il y a la couche inférieure, qui est responsable de la connexion sans fil (V2V, V2I, etc.) [11] [12].

En résumé, l'architecture SDN en trois couches dans les réseaux véhiculaires vise à offrir une gestion centralisée, flexible et sécurisée des communications entre les véhicules et l'infrastructure de transport, permettant des services plus efficaces, sûrs et fiables pour les conducteurs et les passagers.

### 7.3 LE CONTROLEUR SDN :

Dans l'approche SDN, la prise de décision centralisée est effectuée par un nouvel élément intégré à l'architecture appelé le contrôleur SDN. Ce contrôleur est connecté à tous les périphériques du réseau et est responsable de la gestion de leurs tables de routage.

Le contrôleur SDN (Software-Defined Networking) est un élément central dans l'architecture SDN. Il joue un rôle essentiel dans la gestion et le contrôle du réseau en fournissant une interface de programmation et de contrôle pour configurer, surveiller et gérer les périphériques réseau. En résumé, le contrôleur SDN est un élément central dans l'architecture SDN, offrant une gestion centralisée, une programmation, le contrôle du plan de données et l'orchestration du réseau. Il joue un rôle crucial dans la simplification, la flexibilité et l'automatisation de la gestion des réseaux [28].

### 7.4 LES INTERFACES DE COMMUNICATION :

Les interfaces de communication entre les trois plans (API, Application Programming Interfaces) permettent aux contrôleurs de communiquer avec leur environnement [13].

- ❖ **L'interface sud (Southbound API) :** Cette interface correspond à l'interface entre le contrôleur SDN et les périphériques réseau. Elle représente les interfaces de communication, qui permettent au contrôleur SDN d'interagir avec les équipements de la couche d'infrastructure. Elle supporte différentes tâches : remontée régulière d'informations au contrôleur, notification d'un événement, déploiement de règles de flux, etc. Cette interface normalise les échanges entre la couche de contrôle et la couche de données. Différents protocoles ont été conçus pour cette couche. Le plus répandu d'entre eux est OpenFlow [76].
- ❖ **L'interface Nord (Northbound API) :** Cette interface elle permet la communication et l'échange des données entre le contrôleur SDN sur le plan de contrôle et les applications du réseau. Le type d'informations échangées ainsi que leurs formes et fréquences dépendent de chaque application du réseau. Il n'y a pas de standardisation pour cette interface [77].



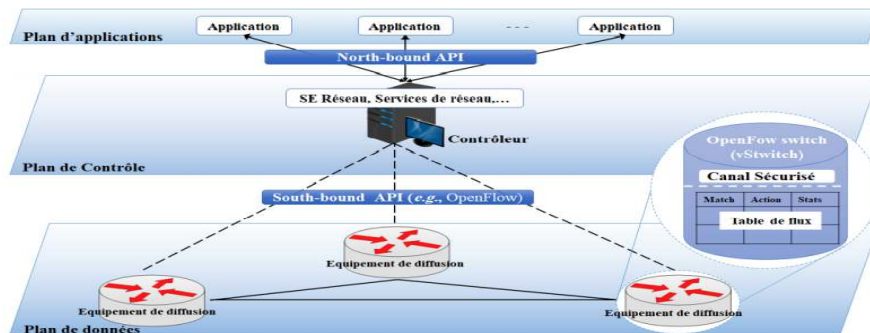


Figure 12: Architecture de SDN/open flow.

## 7.5 OPENFLOW :

Openflow est un protocole de commination open source Il est publié par l'ONF . C'est un protocole est communication qui permet de séparer la partie de contrôle du réseau de la partie de transfert de données dans l'architecture SDN (Software-Defined Networking). Il permet à un contrôleur SDN de prendre des décisions de contrôle de manière centralisée, en fonction des politiques définies, et de les transmettre aux commutateurs réseau qui effectuent le transfert de données [16].

Dans les réseaux véhiculaires, OpenFlow peut être utilisé pour contrôler et gérer le trafic de données entre les véhicules et l'infrastructure de transport. Il permet notamment de :

Définir des politiques de routage et de gestion du trafic pour optimiser la performance du réseau, minimiser la congestion et améliorer la qualité de service (QoS) pour les utilisateurs ;

Gérer les informations de localisation des véhicules pour optimiser le routage et le transfert de données en temps réel ;

Déployer des politiques de sécurité pour protéger les communications entre les véhicules et l'infrastructure de transport, et empêcher les attaques malveillantes ou les violations de données [17].

OpenFlow est donc un outil puissant pour le contrôle et la gestion des réseaux véhiculaires, car il permet une gestion centralisée et flexible des politiques de réseau, qui peuvent être adaptées en temps réel pour répondre aux besoins changeants des utilisateurs [16].

## 7.6 APPORTS DE SDN POUR LES RESEAUX VEHICULAIRE :

SDN (Software Defined Networking) peut apporter plusieurs avantages aux réseaux véhiculaires (VANETs - Vehicular Ad hoc Networks) en permettant une gestion plus efficace du trafic de données et une amélioration de la sécurité des communications. Voici quelques-uns des avantages spécifiques que SDN peut apporter aux réseaux véhiculaires :

❖ **Gestion centralisée du réseau :** Avec SDN, la gestion du réseau peut être centralisée et déléguée à un contrôleur de réseau. Cela permet de surveiller et de contrôler efficacement le trafic de données dans le réseau de véhicules, ce qui peut aider à optimiser les performances du

réseau[18]. De plus, la vue globale du contrôleur SDN sur tout l'état du réseau offre une gestion flexible en temps réel des ressources du réseau ce qui permet l'adaptation dynamique et rapide aux changements fréquents des conditions de réseau .

❖ **Faciliter le passage à l'échelle et réduction les coûts** : En utilisant des équipements de réseau compatibles SDN, les coûts d'exploitation et de maintenance du réseau de véhicules peuvent être réduits, car la gestion centralisée du réseau simplifie le déploiement et la maintenance du réseau et la proposition de nouvelles fonctionnalités à une grande échelle avec un coût minimal. Cela, en intervenant tout simplement au niveau du contrôleur SDN au lieu d'agir au niveau de chaque véhicule séparément. Par exemple, le contrôleur SDN peut facilement gérer une mise à jour logiciel à grande échelle en installant simplement des entrées dans les tables de flux des véhicules à distance [19] [20].

❖ **Accélérer l'innovation et introduire de nouveaux Qualité des services plus avancés** : SDN permet une meilleure gestion de la qualité de service en permettant aux contrôleurs de réseau de définir des politiques de priorité pour le trafic. Cela peut aider à garantir une expérience utilisateur optimale pour les applications de véhicule à véhicule et de véhicule à infrastructure, avec l'aide de la programmabilité et les interfaces de communication unifiées et reconfigurables de SDN, les équipements du réseau deviennent ouverts et configurables avec des logiciels indépendants des fournisseurs. Ce qui va encourager l'innovation et accélérer le développement de nouvelles solutions plus évoluées [21] [22].

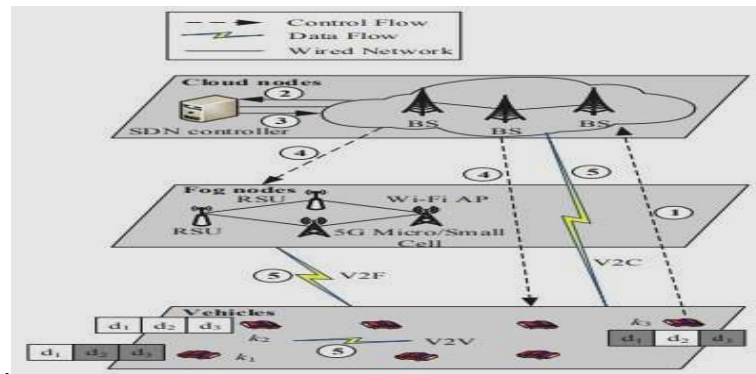
❖ **Flexibilité et Sécurité** : SDN permet une configuration flexible du réseau. Les règles de gestion du trafic peuvent être mises à jour rapidement et facilement, Les réseaux véhiculaires sont vulnérables aux attaques malveillantes, ce qui peut mettre en danger la sécurité des passagers et des véhicules. SDN peut aider à renforcer la sécurité en permettant une segmentation claire du réseau et en offrant des mécanismes de contrôle d'accès granulaires, permettant ainsi une adaptation rapide aux changements dans l'environnement de réseau [17].

En somme, SDN peut apporter des améliorations significatives aux réseaux véhiculaires, en offrant une meilleure gestion du trafic de données, une meilleure sécurité et une meilleure qualité de service, tout en réduisant les coûts d'exploitation.

## 8. LES TRAVAUX CONNEXES :

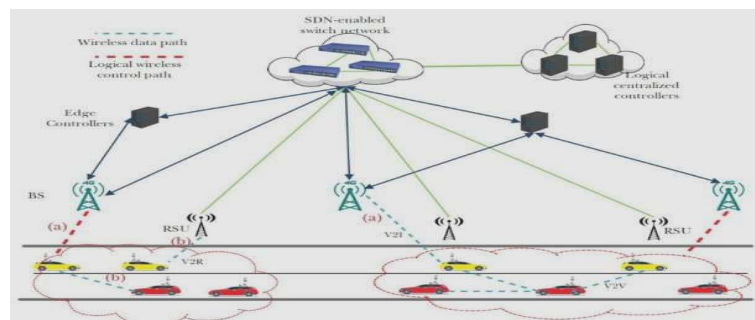
Dans cette partie, nous avons présenté quelques travaux connexes dans le domaine des réseaux de véhicules basés sur le SDN dans le cadre de la dissémination des données.

Liu et d'autres dans cette référence [61], a proposés une architecture de dissémination des données basée sur la transformation des éléments dans des VANETs hétérogènes SDN. Comme le montre la figure 25, le système se compose de trois niveaux, Au niveau supérieur, les nœuds de Cloud tels que les BSS cellulaires sont connectés au contrôleur SDN via le réseau central. Les nœuds Cloud peuvent communiquer avec les véhicules via des interfaces cellulaires telles que "Long-Term Evolution-Véhicule (LTE-V)" et C-V2X. Par souci de simplicité, nous appelons cette communication Véhicule-to-Cloud (V2C), voir la **figure 25**.



**Figure 13:** System architecture.

Abbas et d'autres dans cette référence [62], ils ont introduit une stratégie de routage tenant compte de l'itinéraire : un protocole de routage optimisé pour les performances spécialement conçu pour les réseaux véhiculaires basés sur l'infrastructure. Où les routes sont divisées en segments routiers, avec des unités en bordure de route pour les communications à sauts multiples. La propriété unique du protocole proposé est qu'il analyse le réseau cellulaire pour transmettre des données en transmettant des messages de contrôle à faible latence vers le contrôleur. Le concept de contrôleur de périphérie a été introduit en tant qu'épine dorsale opérationnelle du réseau de véhicules dans l'Internet des véhicules pour obtenir une topologie de véhicule en temps réel, voir la **figure 26**.

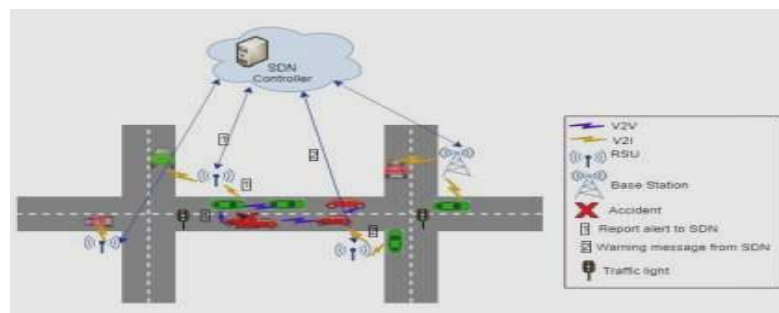


**Figure 14 :** L'architecture proposée pour SD-IOV.

Deyun Gao et autres Dans la référence [63], ils ont proposés une messagerie d'urgence hybride (en couches) qui introduit une technologie de réseau durable dans l'environnement du réseau du véhicule et tire parti de la flexibilité de la communication de véhicule à véhicule, Ils ont mis en place un contrôleur central prenant en charge les commutateurs SDN et Ancre et ont discuté du processus de messagerie comme suit : (1) Lorsqu'un événement se produit, le véhicule source envoie le premier message de détresse à la clé d'amarrage la plus proche. (2) Ce commutateur RSU envoie le paquet correspondant dans le message au contrôleur. (3) Le contrôleur détermine la région de diffusion cible pour le message d'alarme, trouve des cibles RSU, puis calcule les chemins de routage en exécutant d'abord des algorithmes de routage tels que le chemin le plus court. (4) Le contrôleur envoie ensuite des messages de mode de flux aux commutateurs RSU appropriés pour modifier les tables de flux et transmet les messages d'alarme aux cibles RSU. (5) Les RSU concernés interviennent selon leurs propres horaires : envoyer des messages d'urgence aux RSU voisins, ou transmettre directement aux véhicules à proximité, ou transmettre des messages

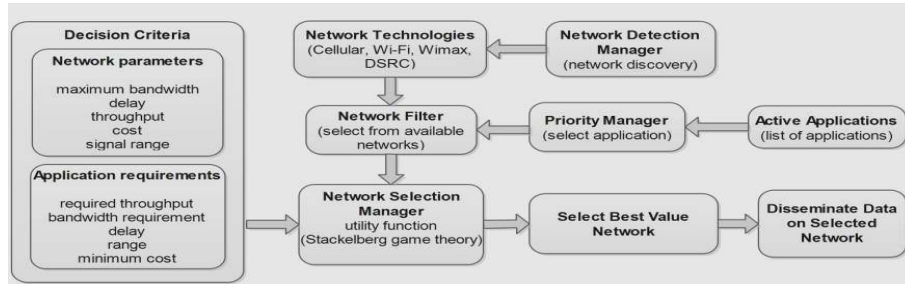
modifiés aux véhicules à proximité, retransmettre des messages à d'autres véhicules en fonction de ceux reçus messages d'avertissement. (7) Les transmissions consécutives de messages d'alarme avec le même ID d'événement n'affectent pas le contrôleur.

