



HAL
open science

Architectures de communications pour les véhicules autonomes connectés

Lucas Bréhon-Grataloup

► **To cite this version:**

Lucas Bréhon-Grataloup. Architectures de communications pour les véhicules autonomes connectés. Réseaux et télécommunications [cs.NI]. Université Paul Sabatier - Toulouse III, 2023. Français. NNT : 2023TOU30212 . tel-04428275

HAL Id: tel-04428275

<https://theses.hal.science/tel-04428275v1>

Submitted on 31 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le 15/11/2023 par :

Lucas BRÉHON--GRATALOUP

Architectures de communication pour les véhicules autonomes connectés

JURY

HIND CASTEL	Professeure, TELECOM SudParis	Présidente
BERTRAND DUCOURTHIAL	Professeur des Universités, UTC	Rapporteur
JÉRÔME HÄRRI	Professeur, Institut Eurecom	Rapporteur
MARIE-PIERRE GLEIZES	Professeure, Université Toulouse III	Examinatrice
HOUDA LABIOD	Professeure, TELECOM Paris	Examinatrice
FABRICE VALOIS	Professeur des Universités, INSA Lyon	Examinateur
ANDRÉ BOTTARO	Responsable d'unité R&D, Orange	Invité
ANDRÉ-LUC BEYLOT	Professeur des Universités, Toulouse INP	Co-Directeur
RAHIM KACIMI	Maître de Conférences, Université Toulouse III	Co-Directeur

École doctorale et spécialité :

MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture

Unité de Recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

André-Luc BEYLOT et Rahim KACIMI

Rapporteurs :

Bertrand DUCOURTHIAL et Jérôme HÄRRI

À eux

REMERCIEMENTS

Une thèse a la réputation d'une aventure personnelle. J'ai toutefois eu la chance de vivre cette expérience sous le regard de nombreuses personnes, fidèles au poste ou rencontrées au cours de ces trois années. À vous qui m'avez accompagné, consciemment ou non, ces mots sont le témoin de ma gratitude.

Je tiens à remercier en premier lieu mes directeurs de thèse, André-Luc Beylot et Rahim Kacimi. Je n'ai toujours pas fini de vous exprimer ma reconnaissance devant la confiance que vous m'avez témoignée, en me proposant ce sujet et en m'accompagnant tout au long de cette aventure. Vous avez vu, avant que j'accepte de le voir, que mon avenir se trouve dans la recherche. Merci pour vos retours, vos enseignements, votre temps, votre patience. Ces trois années de travail avec vous furent un véritable plaisir. Merci André-Luc pour la précision de ton travail de correction sur le manuscrit, digne d'un orfèvre. Merci Rahim pour les heures de relecture, les nombreuses corrections et les envois de papiers à deux heures du matin, j'espère que tu as récupéré tes Miles.

Je veux aussi remercier Bertrand Ducourthial et Jérôme Härri qui ont posé des regards d'experts sur mes travaux de thèse en réalisant des rapports particulièrement complets, pointilleux et instructifs. Je remercie également les membres de mon jury de soutenance, Hind Castel sa présidente, Marie-Pierre Gleizes, Houda Labiod et Fabrice Valois, professeurs et chercheurs respectés, d'avoir pris le temps d'évaluer mes travaux. Vos questions pertinentes après la soutenance ont conduit à des discussions riches qui ont fait mûrir mes réflexions sur les sujets abordés. Merci à André Bottaro d'avoir accepté l'invitation à ma soutenance, et pour ses commentaires constructifs au sujet du manuscrit.

Je remercie aussi l'ensemble du personnel de l'équipe RMESS de l'IRIT, avec une mention spéciale à l'équipe des secrétaires : Sylvie, Vanessa, Léna. Vous m'avez sorti de sacrées épines du pied! Merci aussi à tous les permanents de l'équipe : Manu, Béatrice, Riadh, Katia, Jean-Luc, Jérôme, Julien, Urtzi, Gentian, Rejane. Merci pour l'environnement favorable et les discussions fructueuses.

Malgré le retard administratif lié, entre autres, au covid, j'ai tout de même eu le privilège de conduire une campagne d'évaluation expérimentale dans la navette autonome du projet VILAGIL et de l'action autOCampus, ce qui n'aurait aucunement été possible sans la présence, les compétences et le temps de Marie-José, Mickael et Philippe. Nos dizaines d'heures passées à parcourir ce sacré circuit ont bien fini par porter leurs fruits! Merci à

Toulouse Métropole pour leur confiance, ainsi qu'à Marie-Pierre Gleizes de m'avoir donné l'opportunité de participer à ce grand projet. J'espère que les résultats sont à la hauteur de vos attentes.

Je ne saurais oublier l'ensemble des stagiaires et ingénieurs de recherche qui m'ont chacun aidé, à leur manière, à concrétiser ce projet. Benjamin, tes connaissances techniques et ta curiosité nous ont mené aux discussions qui ont abouti à *l'idée*, la fameuse bonne idée, derrière la deuxième contribution. Merci pour ton répondeur, et désolé pour les journées rallongées. Axel, merci pour ton temps passé à rendre notre bazar logiciel beaucoup plus agréable à manipuler, et surtout merci pour ton apport lors des expérimentations étendues de la deuxième contribution. Pierre, tu n'avais rien demandé à personne et tu t'es retrouvé à faire le poireau pendant nos prises de mesures. Sans toi, jamais nous n'aurions fini à temps, alors merci.

Je remercie bien-sûr ma famille : mes parents Sylvie et Jean-Luc, ma sœur Justine, pour leur patience et leur indulgence devant mes prises de décision, et surtout pour leur présence. Je remercie ma grand-mère Emma, ainsi que Cathy et Marc, de m'avoir donné le sourire à chaque instant passé en leur compagnie. Vous voyez en moi ce qui m'est invisible. Je ne peux m'empêcher de remercier Gérard et Jeannot, qui n'ont jamais quitté mes pensées. Il y a de vous dans ce que je suis devenu. J'espère vous avoir tous rendu fiers.

Merci aussi à ceux qui ont eu l'énergie de rester en contact avec moi malgré les bouleversements de ces trois dernières années. Pierre, Samson, Florian, nous restons liés par la musique, par le courage, par la volonté d'avancer. Simon, nos fameuses discussions de séminaire, ces rayons de fraternité parfaitement inattendus m'ont galvanisé dans les instants les plus nuageux. J'espère qu'ils feront de même pour toi.

Je remercie ici tous ceux que je ne peux pas oublier. Merci Atlas pour ta protection. Merci Hermès de m'avoir transmis ta résilience. Merci Val pour nos moments de tranquillité. Merci Donovan et Georges pour l'inspiration. Merci Belial, Astarté et Nox pour votre force, devenue notre complicité. Continuons d'apprendre à nous connaître.

Car tout a commencé avec elle, je trouve approprié de conclure ces remerciements avec Maya, qui m'a poussé à me découvrir. Sans toi, je ne me serais jamais écouté, et ces trois ans de prises de risque n'auraient sûrement jamais eu lieu. Je continue ma route en me faisant un peu plus confiance. Merci.

Ce travail a été réalisé au sein de l'action autOCampus du projet Vilagil, projet co-financé par Toulouse Métropole et par France 2030 dans le cadre du programme Territoires d'innovation opéré par la Banque des Territoires.

RÉSUMÉ

Les véhicules autonomes représentent une part grandissante du marché automobile. Leur capacité de perception de leur propre environnement améliore la sécurité routière ainsi que le confort à bord. Toutefois, les performances des applications d'autonomie sont limitées, de par l'absence de moyens de communication entre véhicules partageant la même route. Les véhicules autonomes nécessitent également une connectivité.

De nombreux travaux ont proposé des solutions de connectivité véhiculaire visant à étendre les applications pouvant être fournies par les véhicules du futur. Ces applications appartiennent à des domaines variés, allant de la conduite sans chauffeur au divertissement à bord. Toutefois, la diversité des technologies utilisées et des contextes étudiés limite les apports de ces architectures de communication, notamment dans le cadre de déploiements urbains.

Dans cette thèse, inscrite dans le cadre de l'action autoCampus du projet VILAGIL porté par Toulouse Métropole, nous nous proposons d'étudier ces architectures afin de concevoir des solutions plus robustes, tirant profit de technologies de pointe.

Nous proposons dans un premier temps CAVTOMECC, notre solution algorithmique et architecturale d'optimisation de délestage de tâches véhiculaires tirant profit de multiples technologies d'accès radio. En exploitant habilement les ressources de calcul disponibles à proximité des utilisateurs, CAVTOMECC permet de traiter de manière satisfaisante les tâches critiques issues des véhicules, tout en limitant le risque de surcharge au niveau des serveurs. Nous avons procédé à une évaluation par simulation, mettant en évidence une réduction des délais de traitement des tâches prioritaires, aboutissant à de meilleurs taux de succès.

Nous proposons dans un second temps CERES-V2X, un mécanisme de stabilisation de connectivité véhiculaire robuste face aux situations de perturbation physique sur les liens de communication. Un système de mesure de qualité de service permet de basculer proactivement d'une technologie radio à une autre, afin de limiter les pertes de connectivité. Nous avons déployé CERES-V2X en conditions réelles dans un campus universitaire, accompagné d'un véhicule autonome connecté. Les performances observées montrent d'importantes améliorations de performances réseau en matière de stabilité et de délai.

ABSTRACT

Autonomous vehicles hold a growing share of the automotive market. Their ability to perceive their own environment improves road safety as well as comfort on board. However, the performance of autonomous applications is limited by the lack of communication between vehicles sharing the same road. Autonomous vehicles also require connectivity.

Numerous studies have proposed vehicular connectivity solutions designed to extend the applications that may be provided by future vehicles. These applications range from robust autonomous driving to on-board entertainment. However, the diversity of technologies and studied contexts limits the contributions of these communication architectures, particularly in the context of urban deployments.

In this thesis, embedded within the autOCampus action of the VILAGIL project led by Toulouse Métropole, we aim at studying these architectures in order to design more robust solutions, taking advantage of state-of-the-art technologies.

Firstly, we propose CAVTOMECA, our algorithmic and architectural solution to optimize vehicular task offloading, taking advantage of multiple radio access technologies. By skilfully exploiting the computing resources available close to users, CAVTOMECA enables critical vehicular tasks to be handled in a timely manner, while limiting the risk of server overload. We have carried out a performance evaluation by simulation, demonstrating a reduction in processing delays for high-priority tasks, leading to improved success rates.

Secondly, we propose CERES-V2X, a vehicular connectivity stabilization mechanism that is robust to physical disturbances occurring on communication links. A quality-of-service measurement system enables proactive handover from one radio technology to another, to limit loss of connectivity. We deployed CERES-V2X under real-life conditions in a university campus, with the help of a connected autonomous vehicle. Performance evaluation show significant improvements in network performance in terms of stability and delay.

TABLE DES MATIÈRES

Liste des figures	xv
Liste des tableaux	xvii
Glossaire	xxi
1 Introduction	1
1.1 Contexte	1
1.1.1 L'automobile dans notre quotidien	1
1.1.2 Les équipements des automobiles modernes	2
1.1.3 L'autonomie intégrée aux véhicules	2
1.1.4 Les usages des véhicules autonomes dans les transports urbains	4
1.2 Motivations	5
1.3 Méthodologie	6
1.3.1 Un œil sur le concret	6
1.3.2 Différents prismes technologiques abordés lors du développement	7
1.4 Contributions et plan du mémoire	7
1.4.1 CAVTOMECA : faciliter le délestage de tâches de calcul véhiculaire	8
1.4.2 CERES-V2X : Stabiliser la connectivité véhiculaire dans un contexte urbain	8
I Optimisation de délestage de calcul véhiculaire dans un contexte MEC urbain	11
2 Etat de l'art	15
2.1 Les niveaux d'autonomie de conduite	15
2.2 Les capteurs véhiculaires embarqués	16
2.2.1 Les capteurs de position	16
2.2.2 Les capteurs de comportement routier	17
2.2.3 Les capteurs d'environnement	17
2.3 Le traitement des données de capteurs	19

2.3.1	Le cloud computing	19
2.3.2	L'edge computing	20
2.3.3	Le multi-access edge computing	21
2.3.4	Le vehicular edge computing	26
2.4	Les mécanismes de calcul dans le vehicular edge computing	28
2.4.1	Intelligence de périphérie	28
2.4.2	Intelligence collective entre hôtes	30
2.4.3	Optimisation des ressources de calcul	30
2.4.4	Traitement des données massives	31
2.4.5	Flexibilité des performances	31
2.5	Notre positionnement	32
3	CAVTOMECC : optimiser le délestage de tâches dans une architecture VEC	35
3.1	Étude du besoin	36
3.1.1	Différenciation D2D et longue portée	36
3.1.2	Faisabilité d'une solution de délestage hybride	37
3.1.3	Réponse au besoin	38
3.2	Le délestage	38
3.3	Vue d'ensemble de CAVTOMECC	40
3.4	Circulation des données de contexte	40
3.4.1	Contenu	40
3.4.2	Mécanisme	43
3.4.3	Comportement suite à la réception d'une tâche	44
3.5	Choix de la destination	46
3.5.1	Problème de minimisation de délai	46
3.5.2	Solution à une destination et un canal	47
3.5.3	Transformation en problème d'optimisation	49
3.5.4	Preuve de NP-complétude	50
3.5.5	Heuristique	51
3.6	Implantation	51
3.6.1	Délestage pour traitement immédiat	52
3.6.2	Délestage avec gestion de files d'attente	53
3.6.3	Élaboration du simulateur	55
3.7	Évaluation	57
3.7.1	Configuration de l'évaluation	58
3.7.2	Variation des paramètres	59
3.7.3	Intervalles de confiance	61
3.7.4	Utilisation des ressources	61
3.7.5	Temps d'exécution	64

3.7.6	Taux de succès	66
3.7.7	Effet de la prise en charge des files d'attente	67
3.8	Limites et discussion	68
3.9	Conclusion	69
 II Stabilisation de connectivité V2X par handover proactif multi-RAT		71
4	Etat de l'art	73
4.1	La communication véhicule-à-tout	73
4.2	Les technologies V2X héritant du WiFi	74
4.2.1	802.11p, la première technologie radio fondamentalement véhiculaire	74
4.2.2	802.11bd, un perfectionnement qui se fait attendre	77
4.3	Les technologies V2X héritant du cellulaire mobile	77
4.3.1	Le LTE-V2X	78
4.3.2	L'arrivée de la 5G	79
4.3.3	La 5G-V2X	80
4.3.4	La prometteuse Release 17	82
4.4	L'arrivée du Cloud-RAN	83
4.5	Les difficultés des communications V2X	84
4.5.1	Zones de couverture	84
4.5.2	Forte mobilité et bascule horizontale	86
4.6	Les expérimentations réelles	90
4.7	Notre positionnement	91
5	CERES-V2X : Stabiliser la connectivité V2X par handover proactif multi-RAT	95
5.1	Travail préliminaire : couverture cellulaire	96
5.2	Vue d'ensemble de CERES-V2X	98
5.3	Construction de la grille	100
5.4	Mécanisme de bascule	103
5.5	Évaluation des performances	103
5.5.1	Conditions expérimentales	105
5.5.2	Comparaison DSRC / C-V2X	107
5.5.3	Optimisation de bascule	108
5.6	Limites et discussion	112
5.7	Conclusion	113
 Conclusions et perspectives		115

LISTE DES FIGURES

1.1	Méthodologie suivie pendant la thèse	7
2.1	Structure <i>Multi-access Edge Computing</i> (MEC) selon [60]	23
2.2	Comparaison des architectures MEC (a) et <i>Vehicular Edge Computing</i> (VEC) (b).	27
3.1	Scénario de communications véhiculaires multi-RAT	37
3.2	Structure des données utiles des différents messages impliqués.	42
3.3	Mécanisme de transmission du contexte après réception d'une tâche.	43
3.4	Chronologie de l'exploitation du contexte fourni par les RSU.	44
3.5	Diagramme de séquence : informations et délestage.	45
3.6	Vue simplifiée du circuit simulé.	59
3.7	Utilisation des ressources de calcul, pour chaque <i>Road-Side Unit</i> (RSU), $M = 5$, $N = 2$	62
3.8	Délai de traitement total : $T^{1,0,s}$ (a) et $T^{1,0,s,*}$ (b), par priorité de tâche, $M = 5$, $N = 2$	63
3.9	Délai de traitement total : $T^{10,s}$ (a) et $T^{10,s,*}$ (b), par priorité de tâche, $M = 5$, $N = 2$	64
3.10	Délai de traitement total : $T^{10,s}$ (a) et $T^{10,s,*}$ (b), $s = CAVTOMECE$	65
3.11	Taux de succès des tâches délestées, $M = 5$, $N = 2$	66
3.12	Délai de traitement total : $T^{10,s}$ (a) et $T^{10*,s}$ (b), par priorité de tâche, $M = 20$, $N = 5$, avec files d'attente.	67
4.1	Visualisation d'échanges V2X.	74
4.2	Structure protocolaire de la technologie ITS-G5.	76
4.3	Modes de communication 3 et 4 du C-V2X.	79
4.4	Aperçu des communications V2X sur les RAT évoqués.	81
4.5	Chronologie des protocoles de communication V2X.	83
5.1	Nissan Micra K11E (2000) équipée d'une <i>On-Board Unit</i> (OBU).	96
5.2	Cartographie du SNR sur le circuit	97
5.3	Principe de la cartographie de la RSU.	99
5.4	Structure du contenu de grille, case et générateur.	101

5.5	Vue du dessus	106
5.6	Matériel d'expérimentation	106
5.7	Comparaison des zones de couverture	107
5.8	Cartographie des lieux de VHO	109
5.9	Évolution des latences paquet sur la durée d'un tour de circuit	110
5.10	Distribution des latences paquet, par solution	111

LISTE DES TABLEAUX

2.1	RÉSUMÉ DES KPI DES PARADIGMES CLOUD ET EDGE COMPUTING	24
2.2	Comparaison des solutions MEC/VEC	33
3.1	Paramètres du système	41
3.2	Paramètres de simulation	60
4.1	Comparaison des caractéristiques 802.11p et C-V2X	93
5.1	Paramètres de la campagne d'expérimentation	104
5.2	Évaluation de performances, par <i>Radio Access Technology</i> (RAT), par solution	108

GLOSSAIRE

5GAA *5G Automotive Association.*

ADAS *Advanced Driver-Assistance Systems.*

AP *Access Point.*

BBU *Base Band Unit.*

BER *Bit Error Rate.*

BLER *Block Error Rate.*

BS *Base Station.*

BWP *Bandwidth Portions.*

CAM *Cooperative Awareness Message.*

CBS *Coding-Based Scheduling.*

CC *Cloud Computing.*

C-RAN *Cloud-RAN.*

CSMA-CA *Carrier Sense Multiple Access - Collision Avoidance.*

C-V2X *Cellular Vehicle-to-Everything.*

D2D *Device-to-Device.*

DCM *Dual Carrier Modulation.*

DENM *Decentralized Environmental Notification Message.*

DL *Downlink.*

DMRS *Demodulation Reference Signals.*

DSRC *Dedicated Short-Range Communications.*

EC *Edge Computing.*

EDCA *Enhanced Distributed Channel Access.*

eMBB *enhanced Mobile Broadband.*

FC *Fog Computing.*

- FCFS** *First Come First Serve.*
- GLOSA** *Green Light Optimal Speed Advice.*
- GNSS** *Global Navigation Satellite System.*
- HARQ** *Hybrid Automatic Repeat reQuest.*
- HHO** *Horizontal Handover.*
- IMU** *Inertial Measurement Unit.*
- IoT** *Internet of Things.*
- IoV** *Internet of Vehicles.*
- KPI** *Key Performance Indicators.*
- LiDAR** *Light Detection And Ranging.*
- LOS** *Line of Sight.*
- MAC** *Medium Access Control.*
- MCC** *Mobile Cloud Computing.*
- MEC** *Multi-access Edge Computing.*
- MIMO** *Multiple-Input Multiple-Output.*
- ML** *Machine-Learning.*
- MLME** *MAC Layer Management Entity.*
- MMC** *Micro-Macro-Clouds.*
- multi-RAT** *multiple Radio Access Technology.*
- NACK** *Negative Acknowledgements.*
- NFV** *Network Function Virtualization.*
- NLOS** *Non Line of Sight.*
- NR** *New Radio.*
- NSA** *Non-Standalone.*
- OBU** *On-Board Unit.*
- OFDM** *Orthogonal Frequency Division Multiplexing.*
- OFDMA** *Orthogonal Frequency Division Multiple Access.*
- PDR** *Packet Delivery Ratio.*
- PLME** *Physical Layer Management Entity.*
- QoE** *Quality of Experience.*

- QoS** *Quality of Service.*
- RACH** *Random Access Channel.*
- RAN** *Radio Access Network.*
- RAT** *Radio Access Technology.*
- RRH** *Remote Radio Head.*
- RSSI** *Received Signal Strength Indicator.*
- RSU** *Road-Side Unit.*
- RTK** *Real Time Kinematic.*
- SA** *Standalone.*
- SB-SPS** *Sensing-Based Semi-Persistent Scheduling.*
- SC-FDMA** *Single Carrier - Frequency Division Multiple Access.*
- SDN** *Software-Defined Networking.*
- SL** *Sidelink.*
- SNR** *Signal-to-Noise Ratio.*
- UDP** *User Datagram Protocol.*
- UE** *User Equipment.*
- UL** *Uplink.*
- URLLC** *Ultra Reliable Low Latency Communications.*
- V2I** *Vehicle-to-Infrastructure.*
- V2N** *Vehicle-to-Network.*
- V2P** *Vehicle-to-Pedestrian.*
- V2V** *Vehicle-to-Vehicle.*
- V2X** *Vehicle-to-Everything.*
- VAC** *Véhicule Autonome Connecté.*
- VCC** *Vehicular Cloud Computing.*
- VEC** *Vehicular Edge Computing.*
- VFC** *Vehicular Fog Computing.*
- VHO** *Vertical Handover.*
- VIS** *V2X Information Service.*
- VNF** *Virtualized Network Function.*
- WAVE** *Wireless Access in Vehicular Environments.*
- WLAN** *Wireless Local Area Network.*
- WME** *WAVE Management Entity.*
- WSMP** *WAVE Short Message Protocol.*

INTRODUCTION

1.1 Contexte

1.1.1 L'automobile dans notre quotidien

Depuis l'arrivée de la voiture de série au début du XXème siècle, les automobiles n'ont cessé d'affirmer leur place prédominante dans notre quotidien, facilitant grandement la mobilité de chacun. Bien que sa fonction se définisse simplement par le transport motorisé d'un petit nombre de personnes et éventuellement de leurs bagages, l'usage de la voiture est grandement diversifié. La voiture nous accompagne au travail, vers les commerces, en vacances, sur le lieu de nos activités sociales ou de divertissement, et peut même représenter une fin en soi pour les sports automobiles. La démocratisation de la voiture, ainsi que la sous-jacente évolution de nos moyens de transports, s'inscrivent dans la recherche profondément humaine de liberté par le mouvement, déjà explorée de manière similaire avec la locomotive à vapeur au début du XIXème siècle.

En effet, avec le succès de la Ford T et son nouveau modèle de production à la chaîne, des prix particulièrement bas permirent à une toute nouvelle catégorie sociale d'accéder à l'automobile dès les années 1910 [1]. Les véhicules plus confortables et performants demeuraient l'apanage des classes sociales les plus aisées. Cependant, une large portion de la population bénéficiait désormais d'un moyen de locomotion personnel. En l'espace d'un siècle, les immenses progrès technologiques réalisés ont sécurisé nos véhicules, tout comme l'évolution de nos infrastructures urbaines a grandement facilité leur usage. Aujourd'hui, la vaste majorité des foyers français, 84,1 %, est équipée d'au moins une voiture [2]. De la même manière, l'automobile est devenue de plus en plus attrayante pour une variété grandissante d'utilisateurs, grâce à d'immenses progrès ergonomiques et un large panel d'équipements améliorant le confort à bord.

1.1.2 Les équipements des automobiles modernes

À mesure que les ménages se sont équipés de voitures, les profils des utilisateurs se sont diversifiés, appelant de multiples types d'équipements supplémentaires. Bien que ces derniers ne soient en rien nécessaires au principe de fonctionnement de l'automobile, il est indéniable que la présence d'équipements optionnels dans nos véhicules est paradoxalement devenue inévitable. Ils permettent de remplir de très nombreuses fonctionnalités allant du confort dans l'habitacle au stationnement autonome. Dans ce dernier cas, ainsi que dans ceux liés à la conduite autonome, un simple dispositif ne suffit pas : une multitude de capteurs doit être intégrée afin de permettre au véhicule de se positionner dans son environnement. Nous différencions donc les commandes, qui actionnent les fonctionnalités, et les capteurs, qui rassemblent des informations.

Si l'intégration de fonctionnalités supplémentaires progresse à grande vitesse dans le domaine automobile, c'est en raison de la présence de multiples facteurs favorisant leur développement. En premier lieu, nos véhicules modernes possèdent déjà un ordinateur embarqué. Ce dernier n'est pour le moment pas capable d'effectuer des tâches de calcul très complexes, mais nous estimons que le développement de nouvelles applications motiverait le renforcement des capacités de calcul. En outre, l'espace disponible est suffisamment vaste pour intégrer de nouveaux éléments. Enfin, l'utilisateur est actuellement contraint d'organiser son temps autour de son expérience automobile : une fois au volant, il lui est impossible de relâcher son attention tant que sa destination n'est pas atteinte. Se pose alors la question du stationnement, du trajet retour, et ainsi de suite. La voiture personnelle représente à la fois une grande liberté de mouvement, mais apporte également son lot de contraintes, auxquelles l'autonomie souhaite apporter des solutions.

1.1.3 L'autonomie intégrée aux véhicules

Nous pouvons retracer les origines du véhicule radiocommandé à sa première expérimentation en 1925, et celles du véhicule autonome à 1958. Le premier fut un prototype sans opérateur à bord, dont la démonstration dans les rues de New York s'arrêta net suite à de nombreuses collisions [3]. Le second intégrait des capteurs à bord capables de percevoir le courant passant dans un fil incorporé à la route, adaptant sa direction selon le courant perçu [4]. Depuis ces premiers balbutiements, de nouvelles solutions furent explorées afin d'approcher d'une solution efficace et fiable. Les contraintes de la route étant nombreuses, les innovations du véhicule autonome sont rapidement devenues atomiques, individualisées, de sorte à réduire leurs risques. La conception de la voiture du futur évolua d'un véhicule dont la conduite est autonome (l'image de la *self-driving car*) à un véhicule intégrant de plus en plus de fonctionnalités annexes rendues autonomes (ou "autonomisées"), et ne s'orientant que progressivement vers la conduite.

Des protocoles liés aux équipements

L'autonomie apportée par les véhicules modernes est étroitement liée aux équipements fournis à bord. La diversité des tâches que l'on peut déléguer au véhicule commence par des actions simples telles qu'allumer les feux, puis s'étend à des actions complexes comme tourner le volant. Cependant, il semble clair que ces deux tâches ne nécessitent pas les mêmes informations de contexte, ni la même complexité calculatoire afin d'atteindre un résultat fiable.

Nous pouvons dès lors différencier le type d'actes autonomes qu'un véhicule peut prendre en charge en les associant à un besoin de précision d'informations de contexte.

- **Actions binaires.** Lorsqu'une variable environnementale franchit un seuil, le véhicule agit à la place du conducteur en activant ou désactivant une fonctionnalité. Une baisse de luminosité au-delà d'une limite fixée déclenche l'allumage des feux, la détection d'une quantité suffisante d'eau sur le pare-brise actionne les essuie-glaces. Ces actions ne nécessitent pas d'information supplémentaire, et le traitement se fait de manière immédiate.
- **Actions graduelles sur un paramètre du mouvement.** Généralement associées aux situations d'urgence, ces actions concernent les commandes sur lesquelles le véhicule peut agir de manière autonome afin de corriger le mouvement prévu par le conducteur, en jouant sur l'un de ses paramètres (direction, vitesse, accélération). La correction de franchissement de ligne ou le freinage d'urgence font partie de ces actions. Elles nécessitent une granularité d'informations plus fine que les précédentes, provenant notamment des caméras ou de lidars, mais l'intelligence nécessaire reste modérée par l'absence de méthode d'anticipation. En effet, les situations concernées n'étant pas particulièrement complexes, leur détection se fait sur l'instant, au sein de l'ordinateur embarqué.
- **Actions graduelles simultanées.** Ces actions sont les plus proches de la perception publique d'un véhicule autonome. Elles permettent au véhicule d'agir sur plusieurs paramètres de son mouvement, en réponse à une grande variété de situations, incluant l'anticipation d'événements imprévus. Afin de permettre à un véhicule de se déplacer de manière véritablement autonome, il est primordial de franchir la barrière de son espace personnel, afin de l'inscrire dans le contexte d'un trafic routier changeant, parfois imprévisible. Ainsi, les équipements embarqués ne suffisent plus, tant pour des raisons de limitations de puissance de calcul que de besoins de communication avec l'extérieur.

1.1.4 Les usages des véhicules autonomes dans les transports urbains

Dès les travaux préliminaires visant l'autonomie automobile au début des années 2010, il fut établi que l'erreur humaine était responsable de la majorité des accidents de la route. Bien que la source régulièrement citée en France soit ancienne, les données américaines récentes semblent corroborer le fait qu'un parc automobile autonome permettrait une réduction de 90 % des accidents de la route [5], [6]. De même, les véhicules autonomes pourraient participer à une réduction des émissions de gaz à effet de serre allant jusqu'à 60 % [7].

