



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU-BEKR BELKAID – TLEMCCEN

THÈSE LMD

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT

Spécialité : *Informatique Distribuée et Réseaux*

Par :

GAOUAR Nihal

Sur le thème

Étude de solutions Cloud pour les communications véhiculaires

Soutenue publiquement le 28 Octobre 2023 à Tlemcen devant le jury composé de :

Pr BENAMAR Abdelkrim	Professeur	Université de Tlemcen	Président
Pr LEHSAINI Mohamed	Professeur	Université de Tlemcen	Directeur de thèse
Pr FOUCHAL Hacène	Professeur	Université Champagne-Ardenne	Co-directeur de thèse
Dr KHALFI Mohammed Fethi	MCA	Université de Sidi Bel Abbes	Examineur
Dr MERAD BOUDIA Omar Rafik	MCA	Université d'Oran 1	Examineur
Dr AMRAOUI Asma	MCA	Université de Tlemcen	Examineur

*Laboratoire Systèmes et Technologies de l'Information et de la Communication (STIC)
BP 119, 13000 Tlemcen - Algérie*

Remerciements

Je tiens, avant toute chose, à remercier ALLAH TOUT PUISSANT de m'avoir donné la force, le courage, la volonté et la patience afin de réaliser ce travail de thèse.

J'exprime ma profonde gratitude à mon directeur de thèse, Monsieur LEHSAINI Mohamed pour son esprit scientifique, sa pédagogie, sa disponibilité, sa patience, ses conseils et ses idées qui m'ont permis de mener à bien cette thèse.

Mes grands remerciements à mon co-directeur de thèse Monsieur FOUCHAL Hacène, Professeur à l'université Champagne-Ardenne pour son soutien et son aide durant son passage au laboratoire STIC.

Mes remerciements les plus sincères sont adressés à Monsieur BENAMAR Abdelkrim de me faire l'honneur de s'intéresser à ce travail et d'avoir présidé le jury

J'exprime ensuite ma plus profonde gratitude à Monsieur KHALFI Mohammed Fethi, Monsieur MERAD BOUDIA Omar Rafik et Mme AMRAOUI Asma qui m'ont fait l'honneur d'accepter d'être les examinateurs de cette thèse.

Ce travail a été réalisé au sein du laboratoire "Systèmes et Technologie de l'Information et de la Communication (STIC)" de l'université de Tlemcen. Je souhaite également remercier les membres de ce laboratoire. A commencer par les responsables de ce laboratoire Monsieur LEHSAINI et Monsieur FEHAM de m'avoir accueilli au sein de ce laboratoire pour leur disponibilité, précieux conseils et gentillesse. A Monsieur NEBBOU Tawfiq pour son aide, son encouragement, sa disponibilité et ses conseils pertinents. Enfin, à tous mes collègues du laboratoire à qui je souhaite beaucoup d'épanouissement dans leurs travaux.

Enfin, J'adresse mes sincères remerciements à toutes les personnes qui me sont très chères qui m'ont encouragée et soutenue durant toutes ces années de réalisation de cette thèse.

A vous tous, Merci !

Dédicace

Je dédie ce travail

A mes très chers parents qui m'ont poussée, soutenue et épaulée pour que je puisse
atteindre mes objectifs,

A mon très cher mari qui a cru en moi et qui a toujours été là quand j'en avais besoin,

A mes petites princesses Ferial et Meryam qui sont ma raison de vivre,

A ma soeur, mon frère, ma grand mère, et ma belle famille qui m'ont toujours
encouragée et soutenue.

En dernier, j'adresse ma plus grande dédicace à mon très cher grand père qui attendait
avec impatience de me voir enfin docteur, qui n'a jamais cessé de m'encourager et de
prier pour moi. Malheureusement il a rejoint un monde meilleur avant que je ne puisse
aller jusqu'au bout de ce travail de thèse. A toi qui vivras toujours dans mon cœur
reçois cette dédicace symbole de ma reconnaissance et de mon amour éternel.

Table des matières

Liste des figures	v
Liste des tableaux	vi
Introduction Générale	1
1 Les réseaux ad hoc Cloud véhiculaires : Généralités et Concepts	4
1.1 Introduction	4
1.2 Les réseaux ad hoc véhiculaires	5
1.2.1 Définition des réseaux ad hoc véhiculaires	5
1.2.2 Architecture de communication dans les VANETS	6
a) Architecture de communication véhicule à véhicule	6
b) Architecture de communication véhicule à infrastructure	7
c) Architecture de communication hybride	7
1.2.3 Services offerts par les VANETS	7
a) Services de sécurité	8
b) Services de surveillance urbaine	8
c) Services de management de trafic routier	8
1.2.4 Principaux défis des VANETS	8
1.2.5 Protocoles et normes dédiés aux VANETS	9
a) Communications dédiées à courte portée (DSRC)	10
b) Accès sans fil en environnements véhiculaires (WAVE)	10
c) ETSI-ITS G5	11
1.2.6 Projets VANETS	11
a) Projets VANET en Europe	12
b) Projets VANETS aux États-Unis	13
c) Projets VANET au Japan	13
1.3 Le Cloud computing	14
1.3.1 Définition du Cloud computing	14
1.3.2 Le Cloud computing mobile	14
1.3.3 Caractéristiques du Cloud computing	14
1.3.4 Modèles de déploiement du Cloud computing	15
1.3.5 Modèles de service du Cloud computing	16
1.4 Le Fog computing	17
1.4.1 Définition du Fog computing	17
1.4.2 Caractéristiques du Fog computing	18
1.4.3 Architecture du Fog Computing	19

1.4.4	Applications du Fog computing	20
1.5	Le Cloud et Fog véhiculaire	22
1.5.1	Le Cloud Véhiculaire	22
1.5.2	Architecture du Cloud Véhiculaire	24
	a) Le niveau interne du véhicule	24
	b) Le niveau de communication	24
	c) Le niveau Cloud	25
1.5.3	Le Fog Véhiculaire	25
1.5.4	Architecture du Fog véhiculaire	26
	a) La couche génération de données	27
	b) La couche Fog	27
	c) La couche Cloud	27
1.5.5	Comparaison entre le Cloud véhiculaire et le Fog véhiculaire	27
1.5.6	Services offerts par le Cloud/Fog computing dans les réseaux véhiculaires	28
1.6	Conclusion	29
2	État de l'art sur la dissémination de données dans les VANETs via le Cloud/Fog computing	31
2.1	Introduction	31
2.2	État de l'art sur la dissémination de données dans les VANETs	31
2.3	Classification des schémas de dissémination de données dans les VANETs	33
2.3.1	Schémas de dissémination de données dans les VANETs	33
	a) Taxonomie des méthodes de dissémination de données dans les VANETs	33
	b) Inconvénients de la dissémination de données dans les VANETs	34
2.3.2	Schémas de dissémination de données dans les réseaux Cloud véhiculaires	35
2.3.3	Schémas de dissémination de données dans les réseaux Fog véhiculaires	35
2.3.4	Récapitulatif de la classification ci-dessus	36
2.4	Architecture de communication dans les réseaux Cloud/Fog véhiculaires	37
2.4.1	Architecture générale des réseaux Cloud/Fog véhiculaires	37
2.4.2	Architecture de communication des réseaux Cloud/Fog véhiculaires	38
	a) Architecture de Masip-Bruin et al.	38
	b) Architecture de Bitam et al.	39
	c) Architecture de Meneguette [1]	39
	d) Architecture de Santa et al. [2]	40
	e) Architecture d'Agarwal et al.	41
	f) Architecture de Yu et al.	41
	g) Architecture de Huang et al.	42
	h) Architecture de Pasha et al.	42
	i) Architecture de Hou et al. [3]	43
2.5	Solutions de routage dans les VANETs basées sur le Cloud et le Fog computing	44
2.5.1	Solutions de routage basées sur le Cloud computing	44
2.5.2	Solutions de routage basées sur le Fog computing	53

2.5.3	Récapitulatif des solutions de routage dans le Cloud/Fog véhiculaire	58
2.6	Conclusion	60
3	Déploiement efficace des RSUs dans les VANETs basé sur les méthodes d'optimisation	61
3.1	Introduction	61
3.2	Contexte	62
3.3	Méthodes d'optimisation combinatoire	64
3.3.1	Optimisation combinatoire	64
3.3.2	Méthodes de résolution des problèmes d'optimisation combinatoire	65
a)	Méthodes exactes	65
b)	Méthodes approchées	66
3.3.3	Métaheuristiques : recuit simulé et algorithmes génétiques	67
a)	Recuit simulé (Simulated Annealing)	67
b)	Algorithmes génétiques	68
3.4	Approches proposées pour un déploiement efficace des RSUs dans les VANETs	70
3.4.1	Approches basées sur les algorithmes génétiques	74
3.4.2	Approches basées sur le recuit simulé	75
3.5	Évaluation des performances	79
3.5.1	Environnement de simulations	80
3.5.2	Résultats des simulations	81
a)	Taux de couverture	82
b)	Taux de chevauchement des zones couvertes	82
c)	Taux de livraison des paquets (PDR)	84
d)	Délai de bout en bout	85
3.6	Conclusion	86
4	Gestion du trafic routier dans les VANETs basée sur le cloud computing	87
4.1	Introduction	87
4.2	Contexte	87
4.3	Travaux connexes	89
4.4	Méthodes de contrôle des feux de circulation	91
4.4.1	La méthode webster	92
4.4.2	La méthode GreenWave	93
4.4.3	La méthode Maxband	93
4.4.4	La méthode Max-pressure	94
4.4.5	La méthode Actuated control	95
4.4.6	La méthode Self-Organizing Traffic Lights (SOTL)	95
4.4.7	La méthode Split Cycle Offset Optimization Technique (SCOOT)	95
4.4.8	La méthode Sydney Coordinated Adaptive Traffic (SCAT)	96
4.4.9	La méthode Optimized Policies for Adaptive Control (OPAC)	97
4.4.10	Comparaison entre différentes méthodes de contrôle des feux de circulation	97
4.5	Contribution : Le protocole CCITL	97
4.5.1	Architecture du protocole CCITL	98
4.5.2	Fonctionnement du protocole CCITL	99
4.5.3	Environnement de travail	103

TABLE DES MATIÈRES

4.5.4	Evaluation des performances et analyse des résultats	103
a)	Temps d'attente moyen des véhicules dans le réseau routier	104
b)	Temps d'attente moyen dans les différents segments de la route	105
c)	Nombre de véhicules dans les différents segments de la route	106
d)	Récapitulatif des résultats de la simulation	109
4.6	Conclusion	109
	Conclusion Générale	110
	Bibliographie	113

Liste des figures

1.1	Exemple de réseau ad hoc véhiculaire	5
1.2	Modes de communication dans les VANETs [4]	7
1.3	Modèles de déploiement du Cloud [5]	15
1.4	Modèles de service du Cloud computing [6]	16
1.5	Illustration de Fog computing [7]	19
1.6	Architecture générale du Fog computing [7]	21
1.7	Exemple de Cloud véhiculaire	22
1.8	VCC architecture [8]	24
1.9	Exemple de Fog véhiculaire	26
1.10	Architecture du VFC [9]	26
2.1	Exemple d'architecture Cloud/Fog véhiculaire [10]	38
2.2	Solutions de routage dans VCC/VFC	59
3.1	Classification des principales méthodes d'optimisation combinatoire	66
3.2	Modélisation de la problématique	72
3.3	Zones couvertes par les RSUs déployées	72
3.4	Zones de chevauchement : zones couvertes par deux RSUs ou plus	73
3.5	Division des segments en sous segments de 1 mètre	73
3.6	Étapes de traitement par GA-Basic et GA-Improved	75
3.7	Zone du réseau (Abou Tachfine, Tlemcen)	81
3.8	Taux de couverture dans GA-Basic, GA-Improved, SA-Basic et SA-Improved	83
3.9	Taux de chevauchement dans GA-Basic, GA-Improved, SA-Basic et SA-Improved	83
3.10	Taux de livraison de paquets vs. Nombre de RSUs	84
3.11	Délai de bout en bout vs. Nombre de RSUs	86
4.1	Architecture du protocole CCITL	99
4.2	Carte routière du réseau véhiculaire implémenté	104
4.3	Evolution du temps de séjour des véhicules après application du protocole CCITL	105
4.4	Temps de séjour des véhicules dans le réseau avec CCITL et AGTLS-P	106
4.5	Temps de séjour des véhicules dans différents segments avec CCITL et AGTLS-P	107
4.6	Nombre de véhicules traversant différents segments avec CCITL et AGTLS-P	108
4.7	Nombre de véhicules ayant transité à travers les segments gneE18 et gnE19 CCITL et AGTLS-P	108

Liste des tableaux

1.1	Comparaison entre le Cloud véhiculaire et le Fog véhiculaire	28
2.1	Classification des schémas de dissémination de données dans les communi- cations véhiculaires	37
2.2	Récapitulatif des caractéristiques des architectures VCC et VFC	43
2.3	Techniques de dissemination de données dans les réseaux VCC et VFC . . .	60
3.1	Paramètres de simulation	81
3.2	Taux de couverture dans GA-Basic, GA-Improved, SA-Basic et SA-Improved	82
3.3	Taux de chevauchement dans GA-Basic, GA-Improved, SA-Basic et SA- Improved	84
3.4	PDR dans GA-Basic, GA-Improved, SA-Basic and SA-Improved	85
4.1	Récapitulatif des méthodes de contrôle du trafic	98
4.2	Paramètres de simulation pour l'évaluation du protocole CCITL	103
4.3	Récapitulatif des résultats de simulation	109

Glossaire

ACP Automatic Cell Planning. 63

AE Authorized Entity. 50

AES Advanced Encryption Standard. 46

AFP Automatic Frequency Planning. 63

AODV Ad hoc On Demand Distance Vector. 34

AP Access Point. 45

ARTSYS ARTerial network SYStem. 90

ASV Advanced Safety Vehicle. 13

ATL Arterial Traffic Lights. 89

AU Application Unit. 6

BS Base Station. 48

BSM Basic Security Message. 45

BW BandWidth. 50

C2C-CC Car to Car Communication Consortium. 11

CATLS Context-Aware Traffic Light Scheduling. 91

CCH Control CHannel. 11

CCITL Cloud Computing based Intelligent Traffic Light Protocol. 87, 110

Coopers CO-OPERative SystEMs for Intelligent Road Safety. 12

CPU Central Processing Unit. 53

CV Cooperating Vehicle. 47

- DaaS** Data as a Service. 51
- DFCV** Dynamic Fog for Connected Vehicles. 53
- DHT** Distributed Hash Table. 46
- DM** Decision Module. 49
- DSDV** Destination-Sequenced Distance-Vector Routing. 34
- DSR** Dynamic Source Routing. 34
- DSRC** Dedicated Short Range Communications. 10, 50
- DTN** Delay Tolerance Network. 36
- EBI** Event Building Interface. 52
- EEMSFV** Energy Efficient Multicast routing protocol based on SDN and FC for VANETs. 55
- ETSI** European Telecommunications Standards Institute. 40
- EVITA** E-Safety Vehicle Intrusion Protected Applications. 12
- F2C** Fog-to-Cloud. 38
- FNs** Fog Nodes. 18
- FSR** Fisheye State Routing. 34
- GA** Genetic Algorithm. 89
- GAs** Genetic Algorithms. 68
- GeoNet** Geographic addressing and routing for vehicular communications. 12
- GPS** Global Positioning System. 41
- GSRP** Greedy Perimeter Stateless Routing. 34
- HARP** Hybrid ad hoc routing protocol. 34
- HC** Hybrid Cloud. 49
- HITUL** Holistic Intelligent for Traffic Urban Lights. 90
- HNN** Hopfield Neural Network. 90
- HSPA** High-Speed Packet Access. 40
- IaaS** Infrastructure as a Service. 17

- ICANP** Intelligent Context-Aware Negotiation Protocol. 89
- ICAPP** Intelligent Context-Aware Priority Protocol. 89
- IGR** Improved Geographic Routing. 54
- IoT** Internet of Things. 4, 88
- IoV** Internet of Vehicles. 88
- IPv4** Internet Protocol version 4. 103
- IPv6** Internet Protocol version 6. 40, 103
- ISO** International Organization for Standardization. 40
- ITLC** Intelligent Traffic Light Controlling. 89
- ITM** Intelligent Truck Monitoring. 53
- ITS** Intelligent Transportation Systems. 1, 4, 47, 88
- IVI** Intelligent Vehicle Initiative. 13
- JARI** Japan Automobile Research Institute. 13
- LBMM** Load Balancing Max-Min. 48
- LTE** Long Term Evolution. 6
- MAB** Multi-Armed Bandit. 45
- MAC** Medium access control. 9
- MANETs** Mobile Adhoc NETWORKs. 4
- MARTA** Mobility and Automotion through Advanced Transport Networks. 12
- MPSO-CO** Modified Particle Swarm Optimization-COnstrained. 48
- MSN** Mobile Service Node. 49
- NaaS** Network as a Service. 50
- NOLIP** Normalized OverLoad. 44
- NoW** Network On Wheels. 12
- NS-3** Network Simulator 3. 46
- OBU** On-Board Unit. 5
- OLSR** Optimized Link State Routing Protocol. 34

- OPAC** Optimized Policies for Adaptive Control. iii, 97
- OSI** Open Systems Interconnection. 10
- PaaS** Plateform as a Service. 17
- PDR** Packet Delivery Ratio. 44, 63
- PGB** Preferred Group Broadcasting. 34
- PRAODV** PRdiction based Ad hoc On Demand Distance Vector. 34
- PSDS** Proficient Security over Distributed Storage. 33
- PSO-CO** Particle Swarm Optimization-COnstrained. 48
- QoS** Quality of Service. 39
- RI-REP** Routing Information REPLY. 44
- RI-REQ** Routing Information REQuest. 44
- RIVER** Reliable Inter-VEhicular Routing. 44
- ROI** Region Of Interest. 47
- RREP** Route REPLY. 44
- RREQ** Route REQuest. 44
- RSC** RoadSide Cloud. 45
- RSU** RoadSide Unit. 5
- RSUC** Road Side Unit Controller. 57
- RTA** Roads and Traffic Authority. 96
- SA** Simulated Annealing. 67
- SaaS** Software as a Service. 16
- SCAT** Sydney Coordinated Adaptive Traffic. iii, 96
- SCOOT** Split Cycle Offset Optimization Technique. iii, 95
- SDCFN** Software-Defined Cloud/Fog Networking. 48
- SDN** Software Defined Network. 48
- SDNC** Software Defined Network Controller. 57
- SEVECOM** SEcure VEhicle COmmunication. 13

- SLA** Service Level Agreement. 51
- SOTL** Self-Organizing Traffic Lights. iii, 95
- STMP** Smart Traffic Management Protocol. 90
- SUMO** Simulation of Urban MObility. 47, 80, 87
- TA** Trusted Authority. 45
- TBRPF** Topology Broadcast based on Reverse-Path Forwarding. 34
- TCP** Transmission Control Protocol. 103
- TLS** Traffic Light System. 103
- TMC** Traffic Management Center. 45
- ToA** Time of Arrival. 41
- TORA** Temporally Ordered Routing Algorithm. 34
- TSLG** Time-Space Link Graph. 49
- TTL** Time To Life. 100
- UDP** User Datagram Protocol. 103
- V2I** Vehicle To Infrastructure. 7
- V2V** Vehicle To Vehicle. 6
- VANETs** Vehicular Adhoc NETworks. 2, 4, 88
- VC** Vehicular Cloud. 31
- VCC** Vehicular Cloud Computing. 22, 41
- VCP** Virtual Cord Protocol. 46
- VCPS** Vehicular Cyber-Physical System. 50
- VCR** Vehicular Cloud for Road side scenarios. 51
- VF** Vehicular Fog. 31
- VFC** Vehicular Fog Computing. 25, 43
- VII** Vehicle Infrastructure Integration. 13
- VSC** Vehicle Safety Communication. 13
- VSC-2** Vehicle Safety Communication-2. 13

VuC Vehicle using Cloud. 49

WAVE Wireless Access in Vehicular Environments. 10

WIF Waypoint Information Framework. 49

WiSafeCar Wireless traffic Safety network between Cars. 12

WLAN Wireless Local Area Network. 54

WRP Wireless Routing Protocol. 34

ZRP Zone Routing Protocol. 34

Résumé

Cette thèse traite le problème de la dissémination de données dans les réseaux véhiculaires (VANETs). Cette dissémination se fait soit de véhicule à véhicule, soit de véhicule à diverses infrastructures du réseau véhiculaire telles que les RSUs, les serveurs Cloud, ... etc. Motivés par les exigences des communications véhiculaires, nous avons visé à travers cette thèse d'améliorer les performances des VANETs via l'utilisation du Cloud computing et nous avons proposé de nouvelles approches en guise de solutions pour deux problèmes différents. La première vise à garantir une couverture réseaux et des performances de routage acceptables dans un réseau routier tout en minimisant le nombre de RSUs déployés au niveau des intersections de la route, tandis que la deuxième se concentre sur le problème de la congestion routière et du délai pour atteindre la destination et représente un nouveau protocole de gestion du trafic routier qui se base sur le Cloud computing.

Dans la première contribution, nous avons proposé une solution de déploiement efficace des RSUs dans les VANETs basée sur les versions améliorées des métaheuristiques algorithmes génétiques et recuit simulé. Les résultats obtenus étaient prometteurs puisque le nombre de RSUs déployées au niveau des intersections a été grandement diminué tout en gardant de bonnes performances de routage et un bon taux de couverture dans le réseau. Dans la deuxième contribution, nous avons proposé le protocole CCITL; une technique de gestion du trafic routier intelligente qui se base sur les feux de circulation tricolores et le Cloud computing pour fluidifier le trafic et minimiser les délais d'attente. Dans cette solution de la gestion de trafic, CCITL implique le Cloud conventionnel comme fournisseur de services dont les serveurs sont dédiés aux calculs de la densité et du temps d'attente des véhicules dans chaque voie de la route. Les résultats obtenus nous ont permis de constater que le protocole proposé constitue une solution fiable puisque le trafic est devenu plus fluide, les embouteillages ont grandement diminué, et le temps d'attente des véhicules dans plusieurs segments a été bien réduit.

Mots clés : Algorithmes génétiques, Cloud Computing, Dissémination de données, Déploiement des RSUs, Gestion du trafic routier, Recuit simulé, VANETs.

Abstract

This thesis addresses the problem of data dissemination in vehicular networks (VANETs). This dissemination occurs either from vehicle to vehicle, or from vehicle to various vehicular network infrastructures such as RSUs, Cloud servers, etc. Motivated by the requirements of vehicular communications, we aimed in this thesis to improve the performance of VANETs through the use of Cloud computing and proposed new approaches as solutions to two different problems. The first aims for ensuring acceptable network coverage and routing performance in a road network while minimizing the number of RSUs deployed at road intersections, while the second focuses on the problem of road congestion and delay in reaching the destination and represents a new road traffic management protocol based on Cloud computing.

In the first contribution, we proposed a solution for the efficient deployment of RSUs in VANETs based on improved versions of the genetic algorithm and simulated annealing metaheuristics. The results obtained were promising, since the number of RSUs deployed at intersections was greatly reduced while maintaining good routing performance and a good coverage rate in the network. In the second contribution, we proposed the CCITL protocol, an intelligent road traffic management technique that uses traffic lights and cloud computing to smooth traffic flow and minimize waiting times. In this traffic management solution, CCITL involves the conventional Cloud as a service provider whose servers are dedicated to calculating the density and waiting time of vehicles in each lane of the road. The results obtained show that the proposed protocol is a reliable solution, since traffic has become more fluid, traffic jams have been greatly reduced, and the waiting time for vehicles in several segments has been greatly reduced.

Keywords: Genetic algorithms, Cloud Computing, Data dissemination, RSU deployment, Road traffic management, Simulated annealing, VANETs.

ملخص

تتناول هذه الأطروحة مشكلة نشر البيانات في شبكات المركبات (VANETs) حيث يتم هذا النشر إما من مركبة إلى مركبة، أو من مركبة إلى البنية التحتية المختلفة لشبكة المركبات مثل RSUs، الخوادم السحابية،... إلخ. بدافع من متطلبات اتصالات المركبات. استهدفنا من خلال هذه الأطروحة تحسين أداء VANETs من خلال استخدام الحوسبة السحابية واقتراحنا أساليب جديدة كحلول لمشكلتين مختلفتين. يهدف الأول إلى ضمان تغطية الشبكة المقبولة وأداء التوجيه في شبكة الطرق مع تقليل عدد وحدات RSUs المنتشرة عند تقاطعات الطرق، بينما يركز الثاني على مشكلة الازدحام والتأخير على الطرق للوصول إلى الوجهة ويمثل إدارة جديدة لحركة المرور على بروتوكول قائم على الحوسبة السحابية.

في المساهمة الأولى، اقترحنا حلاً للنشر الفعال لـ RSUs في VANETs استناداً إلى الإصدارات المحسنة من الخوارزميات الجينية والتلدين المحاكي. كانت النتائج التي تم الحصول عليها واعدة حيث تم تقليل عدد وحدات RSUs المنتشرة في التقاطعات بشكل كبير مع الحفاظ على أداء التوجيه الجيد ومعدل التغطية الجيد في الشبكة. في المساهمة الثانية، اقترحنا بروتوكول CCITL؛ تقنية ذكية لإدارة حركة المرور على الطرق تعتمد على إشارات المرور والحوسبة السحابية لتبسيط حركة المرور وتقليل أوقات الانتظار. في حل إدارة حركة المرور هذا، يتضمن CCITL السحابة التقليدية كمزود خدمة خوادمه مخصصة لحساب كثافة المركبات ووقت انتظارها في كل ممر من الطريق. سمحت لنا النتائج التي تم الحصول عليها أن البروتوكول المقترح هو حل موثوق لأن حركة المرور أصبحت أكثر مرونة، وانخفضت الاختناقات المرورية بشكل كبير، كما تم تقليل وقت انتظار المركبات في عدة قطاعات.

الكلمات المفتاحية: الخوارزميات الجينية، الحوسبة السحابية، نشر البيانات، نشر RSUs، إدارة حركة المرور على الطرق، التلدين المحاكي، VANETs

Introduction Générale

Introduction Générale

Contexte

L'évolution constante des réseaux sans fil avec internet a conduit à l'émergence de l'internet des objets. En effet, les objets connectés ont envahi notre vie quotidienne et deviennent omniprésents partout où nous sommes. Parmi les objets connectés les plus utilisés se trouvent les véhicules. Effectivement, de nos jours, le véhicule le plus basique est doté d'options et est de plus en plus sophistiqué et les véhicules haut de gamme les plus sophistiqués sont blindés de technologies, de ressources informatiques et de capteurs. Ces véhicules intelligents permettent de communiquer avec les infrastructures de la route et de mettre en oeuvre des applications de sécurité routière telles que l'évitement de collisions, la prévention de travaux,...etc, des applications de confort du conducteur telles que l'accès au contenu multimédia en ligne, le péage automatique des autoroutes, la réservation d'hôtel ou de restaurant,...etc, et des applications des systèmes de transport intelligent "Intelligent Transportation Systems (ITS)" [11], à savoir, la gestion du trafic routier, la proposition d'un détour ou d'une route de secours...etc.

Par ailleurs, les véhicules sont au coeur de notre société, nous les utilisons tous les jours et en toutes circonstances ; pour se rendre au travail, pour voyager, pour partir en vacances ou encore pour sortir en balade. De ce fait, les ressources des véhicules sont constamment sollicitées afin de profiter pleinement des avantages que nous offrent les VANETs en termes d'applications et de communications entre véhicules et avec les infrastructures de la route.

Afin de fonctionner pleinement, les réseaux véhiculaires font de plus en plus appel au Cloud computing pour s'alimenter en ressources ce qui leur permet de communiquer efficacement entre eux et avec les infrastructures routière et de servir un maximum d'applications en temps réel constituant ainsi le paradigme de Cloud véhiculaire ou VCC en abrégé en référence à Vehicular Cloud Computing. Par ailleurs, le Cloud véhiculaire permet également aux véhicules de se partager leurs ressources telles que la capacité de calcul ou de stockage avec d'autres véhicules ou de profiter de celles des autres. En effet, les véhicules dont les ressources ne sont pas exploitées peuvent louer celles-ci à d'autres véhicules qui en ont besoin afin de satisfaire leurs besoins. Ceci évite que des ressources soient gaspillées ou passives et que des véhicules soient en sous-ressources ce qui engendrerait des coupures dans la communication ou des défaillances dans les applications.

Problématique et contributions

Dans le monde des réseaux véhiculaires, plusieurs défis sont posés en matière de communication, de techniques de dissémination de données, de sécurité, et de ressources informatiques suffisantes pour satisfaire les besoins de diverses applications potentiellement mises en oeuvre via les ordinateurs de bord des véhicules. En effet, les réseaux "Vehicular Adhoc NETWORKS (VANETs)" sont des réseaux à faible bande passante dont le nombre de véhicules et de capteurs pose problème car plus le nombre de paquets échangés est grand plus le nombre de collisions augmente, ce qui cause des retards dans les délais de délivrance des messages. Ceci, en plus des coupures de connexion fréquentes dues à la haute mobilité des noeuds. De plus, le besoin croissant des véhicules en terme de ressources de calcul et de stockage afin de servir les diverses applications qui ont émergé avec les VANETs représente un défi majeur car les demandes des utilisateurs ne cessent de croître avec toutes les technologies de la vie moderne. L'utilisation du Cloud Computing dans les réseaux VANETs s'impose alors comme la solution appropriée dans ce genre de défis. Effectivement, dans le monde des besoins en ressources, ou encore des ressources sous-exploitées, le Cloud est la solution. Il permet d'alimenter diverses technologies en ressources ainsi que de louer les ressources sous-exploitées à d'autres utilisateurs qui en ont besoin. Le Cloud est par ailleurs entravé d'ondes et de technologies de communication, ce qui intrinsèquement facilite la communication et l'accès à l'information par les véhicules.

Motivés par ces considérations, nous proposons dans cette thèse de nouvelles approches en guise de solution pour deux problèmes différents. La première vise à garantir une couverture réseaux et des performances de routage acceptables dans un réseau routier tout en minimisant le nombre de RSUs déployés au niveau des intersections de la route, tandis que la deuxième se concentre sur le problème de la congestion routière et du délai pour atteindre la destination et représente un nouveau protocole de gestion du trafic routier qui se base sur le Cloud computing pour appuyer les ressources des véhicules et propose un algorithme efficace pour planifier la circulation routière tout en évitant les embouteillages et les délais pour atteindre la destination.

Pour la première contribution nous avons tenté de trouver la combinaison parfaite entre le nombre de RSUs à déployer, le taux de couverture réseau, et les performances de routage acceptables. Dans cette optique, nous nous sommes basés sur les techniques d'optimisation combinatoire en utilisant deux types de métaheuristiques d'optimisation ; le premier est les algorithmes génétiques [12], et le second est le recuit simulé [13].

La deuxième contribution proposée, offre une solution efficace pour gérer le trafic routier. Elle se base sur les feux de circulation tricolores intelligents et le Cloud computing. Les véhicules du réseau se chargent de collecter les informations pertinentes relatives à l'état de la route et les transmettent à la RSU la plus proche dans leur rayon de communication. Cette dernière se charge de transmettre ces informations au Cloud conventionnel, qui lui, est en charge de faire les traitements adéquats afin de trouver la combinaison optimale des feux tricolores avec la durée adéquate afin d'avoir une route fluide qui permet aux conducteurs de conduire tranquillement sans se farcir les embouteillages, et d'arrivée à destination en temps optimal.

Organisation de la thèse

Pour ce faire, cette thèse est organisée en quatre chapitres :

- Dans le premier chapitre, nous présentons les réseaux ah-hoc véhiculaires, leur architecture, leurs modes de communication, ainsi que les services qu'ils offrent. Ensuite, nous abordons le paradigme du Cloud computing et du Fog computing avec leurs propriétés respectives. Enfin, nous clôturons ce chapitre par la présentation du Cloud computing véhiculaire et du Fog computing véhiculaire en abordant leurs architectures et leurs apports dans les réseaux VANETs.
- Le deuxième chapitre est un état de l'art sur les techniques de dissémination dans les réseaux véhiculaires en utilisant le Cloud et le Fog computing. Cette étude nous a permis de tirer profit des atouts du Cloud computing ainsi que des avantages qu'apportent ce dernier dans les réseaux véhiculaires et de proposer de nouvelles solutions pour ce type de réseaux.
- Le troisième chapitre présente notre première contribution qui est une solution de déploiement efficace des RSUs dans les réseaux véhiculaires basé sur les méthodes d'optimisation. Dans cette contribution, après la détermination d'un nombre initial de RSUs au niveau des intersections d'un réseau routier, nous nous sommes basés sur les techniques d'optimisation des problèmes polynomiaux, à savoir ; les algorithmes génétiques et le recuit simulé pour proposer une solution optimale qui définit un nombre minimal de RSUs à déployer au niveau des emplacements stratégiques tout en garantissant un taux de couverture et des performances de routage acceptables.
- Le quatrième chapitre expose notre seconde contribution qui propose une technique de gestion du trafic routier intelligente qui se base sur les feux de circulation tricolores et le Cloud computing. Dans cette solution de gestion de trafic, le Cloud conventionnel dispose d'une vue globale du réseau routier grâce aux informations relatives à l'état de la route reçues par le biais des RSUs, et c'est lui qui effectue tous les traitements et calculs afin de définir la meilleure combinaison des feux de circulation ainsi qu'une durée optimale de chaque phase de feu. Ceci permet de libérer les routes des embouteillages et de limiter les retards.

Finalement, une conclusion générale clôture ce manuscrit et rappelle les principales contributions tout au long de ce travail de thèse. Elle présente également les perspectives sous-jacentes aux différentes problématiques abordées dans cette thèse. Les travaux contenus dans ce document ont été publiés dans plusieurs journaux et conférences.

**Les réseaux ad hoc Cloud
véhiculaires : Généralités et
Concepts**

Les réseaux ad hoc Cloud véhiculaires : Généralités et Concepts

1.1 Introduction

Les réseaux mobiles ad hoc (Mobile Adhoc NETWORKS (MANETs)) avec l'émergence des véhicules intelligents ont donné naissance aux réseaux ad hoc véhiculaires (VANETs). Aujourd'hui le domaine des VANETs représente une partie intégrante de l'internet des objets "Internet of Things (IoT)" et a reçu beaucoup d'attention de la part des chercheurs au cours des dernières années car il permet des solutions attrayantes dans des domaines importants du système de transport tels que la sécurité des véhicules et de la route, l'efficacité du trafic et le système de transport intelligent (ITS) [11]. La conduite classique s'est alors transformée en une conduite sûre dans un environnement sain. Les réseaux de véhicules intelligents améliorent l'efficacité des transports et la sécurité en fournissant de l'info-divertissement.

Étant intelligents, et avec l'émergence d'applications qui ont révolutionné notre quotidien, ceci en plus des exigences accrues des conducteurs, les véhicules sont devenus de plus en plus gourmands en capacité d'analyse, en capacité de traitement, capacité de calcul, et espace de stockage. Malgré tous les efforts déployés pour répondre à toutes ces exigences, les réseaux véhiculaires ont de nombreux inconvénients tels que les coûts élevés engendrés par la communication entre les véhicules en raison de la grande mobilité des noeuds (véhicules). Cela a rapidement conduit à un besoin important d'augmenter les ressources des véhicules afin de soutenir et de servir toutes les applications des conducteurs pour leur confort, leur sécurité et bien plus encore. Par conséquent, le concept du Cloud computing véhiculaire (VCC) a émergé. L'usage du Cloud computing dans les réseaux véhiculaires permet aux véhicules d'exploiter les avantages du Cloud computing afin de répondre aux exigences de l'utilisateur.

Ce chapitre est consacré à la présentation des réseaux véhiculaires ainsi qu'à la présentation du Cloud computing avec les principales caractéristiques de chacun des deux domaines. Nous mettons l'accent sur les interactions entre ces deux domaines VANETs et Cloud Computing ainsi que sur l'apport du Cloud dans les réseaux ad-hoc véhiculaires.

1.2 Les réseaux ad hoc véhiculaires

Dans cette section, nous présentons les réseaux véhiculaires, leurs caractéristiques, les architectures et les normes de communication supportées par ces réseaux ainsi que leurs défis.

1.2.1 Définition des réseaux ad hoc véhiculaires

Les réseaux ad-hoc véhiculaires connus sous le nom de Vehicular Ad hoc NETWORKS (VANETs) sont issues des réseaux ad-hoc mobiles "Mobile Adhoc NETWORKS (MANETs)". L'idée a été d'impliquer les MANETs dans la communication dans les rues et les autoroutes. Un réseau VANET est un réseau mobile dans lequel les véhicules représentent les noeuds du réseau. Parmi les composants d'un VANET, on trouve des unités de bord de route (RoadSide Unit (RSU)) qui sont des infrastructures potentielles placées à des endroits stratégiques sur les routes et agissant comme un dispositif de connexion pour les véhicules en déplacement. La Figure 1.1 montre un exemple de VANET.

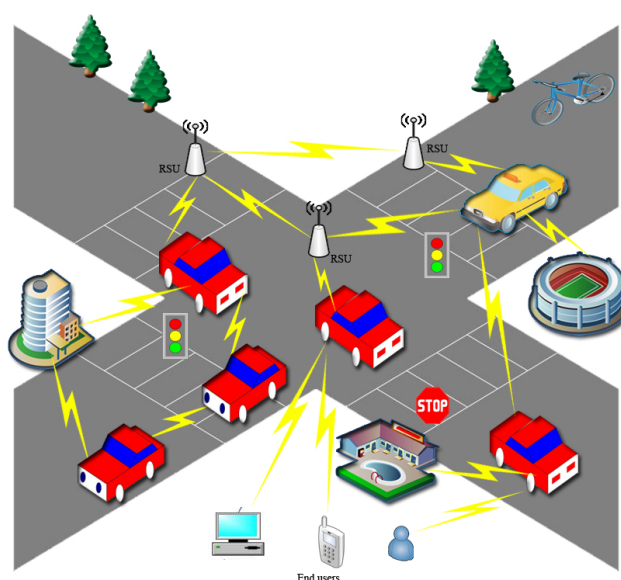


FIGURE 1.1 – Exemple de réseau ad hoc véhiculaire

Les réseaux VANETs sont caractérisés par la haute mobilité de leurs noeuds ainsi que le changement continu et rapide de leurs topologies. De plus, dans le système de transport intelligent, les VANETs sont l'une des technologies les plus émergentes qui permet de fournir des réseaux sans fil à l'aide de véhicules mobiles. Avec l'aide des VANETs, l'ITS fournit de nombreuses applications pour la sécurité des véhicules qui incluent les informations sur la circulation ou un accident qui s'est produit sur le chemin, ou encore la localisation les postes de police ... etc. l'ITS fournit également des messages d'info-divertissement tels que les messages d'urgence, les publicités, ... etc. Il donne ces informations aux véhicules en mouvement par le biais des RSUs et d'unité de bord du véhicule "On-Board Unit (OBU)". Par ailleurs, comme le nombre de demandes pour chaque véhicule peut atteindre un nombre important, chaque RSU devrait pouvoir répondre à toutes les demandes qui

lui sont adressées. Ainsi, une dissémination efficace de données devient un enjeu majeur lorsque le nombre de requêtes destinées à une même RSU est élevé [14].

1.2.2 Architecture de communication dans les VANETs

Les véhicules ont besoin de support de communication pour communiquer entre eux ou avec d'autres infrastructures de la route. Celui utilisé dans les réseaux VANETs est le "Wireless Access in Vehicular Environments (WAVE)" [15]. C'est une technologie de communication qui fournit un large éventail d'informations aux conducteurs et passagers et permet d'assurer une conduite confortable et sécurisée grâce aux applications qu'elle sert et qui fournissent ce genre de service. Le système de communication se compose principalement d'unité d'application "Application Unit (AU)", d'unité embarquée (OBU) et d'unité de bord de route (RSU). C'est généralement la RSU qui héberge les applications fournissant les services, et l'OBU utilise ces services comme étant périphérique homologue. Cependant, les applications peuvent résider aussi bien dans la RSU que dans l'OBU. On appelle Fournisseur le périphérique qui héberge les applications, et celui qui les utilise est décrit comme utilisateur. Tous les véhicules sont équipés d'OBU et de capteurs qui leurs permettent de collecter et de faire des traitements des informations, puis les transmettre aux RSUs ou autres véhicules sous forme de paquets via le support de communication sans fil. Le véhicule transporte également plusieurs AUs utilisant les applications fournies en se basant sur les capacités de connexion de l'OBU. La RSU peut également accéder à Internet ou se connecter à d'autres serveurs, ce qui permet aux AUs de différents véhicules de se connecter à Internet [16, 17].

Ainsi, pour pouvoir échanger les différentes données et informations entre eux, ou avec les infrastructures de la route, les véhicules se connectent d'abord au support de connexion via n'importe quelle technologie de communication sans fil disponible comme la 3G, 4G ou "Long Term Evolution (LTE)", WiMax, ou encore via une radio de courte portée afin d'assurer un lien ou une connectivité avec le réseau. L'architecture de communication dans les VANETs diffère selon que le véhicule souhaiterait communiquer avec un autre véhicule, ou alors avec une infrastructure routière telle qu'une RSU ou encore selon que les véhicules communiquent avec les véhicules et les infrastructures à la fois, ce qui donne lieu à trois architecture de communication comme le montre la Figure 1.2.

a) Architecture de communication véhicule à véhicule

Nommée V2V en référence à "Vehicle To Vehicle (V2V)", cette architecture désigne les communications inter-véhiculaires. Ici, le mode de communication est semblable à celui des réseaux MANETs où les noeuds sont mobiles et changent constamment de localisation. La communication entre les véhicules se fait directement en mode ad hoc à l'aide des dispositifs OBU déjà installés au niveau des véhicules. Les véhicules n'ont pas besoin d'avoir recours aux infrastructures pour leur assurer une connectivité du moment où chaque véhicule se trouve dans la zone radio ou la portée du véhicule avec lequel il communique, dans le cas contraire, les véhicules communiquent via un protocole multi-sauts qui va se charger d'acheminer les paquets de bout en bout en faisant appel à des noeuds voisins intermédiaires qui séparent les deux véhicules en communication pour faire office de relais. En V2V, le support de communication utilisé se caractérise par une latence courte et un

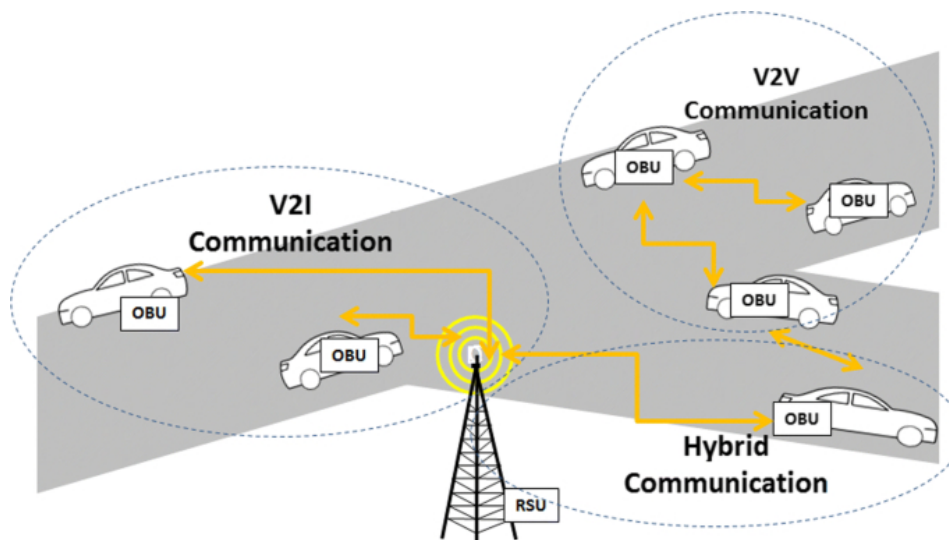


FIGURE 1.2 – Modes de communication dans les VANETs [4]

débit de transmission élevé. Cette architecture est utilisée dans différents scénarios de diffusion d'alertes (freinage d'urgence, collision, décélération, ...etc.) ou dans une conduite coopérative.

b) Architecture de communication véhicule à infrastructure

Nommée V2I en référence à "Vehicle To Infrastructure (V2I)", appelée aussi architecture de communication en mode infrastructure car elle utilise les points d'accès déployés tout au long de la route comme les RSUs pour assurer des services tels que l'accès à Internet, les informations météorologiques et plein d'autres. La connectivité assurée en mode V2I est relativement élevée comparée à celle assurée en mode V2V et on distingue une meilleure exploitation des ressources du réseau. Ainsi, le réseau véhiculaire prend en compte les applications qui utilisent les points d'infrastructure RSU, ce qui multiplie les services fournis à travers les portails internet en commun.

c) Architecture de communication hybride

Cette architecture est une combinaison des deux techniques précédentes. Dans ce cas le véhicule communique avec d'autres véhicules ou avec des infrastructures routières en un saut ou en plusieurs sauts selon qu'il soit proche de l'infrastructure ou pas c'est-à-dire en fonction de son emplacement.

1.2.3 Services offerts par les VANETs

En dehors de la communication entre véhicules et entre véhicules et différentes infrastructures de la route telles que les RSUs, les postes de polices, les hôpitaux, ... etc., l'usage des VANETs a permis à plusieurs applications de voir le jour ; ce qui représente de véritables services pour les conducteurs, les passagers ainsi que pour les autorités. Dans ce qui suit nous allons présenter les principaux services fournis par les VANETs.

a) Services de sécurité

Ces services sont assurés par diverses applications de sécurité ; celles-ci préservent la sécurité des conducteurs et des passagers sur les routes. Ils permettent au conducteur d'éviter certains problèmes tels que les accidents de la route, les incendies, les attentats . . . etc. Par exemple, si dans une zone donnée un attentat terroriste se produit, les véhicules présents sur les lieux de l'attentat préviennent immédiatement les autorités afin qu'elles puissent intervenir au plus vite et avertissent également les autres véhicules du réseau afin que ceux-ci évitent ce chemin pour leur sécurité.

b) Services de surveillance urbaine

Étant équipés de capteurs et de caméras, les véhicules sont parfaitement adaptés à la surveillance urbaine. Ils complètent les caméras de surveillance et les capteurs fixés sur les infrastructures pour surveiller l'environnement. Ceci aide à prévenir des attaques ; par exemple, un véhicule peut détecter un comportement malveillant d'une personne ou d'un autre véhicule et envoie directement l'information aux autorités. Les véhicules aident également les autorités à mener des enquêtes en cas d'accident.

c) Services de management de trafic routier

Le trafic peut être géré efficacement grâce aux VANETs car l'état des routes, les embouteillages, les accidents de la circulation sont directement détectés par les véhicules et cette information est transmise aux infrastructures routières et aux autres véhicules en temps réel. Le trafic peut être facilement géré et les conducteurs gagnent considérablement du temps puisqu'ils disposent des informations d'intérêt directement et peuvent prendre la décision appropriée en temps réel.

1.2.4 Principaux défis des VANETS

Bien que les VANETs soient des réseaux différents grâce à leurs caractéristiques particulières, leur déploiement s'avère délicat et compte de nombreux challenges. Les principaux défis des VANETs sont les suivants :

- **La gestion du réseau et de la communication** : L'un des principaux défis des VANETs réside dans la gestion du réseau et de la communication. En raison de la haute mobilité des noeuds, la topologie du réseau et l'état des canaux changent très rapidement. De ce fait, aucune structure de gestion ne peut être utilisée, car ces structures ne peuvent pas être mises en place et maintenues aussi rapidement que la topologie changeante du réseau. De plus, aucun coordinateur de communication ne peut être assumé dans ce genre de réseau ; les applications devront communiquer d'une façon fiable en utilisant une communication complètement décentralisée.
- **Le contrôle de la congestion et des collisions** : La taille illimitée du réseau ainsi que la densité variable de celui-ci constituent également un défi. La charge de trafic est faible dans les zones rurales et pendant la nuit dans les zones urbaines. Pour cette raison, les partitions du réseau se produisent fréquemment dans les zones à faible densité, tandis qu'aux heures de pointe, la charge de trafic est très élevée, le réseau est donc encombré et des collisions se produisent assez souvent.