Santosh et d'autres dans cette référence [64], ils ont conçu le système SD-IoV pour gérer le trafic intense et éviter le problème de tempête de transmission dû à la grande vitesse de livraison des paquets, Le mécanisme de routage de diffusion proposé utilise le transfert sélectif et la détection de véhicule voisin pour envoyer efficacement des alertes, évitant ainsi la congestion et réduisant les temps de trajet. L'unité embarquée (OBU) dans les véhicules détecte l'accident et démarre l'algorithme de transmission dans le système SD-IoV. Les véhicules équipés d'OBU détectent un accident lorsqu'il y a un changement soudain d'accélération et de position du gyroscope. Le véhicule transmet des informations aux autres véhicules et aux ordures ménagères à proximité via une communication V2V ou V2I. Le RSU envoie des informations au contrôleur SDN et identifie les tronçons routiers qui nécessitent des alertes. De plus, la section de route affectée (lieu de l'accident) est temporairement supprimée de la carte de l'itinéraire. L'alerte est propagée aux RSU appropriées via des règles installées sur les commutateurs. La connexion entre le RSU est connue du SDN et la règle contient un champ de correspondance et un champ d'action. Si l'ID RSU correspond, le champ d'action spécifie comment des paquets sont transmis pour atteindre le RSU cible. Le RSU reçu envoie un message aux véhicules sur le tronçon de route à intervalles réguliers, voir la **figure 27**.



**Figure 15:** Architecture du Système SD-IoV pour la Dissémination des Messages d'Alerte.

Zhu et d'autres dans cette référence [65], ils ont introduit une nouvelle architecture visant à la sélection de réseaux SDN et à la distribution de données dans un environnement de réseau composite hétérogène. Le schéma proposé comprend deux étapes : la sélection du réseau et le flux de données, comme illustré à la **figure 28**. L'étape de sélection du réseau est effectuée en fonction des exigences de l'application et des paramètres du réseau. L'approche de la distribution des données dans un réseau composite dédié est déterminée par le programme. Le contrôleur est responsable de la transmission et de la mise à jour des informations globales pour l'ensemble de la topologie du réseau, assurant un chemin stable de la source à la destination.



**Figure 16:** opération de la sélection du réseau.

Santosh et d'autres dans la référence [66], ils ont mise en œuvre d'une stratégie de dissémination des données utilisant une technologie d'équilibrage de charge Internet embarquée sur réseau définie par logiciel et une latence réduite et une activité Internet embarquée sur les appareils Cloud-Edge (computing) de périphérie à l'aide d'un algorithme d'optimisation Whale intégré. Un appareil utilisateur générant un trafic important entraînera une latence accrue sur le réseau Cloud. Les réseaux définis par logiciel sont le moyen le plus prometteur de fournir un contrôle centralisé et une connaissance globale des réseaux Cloud/brouillard.

**Tableau 2:** comparaison les travaux connexes

Réf	Proposition	Objectif	La technique utilisée	Simulateur	Environment
[61]	Un architecture dissémination de données basée sur son avec Fog computing pour VANET	Cryptage du réseau et de la mise en mémoire tampon des véhicules et augmentez l'efficacité de la bande passante.	Algorithme CSS (Clique Searching based Scheduling) au niveau du contrôleur SDN.	/	VANET
[62]	Approche de sensibilisation à la route basée sur le SD-IoV.	Analyse le réseau cellulaire pour transmettre des données en transmettant des messages de contrôle à faible latence vers et depuis le contrôleur.	Protocole de routage avec Edge.	Mininet-wifi SUMO	IOV
[63]	Un nouvel outil de protocole de routage sensible à la route pour l'IOV basé sur SDN.	Pour profiter de la flexibilité de la communication entre les véhicules et de la rapidité d'arrivée des messages d'urgence.	Algorithme de transmission hybride de messages d'urgence (HEMT).	OMNET++ SUMO	IOV

[64]	Un système SD- IoV SDN.	Ce système a réduit les embouteillages et le temps de trajet.	Algorithme De Détection Des Accidents.	OMNET++ SUMO	IOV
[65]	Une approche de dissémination des données sur le réseau sélectionné.	Optimization of package delivery ratio, average end-to- end delay, average throughput and routing overhead.	Network sélection algorithme.	NS-3 SUMO	VANET
[66]	Une stratégie de dissémination des données utilisant une technologie d'équilibrage de charge Internet.	Minimisation de la latence et amélioration de la Qualité de service (QoS).	Whale Optimization Algorithm.	MATLAB	IOV

## 9. CONCLUSION :

Dans ce chapitre, nous avons présenté les concepts de base des réseaux véhiculaires. Les réseaux véhiculaires sont considérés comme une excellente plateforme pour l'évolution des systèmes de transport intelligents, puis nous avons décrit un bref aperçu de leur évolution, d'abord vers des réseaux hétérogènes et plus récemment vers l'Internet des Véhicules. Enfin, nous avons donné un bref aperçu du paradigme SDN et de leur architecture dans les réseaux véhiculaires. En combinant les réseaux véhiculaires et le SDN, il est possible de créer des infrastructures de transport intelligentes et connectées, capables de fournir des services avancés aux conducteurs, aux passagers et aux piétons. Ces technologies ouvrent la voie à de nouvelles applications et améliorent la sécurité et l'efficacité des systèmes de transport .

10.

## Chapitre

# 2

# *Revue de Littérature sur SDN dans Les Réseaux IOV*

---

## 1. INTRODUCTION :

Les réseaux définis par logiciel basé sur l'internet des véhicules représentent un développement significatif dans le domaine des transports intelligents. Le programme vise à connecter les véhicules et à les intégrer à Internet, offrant de nouvelles possibilités en termes de sécurité, d'efficacité et de services pour les conducteurs et les passagers. Le concept de réseau de développement durable est au cœur de cette transformation. Un réseau de développement durable est une méthode qui sépare le niveau de contrôle du niveau de données du réseau. Dans le contexte de l'IoV, cela signifie que la gestion du réseau et les flux de données sont centralisés et programmables. Grâce au SDN, les décisions de routage et de gestion du trafic peuvent être prises de manière dynamique, en fonction des conditions de conduite, des besoins des utilisateurs et des exigences du réseau.

Dans ce chapitre, nous allons présenter quelques études menées sur le SDN basées sur les réseaux véhiculaires de différentes définitions, architectures, certains types de contrôleurs SDN et diverses méthodes de dissémination des données sur les réseaux véhiculaires.

## 2. DEFINITIONS SDN :

Dans cette partie, nous avons présenté quelques définitions du réseau défini par logiciel (SDN).

(Software-Defined Networking) est une technologie de réseau qui permet de centraliser la gestion et le contrôle des réseaux informatiques en séparant la couche de contrôle (qui prend les décisions de routage et de gestion des flux de données) de la couche de données (qui transfère effectivement les données) [10].

SDN est une fonctionnalité évolutive qui différencie le plan réglementaire de l'Avancement des Connaissances dans le plan de changements et rassemble tous les plans de contrôle chez l'opérateur principal [39].

L'idée clé derrière SDN est de détacher le plan de contrôle du plan de données et d'offrir ainsi une vue de contrôle hautement flexible, automatisée et centralisée de l'architecture du réseau [40].

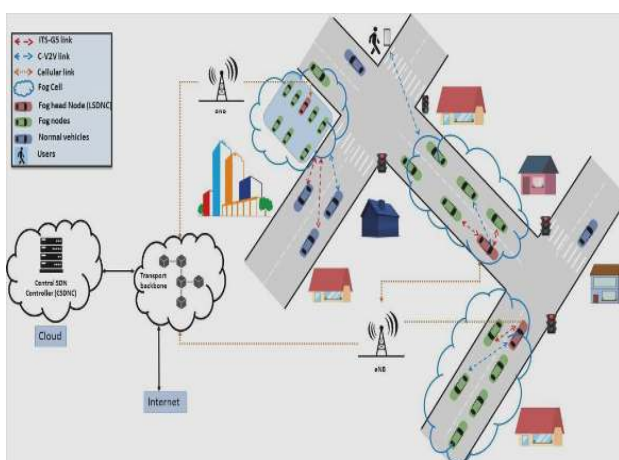
La mise en réseau définie par logiciel (SDN), se caractérise par la séparation du plan de contrôle et du plan de données. Le contrôleur, une entité logiquement centralisée, est utilisé au niveau du plan de contrôle pour surveiller et contrôler les ressources du réseau [41].



### 3. ARCHITECTURES SDN-IOV :

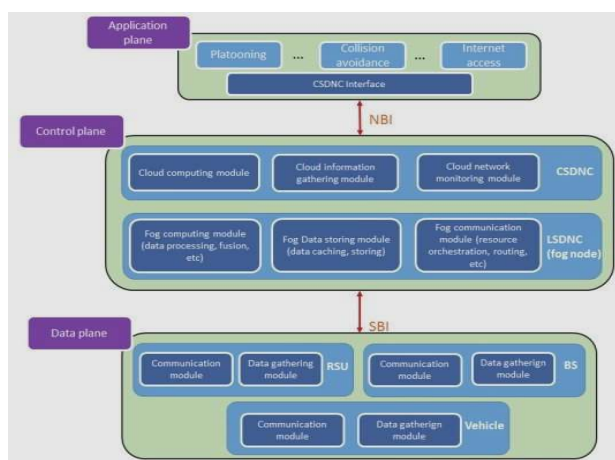
Dans cette partie, nous avons présenté quelques-unes des différentes architectures dans le domaine des réseaux véhiculaires basés sur SDN.

Samir Tohme et d'autres dans cette référence [42], ils ont proposé une approche pour consiste à combiner une architecture de réseau défini par logiciel SDN en couches avec une architecture de calcul de brouillard basée sur des véhicules pour créer un modèle d'infrastructure appelé VISAGE pour les futurs systèmes VANET 5G (réseaux véhiculaires ad hoc basés sur la technologie 5G), voir les **figure 13,14**.



**Figure 17** : Architecture des réseaux SDN.

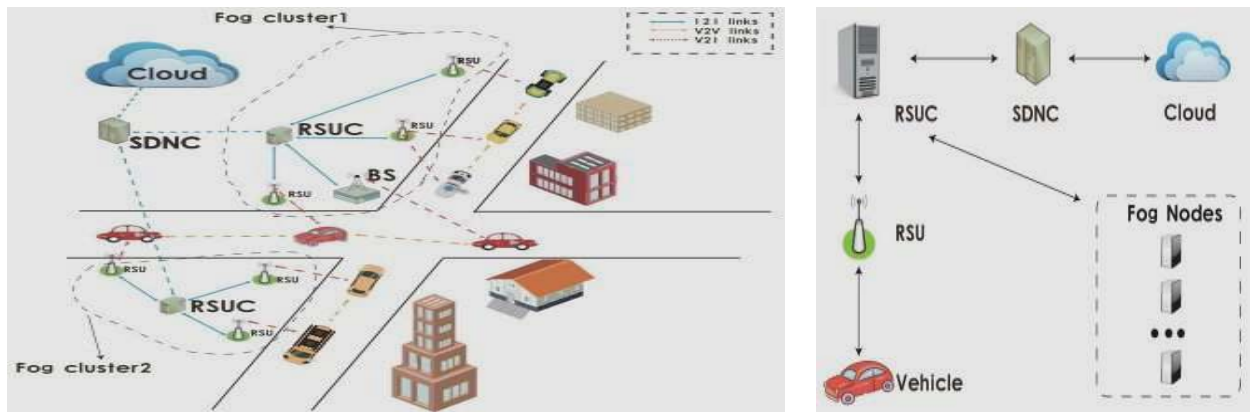
Multiniveaux 5G basés sur la Fog.



**Figure 18** : Architecture de logiciel.

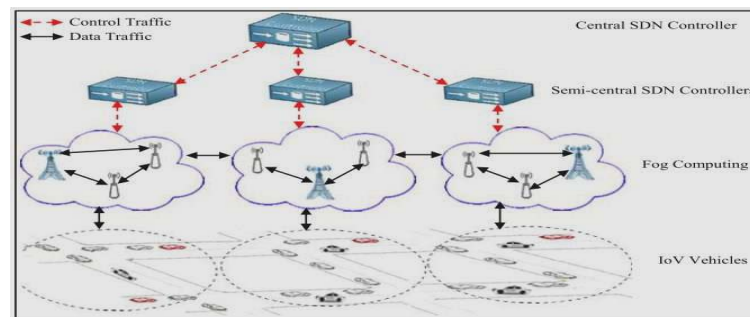
Gao et d'autres dans cette référence [43], ils ont à proposer nouvelle architecture 5G-IoV, Basé sur le Cloud computing flou et les réseaux définis par logiciel (SDN), en plus de prendre pleinement en compte les exigences de service IOV, un modèle à quatre objectifs a été construit et un algorithme d'optimisation multicritères a été proposé. Et révisé l'allocation des ressources des tâches transférées au système de brouillard nuageux. Conformément aux exigences du service IOV, nous créons un modèle qui prend en compte quatre indicateurs : la latence du service, la stabilité de l'exécution des tâches, la consommation d'énergie et l'équilibrage de charge. Comme le montre la **Figure 15**.





**Figure 19:** Architecture de l'IOV 5G basée sur Fog-Cloud et SDN. **Figure 20:** Allocation des ressources

Ahmad Jawid et coll ont à proposées une architecture hiérarchique basée sur les réseaux définis par logiciel (SDN) et le traitement des aliments pour les réseaux IoV. Cette architecture se compose de véhicules bas, d'un environnement de traitement du brouillard, de contrôleurs SDN semi-cœur et de couches principales de contrôleurs SDN. De plus, une stratégie de routage appelée Routage efficace en délai basée sur SDN et Fog Computing pour IoV (DRSFI) a été proposée, Les contrôleurs SDN exécutent DRSFI pour calculer les itinéraires avec un délai minimal, en tenant compte des contraintes de bande passante disponibles, de l'emplacement et de la vitesse du véhicule [44], voir la **figure 17**.