Certaines solutions apportant un premier niveau d'autonomie sont déjà déployées dans de nombreuses municipalités, tandis que d'autres restent à étudier en vue de leur utilisation future. Inspirés des travaux de la *5G Automotive Association (5GAA)* [8], [9], nous classons les différentes applications urbaines liées aux véhicules autonomes connectés selon cinq catégories :

- **Protection.** Cette catégorie concerne la surveillance et la préservation des biens matériels, allant du véhicule lui-même à ce qu'il contient. Elle comprend entre autres les solutions de contrôle d'accès, de suivi à distance de la position et de verrouillage à distance [10]-[13].
- **Sécurité routière.** Concentrée sur la réduction du risque d'accidents graves, cette catégorie comprend les mécanismes de signalisation et d'assistance, comme l'alerte à un incident survenu en amont de l'itinéraire, l'assistance à la navigation, ou encore la signalisation de véhicules d'intervention [14], [15].
- **Conduite autonome et gestion de flotte.** Cette catégorie propose une amélioration du comportement des véhicules seuls ou en petits groupes. Elle comprend les solutions d'aide au dépassement, de synchronisation des usagers d'une intersection, ou de partage d'itinéraire, ce qui se révèle particulièrement intéressant pour les véhicules professionnels. Les applications de gestion de flotte, ou *platooning*, sont étudiées depuis les années 1990 pour leur intérêt sur les longs trajets de poids lourds, notamment pour l'aérodynamisme des véhicules et la réduction de l'impact environnemental. Leur déploiement à large échelle est annoncé pour les prochaines années [16]-[18], en particulier avec l'intégration future des solutions 5G *New Radio (NR)* aux solutions automobiles [19].
- **Fluidification de trafic.** Ce groupement d'applications agit sur deux niveaux : les usagers et l'infrastructure routière. En effet, à l'échelle des véhicules, l'optimisation d'itinéraire en temps réel et l'anticipation de ralentissements permet de limiter les freinages brutaux ainsi que les fortes accélérations. Un trafic fluidifié réduit la consommation, les émissions et le risque d'accidents. De plus, au niveau de l'infrastructure, le comportement des éléments de la route peut également s'adapter au contexte d'usage :

un feu de circulation peut par exemple adapter sa durée de passage au vert en fonction de l'état de congestion d'une route, tout comme un véhicule peut lui-même recevoir des recommandations de vitesse afin de passer au vert au prochain carrefour. Cette dernière application, *Green Light Optimal Speed Advice (GLOSA)*, est également explorée avec des déploiements expérimentaux déjà autorisés. [20]-[23].

- **Facilité d'accès et d'usage.** Les équipements supplémentaires à bord des véhicules, notamment dotés de moyens de communication, représentent bien sûr un confort supplémentaire pour l'utilisateur, pour lequel le ressenti de l'utilisation de son véhicule s'améliore. Pour les véhicules particuliers, la conduite à distance et l'assistance au stationnement en font partie. Pour les applications de groupe, le *geofencing* des transports en commun a déjà fait ses preuves pour la sécurité et la facilité d'usage qu'il apporte, de par la présence de systèmes de géolocalisation à bord et en bordure de route [24], [25].

Les bénéfices de ces applications sont donc nombreux, pour une multitude d'utilisateurs allant des particuliers, aux professionnels de la route, ou aux personnes à mobilité réduite. Cependant, leur développement est fortement ralenti par les limitations techniques des véhicules modernes, qui seront probablement capables dans le futur d'engendrer des volumes de données de perception considérables par l'ajout de capteurs mais incapables de les traiter de manière satisfaisante en interne. Ce problème est actuellement rencontré avec les véhicules purement autonomes, pour lesquels les cas d'usage sont extrêmement restreints et la gestion des cas limites quasi inexistante, ne leur permettant en aucun cas de se passer d'un opérateur humain présent à bord [26]. Il existe donc un fort besoin d'architectures nouvelles, permettant aux usagers de se délester de leurs tâches de calcul via des serveurs externes, par le biais de technologies de communication adaptées aux contraintes d'un usage fortement mobile. Le véhicule autonome connecté s'inscrit donc dans un tout, fondé sur une architecture de communication et de calcul adaptée à des besoins particulièrement demandeurs.

1.2 Motivations

Au delà du véhicule autonome, le marché de l'autonomie véhiculaire est en pleine expansion, avec toujours plus d'équipements et de capteurs intégrés dans le but d'améliorer confort et sécurité. Cependant, nous remarquons qu'une importante portion des technologies véhiculaires peine à être explorée : celle reposant sur les interactions entre les véhicules et leur environnement. Un véhicule autonome ne saurait se contenter de se positionner vaguement par rapport à son entourage, il lui est nécessaire d'échanger des informations afin de préciser son mouvement et s'inscrire dans le trafic routier général. De multiples solutions ont été explorées, mais rencontrent de nombreuses limites liées au déploiement de l'infrastructure nécessaire, aux niveaux de performance requis ou encore au coût supplémentaire induit.

Pour ces raisons, l'autonomie des véhicules modernes semble stagner à une assistance à la conduite fortement contrainte, et y semble cantonnée tant qu'aucune architecture de communication n'est réellement déployée à grande échelle.

L'action autOCampus, du projet VILAGIL porté par Toulouse Métropole, a pour but d'explorer des solutions innovantes de mobilité des biens et des personnes. Le véhicule autonome s'inscrit dans cette démarche. Toutefois, le déploiement d'une architecture de communication spécifique est impératif afin de faciliter et d'approfondir les échanges entre ce véhicule et son environnement. Au sein de cette thèse, nous étudions les nombreuses conditions nécessaires au succès de cette architecture, puis nous recherchons une solution capable de satisfaire une grande variété de services, pour une grande diversité d'utilisateurs et dans un contexte urbain.

Afin d'atteindre nos objectifs, dans cette thèse, nous nous intéressons aux deux parties fondatrices d'une architecture prenant en charge des services véhiculaires, en étudiant pour chacune un problème conceptuel spécifique et emblématique. En effet, nous avons remarqué que les besoins applicatifs liés à l'automobile autonome et connectée se heurtent actuellement à deux freins : la limitation des capacités de calcul embarquées [27] et les contraintes de performance des communications extra-véhiculaires [28]. En réponse, nous choisissons d'étudier deux problèmes fortement corrélés à ces freins :

1. Gestion de quantités massives de données entrantes ;
2. Fiabilisation des communications.

Nous proposons donc deux solutions pour des architectures de communication satisfaisant les contraintes d'applications véhiculaires en vue de leur déploiement futur. La première est un système d'optimisation de délestage de tâches de calcul vers des serveurs de bord de route (solution de calcul), la seconde est un mécanisme de bascule proactive de technologie radio dans un contexte de forte mobilité (solution de communication réseau).

1.3 Méthodologie

1.3.1 Un œil sur le concret

Dans cette thèse, nous explorons différentes solutions d'architectures de communication véhiculaire. Cependant nous complétons cette approche souvent conceptuelle par une étude de leurs conditions d'application réelle. Tout au long de cette thèse, nous gardons à l'esprit les besoins techniques nécessaires à une mise en œuvre pratique ainsi que leurs implications sur les performances finales (limites des capacités de calcul, surcharge, temps de propagation, temps de réponse etc.). De manière générale, nous étudions des problèmes issus de situations réelles, pour lesquels notre analyse nous fait parvenir à des solutions heuristiques ou algorithmiques, que nous affinons et évaluons dans une situation expérimentale ancrée dans le réel, avec des équipements réels quand cela est possible. Lorsque l'analyse par simulation

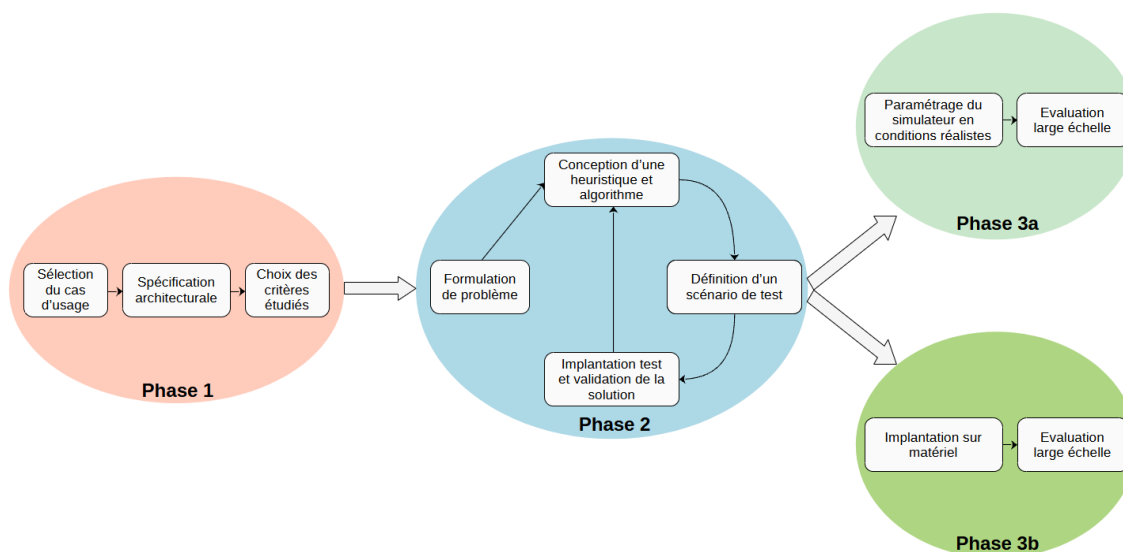


FIGURE 1.1 Méthodologie suivie pendant la thèse

est inévitable, nous la menons de sorte à prendre en compte le plus grand nombre des phénomènes matériels rencontrés si une transposition réelle devait se produire. La figure 1.1 présente un résumé synthétique de la méthodologie que nous suivons.

1.3.2 Différents prismes technologiques abordés lors du développement

Les solutions évoquées sont simultanément fondées sur trois piliers. Les capteurs embarqués relèvent des données liées à l'environnement et au véhicule lui-même. Les architectures de calcul servent à combiner, traiter et exploiter un ensemble de données massives. Enfin, les technologies de communication permettent l'échange de données et de tâches de calcul entre de multiples acteurs du système. Notre recherche se positionne donc à la rencontre de plusieurs disciplines de l'informatique. Nous estimons la présence de capteurs comme un pré-requis aux solutions véhiculaires autonomes connectées, afin de solliciter des connaissances issues des domaines des systèmes distribués, ainsi que des réseaux Internet, mobiles et véhiculaires. Chacune de ces disciplines peut être explorée en vue de concevoir une architecture répondant aux fortes contraintes des applications véhiculaires.

1.4 Contributions et plan du mémoire

Cette thèse a pour objectifs la définition et l'évaluation d'une architecture de communication véhiculaire, afin de faciliter l'insertion de véhicules autonomes dans une infrastructure réseau hétérogène. Comme énoncé précédemment, ces architectures s'articulent autour de deux axes principaux : la gestion de calcul et la communication. Les travaux de cette thèse sont donc présentés en deux parties, la première présentant la première contribution centrée sur l'optimisation de délestage de calcul par les véhicules connectés ; la deuxième se

focalisant sur la stabilisation de connectivité réseau dans un contexte de forte perturbation des liens physiques : un quartier universitaire.

1.4.1 CAVTOMECA : faciliter le délestage de tâches de calcul véhiculaire

Dans notre première contribution, nous nous intéressons aux grandes quantités de données engendrées par les capteurs, nécessitant un traitement externalisé respectant de fortes contraintes temporelles. En effet, afin de permettre au véhicule de participer à la vie du système dans lequel il évolue, les informations engendrées par ses différents capteurs doivent être traitées en vue de leur interprétation et de leur partage éventuel. Nous proposons CAVTOMECA, une solution d'optimisation de délestage de calcul vers des serveurs proches de l'émetteur, exploitant des messages de signalisation sur la quantité de ressources disponibles. Notre objectif est de faciliter l'intégration d'applications nécessitant le traitement de données massives, comme la reconnaissance d'images.

Afin d'y parvenir, notre système fournit à chaque véhicule un mécanisme de sélection de la destination vers laquelle délester leurs tâches. Ce mécanisme s'appuie sur un ensemble de paramètres : la distance à la destination, la charge ajoutée par la tâche à délester, les ressources de calcul disponibles à la destination. Afin de conserver une vision actualisée de ce dernier paramètre, les véhicules reçoivent fréquemment des messages de signalisation transmis par les serveurs de bord de route, comprenant dans un premier temps la quantité de ressources disponibles puis l'état des files d'attente des tâches à traiter. Cette solution est évaluée par simulation sur l'outil OMNET++, selon des scénarios les plus proches possibles de conditions réelles.

1.4.2 CERES-V2X : Stabiliser la connectivité véhiculaire dans un contexte urbain

Dans notre seconde contribution, nous étendons le contexte précédemment étudié afin de nous concentrer sur les phénomènes physiques pouvant perturber la transmission des tâches à traiter. En effet, le déport de calcul ne peut avoir lieu que si le processus de communication est validé dans les deux sens. Cependant, dans des conditions réelles de trafic routier, de nombreux phénomènes peuvent perturber les communications radio : entrée dans un tunnel, sortie de zone de couverture, obstacle temporaire ou permanent sur la ligne de vue entre émetteur et récepteur, présence de bâtiments. Nous proposons donc une solution de stabilisation de la connectivité réseau dans un tel contexte, s'appuyant sur la nature hétérogène des réseaux modernes ainsi que sur la qualité *multiple Radio Access Technology* (multi-RAT) des véhicules, signifiant que diverses technologies radio peuvent être exploitées, simultanément ou non.

Dans les faits, nous concevons un système de relevé de qualité de service, ou *Quality of Service* (QoS), à partir duquel l'ensemble des messages véhiculaires ayant transité par un point d'accès situé en bordure de route permettent de dresser une cartographie des

performances réseau. Cette cartographie générée à chaque point d'accès est ensuite mise en forme et transmise aux usagers de la route, leur permettant de déterminer, en fonction de leur itinéraire, les points où une bascule d'une technologie radio à une autre est nécessaire. Nous déployons cette solution en conditions réelles, sur du matériel commercial et embarqué dans une navette autonome. Nous procédons ensuite à l'évaluation des performances de cette solution dans des conditions routières et urbaines difficiles : celles d'un campus universitaire.

Première partie

Optimisation de délestage de calcul véhiculaire dans un contexte MEC urbain

Avec l'arrivée des réseaux mobiles à très haut débit, les applications Cloud se sont multipliées. L'accès à Internet depuis son smartphone est devenu particulièrement populaire, et les utilisateurs ont orienté massivement leurs choix d'applications vers des modèles fondés sur le *streaming*, remplaçant l'ancien modèle fondé sur le téléchargement local. Ensemble, Cloud et streaming se démarquent par leur absence de besoin de ressources locales, leur disponibilité et la facilité du passage à l'échelle. Ces propriétés proviennent de la possibilité de solliciter et stocker ses données et d'exécuter des tâches de calcul dans des serveurs distants hébergés par un prestataire, lui-même dimensionnant son offre en fonction des besoins de ses clients. Aujourd'hui, le Cloud fait partie de notre quotidien, accessible partout et à tout moment.

Toutefois, les services Cloud se heurtent à des limites architecturales influant sur leurs performances. La centralisation des données induit un nombre de points de stockage relativement faible, donc une faible efficacité de service pour une grande partie des utilisateurs éloignés de ces points. De plus, cette même centralisation rend les ressources disponibles extrêmement robustes, mais uniformes pour tout type d'utilisateurs. Il demeure pourtant une forte demande pour des applications aux usages variés, notamment avec l'arrivée des réseaux mobiles de nouvelle génération et les forts gains de performance réseau motivant d'autant plus le déchargement de charges de calcul et de stockage, notamment dans les milieux urbains où la 5G est déployée en priorité.

Une solution potentielle se trouve dans le *Multi-access Edge Computing (MEC)*, un paradigme informatique visant à répartir les ressources Cloud à proximité des utilisateurs, dans des serveurs de capacité plus faible mais bien plus nombreux. Cette architecture est par nature personnalisable selon le lieu de son déploiement, de sorte à dimensionner les ressources nécessaires aux besoins locaux et à fournir des services adaptés aux utilisateurs locaux. Un cas d'usage rapidement mis en avant consiste en la mise à disposition de données fréquemment demandées par un ensemble d'utilisateurs au niveau d'un serveur de bordure, là où une architecture Cloud aurait besoin de pousser chaque utilisateur à systématiquement

solliciter les serveurs distants, dégradant la *Quality of Experience (QoE)* [29].

Cependant, on ne saurait en déduire que le MEC soit la solution ultime, vouée à détrôner le Cloud. Tout d'abord, les besoins matériels sont sensiblement différents : dès lors qu'une architecture fortement distribuée doit être déployée, des questions d'alimentation électrique, de protection et de connexion au réseau se posent. De plus, la forte personnalisation des solutions MEC appelle un cadre commun, facilitant la mobilité de secteur en secteur, d'architecture à architecture. Enfin, les faibles quantités de ressources disponibles sur chaque serveur comparativement au Cloud imposent une réflexion profonde quant à l'adaptation de la structure aux divers besoins des utilisateurs, et à leurs fluctuations. La solution se doit de répondre aux besoins d'un petit nombre d'utilisateurs, tout comme elle doit pouvoir encaisser la charge en heure de pointe. Ainsi, les déploiements réels d'architectures MEC se font encore rares, ralentissant le développement de solutions pouvant en bénéficier.

Au sein de ces réflexions sur l'avenir des architectures de calcul, les technologies véhiculaires tiennent une place de choix. Ces équipements, s'étendant des ordinateurs embarqués aux capteurs à haut niveau de précision (caméras, *Light Detection And Ranging (LiDAR)*), sont potentiellement généreux dans les volumes de données engendrés en raison en particulier de leur ressources énergétiques plus importantes que de simples capteurs dispersés dans la nature. Ils bénéficieraient grandement d'une architecture capable de fournir les ressources de traitement et de stockage nécessaires à de multiples cas d'usage, même si la question de la surcharge des serveurs de bordure se pose. De plus, le contexte de forte mobilité soulève des défis nouveaux et inhérents aux besoins des véhicules, tels que la fiabilité, la robustesse et la durée de traitement. Cependant, de nombreuses questions liées à l'intégration de véhicules autonomes connectés restent en suspens, notamment pour la gestion combinée des ressources de bordure, de la surcharge, et de la mobilité. De même, peu de travaux traitent de la connaissance du contexte du côté de l'utilisateur. Après quelques années de tâtonnement, lié à une faible part de véhicules équipés, le secteur des communications véhiculaires reste relativement jeune. Il trouve sa place dans le contexte des réseaux urbains modernes, hétérogènes par nature. En effet, dans nos métropoles, nous pouvons bénéficier simultanément d'une couverture cellulaire LTE et 5G, tout en étant à portée d'un ou plusieurs points d'accès Wi-Fi, pour n'évoquer que ces technologies, tout à fait adaptées à un large panel d'applications mobiles. Cependant, les besoins véhiculaires se démarquent de ceux d'un utilisateur humain, ce qui motive la conception de technologies plus adaptées, s'appuyant sur cette hétérogénéité. Dans cette problématique *multi-RAT* (pour *multiplés technologies d'accès radio*), les technologies de communication sans point d'accès, ou *Device-to-Device (D2D)*, se sont démarquées par leurs performances particulièrement satisfaisantes à courte portée, ainsi que par leur généricité apparente. En effet, véhicules et éléments d'infrastructure peuvent être équipés de sorte à échanger par *D2D*, s'affranchissant d'un saut supplémentaire et diminuant considérablement le temps d'échange de données de contexte, de tâches de calcul, ou de résultats de traitement. Si un

tel mode de communication n'est temporairement pas exploitable, alors une bascule peut s'opérer vers les technologies radio classiques disponibles. Les nouvelles technologies de communication s'inscrivent profondément dans le paradigme MEC urbain, ce qui motive l'exploration d'architectures couplant efficacement ces deux branches.

Dans cette partie, nous présentons CAVTOMECE, une solution de délestage de calculs véhiculaires adaptant la sélection de la cible aux ressources disponibles en bordure de route, afin de réduire le temps de traitement des tâches et d'augmenter la proportion de tâches terminées en accord avec leurs critères. Cette solution qui tient compte de son contexte est adaptée aux besoins de véhicules qui peuvent nécessiter de particulièrement fortes charges de calcul, ce qui la positionne en accord avec un grand nombre de cas d'usage urbains.

Organisation de la partie

Cette partie est constituée de deux chapitres. Dans le premier chapitre, nous dressons un état de l'art des systèmes de délestage de calcul MEC, en nous focalisant sur les mécanismes de gestion des tâches. Nous présentons les différentes architectures utilisées, les choix algorithmiques réalisés, et leurs implications sur le déploiement de véhicules connectés. Nous constatons que ces différentes propositions peuvent être regroupées en grandes catégories, nous en relevons les principales caractéristiques puis nous positionnons notre contribution au sein de ces catégories.

Dans le deuxième chapitre, nous présentons CAVTOMECE, notre solution architecturale et algorithmique pour accélérer le calcul des tâches véhiculaires délestées. Nous dressons tout d'abord une étude du contexte urbain et de ses conséquences sur les durées de délestage des tâches, puis nous proposons notre solution exploitant des messages de signalisation pour contrebalancer les comportements imprévisibles, réduire les latences de délestage et le temps passé sur le serveur de bordure. Enfin, nous validons les contributions de notre solution par un ensemble de simulations menées sur de nombreuses valeurs de paramètres, dans un environnement proche du réel.

ETAT DE L'ART

2.1 Les niveaux d'autonomie de conduite

La notion d'autonomie précède celle de connectivité dans le domaine automobile. Nous pouvons associer les origines de l'autonomie véhiculaire à l'apparition des premiers équipements d'assistance à la conduite. De l'allumage automatique des feux à l'assistance au stationnement, chaque nouvelle génération de véhicules apporte des équipements supplémentaires visant à faciliter l'expérience de conduite de l'utilisateur. Cependant, les degrés d'autonomie apportés par un régulateur de vitesse ou par un freinage d'urgence ne sont pas identiques. La classification en six niveaux, aujourd'hui grandement répandue dans l'industrie automobile, apparut initialement en 2016 afin de différencier la capacité des véhicules commerciaux à exécuter des tâches de conduite de manière autonome [30]. Les six niveaux d'autonomie sont présentés comme suit :

- Niveau 0** – Aucune autonomie. Le conducteur gère l'intégralité des fonctionnalités de conduite telles que l'accélération, le freinage, la direction.
- Niveau 1** – Assistance à la conduite. Le véhicule est capable d'opérer un élément de la conduite. Le régulateur de vitesse dynamique remplit cette condition : le conducteur conserve la gestion de la direction et du freinage. Le véhicule ne fait que se maintenir à distance fixe de son prédécesseur.
- Niveau 2** – Autonomie partielle, incluant les *Advanced Driver-Assistance Systems (ADAS)*, systèmes d'assistance avancée à la conduite. Le véhicule prend à la fois en charge les fonctions de direction, d'accélération et de freinage, mais le conducteur demeure en position de conduite, les yeux sur la route, pour reprendre la main si nécessaire. L'assistance au stationnement entre dans cette catégorie, au même titre que les très médiatisées technologies de pilote automatique d'un certain constructeur de véhicules électriques.
- Niveau 3** – Autonomie conditionnelle. Le conducteur conserve sa position de conduite, mais peut

se permettre de quitter la route des yeux tout en restant attentif pour agir en cas d'éventuels imprévus sur la route ou erreurs de jugement de la part du véhicule. Ce niveau se différencie surtout du précédent d'un point de vue technologique, de part la notion de conscience de son environnement. La présence de capteurs à haute précision devient indispensable.

Niveau 4 – Forte autonomie. Le véhicule possède la capacité de corriger les imprévus : le conducteur peut donc quitter sa position de conduite et y revenir dans de rares cas non pris en charge. Ce niveau d'autonomie est adapté aux routes lisibles et aux vitesses limitées. Ainsi, le point faible de ce niveau d'autonomie est sa limitation à une zone géographique prédéfinie, nommée *geofencing* ou géorepérage en français. Ce niveau d'autonomie concerne donc principalement les technologies de transport en commun, le temps que les infrastructures urbaines et la législation évoluent.

Niveau 5 – Autonomie totale. Il s'agit de la généralisation du niveau 4 à toutes les situations de conduite, s'affranchissant donc de géorepérage et se destinant à tout type d'usage.

2.2 Les capteurs véhiculaires embarqués

L'autonomie d'un véhicule, quel que soit son degré, dépend en premier lieu des informations contextuelles mises à disposition. Ainsi, de nombreux capteurs peuvent être embarqués afin de permettre au véhicule d'alimenter sa connaissance du milieu [31]. Certains capteurs ne sont encore que peu répandus, tandis que d'autres sont désormais indispensables à bord. Nous proposons de décrire les principaux capteurs utilisés dans le domaine automobile.

2.2.1 Les capteurs de position

Le récepteur GPS

Les capteurs de localisation GPS, dénomination utilisée par abus de langage pour tous les systèmes *Global Navigation Satellite System* (GNSS), font partie intégrante de nos véhicules, qu'ils soient de série ou nomades. En recevant périodiquement des signaux provenant d'un réseau de satellites disposés sur de multiples plans orbitaux, il est possible d'obtenir une estimation de sa position en trois dimensions avec une précision de l'ordre du mètre [32], cette précision pouvant fluctuer et être améliorée [33]. Bien que ces données très concises soient faciles d'accès et utiles pour un ensemble de cas d'usage liés à la navigation et à la sécurité, les systèmes de pilotage autonome nécessitent une précision bien plus forte, de l'ordre du centimètre.

Le récepteur RTK

Le récepteur *Real Time Kinematic* (RTK) permet de recevoir en temps réel des données de correction de l'estimation de position GPS à partir d'une antenne au sol. Grâce à ce

module, il est possible d'atteindre une précision centimétrique sur le relevé de localisation. Ce module est donc favorable aux véhicules à haut niveau d'autonomie [34].

2.2.2 Les capteurs de comportement routier

La centrale à inertie

La centrale à inertie, ou unité de mesure inertielle pour *Inertial Measurement Unit (IMU)*, permet d'estimer les informations de mouvement d'un véhicule telles que la vitesse et les trois angles constitutifs de son orientation (roulis, cap et tangage). L'IMU comporte trois gyromètres et trois accéléromètres, ce qui lui permet d'intégrer l'accélération et la vitesse angulaire détectées afin d'obtenir les données susmentionnées sans recours à un équipement extérieur. De plus, par comparaison des relevés instantanés au relevé de départ, il est possible d'estimer la position du véhicule [35]. Ainsi, la combinaison des données IMU et GPS/RTK offre au véhicule une précision et une robustesse essentielles à la validité des paramètres instantanés estimés. De nos jours, l'IMU concerne majoritairement les applications aéronautiques et militaires, mais est indispensable aux véhicules à haut niveau d'autonomie.

L'odomètre

Appelé compteur kilométrique dans le langage courant, l'odomètre est l'outil de mesure de la distance parcourue par un véhicule. Là où l'odomètre n'est qu'un outil d'information pour un conducteur humain, il est un élément essentiel à l'évaluation du positionnement d'un véhicule autonome. Alors que la plupart des autres capteurs fournissent une information instantanée, l'odomètre fournit une information dépendante du comportement routier depuis le départ, donc de la globalité du contexte de fonctionnement. Par la connaissance de la distance parcourue, un véhicule peut compléter les informations instantanées relevées par les capteurs de position à des fins de redondance, de sécurisation ou d'optimisation d'itinéraire.

2.2.3 Les capteurs d'environnement

Les caméras

Le sens humain le plus utilisé, au quotidien comme en voiture, demeure la vue. Il semble cohérent d'appliquer ce principe à la perception de l'environnement du véhicule lui-même, au moyen de caméras, ce pourquoi ces dernières font désormais partie du paysage automobile au sein d'applications d'aide au stationnement (caméra de recul) ou de sécurité (caméra-témoin de circulation ou *dash-cam*). Cependant, les progrès technologiques récents ont permis d'aboutir à des captations vidéo grand angle à haute vitesse et ultra-haute définition à un coût maîtrisé. Ainsi, l'apport des caméras à l'autonomie véhiculaire comprend

la perception robuste et précise de l'environnement proche, ainsi qu'une interprétation simplifiée des données engendrées, en comparaison de données brutes provenant d'autres types de capteurs et qui nécessitent un traitement plus poussé en aval afin d'obtenir une interprétation. Cependant, afin de répondre aux besoins des futurs véhicules autonomes, les images captées se doivent de répondre à de stricts critères de qualité [36]. Ainsi, les données produites chaque seconde par ces caméras sont volumineuses, allant, selon les spécifications, de la dizaine de mégaoctets (pour une caméra haute définition à 30 images par seconde) au gigaoctet (pour une caméra 4K à 60 images par seconde). Le stockage et le traitement de ces données induit une charge significative sur le système.

Les radars

Issus de l'acronyme Radio Detection and Ranging signifiant détection et évaluation de distance par radio, les radars font désormais partie du langage commun et sont très répandus. Le système radar propage des ondes radio aux alentours et capte les signaux réfléchis par les obstacles, aboutissant à une cartographie d'ensemble après déduction de la distance à partir de la durée de retour. Ces instruments sont peu coûteux et peuvent bénéficier d'une longue portée de par la faible réflectivité des ondes radio : des distances de l'ordre du kilomètre sont atteignables en usage terrestre, en ligne droite. Dans les courbes, les performances se dégradent en raison de la faible précision angulaire du radar, limitant à deux dimensions la cartographie obtenue. Ainsi, les véhicules commerciaux ont tout d'abord uniquement intégré ces équipements à des fins d'assistance au stationnement, avec les radars de recul. Cependant, afin d'augmenter le potentiel de perception des véhicules modernes, les capteurs radars bénéficient d'un intérêt renouvelé, pour des applications de sécurisation comme l'avertissement d'angle mort ou le freinage d'urgence automatisé [37]. Certains constructeurs souhaitent également exploiter leur potentiel cartographique, bien que l'usage de capteurs complémentaires comme ceux évoqués précédemment soit préconisé afin de contrebalancer leur faiblesses dans ces contextes.