- **L'impact de l'environnement** : Les réseaux VANETs utilisent les ondes électromagnétiques pour communiquer. Ces ondes sont affectées par l'environnement. En effet, un orage ou de fortes rafales de vents peuvent causer des interférences. Ainsi, pour déployer un réseau VANET fiable, il faut prendre en considération l'impact de l'environnement.
- **La conception MAC** : Le contrôle d'accès au support (Medium access control (MAC)) basé sur l'accès multiple par détection de porteuse (CSMA : Carrier Sense Multiple Access) IEEE 802.11 étant un élément clé dans la conception des VANETs, l'un des principaux défis concerne le problème des terminaux cachés.
- **La sécurité** : Les paquets VANETs contiennent des informations vitales, il est donc nécessaire de s'assurer que ces paquets ne sont pas interceptés ou modifiés par un attaquant. De même, la responsabilité des conducteurs doit être établie afin qu'ils informent correctement et en temps réel des conditions de l'environnement et du trafic. La taille du réseau, la mobilité, la pertinence géographique, etc. rendent la mise en oeuvre de la sécurité dans les VANETs difficile et distincte des autres problèmes de sécurité des réseaux.
- **La connectivité intermittente** : Le contrôle et la gestion des connexions réseau entre les véhicules et l'infrastructure constituent un défi majeur. Il faut éviter la perte importante de paquets causée par les coupures de connexion intermittentes dues à la grande mobilité des véhicules et à la présence d'obstacles.
- **La connaissance de l'emplacement** : Il est nécessaire dans les VANETs que les véhicules aient une bonne connaissance de l'emplacement des véhicules avec lesquelles ils communiquent. En effet, chaque véhicule doit connaître la position exacte des autres véhicules du réseau pour pouvoir faire face à une situation d'urgence.
- **La gestion de véhicules hétérogènes** : En raison du grand nombre de véhicules intelligents hétérogènes, la gestion des véhicules hétérogènes et de leurs connexions sporadiques constitue un défi dans les réseaux VANETs.

En plus des défis techniques cités ci-dessus, les challenges sociaux et économiques devraient également être considérés. Il est effectivement difficile de convaincre les fabricants de construire un système qui tend vers la violation des feux de circulation car le consommateur peut rejeter ce type de contrôle. Inversement, le consommateur apprécie le message d'avertissement d'un piège policier. Ainsi, pour motiver le fabricant à déployer les réseaux VANETs, il n'y aura que peu d'incitation [18].

1.2.5 Protocoles et normes dédiés aux VANETs

L'environnement des véhicules prend en charge différentes normes de communication relatives à l'accès sans fil. Ces normes VANET sont utilisées pour faire progresser le transport par véhicule et pour aider les clients à confirmer et à comparer les articles. Il existe diverses normes utilisées dans le cadre des réseaux VANETs, les plus importantes sont : le standard DSRC, la norme WAVE, et la norme ETSI-ITS G5.

a) Communications dédiées à courte portée (DSRC)

Dans le système de transport intelligent, les communications dédiées à courte portée (Dedicated Short Range Communications (DSRC)) sont utilisées par les véhicules pour communiquer entre eux (communication V2V) et avec les infrastructures (communication V2I). Les standards DSRC permettent une communication sans fil de portée moyenne allant de 300m à 1km avec une haute transmission de données de 75 MHz de fréquence du spectre dans la bande 5,9 GHz. Ce spectre est organisé en 7 canaux d'une largeur de 10 MHz chacun. Sur ces 7 canaux, l'un est dédié uniquement pour les communications de sécurité. Deux canaux sont utilisés à des fins particulières telles que la sécurité critique de la vie des personnes et la sécurité du publique à haute puissance, et les autres canaux sont des canaux de service.

Les communications DSRC sont caractérisées principalement par une faible latence, un signal puissant et des interférences limitées ce qui les rend meilleures que les communications d'un réseau WLAN classique. De plus, les standards DSRC comprennent des unités embarquées (OBU) dans les véhicules et des unités de bord de route (RSU) et sont utilisés dans les États Unis et dans la plupart des pays Européens. Cependant, la majorité de ces systèmes ne sont pas compatibles. Il y a donc un besoin important de normalisation afin que ces systèmes hétérogènes puissent être compatibles. Dans cette optique, il existe deux principales approches ; une pour l'Union Européenne appelée l'Institut européen des normes de télécommunication, et l'autre pour les États Unis appelée accès sans fil dans les environnements de véhicules (WAVE). Les systèmes de transport intelligent opèrent dans la bande de fréquence des 5 GHz (ETSI ITS-G5) [19].

b) Accès sans fil en environnements véhiculaires (WAVE)

Également connu sous le nom de IEEE 802.11p, le standard d'accès sans fil en environnements véhiculaires (Wireless Access in Vehicular Environments (WAVE)) représente la normalisation des protocoles pour les applications des systèmes de transport intelligent réalisée par IEEE. La norme WAVE utilise la bande de fréquences 5.85-5.925 GHz pour les communications V2V et V2I et fournit des informations sur le trafic en temps réel améliorant ainsi les performances des réseaux VANETs. L'architecture WAVE assure également la sécurité des messages transférés. Les normes WAVE sont utilisées sur la base des performances d'un large éventail d'applications dans le domaine des transports, parmi elles ; la sécurité des véhicules, la gestion du trafic, les péages automatiques et diverses applications.

La famille WAVE IEEE 1609.x explique une conception et un ensemble correspondant de services, de protocoles normalisés et d'interfaces qui permettent à toutes les stations WAVE de s'activer dans un environnement véhiculaire et d'établir des communications de véhicule à infrastructure (V2I) et de véhicule à véhicule (V2V) [20]. De façon précise, la norme IEEE 1609.2 [21] est portée sur la sécurité. IEEE 1609.3 [22] apporte les services de transport et de réseau, semblablement à la couche de transport et la couche réseau du modèle "Open Systems Interconnection (OSI)". La norme IEEE 1609.4 [23] avec certaines opérations de la sous-couche MAC gère la coordination des canaux et l'acheminement. Les normes IEEE 1609.5 ou IEEE 1609.6 sont axées sur la gestion du réseau et les services de gestion à distance. Enfin, les normes IEEE 1609.11 et IEEE1609.12 spécifient, respectivement, des messages sûrs et sécurisés nécessaires à l'utilisation de paiement électronique

protégé, et la valeur de l'identifiant qui a été allouée pour être utilisée par le système WAVE.

c) ETSI-ITS G5

La norme ETSI ITS-G5¹ [24] basée sur IEEE 802.11p est une extension de la norme WiFi générale, qui a été modifiée et optimisée pour fonctionner dans un environnement automobile dynamique. La norme ETSI ITS-G5 a été définie en 2004 et a fait l'objet d'un processus de normalisation approfondi. Elle correspond à la technique d'accès au support à utiliser dans la bande de fréquence de 5.9 GHz pour les systèmes de transport intelligents européens [25] ainsi qu'à l'ensemble des paramètres et protocoles. Ce processus comprenait des essais approfondis sur le terrain (en commençant en 2008 par l'essai allemand sim1 avec 400 véhicules) et des tests d'interopérabilité multi-fournisseurs (ETSI plug tests depuis 2011). Des implémentations de niveau automobile sont disponibles depuis un certain nombre d'années pour permettre aux fournisseurs de niveau d'effectuer des tests et des validations approfondis, ce qui est absolument essentiel pour un produit lié à la sécurité.

La norme ETSI ITS-G5 est similaire à la norme WAVE en utilisant également les adresses physique MAC basées sur IEEE802.11p et donne une importance capitale à la sécurité et la fiabilité des échanges ainsi qu'à la latence et la puissance de communication. Néanmoins, le canal de contrôle "Control CHannel (CCH)" constitue la principale différence entre les deux standards. En effet, ETSI ITS-G5 repose sur un système multi-radios qui maintient toujours le canal CCH à l'écoute par le biais d'une radio alors que WAVE s'appuie sur des dispositifs radio unique qui s'accordent périodiquement sur des canaux connus afin que les messages importants ne soient pas manqués.

1.2.6 Projets VANETs

L'un des défis majeur des VANETs est sa mise en oeuvre dans un système en temps réel. Ces dernière années ont été marquées par le déploiement de nombreux projets dont l'implémentation s'avère difficile car la mise en oeuvre de tels projets dans un système en temps réel requiert une simulation complète en évaluant les performances du système. En effet, plusieurs gouvernements ont mené de nouveaux projets en vue de développer le système de transport intelligent, notamment l'Europe, le Japan et les USAs qui utilisent les VANETs dans les zones urbaines. Les premières avancées se sont penchées sur le développement des systèmes conventionnels DSRC, WAVE et IEEE 802.11p pour acquérir par la suite les concepts des systèmes d'architecture d'application et de messagerie. Les géants de l'automobile tels que Audi, BMW, Ford, General Motors, Nissan...etc utilisent les ITS pour assurer la sécurité des passagers. Différentes initiatives sont lancées dans divers pays, par exemple la communication sur la sécurité des véhicules aux États-Unis, les projets "Car to Car Communication Consortium (C2C-CC)" dans les pays européens, et le programme de sécurité des véhicules à propulsion au Japan, et font partie des projets d'administration qui s'inscrivent dans le cadre de ces plans [20].

1. ETSI ITS-G5 : European Telecommunications Standards Institute Intelligent Transport Systems

a) Projets VANET en Europe

Dans ce qui suit, nous présentons une liste non exhaustive des projets VANET développés en Europe.

- **Projet "C2C-CC"** : Ce projet ayant débuté en 2001 est conçu pour une communication V2V et utilise le réseau local sans fil IEEE 802.11p sur une portée de 100. Il a pour objectif d'assurer une conduite sécurisée sans accidents et améliorer l'efficacité de la circulation routière en proposant des solutions qui prennent en charge tout type de conduite de la manuelle à l'automatisée. Ce projet se consacre à la mise en œuvre de nouvelles technologies axées sur la concurrence et l'innovation, ce qui favorise la coopération entre les usagers de la route et avec l'infrastructure routière. Il tient compte du type de véhicules et des utilisateurs en considérant les besoins particuliers de chacun et se base sur le partage de l'information, d'intentions et de perception.
- **Projet "Network On Wheels (NoW)"** : NoW est un projet allemand qui a pour objectif de trouver des solutions aux problèmes techniques clés de la sécurité des données et des protocoles de communication dans les communications V2V.
- **Projet "Wireless traffic Safety network between Cars (WiSafeCar)"** : Ce projet a pour objectif de développer une plateforme pour les services de circulation fiable dans le but d'améliorer la sécurité de la route, d'éviter les accidents de la circulation et de fournir de nouveaux services aux véhicules. Cet objectif est atteint grâce à la collecte sécurisée de données provenant des véhicules et des stations fixes, à la diffusion sécurisée des données entre les véhicules et à l'utilisation de ces données pour des applications de services de transport en temps réel.
- **Projet "Mobility and Automation through Advanced Transport Networks (MARTA)"** : Il s'agit d'un projet espagnol qui vise à assurer la sécurité, la durabilité, et l'efficacité des ITS en étudiant les bases scientifiques et les technologies des ITS.
- **Projet "CO-OPERative SystemEms for Intelligent Road Safety (Coopers)"** : Ce projet se concentre sur le développement d'applications télématiques innovantes sur l'infrastructure routière avec pour objectif la gestion coopérative du trafic entre le véhicule et l'infrastructure.
- **Projet "E-Safety Vehicle Intrusion Protected Applications (EVITA)"** : Ce projet a pour objectif de concevoir, de prototyper, et de vérifier des blocs de construction pour les réseaux automobiles embarqués dont les composants importants pour la sécurité sont protégés contre la falsification et dont les données sensibles sont protégées contre la compromission. Le projet EVITA fournira donc, une base pour le déploiement sécurisé de la communication V2V.
- **Projet "Geographic addressing and routing for vehicular communications (GeoNet)"** : Ce projet développe les résultats fondamentaux des travaux du consortium de communication C2C-CC en améliorant ses spécifications et en créant une mise en œuvre logicielle de base qui s'interface avec IPv6. L'objectif de GeoNet

est donc de mettre en oeuvre et de tester formellement un mécanisme de mise en réseau sous la forme d'un module logiciel autonome pouvant être incorporé dans des systèmes coopératifs.

- **Projet "SEcure VEhicle COmmunication (SEVECOM)"** : Ce projet a pour but de fournir une définition complète et une mise en oeuvre des exigences de sécurité pour les communications véhiculaires.

b) Projets VANETs aux États-Unis

Le standard d'accès sans fil dans l'environnement véhiculaire WAVE étend plusieurs projets aux USA parmi eux :

- **Projet "Intelligent Vehicle Initiative (IVI)" [26]** : IVI a pour principal objectif de veiller à assurer la sécurité routière.
- **Projet "Vehicle Safety Communication (VSC)" [27]** : Ce projet travaille en coordination avec l'administration de la sécurité routière pour assurer une communication sécurisée entre les véhicules.
- **Projet "Vehicle Safety Communication-2 (VSC-2)" [27]** : Ce projet est similaire au projet VSC en intégrant en plus, la messagerie, les protocoles, les systèmes et les interfaces pour les communications véhiculaires.
- **Projet "Vehicle Infrastructure Integration (VII)" [28]** : Ce projet a débuté dans la ville de Detroit aux États-Unis, il a pour but d'améliorer les communications entre véhicules et entre véhicules et infrastructures. Pour ce faire, VII a collaboré avec General Motors, Ford, Honda, BMW, Nissan, Volkswagen et Daimler-Chrysler.

c) Projets VANET au Japan

Le Japan mène de nombreux projets et essais dans le domaine de l'exploration automobile. Dans ce qui suit, nous citons quelques uns des principaux projets :

- **Projet "Advanced Safety Vehicle (ASV)" [28]** : C'est un projet de sécurité de pointe. Il a été étendu en 2001 à ASV-3, puis en 2005 à l'ASV-4 et fournit un système de navigation et un système automatique d'évitement de collisions. ASV est soutenu par Mitsubishi, Honda, Toyota, et Suzuki.
- **Projet "Demo 2000 cooperative driving" [29]** : Les premiers essais de ce projet ont débuté en l'an 2000 et fournissent un système coopératif d'aide à la conduite. DEMO se base sur la bande de fréquences 5,8 GHz et les protocoles d'accès multiples CSMA pour la communication.
- **Projet "Japan Automobile Research Institute (JARI)" [28]** : Ce projet se focalise principalement sur la sécurité. Il conduit de nombreux essais pour des projets et évalue les projets de l'union européenne et des États-Unis.

- **Projet "Energy ITS" [30]** : Ce projet a débuté en 2008 et a pour objectif d'économiser l'énergie et prévenir le réchauffement climatique grâce aux technologies des systèmes de transport intelligent. Les essais de ce projet ont pu conclure à une réduction d'environ de 14% de consommation du carburant.

1.3 Le Cloud computing

Dans cette section, nous présentons le Cloud computing, ses caractéristiques, les types de Clouds ainsi que ses domaines d'application.

1.3.1 Définition du Cloud computing

Le Cloud computing [31] est l'utilisation de serveurs informatiques distants via Internet pour exploiter leur capacité de stockage et de calcul. Lorsqu'il est utilisé, les services et applications ne sont pas tenus d'être exécutés localement sur les machines des utilisateurs finaux. Le Cloud computing vise à fournir des ressources informatiques aux organisations, aux entreprises ou aux clients ordinaires en cas de besoin. Des serveurs distants sont alloués à la demande, ce qui permet de réduire les coûts en mettant en oeuvre un modèle de paiement à l'utilisation et en favorisant les économies d'échelle. Le principal avantage des services Cloud est qu'ils fournissent une infrastructure dynamique qui permet de modifier les services fournis en fonction des différents niveaux de demande.

1.3.2 Le Cloud computing mobile

Le Cloud computing mobile [32] fait référence au Cloud computing où le client ou l'utilisateur final est un utilisateur mobile. Cela signifie que l'utilisateur mobile se déplace constamment et utilise les ressources du Cloud de manière dynamique à partir de plusieurs emplacements. On parle aussi de Cloud computing mobile lorsque les ressources du Cloud lui-même ne sont pas statiques, c'est-à-dire lorsque les ressources collectives de différents appareils mobiles sont alloués à d'autres utilisateurs, formant ainsi un Cloud mobile comme pour les ressources d'un téléphone mobile ou d'un véhicule en mouvement.

1.3.3 Caractéristiques du Cloud computing

L'utilisation du Cloud computing permet d'étendre la capacité de calcul et l'espace de stockage mais pas que. En effet, grâce à ses caractéristiques, le Cloud assure une continuité des services, une redondance des services très utile aussi surtout en cas de panne informatique qui pourrait subvenir à la suite de divers problèmes tels que les séismes, les incendies ..., ainsi que la disponibilité des services à tout moment. Ces principales caractéristiques sont :

- **Dynamisme** : Le Cloud fournit une infrastructure dynamique qui permet de modifier et changer les services fournis selon la variation du type et de niveau de la demande.
- **Elasticité** : Cette caractéristique qu'a le Cloud computing lui permet d'assurer une continuité de ses services en augmentant ou en diminuant ceux-ci en fonction des

besoins des utilisateurs. Le système peut alors s'adapter aux variations de la charge de travail en rajoutant des ressources ou en les annulant selon les besoins.

- **Extensibilité** : Le Cloud permet aux applications et aux produits informatiques de continuer de fonctionner même quand ceux-ci changent de volume ou de taille afin de répondre aux besoins des utilisateurs.

1.3.4 Modèles de déploiement du Cloud computing

Le modèle de déploiement du Cloud computing diffère selon la configuration de chaque organisation où chaque entreprise ou organisation déploie ou utilise le Cloud computing en fonction de ses besoins spécifiques. Comme illustré dans la Figure 1.3, on distingue différents modèles de déploiement comme suit :

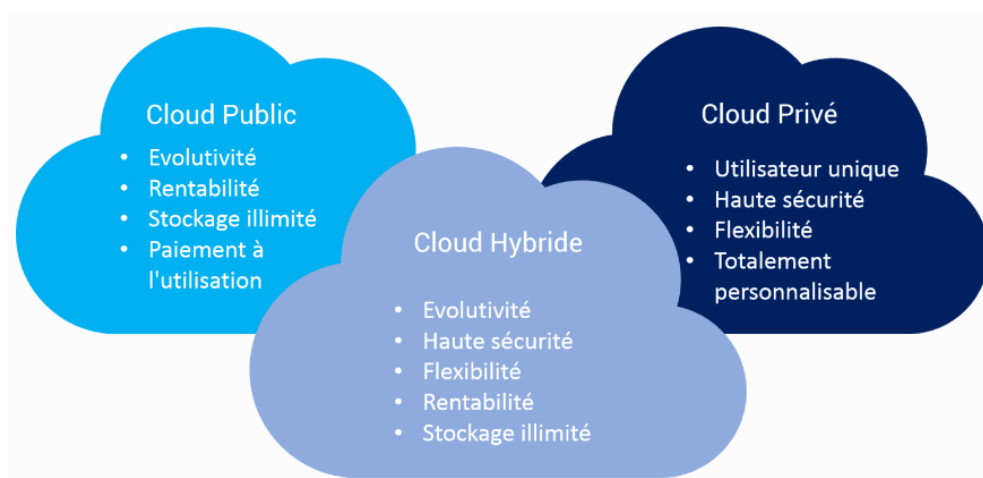


FIGURE 1.3 – Modèles de déploiement du Cloud [5]

- **Le Cloud public** : Désigne une structure informatique où les services web sont hébergés en dehors de l'organisation de l'utilisateur final. Le fournisseur de services met ses ressources à disposition du grand public et ceux-ci sont accessibles simplement via Internet. Ces ressources varient en fonction des fournisseurs et incluent la capacité de traitement, la capacité de stockage, les machines virtuelles ainsi que différentes applications. Le principal avantage de l'utilisation d'un Cloud public réside dans sa facilité de configuration et l'économie réalisée puisque les coûts relatifs aux matériels, à la bande passante ainsi qu'au déploiement des applications sont à la charge du fournisseur.
- **Le Cloud privé** : Désigne l'infrastructure informatique où les ressources et les services sont déployés au sein même de l'organisation ou du réseau d'entreprise. Ces ressources et services sont déployés et configurés et gérés par le service informatique de l'organisation ou de l'entreprise et ne sont accessibles et utilisables que par les membres de cette même organisation. Le déploiement de ce type de Cloud est relativement coûteux et nécessite des compétences informatiques accrues pour sa mise en place, ce qui représente le principal inconvénient du Cloud privé. Cependant, le fait d'être accessible uniquement localement par un ensemble de personnes bien connues

et définies rend la sécurité du Cloud privé relativement élevée en plus du fait qu'il soit totalement personnalisable et adaptable à l'organisation.

- **Le Cloud privé hébergé** : Désigne un environnement Cloud déployé et configuré par une organisation, hébergé dans cette même organisation mais dédié à une autre organisation ou entreprise spécifique. Ce Cloud a été conçu spécialement pour cette organisation spécifique afin de répondre à ses besoins. Les ressources et services du Cloud privé hébergé sont accessibles uniquement par les utilisateurs de l'entreprise pour laquelle il est dédié. L'avantage du Cloud privé hébergé est que les ressources sont gérées par le fournisseur distant qui héberge le Cloud, et donc l'entreprise qui les utilise n'a pas besoin d'avoir des compétences spécifiques en informatique afin de pouvoir administrer le Cloud.
- **Le Cloud hybride** : Désigne une infrastructure Cloud mixte qui combine à la fois le Cloud public et le Cloud privé. Le déploiement de ce type de Cloud comprend au moins un Cloud public et un Cloud privé ; par exemple, au sein d'une organisation, les serveurs peuvent être hébergés à distance sur un Cloud public tandis que certaines ressources sont fournies en interne par le Cloud privé et gérées par le service informatique de l'organisation.

1.3.5 Modèles de service du Cloud computing

Le Cloud computing n'étant pas considéré comme une seule technologie en soi comme c'est le cas pour un téléphone portable ou encore une puce électronique, il est plutôt axé service composé principalement de trois services comme le montre la figure 1.4.

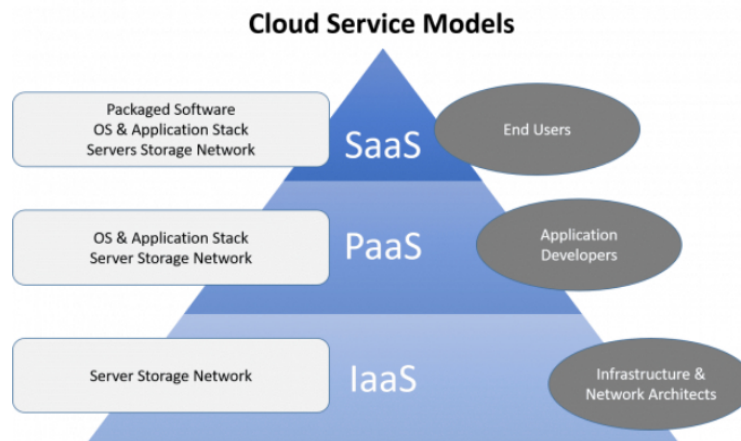


FIGURE 1.4 – Modèles de service du Cloud computing [6]

- **Le service "Software as a Service (SaaS)"** : Contrairement aux méthodes classiques d'utilisation de logiciels où le client devait payer une licence afin de pouvoir installer et utiliser un logiciel sur son ordinateur personnel ainsi qu'un contrat de maintenance afin d'avoir droit aux correctifs et mises à jour associés au logiciel, le service SaaS du Cloud computing permet à l'utilisateur d'avoir les logiciels dont il

a besoin sans pour autant être contraint de payer la licence ni le contrat de maintenance ; l'utilisateur paye seulement des droits d'utilisation des logiciels sous forme d'abonnements. Le fournisseur fournit le service complet (les logiciels et les données qui leur sont associées) et l'utilisateur accède directement au service sans avoir à se soucier des installations, du matériel, de la maintenance ou de l'achat de licences. Il a accès aux services délivrés simplement via une interface web ou par le biais d'un périphérique autorisé. La principale différence qui distingue le modèle SaaS du modèle traditionnel réside dans le fait que dans le modèle traditionnel le client achète le logiciel et donc ce dernier n'est utilisé que par ce même client ou par un ensemble de personnes appartenant à cette même entreprise ou organisation qui a acheté le service. Tandis que pour le modèle SaaS une même application hébergée sur le Cloud est utilisée simultanément par plusieurs clients dans différents endroits et les ressources matérielles sont donc partagées entre ces différents clients en distinguant bien sûr les données propres à chaque client.

- **Le service "Platform as a Service (PaaS)"** : Les méthodes traditionnelles de développement d'applications informatiques requièrent l'achat, la configuration et la mise en réseau de serveurs complexes et coûteux ainsi que d'avoir un personnel qualifié pour déployer tout cela afin d'assurer l'environnement adéquat au bon développement d'applications. Le service PaaS offre aux développeurs la possibilité d'avoir l'environnement de développement tout prêt sans se soucier de l'achat et de la configuration du matériel, simplement en accédant via un navigateur web à la plateforme requise moyennant des frais d'utilisation. Ainsi les fournisseurs offrent aux clients la plateforme de développement qui inclut les serveurs de calcul, la mise en réseaux et les baies de stockage sur laquelle l'utilisateur peut développer et héberger ses propres applications. Le client n'a même plus à se soucier de la sauvegarde de ses données et traitements puisque la plateforme offre même des baies de stockage assurant la sauvegarde et la redondance des données.
- **Le service "Infrastructure as a Service (IaaS)"** : Ce modèle fournit à l'utilisateur l'infrastructure informatique pour l'entreprise ou l'organisation. Le fournisseur héberge toute l'infrastructure informatique de l'entreprise, à savoir les serveurs, le réseau, le système d'exploitation, la virtualisation et le stockage et l'utilisateur peut gérer son infrastructure à distance comme si celle-ci était hébergée localement dans ses propres locaux sans avoir à se soucier des contraintes matérielles.

1.4 Le Fog computing

1.4.1 Définition du Fog computing

Le Fog computing [33] est un nouveau paradigme pour fournir des ressources informatique d'une manière géographiquement distribuée [34]. Il a été introduit comme une extension de l'infrastructure Cloud existante pour fournir des services en temps réel et à faible latence aux terminaux géographiquement distribués. En outre, le Fog computing n'est pas considéré comme un paradigme alternatif aux réseaux de Cloud computing, mais il est introduit pour le compléter. De plus, contrairement au Cloud computing, qui repose sur quelques centres de données de grande capacité, le Fog computing utilise un large

éventail de ressources distribuées et hétérogènes à capacité modérée, appelées noeuds de Fog "Fog Nodes (FNs)" [35]. Les FNs peuvent être déployés dynamiquement avec une connexion réseau n'importe où sur le réseau et à tout moment selon les besoins des utilisateurs de leurs ressources, et tout appareil disposant d'un espace de stockage et d'une capacité de calcul peut faire office de FN. Par conséquent, l'utilisateur peut demander les ressources et les obtenir très rapidement sans surcharger le réseau de paquets de requêtes qui montent jusqu'au Cloud conventionnel, puis retournent tout en bas sous forme de paquets de réponses pour répondre à la demande de l'utilisateur.

Le principal avantage du Fog computing par rapport au Cloud computing est la proximité des terminaux, mais avec une capacité très limitée telle que la capacité des capteurs sans fil, et des caméras intelligentes. Cependant, les applications qu'impliquent les terminaux finaux nécessitent généralement des dispositifs plus puissants. Une des solutions est d'utiliser le Cloud comme support pour ces applications. Par contre, cette stratégie conduit à une latence élevée en raison du long chemin réseau entre les terminaux et les centres de données Cloud. De plus, la bande passante de la connexion réseau des centres de données Cloud peut devenir un goulot d'étranglement majeur. Par conséquent, une stratégie basée sur le Fog computing peut résoudre ces problèmes car l'utilisation des FNs peut entraîner une latence considérablement plus faible et éliminer le goulot d'étranglement au niveau du Cloud.

1.4.2 Caractéristiques du Fog computing

Le Fog computing étant une extension du Cloud computing jusqu'aux bordures du réseau, celui-ci est proche des différents dispositifs réseau qui utilisent l'IoT et rapproche donc les services de traitement et de stockage de données aux bordures du réseau. Les noeuds impliqués dans le Fog computing sont appelés Fog nodes. N'importe quel dispositif ayant une capacité de traitement, une capacité de stockage et une connectivité réseau peut faire office de Fog node. Ces FNs peuvent être déployés n'importe où dans le réseau du moment où la connexion est disponible. Ils agissent sur les données des dispositifs réseaux ; ils traitent eux-mêmes ces données comme ils peuvent faire office d'intermédiaire entre les dispositifs informatiques du réseau et le Cloud conventionnel. La figure 1.5 représente un exemple du Fog computing.

Le Fog computing doit son efficacité aux principales caractéristiques qui lui sont attribuées, parmi elles, on cite :

- **Faible latence** : Elle est la principale caractéristique du Fog computing. Compte tenu du fait que les noeuds Fog sont proches des dispositifs finaux du réseau, le traitement et le stockage se font au bord du réseau et donc à proximité des dispositifs concernés. En effet, dans le Fog computing, la requête n'a pas besoin de traverser tout le réseau pour être traitée comme c'est le cas pour le Cloud computing, ce qui réduit considérablement le temps de réponse.
- **Distribution géographique** : Du fait que les Fog nodes sont géographiquement distribués, les applications et services du Fog computing le sont eux aussi et peuvent être déployés n'importe où dans le réseau.
- **Élasticité** : Le Fog computing a pour caractéristique de pouvoir s'étendre grâce aux noeuds Fog qui peuvent être déployés au besoin n'importe où dans le réseau.

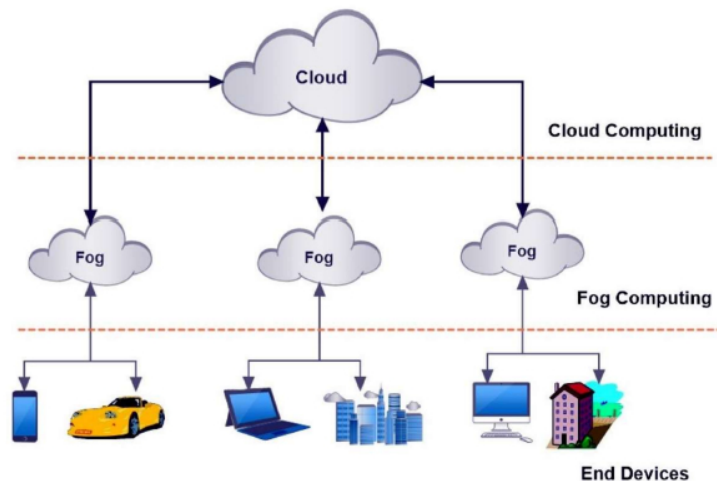


FIGURE 1.5 – Illustration de Fog computing [7]

- **Connaissance de l'emplacement** : le Fog computing supporte la connaissance de l'emplacement grâce aux noeuds Fog qui sont déployés dans différentes localités à travers le réseau.
- **Mobilité** : Le Fog computing supporte la mobilité permettant ainsi les applications mobiles d'une part, et d'autre part, les noeuds Fog sont aptes à se connecter directement aux appareils mobiles tels que les véhicules, les smartphones ou encore les montres connectées.
- **Interopérabilité** : Les noeuds Fog ont la capacité d'opérer et d'interagir avec des dispositifs de différentes plateformes dans différents domaines à travers différents prestataires de service.
- **Interactions en temps réel** : Le Fog computing a l'aptitude de traiter les requêtes des utilisateurs directement sans aucun délais ce qui le rend propice aux applications sensibles au temps et les applications en temps réel.
- **Hétérogénéité** : Les noeuds du Fog computing sont conçus par différents fabricants et viennent donc sous différentes formes et doivent être déployées selon les plateformes de ces fabricants. Le Fog computing a la capacité d'opérer sur différentes plateformes [7].
- **Évolutivité** : Il existe des réseaux de capteurs à grande échelle qui surveillent l'environnement aux alentours. Le Fog computing fournit des ressources de calcul et de stockage distribuées qui peuvent fonctionner avec de tels terminaux à grande échelle [7].

1.4.3 Architecture du Fog Computing

L'approche du Fog computing rapproche le stockage et le traitement de données aux bordures du réseau informatique, mais offre des capacités de traitement et de stockage

très limitées. C'est pourquoi, souvent le Fog computing agit en intermédiaire entre les utilisateurs finaux et le Cloud.

L'architecture générale du Fog computing repose sur une architecture en couches de six niveaux selon Mukherjee et al. [36] et Aazam et Huh [37, 38] :

- La couche physique et virtualisation : Cette couche représente le niveau inférieur de l'architecture ; elle implique la sous-couche physique qui comprend les capteurs physiques et les objets intelligents ainsi que le réseau, et la sous-couche de virtualisation qui comprend les capteurs virtuels avec leur réseau.
- La couche de surveillance : Cette couche est chargée de surveiller la disponibilité des noeuds Fog, des services et des ressources pour savoir quelle ressource est gérée par quel noeud Fog, ainsi que la surveillance de l'utilisation des ressources et éléments du réseau.
- La couche de prétraitement : Elle est chargée de prétraiter les données collectées. Celles-ci sont analysées et filtrées afin que seules les données d'intérêt soient gardées et transmises aux niveaux supérieurs.
- La couche de stockage temporaire : Dans cette couche, nous retrouvons les données filtrées et prétraitées par la couche de prétraitement où elles sont stockées d'une manière temporaire car celles-ci sont directement supprimées après avoir été envoyées au Cloud.
- La couche de sécurité : Les données passent par cette couche en vue de leur assurer une sécurité par chiffrement et mesures d'intégrité avant d'être transmises au Cloud.
- La couche de transport : Cette couche permet la transmission des données sécurisées par la couche de sécurité au Cloud conventionnel.

La figure 1.6 illustre l'architecture générale du Fog computing.

1.4.4 Applications du Fog computing

La proximité des ressources de traitement et de stockage avec les utilisateurs ainsi que le temps de réponse court pour traiter les requêtes des utilisateurs rendent le Fog computing approprié dans plusieurs environnements et propice aux utilisations de beaucoup d'applications tels que :

- **Véhicules intelligents connectés** : Dans le domaine des VANETs, les véhicules produisent et utilisent de plus en plus de données. Et celles-ci ont besoin d'être accessibles et être traitées rapidement en se basant sur les informations disponibles relatives à l'état de la route par exemple ou aux conditions de la conduite et de l'environnement. Ce qui rend le Fog computing réellement approprié pour ce genre de services où les informations des véhicules sont collectées et traitées en temps réel sans la moindre attente par le biais des noeuds Fog.

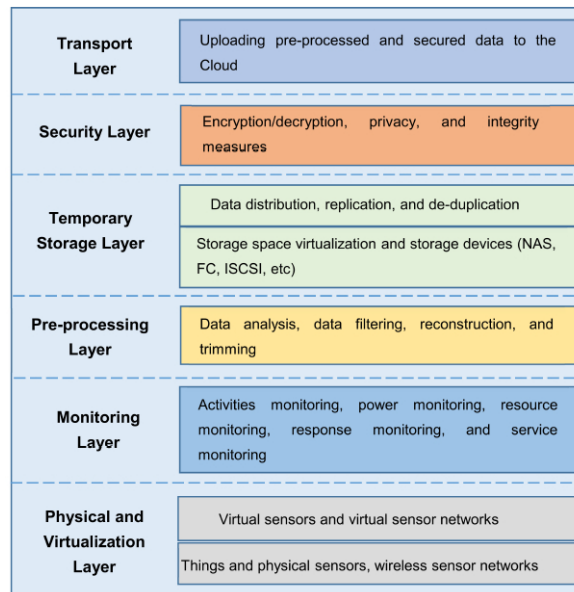


FIGURE 1.6 – Architecture générale du Fog computing [7]

- **Gestion des données de la santé** : Les données relatives à l'état de santé des personnes sont tellement sensibles et importantes que leur gestion doit se faire en temps réel sans aucun retard, et celles-ci doivent être accessibles aux patients et aux médecins afin de suivre l'état général du patient d'une manière très rapide. Ceci rend l'utilisation du Fog computing prometteuse quant à la gestion et la disponibilité des données. Le Fog computing permet également au patient d'accéder aux informations confidentielles relatives à son état de santé localement sans avoir à se déplacer ; celles-ci étant stockées localement dans un noeud fog tel que le smartphone du patient ou son ordinateur personnel.
- **Maisons intelligentes connectées** : Les progrès dans le domaine de l'IoT ont tellement explosé que nous trouvons de nos jours, des objets intelligents partout dans la maison. L'utilisation du Fog computing permet de croître les ressources de ces objets en les alimentant en capacité de traitement ou de stockage au besoin grâce à sa propriété d'être élastique et adaptatif aux besoins des utilisateurs.
- **Grille intelligente** : Étant un réseau de distribution d'électricité qui utilise la technologie de communication numérique pour détecter et réagir aux changements locaux d'utilisation, il utilise les données en temps réel pour une gestion efficace des systèmes. Il est nécessaire de traiter les données distantes à proximité de l'endroit où elles sont produites. Il est également probable que des données provenant de plusieurs capteurs soient produites. Le Fog Computing constitue une solution idéale pour ce genre de traitements, il est construit de telle manière que toutes ces exigences peuvent être satisfaites.
- **Analyse en temps réel** : Les données peuvent être transférées à l'aide des noeuds Fog déployés depuis l'emplacement où elles sont produites vers différents emplacements. Le Fog Computing est utilisé pour l'analyse en temps réel de ces données en

les transmettant aux institutions financières qui utilisent les données en temps réel à partir de réseaux de production.

1.5 Le Cloud et Fog véhiculaire

1.5.1 Le Cloud Véhiculaire

Le paradigme du Cloud Véhiculaire noté "Vehicular Cloud Computing (VCC)" [39] se rapporte à l'utilisation du Cloud computing dans les VANETs. L'idée a été inspirée par le besoin croissant de développer les VANETs en intégrant différentes applications pour les conducteurs pour leur confort et leur sécurité sur la route.

Le VCC peut être effectué en utilisant les ressources du Cloud conventionnel via Internet. Les véhicules connectés qui ont besoin d'être accompagnés en termes de capacité de calcul ou de stockage pour servir leurs applications et les besoins des conducteurs utilisent le Cloud conventionnel pour satisfaire leurs exigences. Le Cloud fournit les ressources nécessaires aux véhicules pour une période précise, le temps que les véhicules répondent à leurs besoins. De plus, on parle de VCC également dans le cas de véhicules qui eux-mêmes louent leurs propres ressources à d'autres véhicules lorsqu'ils ne les utilisent pas. Par exemple, à défaut de gaspiller bêtement ses ressources lorsqu'un véhicule est garé dans le parking d'un centre commercial, le véhicule peut rendre ces ressources accessibles à d'autres véhicules ou utilisateurs afin de les exploiter. La figure 1.7 représente un exemple d'architecture VCC.

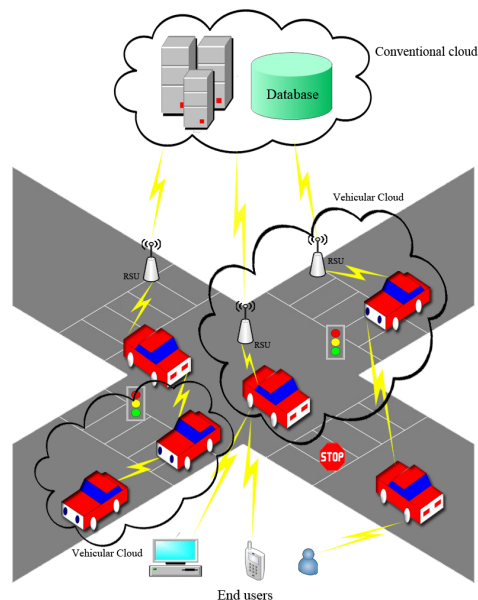


FIGURE 1.7 – Exemple de Cloud véhiculaire

Au-delà de la capacité de traitement ainsi que de l'espace de stockage, le véritable point fort des Clouds véhiculaires réside dans les capteurs qu'ils portent. En effet, les véhicules constituent des plateformes d'observation idéales pour l'environnement et peuvent voir et mémoriser un grand degré de détail (bien au-delà des caméras fixes installées sur les structures latérales). De plus, les informations qu'ils stockent sont pertinentes.

Par exemple, un conducteur souhaitant s'attabler dans un restaurant pour manger, peut obtenir de meilleures recommandations de bons restaurants de la part des véhicules du quartier que s'il cherchait ces recommandations sur le web. Cette capacité illimitée de collecter les informations ainsi que la pertinence de ces informations représentent le principal avantage des VCC. En conservant les informations sur le véhicule, nous économisons les frais de téléchargement des informations sur le Web et le stockage associé. De plus, nous économisons le coût et le temps de téléchargement [39].

Le fait que les informations d'intérêt pertinentes soient localement stockées et donc disponibles et accessibles à tout moment représente un très grand avantage dans plusieurs situations ; notamment en cas de besoins des applications de gestion de trafic routier ou encore de surveillance urbaine ; celles-ci étant particulièrement utiles pour prévenir des attaques terroristes et assurer la sécurité des gens. En effet, les véhicules sont des dispositifs idéals pour assurer ce genre de services. Étant mobiles, donc changeant constamment de position et équipés de capteurs, les véhicules assurent parfaitement la tâche de capturer l'état de la route à tout moment et à différents endroits et stockent ces informations pertinentes localement sur les dispositifs de stockage disponibles grâce au VCC. Aussitôt qu'un problème survient, les autorités sont directement informées et prennent immédiatement les dispositions nécessaires afin de faire face à toute situation difficile. Cela reste vrai aussi pour les investigations des autorités en cas d'enquête criminelle où la police a la possibilité de traquer tout délinquant ou criminel en fuite grâce aux images de vidéo surveillance capturées par les véhicules. Concernant la gestion du trafic routier, grâce aux informations relatives à l'état de la route stockées localement sur le VCC, le conducteur peut facilement faire le choix du chemin à emprunter afin d'atteindre sa destination dans les plus brefs délais en évitant et en contournant toutes les routes où la congestion est présente ou encore en évitant les routes barrées pour cause de travaux ou accidents. Ces informations d'intérêt, le conducteur les retrouvera facilement sans grand effort grâce au VCC.

On souligne également les différences entre les Cloud mobiles orientés mission et les Cloud véhiculaires du fait que ces derniers soient opportunistes et planifiés par coïncidences pour des besoins de commutation ou par leur critère de proximité géographique tandis que les Clouds mobiles orientés mission sont pré-planifiés pour répondre à des objectifs bien précis tels que la surveillance de l'environnement, l'exploration du terrain, la reprise après sinistre, ... etc [39]. Ce qui donne l'avantage au VCC d'être plus flexible. On distingue aussi les Clouds personnels mobiles des Clouds véhiculaires par le fait que les Clouds personnels offrent un type d'applications différent et plus diversifié des Clouds véhiculaires ; principalement des applications de l'informatique sociale, de santé, de détection urbaine et de divertissement. Prenons comme exemple la capture de différentes vues lors d'un concert ou d'un événement sportif et le partage de ces vues avec des spectateurs. Ils offrent également des implications plus fortes sur les réseaux sociaux parmi ses membres que le VCC car les conducteurs sont trop occupés pour aller de A à B pour socialiser. Le principal inconvénient des Clouds mobiles personnels se situe dans la limitation de la puissance de la batterie, il faut donc évaluer soigneusement le compromis entre le traitement et le stockage [39].

1.5.2 Architecture du Cloud Véhiculaire

Comme illustré par la figure 1.8, le système Cloud véhiculaire repose sur une architecture en couches qui s'étale sur trois niveaux : la couche interne du véhicule, la couche de communication et la couche Cloud.

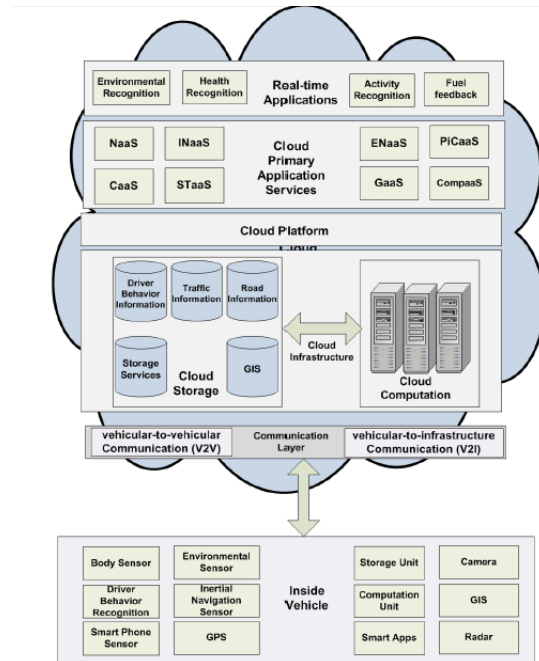


FIGURE 1.8 – VCC architecture [8]

a) Le niveau interne du véhicule

C'est le niveau le plus bas de l'architecture, il concerne les véhicules avec leurs capteurs. Ces différents types de capteurs tels que les capteurs des véhicules, les capteurs environnementaux ou encore les capteurs des smartphones présents dans le véhicule permettent la collecte et le traitement des informations telles que l'état de la route, les facteurs environnementaux, la météo ou encore l'humeur et le comportement des conducteurs [40] par exemple en cas de conduite dangereuse, une alerte est directement envoyée aux véhicules avoisinants. Ces informations seront par la suite envoyées au Cloud pour être utilisées en tant qu'entrée pour diverses applications et logiciels dans la couche application du Cloud. En plus des capteurs qui lui sont associés, chaque véhicule sert aussi de dispositif de stockage et de traitement en complément aux ressources déjà disponibles. Cependant, il est important que le stockage de ces informations soit distribué entre les différentes ressources disponibles et non centralisé afin que les applications en temps réel puissent y accéder facilement en cas de besoin.

b) Le niveau de communication

Le niveau de communication représente le niveau intermédiaire entre les véhicules et le Cloud conventionnel. Cette couche se concentre sur la stratégie de communication afin

d'assurer une connexion du véhicule avec le Cloud. Elle inclut la stratégie de communication inter-véhiculaire c'est-à-dire la communication V2V et la stratégie communication entre véhicules et infrastructures c'est-à-dire la communication V2I. La communication V2V est utilisée et particulièrement utile dans le cas de messages d'alerte ou d'urgence par exemple lorsqu'un véhicule contacte un autre véhicule pour le prévenir qu'une collision est survenue ou alors qu'un changement de voie s'impose, ou encore pour l'alerter sur un danger potentiel sur le chemin comme les incendies . . . etc. Les fonctions de mise en réseau en mode ad hoc sont activées tant que les véhicules sont dans une plage de communication valide. En ce sens, un VCC peut être assemblé dynamiquement, de manière autonome, et opportuniste à travers des véhicules via leur connexions V2V. Cette architecture de communication permet la mise en oeuvre de plusieurs applications flexibles et fluides [41].