**Figure 21:** Architecture hiérarchique basée sur les SDN et les réseaux IoV.

Khoury et d'autres dans cette référence [45], ils ont proposé un protocole de routage robuste pour la transmission de paquets de données qui répond à l'évolution de la topologie et de la mobilité du réseau en adoptant une architecture de réseau hybride définie par logiciel. Pour ce faire, il sépare le plan de contrôle, qui décide de l'endroit où les données sont envoyées, du plan de données, qui achemine les données vers la destination choisie, Les décisions de contrôle sont basées sur différents critères, à savoir la durée du contact, la charge de fluage et les erreurs de communication, qui sont prises de manière décentralisée par chaque véhicule, voir la **figure 18**.

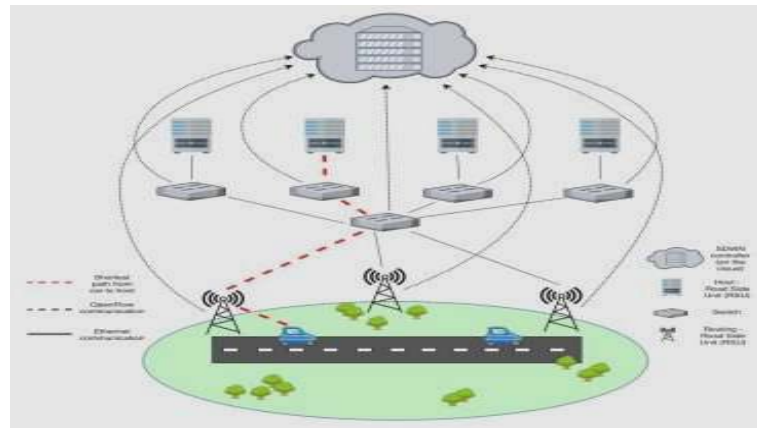


Figure 22: Architecture Générale.

Carlos Renato et d'autres ont proposé un écosystème 5G V2X. L'idée de SDN sert de base à l'écosystème SDN proposé. Étant donné que les principaux lieux de divertissement sont dans les véhicules, l'infrastructure du réseau doit être suffisamment grande pour assurer la livraison et la qualité. Pour ce faire, ils évaluent la communication de véhicule à véhicule (V2V) dans les zones urbaines et rurales ainsi que le trafic sur les services de vidéo sur Internet [46], voir la **figure 19**.

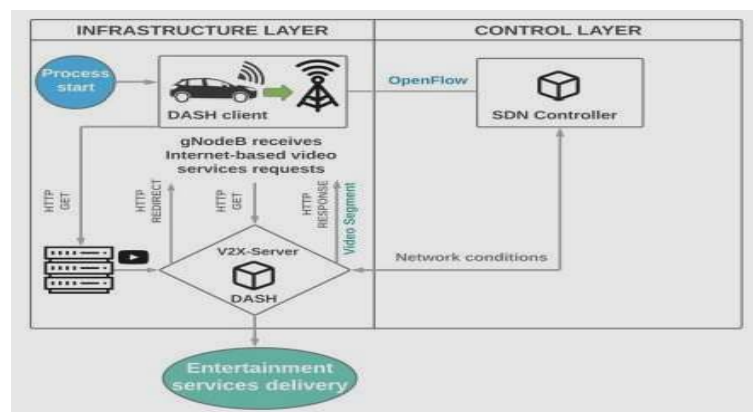
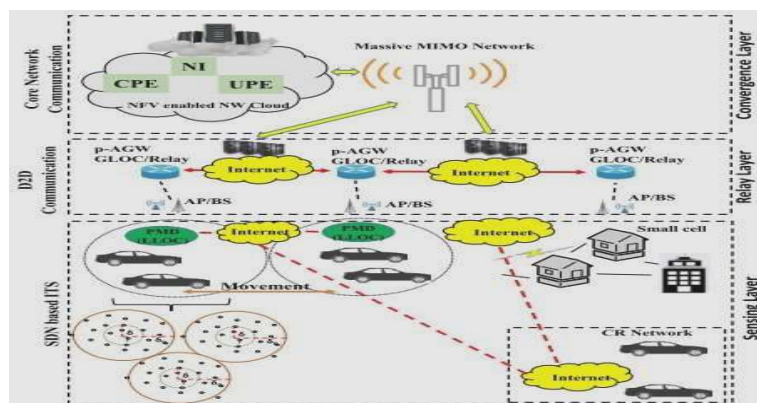


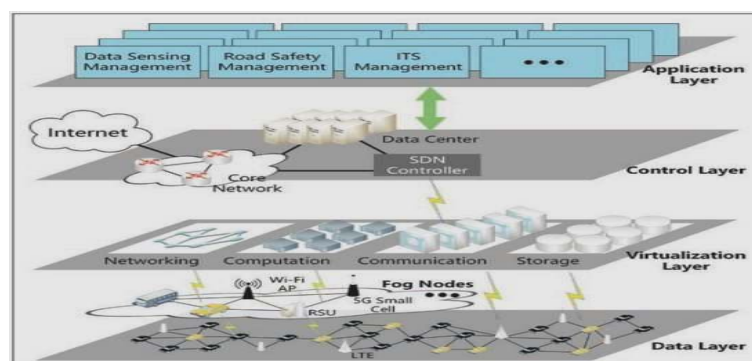
Figure 23: Fonctionnement de l'architecture proposée.

Sadia Din et d'autres, [47] ils ont présenté un nouveau concept pour améliorer ses capacités grâce à une nouvelle architecture SDN de cinquième génération (SDN) pour les systèmes de transport intelligents (5G). L'architecture système proposée repose sur trois couches fonctionnelles : la couche capteur, la couche relais et la couche réseau cœur. De plus, l'architecture 5G ils ont proposées offre des débits et des débits de données élevés, voir la **figure 20**.



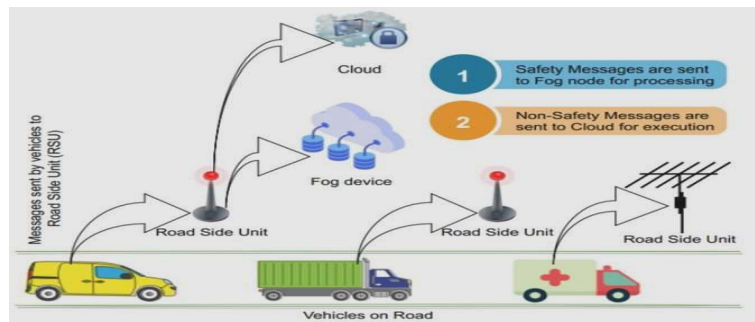
**Figure 24:** Architecture de réseau 5G proposée.

Kai Liu et autres. [48], ils ont proposé une nouvelle architecture hiérarchique pour accroître l'évolutivité et la fiabilité des services d'information, améliorer l'agilité et la flexibilité de la gestion des applications et jeter des bases solides pour les futurs systèmes STI. Comme le montre la **figure 21**, l'architecture se compose de quatre couches, à savoir la couche d'application, la couche de contrôle, la couche de virtualisation et la couche de données. En général, l'architecture hiérarchique est conçue pour intégrer les modèles SDN et Fog et les utiliser au mieux pour réaliser une synergie dans les services d'information IOV.



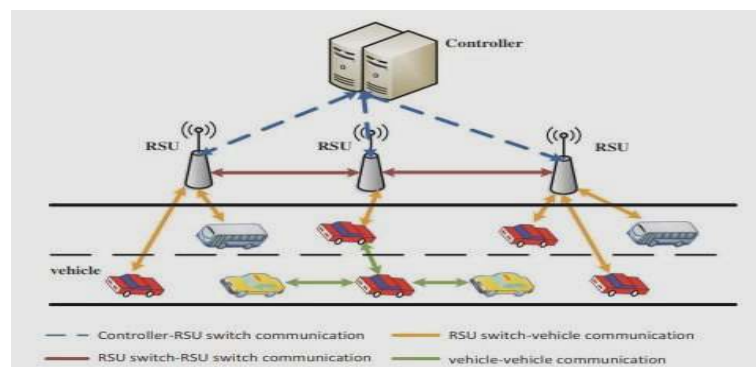
**Figure 25:** Architecture hiérarchique pour les IoV.

Syed et d'autres dans cette référence [49], ils ont proposé une architecture sur le SDVN conscient de la qualité de service et basé sur la tolérance aux pannes utilisant le Cloud-Fog computing (QAFT-SDVN). Le modèle proposé transmet les messages du véhicule via des nœuds SDN situés sur des nœuds Fog, de sorte que les contrôleurs SDN reçoivent les messages des modules SDN voisins et les hiérarchisent de deux manières différentes. L'un dépend du type de lettre, l'autre de la date et de la longueur de la lettre, voir la **figure 22**.



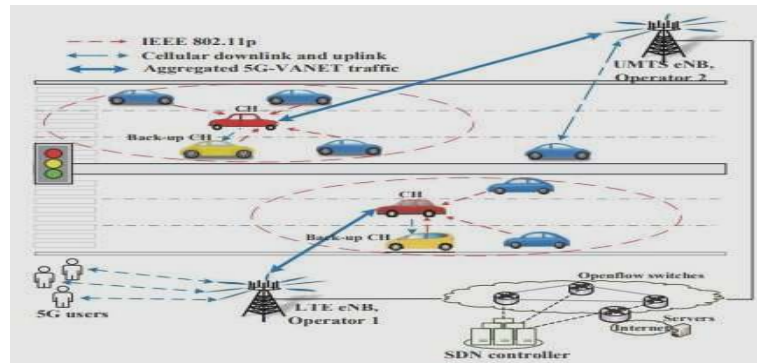
**Figure 26:** Architecture de SDVN.

Voulant Zhu et autres dans cette référence [50], ils ont proposé principes SDN appliqués dans un scénario de réseau de véhicule urbaines pour obtenir une transmission rapide et fiable des messages d'urgence, La nouvelle architecture hybride, illustrée à la **figure 23**, se compose de trois entités : un contrôleur central, un commutateur RSU hybride et une unité embarquée (OBU). Dans cette architecture, le contrôleur et l'alimentation peuvent prendre en charge le protocole Open Flow, tandis que le RSU et le véhicule peuvent prendre en charge les protocoles de communication destinés aux réseaux automobiles.



**Figure 27:** Architecture de l'HÉMT.

Xiaoyu Duan et coll dans cette référence [51] , ils ont présentés la technologie de virtualisation SDN, dans l'architecture VANET 5G, fournissant des politiques programmables de haut niveau pour des communications automobiles personnalisées , Les ressources du téléphone portable sont également économisées en utilisant la connexion IEEE 802.11p entre les véhicules, Grâce à la méthode innovante de regroupement des véhicules avec prise en charge du RNIS et au schéma d'un double chef de cluster, les coûts de signalisation sont réduits et la qualité de la communication est améliorée , La conception proposée offre également aux utilisateurs de véhicules un accès continu aux services de l'opérateur via des CANAUX de clustering et de sauvegarde adaptative , voir la **figure 24**.



**Figure 28:** Architecture de réseau intégrée 5G compatible SDN-VANET.

**Tableau 3:** des architectures SDN.

Réf	Architecture Proposée	Technologie	Contrôleur SDN	Environnement
[42]	Visage SDN pour les futurs systèmes 5G-fanit.	SDN Fog et Cloud	Centralisé	VANET
[43]	Une nouvelle architecture IoV 5G basée sur le calcul de brouillard et le SDN.	SDN Fog et Cloud	Centralisé	IOV
[44]	Une architecture hiérarchique basée sur SDN avec Algorithme de routage (DRFSI).	SDN-Fog	Centralisé et Semi-centralisé	IOV
[45]	Une architecture SDN dans un environnement de calcul de brouillard véhiculaire.	SDN-Cloud	Centralisé et Semi-centralisé	IOV
[46]	Un écosystème 5G V2X basée sur SDN.	SDN	Centralisé	IOV
[47]	SDN basée sur STI Dans l'IOV.	SDN-STI	Centralisé	IOV
[48]	Une nouvelle architecture hiérarchique.	SDN Fog-STI	Centralisé	IOV

[49]	Une architecture SDVN basé sur la tolérance aux pannes utilisant le Cloud-Fog computing (QAFT-SDVN).	SDN Fog et Cloud	Centralisé	IOV
[50]	Une architecture SDN appliqués dans un scénario de réseau de véhicule urbaines des messages d'urgence.	SDN	Centralisé	IOV
[51]	Une l'architecture SDN basée sur technologie de virtualisation dans 5G-VANET.	SDN Clustering	Centralisé	VANET

#### 4. LES TYPES DE CONTROLEURS SDN :

**Beacom** [52] : Beacom est un contrôleur SDN introduit en 2010. Il a été utilisé dans plusieurs études scientifiques. C'est un contrôleur basé sur JAVA. Il peut fonctionner sur plusieurs plateformes, y compris Linux et Android multicœurs haut de gamme.

**DISCO** [53] : DISCO est un contrôleur distribué. Il est principalement utilisé dans les réseaux WAN et overlay. Chaque contrôleur est responsable d'un domaine réseau. Les contrôleurs communiquent entre eux via un canal inter-contrôleur. DISCO peut s'adapter dynamiquement à différentes topologies de réseau hétérogènes.

**IRIS** [54] : IRIS est une plate-forme de contrôleur SDN basée sur OpenFlow. L'informatique peut gérer un grand réseau. IRIS prend en charge les architectes évolutifs horizontalement. Cela permet d'ajouter dynamiquement des serveurs à un cluster de contrôleurs. Cela augmente le facteur d'efficacité des commandes de l'aéronef.

**Maestro** [55] : Maestro est le premier système de contrôle OpenFlow à utiliser le parallélisme. Dans Maestro, les développeurs peuvent modifier les fonctionnalités du plan de données en écrivant de simples programmes à thread unique. Maestro a ses propres projets et techniques pour prendre en charge le protocole OpenFlow. Il s'agit d'un contrôleur basé sur JAVA et très portable pour différents systèmes d'exploitation et architectures.

**OpenDaylight** [56]: OpenDaylight Est inspire par Beacon. Il s'agit d'un contrôleur JAVA dérivé de balises. Il prend en charge OpenFlow et d'autres API orientées vers le Sud. OpenDaylight est présent sur votre machine virtuelle JAVA (JVM).