Les LiDAR

Présentés comme une évolution en trois dimensions du radar, les instruments **LiDAR** exploitent le même principe par le biais de la technologie laser. Leurs ondes appartiennent aux domaines infrarouge, visible ou ultraviolet proche, et sont émises à haute fréquence (jusqu'à 150 000 faisceaux par seconde). Les faisceaux émis sont plus directionnels que les ondes radio, et la quantité d'ondes émises est sensiblement plus élevée qu'avec un radar. Ainsi, la réflexion des obstacles se produit sur de plus nombreux points, aboutissant à une cartographie à haute précision en trois dimensions. Ces capteurs sont bien plus coûteux que leurs homologues radio, sensibles aux conditions météorologiques difficiles et engendrent une grande quantité de données (en moyenne 426 Ko chaque dixième de seconde [38]).

Le traitement en aval de ces données massives présente donc un fort potentiel pour la sécurisation des véhicules autonomes.

2.3 Le traitement des données de capteurs

Ainsi, les véhicules récents comme leurs prédécesseurs sont équipés de capteurs, dont la diversité s'étend au fil du temps. À mesure que la précision de ces derniers augmente, leurs exigences de quantités de stockage et de puissance de calcul ne peuvent qu'augmenter, dépassant les capacités embarquées dans l'unité de calcul à bord (ou *On-Board Unit (OBU)*), généralement équipée de stockage flash de l'ordre du gigaoctet et d'un processeur quadruple-cœur 800 MHz [39]-[41]. Le besoin d'externaliser cette charge informatique est manifeste. De même, la gestion de la mobilité est indispensable, ce qui nécessite le déploiement d'une solution robuste et adaptable afin de minimiser la pression sur le réseau dans le contexte d'un marché florissant. Une plus grande quantité d'appareils signifie plus de capteurs, plus de données et plus de tâches à gérer. Cette situation est analogue au paradigme de l'internet des objets (ou *Internet of Things (IoT)*), où une énorme quantité d'équipements peut être regroupée ou distribuée en fonction de l'application [42], [43]. Étendu aux véhicules connectés et aux villes intelligentes, ce système forme l'Internet des véhicules, ou *Internet of Vehicles (IOV)* [44]. Les possibilités sont nombreuses, allant de la conduite autonome à la sécurité routière collaborative. Cependant, de telles nouveautés nécessitent des garanties strictes en termes de performance du réseau, comme la fiabilité, le délai, le débit, voire la sécurité. De multiples variantes d'architecture réseau cherchent à répondre à ces besoins, nous nous proposons de les étudier.

2.3.1 Le cloud computing

Héritées des références de l'Internet actuel réputées pour leurs performances, comme Amazon AWS ou Microsoft Azure, les premières architectures de réseaux véhiculaires étaient fondées sur le *Cloud Computing (CC)* [45]. Le cloud computing est une architecture centrée autour de puissantes ressources physiques regroupées en data centers, permettant un traitement virtualisé indépendant comme le calcul, le stockage et la communication, directement adapté aux besoins des utilisateurs finaux, pour de nombreux types d'applications. Il est connu pour être évolutif, fiable et abordable, mais implique des temps de latence plus importants car ces data centers sont situés loin du système. En effet, au-delà des considérations de technologie radio, un itinéraire de plusieurs centaines de kilomètres entre l'utilisateur et le cloud induit des délais supplémentaires, en particulier liées au routage.

En outre, lorsque les applications mobiles s'appuyant sur le réseau ont gagné en attrait, le cloud a commencé à intégrer des fonctions mobiles, étendant ainsi la gestion des ressources cloud à un environnement mobile dans le cadre d'un nouveau paradigme : le *Mobile Cloud Computing (MCC)* [46]. Dans le MCC, les serveurs cloud sont toujours centralisés à

longue distance des appareils finaux, mais intègrent également le partage des ressources et la gestion de la mobilité afin de répondre aux exigences des utilisateurs mobiles [47]. Appliqué à l'IoV, le MCC s'efforce de satisfaire la QoS requise par une gamme toujours plus large d'applications dédiées aux véhicules, donnant ainsi naissance au *Vehicular Cloud Computing* (VCC) [48]. Cependant, faire transiter des données sur de longues distances et à travers diverses interfaces augmente le délai et la latence, deux paramètres cruciaux pour les communications mobiles [49]. C'est ainsi qu'émergea la question du rapprochement de la puissance de calcul vers l'équipement utilisateur (UE).

2.3.2 L'edge computing

Alors que le MCC a apporté de nouvelles pistes en matière de réseaux véhiculaires, notamment dans le domaine de la mobilité [50], la question de la gestion du temps est restée en suspens. Par conséquent, l'idée de rapprocher le matériel des utilisateurs mobiles est apparue. Ainsi, ces applications très mobiles pourraient bénéficier de ressources informatiques puissantes dans des nœuds distribués plus près de l'emplacement des utilisateurs. C'est précisément ce que permet le paradigme d'*Edge Computing* (EC), qui consiste à déployer des serveurs à un saut des équipements engendrant les données afin de transmettre et traiter les paquets avec des délais beaucoup plus faibles. Comme dans le cas du CC, l'EC a rapidement été étendu en MEC, que l'ETSI décrit grossièrement comme "*tout ce qui n'est pas une architecture de calcul centrée sur les data centers*" [51]. Plus précisément, une architecture edge computing est définie comme fournissant "*un environnement de service informatique accompagné des capacités de cloud computing à la périphérie du réseau mobile, au sein du réseau d'accès radio et à proximité immédiate des utilisateurs mobiles*" [52]. En rapprochant la puissance de calcul de la route dans des serveurs de périphérie (ou edge servers, ou encore edge nodes), nous pouvons aisément déduire une diminution du délai de transmission des paquets, tout en déplaçant les considérations vers la question du débit du réseau véhiculaire.

De nouvelles architectures furent donc conçues en vue de mener à bien cet affranchissement des centres de données. Ces structures rassemblent les ressources informatiques et de réseau des utilisateurs finaux dans des unités de bord de route, ou *Road-Side Unit* (RSU) [53], tandis que les utilisateurs eux-mêmes disposent d'une certaine puissance de traitement dans leur unité embarquée (OBU) [54]. Le principal avantage est la distribution des ressources, ainsi que des applications et des services qui les accompagnent, par rapport à la centralisation du Cloud. Dès lors, nous étudions les deux directions principales qui ont fait l'objet de recherches afin de résoudre le problème des applications sensibles aux délais : le multi-access edge computing (MEC) et le fog computing.

2.3.3 Le multi-access edge computing

Anciennement *mobile edge computing*, le MEC est une déclinaison du paradigme edge computing dans les réseaux mobiles, notamment cellulaires. En effet, les problématiques de routage telles qu'étudiées dans le cadre du CC ne semblent pas tout à fait transposables au MEC, en ceci que les difficultés rencontrées sur le réseau sont différentes : le MEC est plutôt concerné par les risques liés à la forte mobilité et aux bascules fréquentes (handover). La notion de mobilité est donc inhérente à cette architecture, avec une volonté de déployer les serveurs de bordure à proximité des points d'accès au réseau mobile afin de réduire l'utilisation de bande passante et les latences des demandes applicatives. En distribuant les ressources de calcul et de stockage en bordure du réseau mobile, le MEC s'impose comme une solution architecturale crédible pour les applications sensibles temporellement.

Comme indiqué précédemment, l'ETSI a déployé des efforts de définition et de normalisation pour le MEC, en publiant un cadre de démonstration de faisabilité dès 2015 [55]. Le MEC de l'ETSI repose sur un cadre à trois niveaux, illustré à la figure 2.1, la couche supérieure étant le niveau du système, la couche intermédiaire étant le niveau de l'hôte où résident les applications MEC, et les interfaces de communication réseau étant situées au niveau inférieur. Dans ce cadre, la gestion du système du niveau supérieur surveille et agit sur les blocs situés en dessous, qui sont eux-mêmes séparés entre l'hôte MEC et la gestion du niveau de l'hôte MEC. Les interactions entre les niveaux supérieur et intermédiaire ont une incidence sur le comportement de l'hôte MEC, responsable de l'exploitation de la plateforme virtuelle fournissant des ressources de réseau et de calcul aux applications MEC. Enfin, le niveau inférieur permet des connexions réseau multiples, du niveau local au niveau cellulaire, favorisant ainsi les interactions avec un large éventail d'équipements. L'architecture a été examinée plus en détail dans des travaux antérieurs tels que [56]. La principale conclusion est que, lorsqu'elles sont déployées pour un cas d'utilisation donné, les entités MEC peuvent être distribuées par l'intermédiaire de l'infrastructure de réseau elle-même. En effet, comme la gestion du système et l'orchestrateur ont besoin d'une vue globale de l'activité du système, ils peuvent faire partie du réseau central, tandis que les hôtes MEC et leur propre gestionnaire sont déployés à la périphérie du réseau, à proximité des stations de base, pour bénéficier d'une architecture distribuée.

Dans son déploiement physique, le MEC peut aussi se manifester sous la forme d'une architecture à trois couches, comme le montre la figure 2.2(a), allant du réseau central aux dispositifs finaux [57] :

- Sur la couche de cœur se trouvent les serveurs cloud : de grands centres de données dotés de ressources informatiques presque illimitées par rapport aux dispositifs finaux. Ces serveurs centraux sont également connectés à des nuages privés.
- Sur la couche des appareils, on trouve les équipements mobiles et les utilisateurs

finaux, qui sollicitent souvent les ressources du nuage pour répondre aux besoins de diverses applications.

- Entre les deux se trouve la couche "edge-cloud", où sont insérés les dispositifs MEC, également appelés "edge servers" ou même "*cloudlets*". Ils servent d'intermédiaires entre les utilisateurs finaux et les services cloud, en hébergeant eux-mêmes des ressources de calcul analogues à celles disponibles dans le cloud. De par leur conception, ces petits serveurs ont moins de capacité que les serveurs cloud, mais sont toujours capables de gérer des tâches gourmandes en ressources. Les serveurs MEC sont situés à proximité des stations de base cellulaires et des points d'accès au réseau, de manière à réduire la distance entre le fournisseur de ressources et l'utilisateur.

L'architecture MEC contribue à la mise en œuvre d'applications sensibles à la latence, car elle fournit des ressources cloud à la périphérie du réseau mobile. Sans s'appuyer sur le réseau de cœur, les demandes des utilisateurs et les tâches déchargées n'ont qu'à transiter par les deux couches les plus proches de la périphérie du réseau [58], ce qui implique des avantages considérables en termes de délai de communication, et donc une meilleure QoS et une meilleure QoE. Les réseaux de véhicules peuvent bénéficier de ces mesures améliorées, c'est pourquoi l'ETSI a étendu ses normes MEC aux applications *Vehicle-to-Everything* (V2X) potentielles, en établissant des moyens de faciliter l'interopérabilité dans des environnements hautement dynamiques, multi-fournisseurs, hétérogènes et multi-accès [59]. Avec cette norme, le service d'information V2X (ou *V2X Information Service* (VIS)) assure la continuité du service quelle que soit la zone de couverture de l'opérateur, y compris dans le cas d'opérateurs multiples, en exploitant les informations de configuration de PC5, l'interface de communication device-to-device, pour communiquer directement par l'intermédiaire des systèmes MEC au lieu de transmettre les données par l'intermédiaire des serveurs distants. Le VIS réduit considérablement les temps d'attente pour l'échange de données entre les véhicules couverts par différents opérateurs, ce qui représente des améliorations non négligeables en termes de performances et de polyvalence. En outre, la souplesse liée à la possibilité de déployer des *cloudlets* à l'endroit le plus approprié sur la route permet d'adapter les ressources en fonction des services demandés à un endroit donné, ce qui se traduit par des économies pour les opérateurs de réseau mobile.

Lorsque des tâches sont envoyées à la couche périphérique, le point d'accès sert de relais entre l'équipement et le serveur périphérique. Les données parviennent au point d'accès par le biais de la communication cellulaire, puis sont acheminées vers le *cloudlet* au moyen d'une liaison à large bande, généralement optique, afin d'éviter les problèmes de fiabilité sans fil. À la fin du calcul, les résultats sont renvoyés au terminal. En outre, en cas de pénurie de ressources de calcul, les serveurs périphériques peuvent être interconnectés par liaison terrestre à des fins d'équilibrage de la charge. Le même mécanisme s'applique aux stations de base. Enfin, les applications à forte demande de calcul sont toujours envoyées à la couche périphérique, mais elles sont ensuite relayées vers le réseau central par liaison

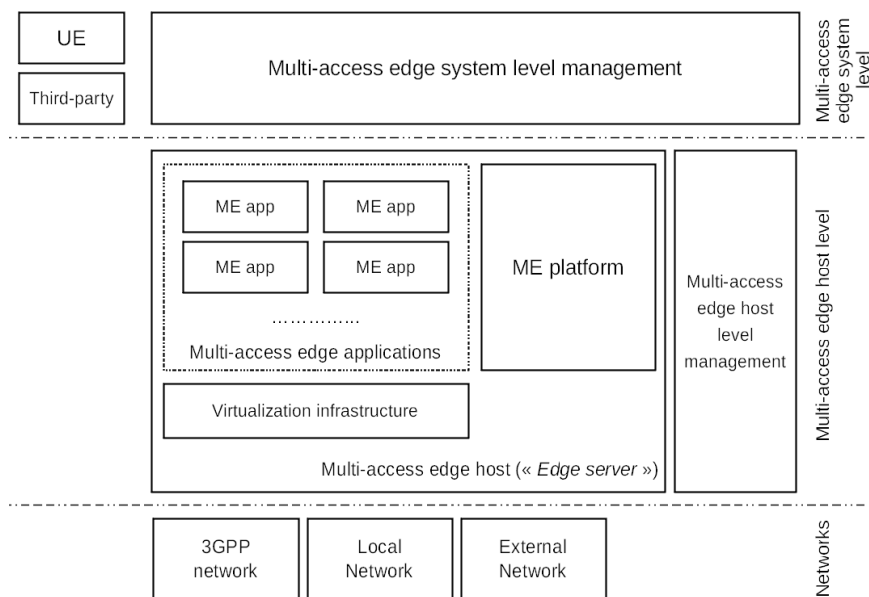


FIGURE 2.1 Structure MEC selon [60]

terrestre pour être traitées par des serveurs cloud centralisés selon la méthode MCC.

Les moyens que possède un véhicule connecté pour solliciter des ressources de calcul externes sont donc rendus nombreux par l'arrivée du MEC. Le déploiement à large échelle de la 5G renforce cette capacité de communication [61], tout en rappelant la diversité des données en circulation, en lien avec la grande variété d'applications abordée en introduction. Une classification est établie afin de faciliter le traitement de ces données par le MEC en fonction de leurs propriétés [62] :

- *Hard real-time* : données critiques ou ayant fixé des besoins en latence stricts (informations d'urgence, conduite à distance). Ces données doivent impérativement être traitées à bord ou en périphérie de la route ;
- *Soft real-time* : données dont le délai de traitement requis est faible, mais tolérant une légère dégradation des performances (informations de trafic, contrôle des feux de circulation). Les cloudlets peuvent déléguer une part du traitement au Cloud si les ressources de périphérie sont faibles ;
- *Non-real-time* : données dont le traitement n'a pas de contrainte de temps (télémétrie, informations de diagnostic à destination du constructeur). Le Cloud prend en charge ces données afin de libérer les ressources edge.

Au vu de la diversité des possibilités de déploiement pour les architectures *edge*, tant dans le domaine du matériel (quantité d'équipements, positionnement) que dans celui des mécanismes (gestion de la charge de calcul, de la sécurité, de l'énergie), les métriques

TABLEAU 2.1 RÉSUMÉ DES KPI DES PARADIGMES CLOUD ET EDGE COMPUTING

KPI	CC	MCC	EC	MEC
Localisation serveur	Centralisée	Centralisée	Distribuée	Distribuée
Capacité calcul	Très haute	Très haute	Limitée	Limitée
Capacité stockage	Très haute	Très haute	Limitée	Limitée
Temps transmission	Long	Long	Court	Court
Gestion mobilité	Basique	Avancée	Basique	Avancée
Réallocation ressources	Très rare	Sporadique	Fréquente	Très fréquente

d'évaluation associées sont nombreuses [63]. Dans ce chapitre de contexte, nous faisons le choix d'illustrer les *Key Performance Indicators* (KPI) généraux permettant de différencier simplement les types d'architectures abordés. Le tableau 2.1 présente une brève comparaison de ces KPI pour CC, MCC, EC et MEC. Une fois encore, la transposition de ce nouveau paradigme à des considérations liées aux véhicules, ainsi que la possibilité de transmissions d'appareil à appareil (D2D), de prise de décision locale et de calcul local, ont marqué la naissance du *Vehicular Edge Computing* (VEC) [27].

Remarque : le fog computing

D'une certaine manière, MEC est une partie du *Fog Computing* (FC), dans la mesure où ce dernier englobe toutes les communications dynamiques entre le nuage et les terminaux, tandis que l'edge computing se concentre spécifiquement sur la périphérie du réseau. Comme indiqué précédemment, les deux sont focalisés sur le traitement en périphérie, ce qui leur permet de raccourcir les chemins de transmission des données et d'atteindre des temps de latence plus faibles, tout en restant capables de s'appuyer sur les ressources du Cloud si les applications nécessitent une quantité de ressources qui dépasse ce que les appareils finaux peuvent fournir. Cependant, plusieurs facteurs séparent MEC et FC. Tout d'abord, ils sont supervisés par des groupes de recherche différents : respectivement l'ETSI (voir 2.3.3) et le consortium OpenFog [64]. Deuxièmement, le MEC peut extraire des ressources des RSU exclusivement, sans s'appuyer sur les véhicules. En effet, certains déploiements préfèrent ne pas s'appuyer sur le calcul local pour accomplir des tâches, tandis que d'autres permettent de décharger certaines de leurs demandes sur des véhicules proches si ce processus peut réduire la taille de la file d'attente et le temps de traitement global [65]. Troisièmement, alors que les edge nodes doivent être proches d'une station de base, les fog nodes peuvent être positionnés n'importe où dans le réseau, entre le cloud et les appareils finaux, afin d'apporter autant d'intelligence que possible au niveau local sans fil [66]. Le fog brouille donc les frontières entre les trois tiers du MEC. Cette différence a motivé OpenFog à mettre en œuvre des mécanismes de communication inter-nœuds avancés, permettant la communication à travers les fogs et un handover plus rapide [64]. Cette fonctionnalité n'est que partiellement prise en charge par les architectures MEC, qui encouragent les appareils à se déconnecter d'un serveur périphérique pour se connecter au second [60].

De ces différences conceptuelles découlent d'importantes variations de performances. L'extension *Vehicle Fog Computing (VFC)* est évolutive et polyvalente du point de vue du trafic routier, ce qui signifie que des conditions de trafic plus denses apportent des ressources informatiques plus importantes en intégrant rapidement les nouveaux arrivants dans le réseau. Chaque donnée passant par la couche véhicule, le délai est considérablement réduit car aucune transmission ne doit atteindre le cœur du réseau, et l'équilibrage des charges s'effectue indépendamment d'un contrôleur central [67]. Toutefois, l'inconvénient clair est que les performances dépendent directement de la densité des véhicules, ce qui peut être une excellente occasion de solliciter des ressources auprès de véhicules voisins ou inactifs dans les environnements urbains, mais perd son avantage dans les contextes très mobiles ou épars. Dans ce dernier cas, il est nécessaire de se connecter au Cloud pour obtenir des ressources supplémentaires quand la demande dépasse les ressources disponibles, au prix d'une forte augmentation de latence.

Ce problème est moins susceptible de se poser dans le cadre du *MEC*, où l'assistance des edge servers peut être sollicitée comme solution intermédiaire entre les ressources des dispositifs finaux (rapides mais limitées) et l'assistance des serveurs Cloud (illimitée mais lente), en fonction de la quantité de traitement nécessaire à une tâche et selon qu'elle soit sensible au temps ou non. Il existe un intermédiaire entre ces deux paradigmes, au carrefour du calcul embarqué distribué et des infrastructures edge-Cloud : les *Micro-Macro-Clouds (MMC)*. Dans ce paradigme, les nœuds d'extrémité voisins sont regroupés en grappes afin de se synchroniser et de partager les ressources inutilisées, qu'il s'agisse de calcul ou de communication. Ce comportement peut rappeler le *VFC*, mais le *MMC* se différencie par l'apport d'un macro-Cloud superviseur, géré sur une couche virtuelle qui prendrait la place de la couche edge, en s'appuyant sur la connexion backhaul¹ pour servir de point de collecte de données entre le Cloud et les appareils. En effet, comme tous les véhicules partagent leurs propres ressources de communication et de calcul, le besoin de ressources physiques supplémentaires disparaît, ce qui permet d'éviter le déploiement de serveurs en périphérie [68]. La diffusion des données est assurée entre les membres du micro-cloud. Lorsque les services du Cloud ne sont pas nécessaires, toutes les communications restent dans la couche la plus basse, ce qui permet de réduire considérablement les temps de latence.

1. Les éléments d'infrastructure capables de communiquer peuvent être interconnectés par un lien physique nommé backhaul. Ces liens bénéficient en général de débits particulièrement élevés, ce qui permet des échanges de données internes à l'infrastructure à haute vitesse (voir figure 2.2).

2.3.4 Le vehicular edge computing

Besoin

Le **VEC** étend l'architecture **MEC** à l'**IoV**, résumant ainsi la couche *device* à tous les véhicules connectés. La couche *edge* assure toujours le lien entre la couche de cœur et la couche véhicule, et rapproche les ressources *cloud* de la route afin de réduire la latence autant que possible. Dans le cas du **VEC**, les *cloudlets* apportent une autre contribution intéressante dans le domaine de la connaissance de l'environnement. En effet, compte tenu de la multiplicité des serveurs de bordure et de leur proximité d'une station de base, le concept de localisation des véhicules en fonction de leur position par rapport à ces dispositifs périphériques est plus satisfaisant en termes de précision et de temps d'exécution [69]. En outre, nous avons constaté que les véhicules peuvent être équipés d'une grande diversité de capteurs, dont une conséquence directe est la grande quantité de données fournies [70]. Ce contenu ne peut être ni stocké ni traité sur le long terme en utilisant exclusivement la capacité embarquée. L'abstraction et le déchargement des tâches vers les *cloudlets* permettent à une diversité d'applications de cartographie de continuer à fonctionner dans des situations de forte mobilité.

Principe

Néanmoins, l'informatique embarquée ne se limite pas à une simple redéfinition de la nature du terminal. Au contraire, les véhicules connectés peuvent utiliser leur interface de transmission à courte portée pour communiquer avec d'autres véhicules, tout comme ils peuvent communiquer avec des **RSU** bien plus proches d'eux qu'une station de base cellulaire. Cette interface *Vehicle-to-Vehicle (V2V)* permet à ces dispositifs de partager des données, mais surtout des ressources informatiques et de stockage. La capacité d'échanger des ressources (sous la forme de données de capteurs abstraites, d'informations contextuelles, de services ou même de tâches) entre véhicules est le premier apport du **VEC** par rapport au **MEC**. La deuxième contribution est la possibilité d'utiliser cette même interface à courte portée pour communiquer avec les unités de bord de route si nécessaire. Comme ces unités sont les éléments d'infrastructure les plus proches de la route, elles servent de point d'accès au réseau disponibles pour les véhicules avec une interface à courte portée, tout comme les stations de base le font avec leurs interfaces cellulaires. En outre, étant donné que les **RSU** peuvent, comme les stations de base, être connectées à un serveur périphérique via un lien *backhaul*, le **VEC** a le potentiel d'apporter des ressources informatiques encore plus près de la route. Enfin, un véhicule intelligent peut calculer lui-même un certain nombre de ses propres tâches tout en restant conscient de son environnement et des dispositifs qui l'entourent, ce qui est une exigence cruciale en matière de sécurité routière et de continuité de service. L'architecture suivante est décrite à la figure 2.2, où l'on peut apprécier la flexibilité du positionnement des serveurs périphériques.

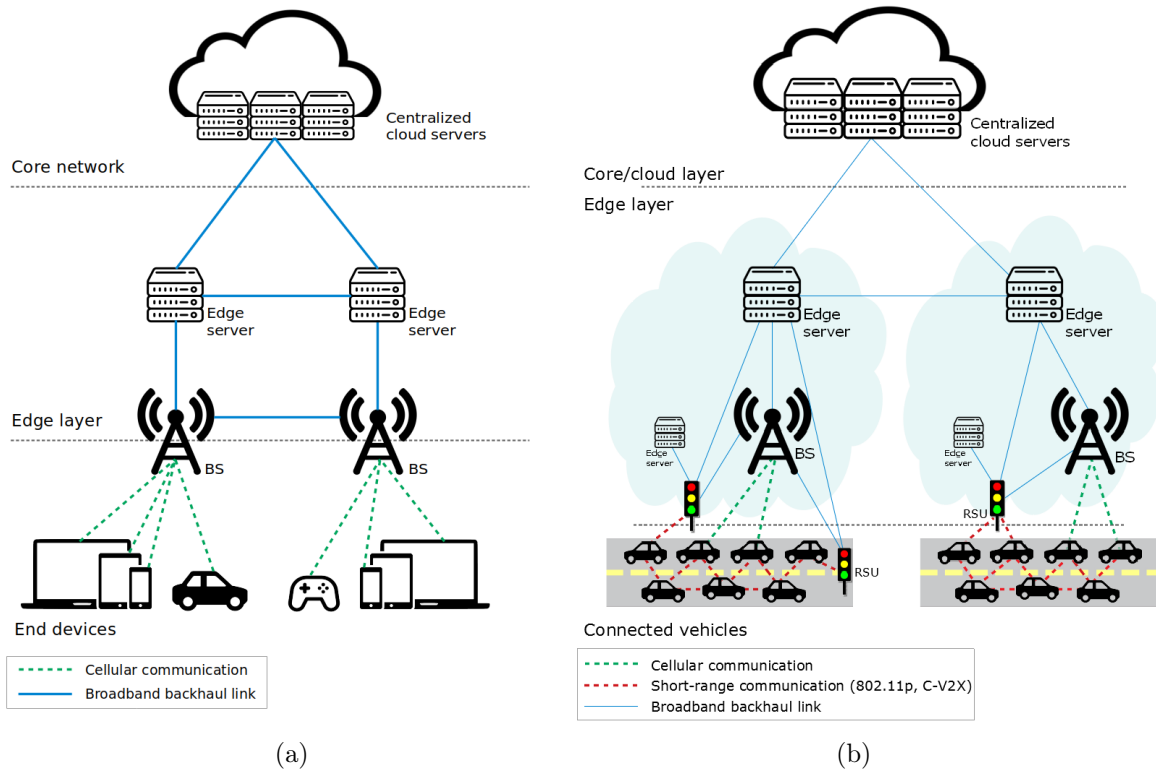


FIGURE 2.2 Comparaison des architectures MEC (a) et VEC (b).

Standard

Bien que le VEC soit inextricable du MEC d'un point de vue conceptuel, il ne semble pas y avoir eu d'effort de standardisation spécifique jusqu'à présent. Cependant, même si des efforts ont été faits pour étudier les applications VEC et leurs problématiques techniques [71], une part considérable de la recherche sur l'informatique de périphérie est consacrée à l'intégration des architectures informatiques de périphérie pour les utilisations V2X. Néanmoins, ce domaine de recherche semble encore être axé sur la communication [72] ou sur le MEC, c'est-à-dire que les services dépendent plutôt du soutien d'un serveur périphérique, contrairement à l'informatique véhiculaire qui s'efforce d'exploiter autant que possible les ressources des dispositifs finaux eux-mêmes. Pour résumer les nouveautés apportées par le VEC, ce paradigme d'informatique périphérique diversifie grandement les possibilités de calcul des tâches pour les véhicules :

- L'OBU du véhicule contient suffisamment de ressources pour calculer plusieurs tâches immédiates nécessaires à son bon fonctionnement. La présence de capteurs embarqués permet la prise en charge de la géolocalisation du côté du véhicule.
- Si l'OBU dispose d'une puissance de calcul supplémentaire, il peut être sollicité par un véhicule proche à l'aide de communications à courte portée.

- En ce qui concerne les ressources de communication, l'interface à courte portée peut être utilisée pour transmettre des requêtes provenant d'autres véhicules, soit vers un véhicule voisin (V2V), soit vers une RSU (*Vehicle-to-Infrastructure* (V2I)). En effet, le canal à courte portée permet les deux utilisations.
- Tout comme n'importe quel autre appareil mobile, l'interface cellulaire permet un accès longue distance aux stations de base du réseau mobile. Les RSU et les stations de base disposent toutes deux d'une connectivité vers le serveur périphérique le plus proche.

La principale contribution du MEC est la réduction de latence lors de l'accès aux ressources informatiques distantes. Dans le cadre du MEC et du VEC, les équipements sont encouragés à exploiter les ressources des appareils environnants par le biais du délestage des tâches. Le délestage repose sur deux aspects : le calcul des tâches et la fourniture de contenu.

2.4 Les mécanismes de calcul dans le vehicular edge computing

Du point de vue de la décision de délestage, de nombreuses stratégies ont été élaborées dans le domaine de l'affectation des tâches dans le cloud computing [73] et dans le cas de ressources disponibles finies [74]. Parallèlement aux travaux historiques sur l'équilibrage de charge ([75], [76]), ces contributions appellent une adaptation à des contextes très mobiles, auxquels les applications de délestage de tâches fondées sur les MEC V2X semblent correspondre.