Le deuxième composant de la couche communication est la communication V2I. Ce type de communication est particulièrement utile dans les applications et services de transport où il est plus approprié d'utiliser les infrastructures de communication afin d'échanger les informations de contrôle entre véhicules, avec le Cloud, ou avec les infrastructures. Cette stratégie permet aussi d'avoir une plateforme de partage de données sur Internet. Par conséquent, les conducteurs accèdent directement aux informations relatives à l'état de la route par exemple ? en ayant une vue plus large du réseau routier. Ceci leur permet de contourner les routes à danger et la sécurité des conducteurs et des passagers se voit alors nettement améliorée.

c) Le niveau Cloud

C'est la couche supérieure de l'architecture où se font les calculs intensifs et complexes en un temps réduit ainsi que le stockage des gros volumes de données. Cette couche à son tour se compose de trois couches essentielles, la couche application, la couche plateforme et la couche infrastructure comme illustré dans la figure 1.8. La couche application prend en charge diverses applications en temps réel ainsi que des applications primaires du Cloud et la couche infrastructure consiste en infrastructures de traitement du Cloud où se font tous les traitements qui requièrent des ressources accrues de calcul et en infrastructures de stockage du Cloud où toutes les informations et données sont enregistrées et stockées.

1.5.3 Le Fog Véhiculaire

Le paradigme du Fog Véhiculaire noté "Vehicular Fog Computing (VFC)" [9] est l'utilisation du Fog computing dans l'environnement des véhicules. Il réduit la charge et le trafic généré pour atteindre les centres de données du Cloud en déplaçant les ressources de stockage de données et de calcul à la périphérie du réseau. De plus, les RSUs et autres composants du réseau tels que les feux de circulation peuvent constituer des FNs et traiter les demandes de véhicules en temps réel sans surcharger le réseau. Cette solution est la plus adaptée aux applications à latence sensible, comme dans le cas d'un accident de la route où le conducteur doit immédiatement informer les voisins et les autorités de la situation. L'utilisation du Fog Computing dans les environnements véhiculaires est également utile en termes de bande passante; elle est considérablement économisée par rapport à l'utilisation du Cloud conventionnel. La figure 1.9 illustre un exemple de VFC.

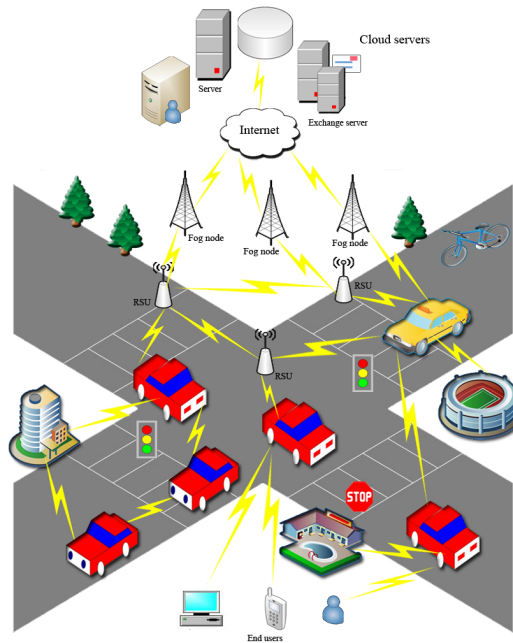


FIGURE 1.9 – Exemple de Fog véhiculaire

1.5.4 Architecture du Fog véhiculaire

Le Fog véhiculaire se distingue du Cloud véhiculaire par la proximité des ressources informatiques avec les utilisateurs finaux. En effet, les ressources se situent à la périphérie du réseau ce qui réduit considérablement le temps de réponse lors des traitements des requêtes des utilisateurs.

D'une manière générale, l'architecture du Fog véhiculaire est classifiée comme étant une architecture à couches qui se compose principalement de trois couches superposées comme le montre la figure 1.10.

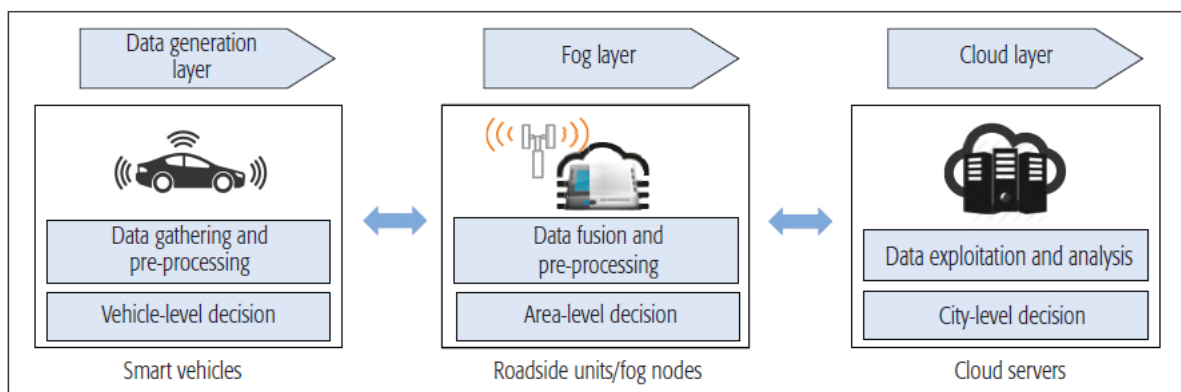


FIGURE 1.10 – Architecture du VFC [9]

a) La couche génération de données

C'est la couche périphérique où sont générées les données des utilisateurs et véhicules. Elle comprend principalement les véhicules connectés du réseau ; celles-ci ayant l'aptitude à produire une quantité considérable de données grâce aux dispositifs qu'elles comportent (capteurs, radars, GPS, caméras, ...), mais aussi tous les dispositifs et objets intelligents pouvant générer des données tels que les smartphones, les montres connectées, les tablettes ... etc. Certaines de ces données peuvent être traitées directement par le véhicule lui-même en exploitant ses propres ressources afin de prendre des décisions au niveau du véhicule en temps réel, tandis que d'autres données générées à ce niveau sont transmises via le réseau à la couche supérieure en vue de leur traitement et utilisation pour d'autres fins comme par exemple la surveillance urbaine ou encore la planification d'infrastructure ou autre.

b) La couche Fog

Dans cette couche sont traitées les données transmises par la couche de génération de données. Les noeuds Fog se chargent de collecter les données envoyées par les véhicules, de traiter ou pré-traiter ces données et de les envoyer au Cloud. Les noeuds Fog font également office d'intermédiaire entre les utilisateurs finaux et le Cloud conventionnel ; il peut s'agir d'unités de route RSUs qui assurent la connexion entre les deux, mais pas que. Ils ont également pour fonction de traiter l'information et de stocker les données comme pour le cas de la gestion des feux de circulation.

c) La couche Cloud

Dans cette couche se trouvent les centres de données et les serveurs lourds en performance afin d'assurer les traitements les plus complexes. Ces serveurs utilisent les données téléchargées et pré-traitées par les noeuds Fog pour performer les calculs intensifs et analyses afin de prendre les grandes décisions de manière optimale. A titre d'exemple, nous citons la gestion du trafic routier d'une ville qui peut être assurée par le Cloud en prenant les décisions adéquates de libérer certains tronçons de route ou par le biais de feux de circulation intelligents en commandant le changement des feux en fonction du taux de congestion des différentes routes à travers la ville.

1.5.5 Comparaison entre le Cloud véhiculaire et le Fog véhiculaire

L'utilisation du Cloud computing et du Fog computing dans les réseaux de véhicules a un impact remarquable sur la communication entre les véhicules. Cela a conduit à de nombreuses applications très utiles pour les conducteurs et pour les autorités telles que la gestion de la circulation et la surveillance urbaine [39]. Cependant, bien que le VCC et le VFC aient beaucoup de similitudes, ils ont également quelques différences à commencer par leur situation géographique. En effet, le Fog computing est situé au bord du réseau véhiculaire, tandis que le Cloud computing est plus éloigné. Effectivement, dans le Fog computing, on a une proximité avec les utilisateurs finaux. De plus, le Fog computing rapproche également les ressources aux utilisateurs, ce qui le rend plus adapté pour les applications sensibles au délai [42], contrairement au VCC qui ne supporte pas tellement

les applications en temps réel; il met souvent du retard pour répondre à une demande des utilisateurs en raison de son éloignement de ces derniers. Cette fonctionnalité rend le Cloud computing inadapté aux messages d'urgence et aux applications en temps réel. De plus, le Fog computing a une grande distribution géographique et prend en charge une mobilité élevée, contrairement au Cloud qui est centralisé. Ainsi, le Fog computing est adapté à l'environnement véhiculaire où les véhicules sont très dynamiques et se déplacent constamment dans diverses directions. La décision de diffusion en VFC est prise localement, ce qui réduit le délai et la consommation de la bande passante, puisque nous n'avons pas besoin d'envoyer les informations aux serveurs distants dans le Cloud. De plus, le Fog computing utilise également l'équilibrage de charge pour gérer les communications, contrairement au Cloud computing où les communications sont limitées par la bande passante. Cependant, pour les applications qui nécessitent un traitement important et une grande capacité de stockage, le Cloud computing est plus approprié que le Fog computing car il a des serveurs plus puissants et une plus grande capacité de stockage. Le tableau 1.1 résume les différences entre VCC et VFC.

TABLE 1.1 – Comparaison entre le Cloud véhiculaire et le Fog véhiculaire

	Cloud véhiculaire	Fog véhiculaire
Latence	Moyenne à élevée	Faible
Consommation de la bande passante	Élevée	Faible
Distribution géographique	Non distribué, plutôt centralisé	Grandement distribué
Prise de décision	Se fait à distance	Se fait localement
Capacité de traitement	Grande	Moyenne
Capacité de stockage	Grande	Moyenne
Communication	Contrainte par la bande passante	Efficace grâce à l'équilibrage de charge
Proximité avec les utilisateurs finaux	Loin des utilisateurs	Proche des utilisateurs

1.5.6 Services offerts par le Cloud/Fog computing dans les réseaux véhiculaires

L'usage du Cloud et Fog computing dans les réseaux véhiculaires améliore grandement les performances de ceux-ci et offre un grand nombre de services en plus de ceux cités plus haut assurés par les VANETs. En effet, les applications de sécurité des conducteurs ainsi que des passagers sont grandement améliorées grâce aux ressources procurées par le Cloud [8]. La surveillance urbaine aussi a connu un progrès remarquable avec l'usage du Cloud. En effet, nous citons l'application de surveillance urbaine appelée MobEyes conçue par Bellavista et al. [43]. Cette application stocke les données détectées dans le noeud de stockage mobile du véhicule, et après cela, le processeur extrait les informations d'intérêt

et les envoie périodiquement au Cloud. Les agents mobiles peuvent ensuite rechercher ces informations auprès des véhicules voisins en cas de besoin. Ce Cloud contenant les données des véhicules peut également être recherché via Internet. De nombreuses applications de gestion du trafic aussi ont été conçues grâce au Cloud computing et Fog Computing dans les VANETs telles que le contrôle intelligent des feux de circulation, le freinage automatique,... etc. De plus, pour la surveillance et l'assistance des véhicules intelligents, Agarwal et al. [44] ont conçu une application d'heure d'arrivée "Time of Arrival (ToA)" pour les zones où les signaux GPS puissants ne sont pas disponibles et abordent également la prévention des collisions à l'aide du freinage automatique et de surveillance par caméra. Ils ont prototypé un matériel et l'ont testé sur des véhicules réels pour évaluer l'efficacité du système proposé. Les chercheurs ont également conçu une interface d'application mobile pour l'OBU pour un trafic intelligent et efficace et pour une surveillance du trafic à distance. Ils ont proposé des algorithmes pour le changement de voie basé sur la vitesse, l'évitement des collisions, la surveillance vidéo/photo, la détection des véhicules/accidents, la localisation basée sur les ToA et la diffusion des messages d'urgence.

L'utilisation du Cloud computing dans les VANETs assure également des applications de confort aux conducteurs [41]. Par exemple, sur une autoroute, au moment de payer, le conducteur peut payer automatiquement via le réseau en effectuant sa transaction directement en ligne depuis son compte bancaire sans s'arrêter et payer sa transaction en espèces. De plus, lorsqu'un conducteur a un long chemin à parcourir et éprouve le besoin de s'arrêter et de prendre un hôtel pour se reposer, il peut directement depuis sa voiture trouver l'hôtel le plus proche, à quel prix et voir s'il y a des chambres disponibles. Il peut directement réserver son hôtel depuis sa voiture en utilisant les ressources du véhicule et d'autres ressources fournies par le Cloud. La même chose aussi pour un restaurant ; quand le chauffeur veut manger, il peut chercher un bon restaurant directement depuis son véhicule. Il peut même consulter la carte du restaurant et décider de ce qui lui conviendrait. En outre, l'utilisation du Cloud computing dans l'environnement véhiculaire permet également des applications de divertissement [45] telles que la publicité commerciale, dans laquelle des annonces d'intérêt public sont publiées et des services multimédias comme la musique et la vidéo sont fournis pour les véhicules.

Les applications en temps réel [41] dans l'environnement véhiculaire se voient aussi grandement améliorées avec l'usage du Cloud et Fog computing. En effet, l'aide à la conduite automatisée devient de plus en plus exigeante en termes de ressources comme la puissance de calcul requise. L'utilisation du Cloud/Fog computing atténue ce problème et garantit un haut niveau de performance et un temps de traitement et de réponse court.

1.6 Conclusion

Dans ce chapitre, nous avons présenté les réseaux véhiculaires, leur architecture et communication ainsi que les services offerts par ce type de réseaux. Nous avons également parcouru le Cloud computing avec ses caractéristiques, ses modèles de déploiement ainsi que ses modèles de service. Puis, nous avons abordé le Fog computing avec ses caractéristiques, son architecture ainsi que ses domaines d'application. Par la suite, nous avons mis l'accent sur Cloud véhiculaire et le Fog véhiculaire avec leurs architectures respectives. Nous avons également survolé les domaines d'application des Cloud/Fog véhiculaires et avons présenté une comparaison exhaustive entre le Cloud et Fog véhiculaire en mettant

en avant les avantages de chaque architecture par rapport à l'autre. Et pour finir, nous avons présenté les services offerts par le Cloud computing et le Fog computing dans le domaine des réseaux ad hoc véhiculaires.

Dans le chapitre suivant, un état de l'art sur les solutions de routage dans les réseaux Cloud/Fog véhiculaires sera présenté. Cette étude nous permettra de tirer profit des avantages des meilleures contributions afin de proposer de nouvelles solutions efficaces dans le domaine du Cloud/Fog computing véhiculaire.

État de l'art sur la dissémination de
données dans les VANETs via le
Cloud/Fog computing

État de l'art sur la dissémination de données dans les VANETs via le Cloud/Fog computing

2.1 Introduction

Avec l'apparition des véhicules intelligents et l'émergence du domaine des VANETs, celui-ci est vite devenu gourmand en terme de capacité d'analyse, de traitement, de calcul, et d'espace de stockage. L'usage du Cloud computing en plus du Fog computing s'avère être une solution prometteuse pour répondre à ces exigences. Les communications inter-véhiculaires et les communications entre véhicules et Cloud et Fog ont donc considérablement évolué et ont donné aux VANETs une nouvelle dimension. Ces communications sont passées de simples communications entre véhicules situés dans la même zone géographique, aux communications entre véhicules dispersés sur les routes et les Cloud/Fog computing. De là sont apparus les problèmes de communication dans les réseaux Cloud véhiculaires "Vehicular Cloud (VC)" et Fog véhiculaires "Vehicular Fog (VF)". A cet effet, il faudrait donc bien établir ses besoins en termes de délai, de sécurité, de capacité de stockage, de vitesse de traitement, ... etc, et se référer à la solution de routage qui répond le mieux aux besoins de l'utilisateur avant de déployer son réseau VC ou son réseau VF. Par conséquent, nous avons trouvé utile de s'étaler sur des schémas de routage du Cloud computing et du Fog computing pour les VANETs dans ce chapitre et de faire une classification des méthodes de dissémination de données dans les réseaux Cloud/Fog véhiculaires.

2.2 État de l'art sur la dissémination de données dans les VANETs

La dissémination de données dans les VANETs est désignée par l'échange d'informations entre les différents acteurs du réseau véhiculaire. Cet échange se fait soit de véhicule à véhicule; ce qui désigne la communication V2V, soit de véhicule à diverses infrastructures du réseau véhiculaire telles que les RSUs, les serveurs Cloud, les noeuds Fog...etc;

et là, nous parlons de communication V2I. Dans ce contexte, de nombreux chercheurs ont passé en revue et reconsidéré les différentes techniques de dissémination de données dans les VANETs.

Dans [46], les auteurs ont passé en revue la transmission des données dans les VANETs en structurant leur étude en trois grandes catégories : l'approche de transfert de données géocast/broadcast où les véhicules diffusent des informations à tous les autres véhicules du réseau situés dans la même zone géographique, l'approche de transfert de données multicast où le véhicule envoie des données à un groupe spécifique de véhicules, et l'approche de transfert de données unicast où le véhicule expéditeur n'a qu'un seul destinataire pour un message donné.

Dans [47], les auteurs ont présenté une vue d'ensemble des techniques de diffusion existantes classées en trois modèles de base : basé sur la traction, basé sur la poussée, et hybride. Ensuite, les auteurs ont passé en revue les approches de modélisation des performances existantes pour analyser la diffusion de données dans les VANETs et ont étudié les stratégies d'optimisation de la dissémination de données.

Dans leur travail, Sharef et al. [48] ont discuté les principales caractéristiques et les principaux défis de la dissémination des données dans les VANETs. Ils ont proposé une taxonomie des protocoles de routage actuels, et ont comparé les instances symbolisées pour toutes les classes de ces protocoles.

Dans [49], les auteurs ont mené une étude comparative de la diffusion des données dans les réseaux véhiculaires. Ils se sont concentrés sur les caractéristiques de la mobilité bidirectionnelle des véhicules dans les VANETs et ont montré que les diffusions utilisant uniquement les directions opposées augmentent significativement les performances de la dissémination de données.

Dans [50], les auteurs ont passé en revue les trois positions où les données peuvent être traitées pour être transmises dans les VANETs ; à savoir, le chef de cluster (cluster-head), les unités en bordure de route (RSUs) et le Cloud. Ils ont proposé un schéma rentable pour sélectionner la meilleure position pour le traitement et la diffusion de données.

Le travail présenté dans [51] est dédié à la découverte de services où la diffusion de données devrait servir de base. Les auteurs ont proposé une taxonomie et une vue d'ensemble des approches du schéma de routage et ont classé les protocoles de diffusion de données les plus populaires disponibles en fonction des indicateurs proposés.

Dans [52], les auteurs ont analysé les performances des techniques de diffusion de messages de sécurité et de non-sécurité dans les VANETs basées sur la technique du Fog computing. Ils ont pris en compte trois principaux paramètres dans leur analyse ; à savoir, le délai, le débit et la probabilité de livraison des messages avec succès.

Dans [53], les auteurs ont abordé le rôle du Cloud computing dans les réseaux ad hoc véhiculaires. Ils ont souligné le fait que le Cloud computing améliore considérablement la diffusion des données, la gestion du trafic et le contrôle de la congestion.

Dans [54], les auteurs ont souligné l'importance de sélectionner le bon protocole de routage avant d'envoyer des données. Ils ont étudié différents protocoles de routage à la demande dans les systèmes de Cloud computing véhiculaires et se sont concentrés sur l'identification du protocole de routage à la demande le plus efficace qui peut offrir de meilleures performances dans des environnements réalistes.

Dans le travail présenté dans [55], les auteurs ont discuté des défis de la gestion économe en énergie et en temps réel des ressources distribuées des appareils mobiles dans un

environnement mobile. Ils ont présenté un ordonnanceur adaptatif à faible consommation d'énergie dans les réseaux Fog véhiculaires. Cet ordonnanceur exploite l'hétérogénéité des noeuds Fog pour optimiser l'énergie à partir du moment où le noeud Fog maximise le taux d'admission des tâches par rapport au calcul et au transfert de données. Ils ont conçu une architecture basée sur le Fog computing qui associe la mise en réseau, le transfert de données et la gestion des ressources informatiques et améliore la latence globale tolérée des traitements en plus de la communication.

Dans [56], les auteurs se sont concentrés sur la sécurité des données transmises dans le Cloud sur Internet et ont mis l'accent sur les problèmes de sécurité des données sur le multi-Cloud. Ils ont proposé la méthode PSDS "Proficient Security over Distributed Storage (PSDS)" dans laquelle les données sont divisées en deux catégories; données normales et données sensibles. Chaque partie est chiffrée et distribuée sur plusieurs Clouds. Les données normales sont téléchargées sur un seul Cloud sous forme chiffrée tandis que les données sensibles sont divisées en deux parties. Les deux parties sont chiffrées et téléchargées séparément sur différents Clouds.

2.3 Classification des schémas de dissémination de données dans les VANETs

2.3.1 Schémas de dissémination de données dans les VANETs

Les VANETs permettent la communication entre les véhicules et entre les véhicules et les dispositifs fixes à proximité (RSUs) [48]. Néanmoins, le principal problème des VANETs est d'assurer des communications fiables entre les véhicules en raison de la grande mobilité des noeuds et des changements fréquents de la topologie. Par conséquent, il est essentiel de trouver un chemin fiable par lequel les messages seront transmis avant de commencer le processus de communication entre les véhicules. Sharef et al [48] ont classé les protocoles de routage conçus pour les VANETs selon leur architecture de communication : communication V2V où les véhicules communiquent uniquement entre eux, et communication V2I où les véhicules communiquent avec les différentes infrastructures du réseau. De plus, le service de localisation est primordial pour la diffusion des données dans les VANETs puisque les noeuds du réseau se déplacent constamment et à une vitesse relativement élevée. De ce fait, la dissémination de données dépend grandement de la topologie du réseau.

a) Taxonomie des méthodes de dissémination de données dans les VANETs

Les méthodes de dissémination des données dans les VANETs sont classifiées en trois majeures catégories : la dissémination de données proactive, la dissémination de données réactive et la dissémination de données hybride. Chacune de ces trois catégories comprend un ensemble de protocoles pour la transmission efficace de données dans les VANETs.

1. Méthodes de dissémination de données proactives :

Les protocoles de dissémination de données proactifs implémentent des stratégies de distance vectorielle et d'état de lien pour trouver le chemin adéquat d'un noeud source à un noeud de destination en utilisant des algorithmes pour déterminer le

chemin le plus court tels que l'algorithme de Dijkstra [57]. Dans ce type de protocoles, les noeuds stockent des routes pour atteindre n'importe quelle destination dans le réseau. Cette stratégie de routage nécessite une maintenance continue des informations de topologie. De plus, le processus de diffusion de données qui utilise ces protocoles consomme une bande passante importante pour distribuer les mises à jour lorsque la topologie change. Ceci nécessite des capacités de stockage et de traitement considérables dans les grands réseaux. Les protocoles de routage proactifs les plus répandus sont "Destination-Sequenced Distance-Vector Routing (DSDV)" [58], "Greedy Perimeter Stateless Routing (GSRP)" [59], "Fisheye State Routing (FSR)" [60], "Optimized Link State Routing Protocol (OLSR)" [61], "Wireless Routing Protocol (WRP)" [62] et "Topology Broadcast based on Reverse-Path Forwarding (TBRPF)" [63].

2. Méthodes de dissémination de données réactives :

Les protocoles de dissémination de données réactifs sont des protocoles de transmission à la demande dans lesquels chaque noeud source ne conserve que la dernière route utilisée vers chaque noeud de destination. Dans ce genre de protocoles, pour trouver une route afin d'atteindre le noeud de destination à partir d'un noeud source, ce dernier inonde le réseau de messages "Route Request" et le noeud destination répond par le chemin inverse à l'aide de messages "Route Replay". Cependant, le principal inconvénient de ces protocoles est qu'ils doivent trouver rapidement une route alternative en cas de défaillance de la route utilisée; ce qui peut entraîner des retards de communication. Les principaux protocoles de routage réactifs sont "Temporally Ordered Routing Algorithm (TORA)" [64], "Ad hoc On Demand Distance Vector (AODV)" [65], "PRdiction based Ad hoc On Demand Distance Vector (PRAODV)" [66], "Dynamic Source Routing (DSR)" [67] et "Preferred Group Broadcasting (PGB)" [68].

3. Méthodes de dissémination de données hybrides :

Les protocoles de dissémination hybrides sont une combinaison de protocoles de dissémination réactifs et proactifs pour rendre le routage plus évolutif et efficace [48]. Ces protocoles sont proactifs avec les noeuds voisins directs et utilisent l'aspect réactif avec ceux qui sont plus éloignés. La caractéristique principale des protocoles de dissémination de données hybrides est qu'ils réduisent la surcharge du réseau causée par les protocoles de dissémination proactifs et le retard causé par les protocoles de dissémination réactifs. Ces protocoles effectuent la découverte de route plus efficacement [48]. Les principaux protocoles de dissémination hybrides sont Hybrid ad hoc routing protocol (HARP)" [69] et Zone Routing Protocol (ZRP)" [70].

b) Inconvénients de la dissémination de données dans les VANETs

Comme mentionné dans le travail de Dhankhar et Agrawal [71], des problèmes et des défis ouverts existent dans le routage de données dans les VANETs. Le principal problème dans la réalisation de la communication entre les véhicules et entre les véhicules et différentes infrastructures du réseau réside dans la perte de connectivité réseau due à la mobilité constante des véhicules et aux changements fréquents de la topologie. Par conséquent, il est essentiel de trouver un chemin qui puisse garantir un haut degré de

fiabilité pour délivrer les messages aux noeuds destinataires avant de commencer tout processus de transmission de données. Dans ce contexte, il est recommandé de choisir les routes qui contiennent le plus de véhicules entre les deux noeuds communicants afin d'assurer la connectivité entre les noeuds intermédiaires impliqués dans le processus de transmission de données.

2.3.2 Schémas de dissémination de données dans les réseaux Cloud véhiculaires

Dans le Cloud computing véhiculaire, la dissémination des données par les protocoles de routage devrait être plus efficace grâce aux ressources louées à d'autres véhicules ou au Cloud conventionnel, ce qui permet de meilleures performances en termes de capacité de calcul des meilleurs chemins et de prise de décision pour délivrer les messages, en plus du gain de temps considérable. En outre, l'utilisation du Cloud Computing dans les VANETs facilite plusieurs tâches pour les conducteurs et les autorités. Il sera plus facile d'acquérir certaines informations, qui sont souvent d'une importance primordiale, sans faire beaucoup de traitements, comme acheminer un grand nombre de paquets vers des véhicules dans la même zone et attendre la réponse, ce qui peut prendre du temps. Il suffit seulement de rechercher les données demandées dans le Cloud. Par exemple, il est plus commode pour un véhicule de rechercher directement les informations qui l'intéressent, comme la présence d'un embouteillage au prochain carrefour, au niveau du Cloud conventionnel dans lequel tous les véhicules en circulation ont stocké leurs données détectées et informations de trafic, plutôt que de rechercher ces informations chez les véhicules voisins. En effet, un véhicule peut demander à son véhicule voisin direct s'il dispose de l'information appropriée, à titre d'exemple l'état du trafic au prochain carrefour, son voisin peut également être trop éloigné du carrefour, il va donc demander à son tour à ses voisins s'ils ont les informations appropriées pour répondre au premier véhicule. Ce qui engendre une charge sur le réseau et un délai de réponse ainsi qu'un retard pour obtenir enfin l'information demandée.

De plus, de nombreuses applications sont désormais possibles et efficaces grâce à l'utilisation du Cloud computing dans l'environnement du véhicule, comme l'application de surveillance urbaine MobEyes [43]; le middleware de capteurs qui prend en charge les applications de surveillance urbaine proactive [39]. MobEyes détecte et traite les données pour la surveillance urbaine, il génère des métadonnées qui rajoutent aux données du capteur des informations contextuelles critiques telles que les horodatages et les coordonnées de position [39]. Ces métadonnées sont périodiquement transmises au Cloud conventionnel. Par la suite, si nécessaire, le Cloud contenant les informations du véhicule peut être consulté sur Internet grâce à l'application personnalisée MobEyes [39].

2.3.3 Schémas de dissémination de données dans les réseaux Fog véhiculaires

Lors de l'utilisation du Fog computing dans les réseaux véhiculaires, la communication devient de plus en plus efficace puisque les données sont traitées et stockées à la périphérie du réseau, ce qui facilite leur accès qui se fait sans difficulté et en un temps très court. Les applications de l'internet des objets sont particulièrement sensibles au retard et doivent

avoir une répartition géographique et une prise en compte de l'environnement local et de la mobilité des objets. Ceci rend l'utilisation du Fog computing très importante. De plus, le Fog computing est adapté à un grand nombre d'applications sensibles aux retards telles que les applications de sécurité routière et est aussi très approprié pour la diffusion des messages d'urgence. Par exemple, lorsqu'un accident de la circulation survient, des messages d'urgence sont envoyés et doivent être reçus en temps réel sans aucun délai pour sauver les personnes impliquées, et aussi pour informer les autres véhicules de l'état actuel de la route afin ne pas emprunter cette route et de dévier leur chemin pour atteindre leur destination. Par ailleurs, l'utilisation du Fog computing permet d'économiser de la bande passante et de ne pas surcharger le réseau. Par exemple, pour les petites applications qui n'ont pas besoin d'avoir une grande capacité de traitement et de stockage, l'utilisation du Fog computing est plus pratique ; elle ne gaspille pas de bande passante et l'application est servie très rapidement sans surcharge. Cependant, pour les données qui ne sont pas urgentes ou qui sont d'un volume élevé, comme le partage de films en haute résolution, il n'est pas judicieux d'utiliser des moyens coûteux de dissémination des données. La technique "Delay Tolerance Network (DTN)" [72] serait plus appropriée dans ce cas pour la dissémination de données et peut être utilisée pour transférer des données entre les serveurs du Fog et entre les véhicules et les noeuds Fog.

Dastjerdi et al. [33] ont comparé les performances d'une requête de détection d'accident de la circulation sur des infrastructures Fog par rapport à des infrastructures Cloud. Le résultat de cette comparaison montre clairement qu'une fois les opérateurs placés sur les dispositifs Fog, le délai de bout en bout est fortement réduit par rapport au délai de traitement dans le Cloud, car les données sont plus proches des sources. Ils ont également comparé l'utilisation du réseau central, qui est beaucoup plus faible dans l'utilisation du Fog que dans le Cloud. Ainsi, l'utilisation du Fog computing réduit considérablement la charge sur le réseau.

Gao et al. [73] quant à eux, ont proposé le modèle FogRoute ; un modèle hybride de dissémination de données qui tire profit de la technique DTN pour décharger le contenu entre les serveurs Fog soit par un être humain, soit par un véhicule. Ce modèle hybride comprend non seulement une dissémination normale des données entre les serveurs Cloud et les serveurs Fog, mais permet également une dissémination à grande échelle de données basée sur la technique DTN à faible coût pouvant être utilisée aussi bien entre les serveurs Fog et les utilisateurs mobiles qu'entre les serveurs Fog entre eux.

On peut résumer les avantages de l'utilisation du Fog computing dans les VANETs dans les points suivants :

- Il réduit le trafic sur le réseau.
- Il est adapté aux tâches et requêtes de l'IoT.
- Il répond aux exigences de faible latence.
- Il s'adapte à l'évolutivité du réseau.

2.3.4 Récapitulatif de la classification ci-dessus

Le tableau 2.1 suivant regroupe les trois schémas de dissémination de données et résume les propriétés de chacun d'entre eux.

TABLE 2.1 – Classification des schémas de dissémination de données dans les communications véhiculaires

Critères	VANETs	VCC	VFC
Dissémination de données	Proactive, Réactive, Hybride	Proactive, Réactive, Hybride	Réactive, Hybride
Latence	Moyenne	Moyenne	Très courte
Consommation de la bande passante	Moyenne	Élevée	Faible
Surcharge sur le réseau	Pas très importante	Importante	Pas très importante
Application en temps réel	Peu supportée	Peu supportée	Supportée
Grand traitement	Non supporté	Supporté	Non supporté
Distribution géographique	Distribué	Non distribué	Distribué
Prise de décision	Locale	À distance	Locale
Communication	Contrainte par la bande passante	Contrainte par la bande passante	Temps réel
Capacité de traitement	Moyenne	Très grande	Moyenne à grande
Capacité de stockage	Moyenne	Très grande	Moyenne à grande
Coût de déploiement	Bas	Élevé	Moyennement bas

2.4 Architecture de communication dans les réseaux Cloud/Fog véhiculaires

Dans cette section, nous présentons l'architecture de communication dans les réseaux Cloud/Fog véhiculaires tout mettant l'accent sur leurs avantages et leurs inconvénients.

2.4.1 Architecture générale des réseaux Cloud/Fog véhiculaires

D'une manière générale les réseaux véhiculaires qui se basent sur le Cloud et/ou le Fog computing reposent sur une architecture en couches composée principalement de trois niveaux :

- **La couche client** : Cette couche désigne l'utilisateur final du réseau ou l'application du Cloud computing véhiculaire. Il peut s'agir du véhicule lui-même, de tout objet connecté utilisant le réseau comme par exemple un smartphone, une tablette d'un passager du véhicule, ou de tout autre dispositif d'accès.
- **La couche communication** : Cette couche désigne le niveau intermédiaire entre l'utilisateur final et le Fog/Cloud conventionnel. Internet, réseau privé GPS, réseau 3G/4G, ou les RSUs peuvent tous être des supports de communication.

- **La couche Cloud** : Cette couche peut désigner le Cloud classique constitué de centres de données (data centers) stationnaires et virtualisés comprenant l'infrastructure de Cloud computing traditionnelle, ou bien le Cloud temporaire où sont implantés les serveurs et les centres de données pour répondre aux besoins des clients.

La figure 2.1 représente un exemple d'architecture Cloud/Fog véhiculaire.

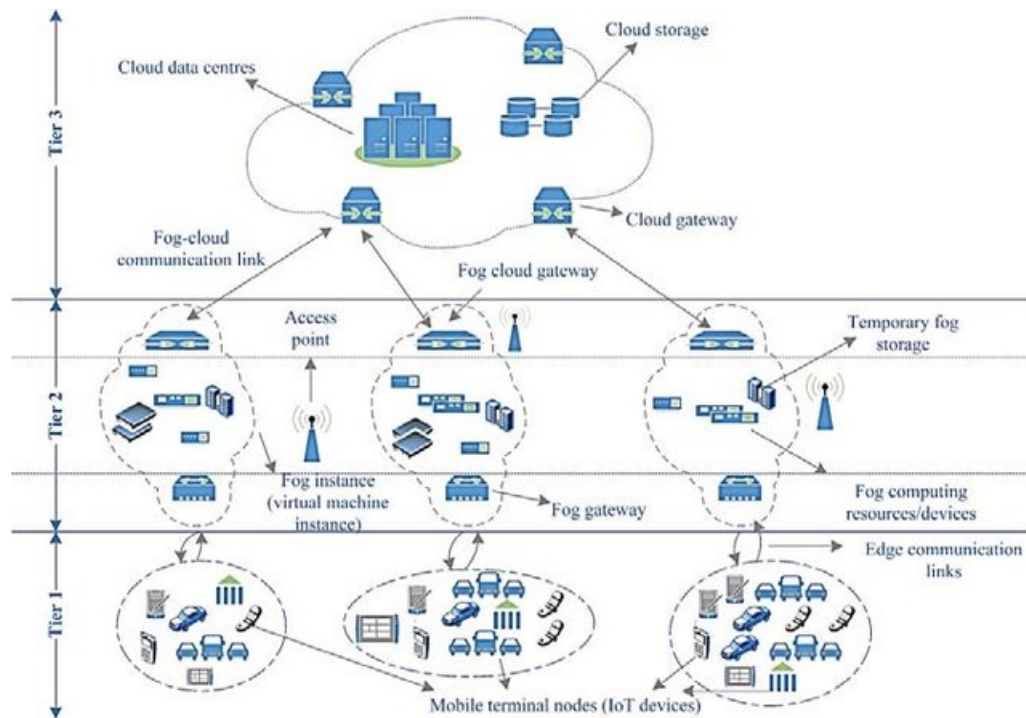


FIGURE 2.1 – Exemple d'architecture Cloud/Fog véhiculaire [10]

2.4.2 Architecture de communication des réseaux Cloud/Fog véhiculaires

Parmi les différentes architectures proposées dans le domaine des Cloud/Fog computing véhiculaires, nous citons les architectures suivantes avec leurs avantages et inconvénients.

a) Architecture de Masip-Bruin et al.

Masip-Bruin et al. [74] ont proposé une architecture "Fog-to-Cloud (F2C)" à niveaux pour assurer une gestion coordonnée des systèmes informatiques F2C où les services Cloud, Fog et réseau fonctionnent simultanément dans chaque scénario. Ainsi, les services peuvent être exécutés sur la périphérie du réseau, à la fois en parallèle et de manière coordonnée, tout en étant supportés par l'imparable évolution de la technologie.

Les principaux points forts de cette architecture sont :

- La dynamicité du modèle.

- L'aptitude du service exécuté sur l'appareil des utilisateurs qui permet de connaître laquelle des ressources du Fog/Cloud disponibles est la mieux adaptée.

L'inconvénient de cette architecture réside dans la nécessité de concevoir une gestion coordonnée pour relever certains défis tels que : l'identification des Clouds/Fogs, l'adaptation sémantique pour le partage des ressources, la garantie de la qualité de service "Quality of Service (QoS)", la planification de l'exécution des services, la sécurité et la confidentialité des données, la découverte et l'allocation des ressources, ainsi que la gestion transparente.

b) Architecture de Bitam et al.

Bitam et al. [75] ont proposé un modèle Cloud computing pour les réseaux véhiculaires appelé VANET-Cloud qui se base sur deux sous-modèles : les Clouds permanents et les Clouds temporaires, dans le but d'améliorer la sécurité routière et fournir différents services numériques tels que les logiciels, les infrastructures de calcul et les plateformes aux usagers de la route à un coût réduit.

Les points forts de cette architecture sont :

- L'intégration d'une nouvelle ressource informatique installée sur les véhicules, ce qui permet aux conducteurs d'accéder aux ressources en utilisant à la fois, les noeuds mobiles et les noeuds fixes ce qui réduit les coûts.
- Le modèle fournit les ressources du véhicule non seulement au conducteur mais également aux autres utilisateurs.

Quant aux inconvénients de cette architecture, ils résident dans les points suivants :

- L'allocation et le partage des ressources.
- Le problème de confidentialité et de sécurité.
- Les données d'agrégation et de détection.
- La coordination et la communication entre les sous-modèles de VANET-Cloud.
- La qualité des prestations fournies.
- L'interopérabilité et la standardisation.
- La consommation de quantité d'énergie importante.

c) Architecture de Meneguette [1]

C'est une architecture Cloud véhiculaire conçue pour aider à la gestion des grandes villes. Elle fournit un cadre qui permet de prendre en charge différents types de services ainsi que de fournir des mécanismes de stockage, d'accès et de gestion de l'information qui comprend des outils pour différents modes de transport non seulement pour les citoyens mais aussi pour les véhicules utilitaires et les services d'urgence comme les ambulances.

Les principaux points forts de cette architecture sont :

- Le Cloud omniprésent.
- Les véhicules choisissent les services Cloud de manière résiliente.
- La fiabilité et la continuité des services Cloud pour les véhicules.
- La rapidité des services.
- La possibilité de profiter pleinement des ressources du réseau.
- L'utilisation de diverses technologies de communication.
- L'utilisation de machines virtuelles dans le Cloud.
- L'allocation et la gestion des ressources.

Quant aux inconvénients de cette architecture, ils se situent dans :

- Le problème de partition réseau où l'on utilise l'approche sweet spot.
- La durée de vie du message qui peut affecter l'efficacité de la propagation.
- Dans la couche consommateur, seuls les véhicules sont pris en compte ; le cas des utilisateurs humains n'est pas pris en considération.

d) Architecture de Santa et al. [2]

Ce travail fait le point sur le déploiement de réseaux véhiculaires sur de véritables autoroutes en Espagne, au Portugal et en Grèce, et le fonctionnement des infrastructures de communication qui se basent sur les technologies IEEE 802.11p, 3G et "Internet Protocol version 6 (IPv6)" suivant les directives "International Organization for Standardization (ISO)" et "European Telecommunications Standards Institute (ETSI)" dans les systèmes coopératifs pour les communications de véhicule à l'infrastructure (V2I).

Les principaux points forts de cette architecture sont :

- Les VANETs déployés sur les autoroutes d'Espagne, du Portugal et de Grèce offrent un assez bon débit en utilisant le réseau 3G courant prenant en charge l'accès par paquets à haut débit "High-Speed Packet Access (HSPA)".
- Un large éventail d'applications V2I est possible. Ces applications pourraient être déployées sur l'architecture de réseau présentée telles que : le guidage de route, la limitation dynamique de vitesse, l'évitement et l'atténuation des embouteillages, la surveillance du trafic routier, ainsi que les applications de sécurité.

Tandis que les inconvénients relatifs à cette architecture sont :

- Réseaux basés sur la technologie de communication 3G et donc un manque de couverture est observé dans plusieurs zones des autoroutes où la technologie 802.11p n'est pas disponible.
- Pour les communications V2V, si les véhicules ne sont pas adressés IPv6, ils ne peuvent pas communiquer puisque les communications réseaux sont basés sur la technologie IPv6.
- Le coût des solutions proposées pour le problème de connectivité est élevé.

e) Architecture d'Agarwal et al.

Agarwal et al. [44] ont proposé une architecture qui permet la surveillance et l'assistance intelligentes des véhicules à l'aide du Cloud computing dans les VANETs. Elle introduit de nouvelles techniques qui ont été proposées pour servir le changement de voie basé sur la vitesse, l'évitement des collisions et la localisation basée sur l'heure d'arrivée (Time of Arrival (ToA)) dans les VANETs. Des applications de localisation ont été déployées basées sur un algorithme qui s'appuie sur l'heure d'arrivée (ToA) pour les zones où les signaux "Global Positioning System (GPS)" puissants ne sont pas disponibles. Cette architecture a aussi traité d'autres applications comme l'évitement des collisions à l'aide du freinage automatique et la surveillance urbaine par caméra.

Les avantages de cette architecture résident dans le fait que l'architecture "VCC" agit comme plateforme pour le changement de voie basé sur la vitesse, l'heure d'arrivée (ToA) et l'évitement des collisions à l'aide d'applications de freinage automatique et de surveillance par caméra.

Quant aux inconvénients, ils se situent dans :

- Les retards dans la diffusion des messages d'urgence.
- La sécurité du réseau VCC car aucun dispositif de sécurité n'a été mis en place.
- L'anticipation du comportement du conducteur face à une situation critique qui n'a pas été traitée.

f) Architecture de Yu et al.

L'architecture proposée par Yu et al. [76] intègre le Cloud computing dans les réseaux de véhicules de sorte que les véhicules puissent partager les ressources de calcul, de stockage, et de bande passante. Cette architecture comprend un Cloud véhiculaire, un Cloud en bordure de route et un Cloud central. Elle permet de traiter l'allocation des ressources Cloud et la migration des machines virtuelles pour une gestion efficace des ressources. Une approche basée sur la théorie des jeux est alors présentée pour allouer de manière optimale les ressources du Cloud. La migration de machines virtuelles en raison de la mobilité des véhicules est résolue sur la base d'un schéma de réservation de ressources.

Les principaux points forts de ce modèle sont :

- Le Cloud omniprésent.
- Les véhicules sélectionnent les services Cloud de manière résiliente.
- La rapidité des services.
- L'utilisation de diverses technologies de communication.
- Les ressources du réseau sont pleinement exploitées.
- La fiabilité et la continuité des services Cloud pour les véhicules.

Quant aux inconvénients de cette architecture, ils sont les suivants :

- Le problème de partition réseau où l'on utilise l'approche sweet spot.
- La durée de vie des messages qui affecte l'efficacité de la propagation.
- Seuls les véhicules sont pris en compte en tant que consommateurs, le cas des consommateurs humains n'est pas traité.

g) Architecture de Huang et al.

L'architecture de Huang et al. [9] intègre le Fog computing dans les réseaux VANETs, et un système de contrôle de trafic intelligent qui se base sur la stratégie du Fog comme cas d'utilisation concret pour tester la faisabilité et l'efficacité de ce modèle.

Les points forts de cette architecture sont :

- Le temps de réponse court ce qui réduit la latence.
- Le volume et la fréquence des données de communication réduits.
- La capacité de stockage disponible.

Tandis que les points faibles de cette architecture sont :

- La sécurité : comment détecter le flux de trafic anormal et comment détecter les noeuds Fog compromis en se basant sur un volume important des données de trafic.
- Comment trouver un équilibre efficace entre fonctionnalité, sécurité et confidentialité dans une application spécifique du véhicule.

h) Architecture de Pasha et al.

L'architecture de Pasha et al. [77] intègre le Cloud computing dans les réseaux véhiculaires en se basant sur la capacité des noeuds véhiculaires à utiliser les ressources en termes de puissance de calcul embarquée et de ressources de stockage et de communication pour interagir avec les différents éléments cyber-physiques possibles en formant ainsi un réseau Cloud véhiculaire.

Son principal point fort réside dans le fait que le modèle VCC emprunte son architecture au Cloud computing mobile, et en tire ainsi pleinement parti. Quant à ses inconvénients, ils résident dans :

- Le manque de coordination et de communication.
- Le besoin de développement et d'amélioration des plateformes middleware.
- Le besoin de création d'un environnement de confiance pour les éléments du véhicule sur internet.

i) **Architecture de Hou et al. [3]**

Hou et al. [3] ont proposé une architecture basée le Fog véhiculaire "VFC". Cette architecture utilise les véhicules comme infrastructures de communication et de calcul. Elle utilise une multitude collaborative de clients utilisateurs finaux ou de dispositifs proches de l'utilisateur pour effectuer la communication et le calcul sur la base d'une meilleure utilisation des ressources de communication et de calcul individuelles de chaque véhicule.

Les principaux points forts de cette architecture sont :

- La bonne utilisation des moyens de communication et de calcul individuels de chaque véhicule.
- Une bonne QoS et des applications grâce à l'agrégation des ressources abondantes des véhicules individuels.
- La construction de modèles adaptés à la mobilité des véhicules.

Les points faibles de cette architecture sont :

- Pas d'analyse de la capacité de calcul.
- Pas de construction de système d'exploitation pratique.
- Pas de gestion des problèmes de sécurité et des applications associées.

Le tableau 2.2 résume les caractéristiques des architectures de communication VCC et VFC.

TABLE 2.2 – Récapitulatif des caractéristiques des architectures VCC et VFC

	VCC	VFC
Fiabilité	++	+
Latence	-	++
Consommation de la bande passante	++	+
Distribution géographique	-	++
Capacité de calcul	++	+
Capacité de stockage	++	+
Sécurité	++	+
Robustesse	++	-
Rapidité de communication	-	++

2.5 Solutions de routage dans les VANETs basées sur le Cloud et le Fog computing

Dans un environnement VC/VF, le routage et la dissémination des données peuvent être difficiles car la connectivité est intermittente en raison de la grande mobilité des véhicules, des changements fréquents dans la topologie du réseau, et de la densité des véhicules. Ces paramètres peuvent influencer le succès de la livraison des messages et augmenter le taux de perte de paquets. Par conséquent, pour surmonter ce défi, les chercheurs se sont concentrés pour trouver un compromis entre la mobilité des noeuds, la connexion discontinue et la garantie de la livraison des messages tout en minimisant les délais de livraison. Dans cette section, nous étudions différentes solutions de routage basées sur le Cloud et le Fog computing puis nous présentons une synthèse générale de ces solutions de routage sous la forme d'un tableau récapitulatif.