**NOX** [57] : NOX est la première plate-forme SDN de contrôleur OpenFlow pour la création d'applications de contrôle de réseau. Il a été initialement développé par Nicira Networks, aux côtés d'OpenFlow. Plus tard, NOX a été donné à la communauté SDN. Les applications peuvent être en python ou C++ et peuvent être chargées dynamiquement.



**POX** [58] : POX est similaire au contrôleur de NOX. POX est un contrôleur SDN qui permet le développement et le prototypage rapides du réseau. Il suit le protocole OpenFlow, et qui sert de cadre entre les commutateurs OpenFlow.

**Floodlight** [59] : Floodlight est un contrôleur SDN open source. Il s'agit d'un contrôleur Open Flow de classe entreprise, basé sur JAVA. Il fonctionne à la fois avec des commutateurs physiques et des commutateurs virtuels qui utilisent le protocole OpenFlow.

**Ryu** [60] : Ryu est un contrôleur SDN basé sur des composants. Ryu signifie "flux" en japonais. Ryu fournit de nombreux composants logiciels et API étendus pour créer et gérer des applications réseau. Ryu prend en charge des protocoles tels qu'OpenFlow, Netconf, OF-config, etc. Ryu est entièrement implémenté en langage python. Il est sous licence Apache 2.0.

**Tableau 4:** Comparaison entre les propriétés des contrôleurs SDN.

Propriétés	Contrôleurs SDN								
	Beacon	DISCO	IRIS	Maestro	OpenDaylight	NOX	POX	Floodlight	Ryu
Support du Parallélisme	Oui	Oui	Oui	Oui	Oui	Non	Non	Oui	Oui
Protocole	Open flow	Open flow	Open flow	Open flow	Open flow	Open flow	Open flow	Open flow	Open flow
L'architecteur	Centralisé	Distribuée	Centralisé	Centralisé	Distribuée	Centralisé	Centralisé	Centralisé	Centralisé
Langage de programmation	JAVA	JAVA	JAVA	JAVA	JAVA	C++	Python	JAVA	Python
API supportés	Ad-hoc API	REST API	REST API	Ad-hoc API	REST API	Ad-hoc API	Ad-hoc API	REST API	Ad-hoc API
Plateformes Supportées	Linux Windows	Linux Windows	Linux Windows	Linux Windows MacOs	Linux	Linux Windows MacOs	Linux Windows MacOs	Linux Windows MacOs	Linux

## 6. CONCLUSION :

Dans ce chapitre, nous avons effectué une analyse détaillée des recherches antérieures sur le SDN dans les réseaux véhiculaires utilisant la technologie 5G embarquée et examiné les recherches antérieures pour comprendre comment ces réseaux sont définis et classés sous différents angles. Cette étude a été réalisée en examinant les différentes définitions de ces réseaux et les différentes architectures utilisées, ainsi que les méthodes de dissémination des données qui leur sont liées, en mentionnant certains des contrôleurs utilisés dans SDN. Ces études sont

---

résumées dans des tableaux pour illustrer la diversité actuelle des concepts, cadres et techniques utilisés pour le Réseau du développement durable.

Sur la base des études mentionnées, on peut conclure que le rôle décisif dans la création de ce type de réseau appartient à la technologie 5G, qui garantit une vitesse de connexion élevée, une faible latence et la possibilité de connecter plusieurs appareils en même temps. Dans le même temps, l'intégration avec les réseaux SDN a considérablement amélioré le taux de livraison des paquets (PDR), réduit le trafic et évité les accidents.



# Chapitre

## 3

# *Implémentation SDN in IOV*

## 1. INTRODUCTION :

Toute nouvelle solution passe par un processus d'évaluation et de validation avant de pouvoir être potentiellement déployée. Le moyen idéal pour accomplir cette tâche est de pouvoir effectuer des tests dans un environnement réel. Cependant, en raison de l'environnement distribué et complexe et de la structure topologique des réseaux véhiculaires et Pour contourner ce problème, la simulation est actuellement le moyen le plus utilisé. En fait, il est plus facile et moins coûteux de concevoir d'analyser et d'évaluer les performances de n'importe quelle solution par simulation [67].

Dans ce chapitre, nous allons proposer une approche de la dissémination des données basée sur SDN pour l'internet des véhicules. Nous présenterons en détail les différentes composantes et la structure proposée de ce type de réseau. Nous procéderons ensuite à la création d'une topologie de réseau spécifique, suivie d'une simulation approfondie du réseau. Nous évaluerons ensuite les performances du réseau, et enfin nous discuterons des résultats obtenus et les comparerons avec le réseau véhiculaire traditionnel.

## 2. L'APPROCHE PROPOSEE :

Nous avons proposé une approche hiérarchique pour la dissémination des données basées sur SDN pour l'Internet des véhicules. Cette approche vise à améliorer l'efficacité et la flexibilité de la transfert des données au sein du réseau. Elle facilite la coordination entre les différentes couches du réseau et permet une utilisation optimale des ressources disponibles. L'approche se compose de trois couches interconnectées par des interfaces Sud-Nord. La première couche, la couche de données, gère l'échange de données entre les véhicules du réseau. La deuxième couche, la couche de contrôle, gère les opérations et les fonctionnalités du réseau en utilise deux contrôleurs POX pour éviter les panes dans le réseau. Enfin, la troisième couche, la couche d'application, héberge les applications spécifiques qui utilisent les fonctionnalités du réseau, voir la **figure 29**.

### 1. La couche de données :

Cette couche comprend tous les éléments physiques du réseau de véhicules, tels que les véhicules équipés de dispositifs de communication DSRC et les unités routières (RSU) installées le long des infrastructures routières. Les véhicules envoient et reçoivent des messages centralisés par un module interne (OBU) basée sur IEEE 802.11p pour communiquer directement avec les autres véhicules à proximité (V2V) et avec les modules routières(V2I) par des communications LTE/4G.

## 2. La couche de contrôle :

Cette couche est composée de stations de base qui ont un rôle crucial dans la communication sans fil avec les véhicules. Elles assurent le relais des données lorsqu'elles sont connectées aux unités de contrôle POX1 et POX2 pour éviter les panes dans le réseau. Ces dernières jouent un rôle central dans la gestion du réseau de véhicules en utilisant le protocole OpenFlow. Elles collectent des informations sur le réseau, prennent des décisions en matière de routage, de gestion des ressources et de sécurité, et permettent une programmation personnalisée du comportement du réseau.

## 3. La couche d'application :

Cette couche fait référence à la couche qui héberge les applications et services spécifiques qui utilisent les réseaux véhiculaires. Cette couche est située au-dessus de la couche de contrôle et exploite les fonctions fournies par cette dernière pour fournir des services aux utilisateurs (contrôle de congestion, gestion des ressources, gestion de la navigation).

La liaison (**Southbound**) représente l'interface de communication entre la couche de contrôle et la couche d'infrastructure, permettant au contrôleur de programmer les équipements réseau. La liaison (**Northbound**) représente l'interface de communication entre la couche de contrôle et la couche d'application, permettant aux applications et aux services d'interagir avec le contrôleur et de recevoir des informations sur l'état du réseau.

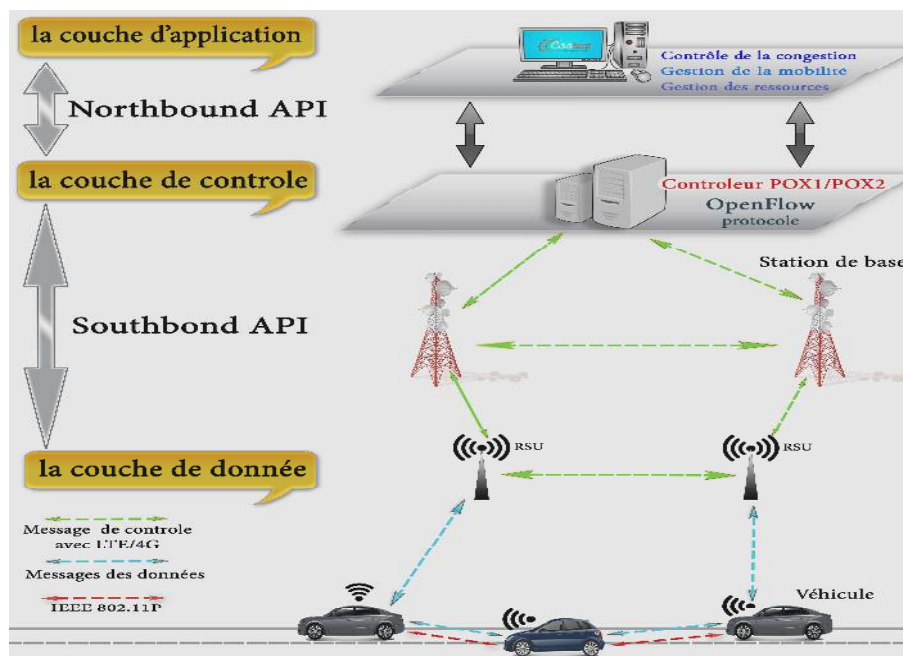


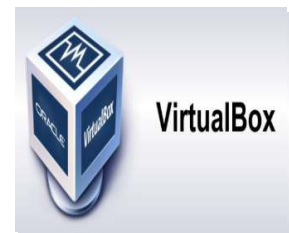
Figure 29: L'approche proposée.

### 3. ENVIRONNEMENTS DE TRAVAIL :

Pour implémenter le scénario de dissémination de données sur le réseau SDN dans les réseaux véhiculaires, nous avons utilisé le simulateur Mininet-wifi sur le système d'exploitation Ubuntu et le simulateur de trafic routier SUMO, et le contrôleur POX avec langage de programmation Python.

#### 3.1 VIRTUELLE BOX :

Virtual Box est un produit de virtualisation x86 et AMD 64/Intel 64 pour un usage professionnel et domestique. La Virtual Box n'est pas seulement un produit riche en fonctionnalités et hautes performances pour les entreprises clientes, mais également une solution professionnelle disponible gratuitement en tant que logiciel open source sous les termes de la Licence Publique Générale GNU (GPL) [68].



#### 3.2 UBUNTU :

Système d'exploitation GNU/LINUX basé sur la distribution Debian Linux. Il est utilisé par des millions de PC dans le monde avec une interface simple, intuitive et sécurisée. Selon le site Web d'Alexa, il s'agit de la version la plus consultée sur Internet et du système d'exploitation le plus utilisé sur les systèmes Cloud et les serveurs informatiques, avec des versions stables majeures appelées LTS [69].



#### 3.3 MININET-WIFI :

Est un outil d'émulation de scénario sans fil OpenFlow/SDN qui permet des expériences haute-fidélités qui émulent des environnements réseau réels. Mininet-wifi étend le populaire émulateur Mininet avec des stations sans fil virtuelles et des points d'accès tout en conservant les fonctionnalités SDN d'origine et une architecture logicielle de virtualisation légère [70].



#### 3.4 POX CONTROLLER :

Est un contrôleur OpenFlow/SDN (Software Defined Networking) open source basée sur Python. POX est utilisé pour accélérer le développement et le prototypage de nouvelles applications réseau. Le contrôleur POX est préinstallé avec la machine virtuelle Mininet. Avec le contrôleur POX [71].



- **Protocole OpenFlow** : est une norme dans l'architecture de réseau (SDN). Ce protocole définit la connexion entre le contrôleur SDN et le périphérique / agent réseau. Il prend les informations de l'application et les convertit en entrée de streaming qui est transmise au commutateur. Il peut également être utilisé pour surveiller les statistiques des commutateurs et des ports dans la gestion du réseau [72].

### 3.5 SUMO :

Est un package de simulation de trafic open source comprenant des composants d'importation nette et de modélisation de la demande. Nous décrivons l'état actuel du package ainsi que les développements et extensions futurs. SUMO aide à étudier plusieurs sujets de recherche, par exemple le choix de l'itinéraire et l'algorithme des feux de circulation ou la simulation de la communication des véhicules [73].



### 3.6 PYTHON :

Python est un langage de programmation interprété, de haut niveau et orienté objet avec une sémantique dynamique. Ses structures de données intégrées de haut niveau combinées à un typage dynamique et à une liaison dynamique le rendent très attrayant pour le développement rapide d'applications ainsi que pour une utilisation en tant que langage de script ou de liaison pour relier des composants existants [74].



## 4. PROCESSUS DE SIMULATION :

Le processus de simulation se fait en plusieurs étapes :

### 4.1 L'INTEGRATION DES PROGRAMMES ET LES VERSIONS :

Pour simuler les réseaux VANETs, nous avons utilisé le simulateur réseau **Mininet-Wifi** sous **UBUNTU** et le simulateur de trafic routier **SUMO** , qui les montrons dans le tableau suivant :

**Tableau 5:** LES PROGRAMMES ET LES VERSIONS.

	LES PROGRAMMES	Les versions
1	VIRTUELLE BOX	6.1
2	UBUNTU	18.04 LTS
3	Mininet-Wifi	2.6
4	SUMO	1.17.0
5	POX contrôleur	0.7.0
6	Python	3.6

Et nous avons effectué des simulations basées sur de nombreux paramètres spécifiques, dont les plus importants sont répertoriés dans le tableau ci-dessous :

**Tableau 6:** Les PARAMETRES

LES PARAMETRES	LA VALEUR
Zone de simulation	100m*100 m
Temps de simulation	60 s
Nombre de véhicules	2-8
Intervalle de paquets	20-50
Norme de communication	IEEE 802.11p

## 5. LES ETAPES D'INSTALLATION :

### 5.1 LES COMMANDES PRINCIPALES D'INSTALLATION MININET-WIFI :

Pour installer Mininet-wifi, nous écrivons la commande suivante dans le terminal, comme illustré à la **Figure 30** :

« ~ / **Mininet-wifi** \$ **sudo util /install.sh -wlnfv** »

- ✓ **install.sh** : le script se trouve dans le répertoire outil du Mininet-wifi. L'exécution de ce script avec les options fournies installera les composants et dépendances requis pour Mininet-wifi.

- ✓ **Les options install.sh :**

- W** : Installe les dépendances sans fil.
- n** : Installe les dépendances Mininet-Wifi.
- f**: Installe OpenFlow.
- v** : Installe le commutateur OpenvSwitch.
- l** : Installe medium, un démon médium sans fil.