Le problème de la fourniture de contenu dans l'edge computing est le manque d'efficacité lorsque les paquets n'ont pas été mis en cache sur les serveurs de périphérie et doivent donc être récupérés au niveau du Cloud. Ce processus implique une augmentation de la latence (les centres de données étant situés loin du serveur MEC), ce qui dégrade l'efficacité de la diffusion.

2.4.1 Intelligence de périphérie

Pour contrebalancer ce comportement, il existe une architecture edge computing dans laquelle les RSU regroupent les informations contextuelles fournies par les véhicules environnants, afin d'aider à allouer l'espace de stockage et à récupérer le contenu en fonction de la densité des utilisateurs [77]. En outre, l'intelligence de périphérie est coordonnée et partagée entre les serveurs et les véhicules. Les serveurs en périphérie peuvent informer les serveurs Cloud de leur comportement en matière de mise en cache afin d'optimiser les intervalles d'extraction et de mise à jour du contenu [78]. Ainsi, comme le contenu populaire

est récupéré de manière anticipée sur la couche edge (il descend donc préventivement des serveurs Cloud et est envoyé aux serveurs edge, comme le montre la figure 2.2(b)), il peut être fourni immédiatement après la première demande de l'utilisateur, et le véhicule récepteur peut ensuite diffuser le contenu récupéré parmi ses pairs. [79] propose également un schéma reposant sur la collaboration pour la conduite automatisée, avec les tiers edge et Cloud partageant leurs connaissances sur le comportement du système. Un serveur edge est positionné à chaque intersection. Un modèle d'interaction est formé par apprentissage par renforcement, un mécanisme de *Machine-Learning (ML)*, pour orchestrer l'arrivée des véhicules. La multiplicité des équipements qui peuplent le réseau constitue le cœur de cette structure. Cependant, peu de travaux de recherche intègrent la possibilité d'avoir très peu d'équipements disponibles dans l'architecture. En outre, la question du délai de communication supplémentaire qu'implique l'accès fréquent au Cloud pour prendre une décision reste à traiter.

Par ailleurs, la collaboration entre les utilisateurs du réseau permet de mettre en commun les données des capteurs et les résultats des calculs afin de laisser chaque acteur décider du contenu à mettre en cache [80]. Les processus de calcul et de mise en cache sont influencés par la disponibilité du débit, qui est attribué par enchères. Les contenus populaires et gourmands en ressources peuvent être traités par les gagnants, laissant les contenus plus modestes aux perdants, ce qui permet d'éviter les doublons dans le stockage et le calcul des tâches. Par conséquent, dans les architectures collaboratives, l'intervention d'un serveur edge est nécessaire à la fois au niveau de l'utilisateur et au niveau du Cloud, pour la planification de la diffusion et la gestion de la politique de mise à jour du contenu, ainsi qu'à des fins d'apprentissage profond [81].

L'intelligence de périphérie peut également être exploitée pour estimer le délai sur chaque liaison. En réponse à l'incertitude des performances du lien backhaul (largement due à l'intervention du Cloud et du cœur, comme nous l'avons indiqué), un processus de décision concentré sur la liaison fronthaul pour un déchargement efficace des tâches est établi [82]. Étant donné que les délais de délestage entre l'UE et la station de base dépendent de moins de variables que les échanges entre la station de base et le cœur, l'algorithme procède à l'estimation du délai maximal. Ainsi, les décisions de déchargement sont orientées vers les estimations de délai les plus faibles sur le fronthaul. Avec le taux de transmissions réussies, ces métriques sont déterminantes dans l'établissement d'une solution *VEC*, étant donné qu'un échec implique des retransmissions.

2.4.2 Intelligence collective entre hôtes

Le **VEC** proposant une plus grande exploitation des ressources embarquées, les usagers peuvent solliciter les capacités d'autres véhicules pour accomplir des tâches [83]. [84] étend l'utilisation de la couche véhicule en exploitant les technologies de clustering, de micro-service et de conteneurisation pour fournir des ressources mobiles de type edge fondées sur la puissance de traitement disponible dans les **OBU**. [85] suit un délestage hybride similaire, en utilisant les **RSU** comme juges qui sélectionnent la destination d'une tâche délestée, parmi les clusters de véhicules, les serveurs edge ou le Cloud. D'autre part, [86] souligne l'importance des communications edge en trouvant de nouvelles utilités aux transmissions mobiles *Vehicle-to-Network* (**V2N**) autrefois sporadiques, avec un positionnement des **RSU** directement au niveau des gNodeB. [87] poursuit cette étude en combinant le soutien du Cloud avec un mécanisme d'agrégation de données de contextes par le regroupement de véhicules, puis par leurs remontées au moyen d'un véhicule passerelle élu dynamiquement.

Par ailleurs, dans la situation de forte mobilité inhérente au **VEC**, les portées de communication **D2D** deviennent problématiques pour l'échange de données entre utilisateurs, ainsi que pour l'échange de tâches avec les **RSU**. Ainsi, le **V2V** multi-saut a été étudié afin de propager les messages en amont de l'hôte émetteur, ce qui permet aux tâches d'être calculées au niveau d'une **RSU** située plus loin sur la route avant que le véhicule de déchargement ne l'atteigne pour recueillir les résultats [88]. Cependant, aucun itinéraire ni aucune destination ne sont clairement définis lorsque le véhicule demande le déchargement d'une tâche, ce qui augmente grandement le risque d'échec dans des situations urbaines.

De la même manière, du point de vue de la fiabilité et de l'évolutivité, [89] vise à coordonner tous les éléments sur la route dans une architecture centrée sur les **RSU**. Les véhicules s'appuient sur l'infrastructure pour solliciter les ressources **MEC** et **VEC** voisines. Les tâches sont ensuite toujours déchargées via des communications à courte portée, ce qui oblige les véhicules à communiquer soit avec une **RSU** à portée, soit entre eux, même lorsqu'ils relaient des données vers le Cloud.

2.4.3 Optimisation des ressources de calcul

Ainsi, dans un scénario **V2X**, si la capacité de calcul embarquée ne permet pas d'assurer le calcul local et si les ressources des serveurs périphériques sont sollicitées trop fréquemment, deux solutions peuvent être imaginées. Premièrement, ajouter d'autres possibilités de déchargement vers des **RSU** proches ou même des BS, et deuxièmement, optimiser l'utilisation de la capacité de calcul des serveurs de périphérie. Il est inconcevable d'augmenter la capacité de calcul à l'intérieur des véhicules, car les conséquences sur le coût global de la solution ne seraient économiquement pas soutenables [90].

En permettant aux utilisateurs de solliciter eux-mêmes des ressources **CC** par l'intermédiaire de leur liaison **MEC**, au lieu de laisser les serveurs edge prendre cette décision, il est possible d'obtenir une architecture adaptée au traitement rapide des demandes gourmandes en ressources [91]. Elle offre une décision de délestage supplémentaire en utilisant la couche périphérique décrite à la figure 2.2(a) comme une simple passerelle pour compenser la sous-optimalité d'un processus de délestage **V2V**, moins approprié pour les tâches ayant des besoins élevés en matière de calcul ou de stockage.

2.4.4 Traitement des données massives

En ce qui concerne le traitement des données volumineuses, les processus de décision sont nombreux. Nous avons vu précédemment comment les **RSU**, les contrôleurs centraux ou même les véhicules peuvent gérer des stratégies de délestage en fonction de la charge de données entrantes. Mais les décisions d'ordonnancement et d'équilibrage de la charge ne doivent pas toujours être prises en fonction des files d'attente de données. En effet, la question finale en matière de **VEC** reste la **QoS**, et non des mesures intermédiaires telles que l'état des files d'attente [92]. Une file d'attente entièrement remplie de tâches rapidement exécutables peut être vidée plus rapidement qu'une file d'attente plus courte mais chargée de tâches complexes. À la lumière de cette observation, un système hybride d'ordonnancement des tâches (**HDSS**) peut s'adapter à la fois aux mécanismes de prise de décision fondés sur les files d'attente et sur le temps. Le passage d'un mécanisme à l'autre peut garantir une stabilisation des temps de réponse, tout en évitant ponctuellement une utilisation accrue des ressources du processeur, étant donné que l'ordonnancement fondé sur le temps nécessite une certaine quantité de calcul pour estimer le temps nécessaire au traitement de diverses files d'attente.

2.4.5 Flexibilité des performances

Enfin, si la principale préoccupation est la satisfaction des contraintes d'une application, l'architecture **VEC** déployée doit constamment redéfinir les priorités de performance du réseau en fonction de la demande en cours. En réattribuant les ressources des serveurs périphériques à chaque demande de tâche envoyée par chaque véhicule, une telle flexibilité peut être proposée [93]. Par conséquent, chaque application véhiculaire doit non seulement déterminer si ses tâches doivent être déchargées, mais aussi la quantité de blocs de calcul des serveurs edge que ces tâches vont nécessiter. La latence supplémentaire que cette estimation implique est contrebalancée par la flexibilité apportée par cette innovation, en particulier dans le cas d'applications gourmandes en ressources. Plus récemment, une approche online distribuée de la gestion des ressources périphériques permet de réduire le temps de latence tout en restant conscient de l'impact des demandes de service entrantes sur l'état du réseau [94]. Son objectif est de minimiser le temps de réponse moyen de manière rentable, en exploitant une structure online et distribuée pour étudier en temps réel la taille des files

d'attente et l'état des liaisons. En utilisant des données fraîches à chaque intervalle de temps, l'algorithme de sélection économise un espace de stockage crucial et reste polyvalent, au prix d'une utilisation supplémentaire des ressources. Le suivi de la nature et du comportement des demandes entrantes en temps réel permet l'optimisation conjointe de la programmation et de l'allocation des ressources, en réaction à des changements soudains dans l'utilisation du réseau, ce qui est approprié pour des événements inattendus et demandeurs qui caractérisent les réseaux véhiculaires.

2.5 Notre positionnement

Suite à l'état de l'art dressé dans ce chapitre, nous retenons que les mécanismes de calcul étudiés dans le contexte du VEC visent comme objectif principal la réduction des délais de traitement. Toutefois, nous avons remarqué qu'une majorité de travaux est concentrée sur des contextes expérimentaux simplifiés, ce qui réduit de fait leur généralité. En connaissance des applications décrites dans la première partie de ce chapitre, nous poursuivons le même objectif de réduction des délais, en ajoutant une considération pour les contextes de déploiement plus complexes (forte urbanisation, rareté des équipements, charge élevée). L'objectif principal de notre contribution est donc la conception et l'évaluation approfondie d'un mécanisme de délestage des tâches véhiculaires hétérogènes à l'échelle urbaine, en vue de minimiser les délais de traitement. Il existe de nombreuses différences significatives entre notre travail et la littérature existante concernant le déchargement des tâches véhiculaires :

1. **Gestion des tâches.** Les tâches engendrées par les véhicules sont délestées vers une destination edge sélectionnée dynamiquement en fonction des informations de contexte disponibles, notamment la position et les ressources de calcul disponibles au moment de l'envoi. L'algorithme de décision relève d'une heuristique en réponse à un problème de minimisation du nombre de tâches en retard pour de multiples machines.
2. **Longue portée.** L'utilisation du réseau cellulaire fait partie intégrante de notre solution sans pour autant dépendre du Cloud, là où les contributions existantes utilisent rarement les communications V2N ou nécessitent de nombreuses interventions du Cloud, ce qui dégrade la performance.
3. **Contexte expérimental.** Nous avons observé que la plupart des systèmes de déchargement des tâches des véhicules ont été conçus pour des terrains de jeu simples tels que des segments d'autoroute. Notre système vise une architecture urbaine avec des attributs similaires à ceux d'un quartier, tels qu'une grande quantité d'obstacles immobiles et en mouvement, ou bien de multiples axes routiers empruntés par les véhicules.

TABLEAU 2.2 Comparaison des solutions MEC/VEC

Pour chaque système (**ligne**), ses caractéristiques edge computing sont détaillées, une astérisque (***) notant les aspects pris en charge. Nous juxtaposons les astérisques en fonction du niveau de précision accordé à chacun des paradigmes MEC et VEC. Pour la gestion en cache, nous différencions le lieu de mise à disposition des données : ‘V’ signifie véhiculaire, ‘E’ signifie edge, ‘C’ signifie Cloud. La même logique s’applique pour les meilleures conditions d’exécution selon la densité d’utilisateurs : ‘D’ signifie que le système est plus adapté aux environnements denses comme les centres villes ou les périphériques, ‘S’ signifie les milieux épars et fortement mobiles comme les autoroutes, ‘A’ signifie que le système est robuste pour toutes les densités.

Ref.	MCC	MEC/VEC	ML	Cellulaire	Caching	Densité
[67]		*/***			V	D
[77]	*	*/**			E,V	D
[78]		***/*	*		E,V	D
[79]	*	***/*		*	C,E	D
[80]		**/**	*		E,V	A
[81]		**/**	*		E,V	A
[82]		**/**	*		E	A
[84]		*/***			V	D
[85]	*	**/**			E,V	A
[86]		**/**		*	E,V	D
[87]	*	*/**		*	C,V	D
[88]		***/**			E,V	D
[89]		*/***			E,V	D
[68]	*	*/***			C,V	D
[91]	*	*/**			E	A
[92]		**/*			E	A
[93]	*	*/**			C,E	A
[94]		*/***			E	A
Notre positionnement		***/*		*	E	A

4. **Temps réel et file d’attente.** Nous avons décidé d’évaluer les performances de notre algorithme selon deux modèles de gestion de tâches du côté des serveurs : premièrement en temps réel, où les serveurs exécutent les tâches à la volée, puis deuxièmement avec une possibilité de différé, lorsque les serveurs peuvent stocker les tâches reçues dans des files d’attente.

CAVTOMECC : OPTIMISER LE DÉLESTAGE DE TÂCHES DANS UNE ARCHITECTURE VEC

Dans ce chapitre, nous présentons CAVTOMECC, notre système d'optimisation de délestage de tâches par intelligence collective, au sens présenté dans le chapitre précédent, dans une architecture véhiculaire hétérogène. Pour y parvenir, nous exploitons l'interface de communication à longue portée de nos éléments d'infrastructure afin de disséminer vers les véhicules les informations de contexte nécessaires à une sélection appropriée de la destination. Dans une architecture simple, un véhicule produit des données et des tâches de sorte à soit les traiter à la volée, soit à s'en délester immédiatement vers le serveur périphérique le plus proche. Le premier cas induit une limitation des cas d'usage puisque, comme nous l'avons énoncé précédemment, les capacités de calcul à bord sont trop limitées pour prendre en charge les applications les plus complexes et critiques. Le second cas permet l'exploitation de ressources *edge computing*, mais ne permet pas d'équilibrage de charge ni d'adaptation au contexte d'utilisation de la route. Avec CAVTOMECC, les véhicules engendrant de la charge de calcul sont capables, par la réception fréquente des informations de capacité de calcul disponibles dans chaque serveur de périphérie, d'adapter leur choix de la destination vers laquelle envoyer leurs requêtes en fonction de leurs besoins applicatifs. Cette solution permet un équilibrage de charge sans ordonnancement centralisé, dynamique et facilitant le traitement de tâches critiques face à de multiples conditions d'utilisation de la route. Nous déployons CAVTOMECC comme une solution complémentaire au sein des OBU et RSU, et en évaluons les performances par simulation, sur un terrain d'expérimentation analogue au campus universitaire de Toulouse III Paul Sabatier.

Ce chapitre comprend trois grandes parties. Nous étudions en premier lieu l'importance d'une interface de communication à longue portée pour le partage d'informations de contexte entre éléments d'une infrastructure véhiculaire urbaine. Nous présentons ensuite un descriptif détaillé de la conception du système, puis nous en évaluons les performances avant de conclure le chapitre.

3.1 Étude du besoin

Avec l'intérêt croissant porté aux smart cities, le **Véhicule Autonome Connecté (VAC)** devient de plus en plus attractif. Néanmoins son acceptation par le grand public semble progresser lentement, appelant généralement des cas d'usage sécurisants, polyvalents et fiables [95]. En effet, malgré les gains de performances offerts par le **MEC**, les véhicules connectés demeurent particulièrement mobiles par nature et abritent des applications critiques, pour lesquelles les questions de stabilité de connectivité, d'efficacité de communication, de gestion du temps et des ressources doivent être résolues. Il est fréquent que les serveurs périphériques soient surchargés par la quantité de tâches complexes à exécuter de la part d'appareils environnants, ce qui augmente grandement le temps total de traitement [96]. Nous avons remarqué dans le chapitre précédent que cette situation appelle à l'étude de mécanismes de gestion de charge afin de répartir les demandes transmises par les véhicules. Toutefois, les véhicules eux-mêmes disposent intrinsèquement d'éléments pouvant participer activement à de tels mécanismes, avec un potentiel remarquable de robustesse et de réduction de complexité. Parmi ces éléments, les interfaces de communication elles-mêmes représentent une diversité de modes de transmission : le **D2D** pour son délai réduit, ou bien la longue portée pour sa fiabilité. Nous nous proposons d'impliquer ces éléments dans l'élaboration d'une solution de délestage.

3.1.1 Différenciation D2D et longue portée

Comme nous l'avons décrit précédemment, les véhicules eux-mêmes peuvent servir de centres de calcul, ce qui implique un besoin de joignabilité et donc de flexibilité dans les capacités de communication. Par conséquent, le premier défi à relever pour améliorer les performances est la sélection du réseau sur lequel envoyer les données. En effet, les applications actuelles gérées par les véhicules reposent principalement sur deux types de communication : **V2V** et **V2I**, qui bénéficient toutes deux des technologies de communication **D2D** grâce à leur traitement rapide des communications à courte portée. Cependant, les portées de communication de ces technologies sont limitées par nature, ce qui entre en conflit avec la nature fortement mobile des hôtes. D'autres types de communication sont à l'étude, tel que le **V2N**.

Les réseaux véhiculaires étant principalement inquiétés par les limitations de portée et de capacité, il s'ensuit naturellement que les réseaux cellulaires, à commencer par le **LTE** [97], soient perçus comme des candidats idéaux pour résoudre ces problèmes. En émettant via ces interfaces cellulaires, le véhicule obtient la capacité de communiquer sur de longues distances comme n'importe quel autre appareil cellulaire. Dans ce cas, les communications passent par une *Base Station (BS)* : eNodeB ou gNodeB, pour **LTE** et **5G** respectivement, aboutissant à un minimum de deux transmissions, l'une selon un lien montant (*Uplink (UL)*) et l'autre descendant (*Downlink (DL)*). Comme nous l'avons remarqué dans notre

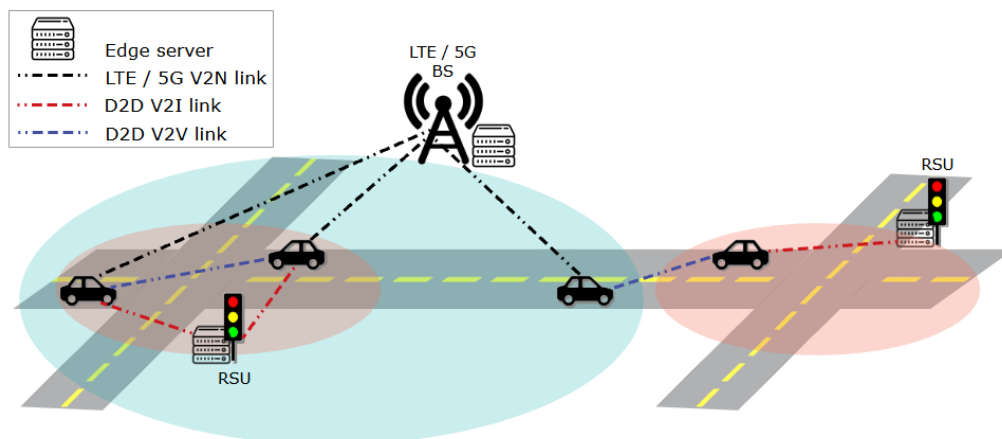


FIGURE 3.1 Scénario de communications véhiculaires multi-RAT

état de l'art, les communications **V2N** ont tendance à être négligées dans la plupart des applications véhiculaires en raison de leur délai plus important que le **D2D**, dû à ce saut supplémentaire et au passage dans le cœur de réseau. Dans le cadre des communications **D2D**, la notion de montant et descendant disparaît, au profit d'une unique transmission selon un lien horizontal (*Sidelink (SL)*).

Ainsi, suite à la caractérisation d'une multitude de cas d'usages véhiculaires par des groupes tels que **5GAA** [8], [9], il devint indéniable que les critères de performance associés ne correspondaient pas à l'offre actuelle des réseaux cellulaires mobiles. La diversité d'usages exige une architecture de réseau souple et polyvalente, garantissant la continuité du service quelle que soit l'application qui en dépend, le nombre d'hôtes qui doivent communiquer par son intermédiaire et la durée de la communication. Par conséquent, les transmissions V2X à longue distance sont limitées dans les standards à des mises à jour sporadiques du trafic routier dans le cas d'un événement inattendu se produisant en amont de l'itinéraire [98]. Cependant, nous pourrions imaginer une solution dans laquelle la centralisation des informations au niveau de la **BS** contribuerait à la gestion des ressources. La présence d'une interface cellulaire représente, à notre avis, un potentiel inexploité pour le déchargement optimisé des tâches, étant donné sa faculté à partager des données utiles sur distances et avec des nombres d'hôtes inatteignables pour les interfaces à courte portée. Un exemple de communications conjointes cellulaires et **D2D** est présenté sur la figure 3.1.

3.1.2 Faisabilité d'une solution de délestage hybride

Les capacités informatiques embarquées des véhicules sont limitées par nature et ne sont généralement pas conçues pour gérer plusieurs applications à la fois et pendant des périodes prolongées. Dans ces situations, les tâches sont placées dans des files d'attente en attendant la libération des ressources de traitement. Par conséquent, pour éviter l'accumulation dans les files d'attente et les temps d'attente excessifs, les grandes quantités de données et les

tâches et opérations les plus complexes doivent être déchargées du véhicule vers des serveurs MEC de grande capacité situés à proximité de la route, couplés aux RSU. Nous dénommons cette opération "délestage" par traduction du terme *offloading*.

Le délestage fait partie intégrante des architectures véhiculaires, jusqu'au matériel commercial proposé dans ce secteur. En effet, au vu de l'importance du multi-RAT dans les conditions de communications variées et contraignantes caractéristiques des applications véhiculaires, les interfaces réseau exploitables dans le cadre du délestage sont d'ores et déjà multiples dans les solutions OBU et RSU commerciales [39], [40]. La plupart des véhicules hôtes sont donc actuellement équipés de plusieurs *Radio Access Technology (RAT)*, leur permettant de s'inscrire dans un réseau hétérogène. La présence de ces équipements permet à l'architecture de couvrir de multiples cellules avec diverses zones de couverture, ainsi que de proposer des niveaux de qualité de service adaptés à de multiples applications.

Nous concluons donc que la conception d'une solution de délestage hybride, fondée à la fois sur les transmissions courte portée et longue portée, est pertinente dans le contexte matériel actuel.

3.1.3 Réponse au besoin

Dans les réseaux véhiculaires modernes, les hôtes s'appuient sur le délestage de tâches et de données, afin d'améliorer les performances en réduisant les délais. Toutefois, si une situation exige que des tâches hautement prioritaires soient relayées et calculées dans des délais très courts, il peut arriver que le serveur MEC le plus proche soit surchargé avant la réception de la tâche, et donc incapable de calculer d'autres opérations sans avoir préalablement terminé les opérations en cours. Dans ce cas, les tâches critiques nouvellement arrivées doivent être soit mises en file d'attente le temps que les ressources soient libérées, soit déchargées vers une RSU plus éloignée, ce qui entraîne dans les deux cas un important retard supplémentaire.

En réponse à cette situation, nous présentons CAVTOMECC, un mécanisme multi-RAT de délestage des tâches V2X en fonction du contexte pour les réseaux MEC. Cette solution tire profit de la nature hétérogène des réseaux urbains, en exploitant les interfaces courte et longue portée disponibles à bord, à des fins de délestage et d'intelligence collective. Les paramètres pris en compte dans le cadre du délestage contextuel sont les suivants : priorité de la tâche, ressources de calcul actuellement disponibles sur les serveurs MEC, et distance entre l'hôte demandeur et les candidats au délestage.

3.2 Le délestage

Le délestage de tâches d'un véhicule vers un serveur MEC s'effectue généralement en trois étapes : le véhicule prépare sa demande et transfère la tâche ainsi que les données associées à l'unité de service régionale qu'il a choisie comme meilleure destination. Ensuite,

la tâche est calculée après réception par le serveur MEC. Enfin, les résultats sont renvoyés à l'initiateur. Pour une tâche donnée u , nous pouvons définir le délai total de délestage t_u comme suit :

$$t_u = \delta t_u^{UL,c} + \Delta t_u^{comp} + \delta t_u^{DL,c} \quad (3.1)$$

où $\delta t_u^{UL,c}$ est le délai de communication pour la transmission de la tâche du véhicule au serveur MEC par le canal c (sens *Up*), Δt_u^{comp} est le temps de calcul au serveur MEC, et $\delta t_u^{DL,c}$ est le délai de communication pour la transmission des résultats du serveur MEC au véhicule par le canal c (sens *Down*).

Dans ce paradigme, un obstacle majeur est rencontré lors de la première étape, avant même l'envoi de la tâche. En effet, le fait de laisser le véhicule sélectionner de lui-même le serveur périphérique le mieux adapté implique un manque d'intelligence dans le mécanisme de sélection. Par conséquent, soucieux des délais et ne disposant que de peu ou pas d'informations contextuelles, l'hôte déleste généralement ses tâches sur la RSU la plus proche, sans tenir compte de la priorité ni de l'état de charge de la destination. Ce comportement augmente le risque de rejet de la tâche ou de ralentissement de l'exécution, ce qui implique des retransmissions et un délai de traitement beaucoup plus long. À l'inverse, déporter la responsabilité de la sélection de destination au niveau d'un orchestrateur externe nécessite de nombreuses communications supplémentaires ainsi qu'une connectivité permanente.

Dans tous les cas, le transfert de tâches vers les serveurs périphériques implique des délais de transmission supplémentaires pour atteindre le serveur périphérique, délais qui n'interviendraient pas si ces tâches étaient exécutées localement. Nous devons donc admettre que le délai total de délestage doit être inférieur au délai local de mise en file d'attente et d'exécution pour que ce paradigme soit pertinent. C'est pourquoi, du point de vue de la transmission, le positionnement de la RSU à proximité physique de la route et des véhicules réduit considérablement les délais de calcul et de communication. En outre, les serveurs MEC des RSU sont équipés de puissants processeurs, et parfois même de processeurs graphiques, de manière à surpasser les OBU dans leurs capacités de calcul. En temps réel, la quantité de ressources informatiques disponibles sur les serveurs MEC est notée A_n . Nous pouvons formuler le délai de calcul d'une tâche donnée u sur le serveur MEC t_u^{comp} comme suit [80] :

$$\Delta t_u^{comp} = \tau_u + \frac{Cr_u \times \epsilon}{F_{CPU}} \quad (3.2)$$

où τ_u est la durée que la tâche passe sur le serveur en attente d'être calculée, Cr_u est le besoin en ressources de calcul de la tâche en pourcentage des ressources de processeur du serveur MEC pendant le temps alloué, ϵ est le coefficient reliant cette valeur à un nombre d'opérations, et F_{CPU} est la fréquence d'horloge du processeur du serveur MEC.

3.3 Vue d'ensemble de CAVTOMECC

Dans le contexte des applications véhiculaires, où chaque milliseconde compte, des solutions doivent être mises en place pour atténuer les délais. Ainsi, pour une sélection plus appropriée du serveur MEC en fonction de chaque tâche, nous proposons un mécanisme de délestage tenant compte d'informations contextuelles étendues. Cette solution comprend la mise en circulation de messages de contexte, ainsi qu'un algorithme de prise de décision avec provisionnement de QoS, afin d'équilibrer la charge, de s'adapter à la priorité de chaque tâche et d'accroître la fiabilité du délestage des tâches par V2I.

En vue de l'étude formelle de notre architecture, nous modélisons donc un petit réseau de véhicules composé de quelques hôtes multi-RAT à interfaces longue portée et courte portée. Soit $\mathcal{M} = [1 \dots M]$ l'ensemble de tous les hôtes véhiculaires du réseau, indexé par i . Soit $\mathcal{N} = [1 \dots N]$ l'ensemble de tous les serveurs MEC situés dans les hôtes RSU du réseau et indexés par j . Les véhicules peuvent engendrer des tâches de complexité et de priorité variables et demander à délester ces tâches sur les serveurs MEC. Les niveaux de priorité des tâches sont haut, indifférent et bas, en lien avec la classification des données *hard*, *soft* et *non real-time* énoncée dans l'état de l'art. Les tâches de priorité haute doivent être traitées avant les autres dans la mesure du possible, tandis que les tâches de priorité basse doivent retarder leur traitement afin de favoriser celui des autres. Les hôtes RSU reçoivent les tâches des véhicules et allouent les ressources de leur serveur MEC pour calculer ces tâches. Ces ressources sont finies.

Notre solution étant fortement corrélée au contexte d'un petit nombre de VAC se déplaçant autour d'un quartier, elle contribue essentiellement à améliorer la sollicitation des ressources pour les hôtes particulièrement impliqués dans la vie d'un petit réseau, avec un besoin de délestage de tâches complexes ainsi que de grandes quantités de données. Dans cette situation, l'utilisation du V2N ouvre la voie à une exploitation efficace des deux types d'interfaces, robuste y compris dans des situations de charge plus élevée ou de populations plus denses, en rendant les véhicules capables d'envoyer et de recevoir des données au-delà de leur seule portée de communication D2D.

Le tableau 3.1 présente un résumé des paramètres impliqués dans notre système de délestage des tâches.