2.5.1 Solutions de routage basées sur le Cloud computing

Cette partie s'intéresse à la diffusion de données dans les VANETs qui se base sur le Cloud computing. Pour cela, nous étudions les principaux protocoles de diffusion de données basés sur le Cloud computing dans les réseaux véhiculaires.

- **Bitam et Mellouk [78]** ont proposé un protocole de routage géographique qui s'appuie sur le Cloud computing pour assurer la diffusion des données dans les VANETs appelé ClouDiV. Le protocole proposé prévoit une diffusion adaptative de tous les types de messages, nécessitant ou non une sécurité via une architecture de Cloud computing. Il est considéré comme un protocole hybride de diffusion de messages dans lequel une approche proactive est utilisée par le centre de données (data center) pour découvrir les routes récentes et les mises à jour en générant un paquet de contrôle appelé "Routing Information REQuest (RI-REQ)" pour recueillir les informations de routage. Chaque noeud reçoit un message RI-REQ et génère un paquet de contrôle appelé "Routing Information REPLY (RI-REP)" contenant les informations de routage demandées. L'approche réactive est effectuée par chaque véhicule pour trouver le plus proche centre de données comme noeud intermédiaire. Le véhicule source génère un paquet pour demander des informations relatives aux routes appelé Route REQuest (RREQ), qu'il envoie à ses voisins directs. Ceux-ci vérifient s'ils ont la route demandée dans leur table de routage. Si ce n'est pas le cas, ils rediffusent le paquet RREQ à leurs voisins directs de la même manière. Lorsque que la route demandée est trouvée, un paquet de réponse appelé Route REPLY (RREP) est généré et transmis au noeud source. Des simulations sur des scénarios urbains pour évaluer les performances du protocole ClouDiv ont été menées sur le simulateur de réseau NS-2, étendu par le framework green-Cloud [79] pour l'infrastructure du Cloud computing. Les performances du le protocole ClouDiv ont été mesurées selon plusieurs métriques telles que le délai moyen de bout en bout, la surcharge normalisée "Normalized OverLoad (NOLIP)" et le taux de livraison des paquets "Packet Delivery Ratio (PDR)". Les auteurs ont comparé ces performances avec le protocole "Reliable Inter-VEhicular Routing (RIVER)" [80]; les évaluations ont montré clairement la diminution du délai de bout en bout et de la surcharge

normalisée ainsi que l'augmentation du PDR. En effet, pour 75 noeuds, le délai moyen de bout en bout avec ClouDiv est de 1,852 s, tandis qu'avec le protocole RIVER, il est de 2,869 s. La NOL pour ClouDiv étant de 58,52 contre 69,07 pour RIVER, et le PDR, quant à lui, est de 97,73% pour ClouDiv, alors qu'il est de 96,37% pour RIVER.

- **Kim et al. [81]** ont proposé un schéma de prélecture de données basé sur les itinéraires des véhicules afin de maximiser la probabilité de diffusion réussie de données lorsque la taille du stockage local des données est limitée et que la connectivité sans fil n'est pas garantie. Ils ont conçu un algorithme glouton pour les cas déterministes et un algorithme d'apprentissage en ligne pour les cas stochastiques afin de décider comment pré-extraire les données d'intérêt d'un centre de données vers des points d'accès sans fil "Access Point (AP)" routiers. Ils ont envisagé un système Cloud véhiculaire qui se compose de serveurs Cloud dans un centre de données et de points d'accès sans fil routiers avec stockage de données local en supposant qu'il existe une limite maximale de stockage de données dans les AP sans fil routiers. Chaque point d'accès prélève certaines données du centre de données avant qu'elles ne soient demandées par le véhicule. Le problème de diffusion des données est formulé comme un problème d'optimisation combinatoire qui maximise la probabilité de succès de la diffusion des données agrégées. L'évaluation des performances des algorithmes a été réalisée en situation réelle (testbed) en utilisant la trace de véhicule de 2 060 taxis grâce aux traces GPS et 40 points d'accès routiers déployés aléatoirement aux carrefours des routes à Pékin. L'algorithme glouton pourrait trouver une solution sous-optimale en un temps polynomial. L'algorithme d'apprentissage en ligne basé sur le bandit multi-armé "Multi-Armed Bandit (MAB)" détermine une matrice de décision binaire optimale en apprenant progressivement à chaque itération, le taux de réussite de la connectivité réseau inconnu. La solution proposée converge vers une solution globalement optimale après un certain nombre d'itérations.
- **Nkenyereye et al. [82]** ont abordé le problème de la dissémination sécurisée des données dans un environnement extrêmement mobile et imprévisible. Ils ont utilisé des techniques de pseudonymes telles que la signature basée sur l'ID, les informations d'identification anonymes, la technique de vérification par lots et la liste de révocation de pseudonymes pour assurer la confidentialité de l'identité des véhicules et de leurs messages générés. L'objectif principal des auteurs était de répondre aux exigences de sécurité telles que la préservation de la confidentialité de l'identité, l'authentification, la confidentialité, la traçabilité, et la résistance aux attaques courantes lors de la diffusion des données. Pour ce faire, le protocole implique certaines entités : l'autorité de confiance "Trusted Authority (TA)" ; en charge de l'enregistrement des autres entités dans le système et de la transmission du matériel cryptographique lors de l'initialisation du système, les Clouds en bordure de route "RoadSide Cloud (RSC)" ; situés le long de la route et servent de bases de données accessibles aux utilisateurs, et le centre de gestion du trafic "Traffic Management Center (TMC)" qui est un serveur situé dans le Cloud conventionnel. Les véhicules diffusent périodiquement un message de sécurité de base "Basic Security Message (BSM)" composé d'éléments et de trames de données. Lorsque le RSC reçoit les messages BSM, il stocke tous les éléments réussis dans sa base de données et fixe un

intervalle d'extraction Tmi allant de 100 à 1000 ms selon le débit de la zone et calcule la densité de véhicules au temps T dans sa zone de couverture. Par la suite, le RSC génère un fichier d'estimation de densité $FTA = \{k_{t_1}, k_{t_2}, k_{t_3}, \dots, k_{t_n}\}$ représentant les informations d'estimation de densité aux instants t_1, t_2, \dots, t_n . Le RSC compose $m_1 = \{FTA\}$ pour le voisinage et envoie $\{\delta_2, C_2\}$ où $C_2 = Enc_{IDRSC}(m_1)$ est le cryptogramme de m_1 sous la clé publique du recevant RSC et δ_2 est une signature générée sur m_1 . Le RSC envoie également $C_3 = Enc_e(m_1)$ au TMC où e est une clé partagée symétrique telle que Advanced Encryption Standard (AES). TMC reçoit les fichiers de différents RSC et prédit la densité ou la congestion en appliquant des techniques d'estimation. Il envoie ensuite ses estimations de densité calculées au RSC pour diffusion. L'efficacité de ce protocole a été démontrée par des simulations avec le simulateur VANET-SIM pour la mobilité des véhicules couplé au simulateur "Network Simulator 3 (NS-3)" pour la simulation de réseau. Les auteurs concluent que les objectifs de sécurité fixés ont été atteints.

- **Dressler et al. [83]** ont proposé un Cloud véhiculaire utilisant le protocole "Virtual Cord Protocol (VCP)"; une solution de routage basée sur les coordonnées virtuelles pour permettre des services Cloud dynamiques distribués et intégrer le routage gourmand. Ils ont utilisé les ressources des véhicules garés pour établir un Cloud véhiculaire pour le stockage des données et la fourniture de connectivité réseau. La principale motivation dans le développement de ce protocole était de trouver une technique efficace et tolérante aux pannes pour le routage avec peu de surcharge. Le protocole VCP offre des services de table de hachage distribuée "Distributed Hash Table (DHT)" et de routage glouton intégré avec livraison garantie et fournit un stockage de données déterministe. Pour configurer le cordon virtuel connectant tous les noeuds du réseau, chacun des noeuds diffuse périodiquement un message Hello qui inclut la position de l'expéditeur, son prédécesseur et son successeur dans le cordon. Dès réception du message Hello, le noeud met à jour sa table de voisinage en ajoutant l'entrée correspondante. Pour rejoindre un cordon, le noeud doit recevoir au moins un message Hello d'un noeud ayant une position dans le cordon. Les données sont ensuite transmises sur la base des informations de voisinage en combinaison avec la structure du cordon à l'aide d'un routage gourmand. Les auteurs ont considéré les véhicules en stationnement pour établir un Cloud véhiculaire dynamique afin de fournir des capacités de stockage et de réseau. Chaque fois qu'une voiture se gare, soit elle crée un nouveau Cloud, soit elle rejoint un Cloud existant. La procédure repose sur deux opérations de base : la configuration et la gestion des réseaux VCP couvrant un groupe de véhicules en stationnement connectés, et l'intégration temporaire des véhicules en mouvement dans le Cloud véhiculaire pour télécharger et téléverser des informations. La gestion des données est basée sur la fonctionnalité DHT. L'application fournit les opérations *LOOKUP* et *PUBLISH* pour rechercher des données ou pour les stocker respectivement, et le routage gourmand de VCP est utilisé pour transmettre les messages au domaine et au noeud de destination. Une fois un message *LOOKUP* reçu, l'application hache l'identifiant des données demandées et charge les données depuis son stockage local. S'il n'y a pas de données correspondantes, la demande est ignorée. Et si, au contraire, la donnée est trouvée, elle est encapsulée dans un message *DATA* et est transmise au noeud demandeur. Lors de la réception d'un message *PUBLISH*, l'application mappe également les

données vers le magasin de stockage local et enregistre les pointeurs. Si le magasin de données a déjà atteint sa capacité maximale, la demande est abandonnée sinon, l'application envoie un message *ACKNOWLEDGE* au noeud source. L'implémentation du VCP proposé a été réalisée sur le simulateur OMNET++ en utilisant le framework Veins [84] avec "Simulation of Urban MObility (SUMO)". Les résultats de simulation indiquent clairement la faisabilité de cette approche, qui est capable de faire face à la dynamique des véhicules qui partent même aux heures de pointe. Utilisant un délai d'attente de 100s, cette solution de Cloud dynamique est capable d'intégrer au moins 80% des véhicules du parking dans le réseau. Le taux de réussite augmente également avec une forte mobilité des véhicules. Le nombre de véhicules ayant un taux de réussite supérieur à 25% est environ 5% plus élevé à une vitesse de 50km/h qu'à une vitesse de 20km/h. La performance globale ne dépend donc que marginalement de la vitesse des véhicules.

- **Chaqfeh et al. [85]** ont proposé un nouveau modèle de collecte de données dans le Cloud véhiculaire pour les systèmes de transport intelligents "ITS". Ils ont exploité les avantages du Cloud computing véhiculaire pour améliorer l'efficacité du routage des données en proposant une architecture en clusters où le réseau est divisé en plusieurs clusters. De plus, s'il n'y a pas d'infrastructure, la diffusion des données se fera à l'aide des protocoles de routage dédiés aux VANETs. Ces protocoles visent généralement à pallier le problème de la grande mobilité des véhicules qui entraîne une topologie très dynamique et des déconnexions fréquentes, en utilisant le paradigme de la diffusion dirigée ou le routage géographique. Dans ce travail, les auteurs ont illustré via des simulations que la participation d'un faible pourcentage de véhicules dans un Cloud véhiculaire dynamique est suffisante pour offrir des collectes de données significatives pour les ITS. Dans ce protocole, un véhicule désigne à l'avance ses préférences pour un service de collecte de données, afin qu'il puisse demander des informations sur le trafic à la demande. Un véhicule qui recherche des informations sur le trafic définit sa destination souhaitée et crée un message de demande d'itinéraire appelé Route Request (*RREQ*) et le diffuse à ses voisins directs. La destination souhaitée, appelée "Region Of Interest (ROI)", peut être étendue à une carte routière détaillée ou limitée à un seul segment de route. Lorsque dans la *ROI* souhaitée, le message *RREQ* est reçu par les véhicules, ceux qui souscrivent au même service appelé véhicules coopérants "Cooperating Vehicle (CV)" et vont coopérer pour fournir une réponse au véhicule demandeur en annonçant au réseau les données de capteur requises. Les CV essaient de former un Cloud dynamique pour publier leurs données de capteurs et sont en compétition pour être élus en tant que courtier. Le courtier collecte les données et communique avec l'infrastructure. Il forme un Cloud véhiculaire (VC) et ce dernier accumule ses ressources informatiques pour accomplir la tâche et répondre à la demande. Si les ressources du VC ne sont pas suffisantes pour fournir une réponse, le courtier stocke les données dans une table de hachage et les envoie à un serveur dans le Cloud d'Internet pour un traitement ultérieur et le serveur du Cloud d'Internet se chargera de répondre au courtier. Sinon, en cas de demande simple, les ressources du VC peuvent être suffisantes et une fois la réponse est prête, le courtier crée un message de réponse de route (*RREP*) et le transmet au véhicule demandeur via un autre VC ou via le réseau des VANETs.

Le scénario proposé a été implémenté à l'aide d'OMNET++ avec Veins et SUMO. Les résultats de la simulation indiquent clairement que la vitesse moyenne recueillie dans le cadre d'un trafic moyen n'est pas très éloignée de la moyenne globale sous tous les taux de pénétration. Le meilleur résultat est obtenu lorsque 90% des véhicules participent à la collecte de données, mais même si le pourcentage de véhicules participants est faible, les approches véhiculaires basées sur le Cloud peuvent soutenir efficacement les ITS avec des données en temps réel.

- **He et al. [86]** ont proposé "Modified Particle Swarm Optimization-CONstrained (MPSO-CO)"; un algorithme d'optimisation basé sur le réseau "Software Defined Network (SDN)" avec contrainte d'optimisation par essaims de particules pour résoudre les problèmes de latence de traitement élevée, de faible support de mobilité et de connaissance de l'emplacement dans une architecture de réseau Cloud/Fog définie par logiciel "Software-Defined Cloud/Fog Networking (SDCFN)" dans les environnements d'Internet des véhicules (IoV). MPSO-CO combine le SDN et le Fog computing pour surmonter ces problèmes. Le Fog computing permet d'étendre le calcul et le stockage à la périphérie du réseau, ce qui pourrait réduire la latence et permettre la prise en charge de la mobilité et la connaissance de l'emplacement, tandis que le SDN permet de fournir un contrôle centralisé flexible et une connaissance globale du réseau. Les résultats de la simulation montrent que l'algorithme MPSO-CO réduit la latence et améliore la qualité de service dans l'architecture SDCFN. Dans ce protocole, lorsqu'une tâche est transférée des utilisateurs finaux aux RSUs et à la station de base "Base Station (BS)", elle est divisée en un ensemble de sous-tâches et chaque sous-tâche est transférée au noeud du Cloud ou au noeud du fog pour le traitement. Enfin, les résultats des sous-tâches sont fusionnés et renvoyés aux utilisateurs finaux. D'où le problème de latence dans le traitement des demandes dont la formulation est représentée par l'équation (2.1).

$$t(x) = \max\left\{\frac{u_1}{C_{v_1}} + \tau_{v_1,v_1}m_{v_1,v_1}, \dots, \frac{u_k}{C_{v_k}} + \tau_{v_1,v_k}m_{v_1,v_k}\right\} \quad (2.1)$$

où $t(u)$ est la latence, u_i est le numéro de la sous-tâche I , et C_{v_i} est la capacité de calcul du noeud Fog (FN) et $\tau_{v_i,v_j}m_{v_i,v_j}$ désigne la latence de communication entre les FNs v_i et v_j . Elle est considérée comme un problème d'optimisation formulé par l'équation (2.2).

$$u = \arg_{u \in I} \min\{t(u)\} \quad (2.2)$$

Les auteurs ont utilisé l'optimisation sous contrainte d'essaims de particules "Particle Swarm Optimization-CONstrained (PSO-CO)" [87] pour résoudre ce problème d'optimisation car PSO a l'avantage d'avoir une convergence rapide, une forte capacité de recherche globale et une programmation facile. L'algorithme a été mis en oeuvre par simulation et comparé à d'autres algorithmes d'équilibrage de charge tels que l'algorithme glouton d'équilibrage de charge (Greedy-LB) [88], et l'algorithme d'équilibrage charge Max-Min "Load Balancing Max-Min (LBMM)" [89]. Les résultats obtenus montrent que l'algorithme MPSO-CO diminue la latence et améliore la QoS dans l'architecture SDCFN par rapport à ces algorithmes.

- **Qin et al. [90]** ont conçu une architecture Cloud computing appelée VehiCloud pour le routage de paquets inter-véhicules afin de traiter le problème de communications inter-véhicules peu fiables et d'étendre les capacités de calcul limitées des dispositifs mobiles. Dans l'architecture VehiCloud, un framework d'information sur les points de cheminement "Waypoint Information Framework (WIF)" est conçu pour offrir un service de routage pour les réseaux VANETs, où chaque véhicule agit comme un noeud de service mobile "Mobile Service Node (MSN)" et prévoit ses emplacements futurs en produisant des messages de points de passage, qui décrivent la trajectoire du véhicule. De plus, dans VehiCloud, un module de décision "Decision Module (DM)" collecte les points de passage des véhicules et prend des décisions de routage pour les communications V2V. Un point de passage représente une paire de deux valeurs : l'heure (heure actuelle ou future) et la position du véhicule correspondant à cette heure. Il est défini par un tuple de cinq attributs : identifiant du véhicule (vid), temps de génération (gt), temps futur (ft), position future (fp) et probabilité (pb). Une séquence de points de passage est générée dans un vecteur de points de passage (W-Vector) et le véhicule l'envoie à ses terminaux les plus proches qui transmettent ce W-vector au DM du VehiCloud. Une fois les points de passage collectés sur les véhicules et l'emplacement de tous les terminaux fixes, le DM construit un graphe de liens espace-temps "Time-Space Link Graph (TSLG)" conçu pour cartographier l'évolution de la topologie du réseau (sur une période de temps) en un seul graphe afin de prendre la décision d'acheminement. Lorsqu'un point de cheminement expire, les sommets correspondants et les arêtes associées dans le TSLG doivent être supprimés et les nouveaux points de cheminement, dès qu'ils sont collectés, sont incorporés dans le graphe. Une fois le TSLG construit, le DM modélise le routage comme un problème de recherche de chemin. Tout d'abord, le DM détermine le sommet source et le sommet de destination en fonction des informations courantes de la session demandée. Il détermine alors la taille maximale des données pouvant être transférées, le délai maximal de bout en bout acceptable, et le coût maximal que peut atteindre cette opération de transfert. Par conséquent, le problème complexe de routage du réseau véhiculaire est modélisé comme un problème de recherche du chemin le plus court et les auteurs ont utilisé un solveur de programme linéaire pour résoudre ce problème en temps polynomial. Le chemin résultant est une route globalement optimisée en termes de ratio de livraison de messages, avec un délai de bout en bout et un coût de livraison de messages délimités. Les auteurs ont mené des expériences dans le monde réel pour prouver l'efficacité de VehiCloud. L'architecture a été mise en oeuvre dans une zone résidentielle et l'expérience a démontré la capacité de VehiCloud à effectuer un routage fiable en tenant compte de contraintes variantes.
- **Hussain et al. [91]** ont proposé dans leur solution trois grandes entités Cloud ; Cloud véhiculaire (VC), Véhicules utilisant le Cloud "Vehicle using Cloud (VuC)" et Cloud hybride "Hybrid Cloud (HC)". Ils ont divisé le VC en deux scénarios : le premier est un Cloud statique, qui fait référence à des véhicules stationnaires et fournit des services Cloud, et le second est un Cloud dynamique configuré à la demande de manière ad hoc. VuC permet aux véhicules de se connecter au Cloud via des RSUs tandis que le HC combine le VC et le VuC. Ce modèle proposé suppose que les ressources des véhicules n'appartiennent au VC que si ces véhicules sont à l'arrêt. Pour

former un VC, les véhicules lancent d'abord un protocole pour sélectionner le ou les courtiers parmi eux et identifier les limites à ne pas dépasser des Clouds en élisant parmi les courtiers, une entité autorisée "Authorized Entity (AE)". L'AE invite ensuite les nœuds des véhicules situés à la frontière du Cloud à participer au Cloud. Les véhicules intéressés répondent par un accusé de réception. Les participants au Cloud mettent ainsi en commun leurs ressources pour former un environnement virtuel riche. L'AE dissout le Cloud lorsque le travail est terminé. Dans VuC, le réseau VANET utilise des services Cloud et les RSUs agissent comme des passerelles entre les véhicules et les services Cloud. Une communication filaire à haut débit entre les RSUs et les services Cloud est possible. Le HC est la combinaison du VC et du VuC où le VC sert à la fois de fournisseur de services pour d'autres véhicules et de consommateur. Par exemple ; un véhicule peut servir de point d'accès pour d'autres utilisateurs sur le VC et utiliser en même temps les ressources du Cloud classique pour certains traitements.

- **Abid et al. [92]** ont proposé l'architecture V-Cloud. Cette architecture comprend trois couches : système cyber-physique véhiculaire "Vehicular Cyber-Physical System (VCPS)", réseau V2V et réseau V2I. Le VCPS assure la sécurité du conducteur, notamment par la conscience du contexte, la détection de l'humeur du conducteur, et la surveillance des soins de santé du conducteur pendant la conduite. V-Cloud offre des services en temps réel et pourrait être étendu pour profiter des ressources installées sur les OBUs des véhicules. Les véhicules de cette architecture forment un cluster dans lequel ils communiquent entre eux et avec d'autres infrastructures sur la route via la communication dédiée à courte portée "DSRC", les réseaux cellulaires (WiMax/3G/4G) et le WiFi. Les smartphones des conducteurs sont utilisés à l'intérieur des véhicules et servent à surveiller les conditions d'humeur et la santé du conducteur grâce à des capteurs intégrés. Ils envoient également les informations recueillies sur le conducteur au Cloud. Ce dernier stocke l'historique à long terme du conducteur, ce qui lui permet de prédire l'humeur du conducteur à l'avance et d'informer le système d'intervention d'urgence en cas de maladie ou d'accident.
- **Mershad et Artail [93]** ont proposé un protocole de découverte de services Cloud pour les VANETs appelé CROWN. Dans ce système, les RSUs sont utilisées comme des répertoires Cloud. Elles stockent des informations sur les ressources informatiques disponibles des véhicules dans leur zone de transmission. En outre, une RSU stocke pour chaque STAR (serveur Cloud mobile) le type de ressources qu'elle offre, les attributs de chaque ressource et le prix requis par unité de ressource. Ensuite, chaque RSU partage ces informations avec les autres RSUs pour permettre aux véhicules de découvrir et de consommer les services des serveurs Cloud mobiles. Ce protocole présente les STARs comme des serveurs Cloud mobiles. Lorsqu'un véhicule souhaite participer au Cloud véhiculaire en tant que STAR, il définit les ressources qu'il a à offrir ainsi que les attributs relatifs à chaque ressource. S'il peut offrir un accès à Internet ou un réseau en tant que service "Network as a Service (NaaS)" par exemple, le STAR doit préciser ses paramètres de QoS, son délai d'accès (Da) et sa bande passante "BandWidth (BW)". Le STAR fixe également le prix qu'il facturera pour ses services ; par exemple, le coût d'accès par heure (Ca). Si le STAR propose un logiciel en tant que service SaaS, il doit définir la quantité maximale de stockage

par consommateur qu'il peut offrir (S_m), ainsi que les heures pendant lesquelles ses ressources seront disponibles (TSTAR). En outre, le STAR définit la durée globale maximale pendant laquelle il peut prêter sa capacité de stockage à un consommateur (T_t) et le prix par unité de stockage (C_s). Enfin, s'il est en mesure de fournir des données en tant que service "Data as a Service (DaaS)", il définit les types de données qu'il peut mettre en cache (DT), la capacité de stockage maximale qu'il peut offrir par utilisateur (D_c) et le coût par unité de données (C_d). Ensuite, le STAR formule un paquet d'enregistrement qui contient toutes les ressources qu'il peut offrir avec leurs attributs et envoie ce paquet à sa RSU la plus proche. Une fois que la RSU reçoit le paquet d'enregistrement, elle définit une zone appelée zone d'influence du STAR ; la zone dans laquelle le STAR offre efficacement ses ressources aux consommateurs. Ensuite, elle calcule les ID de toutes les RSUs qui sont dans la zone d'influence du STAR, en utilisant la carte numérique et envoie les données d'enregistrement (RD) du STAR à ces RSUs. Chaque RSU ajoute ce RD à son cache, ce qui permet au STAR d'être connu par toutes les RSUs qui se trouvent dans la zone d'influence du STAR. Lorsqu'un utilisateur est intéressé par une donnée, il envoie une requête à la RSU la plus proche en spécifiant dans son paquet de requête les ressources désirées avec leurs attributs et ses coordonnées géographiques. Dès que la RSU reçoit un paquet de requête de l'utilisateur, elle recherche dans son cache un ou plusieurs STARs qui pourraient satisfaire les besoins de l'utilisateur. Pour répondre à l'utilisateur demandeur, la RSU choisit le ou les STAR(s) correspondant(s) en fonction de la ou des valeurs des ressources et formule un paquet de réponse qui contient : l'ID du STAR, la dernière localisation du STAR, la dernière zone d'influence estimée du STAR et les ressources offertes par le STAR avec les attributs de chaque ressource. L'utilisateur choisit ensuite un ou plusieurs STAR à partir du paquet de relecture et spécifie les ressources avec leurs attributs qu'il demande à chaque STAR et envoie un paquet de service contenant sa demande au STAR choisi. Lorsqu'un STAR reçoit le paquet de service, il répondra soit par un accusé de réception négatif s'il ne peut pas satisfaire la demande de l'utilisateur, soit par un paquet de réponse de service dans le cas contraire. Les auteurs ont implémenté le protocole CROWN en utilisant le simulateur NS-2 avec SUMO pour générer la mobilité des véhicules. Comme CROWN est le premier protocole de découverte de services Cloud proposé pour le Cloud véhiculaire, ils le comparent à un protocole de découverte de services Cloud dans lequel ils remplacent les diverses opérations de CROWN par des opérations de diffusion. Ils ont prouvé son efficacité et ont conclu que la distance entre l'utilisateur et le STAR qui va le servir est le facteur le plus important qui affecte à la fois CROWN et le protocole de diffusion-CROWN.

- **Baby et al. [94]** ont présenté un Cloud véhiculaire pour les scénarios de bord de route "Vehicular Cloud for Road side scenarios (VCR)". Le VCR fournit des services de sécurité et de non-sécurité dans les applications véhiculaires par le biais de Clouds véhiculaires privés et publics. L'architecture proposée assure la communication complète entre V2V et V2I et peut être divisée en deux : la communication par le biais du Cloud public et la communication par le biais du Cloud privé. Le Cloud privé est formé par les OBUs des véhicules pour maintenir une mise à jour fréquente de leurs informations à proximité, tandis que le Cloud public est une partie des RSUs. Les auteurs se sont concentrés sur l'accord de niveau de service "Service Level Agree-

ment (SLA)" [95] qui mesure la performance de l'infrastructure et des applications puisque les performances d'une application dépendent d'un ensemble de ressources dans un environnement Cloud. Le Cloud public offre des services logiciels (SaaS), notamment le V-call ; qui permet d'appeler les services d'urgence en cas d'accident, le service de circulation ; qui signale l'état de la route (embouteillages, accidents, routes fermées, etc.), et les services de localisation ; qui identifient les infrastructures importantes dont les utilisateurs peuvent avoir besoin, comme les hôpitaux, les stations de service, les distributeurs automatiques de billets pour retirer de l'argent ou même les caisses sur une autoroute. Le Cloud privé concerne les Clouds formés par les véhicules pour les mesures de sécurité et pour résoudre les problèmes d'interopérabilité. Ce Cloud réside sur l'OBU du véhicule pour maintenir une mise à jour fréquente des informations à proximité. Les auteurs considèrent deux interfaces programmables d'application (API) : l'API de recherche (Inquiry API) ; pour rechercher des informations et les partager entre véhicules, et l'API éditeur (Publisher API) ; pour stocker, mettre à jour, supprimer ou cacher des informations. Ainsi, un ensemble de services est fournis pour le V2V tels que des services d'échange de données et les services d'alerte ; telles que les messages d'accident, d'état de la route, des embouteillages. Et chaque véhicule peut diffuser des messages par le biais d'un mécanisme d'échange de données aux autres véhicules, partageant ainsi des services liés à la sécurité et des services d'info-divertissement via les tableaux de bord pour les passagers. Les auteurs ont testé les performances de leur proposition en utilisant l'environnement Google App Engine Cloud pour les services du Cloud public et le registre privé JUDDI [96] pour le Cloud privé.

- **Zingirian et Valenti [97]** ont proposé un nouveau paradigme de service pour les plates-formes de communication des véhicules (VCP), basé sur le concept de Cloud de capteurs. Le paradigme proposé permet aux VCP de mettre leurs composants disponibles (capteurs et dispositifs de véhicules) à la disposition d'applications tierces de surveillance des véhicules en tant que ressources virtuelles. Les principaux composants de l'architecture de service Cloud de capteurs sur VCP sont l'Event Building Interface (EBI) ; la définition des règles et le mappage d'exécution des règles. L'EBI prend en charge le paradigme de l'abonnement régulier mis en oeuvre par des services Web. Les paramètres de la demande d'abonnement sont les suivants : Rule Scope ; spécifie les véhicules sur lesquels la règle doit être évaluée, Rule Body ; définit la sémantique de la règle, Rule Trigger ; définit le moment où la règle d'évaluation a eu lieu, véhicule de notification ; contenant l'identifiant du véhicule qui a généré l'événement notifié, propriété de notification ; coordonnées géographiques et horodatage dans lequel la notification est déclenchée, corps de notification ; construit par la fonction de notification définie dans le corps de la règle qui est une chaîne formatée. Pour le mappage d'exécution de la règle, chaque règle peut être exécutée à différents niveaux : au niveau du sous-système, la règle s'exécute dans les routines de plus bas niveau du processeur central de l'OBU ou dans les processeurs périphériques. Au niveau du système, la règle s'exécute en tant que routine activée par un drapeau ou en tant qu'appel périodique dans la boucle principale RT-OS. Au niveau du noyau, la règle s'exécute directement dans le thread du gestionnaire de messages, accédant aux données du message traité, aux tables de la base de données ou aux fichiers d'état temporaires de la règle. Au niveau de l'EBI, elle accède à la

base de données principale sous forme de lots périodiques. L'EBI analyse ensuite la règle déterminant sa faisabilité, le niveau approprié auquel elle doit être exécutée selon deux critères : la faisabilité de la planification à chaque niveau d'exécution et la sélection de la planification faisable au niveau d'exécution le plus élevé. Il choisit le niveau approprié en fonction de la contrainte de mémoire, de la contrainte "Central Processing Unit (CPU)", de la contrainte de rafraîchissement, de la contrainte de message et de bande passante. Les auteurs ont conclu que le modèle de Cloud de capteurs est particulièrement adapté aux VCP en raison du fait que les VCP implémentent déjà des fonctions telles que la classification et l'enregistrement des données des capteurs menant au paradigme du Cloud de capteurs, et les nouvelles applications de surveillance intelligente des camions "Intelligent Truck Monitoring (ITM)" insistent fortement sur la vitalisation des ressources afin d'en réduire le coût.

2.5.2 Solutions de routage basées sur le Fog computing

Dans cette section, nous étudions quelques protocoles de diffusion de données dans les VANET basés sur le Fog computing.

- **Paranjothi et al. [98]** ont proposé "Dynamic Fog for Connected Vehicles (DFCV)" un Fog dynamique pour véhicules connectés. Il s'agit d'un schéma pour assurer la communication qui crée, incrémente et détruit dynamiquement les nœuds Fog en fonction des besoins de communication. Il réduit les délais et garantit la livraison des messages même lorsque la densité de véhicules est élevée.

Pour assurer la communication avec les véhicules voisins à l'aide de la technique de calcul du Fog, DFCV se concentre sur deux scénarios : le scénario divisé et le scénario de fusion. Dans le scénario divisé, le Fog est divisé en plusieurs Fogs ; il s'agit d'une opération primitive effectuée par DFCV. Cela se produit lorsque la distance entre les véhicules augmente du point de vue de l'expéditeur ou lorsque la capacité de DFCV est supérieure à la capacité seuil. La distance entre les véhicules est d'abord calculée, puis la capacité de DFCV est calculée selon l'équation (2.3).

$$fc = \frac{n_v}{t_v} \quad (2.3)$$

où fc représente la capacité de DFCV, n_v est le nombre de véhicules connectés au Fog et t_v est le nombre total de véhicules situés dans la région. Ensuite, la capacité est comparée à la capacité seuil du DFCV (th_{cap}) telle que présentée dans l'équation (2.4).

$$\begin{cases} fc \leq th_{cap} & \text{Etablir un lien de communication entre les véhicules} \\ fc > th_{cap} & \text{Diviser le Fog} \end{cases} \quad (2.4)$$

S'il y a une division, les messages sont relayés à la station de base à l'aide de la RSU et sont envoyés aux véhicules. Dans le scénario de fusion, deux ou plusieurs couches de Fog sont combinées sous les mêmes stations de base en une seule couche Fog. Après cela, les messages sont diffusés aux véhicules voisins. Cela se produit lorsque

la distance entre les véhicules est inférieure à la distance minimale ou lorsque la capacité du DFCV est inférieure à la capacité seuil.

La mise en oeuvre de DFCV a été réalisée par simulation à l'aide du simulateur de trafic open source SUMO avec simulateur NS-2. Les résultats montrent que DFCV réduit la latence et offre des performances jusqu'à 35 % supérieures à celles des techniques courantes lorsque la densité des véhicules est élevée et que la livraison des messages est garantie.

- **Ullah et al. [99]** ont proposé un schéma de diffusion basé sur l'évitement de la congestion pour assurer une diffusion efficace des données dans les VANETs. Ils ont proposé une taxonomie de schémas qui traitent l'évitement de la congestion des messages. Ils se sont concentrés sur la diffusion de messages d'urgence parmi les véhicules ou entre les véhicules et les RSUs. Ils ont d'abord exploré l'architecture VANET en expliquant les trois types qu'elle implique : l'architecture "Wireless Local Area Network (WLAN)", l'architecture ad hoc cellulaire et l'architecture hybride. Ils ont également souligné les différents défis auxquels sont confrontés les réseaux véhiculaires tels que la congestion, la latence, le débit, la disponibilité, le routage et la sécurité. Ensuite, ils ont présenté un résumé des différentes méthodes utilisées dans la transmission des messages dans les VANETs. Les auteurs se sont intéressés aux problèmes de congestion des messages lors de la communication entre véhicules ou entre véhicules et autres infrastructures sur la route, ils ont ainsi examiné différents protocoles et algorithmes d'évitement de congestion, les ont comparés et ont présenté un examen approfondi des méthodes et schémas existants sur la façon d'assurer une diffusion efficace des messages de données tout en évitant la congestion. Enfin, ils ont exploré l'architecture assistée par le fog pour les VANETs et ont souligné son rôle important dans l'amélioration des capacités de communication entre les véhicules, en particulier pendant les conditions d'urgence ou les heures de pointe. Les auteurs ont clairement identifié les progrès des VANETs vers les VANETs assistés par le Fog, ainsi que les principales opportunités et défis.
- **Lu et al. [100]** ont proposé un routage géographique heuristique, c'est-à-dire un routage géographique amélioré "Improved Geographic Routing (IGR)". C'est un schéma de routage basé sur la position pour les communications V2V en milieu urbain. L'IGR utilise le calcul du Fog véhiculaire pour tirer le meilleur parti des ressources de communication et de calcul véhiculaires. Il se compose de deux modes. Le premier sélectionne le carrefour en fonction de la destination et de la densité des véhicules sur chaque route, tandis que le second est un mode de transfert glouton amélioré pour transmettre les données entre les carrefours. Dans l'IGR, seul le véhicule source d'un carrefour sélectionne le carrefour de transfert suivant à l'aide du plan des rues pour déterminer les positions des carrefours voisins. Le véhicule source utilise le score pour sélectionner la prochaine jonction de transfert. Plus le score est élevé, meilleure est la sélection. La jonction avec le score le plus élevé est la plus proche de la destination et a la plus forte densité de véhicules. Le score prend en compte la distance euclidienne jusqu'à la destination et la densité des véhicules entre le carrefour actuel et le carrefour suivant et est calculé comme exprimé par l'équation (2.5).

$$score(j_{next}^{(i)}) = \frac{\rho(j_{current}, j_{next}^{(i)})}{\frac{d_i}{d_{max}}} \quad (2.5)$$

où j_{next} et $j_{current}$ sont respectivement la jonction suivante et la jonction actuelle. d_{max} est la distance maximale depuis la jonction candidate j_{next} et d_i est la distance euclidienne depuis la jonction j_{next} . $\rho(j_{current}, j_{next}^{(i)})$ est la densité de véhicules entre la jonction actuelle $j_{current}$ et la jonction candidate j_{next} et est obtenue par l'équation (2.6).

$$\rho(j_{current}, j_{next}^{(i)}) = \sum_{j=1}^{k^i} \frac{N_j}{N_{max}} \quad (2.6)$$

où N_{max} est le nombre maximum de véhicules entre $j_{current}$ et les carrefours voisins, et N_j est le nombre de véhicules dans le segment j_{th} sur la rue, il est calculé selon l'équation (2.7).

$$N_j = ceil\left(\frac{l_i}{R}\right) \quad (2.7)$$

$ceil()$ étant la fonction d'arrondi, R la distance de transmission des véhicules et l_i désigne la distance euclidienne entre $j_{current}$ et la jonction candidate j_{next} .

Une fois la jonction suivante sélectionnée, le paquet de données est transmis à l'aide d'un routage glouton amélioré. Chaque véhicule maintient une table pour ses véhicules voisins et enregistre la position des véhicules voisins, la direction de conduite et le taux de perte de paquets. Lorsqu'un véhicule reçoit un paquet de données, il le transmet à son véhicule voisin en fonction du coût; il choisit le voisin qui a le moindre coût. Le coût de transmission est calculé selon l'équation (2.8).

$$cost(v_i) = \alpha * \frac{d(V_i, j_{next}^{(i)})}{d(j_{current}, d_{next}^{(i)})} + (1 - \alpha) * r_{err}(V_i) \quad (2.8)$$

où $d(V_i, j_{next}^{(i)})$ est la distance euclidienne entre le véhicule V_i et la prochaine jonction j_{next} et $d(j_{current}, d_{next}^{(i)})$ est la distance euclidienne entre la jonction actuelle $j_{current}$ et la jonction suivante j_{next} . $r_{err}(V_i)$ est le taux d'erreurs des paquets du véhicule source au véhicule V_i . α est le poids accordable ($0 \leq \alpha \leq 1$). Pour évaluer les performances du protocole proposé, les auteurs ont implémenté l'IGR via un simulateur de réseau NS-2 en utilisant une topologie urbaine réelle faisant partie de la ville de Shanghai, en Chine. Ils ont comparé l'IGR avec les protocoles GPSR et GPCR et ont prouvé que l'IGR est le plus efficace en terme de délai de bout en bout et de taux de livraison des paquets.

- **Kadhim et Seno [101]** ont présenté un protocole de routage multicast économe en énergie basé sur les SDN et le fog computing pour les réseaux véhiculaires, appelé "Energy Efficient Multicast routing protocol based on SDN and FC for VANETs (EEMSFV)". EEMSFV tient compte des contraintes de la bande passante et des

délais pour minimiser la consommation d'énergie et le délai. De plus, les auteurs ont proposé un algorithme de planification pour classer les demandes multicast entrantes en fonction du type d'applications et un algorithme de classification basé sur la priorité pour planifier les demandes multicast. Lorsqu'un véhicule veut envoyer des données vers un ensemble de noeuds destinataires en mode multicast, il envoie d'abord une requête contenant les adresses IP de la source et des destinations, le type d'application, la taille des données et la contrainte de délai au noeud fixe le plus proche (RSU ou BS) appelé noeud fixe source. Ce noeud fixe transmettra cette demande au contrôleur local en ajoutant son adresse IP. Si tous les véhicules destinataires sont dans la zone de couverture du contrôleur, celui-ci utilisera EEMSFV pour construire l'arbre multicast qu'il enverra ensuite à tous les noeuds fixes appartenant à l'arbre multicast construit. Le contrôleur stockera alors les informations de cette requête et de l'arbre multicast dans une table appelée table de session qui comprend le numéro de session, le type d'application, l'adresse IP source, les adresses IP de destination, la contrainte de limite de temps, la taille des données et la représentation binaire de l'arbre de multidiffusion. Ensuite, il enverra cette table au contrôleur SDN. Si les véhicules de destination ne se trouvent pas tous dans la zone de couverture du contrôleur local, il transmettra la demande de multidiffusion au contrôleur SDN. Ce contrôleur SDN détermine la zone géographique d'où provient la requête multicast. Il vérifie ensuite, si tous les véhicules de destination se trouvent ou non dans la même zone géographique. Si tel est le cas, il s'appuie uniquement sur les informations qui sont stockées dans la table spécifique à cette zone géographique EEMSFV afin de sélectionner rapidement le chemin de multidiffusion optimal, sinon il sélectionne le chemin multicast optimal en exécutant EEMSFV sur les informations stockées dans la table générale de toute la zone géographique pour sélectionner le chemin multicast optimal. Ensuite, il stocke la requête et les informations de l'arborescence de multidiffusion dans sa propre table de session. Enfin, le contrôleur SDN envoie les tables de flux à tous les noeuds intermédiaires appartenant à l'arbre multicast construit. En règle générale, plusieurs demandes de session multidiffusion atteignent le contrôleur local ou le contrôleur SDN, qu'il doit traiter en fonction de leur priorité. Dans un premier temps, le contrôleur classe ces requêtes selon leur type d'application et place chaque groupe d'un même type d'application dans une file d'attente spécifique. Pour ce schéma, les auteurs ont considéré qu'il existe quatre types d'applications : pompiers, police, ambulances, véhicules normaux. Ainsi, chaque contrôleur dispose de quatre files d'attente et planifie les demandes à l'intérieur de chaque file d'attente en fonction de la priorité. La priorité est déterminée en fonction de la contrainte de temps où les demandes urgentes ont une priorité élevée. Ensuite, le contrôleur construit l'arbre de multidiffusion en sélectionnant une requête à la fois en fonction de la priorité des têtes de ces files d'attente. Après cela, le contrôleur envoie des tables de flux aux commutateurs OpenFlow, aux contrôleurs locaux, aux stations de base et aux RSUs qui se trouvent dans cet arbre de multidiffusion. Une fois que le noeud fixe source a reçu la table de flux, il répond au véhicule source pour démarrer la transmission de données. Dans ce cas, l'environnement Fog computing est exploité et pourrait prendre en charge la mobilité et augmenter la connaissance de la localisation dans les réseaux de véhicules ainsi que le principe du clustering qui réduit le nombre de

paquets de contrôle échangés. Cette approche a été implémentée par simulation sur OMNET++ avec SUMO. Les auteurs ont comparé leur approche proposée avec les protocoles de routage multidiffusion CVLMS [102] et MABC [103] et ont prouvé qu'elle est meilleure que CVLMS et MABC en terme de taux de transmissions réussies, de surcharge normalisée, de délai moyen de bout en bout, de pourcentage de sessions de multidiffusion respectant leurs délais, et de taux de perte de paquets.

- **Arif et al.** [104] ont proposé une communication VANET sécurisée basée sur le SDN avec Fog computing. L'environnement basé sur le SDN offre une évolutivité, une flexibilité, une capacité de calcul et des informations globales, tandis que le Fog computing offre des services délicats et sensibles à la localisation qui peuvent répondre aux demandes des VANETs. L'utilisation d'une architecture basée sur le SDN et d'un plan de sécurité permet de fournir un mécanisme sécurisé pour les communications. Pour assurer la sécurité des communications entre les véhicules, le Cloud et les serveurs de transport, les auteurs ont utilisé un contrôleur de réseau défini par logiciel "Software Defined Network Controller (SDNC)" et Fog computing. Les principaux composants de ce système sont : les RSUs du SDN, les contrôleurs routiers SDN "Road Side Unit Controller (RSUC)". Les RSUs sont fixées le long des routes et contrôlées par le SDNC, elles sont utilisées pour transmettre des données et fonctionnent comme des dispositifs Fog. Les RSUs se connectent d'abord aux RSUC pour approcher le SDNC. En plus d'être responsables de la transmission des données, les RSUC stockent également les informations sur le trafic et fournissent les services d'urgence et nécessaires. Tous les RSUC fonctionnent comme des dispositifs Fog sous l'orchestration du contrôleur SDN. Le Fog Computing agit comme un intermédiaire entre SDNC et le serveur Cloud pour assurer les communications sécurisées dans les VANETs. Les communications sont composées de trois parties : la sécurité, les plans de données et le contrôle. Le contrôle SDN hybride est utilisé avec peu de modifications du plan de sécurité pour fournir une meilleure sécurité. La communication du plan de données est destinée au transfert de données, tandis que la communication de contrôle des données concerne les règles et les politiques. Le plan de contrôle de sécurité envoie les clés de sécurité aux RSUs, et les RSUC collectent toutes les clés, les combinent et les envoient au SDNC. Ensuite, les données du RSUC sont envoyées au serveur Fog via SDNC. Le framework Veins avec OMNET++ et SUMO est utilisé pour la mise en oeuvre de ce protocole. Les résultats ont montré que le système proposé fournit de bons résultats pour les deux types de communications (V2V et V2I) dans les réseaux VANETs et a un coût de communication très faible contrairement à d'autres cadres de sécurité qui, généralement, imposent un coût élevé et une surcharge importante sur le réseau.
- **Truong et al.** [105] ont proposé une architecture appelée FSDN qui combine l'architecture SDN et le Fog Computing. FSDN se compose de quatre couches : véhicules, Fog computing, contrôleur SDN et Cloud computing. Le SDN offre la flexibilité, l'évolutivité, les connaissances globales et la programmabilité, tandis que le Fog computing fournit des services sensibles aux retards et des services de détection de localisation. Le principe de fonctionnement est le même que dans [104] avec le SDNController et le RSUController. La différence est qu'ici, il s'agit uniquement de communication, d'envoi de données entre les composants du réseau sans la

fonction de sécurité ajoutée par le protocole de Arif et al. [104]. Dans ce protocole, les RSUCs et BSs sont des dispositifs Fog qui s'intègrent au Cloud via le SDNC et partagent leurs ressources pour contrôler les véhicules. Le contrôleur SDN orchestre également les noeuds Fog et gère les ressources.

- **Hou et al. [3]** ont proposé une architecture appelée VFC dans laquelle ils combinent le Fog computing avec les VANETs. Dans l'architecture VFC, les véhicules sont utilisés comme infrastructures de communication et de calcul où les ressources de communication et de calcul individuelles de chaque véhicule sont impliquées pour effectuer la communication et le calcul. Les auteurs ont envisagé quatre scénarios ; les deux premiers scénarios utilisent des véhicules en mouvement comme infrastructures de communication et de calcul et les deux autres scénarios considèrent les véhicules en stationnement comme des infrastructures pour augmenter les capacités de calcul et de communication. Pour utiliser les véhicules en mouvement comme infrastructure, les auteurs ont souligné l'importance de connaître la vitesse des véhicules dans les domaines spatial et temporel. Ainsi, il est possible d'explorer la relation entre connectivité et mobilité des réseaux de véhicules afin d'appréhender globalement les circonstances de la communication en milieu urbain et ainsi d'évaluer précisément les possibilités de cette infrastructure. Concernant les véhicules stationnés en bord de route, ils sont intéressants à être utilisés comme infrastructures car ils peuvent largement relayer les paquets et améliorer la connectivité. En raison des caractéristiques d'une large distribution géographique et d'emplacements spécifiques, un grand nombre de véhicules en stationnement de longue durée peut devenir crédible, abondant et pratique à utiliser comme noeuds de communication et de traitement dans les zones urbaines. Les auteurs ont étudié la mobilité et la connectivité des véhicules dans des scénarios urbains, ils ont mené leurs essais sur deux traces de véhicules urbains à grande échelle dans le monde réel, la trace de la ville de Shanghai et la trace de la ville de Pékin en utilisant le protocole GPRS pour la communication. Ils ont démontré l'énorme potentiel d'amélioration réalisé par VFC dans la capacité de communication et de calcul.