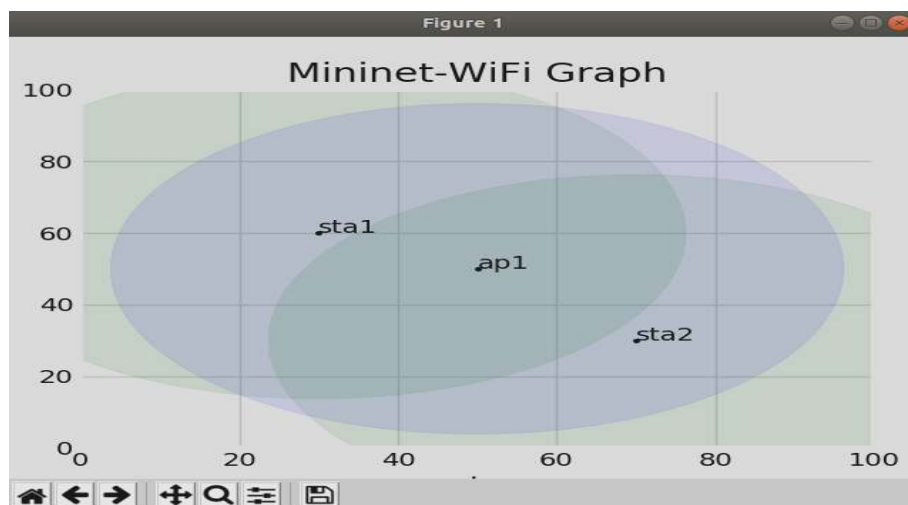
```

ubuntu@ubuntu-VirtualBox: ~/mininet-wifi
File Edit View Search Terminal Help
ubuntu@ubuntu-VirtualBox:~/mininet-wifi$
ubuntu@ubuntu-VirtualBox:~/mininet-wifi$ sudo util/install.sh -Wlnfv
Detected Linux distribution: Ubuntu 18.04 bionic amd64
Ubuntu
sys.version_info(major=2, minor=7, micro=17, releaselevel='final', serial=0)
Detected Python (python) version 2
Installing Mininet dependencies
Reading package lists... Done

```

**Figure 30:** installer Mininet wifi.

- La figure ci-dessous montre un exemple fonctionnant en Mininet-wifi pour confirmer le processus d'installation :



**Figure 31:** exemple topologie dans Mininet-wifi.

## 5.2 LES COMMANDES PRINCIPALES POUR D'INSTALLATION DU SUMO :

Pour installer sumo On tape dans le terminal la commande :

« Sudo apt-get install sumo sumo-tools sumo-doc ».

La figure 32 montre le discours ci-dessus :

```

ubuntu@ubuntu-VirtualBox: ~
File Edit View Search Terminal Help
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install sumo sumo-tools sumo-doc
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done

```

**Figure 32:** installation sumo.

## 6. LES ETAPES DU SIMULATION :

### 6.1 LA CREATION DE LA TOPOLOGIE :

Le châssis utilisé dans notre projet se compose deux contrôleurs, quatre points d'accès, quatre unités routières et de huit véhicules (car). Cette structure a été créée grâce à l'émulateur Mininet-Wifi. Après avoir lancé ce dernier, nous avons créé la topologie sous la forme d'un fichier python contenant les parties suivantes :

#### 1) Importation de quelques bibliothèques de Mininet-wifi :

Cette figure montre les bibliothèques utilisées en topologie :

```
from mininet.node import RemoteController
from mininet.log import info
from mn_wifi.node import UserAP
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi
from mn_wifi.sumo.runner import sumo
from mn_wifi.link import wmediumd, mesh
from mn_wifi.wmediumdConnector import interference
```

**Figure 33:** les bibliothèques de mininet-wifi.

#### 2) La définition les composants et les paramètres de topologie :

Cette figure montre des paramètres et des composantes utilisées en topologie :

```
def topology():
    "Create a network."
    net = Mininet_wifi(controller=RemoteController, accessPoint=UserAP,
                      link=wmediumd, wmediumd_mode=interference)
```

**Figure 34:** la définition les paramètres de topologie.

#### 3) Création des nœuds véhicules et les unités routières :

Cette figure montre les créations des nœuds véhicules(car) et unités routières (rsu) :

```
info("*** Creating nodes: car\n")
cars = []
for id in range(0, 21):
    cars.append(net.addCar('car%s' % (id + 1), wlans=2, encrypt='wpa2,'))

info("*** Creating nodes: rsu\n")
rsus = []
for id in range(0, 4):
    rsus.append(net.addCar('rsu%s' % (id + 1), wlans=2, encrypt='wpa2,'))
```

**Figure 35 :** la création des Nœuds (car & rsu).



#### 4) Définition du classe des contrôleurs POX :

Dans les **figures 36** et **37**, nous avons défini les classes pour les deux contrôleurs

Avec des adresses IP et des numéros de port déférant :

```
from network_controller.poxcontroller import PoxController

if __name__ == '__main__':
    pox = PoxController(cmd='/home/ubuntu/pox/pox.py',
                        script='forwarding,ls_learning',
                        ofst='openflow.spanning_tree --no-flood --hold-down',
                        debug='log.level --DEBUG',
                        pretty_log='samples.pretty_log',
                        ofdc='openflow.discovery_host_tracker info.packet_dump',
                        ofport='openflow.of_01 --port=6653',
                        log='log --file=/home/ubuntu/SDVANETS/log_files/pox.log'
    )

    pox.run()
```

**Figure 36:** main de POX n°1.

```
from network_controller.poxcontroller import PoxController

if __name__ == '__main__':
    pox = PoxController(cmd='/home/ubuntu/pox/pox.py',
                        script='forwarding,ls_learning',
                        ofst='openflow.spanning_tree --no-flood --hold-down',
                        debug='log.level --DEBUG',
                        pretty_log='samples.pretty_log',
                        ofdc='openflow.discovery_host_tracker info.packet_dump',
                        ofport='openflow.of_01 --port=6654',
                        log='log --file=/home/ubuntu/SDVANETS/log_files/pox.log'
    )

    pox.run()
```

**Figure 37:** main de POX n°2.

#### 5) Création des nœuds contrôleurs :

Dans la **figure 38**, nous avons créé deux contrôleurs et quatre points d'accès, le premier contrôleur est relié avec cette topologie à l'adresse IP 127.0.0.1 et le port =6653 du le protocole TCP et le deuxième contrôleur est relié avec à l'adresse IP 127.0.0.2 et le port=6654 :

```
info("*** Creating nodes: controllers\n")
c1 = net.addController(name='c1', ip='127.0.0.1', port=6653, protocol='tcp')
c2 = net.addController(name='c2', ip='127.0.0.1', port=6654, protocol='tcp')

e1 = net.addAccessPoint('e1', ssid='vanet-ssid', mac='00:00:00:11:00:01',
                        mode='g', channel='1', passwd='123456789a',
                        encrypt='wpa2', position='215.35,300.51,0.0')
e2 = net.addAccessPoint('e2', ssid='vanet-ssid', mac='00:00:00:11:00:02',
                        mode='g', channel='6', passwd='123456789a',
                        encrypt='wpa2', position='300.81,206.09,0.0')
e3 = net.addAccessPoint('e3', ssid='vanet-ssid', mac='00:00:00:11:00:03',
                        mode='g', channel='1', passwd='123456789a',
                        encrypt='wpa2', position='408.97,304.93,0.0')
e4 = net.addAccessPoint('e4', ssid='vanet-ssid', mac='00:00:00:11:00:04',
                        mode='g', channel='6', passwd='123456789a',
                        encrypt='wpa2', position='519.27,206.09,0.0')
```

**Figure 38 :** La création les Nœuds de contrôle.

#### 6) Configuration du modèle de propagation et des nœuds wifi :

Nous allons maintenant configurer les nœuds et le modèle de mobilité «log Distance », voir la **figure 39** :

```
info("*** Configuring Propagation Model\n")
net.setPropagationModel(model="logDistance", exp=4)

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

**Figure 39 :** la configuration de modelé.

#### 7) Création des liens entre les unités routières et les points d'accès :

Dans la **figure 40**, nous avons créé des liens entre les unités routières (rsu) et les points d'accès(e) :

```

info("*** Adding link\n")
net.addLink(rsus[0], rsus[1])
net.addLink(rsus[1], rsus[2])
net.addLink(rsus[2], rsus[3])
net.addLink(e1, e2)
net.addLink(e2, e3)
net.addLink(e3, e4)

for rsu in rsus:
    net.addLink(rsu, intf=rsu.wintfs[1].name,
                cls=mesh, ssid='mesh-ssid', channel=5)

for car in cars:
    net.addLink(car, intf=car.wintfs[1].name,
                cls=mesh, ssid='mesh-ssid', channel=5)

```

Figure 40 : Création des liens.

## 8) Démarrage le SUMO :

Ce qui permet au script Python d'interagir avec le simulateur de trafic SUMO pour contrôler les véhicules, récupérer des données de simulation avec le fichier « map.sumocfg » et connecté au le port "8813", voir la **figure 41**.

```

info("*** Starting sumo\n")
# change config_file name if you want
# use --random for active the probability attribute of sumo
net.useExternalProgram(program=sumo, port=8813,
                       # config_file='map.sumocfg',
                       extra_params=["--start --delay 1000"],
                       clients=1, exec_order=0)

```

Figure 41 : Démarrage le sumo.

## 9) Démarrage le réseau :

Dans la **figure 42**, nous avons configuré les adresses IP des unités routières (rsu) et des véhicules (car) dans le réseau virtuel simulé, définit les positions des unités routières et démarre le réseau avec les deux contrôleurs et les points d'accès :

```

info("*** Starting network\n")
net.build()
c1.start()
c2.start()
e1.start([c1])
e2.start([c1])
e3.start([c2])
e4.start([c2])

for rsu in rsus:
    rsu.setIP('192.168.0.%s/24' % (int(rsus.index(rsu)) + 101),
             intf='%s-wlan0' % rsu)
    rsu.setIP('192.168.1.%s/24' % (int(rsus.index(rsu)) + 101),
             intf='%s-mpi' % rsu)

pos = ['215.35,300.51,0.0', '300.81,206.09,0.0', '408.97,304.93,0.0', '519.27,206.09,0.0']
for rsu, pos in zip(rsus, pos):
    rsu.setPosition(pos=pos)

for car in cars:
    car.setIP('192.168.0.%s/24' % (int(cars.index(car)) + 1),
             intf='%s-wlan0' % car)
    car.setIP('192.168.1.%s/24' % (int(cars.index(car)) + 1),
             intf='%s-mpi' % car)

```

Figure 42 : Démarrage le réseau.

## 10) Démarrage la télémétrie :

Initialisation la télémétrie dans le réseau virtuel pour collecter les positions des nœuds (véhicule et points d'accès et rsu), voir la **figure 43**.

```

info("*** Starting telemetry\n")
# Track the position of the nodes
nodes = net.cars + net.aps
print(nodes)
net.telemetry(nodes=nodes, data_type='position',
             min_x=0, min_y=0,
             max_x=850, max_y=650)

```

Figure 43 : Démarrage la télémétrie.

## 6.2 LANCEMENT DE LA TOPOLOGIE :

Après avoir créé la topologie, nous activons le contrôleur POX1 en tapant la commande illustrée dans la figure ci-dessous :

```
ubuntu@ubuntu-VirtualBox:~/pox$ ./pox.py forwarding.l2_learning
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
```

**Figure 44** : Le lancement du contrôleur POX1.

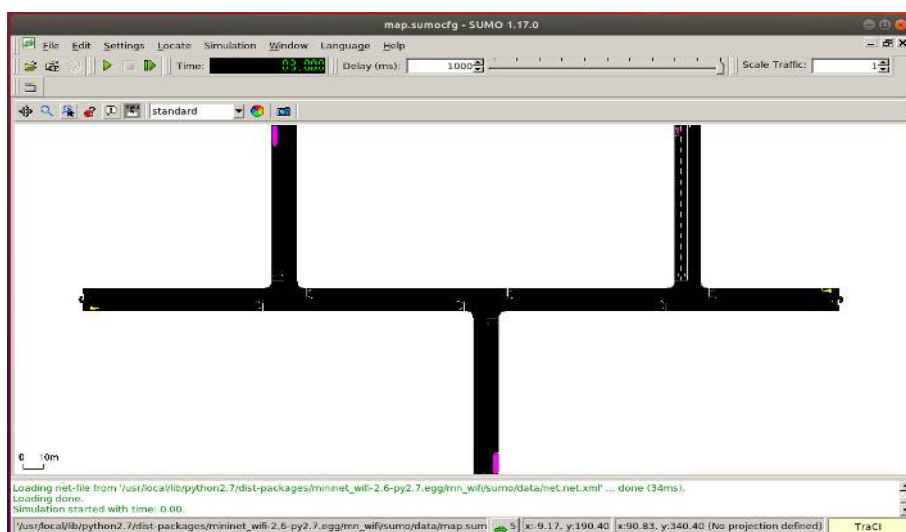
Après avoir créé une topologie, nous ouvrons un terminal, nous entrons dans le fichier "sdvanets", après ont lancé notre fichier Python "sdvanets" grâce à la commande de **la figure 45** :