3.4 Circulation des données de contexte

3.4.1 Contenu

Les messages de contexte sont différenciés selon deux catégories : les mises à jour individuelles émises par les serveurs périphériques des RSU, et les récapitulatifs généraux transmis par celui de la BS. La mise en forme et l'envoi de ces messages sont corrélés à la circulation d'un autre type de message : les demandes de délestage de tâches. À des fins de

TABLEAU 3.1 Paramètres du système

Dénomination	Description
λ	Fréquence de génération de tâches
k	Niveau de priorité d'une tâche
γ_u	Taille des données associées à la tâche u
Cr_u	Besoin en ressources de calcul de la tâche u
A_n	Ressources de calcul disponibles sur la RSU n
σ_n	Écart-type des ressources de calcul disponibles sur la RSU n
P_u	Niveau de priorité de la tâche u
Δt_u^{comp}	Temps de calcul de la tâche u
$\delta t_u^{UL,c}, \delta t_u^{DL,c}$	Délai de communication de la tâche u sur le canal c
τ_u	Temps passé par la tâche u dans la file d'attente de calcul
t_u	Délai total de délestage de la tâche u
t_{max}^P	Délai de délestage maximal toléré par une tâche de priorité P
$T^{\lambda,s}$	Délai total de délestage pour toutes les tâches du scénario s
$R^{\lambda,s}$	Taux de succès de délestage des tâches pour le scénario s
\mathcal{M}/M	Ensemble/Nombre de véhicules
\mathcal{N}/N	Ensemble/Nombre de RSU
F_{CPU}	Fréquence d'horloge des processeurs des serveurs MEC
$l_{m,n}$	Distance entre le véhicule m et la RSU n

suivi, chaque message possède son propre numéro de séquence, horodatage et indicateur de taille. Nous décrivons à présent le reste de leur contenu. Un visuel de la structure des messages impliqués dans notre solution est fourni en Figure 3.2.

Lorsqu'un délestage est nécessaire, il est indispensable de fournir les données nécessaires à son calcul. Ainsi, dans un message de demande de délestage, un ensemble d'informations sont disponibles :

- **Number of Data Frames** : Indication du nombre de trames à transmettre afin de faire parvenir l'intégralité des données nécessaires au calcul de la tâche (*1 octet*)
- **Frame Number** : Numéro de la trame de données courante (*1 octet*)
- **Task Usage Value** : Quantité d'opérations nécessaires au calcul de la tâche telles que formulées dans 3.2 (*4 octets*)
- **Sender ID** : Identifiant du véhicule à l'origine du délestage, de sorte à lui renvoyer les résultats de la tâche à l'issue de son traitement (*1 octet*)
- **Priority** : Indice de priorité de la tâche délestée (*1 octet*)
- **Content** : Les données contenues dans la trame, devant totaliser γ_u une fois toutes les trames reçues (*K octets*)

Les messages de mise à jour individuelle sont des messages courts envoyés par les serveurs périphériques à chaque variation de leur quantité de ressources de calcul disponibles en vue

Seq. No.	Timestamp	Length	Nb of Data Frames	Frame No.	Task Usage Value	Sender ID	Priority	Content
Unsigned int 4 bytes	Simtime 64 bytes	Int 4 bytes	Byte 1 byte	Byte 1 byte	Int 4 bytes	Byte 1 byte	Byte 1 byte	String K bytes

Task data message from vehicle (payload size 80 + K bytes)

Seq. No.	Timestamp	Length	Server ID	Usage Value	Time to Future (optional)	Future Usage (optional)
Unsigned int 4 bytes	Simtime 64 bytes	Int 4 bytes	Byte 1 byte	Byte 1 byte	Simtime 64 bytes	Byte 1 byte

Usage update message from RSU (payload max size 139 bytes)

Seq. No.	Timestamp	Length	Server ID List	Usage List	Future Usage List (optional)	Time to Future List (optional)
Unsigned int 4 bytes	Simtime 64 bytes	Int 4 bytes	List of byte $2 * N - 1$ byte	List of byte $2 * N - 1$ byte	List of byte $2 * N - 1$ byte	List of simtime $65 * N - 1$ bytes

Recap message from BS (payload max size 68 + 71 * N bytes)

FIGURE 3.2 Structure des données utiles des différents messages impliqués.

de délestage, ou périodiquement en cas de non-utilisation prolongée. Les champs auxquels nous nous intéressons ici sont :

- **Server ID** : L'identifiant unique du serveur périphérique et de sa RSU (*1 octet*)
- **Usage Value** : La quantité de ressources disponibles, en pourcentage des ressources de calcul de l'équipement (*1 octet*)
- **Time to Future** (optionnel) : Si le mécanisme le prend en charge, temps estimé avant augmentation des ressources disponibles (*64 octets*)
- **Future Usage** (optionnel) : Si le mécanisme le prend en charge, future quantité de ressources disponibles après expiration du temps estimé précédemment (*1 octet*)

Les messages récapitulatifs généraux, que nous appellerons aussi balises de contexte, sont des messages plus longs envoyés par le serveur périphérique associé à la BS cellulaire suite à la réception d'une mise à jour individuelle. Les champs auxquels nous nous intéressons sont :

- **Server ID List** : liste récapitulative de chaque identifiant de RSU ($2N - 1$ octets)
- **Usage Value List** : liste récapitulative de toutes les quantités de ressources disponibles aux RSU, ordonnée de la même manière que la liste des identifiants. L'indice de chaque valeur correspond donc à celui de l'identifiant associé ($2N - 1$ octets)
- **Future Usage List** (optionnel) : Si le mécanisme le prend en charge, liste de toutes les futures quantités de ressources disponibles aux RSU ($2N - 1$ octets)
- **Time to Future List** (optionnel) : Si le mécanisme le prend en charge, liste des temps estimés pour chaque RSU avant d'atteindre la quantité de ressources annoncée ($65N - 1$ octets)

La somme des tailles maximales de chaque champ, incluant les champs obligatoires de numéro de séquence, horodatage et indicateur de taille, nous fait parvenir à la conclusion que les messages de mise à jour individuelle peuvent atteindre, au maximum, une taille de donnée utile de 139 octets. De même, dans le cadre d'une expérimentation restreinte à un quartier urbain, le nombre de **RSU** impliquées n'est pas estimé dépasser la dizaine, au vu de précédents travaux de mise en œuvre d'architectures **V2X** [99]. Ainsi, dans une situation où la valeur maximale de N est de 5 **RSU**, les données utiles des balises atteignent une taille maximale de 423 octets. Par ces calculs, nous estimons que les messages de contexte impliqués dans **CAVTOMEC** sont légers et ne représentent donc pas une charge significative sur le réseau.

3.4.2 Mécanisme

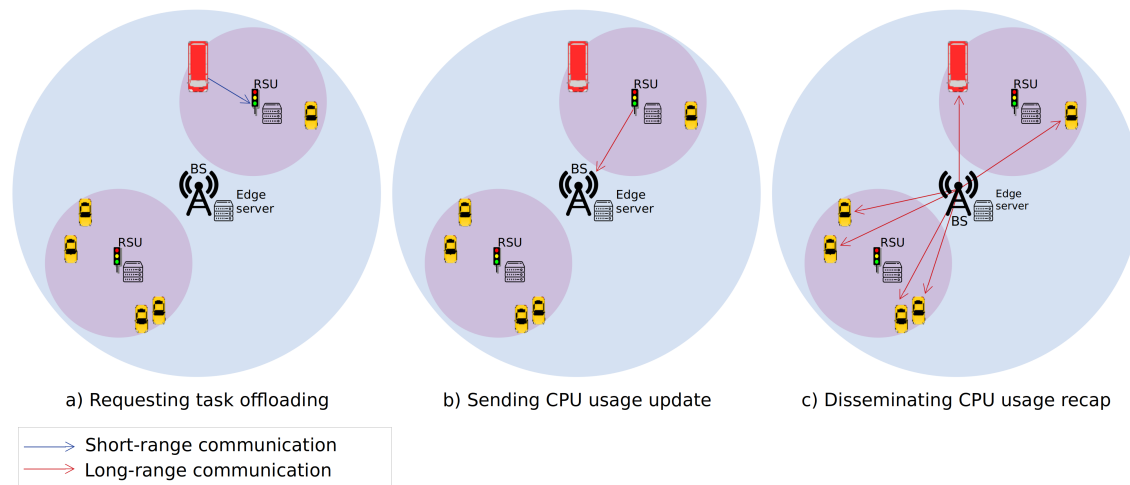


FIGURE 3.3 Mécanisme de transmission du contexte après réception d'une tâche.

L'architecture de mise à jour des données de contexte par les **RSU** est schématisée dans la figure 3.3, avec la visualisation des portées de transmission des **RSU** et **BS**. Lorsque la quantité de ressources de calcul disponibles dans un serveur **MEC** évolue, suite à l'approbation d'une demande de déchargement ou à la fin du traitement d'une tâche, la **RSU** correspondante met à jour le système avec les informations sur la nouvelle quantité de ressources libres, sous forme de pourcentage de ressources CPU. Le choix d'une telle métrique se justifie dans le contexte de notre étude par le fait que les déploiements actuels de smart cities utilisent du matériel provenant d'un fournisseur unique pour des raisons de maintenabilité [100]. Ainsi, toutes les **RSU** partagent les mêmes spécifications, y compris la fréquence d'horloge du processeur.

Le message de mise à jour individuelle est envoyé à la station de base par l'intermédiaire de l'interface cellulaire longue portée. De par la vaste couverture proposée par le

réseau cellulaire, il n'est pas nécessaire d'attendre qu'un hôte-relais se trouve à portée de communication D2D pour permettre à la charge utile d'atteindre sa destination lointaine. Le système est donc temps réel, quel que soit l'emplacement de la RSU. En outre, par analogie avec les RSU et leurs ressources de calcul associées, nous attribuons également un serveur périphérique au niveau de la BS pour le stockage et la gestion des données extraites des messages de mise à jour du contexte issus des RSU. Dès la réception du message, un traitement rapide fournit une actualisation de la connaissance globale de la disponibilité informatique de toutes les RSU de la zone, en maintenant une liste de clé-valeur associant l'identifiant de chaque RSU à un pourcentage de ressources disponibles. Un court message récapitulatif est alors construit, contenant le dernier état connu des ressources CPU de tous les serveurs MEC de la zone. Provenant de la BS elle-même, ces récapitulatifs sont transmis à tous les hôtes véhiculaires via le réseau cellulaire et reçues par les véhicules via l'interface longue portée ; la base de connaissances y est alors également mise à jour sous la forme d'une liste clé-valeur.

Les hôtes partagent désormais des informations sur la capacité de délestage des tâches du système MEC avec un très faible ajout de charge sur le réseau : comme mis en évidence en 3.4.1, la charge utile des messages récapitulatifs est inférieure au kilo-octet. Le mécanisme consistant à exploiter les informations d'occupation des ressources fournies par les RSU est schématisé sur la figure 3.4 : chaque véhicule produisant des tâches à délester combine les informations inhérentes à chaque tâche avec les données de contexte obtenues, afin de déterminer de manière autonome la destination de délestage la plus appropriée. Par ailleurs, afin d'illustrer le déroulement chronologique décrit, un diagramme de séquence de messages est présenté sur la figure 3.5, dans le cas d'un déchargement D2D d'une tâche sur l'interface courte portée ainsi que la transmission des récapitulatifs d'utilisation sur l'interface cellulaire longue portée.

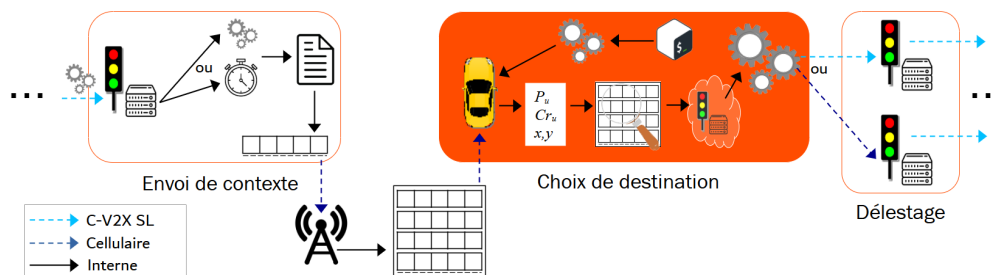


FIGURE 3.4 Chronologie de l'exploitation du contexte fourni par les RSU.

3.4.3 Comportement suite à la réception d'une tâche

Au niveau de la RSU, la réception d'une tâche délestée par un véhicule déclenche généralement l'un des deux comportements suivants : soit la tâche est exécutée immédiate-

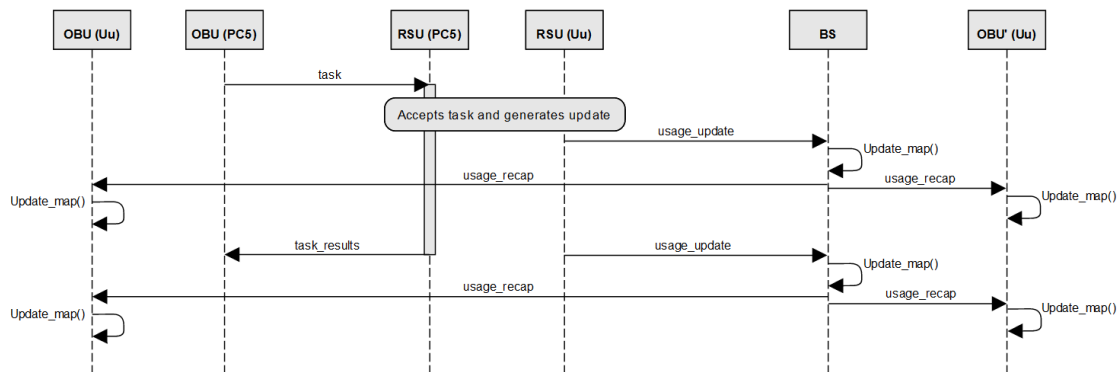


FIGURE 3.5 Diagramme de séquence : informations et délestage.

ment (calcul à la volée [101]), soit la quantité de ressources de calcul au niveau du serveur périphérique est trop faible, ce qui entraîne la mise en attente de la tâche reçue, le temps qu'une quantité suffisante soit libérée. Les deux comportements ont été exploités dans les architectures MEC, chacun apportant ses propres avantages en termes de performance ou de fiabilité, c'est pourquoi nous proposons deux variantes de notre solution. La première variante ne prend en charge que les calculs à la volée, et rejette les tâches entrantes quand les ressources disponibles sont trop faibles ; la seconde fournit aux serveurs la possibilité d'insérer ces tâches dans une file d'attente le temps que les ressources nécessaires soient libérées. Nous détaillons les deux fonctionnements dans la prochaine sous-partie.

Dans une situation de calcul à la volée, la RSU et le serveur périphérique associé doivent maintenir la connaissance du système global et des ressources disponibles en temps réel, en fournissant des mises à jour lorsqu'une tâche à déléster est reçue, ou lorsque les résultats sont renvoyés. Toutefois, dans le cas du calcul avec mise en attente, la quantité en temps réel de ressources disponibles est faussée, car elle masque l'impact des prochaines tâches une fois qu'elles ont été récupérées dans la file d'attente pour être exécutées par la suite. Par conséquent, les informations de mises à jour telles qu'établies dans le cas précédent doivent être complétées par des données provenant d'un autre champ de l'application : le temps estimé nécessaire pour terminer la tâche en cours et la quantité de ressources disponibles lorsque cette tâche sera achevée. De cette manière, les hôtes bénéficient d'une plus grande visibilité en prévision d'un futur délestage, afin de prendre en compte à la fois la disponibilité présente et la disponibilité future des ressources.

Par l'étude de ces multiples informations de contexte, les véhicules obtiennent la capacité de déterminer par eux-mêmes la destination la plus appropriée pour déléster leurs tâches, afin de réduire leur temps de traitement. De nombreuses directions peuvent être explorées, nous nous attelons donc à formuler le problème ainsi posé.

3.5 Choix de la destination

Dans cette section, nous nous concentrons sur le déchargement rapide et fiable des tâches de véhicules vers l'infrastructure. Notre objectif est de minimiser le délai total de déchargement des tâches tout en maximisant les taux de succès de calcul des tâches. Pour cela, considérant que notre système de délestage ne choisit qu'une cible pour une tâche, nous définissons la destination optimale pour une tâche donnée comme la destination garantissant la fin de la tâche dans des délais respectant les contraintes fixées par la priorité de la tâche, comprenant le délai de communication et le temps passé dans la file d'attente, le cas échéant.

Il convient de noter que, dans notre système, les tâches hautement prioritaires peuvent bénéficier d'un traitement préférentiel comparé aux tâches à faible priorité. En effet, en plus de viser à satisfaire les contraintes de délai, et conformément à nos motivations, notre objectif est de minimiser les délais de délestage pour les tâches très urgentes. Nous cherchons donc à élaborer un algorithme distribué entre les véhicules, capable d'optimiser le délestage des tâches au fil de leur création. Afin de parvenir à cette solution online, nous procédons à la formulation du problème de minimisation de délai étudié.

3.5.1 Problème de minimisation de délai

Soit E l'ensemble des tâches, divisé en trois sous-ensembles correspondant aux trois niveaux possibles de priorité : haute (E_H), indifférente (E_I) et basse (E_L). À chaque priorité est associée un maximum de durée totale de délestage : t_{max}^H , t_{max}^I , t_{max}^L . Nous formulons ainsi notre première fonction d'optimisation telle que :

$$(P1) : \min \sum_{u \in E_H} t_u^H \quad (3.3a)$$

$$\text{s.t. } t_u^H > 0, \quad (3.3b)$$

$$C_{r_u} \leq A_n, \quad \forall u \in E_H, \forall n \in [1 \dots N], \quad (3.3c)$$

$$t_u^H \leq t_{max}^H, \quad \forall u \in E_H, \quad (3.3d)$$

$$t_u^I \leq t_{max}^I, \quad \forall u \in E_I, \quad (3.3e)$$

$$t_u^L \leq t_{max}^L, \quad \forall u \in E_L \quad (3.3f)$$

Dans ce problème (P1), l'objectif défini dans (3a) consiste à minimiser la somme de t_u , le délai de délestage total, pour les tâches de priorité haute. Cette expression sous forme de somme est appropriée, étant donné que les délais des tâches sont relativement interdépendants en raison de la présence de files d'attente pour les tâches. La première contrainte (3b) impose la positivité stricte du délai total de délestage. La deuxième contrainte (3c) garantit qu'aucune tâche u ne peut être transférée à un serveur périphérique si les

ressources de calcul disponibles A_n de ce serveur (déterminées à partir du message de mise à jour) ne dépassent pas la quantité de ressources de calcul C_{r_u} nécessaire au calcul de la tâche. Les contraintes (3d) à (3f) sont les critères nécessaires à la sélection de la destination de délestage optimale telle que définie dans le paragraphe précédent : il est nécessaire de veiller à ce que le temps total de délestage d'une tâche soit strictement inférieur au temps de délestage maximal associé à sa priorité. Toutefois, conformément à notre modèle de système, il convient de formuler la notion de destination unique pour le délestage des tâches dans les véhicules. Nous cherchons donc à définir cet aspect au sein du problème.

3.5.2 Solution à une destination et un canal

Dans notre solution, une tâche ne peut être délestée que vers un et un seul serveur périphérique. De plus, à des fins de simplification, une seule interface peut être exploitée lors du délestage d'une tâche donnée. Les véhicules peuvent basculer d'une interface à l'autre entre chaque délestage, en tenant compte de l'état du système (autrement dit, le contexte) : la destination sélectionnée, la proximité par rapport à l'UE, et d'autres paramètres.

Nous définissons une matrice de décision évolutive D , ou "destination", représentant à un instant donné et pour tous les véhicules, la destination de délestage choisie. Nous définissons aussi une autre matrice de décision C , ou "canal", qui représente à un instant donné et pour tous les véhicules, le RAT choisi pour délester la tâche.

Supposons que, $\forall i \in [1 \dots M]$ et $\forall j \in [1 \dots N]$, $d_{i,j} \in \{0, 1\}$ et $c_{i,j} \in \{-1, 0, 1\}$. La valeur binaire $d_{i,j}$ traduit si oui ou non (respectivement 1 et 0) la destination j a été sélectionnée par le véhicule i . La valeur $c_{i,j}$ traduit si la tâche est délestée de i vers j par le biais de transmissions D2D ou longue portée (respectivement -1 et 1), la valeur 0 exprimant une absence de délestage vers cette destination.

$$D = \begin{pmatrix} d_{1,1} & \cdots & d_{1,N} \\ \vdots & \ddots & \vdots \\ d_{M,1} & \cdots & d_{M,N} \end{pmatrix}$$

$$C = \begin{pmatrix} c_{1,1} & \cdots & c_{1,N} \\ \vdots & \ddots & \vdots \\ c_{M,1} & \cdots & c_{M,N} \end{pmatrix}$$

Par conséquent, étant donné qu'aucun véhicule ne peut sélectionner plusieurs destinations de délestage à la fois, ni plusieurs canaux de délestage, les lignes de D ont une et une seule valeur unitaire, celles de C ont une et une seule valeur fixée à 1 ou -1 . Ainsi, soit e le vecteur colonne comprenant N valeurs unitaires (le nombre de RSU), et soit f le vecteur colonne comprenant M valeurs unitaires (le nombre de véhicules). De plus, soit Z la matrice des valeurs absolues de C . Avec S_D et S_C les vecteurs sommes respectifs des lignes de D et Z , nous obtenons les contraintes suivantes :

$$S_D = D \cdot e = \begin{pmatrix} \sum_{j=1}^N d_{1,j} \\ \vdots \\ \sum_{j=1}^N d_{M,j} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = f$$

$$S_C = Z \cdot e = \begin{pmatrix} \sum_{j=1}^N |c_{1,j}| \\ \vdots \\ \sum_{j=1}^N |c_{M,j}| \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = f$$

Il suit donc que, avec \odot le produit matriciel de Hadamard :

$$f^t \cdot D \odot Z \cdot e = \sum_{i=1}^M \sum_{j=1}^N |d_{i,j} c_{i,j}| = M$$

Cela signifie simplement qu'à chaque instant, si une tâche vient d'être créée ou est en attente de délestage, le nombre de décisions de délestage provenant des véhicules est égal à M , de sorte que toute sélection de destination est liée à une sélection d'interface unique.

En outre, du point de vue des délais, nous estimons que les communications D2D sur le sidelink doivent être privilégiées si aucun avantage clair ne peut être dégagé pour l'une ou l'autre décision. En effet, leur délai de communication associé est plus court que les transmissions cellulaires uplink-downlink, ce qui les rend plus appropriées pour les tâches hautement prioritaires et contraignantes. Par conséquent, nous définissons le paramètre c des variables $\delta t_u^{UL,c}$ et $\delta t_u^{DL,c}$, exprimant le canal de délestage sélectionné : $c = -1$ indique l'utilisation d'un canal sidelink, tandis que $c = 1$ indique l'utilisation d'un canal cellulaire. Nous formulons ensuite notre contrainte de la manière suivante, avec ϵ une constante strictement positive arbitrairement faible :

$$\delta t_u^{-1} \leq \delta t_u^1 - \epsilon, \quad \forall u \in E, \quad \epsilon \in \mathbb{R}^{+*} \quad (3.4)$$

Par la même, le problème (P1) peut être précisé en (P2) :

$$(P2) : \min \sum_{u \in E_H} t_u^H \quad (3.5a)$$

$$\text{s.t. } t_u^H > 0, \quad (3.5b)$$

$$C_{r_u} \leq A_n, \quad \forall u \in E_H, \forall n \in [1 \dots N], \quad (3.5c)$$

$$t_u^H \leq t_{max}^H, \quad \forall u \in E_H, \quad (3.5d)$$

$$t_u^I \leq t_{max}^I, \quad \forall u \in E_I, \quad (3.5e)$$

$$t_u^L \leq t_{max}^L, \quad \forall u \in E_L, \quad (3.5f)$$

$$\delta t_u^{-1} \leq \delta t_u^1 - \epsilon, \quad \forall u \in E, \quad \epsilon \in \mathbb{R}^{+*}, \quad (3.5g)$$

$$\sum_{i=1}^M \sum_{j=1}^N |d_{i,j} c_{i,j}| = M, \quad (3.5h)$$

$$\sum_{j=1}^N d_{i,j} = 1, \quad \forall i \in [1 \dots M], \quad (3.5i)$$

$$d_{i,j} \in \{0, 1\}, \quad \forall i \in [1 \dots M], \forall j \in [1 \dots N], \quad (3.5j)$$

$$c_{i,j} \in \{-1, 0, 1\}, \quad \forall i \in [1 \dots M], \forall j \in [1 \dots N] \quad (3.5k)$$

La contrainte (3.5g) exprime le choix de conception favorisant les communications sur le sidelink. Les contraintes (3.5h) à (3.5k) imposent l'unicité de destination et de canal de notre solution tels que nous venons de les décrire.

3.5.3 Transformation en problème d'optimisation

Nous généralisons le critère de délai maximal acceptable en tant qu'une unique contrainte, en définissant un vecteur de valeurs t_{max} , comprenant les valeurs de délai maximal associées aux trois niveaux de priorité. Soit donc :

$$t_{max} = [t_{max}^H, t_{max}^I, t_{max}^L] \quad (3.6)$$

Ainsi, en associant les précédents exposants H , I et L avec les valeurs numériques respectives 0, 1 et 2, nous pouvons combiner les contraintes (3.5d) à (3.5f) en une contrainte commune :

$$t_u^k < t_{max}(k), \quad \forall k \in \{0, 1, 2\} \quad (3.7)$$

Par la combinaison de ces transformations, la formulation précédente permet d'obtenir le problème de minimisation suivant (P3) :

$$(P3) : \min \sum_{u \in E_H} t_u^H \quad (3.8a)$$

$$\text{s.t. } t_u^H > 0, \quad (3.8b)$$

$$C_{r_u} \leq A_n, \quad \forall u \in E_H, \forall n \in [1 \dots N], \quad (3.8c)$$

$$t_u^k \leq t_{max}(k), \quad \forall u \in E, \quad \forall k \in \{0, 1, 2\}, \quad (3.8d)$$

$$\delta t_u^{-1} \leq \delta t_u^1 - \epsilon, \quad \forall u \in E, \quad \epsilon \in \mathbb{R}^{+*}, \quad (3.8e)$$

$$\sum_{i=1}^M \sum_{j=1}^N |d_{i,j} c_{i,j}| = M, \quad (3.8f)$$

$$\sum_{j=1}^N d_{i,j} = 1, \quad \forall i \in [1 \dots M], \quad (3.8g)$$

$$d_{i,j} \in \{0, 1\}, \quad \forall i \in [1 \dots M], \forall j \in [1 \dots N], \quad (3.8h)$$

$$c_{i,j} \in \{-1, 0, 1\}, \quad \forall i \in [1 \dots M], \forall j \in [1 \dots N] \quad (3.8i)$$

3.5.4 Preuve de NP-complétude

(P3) est essentiellement une transformation du problème de minimisation du nombre de tâches en retard sur plusieurs machines (noté P_M), qui est un problème NP-complet connu [102]. La situation consiste en un ensemble de tâches devant être exécutées sur plusieurs machines de manière individuelle afin de minimiser les temps de traitement tout en respectant les contraintes de durée. En reprenant l'équation 3.1, nous savons que même si le concept de ce problème est principalement concentré sur le traitement, c'est-à-dire lié à la portion incluse dans le terme Δt_u^{comp} de la formule de t_u^H , nous pouvons également considérer les délais de communication. En effet, les ressources de communication et les ressources de calcul dépendent toutes deux de la quantité de tâches prise en charge à un moment donné. En ajoutant cette contrainte au problème P_M , le problème d'optimisation (P3) est un problème NP-complet.

En considérant que tous les processeurs du système sont identiques, nous pouvons estimer que les facteurs de décision sont la route empruntée pour le délestage des tâches et la quantité courante de ressources de calcul disponibles. Nous intégrons ces deux facteurs dans notre heuristique et notre algorithme visant à minimiser le délai global de délestage des tâches critiques. Du point de vue de la communication, les variables $\delta t_u^{UL,c}$, $\delta t_u^{DL,c}$ et $\delta t_u^{SL,c}$ dépendent de multiples facteurs tels que la qualité du lien ou la distance à la destination $l_{i,j}$. Nous avons indiqué dans l'équation 3.12 que les canaux sidelink doivent être privilégiés en raison de leur meilleure qualité, bien qu'ils soient limités par leur portée de communication plus courte. Par conséquent, étant donné que les tâches hautement prioritaires doivent être transmises et exécutées dans des délais conformes aux critères, nous cherchons à les délester aussi souvent que possible sur des destinations proches, tant que leurs capacités de calcul disponibles garantissent un traitement rapide. La relation suivante de dépendance entre le délai et la distance émerge de telle sorte que, pour $i \in [1..M]$ et $j, k \in [1..N]$:

$$\delta t_u^{SL,-1}(l_{i,j}) < \delta t_u^{SL,-1}(l_{i,k}) \quad \text{où} \quad l_{i,j} < l_{i,k} \quad (3.9)$$

D'où :

$$t_u^H(l_{i,j}) < t_u^H(l_{i,k}) \quad \text{où} \quad l_{i,j} < l_{i,k}, \quad j \neq k \quad (3.10)$$

3.5.5 Heuristique

Afin de corrélérer la contrainte de distance (3.10) avec (P3) et son heuristique sous-jacente, nous supposons que tous les véhicules d'indice $i \in [1..M]$ sont capables de trier les indices $j \in [1..N]$ des **RSU**, dans un ordre croissant à partir des valeurs $l_{i,j}$. À l'échelle du système, cette supposition induit que les véhicules sont en permanence conscients de la position des **RSU** de l'architecture, y compris lorsque ces dernières sont hors de portée.