2.5.3 Récapitulatif des solutions de routage dans le Cloud/Fog véhiculaire

Par l'organigramme de la figure 2.2 et le tableau 2.3 , nous représentons les techniques de diffusion citées ci-dessus et résumons leurs caractéristiques.

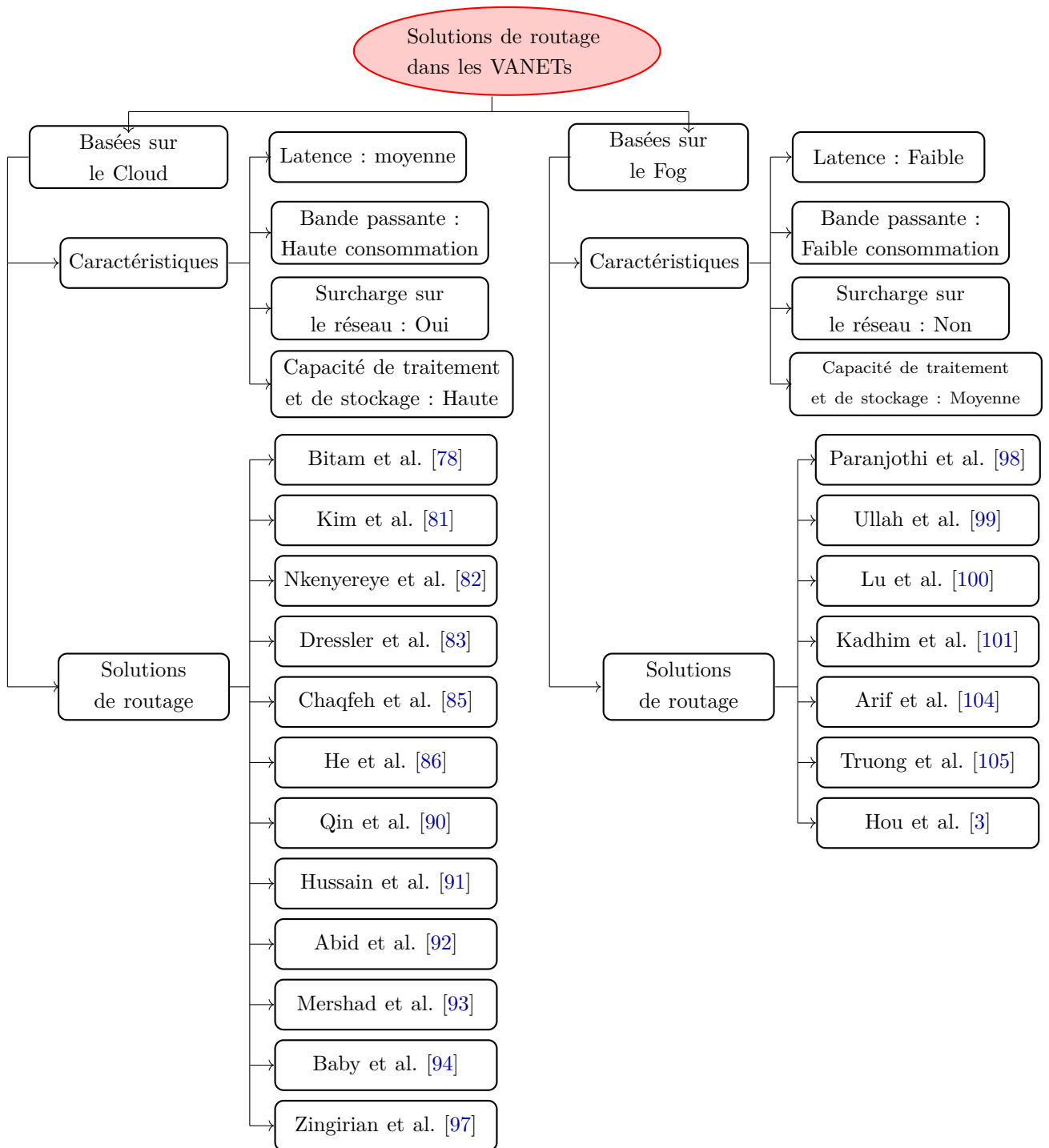


FIGURE 2.2 – Solutions de routage dans VCC/VFC

TABLE 2.3 – Techniques de dissémination de données dans les réseaux VCC et VFC

Protocole	Architecture	Fiabilité	Vitesse de transfert	Sécurité	Bande passante	Complexité
Bitam et al. [78]	Cloud	++	+	-	++	$O(n^2)$
Kim et al. [81]	Cloud	+	+	+	++	$O(n \log n)$
Paranjothi et al. [98]	Fog	++	++	-	+	$O(\log n)$
Nkenyereye et al. [82]	Cloud	+	-	++	++	$O(n^n)$
Ullah et al. [99]	Fog	+	+	-	-	$O(n)$
Dressler et al. [83]	Cloud	+	-	-	+	$O(n^n)$
Chaqfeh et al. [85]	Cloud	++	-	-	++	$O(2^n)$
Lu et al. [100]	Fog	++	++	-	+	$O(\log n)$
Kadhim et Seno [101]	Fog	++	++	-	-	$O(n^2)$
Arif et al. [104]	Fog	++	+	++	+	$O(n \log n)$
Truong et al. [105]	Fog	+	++	-	+	$O(\log n)$
He et al. [86]	Cloud/Fog	++	+	-	++	$O(n \log n)$
Hou et al. [3]	Fog	++	+	-	-	$O(\log n)$
Qin et al. [90]	Cloud	++	+	-	+	$O(n \log n)$
Hussain et al. [91]	Cloud	+	+	-	+	$O(n \log n)$
Abid et al. [92]	Cloud	+	+	++	+	$O(2^n)$
Mershad et Artail [93]	Cloud	++	+	-	+	$O(n \log n)$
Baby et al. [94]	Cloud	++	+	-	+	$O(n \log n)$
Zingirian et al. [97]	Cloud	+	+	-	+	$O(n)$

2.6 Conclusion

Dans ce chapitre, nous avons discuté des différents schémas de dissémination de données dans les VANETs, les réseaux Cloud véhiculaires et les réseaux Fog véhiculaires. Nous nous sommes également intéressés aux architectures proposées puisque l'architecture d'un réseau joue un rôle crucial dans la dissémination des données. Ce point est encore plus important dans le cas d'un réseau véhiculaire où les nœuds sont extrêmement mobiles et l'architecture est en constante évolution. En outre, bien que de nombreuses études se concentrent sur la diffusion des données dans VANETs, cette dernière reste difficile car les nœuds changent constamment d'emplacement et il n'existe pas encore de méthodes standards de dissémination de données dans des environnements mobiles comme les VANETs. De plus, avec le Cloud véhiculaire et le Fog véhiculaire, même les ressources dont nous disposons ne se trouvent pas à des endroits spécifiques du réseau et sont également mobiles en fonction de la localisation du nœud Cloud ou du nœud Fog qui se déplacent constamment comme dans le cas du Cloud mobile où les véhicules en mouvement louent leurs ressources à d'autres véhicules ou utilisateurs.

**Déploiement efficace des RSUs dans
les VANETs basé sur les méthodes
d'optimisation**

Déploiement efficace des RSUs dans les VANETs basé sur les méthodes d'optimisation

3.1 Introduction

Aujourd'hui, les véhicules sont devenus de plus en plus intelligents et équipés de systèmes hautement sophistiqués. Cela leur permet de communiquer entre eux et avec les infrastructures de la route, principalement les unités de bord de route (RSUs). De plus, pour assurer une diffusion efficace des données dans les réseaux VANETs, il est recommandé d'opter pour une architecture véhicule-à-infrastructure (V2I) dans laquelle les RSUs sont installés aux carrefours du réseau routier. Néanmoins, il n'est pas conseillé de placer une RSU à chaque intersection en raison du coût élevé que cela implique. Il convient donc de réduire le nombre de RSUs en choisissant des emplacements aux intersections stratégiques qui maximisent la surface de couverture de la zone urbaine tout en minimisant la surface des zones de chevauchement. En outre, le déploiement d'un nombre optimal de RSUs dans une zone urbaine répondant aux exigences susmentionnées est un défi NP-Difficile puisque le nombre de combinaisons est très élevé lorsque le nombre d'intersections est très grand.

Afin de trouver une solution optimale à cela, nous avons opté pour deux approches métaheuristiques. La première s'appuie sur la version standard des algorithmes génétiques (GA-Basic), puis sur la version améliorée des GAs (GA-Improved), tandis que la seconde approche repose sur la métaheuristique recuit simulé (SA) dans sa version standard (SA-Basic) puis sur sa version améliorée (SA-Improved).

La simulation et l'évaluation des résultats ont été menées à l'aide du simulateur d'événements discrets OMNeT++ et les résultats obtenus ont montré que l'approche GA-Improved déploie un nombre réduit de RSUs (seulement 37,5% de RSUs au niveau de la zone de déploiement) tout en garantissant des performances de routage acceptables par rapport aux approches GA-Basic, SA-Basic et SA-Improved qui déploient respectivement 46,25 %, 65 % et 52,50 % pour les mêmes performances de routage.

3.2 Contexte

Les réseaux ad hoc véhiculaires (VANETs) [14, 28, 106, 107] sont un type particulier de réseaux sans fil dans lesquels les noeuds sont des véhicules en mouvement sur les routes. Ils permettent la communication et l'échange d'informations entre les utilisateurs sur les routes. Ce type de réseaux est une version spéciale des réseaux mobiles ad hoc (MANETs) [108] avec des caractéristiques supplémentaires.

Les nouvelles technologies de communication sans fil rendent les véhicules plus intelligents [109, 110] et favorisent le développement de systèmes de transport intelligents (ITS). L'une des principales fonctions des ITS est de prédire l'état futur du trafic [111] et de gérer plusieurs applications V2X qui augmentent la sécurité routière [111, 112].

Dans les VANETs [113, 114], les véhicules sont très mobiles et sont capables de stocker et de traiter des informations; notamment des messages d'urgence, des alertes de trafic et des informations de localisation. Deux modes de communication sont prévus dans les réseaux VANETs [115] : de véhicule à véhicule (V2V), où les véhicules communiquent entre eux sans passer par les RSUs, et le mode véhicule à infrastructure (V2I), où les véhicules équipés d'unités embarquées (OBU) communiquent directement avec les RSUs. L'organisation IEEE a défini une pile de communication appelée accès sans fil pour l'environnement du véhicule (WAVE) sous l'égide de l'IEEE802.11p [116], qui gère les communications entre véhicules et RSUs.

Les VANETs reposent sur les communications V2I, dans lesquelles les véhicules diffusent périodiquement des données pour mettre à jour les RSUs situées à proximité dans leur voisinage. Les RSUs sont généralement placées aux intersections des routes, en particulier dans les environnements urbains, et sont d'une importance capitale dans la diffusion des informations sur le trafic. Cependant, il n'est pas conseillé de placer une RSU à chaque intersection; d'une part, à cause de leur coût qui est assez élevé et, d'autre part, car, lorsque ces RSUs sont proches les unes des autres, ils génèrent des zones de chevauchement dans lesquelles les véhicules ne peuvent pas communiquer avec les autres véhicules et avec les RSUs. Par conséquent, il convient de placer un nombre optimal de RSUs à des intersections spécifiques afin d'obtenir des performances de routage acceptables.

Dans les réseaux sans fil tels que les réseaux ad hoc véhiculaires et les réseaux de capteurs sans fil (WSNs), les paquets peuvent être perdus en raison de leurs collisions au niveau du support, en particulier lorsque, dans une zone géographique, de nombreuses RSUs qui sont proches les unes des autres tentent de transmettre des données simultanément, ou en raison d'un débordement de noeuds ou de RSUs, c'est-à-dire si la file d'attente utilisée pour contenir les paquets de données devant être transmis déborde en entraînant une perte de paquets [117, 118], ou en raison de la grande mobilité des noeuds. Ce phénomène a un impact négatif sur les performances du réseau, notamment sur le taux de livraison des paquets avec succès [119]. De plus, les caractéristiques des VANETs sont limitées par de nombreux facteurs tels que la couverture de la zone de transmission, la bande passante et la capacité de liaison [120]. Cela peut également entraîner un taux élevé de perte de paquets et une dégradation des performances de routage.

La dissémination fiable des données est considérée comme l'un des défis les plus importants des réseaux sans fil tels que les WSNs et les VANETs. De nombreuses applications dans ces types de réseaux nécessitent une diffusion fiable de données, comme les applications de sécurité routière [121]. Dans cette optique, le placement des RSUs aux

intersections doit être effectué de manière efficace afin d'obtenir des échanges fiables entre les véhicules eux-mêmes et entre les véhicules et les RSUs.

Un déploiement efficace des RSUs dans une zone urbaine doit tenir compte du compromis entre le phénomène de planification automatique des cellules "Automatic Cell Planning (ACP)" et le phénomène de planification automatique des fréquences "Automatic Frequency Planning (AFP)". Selon le phénomène ACP, chaque fois qu'une RSU est ajoutée à la zone urbaine, la surface couverte par toutes les RSUs installées augmente, alors que selon le phénomène AFP, la surface de la zone de chevauchement (la surface de la zone couverte par deux RSUs ou plus) augmente également, ce qui génère du bruit dans la zone de chevauchement. Par conséquent, l'ajout d'une RSU à une intersection dans la zone urbaine devrait augmenter la superficie de la zone couverte et minimiser la superficie de la zone de chevauchement.

Nous avons remarqué que dans la plupart des travaux présentés dans la littérature concernant les VANETs dans lesquels il existe des architectures basées sur les RSUs, les auteurs n'accordent pas beaucoup d'importance au nombre de RSUs qu'ils doivent déployer dans la zone urbaine malgré leur coût très élevé ainsi qu'à leurs positions dans cette zone. A titre d'exemple, citons l'architecture proposée par Liu et al. [112] pour les tâches gourmandes en ressources informatiques et/ou sensibles aux délais. Les auteurs ont proposé une architecture basée sur des serveurs de calcul en bordure du réseau véhiculaire où les requêtes transitent par des RSUs sans tenir compte du coût de cette architecture qui est en fonction du nombre de RSUs déployées en zone urbaine.

A notre connaissance, le travail présenté par Ghorai et Banerjee [122] est le seul travail dans lequel les auteurs ont proposé de placer des RSUs dans une zone urbaine pour en assurer la couverture. Cependant, dans ce travail, les auteurs n'ont pas tenté de réduire le nombre de RSUs et de sélectionner les meilleurs emplacements pour les RSUs dans la zone urbaine.

Cette contribution a pour objectif de minimiser le nombre de RSUs déployées et de trouver les meilleurs emplacements pour celles-ci dans un environnement urbain. Pour ce faire, nous avons pris comme exemple d'environnement la localité d'Abou Tachfine de la wilaya de Tlemcen en Algérie dans laquelle il y a 80 intersections. Nous avons considéré les résultats obtenus par le protocole de routage décrit dans [123] en termes de taux de livraison de paquets "PDR" et de délai de bout en bout comme critères de performance pour les approches proposées. De plus, le nombre de scénarios possibles est très important et l'exploration de tous les scénarios ne peut être réalisée dans un temps raisonnable. Par conséquent, nous avons utilisé des approches métaheuristiques pour trouver le meilleur scénario qui permet de déployer un nombre minimum de RSUs tout en garantissant des performances de routage acceptables. Nous avons utilisé deux métaheuristiques différentes pour réduire le temps de calcul pour trouver le meilleur scénario : les algorithmes génétiques [12] (approches basées sur GAs) et recuit simulé [13] (approches basées sur SA). Les solutions proposées visent à obtenir un taux de couverture élevé sur les routes de la localité concernée avec des performances de routage acceptables tout en déployant un nombre minimal de RSUs aux intersections. En outre, pour accélérer la convergence de ces métaheuristiques et obtenir de meilleurs résultats, nous proposons une version améliorée pour chacune des métaheuristiques (basée sur GA et SA). Dans ce contexte, nous avons exploité chacune de ces métaheuristiques dans un contexte standard (GA-Basic et SA-Basic) ainsi que dans un contexte amélioré (GA-Improved et SA-Improved).

Les principales contributions des approches proposées sont les suivantes :

- Déploiement efficace d'un nombre réduit de RSUs dans une zone urbaine basé sur des approches métaheuristiques tout en ayant des performances de routage acceptables.
- Réduction du nombre de RSUs déployées dans une zone urbaine et sélection de leurs emplacements appropriés aux intersections.
- Réduction du coût de déploiement des RSUs et minimisation des zones de chevauchement (zones couvertes par plus d'une RSU).

Dans ce qui suit, nous décrivons d'une façon générale l'optimisation combinatoire avec les méthodes de résolution de ses problèmes, ainsi que les métaheuristiques recuit simulé et algorithme génétique, puis nous détaillons notre contribution et enfin, nous présentons les résultats obtenus.

3.3 Méthodes d'optimisation combinatoire

Il existe de nombreux problèmes intéressants pour des applications pratiques où l'utilisation d'une méthode déterministe pour trouver une solution est presque impossible en termes de temps de calcul, en particulier lorsque l'espace des solutions réalisables est très grand. Dans ce genre de cas, il s'agit de problèmes d'optimisation combinatoire. Les problèmes d'optimisation combinatoire apparaissent dans plusieurs applications [124]. Parmi ces applications, nous pouvons citer les suivantes :

- Trouver le chemin le plus court de Paris à Rome dans le réseau routier européen,
- Planifier les examens pour les cours donnés dans une université,
- Trouver des chemins dans des réseaux à grande échelle,
- Placer des bornes WIFI pour couvrir une grande zone,
- Placer des RSUs pour couvrir une zone urbaine,
- etc.

Pour ce type de problèmes, il n'y a pas d'algorithmes connus avec un temps de calcul polynomial et les seuls algorithmes permettant d'obtenir une solution optimale sont d'une complexité asymptotique très élevée tels que les problèmes à complexité exponentielle $\theta(2^n)$ ou à complexité factorielle $\theta(n!)$.

3.3.1 Optimisation combinatoire

Les problèmes d'optimisation combinatoire se divisent en deux grandes catégories : les problèmes à variables continues et les problèmes à variables discrètes. Pour la première catégorie, les problèmes à variables continues sont bien connus de l'école dans les cours de mathématiques. Un exemple simple consiste à trouver le minimum de la fonction $F : R \rightarrow R$ avec $f(x) = x^2$. Il est évident que $x_0 = 0$ est la solution unique pour ce problème.

Les problèmes plus complexes sont souvent abordés en calculant les dérivées, en utilisant les méthodes de Newton ou les techniques de programmation linéaire [124]. La deuxième catégorie traite l'optimisation discrète. En outre, un problème d'optimisation combinatoire vise à minimiser ou à maximiser une fonction objectif donnée avec un ensemble donné de contraintes. Formellement, il peut être défini comme un triplet (S, f, Ω) où S est un espace de recherche donné, f est la fonction objectif, qui doit être soit maximisée soit minimisée, et Ω est l'ensemble des contraintes qui doivent être satisfaites pour obtenir des solutions réalisables.

Pour traiter un problème d'optimisation combinatoire afin de trouver une meilleure solution, nous procédons comme suit :

- Modélisation du problème : dans cette étape, nous modélisons le problème en le représentant dans un contexte analytique ou de simulation et nous définissons l'espace des solutions,
- Formulation mathématique : nous formalisons le problème et nous définissons une fonction objectif,
- Résolution du modèle : cette étape consiste à utiliser une méthode d'optimisation pour résoudre le modèle,
- Évaluation de la solution trouvée et nécessité de revenir à la première étape pour une éventuelle amélioration de la solution courante.

Trouver une solution optimale à l'aide d'algorithmes de complexité exponentielle ou factorielle est inconcevable, à moins que les ressources informatiques disponibles soient importantes. Pour ce genre de problèmes, puisqu'il est difficile d'obtenir une solution en un temps raisonnable, l'utilisation d'algorithmes qui fournissent des solutions "pas trop mauvaises" fait partie des approches prometteuses à utiliser pour le traitement.

3.3.2 Méthodes de résolution des problèmes d'optimisation combinatoire

Dans les problèmes d'optimisation combinatoire de la vie réelle, les solutions doivent généralement être évaluées de différents points de vue correspondant à des objectifs multiples et souvent contradictoires [125]. Il existe deux types d'approches pour résoudre un problème d'optimisation combinatoire : les méthodes exactes telles que Branch & bound [126] et la programmation dynamique [127], ainsi que les méthodes approchées telles que la colonie de fourmis [128], les algorithmes génétiques [12], le recuit simulé [13], la recherche tabou [129], etc... La figure 3.1 illustre cette catégorisation en mettant l'accent sur les principales méthodes d'optimisation décrites dans la littérature.

a) Méthodes exactes

Les méthodes exactes explorent l'ensemble de l'espace des solutions réalisables pour trouver la solution optimale. Ces méthodes sont utilisées pour trouver au moins une solution optimale à un problème. Néanmoins, les méthodes exactes ne sont efficaces que pour les petits problèmes et perdent leurs performances lorsqu'elles passent à l'échelle

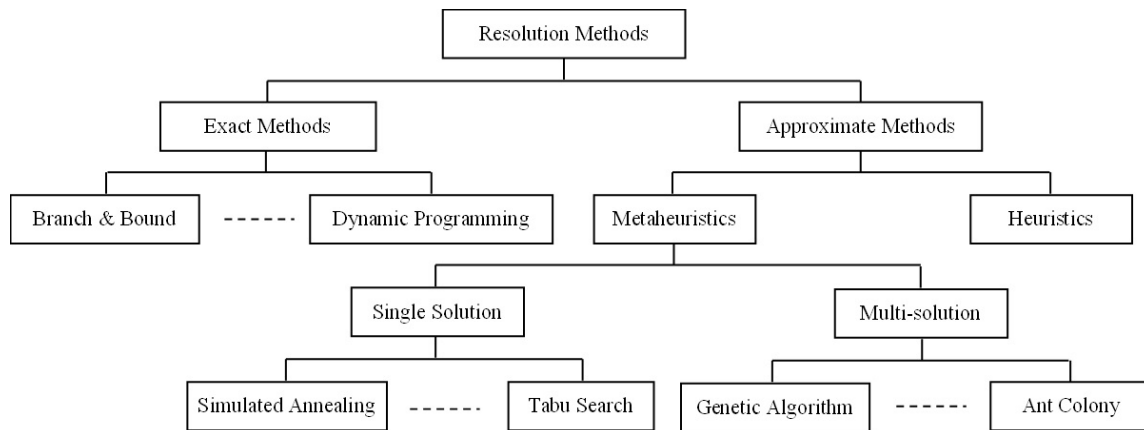


FIGURE 3.1 – Classification des principales méthodes d'optimisation combinatoire

supérieure. Les plus efficaces dans la littérature appartiennent aux paradigmes de deux classes principales : la programmation dynamique et les méthodes Branch & Bound [130].

- **Programmation dynamique** : La résolution d'un problème d'optimisation combinatoire par programmation dynamique consiste à diviser ce problème en plusieurs sous-problèmes, traiter chacun d'entre eux d'une façon indépendante. La solution optimale au problème initiale consiste en la somme des solutions optimales de tous ses sous-problèmes.
- **Branch & Bound** : C'est une méthode par séparation et évaluation. Elle consiste à énumérer toutes les solutions possibles, à diviser l'ensemble des solutions en sous-ensembles et évaluer chaque solution des sous-ensemble en majorant la valeur de la meilleure solution de chaque sous-ensemble.

b) Méthodes approchées

Les méthodes approchées explorent non pas tout l'espace de solutions mais un sous-ensemble de cet espace des solutions réalisables. Parmi ces méthodes, nous citons :

- **Les algorithmes d'approximation** : Ces algorithmes fournissent une solution "pas trop éloignée" de la solution optimale d'où leur nom algorithme d'approximation ou algorithme de rapprochement.
- **Les algorithmes heuristiques** : Ces algorithmes fournissent une solution qui se rapproche de la solution optimale sans que ce soit nécessaire de définir exactement la qualité de la solution produite par rapport à la solution optimale. Une heuristique peut produire une solution qui ne soit pas trop bonne comme elle peut ne pas produire de solution du tout. Les heuristiques peuvent être classées en deux catégories :
 - **Les méthodes constructives** : Ces méthodes de résolution génèrent des solutions de manière incrémentielle, c'est-à-dire à partir d'une solution initiale vide à laquelle on ajoute des éléments jusqu'à l'obtention d'une solution qui peut être plus ou moins complète.

- **Méthodes d'exploration locale** : Ces méthodes partent d'une solution initiale complète et font plusieurs tentatives pour améliorer cette solution en explorant son voisinage direct.
- **Les algorithmes probabilistes** : Ces algorithmes sont non déterministes. En effet, lorsqu'il y a lieu de faire un choix, ce choix se fait de manière aléatoire.
- **Les algorithmes métaheuristiques** : Ces algorithmes utilisent et coordonnent d'autres heuristiques afin d'obtenir la solution souhaitée à un problème compliqué. Une heuristique est conçue pour résoudre un problème spécifique alors qu'une métaheuristique est conçue pour être appliquée à divers problèmes. Dans ce qui suit, nous présentons en détail les principes des deux métaheuristiques algorithmes génétiques et recuit simulé, que nous avons utilisées pour traiter le problème du déploiement des RSUs dans une zone urbaine avec un grand nombre d'intersections.

3.3.3 Métaheuristiques : recuit simulé et algorithmes génétiques

Les métaheuristiques sont des heuristiques inspirées de la nature et dédiées à la résolution de problèmes NP-difficiles et plus particulièrement aux problèmes d'optimisation qui visent à atteindre un optimum global généralement enfoui au milieu de nombreux optimums locaux [131]. Comme le montre la figure 3.1, les métaheuristiques sont divisées en deux sous-classes : les méthodes à solution unique et les méthodes à population de solutions.

Dans ce qui suit, nous détaillons une métaheuristique de chaque classe : le recuit simulé pour les méthodes à solution unique et les algorithmes génétiques pour les méthodes à population de solutions.

a) Recuit simulé (Simulated Annealing)

La métaheuristique recuit simulé "Simulated Annealing (SA)" est une généralisation de la méthode de Monte Carlo; son objectif est de trouver une solution optimale pour un problème donné. Elle s'inspire de l'algorithme de Metropolis, qui permet de décrire l'évolution d'un système thermodynamique [132]. Elle se distingue par deux caractéristiques principales : la facilité de sa mise en oeuvre, et sa rapidité de convergence par rapport aux autres métaheuristiques décrites dans la littérature. De plus, elle fournit de meilleurs résultats pour de nombreux problèmes d'optimisation combinatoire par rapport aux autres métaheuristiques à solution unique.

Nous pouvons considérer la métaheuristique recuit simulé comme une version étendue de la méthode de descente. Elle commence par l'initialisation d'une population X et un paramètre de contrôle θ , appelé température. Ensuite, elle génère aléatoirement une solution de voisinage $X' \in N(X)$ et l'évalue. Après cela, elle compare ses performances avec la solution courante. La population X' sera sélectionnée s'il y a une amélioration ou s'il y a une dégradation minimale de la solution actuelle avec une certaine probabilité. Le processus s'arrête lorsque le paramètre de contrôle devient nul et la meilleure solution trouvée parmi les solutions explorées est alors renvoyée comme résultat.

b) Algorithmes génétiques

Les algorithmes génétiques "Genetic Algorithms (GAs)" sont des algorithmes d'optimisation stochastiques basés sur les phénomènes de génétique et de sélection naturelle. Leur fonctionnement est extrêmement simple ; on part d'une population initiale de solutions potentielles (individus ou chromosomes) choisies arbitrairement. Les performances relatives (fitness) de ces solutions sont évaluées et, sur la base de ces performances, une nouvelle population de solutions est constituée à l'aide d'opérateurs évolutifs simples : la sélection, le croisement et la mutation. Ce processus de recherche est répété jusqu'à ce qu'une solution satisfaisante soit trouvée ou que le nombre de cycles fixé au début soit épuisé.

1. Présentation de l'algorithme : Les GAs sont définis par les propriétés suivantes :
 - L'individu ou le chromosome : représente une solution potentielle au problème posé.
 - La population : une population est un ensemble d'individus (chromosomes) ou de points de l'espace de recherche.
 - L'environnement : représente l'espace de recherche.
 - La fonction fitness : c'est la fonction objectif que nous cherchons à atteindre (fonction à maximiser ou à minimiser).
2. Phases d'exécution de l'algorithme génétique : Les algorithmes génétiques s'exécutent sur la base des phases suivantes :
 - La phase d'initialisation : une population initiale de N chromosomes (N solutions) est choisie aléatoirement.
 - La phase d'évaluation : chaque solution (chromosome) est décodé, puis évalué.
 - La phase de sélection : une nouvelle population de N chromosomes est générée en utilisant une méthode de sélection appropriée.
 - La phase de reproduction : croisement et mutation possibles au sein de la nouvelle population.
 - Revenir à la phase d'évaluation jusqu'à ce que l'algorithme soit arrêté.
3. Éléments fondamentaux des algorithmes génétiques :
 - *Le codage de la population* : La première étape de la mise en oeuvre des algorithmes génétiques consiste à générer une population d'individus initiaux (solutions). En effet, les algorithmes génétiques agissent sur une population d'individus, et non sur un individu isolé. D'une manière analogue à la biologie, chacun des individus de la population est codé par un chromosome appelé aussi génotype. Une population est donc constituée de chromosomes et chacun de ces chromosomes code un point dans l'espace de recherche. L'efficacité des algorithmes génétiques dépend du choix de la méthode de codage des chromosomes et de la qualité de la population initiale. Il existe trois grands types de codage : binaire, Gray ou réel.

- *La fonction fitness* : Il s'agit d'une fonction d'évaluation permettant de calculer le coût d'un point dans l'espace de recherche. L'évaluation d'un individu ne dépend pas de celle des autres individus. Le résultat fourni par la fonction d'évaluation permet de rejeter ou de sélectionner un individu afin de ne garder que les individus avec les meilleurs coûts parmi les individus de la population courante. La fonction de fitness permet de s'assurer que les individus les plus performants seront préservés, tandis que les individus les moins performants seront progressivement éliminés de la population.
- *Le croisement (Hybridation)* : La fonction de croisement permet de créer, à partir de deux individus appelés "parents", de nouveaux individus appelés "enfants" qui héritent de certaines caractéristiques de leurs parents. La probabilité de croisement désigne la fréquence à laquelle les hybridations sont appliquées. Un enfant est une copie exacte de ses parents s'il n'y a pas d'hybridation. Si l'hybridation a lieu, les enfants sont constitués d'une partie de chacun de leurs parents. Si la probabilité d'hybridation est de 0%, la nouvelle génération est une copie de la précédente, alors que si cette probabilité est de 100%, alors l'intégralité des descendants seront générés par hybridation. L'hybridation est mise en oeuvre de telle sorte que les nouveaux chromosomes conservent la meilleure partie des anciens chromosomes afin d'obtenir finalement les meilleurs chromosomes.
- *La mutation* : Le rôle de cette opération est de modifier avec une certaine probabilité, de manière aléatoire, la valeur des gènes d'un individu. Dans le cas d'un codage binaire, chaque bit $\alpha_i \in \{0, 1\}$ concerné par la mutation, sera substitué par son inverse $\alpha'_i = 1 - \alpha_i$ selon une probabilité pm . De la même manière, il est possible, qu'une même chaîne de bits subissent plusieurs mutations. Afin d'éviter que l'AG ne converge vers des extrêmes locaux et pour permettre la création de nouveaux éléments, si la mutation génère un individu d'une valeur plus faible, celui-ci sera éliminé. La fréquence avec laquelle les gènes d'un chromosome sont mutés est représenté par la probabilité de mutation. S'il n'y a pas de mutation, le fils est inséré dans la nouvelle population sans être modifié, tandis que si la mutation a lieu, une partie du chromosome est modifiée.
- *L'itération* : Une population intermédiaire de m individus est sélectionnée à partir d'une population initiale de N individus. Un individu peut être sélectionné à plusieurs reprises comme il ne peut ne jamais être sélectionné, en fonction de sa fitness. Les m individus de la population se croisent par paires (les paires sont formées aléatoirement ou selon une certaine politique) pour générer N nouveaux individus. Ces individus subissent des mutations qui agissent de manière aléatoire avec une faible probabilité (2-3%) pour constituer une nouvelle population. Le processus est répété à partir de cette population jusqu'à ce qu'une solution satisfaisante soit obtenue. La sélection des individus impliqués dans l'opération de croisement peut prendre plusieurs formes :
 - Sélection par roue aléatoire biaisée : Les parents sont sélectionnés sur la base de leurs performances. Par conséquent, les parents ayant de meilleures performances ont plus de chances d'être sélectionnés. Il s'agit d'une roue

aléatoire biaisée sur laquelle tous les chromosomes de la population sont placés en fonction de leurs valeurs de fitness. Les meilleurs chromosomes peuvent être sélectionnés plusieurs fois tandis que les moins bons ont une chance très mince d'être sélectionnés.

- Sélection par rang : Cette sélection trie d'abord la population par fitness et assigne à chaque chromosome un rang en fonction de sa position. Le pire chromosome aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur chromosome qui aura le rang N (N étant le nombre d'individus dans la population). La sélection par rang d'un chromosome est la même que par la roue, mais les proportions sont relatives au rang plutôt qu'à la fitness.
- Sélection steady-state : L'idée principale de cette politique de sélection est qu'une majeure partie de la population puisse survivre jusqu'à la génération suivante. Les algorithmes génétiques fonctionnent de telle sorte que dans chaque génération, quelques chromosomes sont sélectionnés parmi ceux qui ont les meilleurs coûts pour générer des chromosomes descendants. Ensuite, les pires chromosomes sont éliminés et remplacés par les nouveaux. Le reste de la population survit alors à la nouvelle génération.
- Sélection par tournoi : Cette méthode utilise seulement des comparaisons entre individus sans nécessité de trier la population. Elle possède un seul paramètre, appelé la taille du tournoi T . Pour sélectionner un individu, T individus sont extraits uniformément de la population, et le meilleur des T individus sera sélectionné. Si $T = 2$, alors deux individus de la population sont tirés au hasard, et le meilleur sera sélectionné avec une probabilité p comprise entre 0,5 et 1. Ce processus est répété m fois afin d'obtenir les m individus de la population qui serviront de parents. Le choix de la taille T permet de faire varier la pression sélective, c'est-à-dire d'augmenter les chances de sélectionner les plus performants au détriment des plus faibles.
- Élitisme : Lors de la création d'une nouvelle population, il y a de fortes chances de perdre les meilleurs chromosomes après l'hybridation et la mutation. Pour éviter cela, la méthode de l'élitisme est utilisée. Elle consiste à copier un ou plusieurs des meilleurs chromosomes dans la nouvelle génération. Ensuite, l'algorithme de reproduction habituel génère le reste de la population. Cette politique de sélection améliore grandement les algorithmes génétiques puisqu'elle permet de garder les meilleures solutions.

3.4 Approches proposées pour un déploiement efficace des RSUs dans les VANETs

Pour assurer une dissémination efficace des données dans les VANETs, en particulier dans les zones urbaines, il est recommandé de déployer un nombre suffisant de RSUs aux intersections. Cependant, étant donné le coût relativement élevé des RSUs et les frais que cela engendre de les déployer, il est recommandé de déployer un nombre minimum de RSUs, mais suffisant, pour couvrir toute la zone urbaine et minimiser les zones de chevauchement tout en garantissant des performances de routage acceptables. Dans ce

but, nous proposons quatre approches basées sur des métaheuristiques pour optimiser le nombre de RSUs à déployer et choisir des emplacements stratégiques à chacune des RSUs afin de couvrir toute la zone urbaine en minimisant les chevauchements.

Étant donné que le nombre de scénarios possibles pour un nombre donné d'intersections et de RSUs est très élevé, les méthodes d'optimisation combinatoire exactes ne peuvent pas résoudre efficacement ce problème. Par conséquent, nous avons opté pour l'utilisation des métaheuristiques pour traiter ce problème. Nous utilisons deux types de métaheuristiques : une de la classe des "solutions uniques" représentée par la métaheuristique recuit simulé (approche SA) et l'autre de la classe des "solutions multiples" représentée par les algorithmes génétiques (approche AG). Nous utilisons des versions améliorées de chacune de ces métaheuristiques pour accélérer leur convergence et obtenir des résultats de bonne qualité.

Pour notre étude de cas, nous avons considéré comme exemple d'environnement la localité d'Abou Tachfine de la wilaya de Tlemcen d'une superficie de 8171.06 m^2 et dans laquelle se trouvent 80 intersections. Si par exemple, nous voulons placer 10 RSUs à 10 intersections parmi les 80 intersections de la zone de déploiement, nous aurons un très grand nombre de scénarios possibles comme le montre l'équation (3.1).

$$\binom{80}{10} = \frac{80!}{10! * 70!} = 16464921101 \quad (3.1)$$

De plus, si nous supposons que chaque scénario nécessite 1 ms pour être évalué, il faut 190 jours pour tester tous les scénarios. Par conséquent, une méthode approximative est nécessaire pour trouver une solution au problème du déploiement d'un nombre minimum de RSUs au niveau des intersections de la zone urbaine tout en garantissant une performance de routage acceptable. Nous modélisons notre approche comme illustré par la figure 3.2.

Comme mentionné ci-dessus, nous avons utilisé deux types de métaheuristiques pour traiter ce problème d'optimisation combinatoire :

- Approche GA basée sur les algorithmes génétiques.
- Approche SA basée sur le recuit simulé.

Avant de présenter chacune des méthodes de résolution relatives à chacune des métaheuristiques mentionnées ci-dessus, il est approprié de modéliser le problème d'optimisation combinatoire. Dans notre cas, la modélisation consiste à représenter l'environnement urbain concerné par des segments et des intersections où un segment représente une route entre deux intersections. De plus, dans le cas de l'approche par algorithme génétique, la taille du chromosome en termes de nombre de gènes est égale au nombre d'intersections dans la zone urbaine, où un gène correspond à une intersection et un chromosome à une solution réalisable. Dans ce contexte, le nombre de gènes dans un chromosome est égal à 80 où 80 représente le nombre d'intersections dans la zone urbaine considérée, et la valeur assignée à un gène (i) symbolise le placement d'une RSU à l'intersection (i) ou non. Elle est représentée comme suit :

- $gene(i) = 1$ veut dire qu'une RSU est placée au niveau de l'intersection (i).

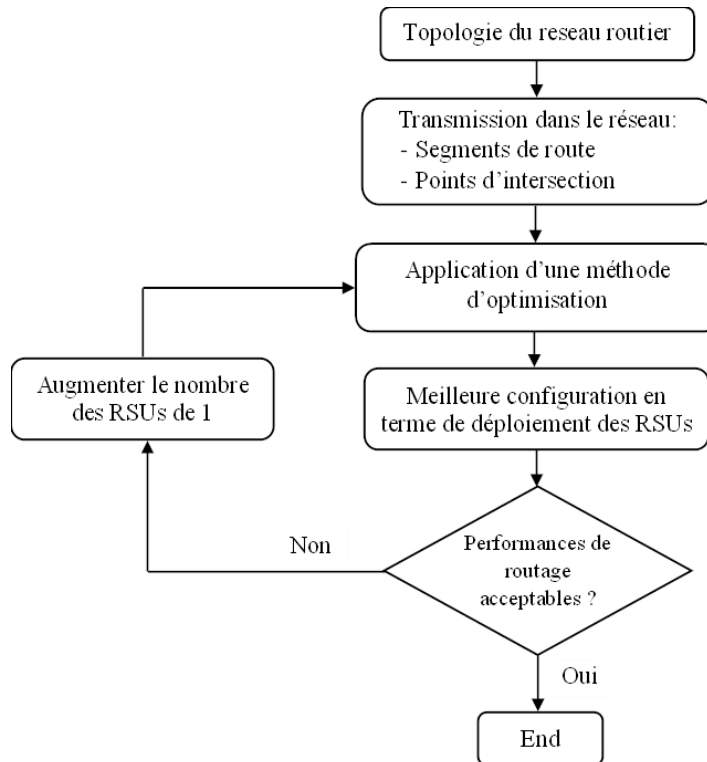


FIGURE 3.2 – Modélisation de la problématique

- $gene(i) = 0$ veut dire qu'aucune RSU n'est placée au niveau de l'intersection (i).
- *La fonction fitness* : elle consiste à calculer la zone totale couverte par toutes les RSUs installées dans la zone urbaine, comme le montre la figure 3.3, et la zone de chevauchement totale où plus d'une RSU couvre chacune des parties de cette zone de chevauchement, comme le montre la figure 3.4.

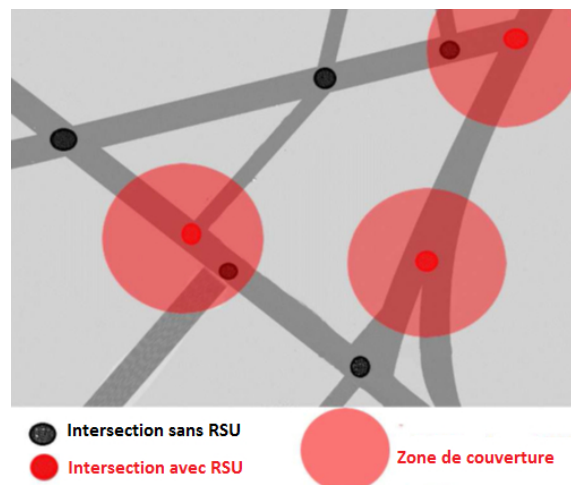


FIGURE 3.3 – Zones couvertes par les RSUs déployées

- *Couverture du voisinage* : L'algorithme considère la table de voisinage de l'intersection qui comporte une RSU et calcule, pour chaque segment de la table de voisinage,

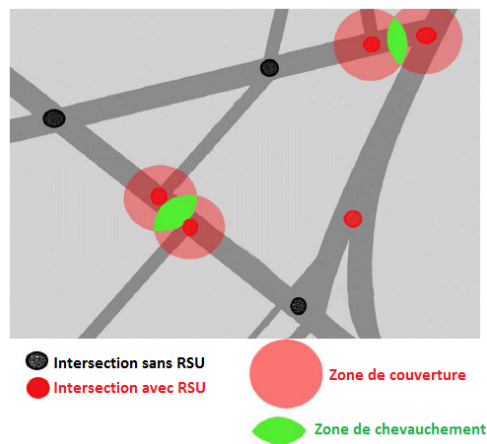


FIGURE 3.4 – Zones de chevauchement : zones couvertes par deux RSUs ou plus

la somme des distances qui sont inférieures ou égales à la portée de la RSU.

- *Couverture d'un segment* : L'algorithme considère le point d'intersection qui contient une RSU comme centre d'un cercle dont le rayon représente la portée de transmission de la RSU et recherche parmi les segments du réseau ceux appartenant au cercle (il calcule la somme de toutes les distances trouvées par l'algorithme).
- *Couverture totale de la zone* : C'est la somme de l'ensemble des distances trouvées en employant les deux algorithmes de couverture.
- *Zone de chevauchement* : Après avoir calculé la zone de couverture pour chaque intersection contenant une RSU, il peut y avoir des segments couverts par plus d'une RSU, comme le montre la figure 3.4. Ces segments sont divisés en plusieurs sous-segments d'une longueur d'un mètre, comme le montre la figure 3.5. Après avoir divisé le segment en plusieurs sous-segments, il peut y avoir deux intersections qui ont la même couverture.

Dans ce cas, le temps de transmission des RSUs est utilisé pour décider entre ces intersections. L'intersection dont le temps de transmission est le plus court sera choisie. Cette solution est ensuite améliorée à l'aide des deux algorithmes d'optimisation mentionnés ci-dessus (algorithmes génétiques et recuit simulé).

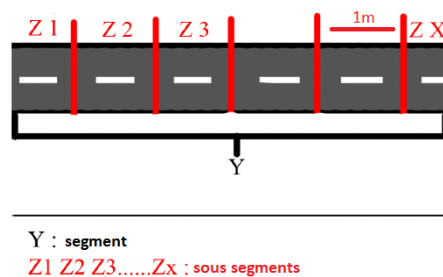


FIGURE 3.5 – Division des segments en sous segments de 1 mètre

3.4.1 Approches basées sur les algorithmes génétiques

Dans cette partie, nous présentons les deux approches basées sur les algorithmes génétiques pour traiter le problème abordé. La première approche concerne la version de base des algorithmes génétiques (GA-Basic) tandis que la seconde approche est une version améliorée de cette métaheuristique (GA-Improved). Cette dernière accélère la convergence et fournit des solutions de bonne qualité qui offrent des performances de routage très acceptables.

Pour illustrer la différence entre les deux approches qui utilisent des algorithmes génétiques, nous supposons que n est le nombre de RSUs à placer aux intersections. Par conséquent, tous les individus d'une génération doivent contenir n bits dont la valeur est 1. Cette différence consiste en les points suivants :

- Dans la version standard, l'initialisation de la population se fait de manière aléatoire, tandis que dans la version améliorée, un traitement préalable est effectué sur toutes les générations, y compris la génération initiale où la population contient des individus avec un très faible taux de chevauchement entre eux.
- Pendant l'opération de croisement, certains individus de la nouvelle génération peuvent contenir un nombre de bits m supérieur ou inférieur à n . Dans l'approche GA-Basic, si m est supérieur à n , alors $(m - n)$ bits sont aléatoirement mis à 0, tandis que dans l'approche GA-Improved, les bits mis à 0 sont ceux qui minimisent la couverture et augmentent la surface de chevauchement. En outre, si m est inférieur à n , alors $(n - m)$ bits seront commutés à 1 parmi les bits qui sont à 0. Dans l'approche GA-Basic, ces bits sont choisis de manière aléatoire, tandis que dans l'approche GA-Improved, leur choix est basé sur la surface de couverture et la surface de chevauchement.
- Lors de l'opération de mutation, dans l'approche GA-Basic, les bits concernés par la mutation sont choisis aléatoirement, tandis que dans l'approche GA-Improved, ce sont ceux qui augmentent la surface couverte par les RSUs et minimisent la surface des zones de chevauchement qui sont concernés par la mutation.

L'évaluation des individus de chaque population est suivie de la sélection des meilleurs individus en fonction de la fonction de fitness. Ensuite, l'opération de croisement sera effectuée, suivie de l'opération de mutation. L'opération de mutation dans l'approche GA-Improved est différente de celle de l'approche GA-Basic. En effet, dans l'approche GA-Basic, il y a une probabilité d'effectuer l'opération de mutation et, dans le cas où cette mutation survient, le choix des deux bits (l'un qui contient 1 et l'autre qui contient 0) est fait de manière aléatoire. Par contre, dans GA-Improved, l'opération de mutation concerne les individus qui augmentent la surface des zones de chevauchement. Le processus de recherche du nombre de RSUs à déployer sera répété jusqu'à ce qu'il y ait une stagnation c'est à dire ; pas de changement dans la qualité de la solution après un certain nombre de générations, ou jusqu'à ce que le nombre de générations fixé au départ soit atteint. Enfin, la meilleure solution parmi les solutions explorées est sélectionnée. L'approche basée sur les algorithmes génétiques dans ces deux versions est modélisée comme le montre la figure 3.6.