```
ubuntu@ubuntu-VirtualBox: ~/sdvanets
File Edit View Search Terminal Help
ubuntu@ubuntu-VirtualBox:~/sdvanets$
ubuntu@ubuntu-VirtualBox:~/sdvanets$ sudo python3 -m sdvanets
[sudo] password for ubuntu:
Running Pox Controller...
Pox controller started.
Running Scenario...
Setting to default scenario...
```

**Figure 45** : lancement la simulation.

Après avoir exécuté la commande simulation run, nous obtenons les fenêtres suivantes :

- La figure suivante montre l'ouverture de la fenêtre SUMO pour démarrer la simulation en coordination avec Mininet-wifi :



**Figure 46** : interface SUMO.

b) 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 :

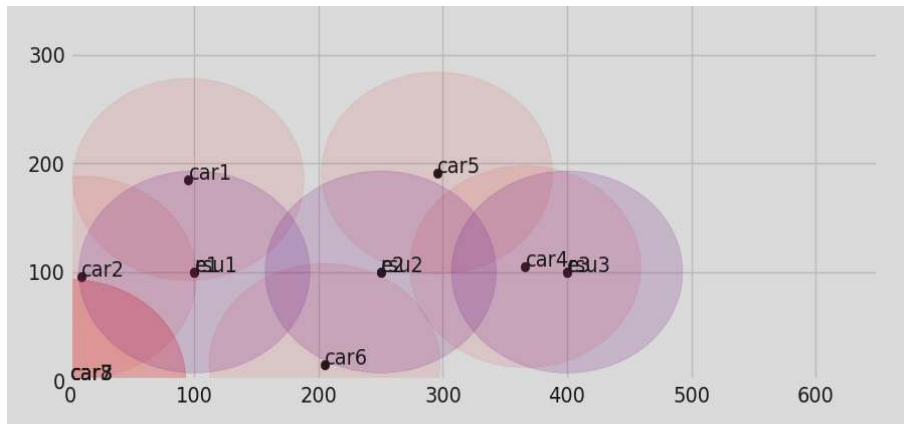


Figure 47 : Visualisation du Graph à l'instant t1.

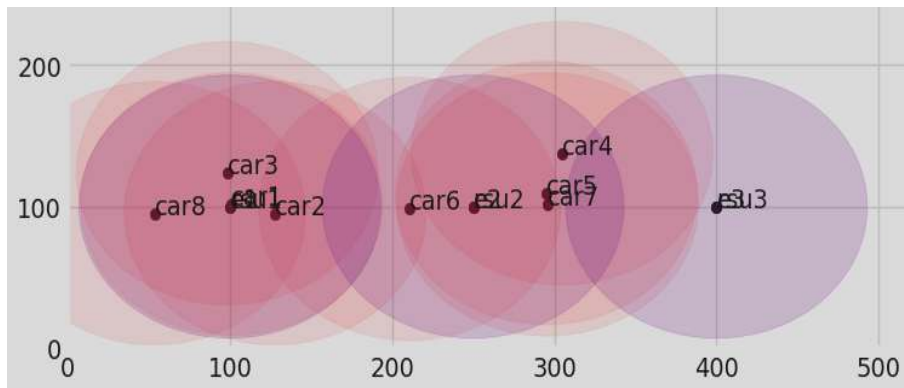


Figure 48 : Visualisation du Graph à l'instant t2.

c) Dans la **figure 49**, la fenêtre de surveillance affiche tous les scénarios et processus en cours dans le réseau, supervisés par le contrôleur POX. Le contrôleur POX joue un rôle essentiel dans la gestion du réseau SDN en utilisant le protocole OpenFlow pour envoyer et transmettre des messages entre les différents éléments du réseau pour la gestion et le contrôle du réseau SDN.

```
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
[openflow.spanning_tree ] Spanning tree component ready
[host_tracker           ] host_tracker ready
[info.packet_dump      ] Packet dumper running
[core                   ] POX 0.7.0 (gar) going up...
[core                   ] Running on CPython (3.6.9/Mar 10 2023 16:46:00)
[core                   ] Platform is Linux-5.4.0-148-generic-x86_64-with-Ubuntu
-18.04-bionic
[version                ] Support for Python 3 is experimental.
[core                   ] POX 0.7.0 (gar) is up.
[openflow.of_01         ] Listening on 0.0.0.0:6653
[openflow.of_01         ] [10-00-00-00-00-03 4] connected
[openflow.discovery     ] Installing flow for 10-00-00-00-00-03
[openflow.spanning_tree] Disabling flooding for 3 ports
[openflow.of_01         ] [10-00-00-00-00-01 2] connected
[openflow.discovery     ] Installing flow for 10-00-00-00-00-01
[openflow.spanning_tree] Disabling flooding for 3 ports
[host_tracker           ] Learned 17592186044419 2 10:00:00:00:02
[dump:10-00-00-00-00-03] [ethernet][ipv6][icmpv6][24 bytes]
[openflow.of_01         ] [10-00-00-00-00-02 3] connected
[openflow.discovery     ] Installing flow for 10-00-00-00-00-02
[openflow.spanning_tree] Disabling flooding for 4 ports
[openflow.discovery     ] Dropping LLDP packet 261
[openflow.discovery     ] Link detected: 10-00-00-00-00-01.2 -> 10-00-00-00-00-0
```

Figure 49 : fenêtre de contrôleur POX.

d) La **figure 50** montre l'interface de ligne de commande (CLI) de Mininet-wifi, où certaines commandes peuvent être exécutées dans cette interface telles que :

- **Mininet-wifi>nodes** : pour afficher les nœuds de notre topologie réseau.
- **Mininet-wifi>links** : cette commande est pour afficher les liens du réseau.
- **Mininet-wifi>intfs** : afin d'afficher les interfaces du réseau.
- **Mininet-wifi>xterm X** : pour ouvrir les fenêtres des nœuds(X) du réseau.

```

Terminal
File Edit View Search Terminal Help
*** Creating nodes: car
*** Creating nodes: rsu
*** Creating nodes: controller
*** Configuring Propagation Model
*** Configuring wifi nodes
*** Connecting to wmediumd server /var/run/wmediumd.sock
*** Adding link
*** Starting sumo
*** Starting network
Retrying in 1 seconds
*** Starting telemetry
*** Starting agents
*** Running CLI
*** Starting CLI:
mininet-wifi>

```

**Figure 50** : ligne interface de commande (CLI).

### 6.3 EVALUATION DE PERFORMANCE DE RESEAU SDN VS LE RESEAUX DE VEHICULAIRES TRADITIONNEL :

Dans cette section, nous avons examiné les performances de déploiement de données basées sur le SDN de l'Internet des véhicules et les avons comparées à celles d'un réseau véhiculaire conventionnel. Nous avons utilisé une topologie qui est un réseau véhiculaire conventionnel avec un nombre de véhicules équivalent par rapport à une infrastructure SDN. Les critères de performance pris en compte incluent la **latence** moyenne, le ratio de livraison des paquets (**PDR**) et le **débit**. Chacun de ces critères est brièvement expliqué. Nous avons mesuré les trois mesures de performance : la latence moyenne, le taux de livraison des paquets (PDR) et le débit (Throughput) sur tous les véhicules.

#### 6.3.1 Test la connectivité entre les nœuds :

Pour ouvrir les fenêtres du véhicule sur l'interface de ligne de commande (CLI) de Mininet-wifi, nous exécutons la commande suivante : « **mininet-wifi>xterm car2 car9** ».

La **figure 51** montre probablement une capture d'écran de la fenêtre du terminal car2 où nous avons exécuté la commande « **ping** » et obtenu une réponse réussie, confirmant ainsi la connectivité entre les nœuds du réseau.



```

"Node: car9"
root@ubuntu-VirtualBox:~# ping -c 10 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=1.84 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=2.21 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=3.79 ms

--- 192.168.0.2 ping statistics ---
10 packets transmitted, 3 received, 70% packet loss, time 9124ms
rtt min/avg/max/mdev = 1.840/2.615/3.791/0.846 ms
root@ubuntu-VirtualBox:~#
    
```

Figure 51 : Test la connectivité.

### 6.3.2 Mesure de latence :

La latence reflète le délai de transmission des paquets de données du véhicule source au véhicule de destination. Cette métrique est très importante dans le cas de l'envoi de messages urgents. Dans notre SDN-IoV, nous définissons la latence comme suit [78] :

$$Latence = \frac{Ps}{Tr} \dots\dots\dots(1)$$

**Ps** : la taille du paquet et **Tr** : le débit de transmission (paquet/ms).

La latence entre les véhicules du réseau SDN a été mesurée en exécutant la commande « ping » entre chaque paire de véhicules. Le temps aller-retour (RTT) a été enregistré pour chaque paquet envoyé. La **figure 52** illustre les variations du RTT après l'exécution du « ping » entre les véhicules dans le réseau SDN.

<pre> "Node: car2" ^C --- 192.168.0.1 ping statistics --- 29 packets transmitted, 10 received, +6 errors, 65% packet loss, time 28433ms rtt min/avg/max/mdev = 1.124/6.276/28.639/7.749 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>	<pre> "Node: car1" ^C --- 192.168.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5013ms rtt min/avg/max/mdev = 1.972/4.841/9.544/2.444 ms root@ubuntu-VirtualBox:~/sdvanets#                 </pre>
<pre> "Node: car4" ^C --- 192.168.0.1 ping statistics --- 27 packets transmitted, 11 received, +9 errors, 59% packet loss, time 26365ms rtt min/avg/max/mdev = 1.438/5.168/12.051/3.836 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>	<pre> "Node: car3" ^C --- 192.168.0.1 ping statistics --- 47 packets transmitted, 18 received, +3 errors, 61% packet loss, time 46706ms rtt min/avg/max/mdev = 1.045/5.727/34.634/7.962 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>
<pre> "Node: car6" ^C --- 192.168.0.1 ping statistics --- 38 packets transmitted, 17 received, +6 errors, 55% packet loss, time 37491ms rtt min/avg/max/mdev = 2.531/8.544/39.185/8.978 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>	<pre> "Node: car5" ^C --- 192.168.0.1 ping statistics --- 38 packets transmitted, 20 received, +3 errors, 47% packet loss, time 37422ms rtt min/avg/max/mdev = 1.143/14.857/66.708/18.800 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>
<pre> "Node: car8" ^C --- 192.168.0.1 ping statistics --- 13 packets transmitted, 13 received, 0% packet loss, time 12030ms rtt min/avg/max/mdev = 1.200/2.613/5.893/1.197 ms root@ubuntu-VirtualBox:~/sdvanets#                 </pre>	<pre> "Node: car7" ^C --- 192.168.0.1 ping statistics --- 28 packets transmitted, 10 received, +3 errors, 64% packet loss, time 27424ms rtt min/avg/max/mdev = 2.043/5.622/17.271/4.918 ms, pipe 4 root@ubuntu-VirtualBox:~/sdvanets#                 </pre>

Figure 52 : les différents RTT de scenario SDN.

### 6.3.3 Mesure le ratio de livraison de paquets (PDR) :

Cela reflète le pourcentage de paquets envoyés avec succès sur le nombre total de paquets envoyés lorsque des informations d'urgence ont été diffusées. Nous avons mesuré cette métrique comme suit [78] :

$$PDR = \frac{Pr}{Pt} \dots\dots\dots(2)$$

**Pr** : le rapport des paquets reçus avec succès par le véhicule de destination et **Pt** : tous les paquets envoyés par le véhicule source.

Pour mesurer le pourcentage de livraison de paquets dans le réseau entre tous les véhicules, nous avons mis en place un scénario où la véhicule "car8" agit en tant que serveur, tandis que les autres véhicules sont des clients. Les véhicules clients envoient un ensemble de paquets fragmentés sous forme de datagrammes en utilisant le protocole UDP vers le véhicule serveur. Pour mettre en œuvre ce scénario, nous avons ouvert les fenêtres du serveur et des clients, et entré les commandes suivantes :

- Dans la fenêtre serveur "car8" : ~ # **iperf -s -u -p 5566 -i 1**
- Dans la fenêtre client "car x" : ~ # **iperf -c 192.168.0.8 -u -t 15 -p 5566**

« **iperf** : qui est utilisé pour mesurer la performance du réseau »

« **-s** : c'est l'option qui indique à iperf de fonctionner en tant que serveur »

« **-u** : C'est l'option qui indique à iperf de fonctionner en mode UDP »

« **-p 5566** : C'est l'option qui spécifie le numéro de port sur lequel le serveur iperf va écouter les connexions entrantes. »

« **-t 15** : la durée est fixée à 15 secondes »

Les **figures 53,54** illustrent la connexion établie en mode client-serveur via le port du protocole UDP. Ces figures peuvent représenter les fenêtres des clients et du serveur que vous avez ouvertes, montrant l'échange de paquets entre eux.

```

"Node: car8"
root@ubuntu-VirtualBox:~# iperf -s -u -p 5566
-----
Server listening on UDP port 5566
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----

```

**Figure 53** : écoute du serveur sur UDP.

```

"Node: car2"
root@ubuntu-VirtualBox:~# iperf -c 192.168.0.8 -u -p 5566 -t 15
Client connecting to 192.168.0.8, UDP port 5566
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 37] local 192.168.0.2 port 44455 connected with 192.168.0.8 port 5566
[ ID] Interval      Transfer      Bandwidth
[ 37] 0.0-15.0 sec  1.88 MBytes  1.05 Mbits/sec
[ 37] Sent 1339 datagrams
[ 37] Server Report:
[ 37] 0.0-15.0 sec  1.88 MBytes  1.05 Mbits/sec  0.000 ms  0/ 1339 (0%)
root@ubuntu-VirtualBox:~#

```

**Figure 54** : la connectivité client-serveur.



La **figure 55** présente un exemple du ratio de livraison de paquets après l'exécution des commandes mentionnées précédemment. Elle montre le résultat de la mesure du ratio de livraison des paquets, qui est calculé en fonction du nombre de paquets reçus par le serveur par rapport au nombre de paquets envoyés par les clients. Le ratio de livraison de paquets est généralement exprimé en pourcentage, indiquant la proportion de paquets qui ont été livrés avec succès.