Enfin, en combinant la prise en compte des ressources de calcul disponibles, le caractère simple destination, le caractère simple canal, et l'avantage donné aux interfaces **D2D** courte portée, une variable de décision $d_{i,j}$ pour chaque véhicule i et **RSU** j peut être définie par la suivante, recalculée chaque fois qu'une tâche $u \in E_H$ nécessite délestage :

$$d_{i,j} = \begin{cases} 1, & \text{si } d_{i,k} = 0 \forall k \neq j \text{ et } A_j > Cr_u \\ & \text{et } l_{i,j} < l_{i,k} \forall k > j \\ & \text{ou si } d_{i,k} = 0 \forall k \text{ et } j = 1 \\ 0, & \text{sinon} \end{cases} \quad (3.11)$$

Dans 3.11, pour chaque véhicule, chaque **RSU** trié de la plus proche à la plus lointaine est étudiée pour déterminer sa quantité de ressources disponibles. Le premier candidat valide est sélectionné et conduit à l'aboutissement du mécanisme de sélection pour cette tâche. Si aucun candidat valide n'est obtenu alors que la valeur de l'indice j atteint N , alors la valeur 1 est attribuée à $d_{i,1}$. Nous entendons que la complexité de cette heuristique pour chaque véhicule est linéaire de l'ordre du nombre de **RSU** : $\mathcal{O}(N)$. Cette déduction provient de l'opération de comparaison qui a lieu pour chaque valeur possible de j , qui a lieu un maximum de N fois dans le pire des cas.

3.6 Implantation

Cette section détaille la mise en œuvre du mécanisme de sélection de destination de délestage en cohérence avec l'heuristique décrite précédemment, sous une forme algorithmique. Comme indiqué, nos deux principaux facteurs de réduction de délai sont la disponibilité et la proximité des ressources de calcul. Nous intégrons ces facteurs dans notre algorithme conscient des priorités de tâches, avec des pondérations variables en fonction de l'urgence de la tâche. Par exemple, pour les tâches hautement prioritaires, les véhicules à l'origine de délestage considèrent en premier lieu une **RSU** proche afin de privilégier les échanges **D2D**, et vérifient la disponibilité des ressources pour confirmer leur présélection.

Deux mécanismes émergent de cette approche, nous les dénommons Phases. La première Phase est concentrée sur la sélection de destination d'un point de vue de calcul immédiat, tandis que la seconde Phase intègre le comportement des files d'attente de calcul et des futures ressources disponibles en périphérie. Nous avons fait le choix de séparer ces deux

mécanismes afin de valider la contribution du produit de base dans un premier temps, puis de concevoir une infrastructure plus complexe dans laquelle notre solution peut être intégrée afin d'en améliorer les performances.

3.6.1 Délestage pour traitement immédiat

Le mécanisme de la Phase 1 se concentre sur les ressources de calcul actuellement disponibles au niveau des serveurs périphériques pour affiner la sélection des destinations de délestage. Nous avons fait le choix de fournir la position de toutes les **RSU** à tous les véhicules à tout moment, dans une liste $List_{pos}$, ce qui leur permet d'intégrer la distance par rapport aux éléments d'infrastructure dans leur algorithme de sélection. Le comportement dépend également de la préférence en matière de **QoS** à un moment donné, et donc de la priorité de la tâche.

Dans notre système, nous représentons une tâche véhiculaire comme une succession d'opérations de calcul devant être prises en charge au niveau du processeur d'un serveur périphérique. Chaque tâche est donc caractérisée par sa priorité, par le nombre d'opérations nécessaire à son traitement, par la contrainte temporelle permettant de la considérer comme traitée avec succès, et par la taille des données qui l'accompagnent. Ainsi, comme la gestion de la charge de calcul dans un serveur consiste en la répartition de portions de ressources d'un processeur, une étape d'estimation est nécessaire au niveau du véhicule afin d'approximer les ressources à requérir auprès du **MEC**, en fonction de la durée maximale de traitement autorisée pour cette tâche.

Par conséquent, lorsqu'une tâche u doit être délestée, le véhicule émetteur estime les ressources de calcul Cr_u nécessaires à la validation de ses contraintes, vérifie l'indice de priorité P_u et compare sa position actuelle à celle des **RSU** de la zone d'étude. Ayant reçu des récapitulatifs d'utilisation de la part du serveur **MEC** corrélaté à l'eNodeB, ils peuvent exploiter ces données pour sélectionner le meilleur candidat **MEC** pour le délestage des tâches par **V2I**.

Le détail du fonctionnement est décrit dans l'algorithme 1, comprenant deux procédures distinctes et concurrentes. Tout d'abord, le véhicule doit récupérer les données d'utilisation des **RSU** à partir des messages récapitulatifs reçus sur l'interface longue portée, et les utiliser pour mettre à jour leur liste d'utilisation $List_{usage}$. Dans cette liste, chaque numéro d'identification de **RSU** est associé à un pourcentage correspondant à sa proportion de ressources CPU disponibles, conformément à la structure décrite en 3.4.1. À chaque mise à jour, la table est triée selon deux critères : distance à la **RSU** (nommée $List_{usage}^{prox}$, tri croissant des valeurs $l_{i,j}$) et pourcentage de ressources libres (nommé $List_{usage}^{desc}$, tri décroissant des valeurs A_n). À la suite de ces opérations, chaque fois qu'une tâche doit être délestée, ces tables sont consultées. Les critères de sélection varient en fonction des exigences de **QoS**, déduites de la priorité P_u de la tâche u . Une tâche de priorité *élevée* nécessite un transfert aussi rapide que possible vers un serveur capable de l'exécuter à la volée, ce qui revient à

parcourir la table $List_{usage}^{prox}$ et à rechercher l'entrée d'utilisation correspondante. Puis, en partant de la **RSU** la plus proche du véhicule, la première à disposer de suffisamment de ressources est immédiatement chargée de la tâche. La priorité *indifférente* est insensible à la proximité aux **RSU**, ce qui permet de sélectionner uniquement la **RSU** disposant d'une disponibilité suffisante du processeur. Enfin, la priorité *faible* déclenche un effort pour trouver délibérément la **RSU** ayant la plus petite quantité de CPU disponible, afin de laisser la meilleure qualité de service aux tâches prioritaires. Lorsque la tâche est transmise au **MEC**, ses caractéristiques détaillées dans la figure 3.2 permettent au serveur de prendre sa décision d'acceptation et d'allocation de ressources le cas échéant.

Pour les besoins de la phase 1, si, dans l'un de ces cas, selon le tableau d'usages dont le véhicule dispose, aucune **RSU** ne dispose de suffisamment de ressources pour traiter la tâche, celle-ci est conservée au niveau de l'**OBU** dans l'attente de la prochaine mise à jour. Par conséquent, les serveurs **MEC** sont supposés calculer toujours les tâches à la volée, sans les mettre en file d'attente en cas de surcharge, ce qui implique $\tau_u = 0$. De cette manière, nous bénéficions d'une plus grande visibilité sur le processus de délestage. En réutilisant la formule 3.2 précédente, nous obtenons :

$$\Delta t_u^{comp} = \frac{Cr_u \times \epsilon}{F_{CPU}} \quad (3.12)$$

3.6.2 Délestage avec gestion de files d'attente

La présence d'un mécanisme de file d'attente influe grandement sur les performances et le délai de déchargement. Pour les expériences de la Phase 1, nous avons supposé que l'algorithme de sélection de la destination était indépendant des files d'attente, se concentrant uniquement sur les ressources de calcul disponibles au moment du délestage. De cette manière, les véhicules étaient les seuls à pouvoir retarder ou suspendre la programmation des tâches. Dans une configuration plus réaliste, l'infrastructure elle-même devrait gérer les tâches et les files d'attente du côté du serveur qui les calcule activement à mesure que de nouvelles tâches font leur entrée. Par conséquent, dans la Phase 2 de notre étude, nous mettons en place des files d'attente du côté du serveur périphérique et mettons à contribution l'intégralité des champs des messages de mises à jour d'utilisation de ressources à destination des véhicules. À la liste d'usage de ressources $List_{usage}$ s'ajoute donc une nouvelle liste $List_{future}$, correspondant à une estimation de la quantité de ressources prochainement disponibles. Cette liste a également vocation à être triée sur le véhicule par ordre de proximité géographique.

Du côté des **RSU**, une tâche reçue peut être calculée immédiatement si les ressources disponibles le permettent. Dans ce cas, un nouveau message de mise à jour est envoyé avec les nouvelles valeurs de ressources disponibles. Sinon, la tâche est mise en file d'attente pour une programmation ultérieure, et un message de mise à jour est construit avec les

Algorithm 1 Délestage pour traitement immédiat

Require: $List_{pos}, List_{usage}$

- 1: **procedure** UPDATE USAGE INTELLIGENCE
- 2: Upon timer expiration
- 3: **if** Usage Recap received **then**
- 4: Update List $List_{usage}$
- 5: New $List_{usage}^{prox} \leftarrow List_{usage}$ insertion-sorted by distance to vehicle
- 6: New $List_{usage}^{desc} \leftarrow List_{usage}$ insertion-sorted by available CPU
- 7: **if** Vehicle has a task u to send **then**
- 8: Select_Destination(P_u, Cr_u)
- 9: **end if**
- 10: **end if**
- 11: Reset timer
- 12: **end procedure**
- 13: **procedure** SELECT DESTINATION(P_u, Cr_u)
- 14: **if** P_u is HIGH **then**
- 15: **for** $k = 0, \dots, N - 1$ **do**
- 16: **if** $List_{usage}^{prox}(k) > Cr_u$ **then**
- 17: Send u to corresponding RSU
- 18: **else**
- 19: **if** $k == N - 1$ **then**
- 20: Wait for next update
- 21: **end if**
- 22: **end if**
- 23: **end for**
- 24: **else if** P_u is MEDIUM **then**
- 25: **if** $List_{usage}^{desc}(0) < Cr_u$ **then**
- 26: Wait for next update
- 27: **else**
- 28: Send u to corresponding RSU
- 29: **end if**
- 30: **else if** P_u is LOW **then**
- 31: **for** $k = N - 1, \dots, 0$ **do**
- 32: **if** $List_{usage}^{desc}(k) > Cr_u$ **then**
- 33: Send u to corresponding RSU
- 34: **else**
- 35: **if** $k == 0$ **then**
- 36: Wait for next update
- 37: **end if**
- 38: **end if**
- 39: **end for**
- 40: **end if**
- 41: **end procedure**

informations complémentaires sur l'état de la file d'attente. Les messages de mise à jour contiennent désormais les informations sur les ressources actuellement disponibles, les

valeurs futures à la fin de la tâche en cours et une estimation du temps d'achèvement de la tâche.

Conformément à notre objectif d'améliorer les performances des tâches hautement prioritaires, le système de gestion des files d'attente favorise les tâches sur une base de priorité. En d'autres termes, le paradigme *First Come First Serve* (FCFS) est mis de côté pour programmer les tâches les plus urgentes dès que possible. En pratique, chaque fois qu'une tâche est terminée, les ressources de calcul correspondantes sont libérées et la file d'attente est parcourue à la recherche d'une tâche dont les attributs sont $Cr_u < A_n$ et $Pu = HIGH$. C'est dans cet ensemble de tâches que FCFS intervient : la première tâche validant ces conditions est immédiatement prise en charge selon la méthode précédemment détaillée. Dans le cas où aucune tâche ne posséderait les attributs mentionnés, d'autres tâches moins prioritaires sont recherchées.

Dans l'algorithme 2, du côté du véhicule, le délestage des tâches conserve une dynamique similaire à celle de l'algorithme de la phase 1, avec la prise en compte supplémentaire des files d'attente des serveurs de périphérie avec le système de gestion susmentionné. Par conséquent, les tâches hautement prioritaires déclenchent toujours la consultation de $List_{usage}^{prox}$ puis de $List_{usage}^{desc}$ à la recherche d'un serveur périphérique disponible à proximité. Toutefois, si aucun n'est trouvé, la liste $List_{future}^{prox}$ est parcourue de la même manière, en considérant que la puissance de calcul des serveurs périphériques justifie d'envoyer des tâches prématurément dans une file d'attente courte. Si aucun candidat adéquat n'est trouvé, la tâche est envoyée à la RSU la plus proche pour être mise en file d'attente. Les priorités indifférente et faible prolongent leur comportement de Phase 1 d'une manière facilitée par la présence des files d'attente, qui annulent l'action d'attendre la prochaine mise à jour de l'utilisation, augmentant ainsi la fréquence du délestage.

3.6.3 Élaboration du simulateur

Outils de programmation

Nous mettons en place notre simulation dans le simulateur à événements discrets OMNET++ [103], sélectionné pour son accessibilité visuelle et sa bibliothèque intégrée iNET proposant les principaux protocoles de communication Internet. À des fins d'interopérabilité multi-RAT, nous intégrons également les éléments suivants :

- la bibliothèque SimuLTE dans sa version 1.2 [104] pour les communications mobiles cellulaires LTE et assimilées, incluant les hôtes véhiculaires pour leurs communications longue-portée ;
- l'outil et la bibliothèque Veins dans sa version 5.1 [105] afin de prendre en charge les communications véhiculaires à courte portée ;
- le simulateur SUMO 1.9.2 [106] en vue de simuler la mobilité ainsi que les visuels associés dans notre scénario urbain.

Ces bibliothèques ont été sélectionnées dans des versions permettant leur interopérabilité, ce qui nous mena à un obstacle dans la réalisation de la simulation du **multi-RAT**. En effet, les versions précédentes d'iNET permettaient aux hôtes d'exploiter plusieurs interfaces réseau ainsi que leurs couches réseau et transport dédiées. Cependant, les versions les plus récentes, comme celle que nous avons choisie, ont retiré cette possibilité, faisant transiter tous les paquets quittant toute interface réseau par des couches supérieures mutualisées. Étant donné que les trames fondées sur Veins et celles sur iNET dépendent de modules de code différents, ce comportement nous a amenés à programmer des méthodes d'harmonisation du contenu passant entre les couches protocolaires, de manière à présenter toutes les trames

Algorithm 2 Délestage avec gestion de files d'attente

Require: $List_{pos}, List_{usage}, List_{future}$

```

1: procedure UPDATE USAGE INTELLIGENCE
2:   while Every T time do
3:     if Usage Recap received then
4:       Update List  $List_{usage}$  and  $List_{future}$ 
5:       New  $List_{usage}^{prox} \leftarrow List_{usage}$  insertion-sorted by distance to vehicle
6:       New  $List_{usage}^{desc} \leftarrow List_{usage}$  insertion-sorted by available CPU
7:       New  $List_{future}^{prox} \leftarrow List_{future}$  insertion-sorted by distance to vehicle
8:       if Vehicle has a task  $u$  to send then
9:         Select_Destination( $P_u, Cr_u$ )
10:      end if
11:    end if
12:  end while
13: end procedure
14: procedure SELECT DESTINATION( $P_u, Cr_u$ )
15:   if  $P_u$  is HIGH then
16:     for  $k = 0, \dots, N - 1$  do
17:       if  $List_{usage}^{prox}(k) > Cr_u$  then
18:         Send  $u$  to corresponding RSU and end
19:       end if
20:     end for
21:     for  $k = 0, \dots, N - 1$  do
22:       if  $List_{future}^{prox}(k) > Cr_u$  then
23:         Send  $u$  to corresponding RSU and end
24:       end if
25:     end for
26:     Send to corresponding  $List_{usage}^{prox}(0)$  RSU for queueing
27:   else if  $P_u$  is INDIFFERENT then
28:     Send  $u$  to corresponding  $List_{usage}^{desc}(0)$  RSU
29:   else if  $P_u$  is LOW then
30:     Send  $u$  to corresponding  $List_{usage}^{desc}(N - 1)$  RSU
31:   end if
32: end procedure

```

de la même manière, quel que soit le **RAT** d'où elles proviennent ou vers lequel elles se dirigent. Nous avons ensuite modifié le mécanisme d'attribution des adresses IP afin de faciliter la sélection de l'interface réseau lorsque les applications envoient des données aux interfaces réseau.

Choix de développement

Comme nous l'avons expliqué précédemment, les paquets d'utilisation contiennent peu d'informations. Afin de simuler la transmission des balises de contexte après leur passage par l'eNodeB, nous avons donné à l'eNodeB en tant que tel la capacité d'héberger une application capable de créer des paquets et de les envoyer aux hôtes véhiculaires. En d'autres termes, dans le simulateur, l'eNodeB devient une partie du serveur **MEC**. Nous avons fait ce choix plutôt que de créer un serveur associé au point d'accès par un lien backhaul, car sa mise en place dans le simulateur aurait impliqué un passage supplémentaire du trafic dans le réseau de cœur, alors que ce passage a déjà lieu dès lors que les messages transitent par une **BS** cellulaire.

Une station de base n'est effectivement pas prévue pour héberger un serveur **MEC** dans les architectures classiques. Toutefois, dans une architecture reposant sur le **MEC V2X**, de tels changements restent raisonnables afin de limiter autant que possible le trafic dans le réseau de cœur. En effet, notre choix de conception de la simulation produit une quantité de trafic périphérique similaire à celle attendue de notre architecture, puisque le recours au Cloud est nul.

Ainsi, du côté des véhicules, nous avons développé trois applications. La première est l'application de réception des balises de contexte présentée précédemment. Elle partage ses résultats avec la seconde : l'application de génération de tâches intégrant nos algorithmes de délestage. Enfin, l'application de réception des résultats intervient une fois le traitement des tâches déchargées terminé, afin d'en capter les résultats et de produire des statistiques. Du côté des **RSU**, nous avons développé le mécanisme d'envoi de balises de contexte ainsi que l'application de réception et de traitement des tâches.

3.7 Évaluation

Dans cette section, nous évaluons la capacité de CAVTOMECC à optimiser le délestage de tâches de sorte à réduire le risque de surcharge des ressources de calcul périphériques. Une telle capacité permettrait de réduire les délais de délestage des tâches véhiculaires, notamment pour les tâches critiques, augmentant leur taux de succès. Nous commençons par détailler le matériel simulé sur chaque hôte, les types d'applications concernées et le volume de trafic attendu pendant la durée de la simulation. Nous comparons ensuite notre solution avec le mécanisme de délestage vers le serveur le plus proche, répandu pour sa facilité calculatoire et son aise de déploiement [107].

3.7.1 Configuration de l'évaluation

Au sujet des réseaux véhiculaires et des systèmes MEC associés, la plupart des études réalisées dans la littérature étaient jusqu'alors concentrées sur de grands réseaux peu denses, sur des circuits dégagés à vitesse constante ou sur de petites parcelles urbaines [27], [71]. En cohérence avec cette bibliographie et notre contexte d'étude, nous faisons le choix de mener notre évaluation sur un quartier urbain hébergeant une quantité d'équipements faible à modérée, de sorte à correspondre à un déploiement futur permettant de mener une étude comparative entre simulation et mise en œuvre réelle.

Concernant les tâches produites par les véhicules, une étude de la littérature ([108], [109]) nous permet de retenir une taille de données inférieure au méga-octet, afin de concentrer notre étude de performance sur la partie calcul plutôt que sur la partie communication. Ainsi, les bornes de complexité des tâches ont été adaptées aux capacités de calcul embarquées dans la plupart des équipements actuellement intégrés aux VEC, tout en correspondant aux paramètres mis en place dans de récents travaux ([109], [110]) : pour un processeur d'une fréquence de 800 MHz, nous estimons que les tâches délestées peuvent nécessiter entre 1 et 50% des ressources de calcul, selon l'application. À des fins de clarté dans les mesures, nous adoptons un modèle où le délestage partiel n'est pas autorisé, les tâches ne sont donc pas subdivisibles lors de leur prise en charge.

Pour des raisons de continuité tout au long du travail de thèse, nous avons fait le choix de nous affranchir des terrains d'expérimentation disponibles par défaut dans les outils de simulation, privilégiant la création complète de notre propre zone d'étude, tirée d'une cartographie du campus de Rangueil de l'Université Toulouse III Paul Sabatier. Nous avons commencé par en extraire une vue de dessus depuis le visualiseur de carte OpenStreetMap [111], puis par retirer l'ensemble des informations ne portant pas sur la disposition des routes. Nous obtenons un fichier XML dont chaque élément correspond à une voie, que nous importons dans SUMO le simulateur de mouvement. Afin de permettre au logiciel d'associer un véhicule à un itinéraire, nous devons associer à chaque véhicule une succession de voies connectées en au moins un point GPS, ainsi qu'un point de départ et un point d'arrivée. Une fois cette manipulation terminée, d'autres paramètres sont à déterminer tels que la vitesse maximale, l'accélération maximale, la quantité d'arrêts intempestifs, entre autres.

La figure 3.6 présente le terrain de jeu obtenu, indique la trajectoire suivie par les véhicules durant les premières étapes de l'évaluation, et met en évidence la position des RSU où se trouvent les serveurs MEC. Dans nos scénarios expérimentaux plus avancés, des serveurs MEC supplémentaires sont déployés en renfort des serveurs initiaux ; ils sont également visualisés.

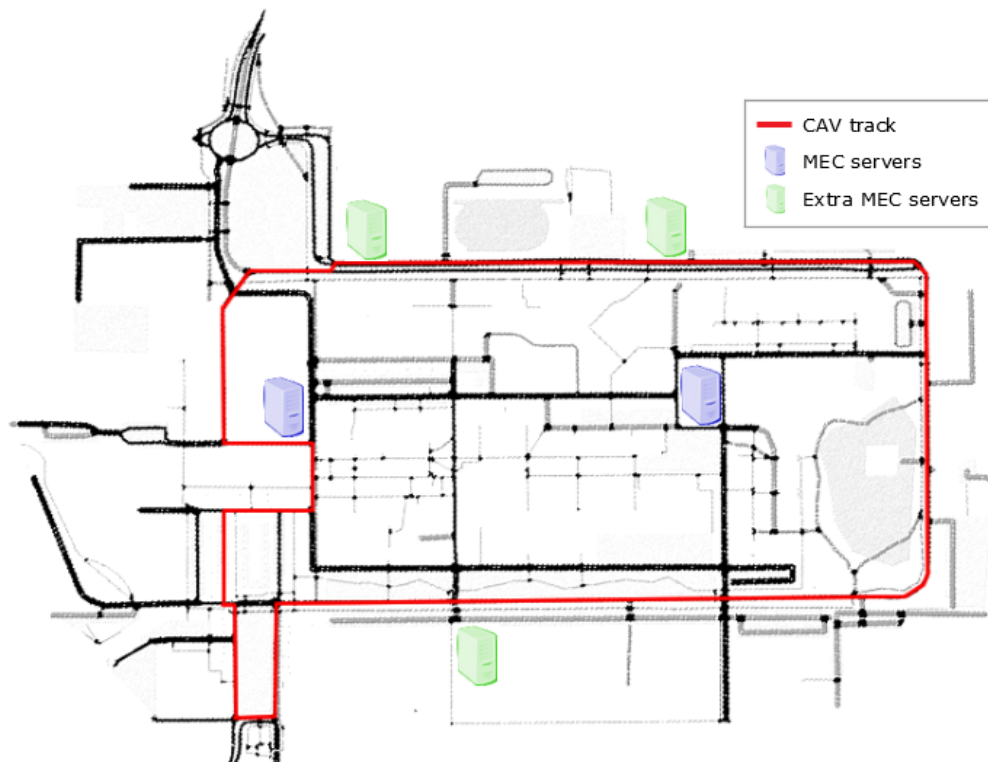


FIGURE 3.6 Vue simplifiée du circuit simulé.

3.7.2 Variation des paramètres

Il ne nous échappe pas qu'une architecture de communication et de calcul reste un outil mis à disposition d'équipements et d'applications en vue d'une amélioration de performances à haut niveau. Malgré le fait que la simulation d'applications véhiculaires quitte le champ de notre étude, nous nous devons d'étudier les performances de notre solution dans une situation réaliste où une multitude d'applications seraient en fonctionnement concurrent, engendrant leurs propres tâches à déléster au système MEC. Une certaine variabilité dans la taille des tâches, leur complexité et leur niveau de priorité doit donc être implanté dans nos scénarios d'évaluation.

Les véhicules produisent de nouvelles tâches selon une fréquence qui suit une loi de Poisson de paramètre λ . Chaque tâche u est délestée avec une quantité de données de taille γ , tirée uniformément entre les contraintes γ_{min} et γ_{max} . La complexité de la tâche Cr est proportionnelle à la taille des données de la tâche, entre les bornes Cr_{min} et Cr_{max} , et exprimée en pourcentage de ressources de calcul. Compte tenu de la relativement petite quantité d'hôtes considérée, nous estimons qu'une valeur de λ fixée à 1 Hz induira probablement une charge suffisamment faible sur les canaux et les ressources de calcul pour transmettre les données d'une tâche dans une direction puis l'autre, permettant le traitement de la tâche délestée avant que le véhicule demandeur n'essaie d'en déléster un

autre. En revanche, si cette valeur venait à augmenter d'un facteur dix, ou si la quantité d'équipements produisant des tâches augmentait, alors le réseau et le processeur risqueraient probablement d'être sur-sollicités, ce qui aurait un impact sur les délais d'exécution des tâches et de transmission, et donc sur les taux de réussite.

La simulation vise donc à comparer les performances des deux solutions susmentionnées, un délestage vers le proche et un délestage optimisé par CAVTOMECC, selon de multiples valeurs de λ , γ_{min} , γ_{max} , Cr_{min} et Cr_{max} . En d'autres termes, notre objectif avec ces scénarios est de recueillir des informations sur, premièrement, l'utilité globale de CAVTOMECC par rapport à un mécanisme de délestage simple. Deuxièmement, en exploitant différents paramètres liés au délestage, tels que la fréquence d'émission et la complexité des tâches, les performances de notre solution sont testées dans des conditions multiples. Plus les tâches sont fréquentes et complexes à traiter, plus le contexte devient difficile. Des groupes de mesures émergent de ces expériences, distingués par le type de mécanisme de délestage utilisé, la valeur de λ , et la complexité calculatoire des tâches délestées. Le tableau 3.2 résume nos paramètres de simulation.

TABLEAU 3.2 Paramètres de simulation

Name	Value
Nombre de répétitions	30
Durée de simulation (s)	1500
Valeurs de M	{5 ; 10 ; 20}
Valeurs de N	{2 ; 5}
Nombre d'eNodeB	1
Fréquence d'horloge MEC (MHz)	800
Vitesse maximale VAC (km.h ⁻¹)	30.0
Contrainte de délai pour succès (s)	20.0
Valeurs de k	{L ; I ; H}
Valeurs de λ (Hz)	{1 ; 5 ; 10}
Valeurs de λ^* (Hz)	{1 ; 5 ; 10}
$\gamma_{min} - \gamma_{max}$ (kO)	[20 ; 200]
$Cr_{min} - Cr_{max}$ (%)	[1 ; 35]
$Cr_{min}^* - Cr_{max}^*$ (%)	[5 ; 50]

Ainsi, nous évaluons chaque solution s pour de multiples scénarios et valeurs de paramètres, afin d'obtenir des conclusions sur les performances de CAVTOMECC selon les critères mentionnés précédemment. Chaque scénario est répété 30 fois avec réinitialisation des générateurs de valeurs pseudo-aléatoires. Les métriques obtenues et leurs interprétations possibles sont les suivantes (voir Tableau 3.1) :

- $A_n^{\lambda,s}$: Quantité de ressources de calcul disponible sur chaque serveur MEC n . Ces valeurs nous permettent d'observer le partage de charge entre RSU.
- $\sigma_n^{\lambda,s}$: Écart-type des quantités de ressources de calcul disponibles sur chaque serveur

MEC n , estimé par moyenne glissante. Ces valeurs nous permettent de vérifier si les ressources mises à disposition sont utilisées de manière régulière ou erratique, donnant une visibilité pour l'éventuelle arrivée de tâches critiques.

- $R^{\lambda,s}$: Taux de succès des tâches délestées. Ces valeurs nous permettent de mesurer l'impact global de notre solution à haut niveau, en prenant en compte les multiples générateurs de délai, qu'ils soient liés aux communications ou au calcul.
- $T^{\lambda,s}$: Délai total de délestage pour toutes les tâches d'un scénario, tel que défini dans l'équation 3.1, incluant donc les temps de transmission, d'attente au niveau du serveur et de calcul. Ces valeurs nous permettent d'étudier l'éventuelle régularité des délais de traitement tout au long du scénario, ainsi que l'impact de la présence ou non de files d'attente.

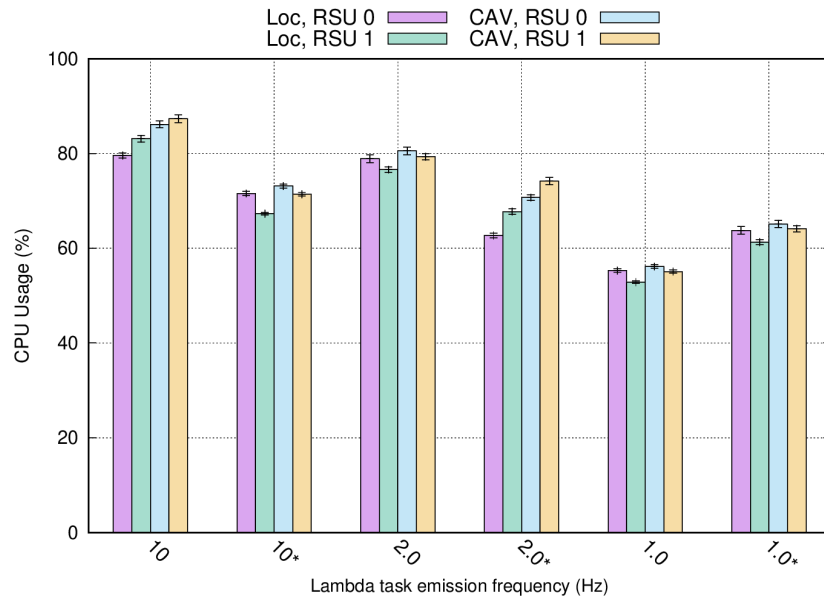
La présence d'une astérisque (*) dans l'intitulé ou la légende d'une courbe signifie que les données concernées ont été relevées dans un scénario où les tâches possèdent des complexités de calcul plus élevées, requérant donc une plus grande quantité de ressources **MEC**.