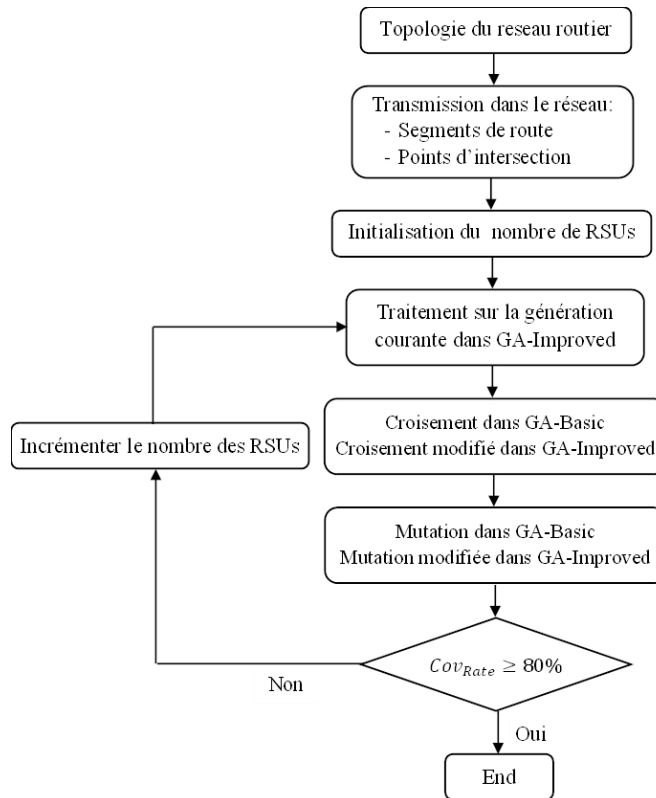


FIGURE 3.6 – Étapes de traitement par GA-Basic et GA-Improved

3.4.2 Approches basées sur le recuit simulé

Dans cette partie, nous exécutons la métaheuristique basée sur le recuit simulé en deux versions : Basique (SA-Basic) et améliorée (SA-Improved). Dans cette métaheuristique, le choix de la solution suivante est fait selon une probabilité qui dépend de la qualité des différents voisins :

- Les meilleurs voisins ont une probabilité élevée d'être acceptés comme solutions pour la prochaine itération.
- Les mauvais voisins ont une faible probabilité d'être admis pour la prochaine itération.

Dans les approches basées sur le recuit simulé, un paramètre θ appelé "Température" est utilisé. Il varie au cours du processus de recherche : Au début, la valeur de θ est élevée, puis elle diminue progressivement après chaque cycle et finit par tendre vers 0. La valeur de θ a un impact sur le processus d'exécution de l'approche :

- Si θ a une valeur élevée, tous les voisins ont presque la même probabilité d'être acceptés avec un avantage pour ceux qui améliorent la solution courante.
- Si θ a une valeur basse, une solution de voisinage qui renvoie une fitness plus petite que celle de la solution courante a une faible probabilité d'être choisie.
- Si la valeur de θ est nulle, aucune dégradation de la fonction de coût n'est acceptée.

Soit C l'ensemble des configurations possibles pour un nombre donné n de RSUs à déployer dans la zone urbaine, comme indiqué dans l'équation 3.2. La taille de l'ensemble C est calculée selon l'équation 3.3 ; elle représente le nombre de configurations possibles. La probabilité p de passer de la configuration C_i à la configuration C_j est donnée par l'équation 3.7 où C_i représente la configuration courante et C_j est une configuration dans le voisinage de C_i . C_j est considérée comme une configuration dans le voisinage de C_i si dans C_j il n'y a qu'une seule RSU dont l'emplacement est différent de celui de C_i .

$$C = \{C_1, C_2, \dots, C_m\} \quad (3.2)$$

$$|C| = \binom{80}{n} = \frac{80!}{n! * (80 - n)!} \quad (3.3)$$

La fonction fitness vise à augmenter la zone couverte Cov_{Area} par les RSUs déployées dans la zone urbaine et à réduire la surface de la zone de chevauchement $Overlap_{area}$, c'est-à-dire la surface des zones couvertes par plusieurs RSUs. Elle est exprimée par l'équation 3.4.

$$\begin{cases} Max(Cov_{Area}) \\ Min(Overlap_{area}) \end{cases} \quad (3.4)$$

A partir de la fonction fitness représentée par l'équation 3.4, nous pouvons déduire une nouvelle fonction fitness qui combine la zone couverte Cov_{Area} et la zone de chevauchement $Overlap_{area}$. Cette nouvelle fonction fitness est exprimée par l'équation 3.5.

$$Re_Cov_{Area} = Cov_{Area} - Overlap_{area} \quad (3.5)$$

La fonction de fitness représentée par l'équation 3.5 peut également être exprimée selon l'équation 3.6.

$$Cov_{Rate} = \frac{Re_Cov_{Area}}{Tot_{Area}} \quad (3.6)$$

où Re_Cov_{Area} représente la zone couverte pour une telle configuration à partir de laquelle la zone de chevauchement sera extraite, et Tot_{Area} représente la surface totale de la zone de déploiement.

La transition d'une configuration C_i à une configuration voisine C_j s'effectue selon la probabilité p présentée dans l'équation 3.7 :

$$p = \exp\left(-\frac{|Cov_{Rate}(C_j) - Cov_{Rate}(C_i)|}{k * \theta}\right) \quad (3.7)$$

où $Cov_{Rate}(C_i)$ et $Cov_{Rate}(C_j)$ représentent respectivement les taux de couverture des configurations C_i et C_j , k est un coefficient qui doit être adapté à l'application, (par exemple, $k = 1$) et θ est un paramètre formel qui représente la température et :

- Une configuration voisine C_j qui améliore Cov_{Rate} est acceptée si $Cov_{Rate}(C_j) > Cov_{Rate}(C_i)$.
- Une configuration voisine C_j dont la fitness $Cov_{Rate}(C_j)$ est inférieure à celle de la configuration courante C_i est acceptée avec une probabilité définie selon l'équation 3.7.

L'approche basée sur le recuit simulé est exécutée selon l'Algorithme 1 où, pour n RSUs à installer, une solution initiale C_0 est générée. Dans l'approche SA-Basic, cette solution initiale est générée de manière aléatoire, alors que dans l'approche SA-Improved, cette solution est générée après un prétraitement dans lequel nous évitons de placer des RSUs dans deux intersections voisines qui ne sont pas très éloignées. Ce processus accélère la convergence de l'algorithme puisqu'il ne tient pas compte de certaines configurations et permet donc d'obtenir de meilleures solutions.

A partir d'une configuration initiale C_0 et d'une température θ_0 , l'approche basée sur le recuit simulé passe d'une configuration C_i à une configuration C_j si C_j est meilleure que C_i ou avec une certaine probabilité si C_j n'est pas meilleure que C_i , comme le montre l'Algorithme 2. De plus, pour chaque température θ , m itérations sont effectuées pour trouver la meilleure configuration qui lui est correspondante. Par la suite, la valeur θ sera décrémentée et le processus est répété jusqu'à ce que θ atteigne une valeur seuil. A la fin de l'exécution de l'algorithme, la meilleure configuration trouvée parmi les configurations explorées est retournée.

Algorithm 1 : Approche basée sur le recuit simulé (SA-based)

```

1: Begin
   Input : Configuration  $C_i$ 
   Output : Configuration  $C_j$  qui maximise  $Cov_{Rate}$ 
           -  $C_i$  représente la configuration courante
           -  $Neighbor(C_c)$  est une fonction qui retourne aléatoirement un voisin ( $X$ ) de  $C_c$ 
2:  $C_c \leftarrow C_i$ ;
3:  $\theta \leftarrow \theta_0$ 
4: while  $\theta \geq Thresh$  do
5:    $m \leftarrow 0$ 
6:   repeat
7:      $X \leftarrow Neighbor(C_c)$ 
8:      $diff_{Rate} = Cov_{Rate}(X) - Cov_{Rate}(C_c)$ 
9:     if  $Accepte(diff_{Rate}, \theta)$  then
10:       $C_c = X$ 
11:     end if
12:      $m \leftarrow m + 1$ 
13:   until ( $m = N_\theta$ )
14:    $\theta = Decrease(\theta)$  (selon l'équation 3.12)
15: end while
16: End

```

Dans les métaheuristiques, le bon choix des paramètres initiaux a un impact positif sur la qualité des résultats et la vitesse de convergence vers une meilleure solution. Comme

Algorithm 2 : Accepter ou pas une transition de C_i à C_j

```

1: Begin
   Input : Configuration  $diff_{Rate}$  and  $\theta_0$ 
   Output : - Vrai si le taux de couverture augmente
             - Vrai si selon une certaine probabilité le taux de couverture diminue
             - Faux dans tous les autres cas
2: if ( $diff_{Rate} < 0$ ) then
3:   Return True;
4: else
5:    $pp = exp-\frac{diff_{Rate}}{\theta}$ 
6:   if ( $Alea(0, 1) < pp$ ) then
7:     Return True;
8:   else
9:     Return False;
10:  end if
11: end if
12: End

```

le nombre de configurations possibles est très élevé, une configuration initiale aléatoire peut nous éloigner de la solution optimale. Par conséquent, nous proposons de choisir plusieurs configurations et de calculer la moyenne de leurs performances. Dans ce cas, nous supposons que la probabilité d'être proche de la solution optimale est d'environ 0,5. Nous procédons formellement comme suit :

- Un certain nombre de configurations aléatoires est généré. Soit C_{Gen} l'ensemble des configurations générées et n la taille de l'ensemble C_{Gen} .

$$C_{Gen} = \{C_0, C_1, C_2, \dots, C_{n-1}\} \quad (3.8)$$

$$n = |C_{Gen}|$$

- Pour chaque configuration $C_i \in C_{Gen}$, on calcule $Cov_{Rate}(C_i)$,
- On trie les configurations en fonction de leurs fitness,
- On calcule la valeur médiane M (la valeur qui divise la distribution en deux parties égales) comme suit :

$$M = \begin{cases} \frac{(Cov_{Rate}(C_{\frac{n}{2}}) + Cov_{Rate}(C_{\frac{n}{2}+1}))}{2} & \text{Si } n \text{ est pair} \\ Cov_{Rate}(C_{\frac{n+1}{2}}) & \text{Si } n \text{ est impair} \end{cases} \quad (3.9)$$

- Le paramètre initial θ_0 est choisi selon l'équation 3.11. Il est déduit de l'équation 3.10.

$$pp = \exp\left(-\frac{M}{\theta_0}\right) = 0.5 \quad (3.10)$$

$$\Rightarrow \ln(pp) = \ln\left(\exp\left(-\frac{M}{\theta_0}\right)\right) = \ln\left(\frac{1}{2}\right)$$

$$\Rightarrow -\frac{M}{\theta_0} = -\ln(2)$$

$$\Rightarrow \theta_0 = \frac{M}{\ln(2)} = 1.45 * M \quad (3.11)$$

Pour la décrémentation du paramètre θ d'une itération à l'autre, une séquence géométrique exprimée par l'équation 3.12 a été proposée. On en déduit qu'après m itérations, la valeur de la température θ_n est donnée par l'équation 3.13.

$$\theta_n = R_\theta * \theta_{n-1} \quad (3.12)$$

$$\theta_n = \theta_0 * R_\theta^n \quad (3.13)$$

où θ_0 représente le premier terme et R_θ est le rapport commun pour cette séquence géométrique tel que présenté dans l'équation 3.16.

Pour le critère d'arrêt, le processus de recherche de la meilleure configuration s'arrête lorsque le paramètre θ_n est inférieur à une certaine valeur seuil, comme le montre l'équation 3.14. Pour chaque valeur de θ , 10 itérations sont effectuées avant de passer à la valeur suivante de θ , comme illustré dans l'algorithme 1 où $N_\theta = 10$. Le ratio commun R_θ pour la séquence géométrique est choisi selon l'équation 3.15. Donc, en fonction de ces paramètres, l'équation 3.16 peut être déduite de l'équation 3.15.

$$\theta_n < \theta_{Ratio} * \theta_0 \quad \text{and} \quad \theta_{Ratio} = 10^{-4} \quad (3.14)$$

$$R_\theta = \theta_{Ratio}^{1/N_\theta} \quad (3.15)$$

$$R_\theta = (10^{-4})^{1/10} = 0.3981 \quad (3.16)$$

3.5 Évaluation des performances

Dans cette partie, nous présentons l'évaluation des performances des approches proposées basées sur les algorithmes génétiques et le recuit simulé pour le déploiement efficace des RSUs dans une zone urbaine. Ces approches visent à déployer un nombre minimal de RSUs à certaines intersections parmi les 80 intersections composant le réseau routier de la zone Abou Tachfine de la wilaya de Tlemcen (Algérie) tout en garantissant des performances de routage acceptables. Ces performances concernent deux métriques :

- Taux de livraison des paquets (PDR) : C'est le rapport entre le nombre total de paquets reçus par la destination et le nombre total de paquets générés par les noeuds sources (véhicules).
- Délai de bout-en-bout : Représente la durée du trajet d'un paquet de la source à la destination.

Comme illustré précédemment dans la figure 3.2, les étapes du processus de résolution de problèmes sont résumées comme suit :

1. Appliquer la méthode d'optimisation choisie (approches basées sur les AGs ou sur le SA) pour trouver le nombre minimum de RSUs et leurs emplacements stratégiques.
2. Sélectionner le meilleur résultat obtenu par chaque méthode d'optimisation.
3. Simulation avec le résultat obtenu et évaluation des performances du protocole de routage développé dans [123] en termes de PDR et de délai de bout en bout.
4. Si les performances du protocole de routage [123] sont acceptables en termes de PDR et de délai de transmission, le processus de recherche du nombre minimal de RSUs s'arrête. Dans le cas contraire, le nombre de RSUs est incrémenté et le processus de recherche est répété (retour à l'étape 1).

Pour atteindre cet objectif, nous avons évalué les configurations retournées par chacune des méthodes d'optimisation sur le protocole de routage [123]. Les critères suivants ont été évalués :

- Le taux de livraison des messages envoyés par les véhicules sur un segment de route à destination de la RSU la plus proche.
- La zone de couverture minimale pour garantir le bon traitement des messages des véhicules par les RSUs.
- Le délai de transmission des messages des véhicules vers les RSUs.

3.5.1 Environnement de simulations

Les performances de routage de l'approche GA-based (GA-Basic et GA-Improved) et de l'approche SA-based (SA-Basic et SA-Improved) sont évaluées à l'aide des simulateurs OMNET-5.0 [133] et SUMO-0.29.0 "SUMO". Les simulations sont réalisées selon les paramètres indiqués dans le tableau 3.1. Chaque scénario est exécuté 10 fois et les résultats renvoyés par chacune des exécutions sont moyennés. Quelques obstacles sont introduits dans la zone de déploiement pour illustrer les simulations dans un environnement réaliste. De plus, chaque véhicule est équipé d'un fichier de carte numérique pour connaître l'environnement de simulation. Pour la couche MAC le protocole 802.11p [116] est utilisé. La zone urbaine est la localité Abou Tachfine située à Tlemcen contenant 80 intersections comme le montre la figure 3.7.

La mise en oeuvre des approches proposées s'effectue en deux phases. La première phase consiste à trouver le nombre minimum de RSUs à déployer et leurs emplacements de telle sorte qu'au moins 80% de la zone de déploiement soit couverte par des RSUs. La seconde phase exploite les résultats obtenus dans la première phase pour évaluer les performances de routage relatives à chaque approche proposée.



FIGURE 3.7 – Zone du réseau (Abou Tachfine, Tlemcen)

TABLE 3.1 – Paramètres de simulation

Paramètres	Valeurs
Durée de simulation	100 sec
Intervalle entre les beacons	0.5 sec
Protocole MAC	802.11p
Zone du réseau	Abou Tachfine, Tlemcen (8171.06 m^2)
Portée de transmission	250 m
Modèle de mobilité	SUMO 0.29.0
Vitesse moyenne des véhicules	40 km/h
Nombre d'intersections	80
Modèle de trafic	UDBBasicAPP
Taille des paquets	128 octets
Nombre de véhicules	50-400

3.5.2 Résultats des simulations

Dans cette partie, nous présentons les résultats de simulation obtenus par chaque approche et nous les comparons en termes de taux de couverture, de taux de chevauchement, de délai de bout en bout et de PDR.

a) Taux de couverture

Le tableau 3.2 et la figure 3.8 illustrent le taux de couverture de la zone urbaine en fonction du nombre de RSUs déployées pour chaque approche. Nous remarquons qu'avec le même nombre de RSUs déployées, l'approche "GA-Improved" fournit un taux de couverture élevé comparé aux autres approches (GA-Basic, SA-Basic, et SA-Improved) grâce à l'opération de prétraitement lors de la génération d'une population. Par exemple, si au cours de l'opération de croisement, le nombre de bits qui sont à "1" dans un enfant généré est supérieur au nombre attendu, les autres bits qui passeront à "0" sont choisis parmi ceux qui minimisent le taux de couverture alors que dans l'approche GA-Basic, ils sont choisis au hasard. De même, dans le cas où le nombre de bits "1" dans un enfant généré est plus faible que prévu, les autres bits "1" qui seront ajoutés sont ceux qui maximisent le taux de couverture de la zone urbaine, alors que dans la version standard (GA-Basic), ces bits sont choisis de manière aléatoire. Par exemple, le déploiement de 30 RSUs permet d'obtenir un taux de couverture de 98,80% avec l'approche GA-Improved, alors qu'avec les autres approches GA-Basic, SA-Basic et SA-Improved le taux de couverture atteint respectivement 93,53 %, 90,10 % et 95,04 % . En outre, pour assurer une couverture complète de la zone urbaine, l'approche GA-Improved déploie 42 RSUs tandis que les approches GA-Basic, SA-Basic et SA-Improved déploient respectivement 52, 61 et 50 RSUs.

TABLE 3.2 – Taux de couverture dans GA-Basic, GA-Improved, SA-Basic et SA-Improved

Nombre de RSUs	GA-Basic		GA-Improved		SA-Basic		SA-Improved	
	S-Couv	T-Couv	S-Couv	T-Couv	S-Couv	T-Couv	S-Couv	T-Couv
10	5721.44	70.02	6284.00	76.91	5473.52	66.99	5777.90	70.71
20	7032.36	86.06	7564.12	92.57	6934.93	84.87	7220.36	88.37
30	7642.61	93.53	8073.19	98.80	7362.19	90.10	7765.80	95.04
40	7902.49	96.71	8131.31	99.51	7806.60	95.54	8063.02	98.68
50	8127.64	99.47	8171.06	100.00	8062.76	98.67	8158.89	99.85
60	8171.06	100.00	8171.06	100.00	8144.08	99.67	8171.06	100.00
70	8171.06	100.00	8171.06	100.00	8171.06	100.00	8171.06	100.00
80	8171.06	100.00	8171.06	100.00	8171.06	100.00	8171.06	100.00

¹ S-Couv : Surface couverte exprimée en (m^2)

² T-Couv : Taux de couverture exprimé en (%)

b) Taux de chevauchement des zones couvertes

Le tableau 3.3 et la figure 3.9 montrent le taux de chevauchement entre les zones couvertes par deux RSUs ou plus en fonction du nombre de RSUs pour chaque approche. Nous remarquons que les deux approches GA-Improved et SA-Improved fournissent un taux de chevauchement très faible par rapport aux deux autres approches GA-Basic et SA-Basic.

Les bonnes performances de l'approche GA-Improved ont été obtenues grâce à des opérations de croisement et de mutation améliorées, tandis que celles de l'approche SA-

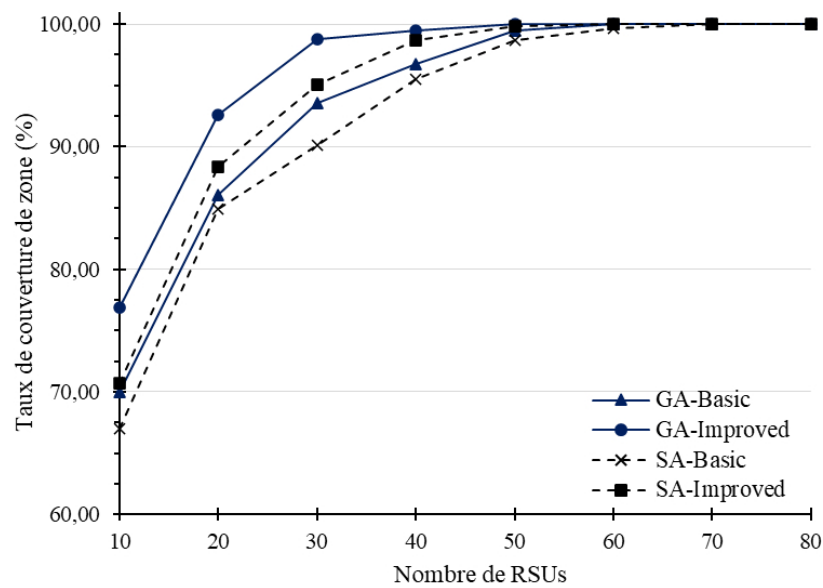


FIGURE 3.8 – Taux de couverture dans GA-Basic, GA-Improved, SA-Basic et SA-Improved

Improved a été obtenue grâce à l'utilisation d'une solution de voisinage qui évite l'installation de RSUs dans deux intersections adjacentes peu éloignées l'une de l'autre. Par exemple, lorsque le nombre de RSUs déployés est de 30, le taux de chevauchement atteint par GA-Improved, GA-Basic, SA-Basic et SA-Improved est respectivement de 1,37%, 1,65%, 1,68% et 1,46%.

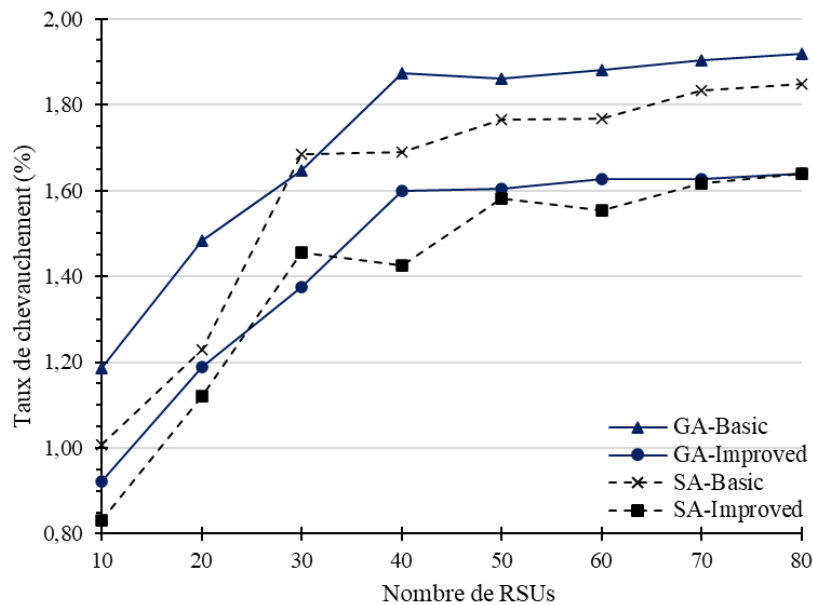


FIGURE 3.9 – Taux de chevauchement dans GA-Basic, GA-Improved, SA-Basic et SA-Improved

TABLE 3.3 – Taux de chevauchement dans GA-Basic, GA-Improved, SA-Basic et SA-Improved

Nombre de RSUs	GA-Basic		GA-Improved		SA-Basic		SA-Improved	
	S-Chauv	T-Chauv	S-Chauv	T-Chauv	S-Chauv	T-Chauv	S-Chauv	T-Chauv
10	67.86	1.19	58.00	0.92	55.20	1.01	48.00	0.83
20	104.30	1.48	90.00	1.19	85.15	1.23	81.00	1.12
30	125.87	1.65	111.00	1.37	123.95	1.68	113.00	1.46
40	148.10	1.87	130.00	1.60	132.25	1.69	115.00	1.43
50	151.27	1.86	131.00	1.60	142.35	1.77	129.00	1.58
60	153.61	1.88	133.00	1.63	144.05	1.77	127.00	1.55
70	155.61	1.90	133.00	1.63	149.80	1.83	132.00	1.62
80	156.78	1.92	134.00	1.64	151.10	1.85	134.00	1.64

¹ S-Chauv : Surface chevauchée exprimée en (m^2)

² T-Chauv : Taux de chevauchement exprimé en (%)

c) Taux de livraison des paquets (PDR)

Le tableau 3.4 et la figure 3.10 montrent le taux de livraison des paquets en fonction du nombre de RSUs pour chaque approche. Nous constatons que l'approche GA-Improved fournit de meilleures performances que les autres approches. En effet, dans l'approche GA-Improved, le PDR a atteint 100 % en déployant seulement 30 RSUs, alors que pour le même nombre de RSUs, le PDR n'atteint que 97,52%, 94,88% et 97,29% respectivement dans GA-Basic, SA-Basic et SA-Improved.

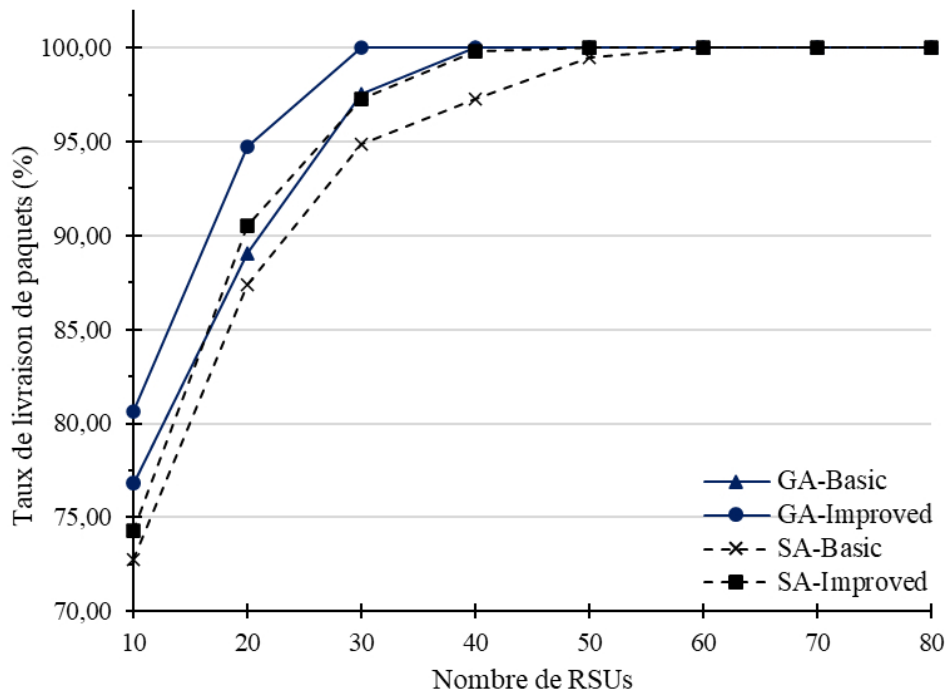


FIGURE 3.10 – Taux de livraison de paquets vs. Nombre de RSUs

Les bonnes performances obtenues par l'approche GA-Improved sont justifiées par le bon déploiement des RSUs qui permet d'avoir un taux de couverture élevé et de minimiser le taux de chevauchement. De plus, l'approche GA-Improved a besoin de déployer seulement 37.5% des RSUs sur les 80 intersections pour atteindre un PDR de 100% alors que les approches GA-Basic, SA-Basic, et SA-Improved ont besoin, respectivement, de 46,25%, 65% et 52,50% pour avoir cette même performance. Les résultats acceptables fournis par GA-Improved sont justifiés par la bonne sélection des intersections où les RSUs seront installés. En effet, la localisation des RSUs a un impact sur la qualité de la solution.

TABLE 3.4 – PDR dans GA-Basic, GA-Improved, SA-Basic and SA-Improved

Nombre de RSUs	Taux de livraison de paquets (%)			
	GA-Basic	GA-Improved	SA-Basic	SA-Improved
10	76.79	80.63	72.72	74.32
20	89.06	94.74	87.37	90.54
30	97.52	100.00	94.88	97.29
40	100.00	100.00	97.29	99.78
50	100.00	100.00	99.50	100.00
60	100.00	100.00	100.00	100.00
70	100.00	100.00	100.00	100.00
80	100.00	100.00	100.00	100.00

d) Délai de bout en bout

La figure 3.11 illustre l'évolution du délai de transmission en fonction du nombre de RSUs lorsque chacune des approches de déploiement des RSUs dans la zone urbaine est utilisée. Le délai de transmission de bout en bout est relativement lié à l'overhead, ce qui est expliqué par la figure 3.11. Nous pouvons voir que le délai fourni par le protocole lors du déploiement des RSUs par chaque approche diminue au fur et à mesure que le nombre de RSUs augmente, car l'overhead entraîne une congestion du réseau lorsque le nombre de RSUs est faible. Ceci entraîne un délai de transmission plus important. Nous pouvons également constater que le déploiement des RSUs effectué par l'approche GA-Improved a un délai plus faible par rapport aux autres approches. Le délai de l'approche GA-Improved est de 0,22 ms alors qu'il est respectivement de 0,24, 0,32 et 0,34 ms pour les approches SA-Improved, GA-Basic et SA-Basic dans les meilleurs cas.

Chacune des approches proposées pour déployer un nombre réduit de RSUs dans une zone urbaine fournit une solution proche de la solution optimale après avoir exploré un grand nombre de solutions.

D'après les résultats obtenus, il a été révélé que l'approche GA-Improved fournit une solution de meilleure qualité par rapport aux autres approches (GA-Basic, SA-Basic, et SA-Improved) en raison de l'amélioration des opérations de croisement et de mutation. En outre, l'approche SA-Improved offre une solution de meilleure qualité que celles fournies par l'approche GA-Basic grâce à l'évitement de l'installation de RSUs au niveau de deux intersections voisines qui ne sont pas très éloignées.

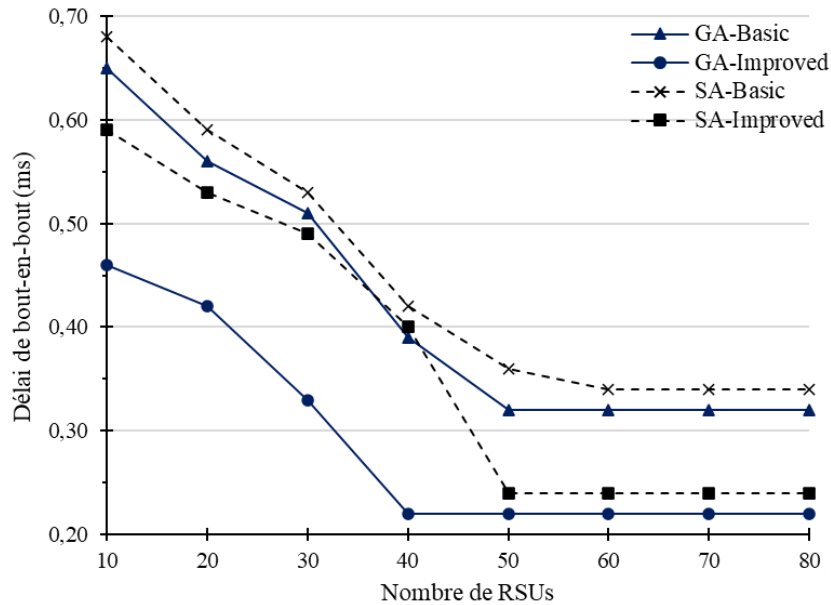


FIGURE 3.11 – Délai de bout en bout vs. Nombre de RSUs

3.6 Conclusion

Dans ce chapitre, nous avons proposé quatre approches (GA-Basic, GA-Improved, SA-Basic et SA-Improved) pour parvenir à un déploiement optimal des RSUs au niveau des intersections dans une zone urbaine tout en garantissant des performances de routage acceptables. Ces approches sont basées sur une métaheuristique d’algorithme génétique et une métaheuristique de recuit simulé dans leurs versions standard (GA-Basic et SA-Basic) ainsi que leurs versions améliorées (GA-Improved et SA-Improved).

Pour illustrer les avantages de chacune de ces approches, les performances d’un protocole de routage en termes de PDR et de délai de bout en bout ont été évaluées en fonction du nombre de RSUs déployés dans la zone urbaine. Par conséquent, nous avons implémenté et évalué les approches proposées à travers des simulations conductives en utilisant le simulateur OMNET++.

Les résultats obtenus ont montré que l’approche GA-Improved permet de réduire le nombre de RSUs tout en permettant une meilleure performance de routage en termes de PDR et de délai de bout en bout par rapport aux autres approches. L’approche GA-Improved n’a besoin de déployer que 37,5 % des RSUs sur les 80 intersections pour atteindre un taux de livraison de paquets de 100 %, alors que les approches GA-Basic, SA-Basic et SA-Improved ont besoin, respectivement, de 46,25%, 65% et 52,50 % pour obtenir cette même performance. De plus, les versions améliorées proposées accélèrent le processus de convergence du déploiement des RSUs et obtiennent de meilleurs résultats.

**Déploiement efficace des RSUs dans
les VANETs basé sur les méthodes
d'optimisation**

Gestion du trafic routier dans les VANETs basée sur le cloud computing

4.1 Introduction

Au cours des deux dernières décennies, le nombre de véhicules a explosé et les routes sont devenues de plus en plus encombrées. Par conséquent, les embouteillages et les congestions sont devenus fréquents et se produisent lorsque la demande de trafic routier dépasse la capacité offerte par les infrastructures de transport communes. Ceci représente un sérieux problème pour le conducteur qui lui cause un stress permanent sur la route et beaucoup de retard pour atteindre sa destination. Motivés par ces considérations, nous avons tenté de trouver une solution en proposant un algorithme basé sur les VANETs et le Cloud computing pour programmer les feux de circulation tricolores afin de fluidifier le trafic routier et limiter les retards, appelé "Cloud Computing based Intelligent Traffic Light Protocol (CCITL)". Dans cet algorithme, nous utilisons le Cloud computing en impliquant le Cloud conventionnel et les Clouds véhiculaires afin d'avoir une vision globale du trafic et des conditions routières et de profiter des capacités de calcul du Cloud pour calculer au mieux les formules les plus adaptées programmant efficacement les feux tricolores à chaque carrefour en synchronisant les feux avec leurs durées de phase de manière dynamique afin d'organiser au mieux le trafic routier en assurant ainsi, la fluidité du trafic et l'arrivée à destination dans des délais courts tout en minimisant les temps d'attente sur les différents segments du réseau routier.

La simulation et l'évaluation des résultats ont été menés à l'aide du simulateur d'événements discrets OMNeT++ avec le simulateur de mobilité urbaine "SUMO". Les résultats obtenus ont montré que CCITL a grandement contribué à la réduction de la congestion routière ainsi qu'à la réduction du temps d'arrêt des véhicules dans différents segments, diminuant ainsi les retards et l'attente dans les sections de route.

4.2 Contexte

Les technologies de communication sans fil et les progrès des systèmes embarqués ont fait émerger l'Internet des objets (IoT) dans notre vie quotidienne ; les objets sont devenus intelligents et dotés de plusieurs technologies et ressources informatiques pour fonctionner

automatiquement rendant ainsi la vie des utilisateurs plus facile.

L'un des domaines les plus pertinents de l'Internet des objets "IoT" est l'Internet des véhicules [134] "Internet of Vehicles (IoV)". Ce domaine a conduit à l'apparition des réseaux véhiculaires, qui sont des réseaux mobiles ad hoc appelés "VANETs" [28, 135] et le système de transport intelligent "ITS" [11] devient un composant indispensable. Il vise à atteindre l'efficacité du trafic en minimisant les problèmes. L'ITS fournit aux utilisateurs des informations préalables sur le trafic, ce qui réduit leur temps de trajet et améliore leur sécurité et leur confort. L'ITS comprend six composants fondamentaux qui sont ; les systèmes avancés d'information des voyageurs, les systèmes avancés de gestion des transports, les systèmes avancés de contrôle des véhicules, les systèmes avancés de transport public, les systèmes avancés de la gestion des véhicules commerciaux, et les systèmes avancés de transport urbain [136].

De nos jours, les chercheurs de l'ITS utilisent de plus en plus le Cloud computing pour améliorer les ressources des véhicules en répondant aux besoins des véhicules en terme de capacité de traitement, de capacité d'analyse, de capacité de calcul et de capacité de stockage afin de supporter et servir toutes les applications des conducteurs pour leur confort et leur sécurité et bien plus encore. Les véhicules peuvent ainsi tirer parti des avantages du Cloud computing pour répondre à certaines exigences [14]. En effet, en utilisant le Cloud computing dans les ITS, nous disposerons d'une vue globale du réseau routier et de capacités de calcul plus grandes pour trouver l'itinéraire le plus approprié pour qu'un conducteur atteigne sa destination tout en évitant les embouteillages dans un laps de temps optimal.

L'utilisation du Cloud computing joue également un rôle important dans la gestion de la sécurité. Ainsi, en ayant une vue globale du réseau routier, les conducteurs sont alertés de tout danger éventuel sur la route et conseillés d'emprunter un autre itinéraire. Le Fog computing contribue également grandement à la sécurité. En effet, les nœuds Fog avertissent les conducteurs dès qu'un accident se produit et traitent les alertes de sécurité en un temps record pour avertir les autorités ou alerter les services d'urgence, comme les ambulances, afin de porter secours aux personnes en danger. Dans [137], les auteurs ont abordé l'aspect sécurité dans les ITS. Ils ont proposé un framework de sécurité utilisant le Fog computing pour le système de contrôle des feux de circulation afin de fournir des services de sécurité avec des hypothèses réalistes. Dans [138], les chercheurs ont également abordé l'aspect sécurité en impliquant le Fog computing ; ils ont proposé deux systèmes sécurisés de contrôle intelligent des feux de circulation utilisant le Fog computing, basés sur la robustesse du puzzle de collision de hachage et de Diffie-Hellman.

Au cours de ces dernières années, le nombre de véhicules a considérablement augmenté et les routes sont désormais de plus en plus congestionnées. En conséquence, les embouteillages et la congestion sont devenus fréquents et se produisent lorsque la demande de trafic routier excède la capacité assurée par les infrastructures de transport communes. Les ITS sont apparus comme un moyen efficace pour palier ce type de problèmes, puisque la technologie des VANETs permet de fournir des réseaux sans fil à l'aide de véhicules mobiles, et les ITS offrent de nombreuses applications pour les véhicules, y compris les informations sur la circulation ou les accidents sur les routes, l'emplacement des postes de police et les messages d'urgence ainsi que les messages d'infodivertissement tels que les publicités, ...etc. L'installation de feux tricolores aux carrefours routiers fait partie des moyens efficaces pour gérer le trafic routier et éviter la congestion des routes et réduire les

temps d'attente sur les routes. Agrémentés par le Cloud computing, les feux de circulation intelligents pour la gestion du trafic apporteraient une amélioration significative en terme de fluidité de la circulation routière et de réduction des temps d'attente sur les différents segments.

Dans ce qui suit, nous décrivons les travaux connexes à cette problématique ainsi que les différentes méthodes de contrôle des signaux de circulation, nous détaillons notre contribution et enfin, nous présentons les résultats obtenus.

4.3 Travaux connexes

Au cours de la dernière décennie, avec l'énorme croissance du nombre de véhicules dans le monde, les embouteillages et la congestion routière ont augmenté considérablement, ce qui a incité les chercheurs à se concentrer sur des solutions intelligentes pour gérer le trafic à l'aide des ITS qui s'appuient sur les véhicules intelligents pour fournir de nombreux services et applications aux conducteurs afin qu'ils puissent conduire de manière sûre et confortable tout en arrivant à leur destination dans les plus brefs délais.

Dans [139], les auteurs ont proposé un modèle d'apprentissage par renforcement profond pour résoudre le problème du contrôle des feux de circulation, appelé IntelliLight. Ils ont mené des expériences approfondies en utilisant des expériences virtuelles et réelles en extrayant un ensemble de données de trafic réel à grande échelle à partir de caméras de surveillance pour tester leur modèle. Ils ont présenté des observations approfondies et des études de cas pour comprendre comment l'agent s'adapte à l'évolution du trafic, en complétant la mesure quantitative des récompenses pour aider à générer des règles de circulation pour une application dans le monde réel.

Dans [140], les auteurs ont proposé un protocole de négociation intelligent tenant compte du contexte, appelé "Intelligent Context-Aware Negotiation Protocol (ICANP)". ICANP permet la négociation de la phase verte entre les véhicules et le feu de circulation à l'intersection signalée. Il optimise le temps de passage en utilisant un algorithme génétique "Genetic Algorithm (GA)" pour réduire le temps d'attente des véhicules. De plus, les auteurs ont introduit un protocole de négociation basé sur la priorité, appelé "Intelligent Context-Aware Priority Protocol (ICAPP)". Ce protocole prend en compte la priorité des véhicules. Il interrompt le temps vert des véhicules en négociation dès qu'un véhicule ayant une certaine priorité arrive à l'intersection et ce véhicule traverse sans temps d'attente ni négociation.

Dans [141], les auteurs ont développé un protocole de contrôle intelligent des feux de circulation appelé "Intelligent Traffic Light Controlling (ITLC)". ITLC recueille des informations sur les caractéristiques du trafic en temps réel des véhicules traversant le carrefour ; la vitesse du trafic, le nombre de véhicules en mouvement, les volumes de flux de trafic et tout le reste. Ceci, en utilisant la communication VANET pour programmer efficacement les phases de chaque feu de circulation isolé. En outre, les auteurs ont considéré "Arterial Traffic Lights (ATL)", un autre protocole plus sélectif pour la gestion des feux de circulation afin de contrôler le trafic aux carrefours. ATL prend en compte les feux voisins et les caractéristiques du trafic en temps réel pour programmer les feux verts et rouges.

Dans [142], les chercheurs ont proposé un modèle de simulation pour résoudre le problème des embouteillages en faisant un contrôle automatique des feux de circulation. Ils

ont divisé le modèle de simulation en six parties, chaque partie contenant une route avec trois carrefours. Ils ont estimé le nombre de véhicules partant et arrivant sur la route. Lorsqu'il est combiné aux carrefours avec un modèle de simulation de contrôle des feux de circulation, le système expert réduit le temps d'attente moyen des véhicules à chaque carrefour.

Dans [143], les auteurs ont développé un système appelé e-NOTIFY pour alerter le centre de contrôle via les réseaux VANETs lorsqu'un accident se produit. e-NOTIFY se concentre sur l'amélioration de la récupération après l'accident grâce à une gestion efficace et rapide des ressources de secours d'urgence. Cela permet d'augmenter la probabilité de sauvetage des personnes blessées dans des accidents de la route.

Dans [144], les auteurs ont proposé un protocole de gestion intelligente du trafic appelé "Smart Traffic Management Protocol (STMP)". Ce protocole réduit le niveau de trafic dans la ville en utilisant les communications V2V et V2I. Dans le protocole STMP, des capteurs de surveillance sont placés sur chaque route. En outre, la connaissance de la surveillance en temps réel et de l'état des routes permet de détecter les routes encombrées et de prendre ainsi des mesures préventives.

Dans [145], les auteurs ont étudié le problème du contrôle adaptatif des feux de circulation. Ils ont pris en considération les facteurs environnementaux et le comportement des conducteurs en tenant compte des routes secondaires urbaines et du trafic routier. Ils ont développé un système de réseau artériel, appelé "ARTerial network SYStem (ARTSYS)", dans lequel chaque carrefour est contrôlé par son propre flux de trafic, son propre type de véhicules et peut coopérer avec les carrefours voisins.

Dans [146], les auteurs ont proposé un algorithme de contrôle adaptatif des feux de circulation. Ils ont utilisé les informations de position et de vitesse en temps réel de chaque véhicule pour améliorer le contrôle des feux aux carrefours. Ils ont développé un algorithme de planification efficace pour planifier les phases de chaque cycle de feux et réduire le retard des véhicules en mouvement.

Le travail présenté dans [138] est dédié à l'aspect sécurité dans les systèmes de signalisation. Dans ce travail, les auteurs ont fait appel au Fog computing pour assurer la protection des conducteurs. Ils ont supposé que les feux de circulation représentent des dispositifs Fog et ont développé deux schémas sécurisés pour le contrôle des feux de circulation. Le premier schéma utilise le puzzle Diffie-Hellman pour faire face aux attaques par déni de service et le second implique le puzzle de hachage de collision. Le deuxième schéma est beaucoup plus efficace que le premier, surtout lorsque la densité du trafic est élevée.

Les auteurs dans [147], optimisent la séquence des feux de circulation pour une seule intersection en utilisant le réseau neuronal Hopfield "Hopfield Neural Network (HNN)" afin de gérer la congestion du trafic et réduire les accidents de la route. Ils ont utilisé une approche basée sur les algorithmes génétiques pour déterminer le meilleur moment pour la phase du feu vert. La meilleure séquence de la phase du feu vert est fournie par le HNN après presque seize itérations. Lorsque le temps de la phase du feu vert atteint le maximum, le flux du trafic est à son maximum ; il augmente tant que la phase de temps du feu vert augmente.

Dans [148], les auteurs ont proposé une approche holistique tenant compte d'une grande zone urbaine appelée "Holistic Intelligent for Traffic Urban Lights (HITUL)". Ce système soutient les décisions des gestionnaires de trafic et génère une meilleure planifi-

cation des feux de circulation. Il calcule les plans de feux de circulation optimaux à l'aide de micro-simulations et de techniques bio-inspirées et prend en compte les conditions de circulation réelles.

Dans [149], les auteurs ont proposé une approche basée sur les algorithmes génétiques pour l'optimisation des signaux de contrôle du trafic. Ils ont pris en compte l'effet de confusion lorsque les conducteurs changent d'itinéraire en faveur d'un itinéraire concurrent, ce qui réduit la durée du trajet en raison de la modification de la synchronisation des signaux. Les auteurs incluent un modèle d'équilibre du réseau pour optimiser le cycle de signalisation en anticipant l'impact sur le modèle d'acheminement du trafic. Un algorithme génétique est conçu pour résoudre le problème résultant ; il utilise une fonction de fitness illustrative représentée par le temps de trajet total sur le réseau.

Les auteurs dans [150], ont considéré le problème de contrôle des feux de circulation aux intersections isolées. Ils ont proposé une méthode de contrôle du signal avec reconnaissance de la technologie des véhicules en considérant le problème comme un problème d'ordonnement d'une seule machine. Ils ont programmé les véhicules qui s'approchent du carrefour pour qu'ils transitent dans les plus brefs délais en proposant une approche *branch and bound*.

Les auteurs dans [150] ont étudié le problème du contrôle des feux de circulation aux intersections isolées. Ils ont proposé une méthode de contrôle des feux avec reconnaissance de la technologie des véhicules en considérant le problème comme un problème d'ordonnement d'une seule machine. Ils ont programmé les véhicules approchant de l'intersection pour qu'ils transitent dans le temps le plus court possible selon une approche basée sur la méthode "branch and bound".

Un protocole d'ordonnement des feux de circulation tenant compte du contexte, appelé "Context-Aware Traffic Light Scheduling (CATLS)", est proposé dans [151]. CATLS prend en compte les caractéristiques du flux de trafic pour programmer efficacement les phases de chaque feu de circulation. La phase verte du feu de circulation peut être interrompue à tout moment par des véhicules d'urgence pour traverser l'intersection. CATLS programme les phases en fonction de la priorité de chaque véhicule d'urgence lorsque plusieurs véhicules d'urgence veulent traverser l'intersection en même temps. Si deux véhicules ont la même priorité sur des voies différentes, CATLS favorise le véhicule qui se trouve sur la voie la plus encombrée.