```

"Node: car2"
Client connecting to 192.168.0.1, UDP port 5566
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 33] local 192.168.0.2 port 34777 connected with 192.168.0.1 port 5566
[  ID] Interval      Transfer    Bandwidth
[ 33]  0.0-15.0 sec  1.88 MBytes  1.05 Mbits/sec
[ 33] Sent 1339 datagrams
[ 33] Server Report:
[ 33]  0.0- 8.0 sec  1.02 MBytes  1.07 Mbits/sec  0.000 ms  610/ 1339 (0%)
root@ubuntu-VirtualBox:~#

```

**Figure 55** : Exemple de résultat PDR.

Après avoir réalisé le scénario décrit précédemment, nous avons analysé les résultats obtenus et les avons représentés sous forme de courbes graphiques. Ces courbes graphiques ont été incluses dans la partie "Résultats de la simulation".

### 6.3.4 Mesure le débit (Throughput) :

Cela représente le nombre de paquets transmis par unité de temps, exprimé comme suit [78] :

$$\text{Débit} = \frac{\text{Nb}}{t} \dots\dots\dots(3)$$

**Nb** : le Nombre de paquets transmis et **t** : période de temps spécifique.

Pour mesurer le débit (Throughput) dans le scénario précédent, vous avez utilisé les valeurs de quantités de données envoyées et reçues avec succès pour chaque période de transmission. Nous avons converti la quantité de données envoyées illustrée à la **figure 55** de Mbytes en Mbits afin d'obtenir des mesures cohérentes.

En convertissant les quantités de données en bits, nous pouvons obtenir une mesure précise du débit de transmission. Le débit est généralement exprimé en bits par seconde (bps) ou en mégabits par seconde (Mbps).

Nous avons collecté les résultats de débit pour chaque période de transmission et les avons présentés sous forme de courbes graphiques dans la section des résultats de la simulation.

## 7. RESULTATS DES SIMULATIONS

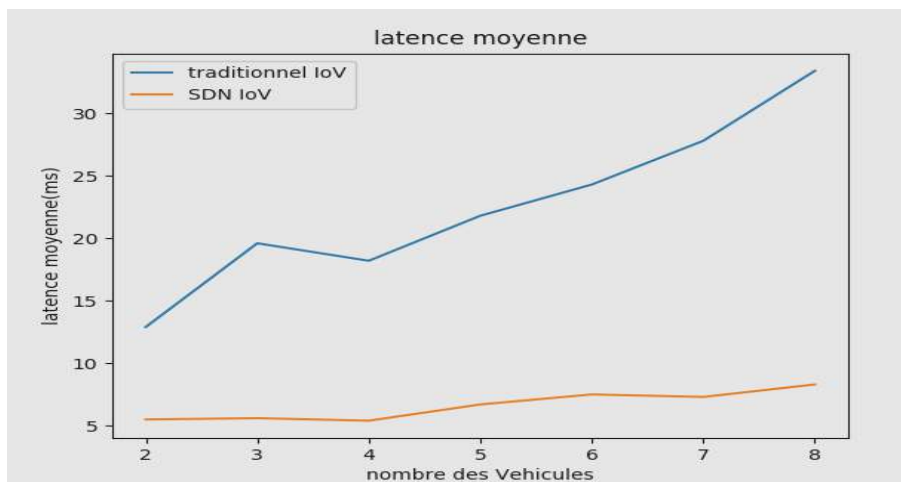
Après avoir évalué les performances des deux réseaux, à savoir le réseau SDN pour l'Internet des véhicules et le réseau véhiculaire traditionnel, en termes de latence moyenne, de taux de livraison des paquets et de débit, nous avons inclus trois figures dans cette section. Chaque figure présente deux courbes graphiques représentant des mesures spécifiques.

- **Latence moyenne :**

Dans les deux courbes de la **figure 56**, il est clairement visible que plus il y a de véhicules, plus le temps de latence moyen augmente. Cependant, il est important de noter que dans un réseau de véhicules conventionnel, avec l'augmentation du nombre de véhicules, on observe une augmentation significative du temps de latence, atteignant jusqu'à 34 ms.

En revanche, dans le réseau SDN, bien qu'il y ait une légère augmentation de la latence avec l'augmentation du nombre de véhicules, atteignant environ 8 millisecondes au maximum, le temps de latence reste généralement beaucoup plus faible par rapport à celui d'un réseau de véhicules classique.

Cela suggère que le réseau SDN offre une meilleure gestion du temps de réponse, même avec une augmentation du nombre de véhicules. Par conséquent, le réseau SDN peut être considéré comme plus efficace pour maintenir des temps de latence plus faibles, ce qui peut être particulièrement bénéfique dans les applications nécessitant une connexion à faible latence, telles que les communications en temps réel ou les systèmes de contrôle critiques.

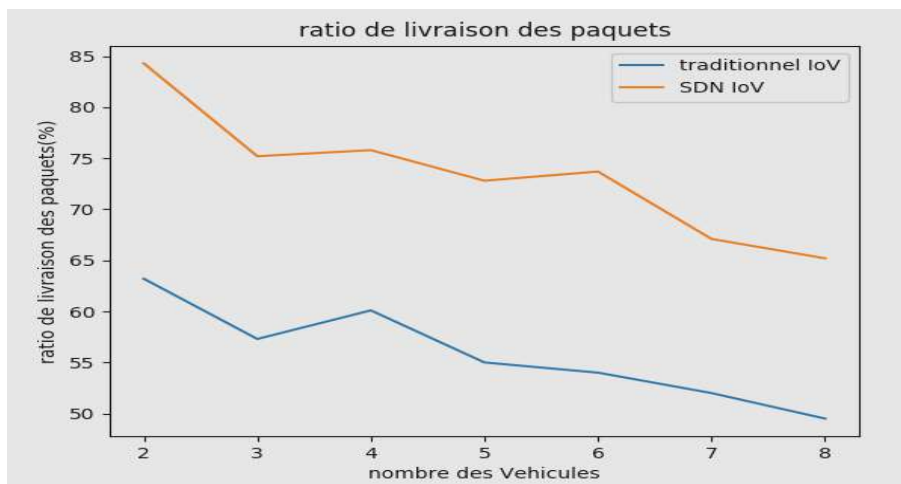


**Figure 56 :** comparaison de Latence entre les deux scénarios.

- **Le ratio de livraison de paquets (PDR) :**

La **figure 57** montre que dans les deux cas, plus il y a de véhicules, plus le pourcentage de livraisons de colis est faible. Dans un réseau de véhicules conventionnel, une augmentation du nombre de véhicules entraîne une réduction significative du pourcentage de livraison de colis, allant de 50% à 60%. En revanche, dans le réseau SDN, bien qu'il y ait une légère diminution du taux de livraison des paquets avec l'augmentation du nombre de véhicules où le pourcentage varie de 85% à 65%, il reste généralement plus élevé que dans le réseau de véhicules traditionnel.

Cela suggère que le réseau SDN offre une meilleure capacité à livrer des colis avec succès, même avec une augmentation du nombre de véhicules. Par conséquent, le réseau SDN peut être considéré comme plus fiable en termes de livraison de paquets, ce qui est nécessaire dans les applications qui nécessitent une communication sans perte de données.



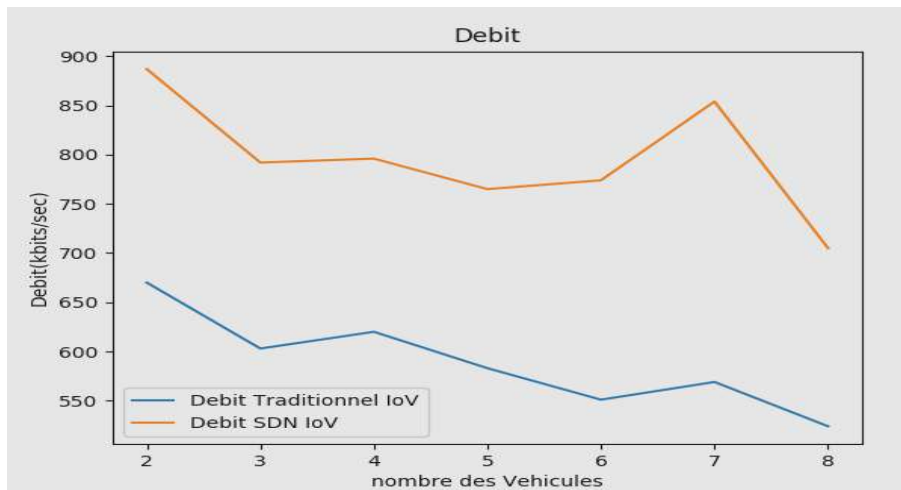
**Figure 57** : comparaison le PDR entre les deux scenarios.

- **Le débit (Throughput) :**

Le débit montre que, dans les deux cas du **figure 58**, plus le nombre de véhicules augmente, plus le taux de productivité diminue. Dans le réseau de véhicules traditionnel, une augmentation du nombre de véhicules entraîne une diminution significative du taux de productivité, passant de 600 kilobits par seconde à 500 kilobits par seconde (0,5 Mégabits par seconde).

En revanche, dans le réseau SDN, bien qu'il y ait une légère diminution du taux de productivité avec l'augmentation du nombre de véhicules, il commence à partir de 900 kilobits par seconde (0,9 Mégabits par seconde). Par conséquent, on observe que les performances du réseau SDN en termes de débit sont généralement supérieures à celles du réseau de véhicules traditionnel.

Cela indique que le réseau SDN offre une meilleure capacité de transmission de données, même avec une augmentation du nombre de véhicules. Ainsi, le réseau SDN peut être considéré comme offrant une meilleure productivité en termes de débit, ce qui est crucial pour des applications nécessitant une transmission rapide et efficace des données.



**Figure 58 :** comparaison le débit entre les deux scenarios.

## 8. CONCLUSION :

Dans ce chapitre nous avons proposé une architecture SDN basée sur le réseau de véhicules avec l'ajout du contrôleur POX. Nous avons évalué les performances du réseau en termes de temps de latence, de ratio de livraison des paquets et de débit en utilisant l'émulateur mininet-wifi, et les avons comparées à un réseau de véhicules traditionnel. Les résultats ont montré que dissémination de données basée sur SDN présente des avantages significatifs, tels qu'une gestion améliorée du temps de réponse (latence), une livraison plus fiable des paquets et des débits plus élevés. Cette approche offre des perspectives prometteuses pour améliorer les communications dans les réseaux de véhicules et faciliter le développement d'applications avancées.

## *Conclusion générale*

---

Les réseaux véhiculaires et les systèmes de transport intelligents (STI) sont encore en phase de développement. Malgré les avancées réalisées dans le domaine des STI, la communication au sein des réseaux automobiles demeure un sujet de recherche actif. Actuellement, il existe de nombreuses technologies de communication conçues spécifiquement pour répondre aux besoins des réseaux automobiles. Par conséquent, la simulation de scénarios réels est la méthode la plus largement utilisée pour concevoir et évaluer les solutions proposées pour la mise en œuvre pratique des réseaux véhiculaires.

Les réseaux véhiculaires dédiés (VANETs), qui sont une forme de réseaux mobiles ad hoc (MANETs), ont été développés pour faciliter les communications entre les véhicules et les équipements fixes au sein d'un groupe ou d'une infrastructure. Cette technologie a émergé pour améliorer la sécurité routière, les conditions de circulation et le confort de déplacement.













L'Internet des véhicules (IoV) a élargi le concept des réseaux ad hoc en incluant Internet et en utilisant des technologies informatiques et des appareils sans fil pour cela. Cependant, il existe des problèmes d'évolutivité car il est difficile de gérer des volumes importants de données. Les architectures actuelles sont rigides, difficiles à contrôler et manquent de flexibilité et d'adaptabilité.













Les réseaux définis par logiciel (SDN) ont suscité un grand intérêt ces dernières années et ont provoqué un changement fondamental dans la conception des réseaux. Le marché a rapidement adopté le SDN comme un ensemble de solutions qui éliminent les frontières entre les applications et les architectures réseau. Le déploiement des applications devient plus facile et plus dynamique grâce au SDN.

Récemment, plusieurs études ont montré que l'intégration du SDN dans les réseaux véhiculaires peut apporter de la flexibilité, de la fiabilité et des API permettant de programmer les périphériques réseau dans différents langages, tout en améliorant la capacité d'évolutivité.












Dans cette mémoire, nous avons proposé une architecture de dissémination de données basée sur le SDN pour l'Internet des véhicules. Une topologie avec de nombreux nœuds véhicules, quatre un point d'accès et quatre unités routière et deux contrôleurs POX a été créé, et un simulateur Mininet-wifi a été utilisé pour mettre en œuvre le projet. L'objectif de ce travail était de garantir une installation pratique et efficace des contrôleurs SDN dans le cadre de la dissémination des données, assurant une réception optimale des paquets sur le réseau et réduisant la latence.













## *Bibliographie*














-  [01] MERZOUGUI, S. E. (2020). Un schéma d'authentification sécurisé pour l'internet des véhicules.
-  [02] Benderradji, D., & Bouanane, I. *Une architecture IoV hétérogène pour le transfert des données dans la communication Véhicule-Infrastructure* (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA).
-  [03] Youcef, M. (2019). Intégration du Software-Defined Networking (SDN) dans les réseaux de véhicules (VANETs).
-  [04] Y. Liu, C. Chen, and S. Chakraborty. A software defined network architecture for geobroadcast in vanets. En 2015 IEEE International Conférence on Communications (ICC), London, 2015.
-  [05] K. Ait-Ali : « Modélisation et étude de performances dans les réseaux VANET, Thèse de Doctorat », Université de technologie Belfort-Montbéliard, 2013.
-  [06] F Mezghani: « La dissémination de contenus dans les réseaux véhiculaires » - 2015 - oatao.univ-toulouse.fr.
-  [07] A. Zekri and W. Jia. Heterogeneous vehicular communications: A comprehensive study. *Ad Hoc Networks*, 75–76():52–79, 2018.
-  [08] BOUKSANI, W. (s.d.). GESTION DE LA PROTECTION DE LA VIE PRIVÉE DANS LES RESEAUX VEHICULAIRES (VANET). Récupéré sur depot-e.uqtr.ca. Consulté le 22 Avril 2023 à 17 : 54 ;pp. 6-18. URL : <https://depot-e.uqtr.ca/id/eprint/8166>.
-  [09] J. Petit, “Surcout de l’authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires,” Ph.D. dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011.
-  [10] T. Soufian, Vers des réseaux véhiculaires (VANET) programmables grâce à la technologie SDN (Software Defined Network), Toulouse, Université Toulouse: Thèse DOCTORAT, 2020.
-  [11] M. A. C. F. K. MENDIBOURE Léo, Distribution géographique de données dans l’Internet des Véhicules : une approche logicielle et sécurisée utilisant les réseaux cellulaires, UNIVERSITÉ DE BORDEAUX : THÈSE DOCTORAT, 2020.
-  [12] EFORT, «Le Protocole OpenFlow dans l’Architecture SDN (Software Defined Network) » 2016.

-  [13] Bendif Nour el houda, Bensalem Rania, “ Implémentation du modèle SDN dans les réseaux VANET” , UNIVERSITE BADJI MOKHTAR – ANNABA , 2020 .
-  [14] Burgod Céline. (2009). Contribution à la sécurisation du routage dans les réseaux ad hoc. Thèse De Doctorat De L'université De Limoges Ecole Doctorale. Chapitre 1 pp 7-17. 125pages.
-  [15] Corson Scott, Macker Joseph. (1999). Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Request for Comments (RFC) 2501, IETF. January 1999.
-  [16] T. A. H. B. G. P. L. P. J. R. S. S. J. T. Nick McKeown, «OpenFlow: Enabling innovation in campus networks, » Computer Communication Review, vol. 38 (2), pp. 69 - 75, 2008.