3.7.3 Intervalles de confiance

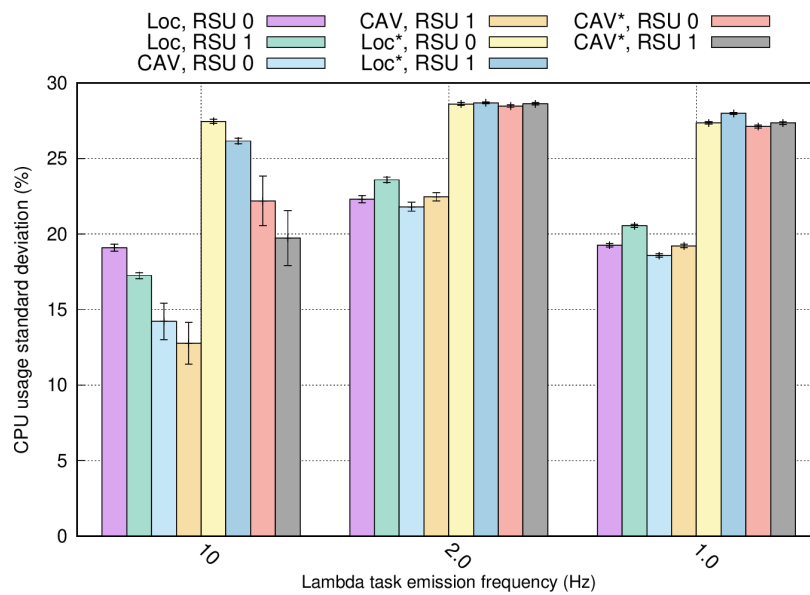
Pour obtenir des intervalles de confiance satisfaisants, nous avons utilisé une méthode de réplication. Avec 30 répétitions par scénario, nous obtenons des intervalles de confiance à 95 % pour nos histogrammes. Nous prenons également en compte la présence de tels intervalles dans notre interprétation des figures obtenues. Les intervalles de confiance nous permettent de tirer des conclusions fiables des données extraites de nos expériences.

3.7.4 Utilisation des ressources

La figure 3.7 concerne l'utilisation des ressources de calcul pour la Phase 1 de notre solution, afin que la gestion des files d'attente n'influence pas nos mesures. La figure 3.7(a) présente une comparaison de la charge du processeur moyenne entre les **RSU**, regroupées pour les deux scénarios, en faisant varier les valeurs de λ et la complexité des tâches. La figure 3.7(b) présente les mesures de l'écart-type de la charge du processeur entre les **RSU**, regroupées pour les deux scénarios et pour les deux niveaux de complexité étudiés, en faisant varier les valeurs de λ . La figure 3.7(a) montre que les serveurs de **MEC** sollicitent davantage de ressources de calcul lorsque les valeurs de λ sont élevées, en particulier pour les tâches moins complexes. Ce comportement s'explique par le fait que les tâches peu complexes sont plus susceptibles d'être acceptées par le serveur **MEC** sans lui faire atteindre la pleine charge de son processeur. En outre, la fréquence d'émission plus élevée des tâches induit une part de traitement parallèle, tandis qu'une valeur de λ égale à 1 Hertz représente un faible nombre de tâches à traiter en même temps, ce qui nécessite moins de ressources. La réciproque s'applique dans le cas où $\lambda^* = 10$: les tâches ayant des besoins en ressources



(a)



(b)

FIGURE 3.7 Utilisation des ressources de calcul, pour chaque RSU, $M = 5$, $N = 2$.

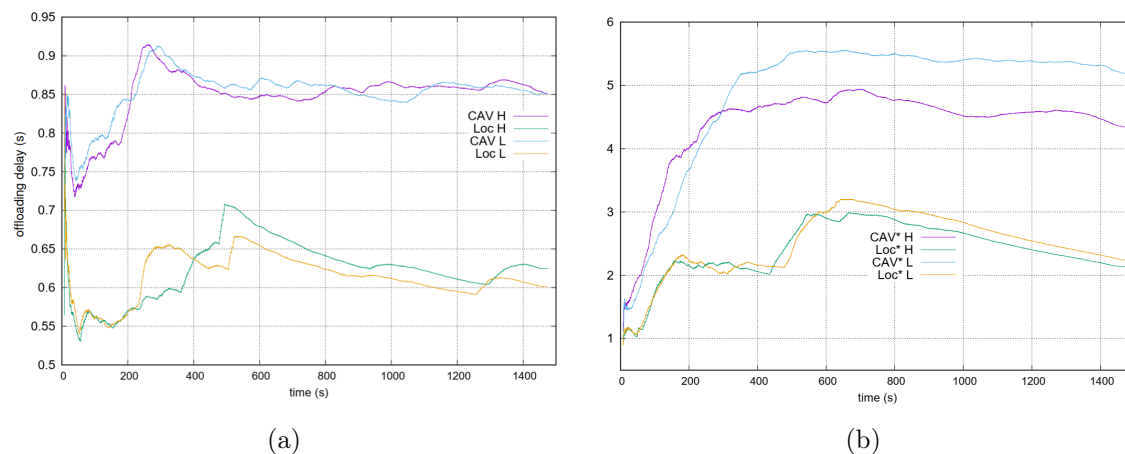


FIGURE 3.8 Délai de traitement total : $T^{1.0,s}$ (a) et $T^{1.0,s,*}$ (b), par priorité de tâche, $M = 5$, $N = 2$

élevés sont moins susceptibles d'être acceptées par le serveur MEC afin d'éviter la surcharge, même si une certaine quantité de ressources informatiques est encore disponible. Cependant, nous constatons que notre mécanisme atteint des niveaux d'utilisation du processeur plus élevés que le mécanisme de base, en particulier pour les tâches plus complexes.

La figure 3.7(b) met en évidence des valeurs de σ élevées pour toute solution et pour toute RSU, pour des valeurs faibles de λ et des complexités élevées. Ce phénomène s'explique par le fait que les serveurs périphériques ne sont pas actifs à tout moment lorsque de petites quantités de tâches sont engendrées, mais que les exigences élevées de ces tâches mènent à l'allocation de quantités massives de ressources lorsqu'elles sont acceptées, ce qui entraîne des fluctuations dans le fonctionnement du processeur. Cependant, lorsque $\lambda = 10$, une différence claire peut être notée, notre solution obtenant des valeurs d'écart-type 25% inférieures à celles de la solution fondée sur la localisation. L'utilisation du processeur est donc plus stable.

Pour résumer, la figure 3.7(a) permet de conclure que notre solution permet de mieux répartir la charge entre les serveurs périphériques, ce qui les rend moins exposés aux surcharges ou aux périodes où la puissance de calcul disponible est faible. De plus, les niveaux d'utilisation atteints sont plus élevés, indiquant le traitement simultané d'un plus grand nombre de tâches et donc une meilleure efficacité du délestage. En outre, la figure 3.7(b) nous permet de conclure que l'utilisation du processeur de chaque serveur MEC fluctue moins avec le temps dans notre architecture. Cela signifie que le système est capable de réagir aux fluctuations des besoins des tâches. L'exploitation des ressources est stabilisée, ainsi, à tout moment, la visibilité fournie aux véhicules permet aux tâches urgentes d'être déchargées immédiatement, accélérant leur traitement.

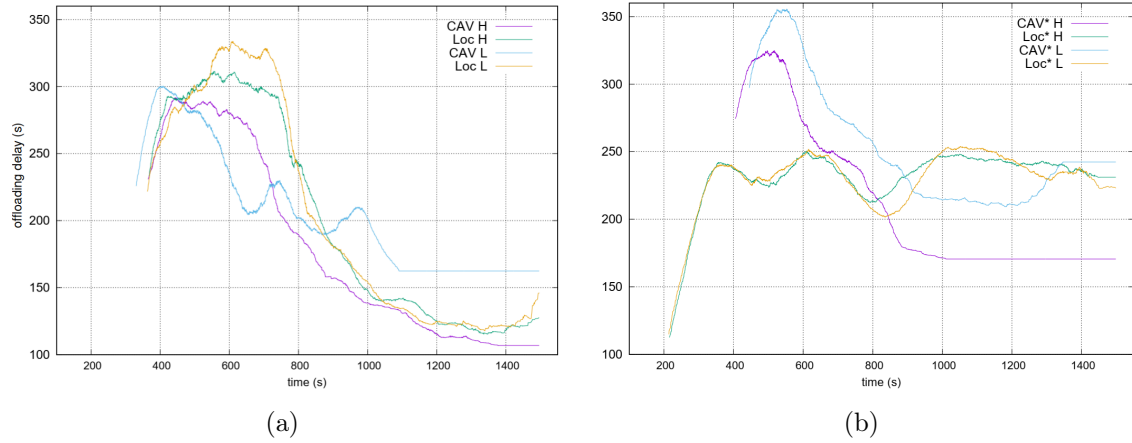


FIGURE 3.9 Délai de traitement total : $T^{10,s}$ (a) et $T^{10,s,*}$ (b), par priorité de tâche, $M = 5$, $N = 2$

3.7.5 Temps d'exécution

La figure 3.8 montre l'évolution du délai d'exécution des tâches pour les deux solutions s pendant la durée de la simulation $T^{\lambda,s}$, en fonction du niveau de priorité des tâches : H signifie élevée et L signifie faible. Les évolutions respectives de $T^{1.0,s}$ et $T^{1.0,s,*}$ sont représentées graphiquement dans les sous-figures 3.8(a) et (b). De la même manière, la figure 3.9 montre les évolutions de $T^{10,s}$ et $T^{10,s,*}$. Les paramètres fixés sont $M = 5$ RSU et $N = 2$ véhicules.

Tout d'abord, une grande différence dans les valeurs de T peut être notée en fonction de la valeur fixée pour λ . Lorsqu'une tâche est engendrée chaque seconde en moyenne, la charge du réseau est minimale pour communiquer avec la RSU, et la charge induite sur le processeur laisse une marge considérable, comme le montre la figure 3.7. Par conséquent, les tâches déchargées mettent peu de temps à atteindre le serveur périphérique et à y être calculées, ce qui entraîne des délais de délestage inférieurs à la seconde pour des valeurs de γ faibles. Avec des tâches plus complexes (figure 3.8(b)), $T^{1.0,s,*}$ augmente en raison de la taille plus importante des données à transmettre lors du délestage, ainsi que de la charge sur le processeur légèrement plus élevée au niveau des RSU. En outre, nous observons déjà une différence de traitement par CAVTOMECC des tâches prioritaires et non prioritaires, produisant un écart de délai allant jusqu'à 0,85 secondes, soit près d'un cinquième des valeurs maximales mesurées. Toutefois, les valeurs restent de l'ordre de la seconde. Dans la situation illustrée par la figure 3.8, l'utilisation de communications V2N induit un retard, visible lorsque de petites quantités de tâches sont générées, mais largement compensé par les meilleures performances obtenues lors de la gestion de quantités plus importantes de tâches, comme nous allons à présent le constater.

Avec une fréquence de délestage 10 fois plus élevée, les délais atteignent des valeurs élevées. En effet, l'absence de files d'attente dans la Phase 1 de notre solution provoque

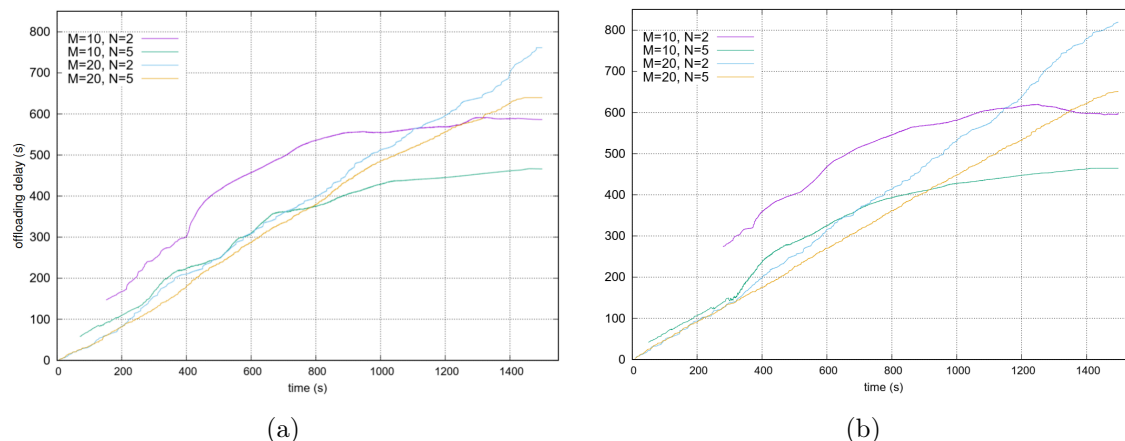


FIGURE 3.10 Délai de traitement total : $T^{10,s}$ (a) et $T^{10,s,*}$ (b), $s = CAVTOME C$

une longue attente des tâches au niveau des véhicules avant leur envoi, voire des rejets au niveau des serveurs MEC. Cela dit, sur la figure 3.9, il devient manifeste que notre solution produit un traitement différent selon le degré de priorité de la tâche, favorisant les transmissions rapides des tâches à haute priorité au détriment des tâches à faible priorité. L'écart peut atteindre jusqu'à 30%. En revanche, la solution fondée sur la seule localisation traite toutes les tâches de la même manière, comme le montre la similitude entre les tracés des priorités L et H. Nous remarquons également, une fois l'étape transitoire franchie, que notre mécanisme fournit des délais de délestage bien plus courts pour les tâches hautement prioritaires et de haute complexité (24% de réduction par rapport au scénario de base), et qu'il maintient son avantage avec des complexités plus faibles, bien qu'avec un écart légèrement moins marqué (15% de réduction par rapport au scénario de base).

Afin de déterminer l'efficacité de CAVTOME C dans des situations diversifiées, nous effectuons ensuite des simulations avec d'autres quantités de véhicules et de RSU (en faisant varier les valeurs M et N), en fixant des paramètres de tâches comme $\lambda = 10$ Hz. Les résultats sont présentés dans la figure 3.10.

Sur la figure 3.10, il apparaît que le principal facteur influençant le profil de délai de délestage des tâches n'est pas le nombre de RSU, mais le nombre de véhicules présents dans le système. Un plus grand nombre de VAC induit un plus grand nombre d'applications, et donc des charges de données plus élevées dans le système. Les canaux subissent une plus grande quantité de trafic, les serveurs périphériques reçoivent plus de tâches à traiter, les ressources du processeur sont davantage consommées, ce qui finit par allonger les temps d'attente et donc les délais de délestage des tâches.

En outre, pour les mêmes valeurs de M , CAVTOME C obtient des délais favorables aux architectures avec un plus grand nombre de serveurs périphériques. En effet, il était entendu dans l'algorithme de décision de la phase 2 qu'un panel plus large de RSU permettait de prendre des décisions plus appropriées. En outre, la figure 3.6 montre qu'avec l'inclusion de

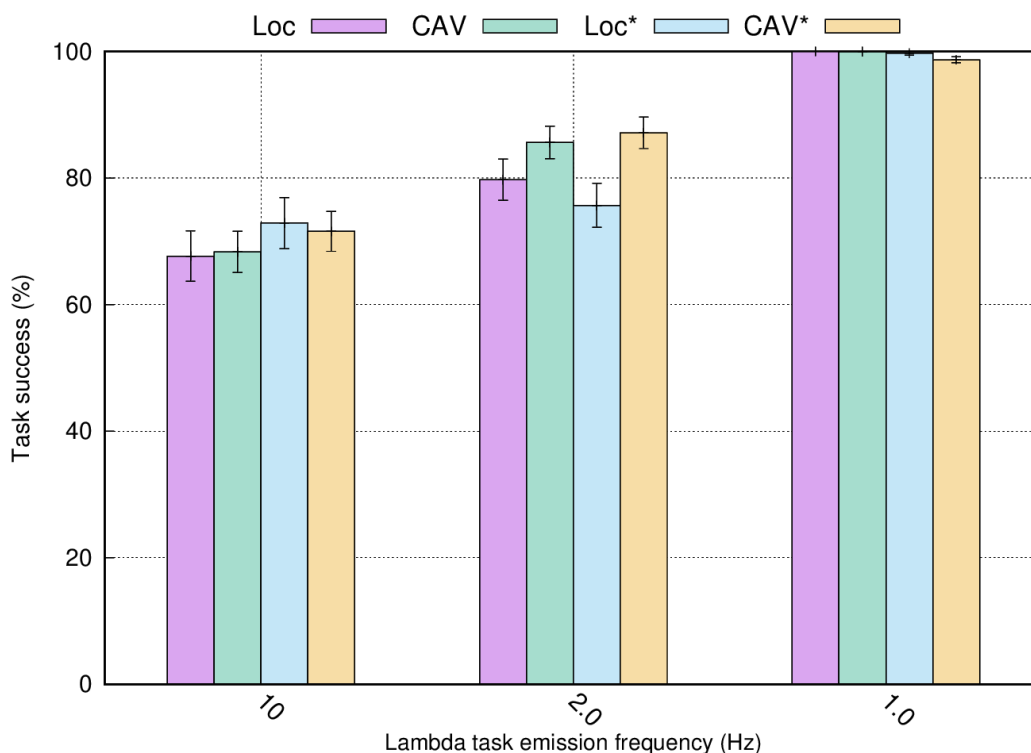


FIGURE 3.11 Taux de succès des tâches délestées, $M = 5$, $N = 2$.

serveurs de périphérie supplémentaires, N passant de 2 à 5, les véhicules circulant le long de la voie atteignent fréquemment des emplacements à portée D2D de plusieurs serveurs de périphérie en même temps. Dans ces situations, la décision optimisée de délestage peut sélectionner le serveur périphérique le plus disponible sans avoir à compromettre ce choix par des distances de communication sensiblement plus longues, ni en choisissant systématiquement le plus proche au risque de le surcharger.

Une similitude entre les deux sous-figures peut être identifiée, tant au sujet des profils de courbes que des valeurs mesurées. Ce comportement exprime la réduction de l'impact de la complexité des tâches sur les valeurs globales du délai, qui évoluent de la même manière et restent dans le même ordre de grandeur : l'importance de la complexité dans le délai de délestage est atténuée par d'autres facteurs tels que la densité de population, le temps d'attente ou le délai de communication.

3.7.6 Taux de succès

La figure 3.11 présente $R^{\lambda,s}$, le taux de réussite moyen des tâches d'un scénario, selon de multiples valeurs de paramètres. Les intervalles de confiance nous permettent de déduire que pour une fréquence intermédiaire d'émission de tâches ($\lambda = 2,0$, une nouvelle tâche est émise toutes les demi-secondes en moyenne), CAVTOMECC obtient de meilleurs taux de réussite des tâches de 6 points de pourcentage pour les tâches non complexes, et de 12

points pour les tâches complexes. Cette observation met en évidence deux contributions : premièrement, dans notre architecture, les tâches tendent à moins souvent subir de l'attente ou des rejets de la part des **RSU**, ce qui leur permet d'être traitées à temps. Deuxièmement, les transmissions sont plus fiables dans l'ensemble, ce qui évite les retransmissions, à l'origine de forts délais. Ainsi, selon les paramètres du système, l'impact de notre solution se traduit par un maintien ou une augmentation du taux de satisfaction des délais pour les tâches déchargées.

3.7.7 Effet de la prise en charge des files d'attente

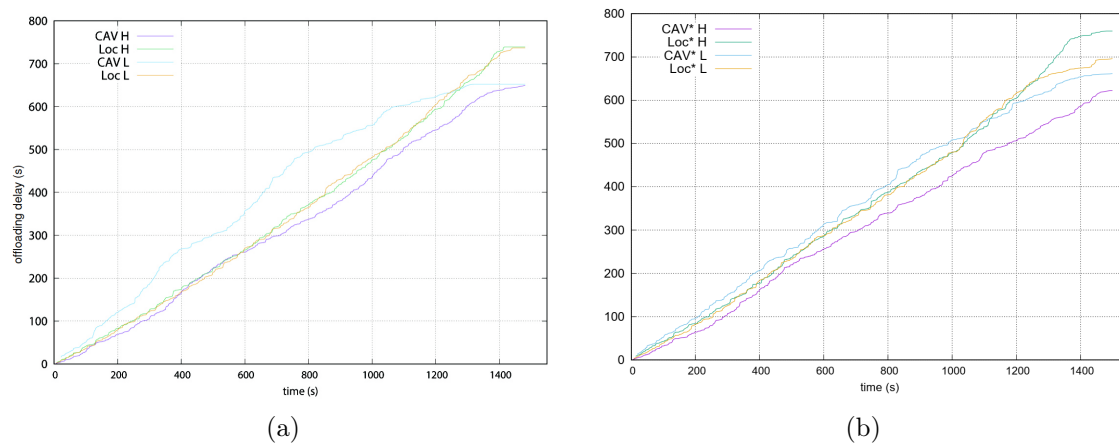


FIGURE 3.12 Délai de traitement total : $T^{10,s}$ (a) et $T^{10*,s}$ (b), par priorité de tâche, $M = 20$, $N = 5$, avec files d'attente.

Afin d'illustrer la robustesse de notre solution avec et sans la présence d'un mécanisme de file d'attente du côté des serveurs **MEC**, nous simulons à présent le comportement de CAVTOMECC dans une architecture où :

1. les files d'attente sont actives ;
2. les serveurs remplissent les champs optionnels de leurs balises (Future Usage, Time to Future, Future Usage List, Time to Future List) ;
3. les véhicules utilisent l'algorithme de Phase 2 pour délester leurs tâches.

Nous évaluons les performances du système dans des conditions de population plus dense que précédemment, soit avec les paramètres suivants : $M = 5$ **RSU**, $N = 20$ véhicules et $\lambda = 10$ Hz. Les résultats sont présentés dans la figure 3.12.

Nous constatons que lorsque le système est plus densément peuplé, la présence de files d'attente induit une forte augmentation du temps total de délestage, dû à la surpopulation des tâches relativement aux ressources disponibles en périphérie. En outre, le même compromis que celui observé sur la figure 3.9 avec les algorithmes de la Phase 1 apparaît à nouveau

avec les algorithmes de la Phase 2 : les tâches hautement prioritaires sont toujours exécutées plus rapidement avec CAVTOMECC qu'avec le délestage par proximité, au détriment des tâches faiblement prioritaires, dans les premières minutes de la simulation. De plus, après 20 minutes, les valeurs de délai favorisent clairement CAVTOMECC. Ces résultats peuvent s'expliquer par le mécanisme de gestion des files d'attente, qui permet aux tâches urgentes d'être toujours exécutées avant les autres, quel que soit le niveau de remplissage des files, ce qui lui donne un avantage sur les solutions fondées sur la localisation, qui peuvent soumettre certains serveurs MEC à une forte surcharge dans les scénarios denses.

Nous observons également que la différence de délai de délestage total entre les tâches plus complexes et les tâches moins complexes n'est pas aussi visible dans ces scénarios que dans la figure 3.8 ou la figure 3.9. Ceci s'explique par le temps que les tâches passent dans les files d'attente avant d'être programmées, ce qui réduit la proportion occupée par la première partie de la formule Δt_u^{comp} (3.2) dans le résultat de la mesure de délai, ce qui atténue l'impact de la complexité des tâches.

Ainsi, nous pouvons conclure que CAVTOMECC continue d'apporter des améliorations de performance globale une fois associé à un trafic dense, bien que les retards globaux soient mécaniquement voués à augmenter avec le nombre de véhicules générateurs de tâches. Cependant, la présence d'un mécanisme de gestion de files d'attente tenant compte des priorités des tâches permet d'atténuer les effets de ce trafic plus important, en favorisant la programmation des tâches urgentes, y compris en cas de charge plus soutenue.

3.8 Limites et discussion

Dans cette section, nous discutons des limites de l'implantation actuelle de CAVTOMECC et des pistes d'amélioration à étudier dans le cadre de travaux futurs.

Sensibilité au temps. Comme il peut être déduit de la description du système, une base de connaissances mise à jour du côté du véhicule signifie une sélection plus fiable de la destination, mais ne signifie pas qu'une carte datée mette en péril le bon fonctionnement du VAC. La demande de délestage de tâches peut cependant avoir besoin d'attendre plus longtemps que prévu avant d'être acceptée au niveau du serveur périphérique. Le fonctionnement fondamental du système ne dépend pas de la livraison ni de la réception en temps voulu des récapitulatifs d'utilisation de ressources. En outre, la propagation des messages de mises à jour et des récapitulatifs via les réseaux cellulaires à longue portée est soumise à des performances différentes, comme des débits de données plus faibles, par rapport aux transmissions de tâches par protocoles D2D. Cet écart de performance entre deux parties interdépendantes du système peut impliquer qu'un récapitulatif d'utilisation peut être reçu par les véhicules peu de temps après qu'une tâche vient d'être acceptée par

un serveur MEC, ce qui peut conduire à une sélection sous-optimale de la destination et à un délai de traitement potentiellement rallongé. Nous soupçonnons qu'un tel comportement pourrait être plus susceptible d'être observé, s'il l'est, dans des scénarios très denses ou dans des milieux particulièrement étendus, lorsque de plus grandes quantités de mises à jour d'utilisation et de récapitulations sont transmises.

Évaluation. Pour des raisons temporelles et techniques, malgré l'importance que nous avons accordée à la répétition des expériences ainsi qu'à la diversité des scénarios évalués, notre évaluation des performances de CAVTOMECC pourrait être étendue. La première extension pourrait être conduite dans le simulateur, en augmentant considérablement la quantité de véhicules et de serveurs périphériques, afin d'étudier le comportement de notre solution à l'échelle. La seconde extension considérable consisterait en une implantation réelle, afin de mesurer l'impact des phénomènes physiques imprévisibles tels qu'ils apparaissent sur le terrain, et ne peuvent pas être simulés de manière réaliste.

3.9 Conclusion

Dans ce chapitre, nous avons présenté CAVTOMECC, un système d'optimisation de délestage de tâches véhiculaires prenant en compte le contexte de disponibilités des ressources ainsi que le niveau de criticité des tâches. Cette solution aborde les thématiques fondamentales des réseaux véhiculaires telles que le délestage de tâches et la réduction des délais, en fournissant des mécanismes dédiés aux hôtes et aux serveurs périphériques afin de leur permettre, respectivement, d'améliorer leur sélection de destination de délestage et d'accélérer le traitement des tâches les plus critiques. Les résultats obtenus montrent que CAVTOMECC permet de délester les tâches véhiculaires avec une plus grande efficacité qu'un mécanisme de délestage classique, menant à des délais réduits, à un meilleur contrôle sur les ressources de calcul utilisées, et à des meilleurs taux de succès.

Suite à ces travaux, il nous est possible de fournir certains éléments de réponse à deux inquiétudes majeures déduites de l'état de l'art : les limitations intrinsèques des capacités de calcul embarquées dans le véhicule, ainsi que les difficultés liées aux communications extra-véhiculaires. Nous émettons la possibilité qu'un système de gestion de tâches combinant une prise en charge embarquée ainsi qu'une solution de délestage optimisé satisfasse ces deux limitations. Toutefois, dans ce travail, nous nous sommes concentrés sur la gestion des ressources de calcul, en supposant une gestion optimale des ressources de communication. Ainsi, dans la prochaine partie, afin d'étudier les deux principaux aspects des architectures véhiculaires, nous nous proposons d'aborder le sujet des ressources de communication réseau.

Deuxième partie

Stabilisation de connectivité V2X par handover proactif multi-RAT

La connectivité réseau des véhicules dans le contexte des smart cities motive le déploiement à grande échelle des réseaux mobiles 5G [112] et de leurs futurs successeurs. Améliorés par des protocoles de communication dédiés, les VAC peuvent proposer une variété de services liés aux échanges avec leur entourage, allant de la sécurité routière à la conduite à distance. La réduction des latences, la transmission de quantités de données plus importantes et l'amélioration de la gestion de mobilité contribuent à satisfaire les exigences de performance élevées des applications *Ultra Reliable Low Latency Communications (URLLC)*, notamment en réponse aux quantités massives d'informations engendrées par les capteurs embarqués dans les véhicules. Ainsi, dans un projet de multiplication des VAC, il est à prévoir qu'un nombre croissant d'équipements implique une plus grande quantité de capteurs, de données et de tâches à traiter. Cette situation est analogue au paradigme de l'internet des objets (IoT), où une quantité massive d'équipements produisant des données peut être regroupée ou distribuée en fonction de l'application [113], [114]. Étendu aux véhicules connectés et aux *smart cities*, ce système constitue l'IOV [115].

Nous avons constaté que le paradigme MEC s'adapte à ce futur contexte, avec des serveurs périphériques positionnés à proximité des utilisateurs pour une disponibilité rapide des ressources de calcul. Les temps de propagation des demandes de délestage sont réduits par rapport aux architectures Cloud, mais crée un besoin de connectivité stable pour assurer une livraison fiable et efficace des données, ainsi qu'un renvoi des résultats respectueux des contraintes de délai.

La connectivité des véhicules est assurée par de multiples interfaces et protocoles de réseau, mais dont le mécanisme de sélection reste un sujet d'étude. Chacun de ces RATs s'accompagne de ressources radio, de méthodes d'accès et de performances contextuelles spécifiques. Associées au MEC, des performances élevées en matière de communication réseau et de calcul peuvent être obtenues, ouvrant la voie à des cas d'usage V2X exigeants [69].