La plupart de ces travaux ont abordé le problème de la congestion routière et du temps d'attente. Cependant, ces travaux s'appuient uniquement sur les réseaux VANETs pour gérer le trafic et n'ont qu'une vue locale de l'intersection pour planifier les feux de circulation et organiser le trafic. Dans notre proposition, nous combinons le Cloud computing avec les réseaux VANETs pour optimiser la planification des feux de circulation dans un réseau routier, ce qui offre l'avantage d'avoir une vue globale du réseau routier et d'ajouter plus de ressources en termes de capacité de calcul et de vitesse de traitement afin de traiter de grandes cartes routières dans un temps optimal.

4.4 Méthodes de contrôle des feux de circulation

Les intersections routières sont coordonnées et contrôlées par des feux de signalisation afin de garantir la sécurité routière et d'améliorer l'efficacité du trafic tout en réduisant la congestion routière. Dans cette section, nous examinons les différentes méthodes de

contrôle des feux de circulation.

4.4.1 La méthode webster

Dans le domaine de l'ingénierie des transports, pour une intersection unique c'est-à-dire une intersection isolée, le plan des feux de signalisation consiste généralement en une longueur de cycle préprogrammée, une séquence de phases basée sur un cycle fixe et une division des phases [152].

Webster [153] a proposé l'un des premiers modèles de contrôle des feux de circulation et une méthode de calcul du temps de cycle et de division en phases pour une intersection individuelle. Si le trafic routier est constant pendant une période donnée, le calcul de la durée du cycle est généré par l'équation (4.1).

$$C_{des}(V_c) = \frac{N \times t_L}{1 - \frac{V_c}{3600 \times FHP \times (\frac{v}{c})}} \quad (4.1)$$

où N est le nombre de phases, t_L est le temps de perte total par phase, qui peut être considéré comme un paramètre lié au temps "tout rouge" et à l'accélération et la décélération des véhicules. h est le plus petit intervalle de temps entre des véhicules successifs passant un point. FHP (Rush Hour Factor) est un paramètre mesurant les fluctuations de la demande de trafic aux heures de pointe. (v/c) est le volume souhaité divisé par le ratio de capacité, qui indique le degré d'occupation de l'intersection dans un contexte de synchronisation des signaux. Ces paramètres varient généralement en fonction des conditions de circulation.

V_c , comme exprimée par l'équation (4.2), est la somme des volumes des voies les plus encombrées.

$$V_c = \sum_1^N V_c^i \quad (4.2)$$

Elle représente la somme de tous les volumes des canaux critiques, ce qui indique le niveau d'occupation de l'intersection dans le contexte de la synchronisation du signal, où V_c^i représente le volume du canal critique pour la phase i . Un canal critique est le chemin d'approche avec le taux de trafic le plus élevé jusqu'au flux de saturation dans une phase, généralement indiqué par la longueur de la file d'attente [152].

Une fois la durée du cycle calculée, le temps vert est divisé par la durée du cycle, comme l'indique l'équation (4.3).

$$\frac{t_i}{t_j} = \frac{V_c^i}{V_c^j} \quad (4.3)$$

où t_i et t_j représentent la durée des phases pour les phases i et j .

Les équations (4.1) et (4.2) peuvent être exprimées différemment dans le cas d'une application en temps réel. Lorsque le trafic routier est régulier, l'approche de Webster peut réduire le temps de transit des véhicules traversant le carrefour ou augmenter la capacité du carrefour. De plus, si le trafic routier est régulier pendant une courte période et qu'aucune répétition du cycle ne subsiste (c'est-à-dire que $FHP = 1$ et $v/c = 1$), l'équation (4.2) peut être présentée comme suit :

$$\left(1 - \frac{N \times t_L}{C_{des}}\right) \frac{3600}{h} = V_c \quad (4.4)$$

Lorsque la capacité est égale à la demande de trafic V_c , la longueur du cycle C_{des} est la valeur minimale qui satisfait étroitement la demande de trafic. Pour une demande de trafic donnée, si une longueur de cycle de signal inférieure à C_{des} est appliquée, la longueur de la file d'attente continuera d'augmenter et l'intersection sera sursaturée. Si la longueur du cycle de signalisation est supérieure à C_{des} , le retard moyen de chaque véhicule à l'intersection augmente linéairement avec la longueur du cycle.

4.4.2 La méthode GreenWave

Le contrôle GreenWave [154] tente généralement de maximiser la largeur de la bande passante des ondes vertes publiques. Cela se produit lorsqu'une série de feux de circulation est coordonnée pour permettre un flux de circulation continu sur plusieurs intersections dans une direction principale ; c'est à dire, passer un nombre moyen d'intersections sans arrêt sur un itinéraire. Alors que la méthode Webster génère le plan de signalisation d'une seule intersection, cette méthode vise à mettre en œuvre la coordination afin d'optimiser les décalages entre les temps de signalisation des feux de circulation adjacents. Cela suppose que toutes les intersections partagent la même longueur de cycle, qui est la valeur maximale des longueurs de cycle données pour les intersections individuelles. Les décalages entre les intersections sont calculés par l'équation (4.5).

$$\Delta t_{i,j} = \frac{L_{i,j}}{v} \quad (4.5)$$

où $L_{i,j}$ est la longueur de la route qui relie les intersections i et j , et v est la vitesse estimée du mouvement des véhicules sur la route.

La méthode GreenWave est populaire dans le domaine du contrôle des signaux en raison de son efficacité et de sa simplicité, elle a le potentiel de :

- Améliorer l'efficacité du trafic.
- Contrôler la vitesse de circulation dans les zones urbaines.
- Réduire le temps d'attente des véhicules sur les routes secondaires.
- Réduire la consommation du carburant des véhicules.

La méthode GreenWave peut également être désavantageuse dans certains cas, notamment dans le cas où le facteur de sécurité au niveau du passage pour piétons est ignoré, ce qui entraîne des risques pour la sécurité des piétons lors de la traversée de la rue.

4.4.3 La méthode Maxband

La méthode Maxband [155] prend en compte les plans de signalisation des différentes intersections afin de maximiser la bande passante des artères urbaines. Il s'agit d'une méthode d'optimisation puissante qui optimise les décalages des feux de circulation adjacents, calcule le cycle temporel, les vitesses de déplacement et l'ordre des phases de virage

à gauche afin d'optimiser les largeurs de bande dans les deux directions. Contrairement à Greenwave, la méthode Maxband réduit le nombre d'arrêts pour les véhicules circulant dans deux directions opposées.

La méthode Maxband exige que toutes les intersections partagent la même longueur de cycle, qui est égale à la valeur maximale de toutes les longueurs de cycle des intersections. Ensuite, un modèle de programmation linéaire en nombres entiers mixtes est formulé pour générer une largeur de bande symétrique de largeur uniforme, sous réserve de contraintes de largeur de bande sur les intersections individuelles qui exigent que pour chaque direction, la largeur de bande soit inférieure au temps vert.

La méthode Maxband a été étendue à un certain nombre de nouvelles méthodes prenant en compte de nouveaux aspects. Gartner et al. [156] ont proposé une version étendue incluant des largeurs de bande asymétriques dans la direction opposée, une durée de cycle et une séquence de phase de virage à gauche variables, ainsi que des vitesses de progression spécifiques aux liaisons. Dans [157], les auteurs présentent une approche multi-poids et multi-bandes qui prend également en compte tous les aspects précédents.

Cette méthode est largement utilisée par les ingénieurs du trafic car elle utilise relativement peu d'entrées et offre une solution globale optimale.

4.4.4 La méthode Max-pression

La méthode de contrôle Max-pression [158] est une stratégie de contrôle distribuée des feux de circulation. Elle vise à réduire le risque de sursaturation en minimisant la pression de phase pour une intersection afin d'équilibrer la longueur de la file d'attente entre les intersections voisines. La méthode Max-pression repose sur un algorithme décentralisé qui ne nécessite que des informations locales à chaque intersection, il maximise également le débit et est prometteur pour le contrôle adaptatif en temps réel. Cette méthode fonctionne selon l'algorithme 3.

Algorithm 3 Max-pression method

```
Begin  
for all time_stamp do  
   $t = t + 1$   
  if  $t \geq t_{min}$  then  
    Calculer la pression  $p_i$  pour chaque phase  $i$  ;  
    Définir la phase suivant comme  $argmax_i\{p_i\}$  ;  
     $t = 0$  ;  
  end if  
end for  
End
```

Cette méthode sélectionne la phase avec la pression maximale, l'active comme phase suivante et conserve la phase sélectionnée pendant une période de temps donnée t_{min} .

Les principales caractéristiques de cet algorithme sont : la stabilité, l'évolutivité et la distribution.

4.4.5 La méthode Actuated control

La méthode Actuated control [152] mesure les "demandes" de signaux verts de la phase actuelle et d'autres phases concurrentes, puis, sur la base de certaines règles, décide de conserver ou de changer la phase actuelle.

La demande de prolongation de la durée du signal vert sur la phase en cours est générée lorsque la durée de la phase en cours n'atteint pas une période minimale, ou lorsqu'un véhicule se trouve sur la voie d'arrivée de la phase en cours et qu'il est proche de l'intersection. Ce véhicule est appelé "approche du signal vert". De plus, la demande de signal vert sur les phases concurrentes est générée lorsque le nombre de véhicules qui attendent dans les phases concurrentes est supérieur à un seuil donné.

4.4.6 La méthode SOTL

Dans la méthode SOTL [159], chaque feu de signalisation, à chaque intersection, conserve un compteur K_i qui est mis à zéro lorsque le feu devient rouge et qui est ensuite incrémenté à chaque pas de temps par les voitures qui s'approchent uniquement du feu rouge ; c'est-à-dire du prochain feu de signalisation, indépendamment de l'état ou de la vitesse des voitures ; peu importe que le véhicule soit en mouvement ou à l'arrêt. Lorsque K_i atteint un seuil θ , le feu vert de la même intersection devient jaune, et au pas de temps suivant, il devient rouge avec $K_i = 0$, tandis que le feu rouge qui a compté devient vert. De cette façon, s'il n'y a plus de voitures qui s'approchent ou qui attendent derrière un feu rouge, celui-ci passera au vert plus rapidement que s'il n'y avait quelques voitures. Ce mécanisme simple permet de réaliser une auto-organisation comme suit :

S'il n'y a qu'une ou quelques véhicules, ils s'arrêteront plus longtemps derrière les feux rouges. Cela donne aux autres véhicules le temps de les rejoindre. Au fur et à mesure que d'autres véhicules rejoignent le groupe, les véhicules attendent moins longtemps derrière les feux rouges. Avec un nombre suffisant de véhicules, les feux rouges passeront au vert avant même que tous les véhicules atteignent l'intersection. Cette méthode n'a pas de phase ou d'horloge interne. Si aucun véhicule ne s'approche d'un feu rouge, le feu peut rester vert. Cette méthode est dite auto-organisée car la performance globale est donnée par les règles locales suivies par chaque feu. Ils sont "inconscients" de l'état des autres intersections et parviennent tout de même à assurer une coordination globale.

La principale différence entre la méthode SOTL et la méthode "actuated control" est la définition de la demande de phase en cours ; bien que les deux méthodes exigent une durée minimale de la phase verte, dans la méthode "actuated control", la demande de phase en cours est générée dès qu'un véhicule s'approche du signal vert, alors que dans la méthode SOTL, la demande n'est générée que si le nombre de véhicules s'approchant du signal vert est supérieur à un seuil, qui n'est pas nécessairement égal à un.

4.4.7 La méthode SCOOT

La méthode SCOOT [160] est un système de contrôle du trafic urbain développé par le Transport Research Laboratory (TRL) en collaboration avec l'industrie britannique des systèmes de circulation. Il s'agit d'un système adaptatif qui réagit automatiquement aux fluctuations du trafic. Son objectif principal est de réduire le nombre moyen de véhicules dans les files d'attente associées aux différentes voies dans une zone donnée. Il s'agit d'un

système de planification flexible qui peut être étendu ou réduit en fonction des conditions de circulation actuelles. Le système mesure en permanence les volumes de trafic sur toutes les voies associées à chaque intersection du réseau et modifie les horaires de signalisation pour minimiser l'indice de performance, qui est une mesure composite du retard, de la longueur des files d'attente, et des arrêts dans le réseau. Dans la méthode SCOOT, chaque cellule peut contrôler jusqu'à 60 intersections et traiter les données d'entrée en utilisant jusqu'à 256 détecteurs de comptage de véhicules dans la rue. Les détecteurs sont généralement placés à 14 m derrière la ligne d'arrêt. Le modèle de trafic et les optimiseurs de SCOOT utilisent ces informations pour programmer les signaux afin d'obtenir le meilleur compromis global pour la coordination le long de tous les liens dans la zone SCOOT.

L'objectif principal du SCOOT est de répondre aux changements de la demande moyenne de trafic observée en effectuant des ajustements fréquents, mais faibles, de la durée des cycles des feux de circulation, de l'allocation du vert et du décalage de chaque intersection contrôlée. Pour chaque zone coordonnée, le système évalue toutes les deux minutes et demie ou les cinq minutes, si le temps de cycle commun en vigueur à toutes les intersections de la zone doit être modifié afin de maintenir un degré de saturation acceptable qui n'atteint pas un chargement d'intersection à 90% pour l'intersection la plus encombrée de la zone.

4.4.8 La méthode SCAT

La méthode SCAT [161] a été mise au point par l'autorité des routes et du trafic "Roads and Traffic Authority (RTA)". Il s'agit d'une stratégie de contrôle automatisé des signaux en temps réel, adaptée au trafic et conçue pour optimiser la circulation à chaque heure. Pendant que des milliers de véhicules traversent les intersections, le système SCAT travaille en arrière-plan. Lorsque l'infrastructure SCAT est installée, des algorithmes intelligents sont configurés pour répondre aux exigences de gestion du trafic du réseau routier afin d'optimiser le flux de circulation. Les contrôleurs SCAT sont en contact avec des serveurs régionaux qui communiquent avec un gestionnaire de centre en optimisant le flux de trafic sur l'ensemble du réseau routier. Ainsi, lorsque les véhicules se déplacent, ils sont détectés par des capteurs situés sous la route à chaque intersection. Le contrôleur de feux de circulation traite ces données et les envoie aux contrôleurs SCAT, qui appliquent des algorithmes et renvoient des instructions de signalisation optimale. Ces instructions déterminent la durée des feux verts ainsi que la séquence des mouvements de circulation. Ce processus se produit pour chaque cycle de feux de circulation à chaque intersection.

La synchronisation des signaux est régie par une logique de contrôle informatique. Elle n'est pas basée sur un modèle, mais dispose d'une bibliothèque de plans dans laquelle elle fait son choix en se basant sur les données de trafic disponibles. Le système offre des possibilités de contrôle et de gestion sans précédent, car il est très flexible, puissant et évolutif. Le système total est divisé en deux parties : un système régional et un système de gestion central. La répartition des ordinateurs régionaux est déterminée par l'économie des communications. Chaque ordinateur régional conserve le contrôle autonome de sa région. Les données d'entrée sont collectées par un système de capteurs de trafic par les biais de caméras de surveillance. Le système est conçu pour s'auto-organiser en fonction des données reçues, afin de réduire la nécessité d'une configuration manuelle. Il prend en charge les quatre modes de fonctionnement suivants :

- Mode normal : fournit un fonctionnement intégré et adapté au trafic.
- Mode de repli : met en oeuvre des plans temporels en cas de défaillance.
- Mode de commande isolée : fonctionnement du véhicule avec une commande indépendante.
- Mode d'indicateur de signal : il clignote en jaune ou en rouge sur toutes les intersections.

La méthode SCAT prend en entrée des plans de signalisation prédéfinis ; à savoir la longueur du cycle, et la division en phases et sélectionne de manière itérative des signaux de circulation, en fonction des paramètres de performance prédéfinis tels que le degré de saturation (DS), qui est détaillé comme suit :

$$DS = \frac{GE}{G} = G - \frac{(H' - H * N)}{G} \quad (4.6)$$

où G est le temps disponible du signal vert en secondes, GE est le temps effectif du vert pendant lequel les véhicules traversent l'intersection. H' est l'écart de temps total détecté, N est le nombre de véhicules détectés, et H est l'unité d'écart de saturation, en secondes, entre les véhicules, qui représente le plus petit intervalle de temps entre des véhicules successifs passant par un point et est généralement fixé par des experts.

4.4.9 La méthode OPAC

OPAC [162] est une stratégie distribuée de contrôle des feux de circulation, dans laquelle le système utilise les données de trafic collectées par des détecteurs situés à la barre d'arrêt à toutes les approches d'une route.

OPAC divise le réseau en sous-réseaux indépendants pour optimiser les performances du réseau routier. Il distingue deux modèles : le premier est pour les réseaux encombrés, et le second pour les réseaux peu encombrés. Cette stratégie met à jour les horaires de signalisation pour réduire les temps d'attente aux carrefours et aux arrêts. Il a été développé pour offrir de meilleures performances que les méthodes hors ligne et pour répondre pleinement à la demande.

4.4.10 Comparaison entre différentes méthodes de contrôle des feux de circulation

Dans cette sous-section, nous présentons une comparaison entre les différentes méthodes de contrôle des feux de circulation en illustrant leur fonctionnement ainsi que les intersections qu'elles supportent. Le tableau 4.1 résume les méthodes de contrôle du trafic citées ci-dessus.

4.5 Contribution : Le protocole CCITL

Dans cette section, nous décrivons notre protocole CCITL en présentant d'abord son architecture et en détaillant, ensuite, son fonctionnement.

TABLE 4.1 – Récapitulatif des méthodes de contrôle du trafic

Méthode	Fonctionnalité	Type d'intersection supporté
Webster [153]	- Calculer la longueur du cycle - Calculer la division de phase	Intersections simples (isolées)
GreenWave [154]	- Maximiser la bande passante des ondes vertes - Optimiser les décalages pour les feux de circulation adjacents	Intersections dans une direction principale
Maxband [155]	- Maximiser la bande passante - Optimiser les décalages pour les feux de circulation adjacents - Calculer le temps des cycles - Calculer la vitesse de progression - Calculer l'ordre des phases de virage à gauche	Intersections de deux directions opposées
Max-pressure [158]	- Réduire le risque de sursaturation - Minimiser la pression de phase	Intersections individuelles
Actuated control [152]	- Réduire le temps d'attente des véhicules	Tous les types d'intersection
SOTL [159]	- Réduire le temps d'attente des véhicules	Tous les types d'intersection
SCOOT [160]	- Possibilité de modifier le trafic en cas d'urgence - Réagir automatiquement aux fluctuations du trafic	Tous les types d'intersection
SCAT [161]	- La synchronisation des signaux est régie par la logique de contrôle - Fournir un bon niveau de précision - Réduire le temps de trajet - Assurer la sécurité de tous les usagers de la route	Tous les types d'intersection
OPAC [162]	- Minimiser le temps d'attente aux intersections	Toutes les intersections d'un même réseau

4.5.1 Architecture du protocole CCITL

L'architecture générale du protocole CCITL est une architecture en couches qui repose sur trois niveaux : La couche véhicule, la couche RSU, et la couche Cloud, comme illustré dans la figure 4.1.

- **La couche véhicule** : il s'agit du niveau le plus bas de l'architecture, il contient les véhicules répartis tout au long de la route ainsi que les feux de circulation aux intersections. Ces véhicules communiquent entre eux et avec les unités routières (RSUs) ainsi qu'avec les infrastructures routières. À ce niveau, les véhicules envoient périodiquement des informations relatives à leurs états actuels aux RSUs qui se trouvent dans leur plage de transmission.

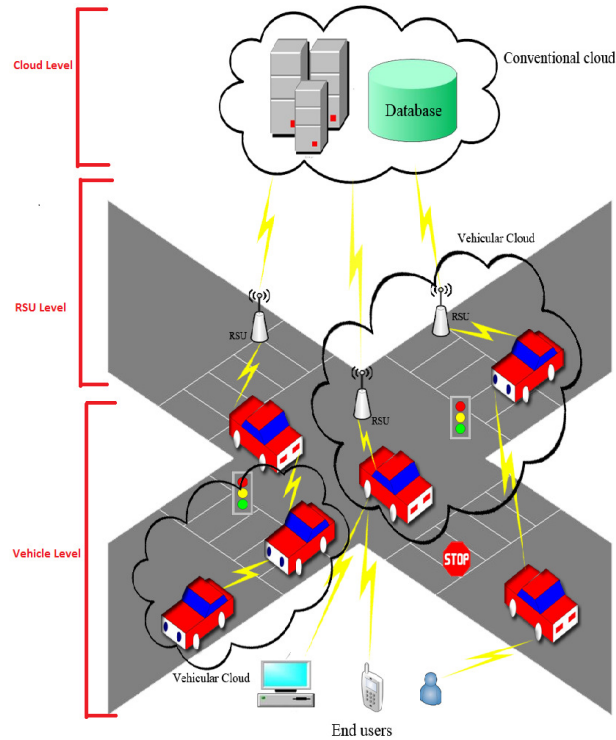


FIGURE 4.1 – Architecture du protocole CCITL

- **La couche RSU :** elle représente l'intermédiaire entre les véhicules et le Cloud. A ce niveau, les RSUs collectent les informations relatives aux véhicules et à l'état de la route et s'engagent, par la suite, à transmettre ces données au Cloud afin que ce dernier puisse les analyser et effectuer les traitements appropriés.
- **La couche Cloud :** il s'agit du niveau le plus élevé de l'architecture dans lequel les différents traitements et calculs sont effectués. A ce niveau, le Cloud collecte les informations relatives aux véhicules et à l'état de la route qui ont été reçues de la part des RSUs, et construit graduellement sa carte du réseau routier pour avoir une vue globale sur le réseau VANET, ce qui lui permet d'effectuer les traitements et les calculs nécessaires atteignant ainsi l'objectif du protocole CCITL qui est d'alléger la circulation routière et diminuer les temps d'attente sur les routes.

4.5.2 Fonctionnement du protocole CCITL

Comme mentionné précédemment, le protocole CCITL proposé est basé sur une architecture à trois niveaux. Il s'exécute donc sur chacune des trois couches suivant un certain nombre d'opérations bien orchestrées afin de garantir le meilleur résultats possible quant à la gestion du trafic routier par le biais de feux de circulation en minimisant les embouteillages et les attentes sur les routes ainsi que les retards qu'ils engendrent. CCITL vise principalement à optimiser le trafic en assurant une conduite fluide tout en arrivant à destination dans un délai optimal. Son mode opératoire ainsi que les détails des traitements qu'il comprend à chaque couche de l'architecture sont expliqués ci-après.

- **Au niveau de la couche véhicule :** À ce niveau, chaque véhicule du réseau

envoie périodiquement des informations le concernant aux RSUs qui se trouvent dans sa même portée de transmission. Ces informations sont envoyées sous forme de messages beacons, et chaque beacon contient les attributs suivants : *VehicleBeacon* (*Seq_Num*, *Source*, *Destination*, *Time_Stamp*, *Time_in*, *Location*, *Lane_id*) où :

- *Seq_Num* désigne le numéro de séquence du beacon,
 - *Source* désigne la source du *VehicleBeacon* qui est l'identifiant propre du véhicule,
 - *Destination* représente la destination du *VehicleBeacon* qui est l'identifiant de la RSU réceptrice,
 - *Time_Stamp* représente l'heure exacte à laquelle le message *VehicleBeacon* est transmis,
 - *Time_In* représente l'heure exacte à laquelle le véhicule a rejoint le segment de la route,
 - *Location* définit les coordonnées de la position où le véhicule est localisé,
 - *Lane_id* désigne le segment de la route où le véhicule se situe à cet instant précis.
- **Au niveau de la couche RSU** : À ce niveau, les RSUs reçoivent dans un premier temps les messages de construction *Msg_const*(*Seq_Num*, *Source*, *Last_hop*, *Hop_count*, *TTL*) de la part du Cloud conventionnel. Les RSUs utilisent ces messages pour construire leur table de routage afin de définir le chemin à prendre pour envoyer un message au Cloud conventionnel. Les attributs contenus dans le message de construction sont :
 - *Seq_Num* représente le numéro de séquence du message *Msg_const*,
 - *Source* désigne la source du message. Elle peut être le Cloud lui-même dans le cas où la RSU réceptrice est un voisin direct du Cloud conventionnel, sinon elle représente une autre RSU voisine qui a transféré le message *Msg_const* à la RSU courante,
 - *Last_hop* désigne l'identifiant de la RSU qui a transféré le message de construction que la RSU actuelle utilise pour emprunter le chemin inverse afin d'atteindre le Cloud,
 - *Hop_count* désigne le nombre de sauts entre le Cloud et la RSU courante,
 - *Time To Life (TTL)* représente la durée de vie du message *Msg_const*.

Dans un second temps, après réception des messages *VehicleBeacon*, les RSU établissent leurs tables de voisinage contenant des informations sur les véhicules voisins et transmettent périodiquement ces informations ainsi que des informations sur l'état de la route au Cloud conventionnel à l'aide de leurs tables de routage.

- **Au niveau de la couche Cloud** : À ce niveau, le Cloud conventionnel construit sa carte du réseau à l'aide des informations collectées relatives aux véhicules et à l'état des routes qui ont été reçues des RSUs. Par conséquent, le Cloud conventionnel aura une vue complète de l'ensemble du réseau VANET. Cette carte du réseau

est caractérisée par les attributs suivants : $map1 \langle map_ID, Location, Time_stamp, Time_in, Veh_ID, Lane_ID \rangle$ où :

- map_ID représente l'identifiant de la carte,
- $Location$ détermine les coordonnées de la position dans laquelle le véhicule se situe,
- $Time_Stamp$ est l'heure à laquelle le véhicule se trouve à une position donnée,
- $Time_In$ est l'heure à laquelle le véhicule a rejoint le segment de la route dont l'identifiant est $Lane_ID$,
- Veh_ID est l'identifiant du véhicule,
- $Lane_ID$ est l'identifiant de la voie dans laquelle se trouve le véhicule.

Ensuite, le Cloud parcourt la carte du réseau et construit progressivement une autre carte où il stocke toutes les informations relatives à la route. Cette deuxième carte est définie comme suit : $map2 \langle map_ID, Lane_ID, Nb_veh, Length, Stay_time, Density \rangle$ où :

- map_ID représente l'identifiant de la carte,
- $Lane_ID$ est l'identifiant de la voie,
- Nb_veh est le nombre de véhicules qui se trouvent sur la voie dont l'identifiant est $Lane_ID$,
- $Length$ est la longueur de la voie dont l'identifiant est $Lane_ID$,
- $Stay_time$ représente le temps pendant lequel le véhicule (v) est resté sur cette voie $Lane_ID$ et est calculé selon l'équation 4.7,
- $Density$ représente la densité des véhicules et est calculée par l'équation 4.8.

$$Stay_time(v) = Time_Stamp - Time_In \quad (4.7)$$

$$Density = \frac{Nb_Veh}{Length} \quad (4.8)$$

Ensuite, le Cloud conventionnel calcule le temps de séjour moyen Avg_Stay_time des véhicules sur chaque voie selon l'équation 4.9.

$$Avg_Stay_time = \frac{\sum_{i=1}^{Nb_veh} Stay_time(v_i)}{Nb_veh} \quad (4.9)$$

Par la suite, le Cloud parcourt cette deuxième carte et extrait la valeur la plus élevée des temps de séjour moyens des véhicules $Max_Avg_Stay_time$ sur les voies du réseau routier qu'il utilisera afin de calculer un seuil de traitement $Threshold$ comme illustré par l'équation 4.10.

$$Threshold = Max_Avg_Stay_time * rd \quad \text{avec } rd \in [0, 1] \quad (4.10)$$

où rd est un nombre aléatoire compris entre 0 et 1.

Le Cloud considère alors que toutes les voies dont le temps de séjour des véhicules est supérieur au seuil calculé précédemment doivent être évacuées. Si deux voies opposées ont des temps d'attente élevés et doivent toutes deux être évacuées, le Cloud conventionnel s'appuie sur la densité pour définir laquelle des deux voies doit être évacuée en premier ; la route ayant la densité de véhicules la plus élevée a la priorité pour l'évacuation. Il répète cette opération périodiquement et calcule à chaque fois le temps de séjour moyen de tous les véhicules dans le réseau VANET. Il transmet ensuite un message d'alerte aux RSUs afin de libérer les voies dont le temps de séjour est le plus long, ou celles dont la densité est la plus élevée dans le cas de voies opposées dont le temps de séjour des véhicules est long.

Les RSUs modifient alors les feux de circulation correspondants qui peuvent influencer sur la congestion de la route concernée en la libérant par un feu vert si le feu est rouge et en fermant l'accès à cette voie, si nécessaire, en mettant au rouge les feux de circulation de ses prédécesseurs afin qu'elles n'évacuent pas leurs véhicules vers la voie congestionnée. Par conséquent, les feux de circulation changent automatiquement et dynamiquement à chaque traitement et la durée des feux rouges et verts varie à chaque itération du traitement par le Cloud de manière dynamique.

L'algorithme 4 représente le traitement Cloud du protocole CCITL.

Algorithm 4 Traitement du Cloud

Begin

Données : $Lanes = \{gneE_1, gneE_2, \dots, gneE_{25}\}$,

Set $V = \{V_1, V_2, \dots, V_n\}$,

Set $Stay_Times = \{stay_time(v_1), stay_time(v_2), \dots, stay_time(v_n)\}$

Résultat : $\min\{\sum_1^n stay_time(v_i)\}$

double $max_time = 0$;

double $threshold$;

$map_1 \quad t_2$;

for ($t = map_2.begin()$; $t <> map_2.end()$; $t ++$) **do**

$t.density = \frac{t.nb_vehicle}{t.length}$;

$t.stay_time = \frac{\sum t_2.stay_time}{t.nb_vehicle}$;

if ($t.stay_time > max_time$) **then**

$max_time = t.stay_time$;

end if

end for

random $rd \in [0, 1]$;

$threshold = max_time * rd$;

for ($t_1 = map_2.begin()$; $t_1 <> map_2.end()$; $t_1 ++$) **do**

if ($t_1.lane_stay_time \geq threshold$) **then**

evacuate $t_1.lane_id$;

end if

end for

End

4.5.3 Environnement de travail

Pour mettre en oeuvre le protocole CCITL, nous avons couplé l'environnement de simulation de réseau OMNeT++ [163] avec l'outil de microsimulation du trafic routier SUMO [164]. Ainsi, le scénario a été mis en œuvre à l'aide d'OMNeT++, et la mobilité et le flux de trafic ont été générés à l'aide de SUMO [164]. Le réseau VANET a été réalisé à l'aide du framework de simulation VEINS [84]. La pile de protocoles Internet (Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol version 4 (IPv4), IPv6, etc...) et les protocoles des couches de liaison filaire et sans fil (Ethernet, IEEE 802.11, etc...) sont fournis par le framework INET [165]. Cet environnement de développement est considéré comme complet car il fournit une simulation très représentative de la réalité, prenant en compte la mobilité des véhicules en combinaison avec une simulation précise du réseau.

Nous avons développé notre propre implémentation du Cloud véhiculaire. Le tableau 4.2 résume tous les paramètres de simulation.

TABLE 4.2 – Paramètres de simulation pour l'évaluation du protocole CCITL

Paramètre	Valeur
Framework de simulation	OMNeT++, SUMO, VEINS
Surface de la zone de simulation	30-80 km^2
Nombre de véhicules	100
Nombre de RSUs	10
Nombre de TLSs	5
Vitesse moyenne des véhicules	80 km/h
Protocole de transport	UDP
Portée de transmission	250m
Couche réseau	802.11p
Durée de simulation	300 <i>sec</i>

4.5.4 Evaluation des performances et analyse des résultats

Afin d'évaluer les performances de notre protocole, nous avons réalisé un ensemble de simulations et de configurations expérimentales via le simulateur OMNeT++ et SUMO. Nous avons implémenté notre propre réseau VANET comprenant 24 segments, 4 jonctions, 5 intersections ; dont une impliquant 8 voies et quatre autres impliquant 6 voies, 5 "Traffic Light System (TLS)" à chaque intersection, des RSUs et des véhicules tout au long de la route comme le montre la figure Fig. 4.2. Nous avons également développé notre propre Cloud computing conventionnel sur OMNeT++.

Étant donné qu'aucun des protocoles et algorithmes récents cités dans la section 4.3 n'implique le Cloud computing, nous avons utilisé comme traitement initial de la gestion du trafic routier avec des feux de circulation les "programmes TLS générés automatiquement" de SUMO [166] (AGTLS-P). Ensuite, nous avons implémenté le protocole CCITL pour comparer et évaluer les différences entre les deux protocoles.



FIGURE 4.2 – Carte routière du réseau véhiculaire implémenté

Dans le travail présenté dans [167], nous devons d'abord détecter si le véhicule est dans un état de congestion ou non afin de décider du contrôle à effectuer : centralisé ou décentralisé. Le contrôle centralisé basé sur les RSUs permet le contrôle du TLS aux intersections dans le cas d'une situation de congestion, tandis que le contrôle décentralisé est utilisé dans le cas d'une situation de non congestion où chaque véhicule décide par lui-même du chemin à emprunter. Par conséquent, contrairement aux travaux présentés dans [167], le protocole CCITL fonctionne uniquement de manière centralisée où le Cloud conventionnel est le pion de la gestion du trafic basée sur le contrôle du TLS ayant à tout moment une vue globale du réseau routier et donc l'information exacte sur l'état de la route.

L'analyse des performances a révélé qu'après le traitement Cloud conventionnel pour trouver les voies routières les plus encombrées avec les temps d'attente des véhicules les plus longs et après exécution des consignes du Cloud par les RSUs pour évacuer les routes encombrées avec des temps d'attente élevés, les résultats obtenus étaient prometteurs.

a) Temps d'attente moyen des véhicules dans le réseau routier

Dans cette sous-section, nous présentons l'évolution du temps de séjour des véhicules dans les différents segments de route avec l'application du protocole CCITL et sans CCITL ; où le trafic est géré par des TLSs qui sont contrôlés par AGTLS-P [166], afin d'illustrer l'avantage de notre contribution.

Les résultats présentés dans les figures 4.3 et 4.4 montrent la réduction du temps de séjour des véhicules sur les routes de l'ensemble du réseau. Nous constatons que le protocole CCITL proposé a considérablement réduit le temps de séjour des véhicules dans les différents segments. En effet, avant l'application du protocole CCITL, c'est-à-dire lorsque le trafic était contrôlé par les feux de signalisation commandés par le protocole AGTLS-P, le temps de séjour moyen des véhicules sur les segments de route était de 5,27

secondes. Comme le montre la figure 4.3, ce temps a dégringolé à 2,75 secondes lorsque le protocole CCITL a été appliqué, c'est-à-dire lorsque le trafic est contrôlé par les feux de circulation dont la stratégie de planification est régie par le protocole CCITL.

Nous remarquons également, en exécutant les deux protocoles AGTLS-P et CCITL séparément, qu'à différents moments de la simulation, le temps de séjour moyen des véhicules sur les segments de route est toujours plus court avec CCITL qu'avec AGTLS-P. En effet, la figure 4.4 illustre bien ce phénomène ; nous pouvons voir qu'après 30 secondes de simulation, le temps de séjour moyen des véhicules avec CCITL est de 4,41 secondes contre 5,24 secondes avec AGTLS-P. De même, après 85 et 120 secondes, les temps d'attente des véhicules avec CCITL sont respectivement de 3,25 secondes et 2,75 secondes, contre 6,62 secondes et 5,43 secondes avec AGTLS-P.

En comparant les deux protocoles CCITL et AGTLS-P, nous concluons que CCITL a considérablement réduit les temps d'attente des véhicules ; par conséquent, les temps d'attente sur les routes ont été réduits de 52% avec CCITL.

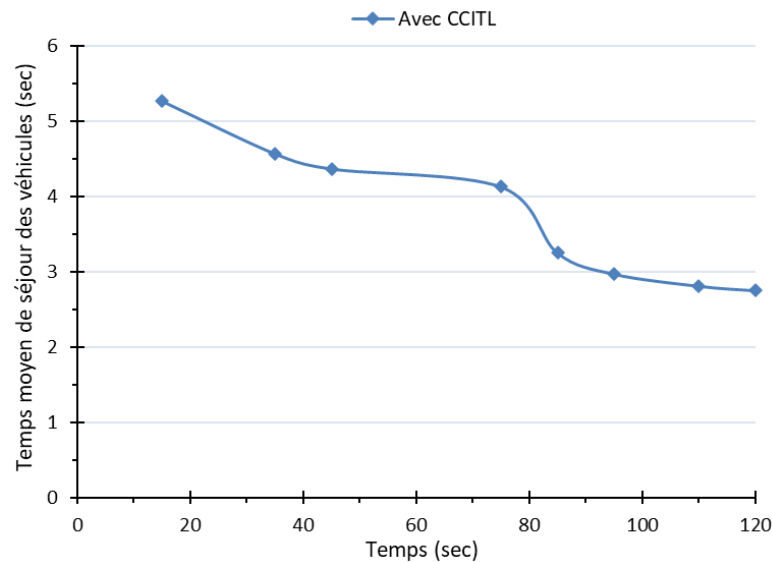


FIGURE 4.3 – Evolution du temps de séjour des véhicules après application du protocole CCITL

b) Temps d'attente moyen dans les différents segments de la route

L'impact du protocole CCITL sur le temps de séjour des véhicules dans les différents segments est sans équivoque. Les figures 4.5a, 4.5b, 4.5c, 4.5d, et 4.5e illustrent en détail l'impact de CCITL sur le temps de séjour des véhicules dans les différents segments.

Les cinq segments les plus encombrés du réseau, *gneE2*, *gneE14*, *gneE18*, *gneE19* et *gneE24*, ont vu le temps d'attente de leurs véhicules considérablement réduit. En effet, le temps d'attente moyen des véhicules dans le segment *gneE2* est passé de 9 secondes à 2,16 secondes après 20 secondes d'exécution de CCITL pour atteindre 1,39 secondes deux minutes après l'exécution de CCITL alors que sans CCITL ; avec AGTLS-P, le temps d'attente moyen le plus court des véhicules est de 1,71 seconde. Dans le segment *gneE14*, le temps d'attente moyen des véhicules est passé de 13 secondes à 7 secondes

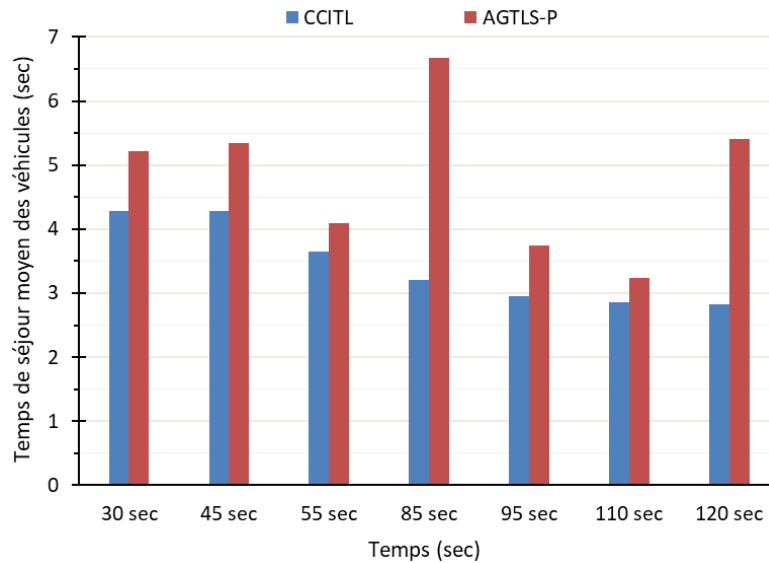


FIGURE 4.4 – Temps de séjour des véhicules dans le réseau avec CCITL et AGTLS-P

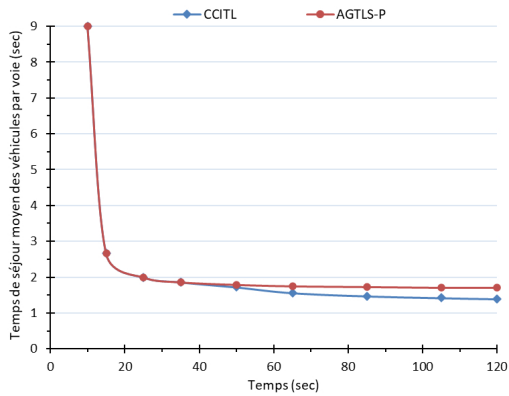
après 30 secondes d'exécution du CCITL, pour atteindre 0,1 seconde 45 secondes après l'exécution de CCITL, alors que sans CCITL ; avec AGTLS-P, sa valeur minimale est de 6,33 secondes. Le temps d'attente moyen des véhicules dans le segment *gneE18* a été réduit de 4,6 secondes à 3,30 secondes 45 secondes après l'exécution de CCITL pour atteindre 2,74 secondes de temps d'attente 2 minutes après l'exécution de CCITL alors que sa meilleure valeur sans CCITL ; avec AGTLS-P, est de 5,8 secondes. Dans *gneE19*, le temps d'attente des véhicules passe de 5 secondes à 2,72 secondes 45 secondes après l'exécution du CCITL pour atteindre 2,56 secondes à 2 minutes, alors que sans CCITL ; avec AGTLS-P, il est de 5,25 secondes à 2 minutes. Dans *gneE24*, le temps d'attente passe de 8 secondes à 1,92 secondes 45 secondes après l'exécution de CCITL pour atteindre 1,72 secondes 2 minutes après l'exécution de CCITL alors qu'il est d'un minimum de 2,68 avec AGTLS-P ; sans CCITL.

c) Nombre de véhicules dans les différents segments de la route

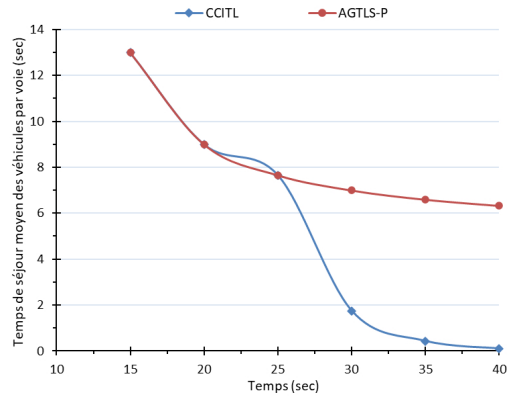
Dans cette partie, nous illustrons l'efficacité du protocole CCITL en se basant sur le critère du nombre de véhicules qui ont transité à travers les différents segments du réseau routier. Pour ce faire nous analysons les différences de flux de véhicules avec et sans protocole CCITL (c'est-à-dire uniquement avec une gestion du trafic basée sur AGTLS-P avant l'exécution de CCITL).

Comme le montrent les figures 4.6, 4.7a, et 4.7b, le nombre de véhicules ayant traversé les différents segments du réseau routier a augmenté après l'application du protocole CCITL. En effet, plus la route est fluide, plus elle est apte à accueillir de nouveaux véhicules.

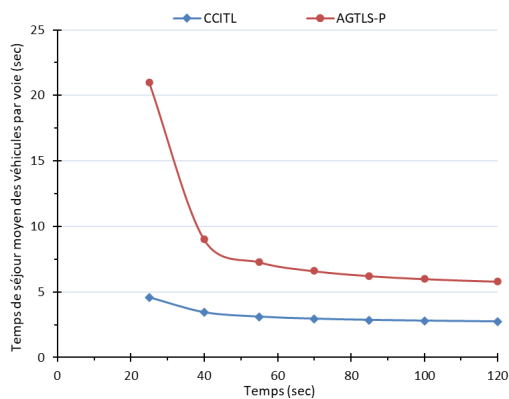
Dans le segment *gneE18* (figure 4.7a), 40 secondes après l'exécution du protocole CCITL, le nombre de véhicules ayant transité par ce segment est de 12 véhicules alors qu'il n'est que de 4 véhicules avec la gestion basée sur AGTLS-P. Ce nombre grimpe à 33 véhicules après 90 secondes avec CCITL contre 14 véhicules avec AGTLS-P pour atteindre 43 véhicules après deux minutes d'exécution du protocole CCITL contre seulement 20



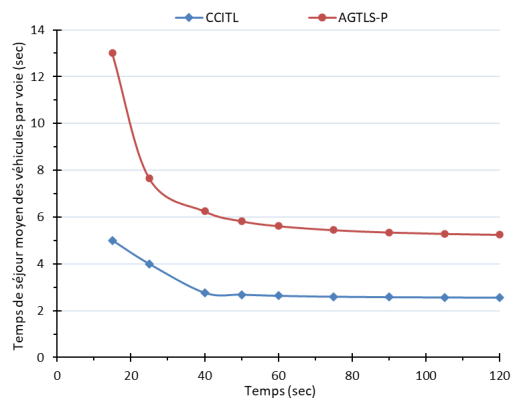
(a) Segment E2



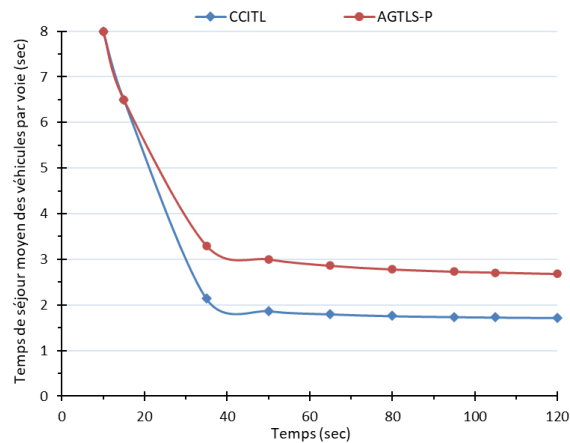
(b) Segment E14



(c) Segment E18



(d) Segment E19



(e) Segment E24

FIGURE 4.5 – Temps de séjour des véhicules dans différents segments avec CCITL et AGTLS-P

véhicules avec AGTLS-P. De même, pour le segment *gneE19* (figure 4.7b), 40 secondes après l'exécution du protocole CCITL, le nombre de véhicules qui traversent est de 9 véhicules alors qu'il n'est que de 4 véhicules sans CCITL ; avec AGTLS-P. Ce nombre grimpe par la suite à 25 véhicules après 90 secondes avec CCITL contre 15 véhicules

avec AGTLS-P pour atteindre 41 véhicules après deux minutes d'exécution du protocole CCITL contre seulement 20 véhicules avec AGTLS-P, c'est-à-dire sans protocole CCITL.

Le nombre total de véhicules ayant traversé les segments de route du réseau en appliquant le protocole CCITL est de 498 contre 468 sans le protocole CCITL à 80 secondes. Ce nombre augmente encore jusqu'à 851 véhicules avec CCITL contre 786 véhicules sans le protocole CCITL après deux minutes.