-  [17] K. B. M. A. A. F. Traore Issa, « Etude du nomadisme dans un Cloud éducatif administré par la technologie SDN/OpenFlow » chez conférence WACREN, Institut de recherches mathématique, Université Félix Houphouët-Boigny, 2016.
-  [18] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane. Software defined networking based vehicular ad hoc network with fog computing. In IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, 2015.
-  [19] M. Zhu, Z. P. Cai, M. Xu, and J. N. Cao. Software-defined vehicular networks: Opportunities and challenges. Energy Science and Applied Technology, CRC Press, pages 247–251, 2015.
-  [20] M. Azizian, S. Cherkaoui, and A. S. Hafid. Vehicle software updates distribution with sdn and cloud computing. IEEE Communications Magazine, 55(8) :74–79, 2017.
-  [21] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane. Software defined networking based vehicular adhoc network with fog computing. In IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, 2015.
-  [22] Z. He, J. Cao, and X. Liu. Sdvn: enabling rapid network innovation for heterogeneous vehicular communication. IEEE Network, 30(4) :10–15, 2016.
-  [23] DETUNJI, gestion de l’anonymat ET de la traçabilité dans les réseaux véhiculaires sans fil. Université du Qué-bec à Trois-Rivières, Département de mathématiques ET informatique appliquées : Thèse de doctorat, 2014.
-  [24] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou. A survey on architecture challenges, and solutions. IEEE Communications Surveys and Tutorials, 17(4) :2377–2396, 2015.



















- [25]  Central Authority. Consulté le 13 avril 2023 à 19 : 12. URL : <https://www.hcch.net/en/states/authorities/details3/?aid=75>.
- [26]  Station de base (en télécommunications). Récupéré sur Glossary. Consulté le 20 Avril 2023 à 14 : 22. URL: <https://boowiki.info/art/emetteurs-radio/station-de-base.html>.
- [27]  A.FitahaA.BadriaM.MoughitaA.Sahela. (s.d.). Performance of DSRC and WIFI for Intelligent Transport Systems in VANET. Consulté le 20 Avril 2023 à 17 : 13. URL : <https://www.sciencedirect.com/science/article/pii/S1877050918301455?via%3Dihub>
- [28]  T. Soufian, Vers des réseaux véhiculaires (VANET) programmables grâce à la technologie SDN (Software Defined Network), Toulouse, Université Toulouse: Thèse DOCTORAT, 2020.
- [29]  Santa J, Gómez-Skarmeta A. F et Sánchez-Artigas M. (2008). Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks. Computer Communications, vol. 31, no. 12. Jul 2008. Pp. 2850-2861.
- [30]  Xu Q., Jiang D. (2003). Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum. Vehicular Technology Conference. VTC 2003-Spring. The 57th IEEE Semiannual, vol. 4. Pp 2451-2455. Apr. 2003.
- [31]  Introduction to Internet of Things (IoT) Consulté le 18 Mai 2023 à 23 : 28  
URL: <https://www.geeksforgeeks.org/introduction-to-internet-of-things-iot-set-1/>
- [32]  A Simulation Approach of the Internet of Intelligent Vehicles for Closed Routes in Urban Environments, sur [ieeexplore.ieee.org](http://ieeexplore.ieee.org). Consulté le 18 Mai 2023 à 22 : 50  
URL: <https://ieeexplore.ieee.org/document/8768841>
- [33]  Ahmed ALIOUA , “ Intégration du Software-Defined Networking (SDN) dans les réseaux de véhicules (VANETs)” , UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE ,2019.
- [34]  Routage dans les réseaux véhiculaire  
URL : <https://www.theses.fr/2014PA132056> Consulté (le 13 Mai 2023)
- [35]  Zeyun, Yao Wenbing, Ni Qiang et Song Yonghua. (2008). DeReQ: A QoS Routing Algorithm for Multimedia Communications in Vehicular Ad Hoc Networks. Department of Electronic and Computer Engineering Brunel University, UK. 2008.
- [36]  Jerbi Moez.(2008). Protocoles pour les communications dans les reseaux de vehicules en environnement urbain : Routage et GeoCast bases sur les intersections. Thèse de doctorat de L'universite D'evry Val D'essonne . 06 Novembre 2008.
- [37]  Bendif Nour el houda - Bensalem Rania, UNIVERSITE BADJI MOKHTAR – ANNABA, “Implémentation du modèle SDN dans les réseaux “ , 2021.

-  [38] M. FAROUK MEZGHANI, UNIVERSITÉ DE TOULOUSE, ‘‘ LA DISSEMINATION DE CONTENUS DANS LES RESEAUX VEHICULAIRES’’, 2015
-  [39] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Veris-simo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76,2015.
-  [40] Ghafoor, H., & Koo, I. (2017). CR-SDVN: A cognitive routing protocol for software-defined vehicular networks. *IEEE sensors journal*, 18(4), 1761-1772.
-  [41] Ku, I., Lu, Y., Gerla, M., Gomes, R. L., Ongaro, F., & Cerqueira, E. (2014, June). Towards software-defined VANET: Architecture and services. In 2014 13th annual Mediterranean ad hoc networking workshop (MED-HOC-NET) (pp. 103–110). IEEE.
-  [42] Soua, Ahmed, and Samir Tohme. "Multi-level SDN with vehicles as fog computing infrastructures: A new integrated architecture for 5G-VANETs."2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). IEEE, 2018.
-  [43] Cao, B., Sun, Z., Zhang, J., & Gu, Y. (2021). Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), 3832-3840.
-  [44] Kadhim, A. J., Seno, S. A. H., & Shihab, R. A. (2018). Routing strategy for Internet of vehicles based on hierarchical SDN and fog computing. *Journal of University of Babylon for Pure and Applied Sciences*, 26(10), 309-319.
-  [45] Khoury, J., Sami, H., Safa, H., & El-Hajj, W. (2019, June). On the use of software defined wireless network in vehicular fog computing environments. In 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 1198-1203). IEEE.
-  [46] STORCK, Carlos Renato et DUARTE-FIGUEIREDO, Fátima. A 5G V2X ecosystem providing internet of vehicles. *Sensors*, 2019, vol. 19, no 3, p. 550.
-  [47] Din, S., Paul, A., & Rehman, A. (2019). 5G-enabled Hierarchical architecture for software-defined intelligent transportation system. *Computer Networks*, 150, 81-89.
-  [48] Liu, K., Xu, X., Chen, M., Liu, B., Wu, L., & Lee, V. C. (2019). A hierarchical architecture for the future internet of vehicles. *IEEE Communications Magazine*, 57(7), 41-47.
-  [49] Syed, S. A., Rashid, M., Hussain, S., Azim, F., Zahid, H., Umer, A., ... & Vargas-Rosales, C. (2022). QoS aware and fault tolerance-based software-defined vehicular networks using cloud-fog computing. *Sensors*, 22(1), 401.

-  [50] Duan, X., Wang, X., Liu, Y., & Zheng, K. (2016, September). SDN enabled dual cluster head selection and adaptive clustering in 5G-VANET. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)* (pp. 1-5). IEEE.
-  [51] Zhu, W., Gao, D., Zhao, W., Zhang, H., & Chiang, H. P. (2018). SDN-enabled hybrid emergency message transmission architecture in internet-of-vehicles. *Enterprise Information Systems*, *12*(4), 471-491.
-  [52] << ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN) -ACM SIGCOMM 2014 >>. [En ligne]. Disponible sur : <https://conferences.sigcomm.org/sigcomm/2014/hotsdn.php>. [Consulté le: 27 Mai 2023].
-  [53] K. Phemius, M. Bouet, et J. Leguay, <<DISCO: Distributed multi-domain SDN controllers>>, in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, p. 104.
-  [54] B. Lee, S. H. Park, J. Shin, et S. Yang, << IRIS: The Openflow-based Recursive SDN controller >>, in *16th International Conference on Advanced Communication Technology*, 2014, p. 1227-1231.
-  [55] Z. Cai, A. L. Cox, et T. S. E. Ng, « Maestro: A System for Scalable OpenFlow Control », déc. 2010.
-  [56] J. Medved, R. Varga, A. Tkacik, et K. Gray, « OpenDaylight: Towards a Model-Driven SDN Controller architecture », in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014, p. 106.
-  [57] N. Gude et al., <NOX Towards an Operating System for Networks >, *SIGCOMM Comput Commun Rev*, vol. 38, n°3, p. 105-110, juill.2008
-  [58] S. Kaur, J. Singh, et N. Singh Ghumman, << Network Programmability Using POX Controller >>, 2014.
-  [59] H. Akçay et D. Yiltas, << Web-Based User Interface for the Floodlight SDN Controller >>, *Int. J. Adv. Netw. Appl.*, vol. 8, p. 3175-3180, avr. 2017.
-  [60] C. El Khalfi, A. El Qadi, et H. Bennis, << A Comparative Study of Software Defined Networks Controllers >>, 2017, p. 15.
-  [61] J. Liu, K., Xiao, K., Dai, P., Lee, V. C., Guo, S., & Cao, J. (2020). Fog computing empowered data dissemination in software defined heterogeneous vanets. *IEEE Transactions on Mobile Computing*, *20*(11), 3181-3193.
-  [62] Abbas, M. T., Muhammad, A., & Song, W. C. (2020). SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach. *Journal of Ambient Intelligence and*

Humanized Computing, 11, 1265-1280.

-  [63] Chahal, M., & Harit, S. (2019). Network selection and data dissemination in heterogeneous software-defined vehicular network. *Computer Networks*, 161, 32-44.
-  [64] DARADE, Santosh A., AKKALAKSHMI, M., et PAGAR, Dr Nitin. SDN based load balancing technique in internet of vehicle using integrated whale optimization method. In: AIP Conference Proceedings. AIP Publishing LLC, 2022. p.020006.
-  [65] Zhu, W., Gao, D., Zhao, W., Zhang, H., & Chiang, H. P. (2018). SDN-enabled hybrid emergency message transmission architecture in internet-of-vehicles. *Enterprise Information Systems*, 12(4), 471-491.
-  [66] Raja, G., Dhanasekaran, P., Anbalagan, S., Ganapathisubramaniyan, A., & Bashir, A. K. (2020, July). SDN-enabled traffic alert system for IoV in smart cities. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHP) (pp. 1093-1098). IEEE.
-  [67] Ayoub Benchabana et Ramla Bensaci, ‘‘Analyse des protocoles de routage dans les réseaux Vanet ’’, Université Kasdi Merbah-Ouargl ,2014.
-  [68] VIRTUELLE BOX. Open-source du logiciel de virtualization. Disponible en ligne :  
URL : <https://www.virtualbox.org/> / 22.00 / (consulté le 20 Mai 2023)
-  [69] UBUNTU. System d’exploitation GNU/LINUX. Disponible en ligne :  
URL : <https://releases.ubuntu.com/> /22.37/ (consulté le 20 Mai 2023)
-  [70] Mininet-wifi URL : <https://intrig.dca.fee.unicamp.br/mininetwifi/> (consulté le 20 Mai 2023)
-  [71] ] Kaur, S., Singh, J., & Ghumman, N. S. (2014, August). Network programmability using POX controller.  
In ICCCS International conference on communication, computing & systems, IEEE (Vol. 138, p. 70). sn.
-  [72] Open flow URL : <https://www.section.io/engineering-education/openflow-sdn/> (consulté le 20 Mai 2023)
-  [73] SUMO URL : <https://www.eclipse.org/sumo/> (consulté le 20 Mai 2023)
-  [74] Python URL : <https://www.python.org/doc/essays/blurbs/> (consulté le 20 Mai 2023)
-  [75] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud. Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, 75(A) :453–471, 2014.

-  [76] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Open flow: enabling innovation in campus networks. *SIGCOMM Comput. Commun.*, 38(2) :69–74, 2008.
-  [77] D. V. Vijay Tijare, «THE NORTHBOUND APIs OF SOFTWARE DEFINED NETWORKS, » *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, October 2016.
-  [78] Azzaoui, N., Korichi, A., Brik, B., & Fekair, M. E. A. (2021). Towards optimal dissemination of emergency messages in Internet of Vehicles: A dynamic clustering-based approach. *Electronics*, 10(8), 979.

# ABRÉVIATIONS

<b>VANET</b>	: Vehicular Ad-hoc Network
<b>MANET</b>	: Mobile Ad hoc Network
<b>IOV</b>	: Internet Of Vehicle
<b>SDN</b>	: Software define Network
<b>OBU</b>	: On-Board Unit
<b>CA</b>	: Central Authority
<b>RSU</b>	: Road Side Unit
<b>SB</b>	: Station de Base
<b>V2V</b>	: Véhicule à Véhicule
<b>V2I</b>	: Véhicule à Infrastructure
<b>V2X</b>	: Véhicule à Hybride
<b>IOT</b>	: Internet Of Thing
<b>PDR</b>	: Packet delivery Ratio
<b>DSRC</b>	: Dedicate Short range communication
<b>STI</b>	: Intelligent transportation system
<b>5G</b>	: Cinquième génération
<b>H4G</b>	: Quatrième génération

<b>LTE</b>	: Long term evolution
<b>SDVN</b>	Software defines vehicular network
<b>QoS</b>	: Quality of service
<b>SD-IoV</b>	: Software defines internet of vehicles
<b>CLI</b>	: Command line interface
<b>TCP</b>	: Transmission Control Protocol
<b>UDP</b>	: User Datagram Protocol