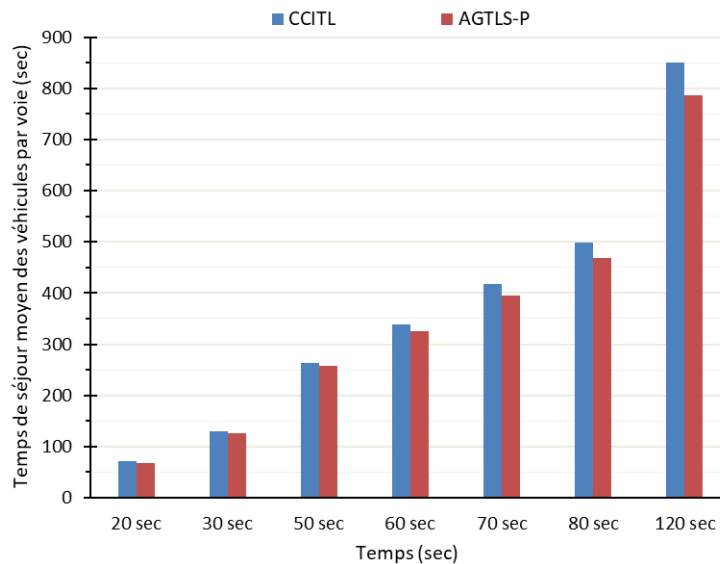
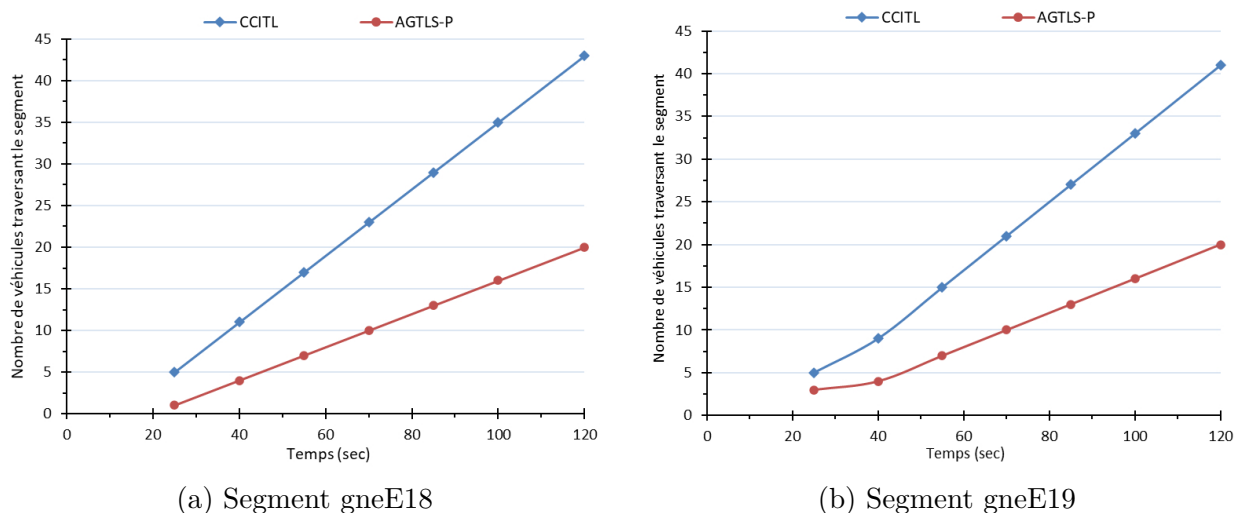


FIGURE 4.6 – Nombre de véhicules traversant différents segments avec CCITL et AGTLS-P



(a) Segment gneE18

(b) Segment gneE19

FIGURE 4.7 – Nombre de véhicules ayant transité à travers les segments gneE18 et gneE19 CCITL et AGTLS-P

d) Récapitulatif des résultats de la simulation

Dans cette partie, nous présentons un récapitulatif des résultats de la simulation sous la forme d'un tableau qui regroupe les résultats obtenus après application du protocole CCITL pour gérer le trafic routier et les compare à ceux obtenus lorsque la gestion du trafic routier est basée sur le protocole AGTLS-P. Le tableau 4.3 résume les résultats des simulations.

TABLE 4.3 – Récapitulatif des résultats de simulation

	CCITL	AGTLS-P
Temps d'attente moyen des véhicules	4.41 sec à 30sec de simulation	5.27 sec à 30sec de simulation
	3.25 sec à 85sec de simulation	6.62 sec à 85sec de simulation
	2.75 sec à 2min de simulation	5.43 sec à 2min de simulation
Meilleure valeur du temps de séjour des véhicules par segment	1.39 sec dans gneE2 0.1 sec dans gneE14 2.74 sec dans gneE18 2.72 sec dans gneE19 1.72 sec dans gneE24	1.71 sec dans gneE2 6.33 sec dans gneE14 5.8 sec dans gneE18 5.25 sec dans gneE19 2.68 sec dans gneE24
Nombre de véhicules ayant transité par le réseau	851	786
Nombre de véhicules ayant transité par segment	43 à travers gneE18 41 à travers gneE19	20 à travers gneE18 20 à travers gneE19

4.6 Conclusion

Dans ce chapitre, nous avons proposé un protocole basé sur le Cloud computing pour gérer le trafic routier dans les VANETs, appelé CCITL. Notre protocole utilise le Cloud computing pour avoir une vue complète du réseau VANET et calculer respectivement le temps de séjour des véhicules dans chaque segment de route ainsi que leur densité dans chaque voie. Le protocole CCITL est basé sur une architecture en couches définie sur trois niveaux : le niveau des véhicules, le niveau des RSUs et le niveau du Cloud. CCITL implique le Cloud conventionnel comme fournisseur de services dont les serveurs sont dédiés aux calculs de la densité et du temps d'attente des véhicules dans chaque voie de la route. En outre, il calcule le temps de séjour moyen des véhicules dans toutes les voies du réseau routier afin de décongestionner de manière optimale les voies dans lesquelles les véhicules ont des temps de séjour longs. Ainsi, afin de rendre le trafic plus fluide, le Cloud envoie des instructions aux RSUs pour qu'elles contrôlent les feux de circulation en allumant les feux verts et rouges selon les instructions du Cloud.

L'implémentation du protocole CCITL a été réalisée par simulations via OMNeT++ avec SUMO. Les résultats obtenus nous ont permis de constater que notre protocole constitue une solution fiable puisque le trafic est devenu plus fluide et le temps d'attente des véhicules dans plusieurs segments est réduit par rapport au protocole AGTLS-P.

Conclusion Générale

Conclusion Générale

Les réseaux véhiculaires dérivent des réseaux ad-hoc mobiles (MANETs) où les véhicules stationnés et en mouvement représentent les nœuds des réseaux. De ce fait, ce type de réseaux est principalement caractérisé par la haute mobilité de ses nœuds et le changement très fréquent de sa topologie. Par conséquent, maintenir une communication stable et sans interruption entre les divers acteurs du réseau représente un défi de taille dans les VANETs. Les recherches récentes se sont plutôt focalisées sur l'aspect technologique du véhicule en rendant celui-ci de plus en plus sophistiqué et doté de plusieurs capteurs et technologies. Cependant, afin d'exploiter au mieux les capacités de ces véhicules en intégrant tous types d'applications pour satisfaire les besoins des conducteurs, les véhicules deviennent de plus en plus gourmands en termes de capacité de calcul et d'espace de stockage. Faire appel au Cloud computing pour satisfaire ce genre de contraintes émerge alors comme une solution triviale et efficace.

Motivés par ces considérations, nous avons tenté de trouver des solutions à deux genres de problématiques dans le cadre de cette thèse. La première concerne la communication fiable dans les VANETs en minimisant le nombre de RSUs à déployer, et la deuxième concerne la gestion de la circulation routière afin d'éviter les congestions de route et les retards pour arriver à destination.

Dans la première contribution, notre travail a été de trouver une solution qui puisse garantir une bonne performance de routage et un taux de couverture satisfaisant tout en minimisant le nombre de RSUs déployées au niveau des intersections de la route. Pour ce faire, nous avons appliqué les méthodes d'optimisation des problèmes polynomiaux sur le nombre total des RSUs à déployer afin de l'optimiser. Nous avons utilisé une méta-heuristique basée sur les algorithmes génétiques dans un premier temps, puis une autre méta-heuristique basée sur le recuit simulé dans un second temps. Les résultats obtenus étaient prometteurs puisque nous avons grandement diminué le nombre de RSUs au niveau des intersections tout en gardant de bonnes performances de routage et un bon taux de couverture dans le réseau. De plus, cette contribution permet aussi d'économiser les coûts qu'engendre le déploiement des RSUs dans un réseau VANET, puisque le nombre de RSUs à déployer diminue ce qui engendre automatiquement une baisse du coût globale de la mise oeuvre du réseau.

Dans la seconde contribution, nous avons tenté de trouver une solution pour le problème de congestion du réseau routier qui engendre une perte de temps considérable pour les conducteurs entraînant des retards pour arriver à leurs destinations. Pour ce faire, nous avons développé un protocole, appelé "CCITL". CCITL implique le Cloud computing

conventionnel afin de gérer le trafic routier en utilisant les feux de circulation intelligent. Il est également capable de gérer efficacement le trafic en se basant sur les traitements faits au niveau du Cloud afin de libérer ou non une route en changeant dynamiquement les feux de circulation et la durée de chaque phase selon les calculs périodiques faits au niveau du Cloud.

Les solutions proposées ont été implémentées en utilisant le simulateur OMNet++ en combinaison avec le simulateur de mobilité urbaine SUMO. Les résultats de simulation montrent les avantages de nos solutions en terme de performance de routage, de taux de couverture et de taux de livraison des paquets pour la première solution, et en terme d'évitement de congestion, de délais pour arriver à destination et de nombre de véhicules pouvant transiter en fluidité sur la route pour la seconde solution.

Comme perspectives à ce travail, nous aspirons à aller encore plus loin dans la gestion intelligente du trafic routier en proposant un protocole de gestion du trafic entièrement distribué qui se base sur le Fog computing. Nous nous intéresserons également à la distribution des ressources du Cloud computing aux utilisateurs de telle sorte à optimiser les ressources fournies afin de satisfaire un maximum de demandes tout en évitant le gaspillage des ressources.

Références Bibliographiques

Bibliographie

- [1] R. I. Meneguetto, “A vehicular cloud-based framework for the intelligent transport management of big cities,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, pp. Article ID 8198597, 9 pages, 2016.
- [2] J. Santa, P. J. Fernández, and F. Pereñíguez, “Deployment of vehicular networks in highways using 802.11 p and IPv6 technologies,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 24, no. 1-2, pp. 33–48, 2017.
- [3] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular Fog Computing : A viewpoint of vehicles as the infrastructures,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [4] J. S. Weber, M. Neves, and T. Ferreto, “Vanet simulators : an updated review,” *Journal of the Brazilian Computer Society*, vol. 27, no. 1, pp. 1–31, 2021.
- [5] StackTrace, “Qu’est-ce que le cloud computing ?” <https://stacktraceback.com/le-cloud-computing/>, December 2020.
- [6] F. Arron, “7 différents types de structures de calcul en nuage que vous devriez savoir.” <https://www.uniprint.net/fr/7-types-cloud-computing-structures/>, March 2017.
- [7] H. F. Atlam, R. J. Walters, and G. B. Wills, “Fog computing and the internet of things : A review,” *big data and cognitive computing*, vol. 2, no. 2, p. 10, 2018.
- [8] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, “A survey on vehicular cloud computing,” *Journal of Network and Computer applications*, vol. 40, pp. 325–344, 2014.
- [9] C. Huang, R. Lu, and K.-K. R. Choo, “Vehicular fog computing : architecture, use case, and security and forensic challenges,” *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, 2017.
- [10] M. Taneja and A. Davy, “Resource aware placement of data analytics platform in fog computing,” *Procedia Computer Science*, vol. 97, pp. 153–156, 2016.

- [11] L. Figueiredo, I. Jesus, J. T. Machado, J. R. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 intelligent transportation systems. Proceedings (Cat. No. 01TH8585)*, pp. 1206–1211, IEEE, 2001.
- [12] M. Srinivas and L. M. Patnaik, "Genetic algorithms : A survey," *computer*, vol. 27, no. 6, pp. 17–26, 1994.
- [13] D. Bertsimas, J. Tsitsiklis, *et al.*, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.
- [14] N. Gaouar and M. Lehsaini, "Toward vehicular cloud/fog communication : A survey on data dissemination in vehicular ad hoc networks using vehicular cloud/fog computing," *International Journal of Communication Systems*, vol. 34, no. 13, p. e4906, 2021.
- [15] W. Xiang, J. Gozalvez, Z. Niu, O. Altintas, and E. Ekici, "Wireless access in vehicular environments," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. Article ID 576217, 2 pages, 2009.
- [16] T. M. Kurihara *et al.*, "Trial-use standard for wireless access in vehicular environments (WAVE)-resource manager," *IEEE Std 1609.2-2006*, pp. 1–105, 2006.
- [17] S. Olariu and M. C. Weigle, *Vehicular networks : from theory to practice*. Crc Press, Chapman & Hall, 2009.
- [18] A. Quyoom, A. A. Mir, and A. Sarwar, "Security attacks and challenges of vanets : a literature survey," *Journal of Multimedia Information System*, vol. 7, no. 1, pp. 45–54, 2020.
- [19] C. Sommer and F. Dressler, *Vehicular networking*. Cambridge University Press, 2014.
- [20] B. Sharma, M. S. P. Sharma, and R. S. Tomar, "A survey : Issues and challenges of vehicular ad hoc networks (VANETs)," in *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*, (Amity University Rajasthan, Jaipur, India), pp. 2491–2503, Elsevier SSRN, 2019.
- [21] J. Mcnew, "Ieee standard," *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages, Approved*, vol. 8, 2006.
- [22] I. T. S. Committee *et al.*, "IEEE trial-use standard for wireless access in vehicular environments (WAVE)-networking services," *IEEE Std*, vol. 1609, pp. 1603–2007, 2007.
- [23] I. T. S. Committee *et al.*, "Ieee trial-use standard for wireless access in vehicular environments (wave)-multichannel operation," *IEEE, Standard*, pp. 1609–4, 2011.

-
- [24] A. Abunei, C.-R. Comşa, and I. Bogdan, “Implementation of ETSI ITS-G5 based inter-vehicle communication embedded system,” in *Proceedings of International Symposium on Signals, Circuits and Systems (ISSCS)*, (Iasi, Romania), pp. 1–4, IEEE, 2017.
- [25] E. G. Strom, “On Medium Access and Physical Layer Standards for Cooperative Intelligent Transport Systems in Europe,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1183–1188, 2011.
- [26] M. Nookala and B. Estochen, “Minnesota, USA intelligent vehicle initiative,” in *Proceedings of Intelligent Vehicle Symposium*, vol. 2, (Versailles, France), pp. 533–536 vol.2, IEEE, 2002.
- [27] S. K. Bhoi and P. M. Khilar, “Vehicular communication : a survey,” *IET networks*, vol. 3, no. 3, pp. 204–217, 2014.
- [28] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, “Vehicular ad hoc networks (VANETS) : status, results, and challenges,” *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.
- [29] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, “Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications,” *IEEE Transactions on intelligent transportation systems*, vol. 3, no. 3, pp. 155–161, 2002.
- [30] S. Tsugawa, S. Kato, and K. Aoki, “An automated truck platoon for energy saving,” in *Proceedings of International Conference on Intelligent Robots and Systems*, (San Francisco, CA, USA), pp. 4109–4114, IEEE, 2011.
- [31] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [32] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing : A survey,” *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [33] A. V. Dastjerdi and R. Buyya, “Fog computing : Helping the internet of things realize its potential,” *Computer*, vol. 49, no. 8, pp. 112–116, 2016.
- [34] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the First edition of the MCC workshop on Mobile cloud computing*, (Helsinki, Finland), pp. 13–16, ACM, 2012.
- [35] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. S. Goren, and C. Mahmoudi, “Fog computing conceptual model,” tech. rep., National Institute of Standards and Technology, U.S. Department of Commerce, 2018.
- [36] M. Mukherjee, L. Shu, and D. Wang, “Survey of fog computing : Fundamental, network applications, and research challenges,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

- [37] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of International Conference on Future Internet of Things and Cloud*, (Barcelona, Spain), pp. 464–470, IEEE, 2014.
- [38] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proceedings of the 29th International Conference on Advanced Information Networking and Applications*, (Gwangju, South Korea), pp. 687–694, IEEE, 2015.
- [39] M. Gerla, "Vehicular cloud computing," in *Proceedings of the 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, (Ayia Napa, Cyprus), pp. 152–155, IEEE, 2012.
- [40] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones : A low-cost platform for driver monitoring," *IEEE Intelligent transportation systems magazine*, vol. 7, no. 1, pp. 91–102, 2015.
- [41] A. Boukerche and E. Robson, "Vehicular cloud computing : Architectures, applications, and mobility," *Computer networks*, vol. 135, pp. 171–189, 2018.
- [42] M. Aazam and E.-N. Huh, "E-HAMC : Leveraging Fog computing for emergency alert service," in *Proceedings of International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, (St. Louis, MO, USA), pp. 518–523, IEEE, 2015.
- [43] P. Bellavista, E. Magistretti, U. Lee, and M. Gerla, "Standard integration of sensing and opportunistic diffusion for urban monitoring in vehicular sensor networks : The mobeyes architecture," in *Proceedings of International Symposium on Industrial Electronics*, (Vigo, Spain), pp. 2582–2588, IEEE, 2007.
- [44] Y. Agarwal, K. Jain, and O. Karabasoglu, "Smart vehicle monitoring and assistance using cloud computing in vehicular ad hoc networks," *International Journal of Transportation Science and Technology*, vol. 7, no. 1, pp. 60–73, 2018.
- [45] M. Saini, K. M. Alam, H. Guo, A. Alelaiwi, and A. El Saddik, "Incloud : a cloud-based middleware for vehicular infotainment systems," *Multimedia Tools and Applications*, vol. 76, no. 9, pp. 11621–11649, 2017.
- [46] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu, "A survey and challenges in routing and data dissemination in vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 787–795, 2011.
- [47] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," *Vehicular Communications*, vol. 1, no. 4, pp. 214–225, 2014.
- [48] B. T. Sharef, R. A. Alsaqour, and M. Ismail, "Vehicular communication ad hoc routing protocols : A survey," *Journal of Network and Computer Applications*, vol. 40, pp. 363–396, 2014.

-
- [49] T. Nadeem, P. Shankar, and L. Iftode, “A comparative study of data dissemination models for VANETs,” *Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems : Networking & Services*, (San Jose, CA, USA), pp. 1–10, IEEE, 2006.
- [50] H. Shahwani, B. A. Mugabarigira, J. Shin, and J. P. Jeong, “An effective data processing and data dissemination in vehicular networks,” *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, (Langkawi, Malaysia), pp. 1–4, ACM, 2018.
- [51] L. Tang, Z. Duan, and Y. Zhu, “Data dissemination in service discovery for vehicular ad hoc networks : a survey,” *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1189–1200, 2018.
- [52] A. Paranjothi, “Performance Analysis of Message Dissemination Techniques in VANET using Fog Computing,” tech. rep., <https://arxiv.org/abs/2003.04354>, University of Oklahoma, USA, 2020.
- [53] I. Ahmad, R. M. Noor, I. Ali, M. Imran, and A. Vasilakos, “Characterizing the role of vehicular cloud computing in road traffic management,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 5, pp. 1–14, 2017.
- [54] R. C. Poonia and L. Raja, “On-Demand Routing Protocols for Vehicular Cloud Computing,” in *Vehicular cloud computing for traffic management and systems*, pp. 151–177, IGI Global (Vehicular cloud computing for traffic management and systems), 2018.
- [55] P. G. Naranjo, Z. Pooranian, S. Shamshirband, J. H. Abawajy, and M. Conti, “Fog over virtualized IoT : New opportunity for context-aware networked applications and a Case Study,” *Applied Sciences*, vol. 7, no. 12, p. 1325, 2017.
- [56] F. Shahid, H. Ashraf, A. Ghani, S. A. K. Ghayyur, S. Shamshirband, and E. Salwana, “PSDS–Proficient Security Over Distributed Storage : A Method for Data Transmission in Cloud,” *IEEE Access*, vol. 8, pp. 118285–118298, 2020.
- [57] F. Benjamin Zhan and C. Noon, “Shortest Path Algorithms : An Evaluation Using Real Road Networks,” *Transportation Science*, vol. 32, pp. 65–73, 02 1998.
- [58] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” *Proceedings of International Conference on Communications architectures, protocols and applications (SIGCOMM ’94)*, (London, United Kingdom), pp. 234–244, ACM, October 1994.
- [59] T.-W. Chen and M. Gerla, “Global State Routing : A new routing scheme for ad-hoc wireless networks,” *Proceedings of IEEE International Conference on Communications (ICC’98)*, (Atlanta, GA, USA), pp. 171–175, IEEE, 1998.
- [60] G. Pei, M. Gerla, and T.-W. Chen, “Fisheye State Routing : A routing scheme for ad hoc wireless networks,” *Proceedings of International Conference on Communications (ICC’2000)*, (New Orleans, LA, USA), pp. 70–74, IEEE, 2000.

- [61] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, “The Optimized Link State Routing Protocol Evaluation through Experiments and Simulation,” *Proceedings of Symposium on Wireless Personal Mobile Communications*, pp. 1–6, IEEE, 2001.
- [62] S. Murthy and J. J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” *Mobile Networks and Applications*, vol. 1, no. 2, pp. 183–197, 1996.
- [63] R. Ogier, F. Templin, and M. Lewis, “Topology dissemination based on reverse-path forwarding (TBRPF),” tech. rep., Network Working Group, IETF RFC 3684, 2004.
- [64] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” *Proceedings of INFOCOM’97*, (Kobe, Japan), pp. 1405–1413, IEEE, 1997.
- [65] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” *Proceedings of the Second Workshop on Mobile Computing Systems and Applications (WMCSA’99)*, (New Orleans, LA, USA), pp. 90–100, IEEE, 1999.
- [66] V. Namboodiri, M. Agarwal, and L. Gao, “A study on the feasibility of mobile gateways for vehicular ad-hoc networks,” *Proceedings of the First ACM International Workshop on Vehicular Ad hoc Networks*, (Philadelphia, PA, USA), pp. 66–75, ACM, 2004.
- [67] D. A. Johnson, David B. and Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, pp. 153–181. Boston, MA : Springer US, 1996.
- [68] V. Naumov, R. Baumann, and T. Gross, “An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces,” *Proceedings of the 7th International Symposium on Mobile Ad hoc Networking and Computing*, (Florence, Italy), pp. 108–119, ACM, 2006.
- [69] N. Nikaein, C. Bonnet, and N. Nikaein, “Harp-hybrid ad hoc routing protocol,” *Proceedings of International Symposium on Telecommunications (IST)*, (Tehran, Iran), pp. 56–67, 2001.
- [70] Z. J. Haas, “A new routing protocol for the reconfigurable wireless networks,” *Proceedings of the 6th International Conference on Universal Personal Communications (ICUPC’97)*, (San Diego, CA, USA), pp. 562–566, IEEE, 1997.
- [71] S. Dhankhar and S. Agrawal, “VANETs : A survey on routing protocols and issues,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 6, pp. 13427–13435, 2014.
- [72] L. Gao, S. Yu, T. H. Luan, and W. Zhou, *Delay tolerant networks*. Switzerland : Springer Cham, 2015.
- [73] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, “FogRoute : DTN-based data dissemination model in fog computing,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, 2016.

-
- [74] X. Masip-Bruin, E. Marin-Tordera, G. Tashakor, A. Jukan, and G. Ren, “Foggy clouds and cloudy fogs : a real need for coordinated management of fog-to-cloud computing systems,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, 2016.
- [75] S. Bitam, A. Mellouk, and S. Zeadally, “VANET-cloud : a generic cloud computing model for vehicular Ad Hoc networks,” *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.
- [76] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, “Toward cloud-based vehicular networks with efficient resource management,” *IEEE Network*, vol. 27, no. 5, pp. 48–55, 2013.
- [77] M. Pasha, M. Farooq, K. Khan, and M.-u.-H. Atif, “Vehicular Cloud Computing : leading towards tomorrow’s Internet of Vehicles,” *Journal of Wireless Sensor Networks*, vol. 2, pp. 1–10, 07 2015.
- [78] S. Bitam and A. Mellouk, “Cloud Computing-Based Message Dissemination Protocol for Vehicular Ad Hoc Networks,” Proceedings International Conference on Wired/Wireless Internet Communication, (Malaga, Spain), pp. 32–45, Springer, 2015.
- [79] S. K. Garg, C. S. Yeo, and R. Buyya, “Green cloud framework for improving carbon efficiency of clouds,” Proceedings of European Conference on Parallel Processing, (Bordeaux, France), pp. 491–502, Springer, 2011.
- [80] J. Bernsen and D. Manivannan, “RIVER : A reliable inter-vehicular routing protocol for vehicular ad hoc networks,” *Computer Networks*, vol. 56, no. 17, pp. 3795–3807, 2012.
- [81] R. Kim, H. Lim, and B. Krishnamachari, “Prefetching-based data dissemination in vehicular cloud systems,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 292–306, 2015.
- [82] L. Nkenyereye, Y. Park, and K.-H. Rhee, “Secure vehicle traffic data dissemination and analysis protocol in vehicular cloud computing,” *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1024–1044, 2018.
- [83] F. Dressler, P. Handle, and C. Sommer, “Towards a vehicular cloud-using parked vehicles as a temporary network and storage infrastructure,” Proceedings of International Workshop on Wireless and Mobile Technologies for Smart Cities, (Philadelphia, Pennsylvania, USA), pp. 11–18, ACM, 2014.
- [84] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, “Veins : The Open Source Vehicular Network Simulation Framework,” in *Recent Advances in Network Simulation*, pp. 215–252, Springer (Recent Advances in Network Simulation), 2019.
- [85] M. Chaqfeh, N. Mohamed, I. Jawhar, and J. Wu, “Vehicular cloud data collection for intelligent transportation systems,” Proceedings of the Third International Conference on Smart Cloud Networks & Systems (SCNS), (Dubai, United Arab Emirates), pp. 1–6, IEEE, 2016.

- [86] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles," *China Communications*, vol. 13, no. 2, pp. 140–149, 2016.
- [87] X. Li, P. Tian, and M. Kong, "A new particle swarm optimization for solving constrained optimization problems," *Journal of Systems & Management*, vol. 16, no. 2, pp. 120–129, 2007.
- [88] B. Sahoo, D. Kumar, and S. K. Jena, "Performance analysis of greedy Load balancing algorithms in Heterogeneous Distributed Computing System," Proceedings of International Conference on High Performance Computing and Applications (ICHPCA), (Bhubaneswar, India), pp. 1–7, IEEE, 2014.
- [89] N. S. Ghumman and R. Kaur, "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system," Proceedings of the 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (Dallas-Fortworth, TX, USA), pp. 1–5, IEEE, 2015.
- [90] Y. Qin, D. Huang, and X. Zhang, "VehiCloud : Cloud computing facilitating routing in vehicular networks," Proceedings of the 11th International Conference on Trust, Security and Privacy in Computing and Communications, (Liverpool, UK), pp. 1438–1445, IEEE, 2012.
- [91] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications : Merging VANET with cloud computing," Proceedings of the 4th International Conference on Cloud Computing Technology and Science, (Taipei, Taiwan), pp. 606–609, IEEE, 2012.
- [92] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-Cloud : vehicular cyber-physical systems and cloud computing," Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, (Barcelona, Spain), pp. 1–5, ACM, 2011.
- [93] K. Mershad and H. Artail, "Finding a STAR in a Vehicular Cloud," *IEEE Intelligent transportation systems magazine*, vol. 5, no. 2, pp. 55–68, 2013.
- [94] D. Baby, R. Sabareesh, R. Saravanaguru, and A. Thangavelu, "VCR : vehicular cloud for road side scenarios," in *Advances in computing and information technology*, pp. 541–552, Springer, 2013.
- [95] M. Alhamad, T. Dillon, and E. Chang, "Conceptual SLA framework for cloud computing," Proceedings of the 4th International Conference on Digital Ecosystems and Technologies, (Dubai, United Arab Emirates), pp. 606–610, IEEE, 2010.
- [96] T. Cunningham, K. Stam, and J. Faath, "Juddi user guide," *Accessed in URL http://juddi.apache.org/docs/3.x/userguide/html/chap-UDDI_Registry.html* dated, pp. 11–12, 2013.
- [97] N. Zingirian and C. Valenti, "Sensor clouds for Intelligent Truck Monitoring," Proceedings of International Conference on Intelligent Vehicles Symposium, (Madrid, Spain), pp. 999–1004, IEEE, 2012.

-
- [98] A. Paranjothi, M. S. Khan, and M. Atiquzzaman, “DFCV : a novel approach for message dissemination in connected vehicles using dynamic fog,” Proceedings of the 16th International Conference on Wired/Wireless Internet Communication, (Boston, MA, USA), pp. 311–322, IFIP, 2018.
- [99] A. Ullah, S. Yaqoob, M. Imran, and H. Ning, “Emergency message dissemination schemes based on congestion avoidance in VANET and vehicular FoG computing,” *IEEE Access*, vol. 7, pp. 1570–1585, 2018.
- [100] T. Lu, S. Chang, and W. Li, “Fog computing enabling geographic routing for urban area vehicular network,” *Peer-to-Peer Networking and Applications*, vol. 11, no. 4, pp. 749–755, 2018.
- [101] A. J. Kadhim and S. A. H. Seno, “Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks,” *Ad Hoc Networks*, vol. 84, pp. 68–81, 2019.
- [102] E. Al-Ezaly, M. Abu-Elkeir, and A. Riad, “Collaborative vehicle location management service for enhanced hybrid reactive and proactive multicast in VANETs,” *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 691–704, 2017.
- [103] X. Zhang, X. Zhang, and C. Gu, “A micro-artificial bee colony based multicast routing in vehicular ad hoc networks,” *Ad Hoc Networks*, vol. 58, pp. 213–221, 2017.
- [104] M. Arif, G. Wang, T. Wang, and T. Peng, “SDN-based secure VANETs communication with fog computing,” Proceedings of the 11th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, (Melbourne, NSW, Australia), pp. 46–59, Springer, 2018.
- [105] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, “Software Defined Networking-based vehicular ad hoc network with Fog computing,” Proceedings of International Symposium on Integrated Network Management (IM), (Ottawa, ON, Canada), pp. 1202–1207, IFIP/IEEE, 2015.
- [106] T. Nebbou, M. Lehsaini, H. Fouchal, and M. Ayaida, “An urban location service for vehicular area networks,” *Concurrency and Computation : Practice and Experience*, vol. 31, no. 24, p. e4693, 2019.
- [107] T. Nebbou and M. Lehsaini, “Greedy curvemetric-based routing protocol for VANETs,” in *Proceedings of International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, (Tangier, Morocco), pp. 1–6, IEEE, 2018.
- [108] I. Chlamtac, M. Conti, and J. J.-N. Liu, “Mobile ad hoc networking : imperatives and challenges,” *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003.
- [109] A. Heidari, M. A. Jabraeil Jamali, N. Jafari Navimipour, and S. Akbarpour, “Internet of things offloading : ongoing issues, opportunities, and future challenges,” *International Journal of Communication Systems*, vol. 33, no. 14, p. e4474, 2020.

- [110] M. U. Ilyas, M. Ahmad, and S. Saleem, "Internet-of-Things-Infrastructure-as-a-Service : The democratization of access to public Internet-of-Things Infrastructure," *International Journal of Communication Systems*, vol. 33, no. 16, p. e4562, 2020.
- [111] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "SDN-based real-time urban traffic analysis in VANET environment," *Computer Communications*, vol. 149, pp. 162–175, 2020.
- [112] S. Liu, J. Tian, X. Deng, Y. Zhi, and J. Bian, "Stackelberg game-based task offloading in vehicular edge computing networks," *International Journal of Communication Systems*, vol. 34, no. 16, p. e4947, 2021.
- [113] R. P. Krishnan and A. R. P. Kumar, "A collaborative strategy for detection and eviction of Sybil attacker and Sybil nodes in VANET," *International Journal of Communication Systems*, vol. 34, no. 3, 2021.
- [114] T. Nebbou, M. Lehsaini, and H. Fouchal, "Partial backwards routing protocol for vanets," *Vehicular Communications*, vol. 18, p. 100162, 2019.
- [115] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R. A. Mini, and A. A. Loureiro, "Data communication in VANETs : Protocols, applications and challenges," *Ad Hoc Networks*, vol. 44, pp. 90–103, 2016.
- [116] D. Jiang and L. Delgrossi, "IEEE 802.11p : Towards an international standard for wireless access in vehicular environments," in *Proceedings of International Conference on Vehicular Technology (VTC Spring 2008)*, (Marina Bay, Singapore), pp. 2036–2040, IEEE, 2008.
- [117] K. Kharb, B. Sharma, and C. Trilok, "Reliable and congestion control protocols for wireless sensor networks," *International Journal of Engineering and Technology Innovation*, vol. 6, no. 1, p. 68, 2016.
- [118] B. Sharma, G. Srivastava, and J. C.-W. Lin, "A bidirectional congestion control transport protocol for the internet of drones," *Computer Communications*, vol. 153, pp. 102–116, 2020.
- [119] T. Chand and B. Sharma, "HRCCTP : a hybrid reliable and congestion control transport protocol for wireless sensor networks," in *Proceedings of International Conference on Sensors*, (Busan, Korea South), pp. 1–4, IEEE, 2015.
- [120] T. S. Gnanasekar and D. Samiappan, "Impact of hybridized rider optimization with cuckoo search algorithm on optimal VANET routing," *International Journal of Communication Systems*, vol. 34, no. 16, p. e4954, 2021.
- [121] B. Sharma and T. C. Aseri, "A hybrid and dynamic reliable transport protocol for wireless sensor networks," *Computers & Electrical Engineering*, vol. 48, pp. 298–311, 2015.

-
- [122] C. Ghorai and I. Banerjee, “A constrained delaunay triangulation based rsus deployment strategy to cover a convex region with obstacles for maximizing communications probability between v2i,” *Vehicular Communications*, vol. 13, pp. 89–103, 2018.
- [123] T. Nebbou, H. Fouchal, M. Lehsaini, and M. Ayaida, “A realistic location service for VANETs,” in *Proceedings of the 17th International Conference on Innovations for Community Services (I4CS’2017)*, (Darmstadt, Germany), pp. 191–196, Springer, 2017.
- [124] F. Neumann and C. Witt, “Combinatorial optimization and computational complexity,” in *Bioinspired Computation in Combinatorial Optimization*, pp. 9–19, Springer, 2010.
- [125] A. Jaszkiewicz, “Genetic local search for multi-objective combinatorial optimization,” *European journal of operational research*, vol. 137, no. 1, pp. 50–71, 2002.
- [126] E. L. Lawler and D. E. Wood, “Branch-and-bound methods : A survey,” *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.
- [127] R. Bellman, “Dynamic programming,” *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
- [128] M. Dorigo and G. Di Caro, “Ant colony optimization : a new metaheuristic,” in *Proceedings of International Congress on Evolutionary Computation (CEC’99)*, vol. 2, (Washington, DC, USA), pp. 1470–1477, IEEE, 1999.
- [129] A. Hertz and D. de Werra, “The tabu search metaheuristic : how we used it,” *Annals of mathematics and artificial intelligence*, vol. 1, no. 1-4, pp. 111–121, 1990.
- [130] M. Palpant, *Recherche exacte et approchée en optimisation combinatoire : schémas d’intégration et applications*. PhD thesis, Atelier national de reproduction des thèses, 2008.
- [131] C. R. Reeves, *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., 1993.
- [132] O. HaJJI, “Contribution au développement de méthodes d’optimisation stochastiques, application à la conception des dispositifs electrotechniques,” *Mémoire de thèse de Doctorat, Université des sciences et technologies de Lille*, 2003.
- [133] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, “Simu5G—An OM-NeT++ library for end-to-end performance evaluation of 5G networks,” *IEEE Access*, vol. 8, pp. 181176–181191, 2020.
- [134] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, “An overview of internet of vehicles,” *China communications*, vol. 11, no. 10, pp. 1–15, 2014.
- [135] S. Olariu, I. Khalil, and M. Abuelela, “Taking VANET to the clouds,” *International Journal of Pervasive Computing and Communications*, vol. 7, no. 1, pp. 7–21, 2011.

- [136] J. S. Sussman, *Perspectives on intelligent transportation systems (ITS)*. Springer Science & Business Media, 2008.
- [137] T. Khalid, A. N. Khan, M. Ali, A. Adeel, A. u. R. Khan, and J. Shuja, “A fog-based security framework for intelligent traffic light control system,” *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24595–24615, 2019.
- [138] J. Liu, J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, and J. Shen, “Secure intelligent traffic light control using fog computing,” *Future Generation Computer Systems*, vol. 78, pp. 817–824, 2018.
- [139] H. Wei, G. Zheng, H. Yao, and Z. Li, “Intellilight : A reinforcement learning approach for intelligent traffic light control,” Proceedings of the 24th SIGKDD International Conference on Knowledge Discovery & Data Mining, (London, United Kingdom), pp. 2496–2505, ACM, 2018.
- [140] N. Bermad, S. Zemmoudj, and M. Omar, “Context-aware negotiation, reputation and priority traffic light management protocols for VANET-based smart cities,” *Telecommunication Systems*, vol. 72, no. 1, pp. 131–153, 2019.
- [141] M. B. Younes and A. Boukerche, “Intelligent traffic light controlling algorithms using vehicular networks,” *IEEE transactions on vehicular technology*, vol. 65, no. 8, pp. 5887–5899, 2015.
- [142] W. Wen, “A dynamic and automatic traffic light control expert system for solving the road congestion problem,” *Expert Systems with Applications*, vol. 34, no. 4, pp. 2370–2381, 2008.
- [143] M. Fogue, P. Garrido, F. J. Martinez, J. C. Cano, C. T. Calafate, P. Manzoni, and M. Sanchez, “Prototyping an automatic notification scheme for traffic accidents in vehicular networks,” IFIP Wireless Days (WD), (Niagara Falls, ON, Canada), pp. 1–5, IEEE, 2011.
- [144] A. F. Santamaria and C. Sottile, “Smart traffic management protocol based on vanet architecture,” *Advances in Electrical and Electronic Engineering*, vol. 12, no. 4, pp. 279–288, 2014.
- [145] O. Tomescu, I. M. Moise, A. E. Stanciu, and I. Batros, “Adaptive traffic light control system using ad-hoc vehicular communications network,” *U.P.B. Sci. Bull., Series D*, vol. 74, no. 2, pp. 67–78, 2012.
- [146] K. Pandit, D. Ghosal, H. M. Zhang, and C. Chuah, “Adaptive Traffic Signal Control With Vehicular Ad hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459–1471, 2013.
- [147] V. Gupta, R. Kumar, K. S. Reddy, and B. K. Panigrahi, “Intelligent traffic light control for congestion management for smart city development,” Proceedings of IEEE Region 10 Symposium (TENSYP), (Cochin, India), pp. 1–5, IEEE, 2017.

-
- [148] Y. Bravo, J. Ferrer, G. Luque, and E. Alba, “Smart mobility by optimizing the traffic lights : A new tool for traffic control centers,” International Conference on Smart Cities, (Málaga, Spain), pp. 147–156, Springer, 2016.
- [149] F. Teklu, A. Sumalee, and D. Watling, “A genetic algorithm approach for optimizing traffic control signals considering routing,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 31–43, 2007.
- [150] F. Yan, M. Dridi, and A. El Moudni, “A branch and bound algorithm for new traffic signal control system of an isolated intersection,” Proceedings of International Conference on Computers & Industrial Engineering, (Troyes, France), pp. 999–1004, IEEE, 2009.
- [151] M. B. Younes, A. Boukerche, and A. Mammeri, “Context-aware traffic light self-scheduling algorithm for intelligent transportation systems,” Proceedings of International Conference on Wireless Communications and Networking, (Doha, Qatar), pp. 1–6, IEEE, 2016.
- [152] H. Wei, G. Zheng, V. Gayah, and Z. Li, “A survey on traffic signal control methods,” *arXiv preprint arXiv :1904.08117*, vol. 1, no. 1, pp. 1–32, 2019.
- [153] F. V. Webster, “Traffic signal settings,” tech. rep., Department of Scientific and Industrial Research, London, England, 1958.
- [154] X. Wu, S. Deng, X. Du, and J. Ma, “Green-wave traffic theory optimization and analysis,” *World Journal of Engineering and Technology*, vol. 2, no. 3, pp. 14–19, 2014.
- [155] J. D. C. Little, M. D. Kelson, and N. H. Gartner, “MAXBAND : a versatile program for setting signals on arteries and triangular networks,” tech. rep., Massachusetts Institute of Technology (MIT), Sloan School of Management, 1981.
- [156] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, “A multi-band approach to arterial traffic signal optimization,” *Transportation Research Part B : Methodological*, vol. 25, no. 1, pp. 55–74, 1991.
- [157] C. Stamatiadis and N. H. Gartner, “Multiband-96 : a program for variable-bandwidth progression optimization of multiarterial traffic networks,” *Transportation Research Record*, vol. 1554, no. 1, pp. 9–17, 1996.
- [158] P. Varaiya, “The max-pressure controller for arbitrary networks of signalized intersections,” *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pp. 27–66, 2013.
- [159] S.-B. Cools, C. Gershenson, and B. D’Hooghe, “Self-organizing traffic lights : A realistic simulation,” *Advances in applied self-organizing systems*, pp. 45–55, Springer, 2013.
- [160] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, “The SCOOT online traffic signal optimisation technique,” *Traffic Engineering & Control*, vol. 23, no. 4, pp. 190–192, 1982.

- [161] P. R. Lowrie, *Scats, sydney co-ordinated adaptive traffic system : A traffic responsive method of controlling urban traffic*. Darlinghurst, NSW, Australia : Roads and Traffic Authority NSW, Traffic Control Section, 1990.
- [162] N. H. Gartner, F. J. Pooran, and C. M. Andrews, “Optimized policies for adaptive control strategy in real-time traffic adaptive control systems : Implementation and field testing,” *Transportation Research Record*, vol. 1811, no. 1, pp. 148–156, 2002.
- [163] A. Varga and R. Hornig, “An Overview of the OMNeT++ Simulation Environment,” International Conference on SIMUTools, (Marseille, France), pp. 1–10, ACM, 2008.
- [164] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO—simulation of urban mobility : an overview,” Proceedings of the 3th International Conference on Advances in System Simulation (SIMUL’2011), (Barcelona, Spain), pp. 63–68, ThinkMind, 2011.
- [165] L. Mészáros, A. Varga, and M. Kirsche, “INET framework,” Recent Advances in Network Simulation, pp. 55–106, Springer, 2019.
- [166] C. Tom, “Traffic lights.” https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html, 11 2021.
- [167] G. Martnez, “Simulation of a virtual Traffic Light System using IEEE 802.11p,” Master’s thesis, Politecnico Di Torino, Italy, Politecnico Di Torino, Italy, 2019.

Résumé

Cette thèse traite le problème de la dissémination de données dans les réseaux véhiculaires (VANETs). Cette dissémination se fait soit de véhicule à véhicule, soit de véhicule à diverses infrastructures du réseau véhiculaire telles que les RSUs, les serveurs Cloud, ... etc. Motivés par les exigences des communications véhiculaires, nous avons visé à travers cette thèse d'améliorer les performances des VANETs via l'utilisation du Cloud computing et nous avons proposé de nouvelles approches en guise de solutions pour deux problèmes différents. La première vise à garantir une couverture réseaux et des performances de routage acceptables dans un réseau routier tout en minimisant le nombre de RSUs déployés au niveau des intersections de la route, tandis que la deuxième se concentre sur le problème de la congestion routière et du délai pour atteindre la destination et représente un nouveau protocole de gestion du trafic routier qui se base sur le Cloud computing.

Dans la première contribution, nous avons proposé une solution de déploiement efficace des RSUs dans les VANETs basée sur les versions améliorées des métaheuristiques algorithmes génétiques et recuit simulé. Les résultats obtenus étaient prometteurs puisque le nombre de RSUs déployées au niveau des intersections a été grandement diminué tout en gardant de bonnes performances de routage et un bon taux de couverture dans le réseau. Dans la deuxième contribution, nous avons proposé le protocole CCITL; une technique de gestion du trafic routier intelligente qui se base sur les feux de circulation tricolores et le Cloud computing pour fluidifier le trafic et minimiser les délais d'attente. Dans cette solution de la gestion de trafic, CCITL implique le Cloud conventionnel comme fournisseur de services dont les serveurs sont dédiés aux calculs de la densité et du temps d'attente des véhicules dans chaque voie de la route. Les résultats obtenus nous ont permis de constater que le protocole proposé constitue une solution fiable puisque le trafic est devenu plus fluide, les embouteillages ont grandement diminué, et le temps d'attente des véhicules dans plusieurs segments a été bien réduit.

Mots clés : Algorithmes génétiques, Cloud Computing, Dissémination de données, Déploiement des RSUs, Gestion du trafic routier, Recuit simulé, VANETs.

Abstract

This thesis addresses the problem of data dissemination in vehicular networks (VANETs). This dissemination occurs either from vehicle to vehicle, or from vehicle to various vehicular network infrastructures such as RSUs, Cloud servers, etc. Motivated by the requirements of vehicular communications, we aimed in this thesis to improve the performance of VANETs through the use of Cloud computing and proposed new approaches as solutions to two different problems. The first aims for ensuring acceptable network coverage and routing performance in a road network while minimizing the number of RSUs deployed at road intersections, while the second focuses on the problem of road congestion and delay in reaching the destination and represents a new road traffic management protocol based on Cloud computing.

In the first contribution, we proposed a solution for the efficient deployment of RSUs in VANETs based on improved versions of the genetic algorithm and simulated annealing metaheuristics. The results obtained were promising, since the number of RSUs deployed at intersections was greatly reduced while maintaining good routing performance and a good coverage rate in the network. In the second contribution, we proposed the CCITL protocol, an intelligent road traffic management technique that uses traffic lights and cloud computing to smooth traffic flow and minimize waiting times. In this traffic management solution, CCITL involves the conventional Cloud as a service provider whose servers are dedicated to calculating the density and waiting time of vehicles in each lane of the road. The results obtained show that the proposed protocol is a reliable solution, since traffic has become more fluid, traffic jams have been greatly reduced, and the waiting time for vehicles in several segments has been greatly reduced.

Keywords: Genetic algorithms, Cloud Computing, Data dissemination, RSU deployment, Road traffic management, Simulated annealing, VANETs.

ملخص

تتناول هذه الأطروحة مشكلة نشر البيانات في شبكات المركبات (VANETs) حيث يتم هذا النشر إما من مركبة إلى مركبة، أو من مركبة إلى البنية التحتية المختلفة لشبكة المركبات مثل RSUs، الخوادم السحابية،... إلخ. بدافع من متطلبات اتصالات المركبات. استهدفنا من خلال هذه الأطروحة تحسين أداء VANETs من خلال استخدام الحوسبة السحابية واقتراحنا أساليب جديدة كحلول لمشكلتين مختلفتين. يهدف الأول إلى ضمان تغطية الشبكة المقبولة وأداء التوجيه في شبكة الطرق مع تقليل عدد وحدات RSUs المنتشرة عند تقاطعات الطرق، بينما يركز الثاني على مشكلة الازدحام والتأخير على الطرق للوصول إلى الوجهة ويمثل إدارة جديدة لحركة المرور على بروتوكول قائم على الحوسبة السحابية.

في المساهمة الأولى، اقترحنا حلاً للنشر الفعال لـ RSUs في VANETs استناداً إلى الإصدارات المحسنة من الخوارزميات الجينية والتلدين المحاكي. كانت النتائج التي تم الحصول عليها واعدة حيث تم تقليل عدد وحدات RSUs المنتشرة في التقاطعات بشكل كبير مع الحفاظ على أداء التوجيه الجيد ومعدل التغطية الجيد في الشبكة. في المساهمة الثانية، اقترحنا بروتوكول CCITL؛ تقنية ذكية لإدارة حركة المرور على الطرق تعتمد على إشارات المرور والحوسبة السحابية لتبسيط حركة المرور وتقليل أوقات الانتظار. في حل إدارة حركة المرور هذا، يتضمن CCITL السحابة التقليدية كمزود خدمة خوادمه مخصصة لحساب كثافة المركبات ووقت انتظارها في كل ممر من الطريق. سمحت لنا النتائج التي تم الحصول عليها أن البروتوكول المقترح هو حل موثوق لأن حركة المرور أصبحت أكثر مرونة، وانخفضت الاختناقات المرورية بشكل كبير، كما تم تقليل وقت انتظار المركبات في عدة قطاعات.

الكلمات المفتاحية: الخوارزميات الجينية، الحوسبة السحابية، نشر البيانات، نشر RSUs، إدارة حركة المرور على الطرق، التلدين المحاكي، VANETs