Le choix optimal d'interface dans le cas du multi-RAT dépend donc fortement des conditions dans lesquelles un véhicule nécessite de transmettre ses messages. Cette situation devient encore plus complexe dans un contexte urbain, les véhicules traversant des quartiers

aux structures variées, et rencontrant divers obstacles dressés entre leur *Access Point (AP)* et eux-mêmes, ce qui peut dégrader la qualité et la performance du lien pendant des périodes pouvant durer plusieurs secondes. Il est donc nécessaire de mener des recherches sur les moyens d'améliorer la sélection de *RAT* dans des conditions de communication particulièrement instables.

Dans cette partie, nous présentons un mécanisme de bascule verticale proactive, laissant aux véhicules la possibilité de sélectionner, dynamiquement et sans orchestration centralisée, l'interface réseau la plus appropriée entre *Dedicated Short-Range Communications (DSRC)*, 5G-V2X PC5, 5G-V2X Uu et LTE-V2X Uu, en fonction de la performance mesurée en une localisation donnée. Cette solution tire profit des messages de positionnement régulièrement diffusés entre usager de la route, et utilise les éléments d'infrastructure à des fins de partage de données de contexte à destination des véhicules.

Organisation de la partie

Cette partie comprend deux chapitres. Dans le premier chapitre, nous présentons un état de l'art des technologies de communication véhiculaire, ainsi que des architectures mises en place pour optimiser la connectivité d'appareils fortement mobiles face aux nombreux phénomènes pouvant dégrader le service. Nous dressons un récapitulatif des spécificités de chaque *RAT*, puis nous regroupons les techniques employées dans la littérature selon plusieurs catégories, afin de positionner notre travail.

Dans le deuxième chapitre, nous présentons CERES-V2X, notre solution de stabilisation de connectivité V2X dans un contexte multi-RAT urbain. Nous menons tout d'abord une étude des particularités d'un territoire urbain sur la fiabilité des performances réseau selon les technologies employées, puis nous proposons notre mécanisme de partage de connaissances et de prise de décision visant à répondre rapidement aux différents phénomènes physiques pouvant gêner, temporairement ou durablement, la propagation de messages critiques. Enfin, nous évaluons les performances de notre architecture par l'expérience, au sein d'un véhicule connecté mettant à contribution du matériel commercial de pointe dédié aux *smart cities*.

ÉTAT DE L'ART

4.1 La communication véhicule-à-tout

Dans l'univers foisonnant des *smart cities*, la grande panoplie des applications repose sur une diversité de méthodes de communication. Les véhicules connectés s'inscrivent intégralement dans ce contexte, de par la complexité des usages les concernant. À mesure qu'un véhicule évolue dans son milieu, la grande variété de situations interdépendantes nécessitent des prises de décision rapides et appropriées. Par conséquent, chaque véhicule se doit d'être en capacité de communiquer avec l'ensemble de son entourage impliqué dans l'utilisation de la route, et ce indépendamment de la nature de l'appareil interlocuteur et de la criticité de l'usage concerné. Nous décrivons les principales catégories de communications, ainsi que les situations globales dans lesquelles elles interviennent :

- Les services d'évitement des collisions physiques entre véhicules sont assurés par les communications [V2V](#).
- L'intelligence routière par récupérations de données de contexte centralisées intervient grâce aux communications [V2I](#).
- La sécurité des piétons est assurée par les communications [Vehicle-to-Pedestrian \(V2P\)](#).
- La plupart des services informatisés liés au Cloud peuvent être gérés par le réseau cellulaire mobile dans le cadre de communications [V2N](#).

Cette multiplicité des modes de communication est définie par le terme général de [V2X](#), ou véhicule-à-tout. La figure [4.1](#) représente les modes de communication abordés.

Les performances de ces échanges dépendent de plusieurs facteurs comprenant, sans s'y limiter, le débit, le délai et la fiabilité. Avant de s'intéresser à la conception d'une architecture de communication, la première décision qui portera un impact important sur ces paramètres est la sélection d'un [RAT](#), lui-même fonctionnant au sein d'un [Radio Access Network \(RAN\)](#) [[116](#)]. Chaque [RAT](#) possède ses propres caractéristiques et implications en

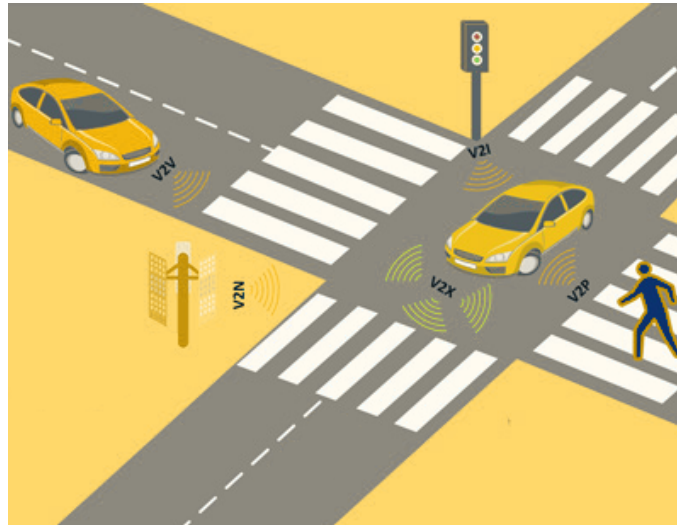


FIGURE 4.1 Visualisation d'échanges V2X.

matière de QoS et de QoE, ce pourquoi le choix de RAT est crucial lors de l'élaboration d'un réseau véhiculaire. Dans le sillage des nouvelles applications véhiculaires, nous pouvons distinguer trois catégories principales de technologies d'accès radio : les technologies fondées sur le Wi-Fi, les technologies cellulaires, et l'arrivée du Cloud-RAN.

4.2 Les technologies V2X héritant du Wi-Fi

4.2.1 802.11p, la première technologie radio fondamentalement véhiculaire

La première norme de communication V2X a été proposée en 2010, puis figée en 2011 sous la forme de l'amendement 802.11p au protocole *Wireless Local Area Network (WLAN)*. Cette spécification provient de la norme 802.11a et y apporte un ensemble de modifications destinées à gérer la mobilité, dans les deux couches inférieures [117] :

- **Paramètres fréquentiels** : la largeur de bande des canaux est réduite de moitié, passant à 10 MHz avec un espacement de sous-porteuse de 156,25 kHz, anciennement 20 MHz avec un espacement de 312,5 kHz en 802.11a.
- **Paramètres temporels** : la durée des symboles est doublée, passant à $6,2 \mu s$
- **Couche *Medium Access Control (MAC)*** : La méthode d'accès découle du mécanisme *Enhanced Distributed Channel Access (EDCA)* de la norme 802.11e, lui-même provenant du *Carrier Sense Multiple Access - Collision Avoidance (CSMA-CA)* du 802.11a. Les modifications apportées concernent notamment le temps d'écoute, de sorte à l'écourter pour les transmissions critiques telles que la voix et la vidéo. Les messages d'arrière-plan, eux, subissent un fort coefficient accompagné de tailles

agrandies de fenêtres de contention [118]. Enfin, afin de favoriser les bascules d'un point d'accès au suivant, les mécanismes d'association sont retirés. Le nouveau plan de gestion est constitué de deux entités protocolaires nommées *Physical Layer Management Entity* (PLME) et *MAC Layer Management Entity* (MLME).

- **En-têtes** : Dans le même souci d'économie de bande passante, afin de réduire la taille des paquets, les équipements ne comportent pas de mécanisme d'authentification ni de confidentialité. Il est prévu dans la norme que la sécurisation des échanges doit être mise en place dans les couches supérieures de la pile protocolaire [119].

Le protocole émergent du standard 802.11p est nommé ITS-G5 en Europe, ou **DSRC** dans le reste du monde. Il fonctionne sur une bande dédiée de 75 MHz dans la zone des 5,9 GHz, et permet des échanges de données directs entre les hôtes dans le nouveau système de communication *Wireless Access in Vehicular Environments* (WAVE), proposé par les normes IEEE 1609.3 et 1609.4 [120]. Ces normes comprennent l'ensemble de fonctionnalités du plan de gestion, nommé *WAVE Management Entity* (WME), permettant la mise en place et la stabilisation des communications D2D impliquées dans les transmissions V2V, V2I, V2P, mentionnées précédemment dans ce travail.

Les messages contextuels sont une autre innovation apportée par le WAVE et provenant des protocoles des couches réseau et transport fournis par 1609.3, nommés *WAVE Short Message Protocol* (WSMP). Ils permettent entre autres l'envoi périodique de messages *Cooperative Awareness Message* (CAM) et l'envoi sporadique de messages *Decentralized Environmental Notification Message* (DENM) [121]. Les CAM sont émis à destination de l'ensemble des appareils environnants, à une fréquence initiale de 1 Hz et pouvant augmenter jusqu'à 10 Hz dans le cas où une situation imprévue est détectée par le véhicule [122]. Leur contenu se limite aux informations de positionnement du véhicule (position, vitesse, direction, dimensions, rôle particulier). Les DENM fournissent des informations plus précises et sont émis plus rarement. En cas d'incident sur la route, ces nombreuses informations vont du type d'événement détecté jusqu'à la direction du trafic routier, et sont diffusées à l'ensemble des équipements proches. Ces deux messages de connaissance du contexte routier sont perçus comme les principaux vecteurs d'information des architectures V2X. Une représentation de la pile protocolaire de l'ITS-G5 est fournie sur la Figure 4.2

De nombreux travaux antérieurs, comprenant des essais expérimentaux, ont validé les performances du **DSRC** pour les communications entre véhicules dans des contextes de faible densité, de courte portée et de ligne de vue directe, ou *Line of Sight* (LOS). Toutefois, de nombreuses questions ont été posées concernant l'évolutivité et la robustesse de ce protocole, le DSRC subissant une grave détérioration de ses performances en raison de la faible largeur de bande du canal et de caractéristiques inadaptées à des scénarios denses [123]. Le besoin d'une technologie de communication V2X robuste, flexible et évolutive émergea, confirmé par la réallocation progressive des bandes de fréquence DSRC à d'autres technologies de communication véhiculaire [124].

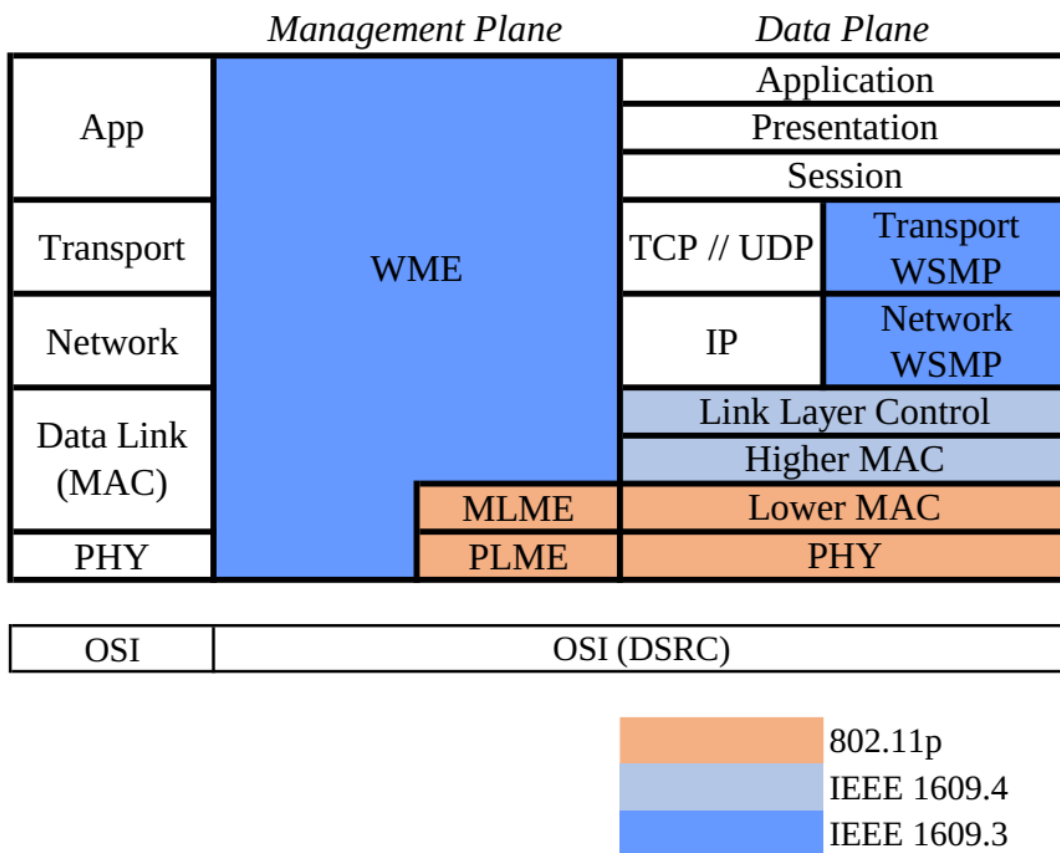


FIGURE 4.2 Structure protocolaire de la technologie ITS-G5.

4.2.2 802.11bd, un perfectionnement qui se fait attendre

Le standard 802.11bd est présenté comme le successeur du 802.11p, conçu pour une fiabilité renforcée et une gestion améliorée des grandes quantités de données. Pour atteindre le premier objectif, l'ajout de midambles permet un suivi plus précis des canaux face à la forte variabilité due à l'ajout de dynamique dans l'espacement inter-sous-porteuses. Le mécanisme de retransmission est également revu, afin d'y ajouter une dynamique en fonction de la congestion des canaux [125]. Enfin, le problème de la portée est abordé sous le même angle que le standard 802.11ax, également en cours de finalisation, avec l'ajout de *Dual Carrier Modulation (DCM)* permettant de diminuer le *Block Error Rate (BLER)* par la duplication des symboles transmis sur deux sous-porteuses distantes. Pour atteindre de plus grandes quantités de bande passante, les ondes millimétriques sont également à l'étude, de même qu'un mécanisme d'accès plus équitable tirant profit de deux largeurs de canal : 10 et 20 MHz, telles que proposées dans les standards 802.11n et 802.11ac [126].

Cependant, la norme 802.11bd peine à se concrétiser, en raison de l'étendue actuelle des équipements compatibles 802.11p. Les questions d'interopérabilité et de rétrocompatibilité sont encore à l'origine de nombreuses modifications. Toutefois, les premières analyses préliminaires sont encourageantes, avec des études montrant une fiabilité accrue de la transmission des messages et des temps de latence réduits de moitié par rapport à la norme 802.11p [127], [128]. De tels gains de performance pourraient grandement contribuer à la performance des services de sécurité routière.

4.3 Les technologies V2X héritant du cellulaire mobile

Les réseaux cellulaires ont toujours été la réponse à la question de la couverture du réseau dans les applications mobiles. Les réseaux véhiculaires ne font pas exception. Compte tenu de la distance couverte par une seule station de base (allant jusqu'à plusieurs kilomètres dans le cas des eNodeB [129]), il est raisonnable de supposer que les environnements urbains sont largement pourvus en accès aux réseaux cellulaires. Toutefois, ces réseaux sont déjà sollicités par un grand nombre d'équipements ainsi que par la grande majorité des utilisateurs mobiles. Par conséquent, ils ne constituent pas une solution pour le déploiement d'applications qui apporteraient une charge supplémentaire massive dans le réseau, sur ses canaux et dans le cœur. Néanmoins, ils peuvent servir de solution de repli pour les architectures de réseaux véhiculaires, par exemple dans le cas d'une défaillance soudaine de la connectivité pendant le calcul d'une tâche ne pouvant être traitée hors ligne. Dans ces situations, la capacité de se connecter à un canal radio à large couverture pourrait permettre la continuité du service, bien qu'en mode dégradé.

4.3.1 Le LTE-V2X

En 2017, avec l'arrivée de la Release 14 de 3GPP, le protocole *Cellular Vehicle-to-Everything (C-V2X)* fut introduit en tant que technologie de communication directe adaptée aux applications automobiles. Contrairement à la norme 802.11p, le *C-V2X* est une technologie de communication naturellement orientée vers la mobilité. En outre, le DSRC n'ayant pas connu d'amélioration depuis son déploiement à grande échelle cinq ans auparavant, la première itération d'un protocole V2X fondé sur le cellulaire, LTE-V2X, s'est confrontée aux limites fondamentales liées à la robustesse [130], par le biais de multiples modes de communication et la méthode d'accès *Sensing-Based Semi-Persistent Scheduling (SB-SPS)*, améliorant la prise en charge de la mobilité à grande vitesse. *C-V2X* peut sélectionner dynamiquement l'une des deux interfaces suivantes en fonction de l'application et du contexte réseau : Uu et PC5.

L'interface Uu fournit une connectivité cellulaire *V2N* par liaison montante/descendante (UL/DL), au prix d'un temps supplémentaire passé au niveau du point d'accès cellulaire (eNodeB ou gNodeB pour LTE ou 5G) et du cœur de réseau LTE [131] comme pour les communications mobiles classiques. Cette interface longue portée démarque grandement le V2X cellulaire du V2X hérité du Wi-Fi, en ceci qu'elle bénéficie de bandes de fréquences exploitables en parallèle de la bande dédiée aux transmissions à courte portée, ainsi que de ses propres mécanismes de modulation (*Single Carrier - Frequency Division Multiple Access (SC-FDMA)* pour UL et *Orthogonal Frequency Division Multiple Access (OFDMA)* pour DL) et de retransmission (*Hybrid Automatic Repeat reQuest (HARQ)*).

L'interface PC5 permet des transmissions D2D sur le *SL* dans la bande de fréquences de 5,9 GHz, bénéficiant de la gestion de canal du LTE : deux largeurs de canal sont sélectionnables dynamiquement parmi 10 MHz et 20 MHz, l'accès au support est géré en *SC-FDMA* pour réduire les interférences entre deux interlocuteurs mobiles. La complexité de l'interface PC5 réside dans son mécanisme d'ordonnancement, intervenant selon l'un de deux modes (voir figure 4.3 : par la station de base cellulaire (mode 3), ou bien de manière autonome entre les équipements eux-mêmes à l'aide de *SB-SPS* (mode 4)).

Ce dernier mode motive la plus grande partie des travaux de recherche sur le V2X, car il permet d'obtenir de faibles latences et de fonctionner hors des zones de couverture des points d'accès cellulaires. Le *SB-SPS* nécessite une écoute constante du support afin d'estimer les futurs blocs de ressources disponibles à partir des écoutes passées. Les messages de positionnement servent donc grandement à l'allocation de ressources, car la transmission de messages *CAM* est périodique. Parmi les blocs disponibles restants, une étude du *Received Signal Strength Indicator (RSSI)* permet de déterminer les blocs les plus satisfaisants, de sorte à les allouer à un nombre déterminé de transmissions à venir. Une fois ce nombre de transmissions écoulé, une nouvelle allocation a lieu. Ainsi, le mode 4 permet d'obtenir un service plus robuste et plus régulier qui convient aux applications de sécurité routière.

La flexibilité du *C-V2X* a conduit à ce que le mode 4 soit fortement étudié pour ses

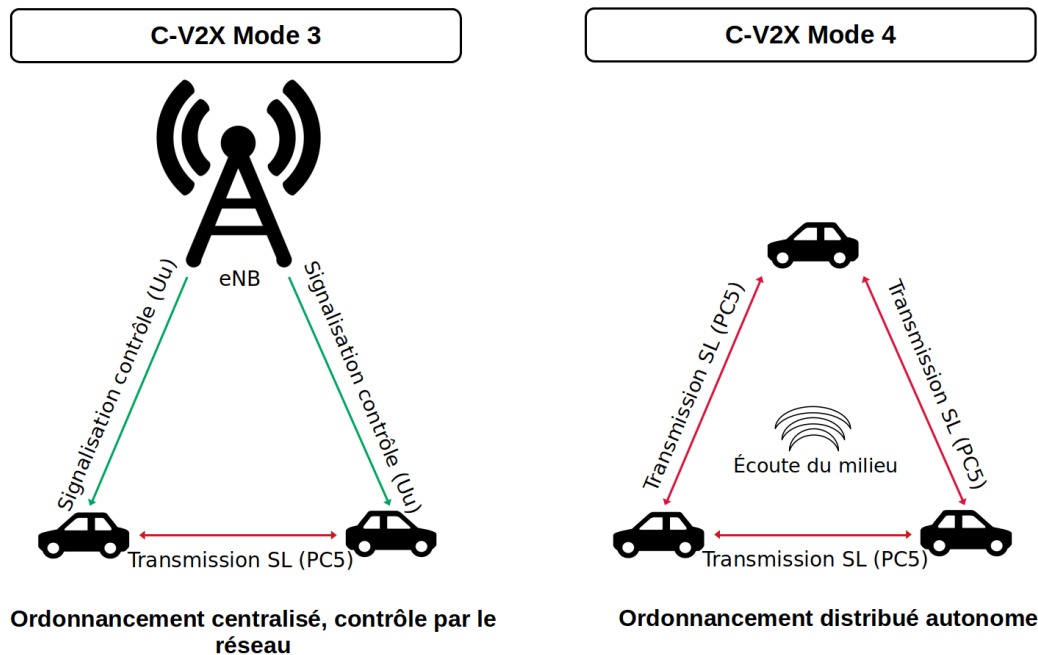


FIGURE 4.3 Modes de communication 3 et 4 du C-V2X.

objectifs de transmission directe, sans l'intervention d'un relais ou d'un orchestrateur. Cependant, peu de valeurs de paramètres jouant sur les performances de ce mode sont privilégiées dans le standard, ce qui mena à des travaux étudiant diverses configurations et leurs effets sur le *Packet Delivery Ratio* (PDR) [132]. La majorité de ces paramètres concernent le contrôle d'accès (durée de rétention, probabilité de réallocation de canal), le principe du mode 4 étant que les appareils gèrent le support par eux-mêmes. Une analyse des performances de ce processus MAC a été menée, à l'aide d'une chaîne de Markov à temps discret et avec la norme 802.11p comme point de comparaison [133]. Il en est déduit que le 802.11p conserve un avantage sur le C-V2X en termes de délai de communication moyen, à condition que la densité des véhicules reste faible. Dans des conditions de réseau plus difficiles, le mode 4 du LTE-V2X prend l'avantage en matière d'évitement des collisions grâce à son mécanisme de contrôle d'accès.

4.3.2 L'arrivée de la 5G

Le développement de la 5G fut grandement motivé par les applications véhiculaires. En effet, le recours aux ondes millimétriques (mmWave), de même qu'au *Multiple-Input Multiple-Output* (MIMO), sont des nouveautés bienvenues dans la poursuite de l'URLLC. De plus, la forte augmentation de bandes passantes induit une mise en commun optimisée des ressources avec les *Bandwidth Portions* (BWP) : afin de ne solliciter que les capacités de calcul et de puissance nécessaires à un usage donné et à un *User Equipment* (UE) donné,

une partie contigüe de la bande d'une porteuse n'utilise qu'une même numérogie OFDM. De cette manière, en subdivisant une même porteuse, l'utilisation des ressources est flexible et fiabilisée, avec un multiplexage intrinsèque de par les configurations différentes à chaque BWP.

En outre, avec l'arrivée de la 5G et le déploiement d'antennes hautement directionnelles, la technologie de formation de faisceaux (ou *beamforming*) a été adaptée aux applications mobiles [134]. Elle consiste à exploiter la directivité des ondes millimétriques pour déployer plusieurs antennes au même endroit, sur des phases différentes, afin de desservir différentes zones. Ainsi, chaque station de base héberge une série d'antennes à ondes millimétriques sans interférence, tout en conservant leurs largeurs de bande colossales, et en étant capable de suivre les utilisateurs mobiles afin d'assurer la continuité du service [135]. Le *beamforming* pourrait alors être optimisé pour réduire les délais [136]. Généralisant le processus aux communications véhiculaires [137] présente la mise en œuvre des technologies de *beamforming*, de broadcast et de multicast dans le cadre du projet 5GCAR, améliorant l'efficacité et la fiabilité de transmission pour les messages communs distribués aux véhicules couverts par le même faisceau.

Dans l'ensemble, la 5G apporte des améliorations substantielles en termes de débits de données, au prix d'une portée plus courte et d'une plus grande sensibilité aux perturbations physiques, ce qui explique le retard de la mise à niveau à partir du LTE dans les réseaux véhiculaires. Les applications ciblées, à forte mobilité et souvent urbaines, exigent une fiabilité et une polyvalence supérieures à celles trouvées jusqu'à présent dans les réseaux LTE. Par conséquent, le déploiement à grande échelle de la 5G a suscité des études sur la gestion de la mobilité dans les réseaux V2X [138], et les deux interfaces 5G-V2X ont été étudiées. [107] les explore selon une approche simultanée, en étudiant l'impact des communications se produisant sur l'interface Uu lorsque l'interface PC5 est déjà utilisée. Un algorithme de changement de mode D2D est utilisé pour maximiser le débit moyen en fonction de la position relative de l'UE par rapport au bord de la cellule. Pour mieux s'adapter aux architectures des réseaux urbains actuels, nous visons à mettre en pratique ces connaissances préliminaires sur le transfert dans un contexte multi-RAT hétérogène.

4.3.3 La 5G-V2X

La mise en œuvre à grande échelle du LTE-V2X a permis l'étude et le déploiement de nombreuses applications véhiculaires dépendantes d'une architecture de communication. Cependant, avec l'évolution et l'affinement des cas d'usage, un autre défi apparut avec l'arrivée de capteurs embarqués à haute précision tels que les caméras à ultra-haute définition et le LIDAR évoqués dans la partie précédente. La bande passante et le débit sont comparables entre LTE-V2X et DSRC, avec pour point commun une faiblesse de performance en ce qui concerne la transmission de grandes quantités de données, ce qui ne permet pas de véritable cohérence avec la démocratisation d'équipements engendrant

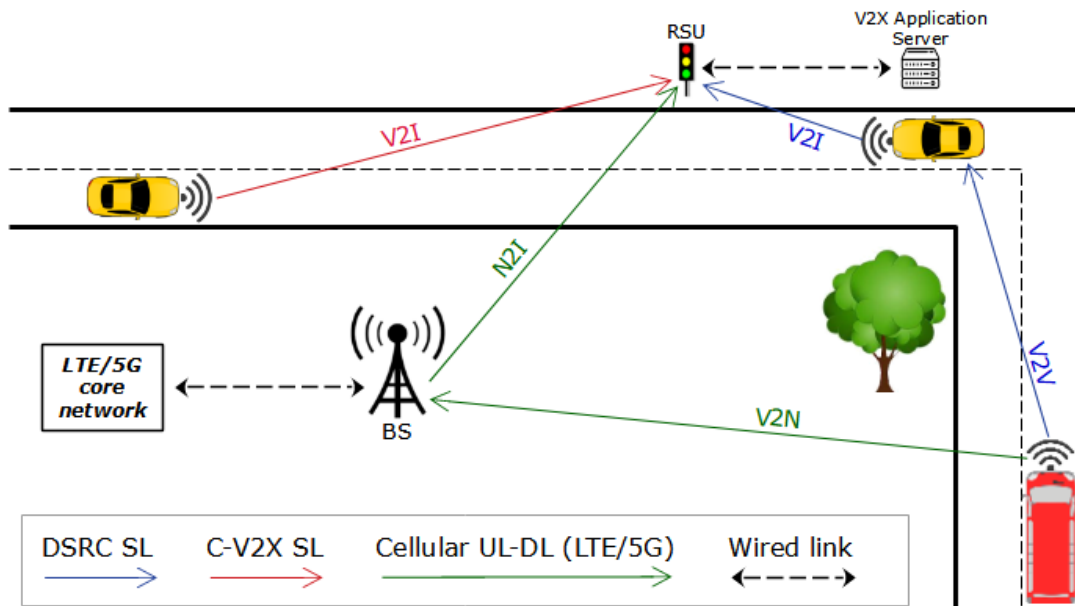


FIGURE 4.4 Aperçu des communications V2X sur les RAT évoqués.

de forts volumes de données. Le développement de la 5G NR pour Uu dans la Release 15 répond à ces problèmes et ouvre la voie à des déploiements V2X massifs [139], qui aboutiront à l'extension du C-V2X du LTE à la 5G pour PC5 lors du lancement de la Release 16, tirant parti d'une largeur de bande et d'un débit accrus. Les interfaces 5G-V2X PC5 et Uu qui en résultent sont mieux adaptées au transfert de données de capteurs à haute précision vers les serveurs périphériques sans détérioration des délais. L'interopérabilité avec le LTE est encouragée par ailleurs, notamment pour la transmission de messages de positionnement (CAM et DENM). La figure 4.4 est une représentation visuelle des communications V2X intervenant sur les interfaces mentionnées.

Les transmissions SL peuvent maintenant avoir lieu sur la gamme de fréquences FR1, y compris la zone des 2,5 GHz en plus de la zone des 5,9 GHz, avec des largeurs de bande de canal dynamiques de 10, 20, 30 ou 40 MHz, et avec une méthode d'accès Mode 4 améliorée (désormais nommée Mode 2) capable de passer du SB-SPS au mécanisme d'ordonnancement dynamique [140]. Par rapport à la norme 802.11p, avec cette nouveauté, les communications V2I et V2V ne chargent pas le même canal et bénéficient d'une plus grande largeur de bande. Par conséquent, les transmissions de données intervéhiculaires sont plus efficaces et plus robustes, car elles peuvent avoir lieu indépendamment de la réception des données de l'infrastructure, permettant la prise en charge des communications D2D à des latences ultra-faibles [62]. Cette avancée représente une grande amélioration pour les applications MEC et VEC critiques, comme la diffusion d'alertes et la conduite à distance, car différents blocs de données peuvent être transmis simultanément aux véhicules et aux serveurs périphériques pour être traités.

En termes de fiabilité, le 5G-V2X met en place un ensemble de solutions en cas de mobilité à grande vitesse. Le passage à un *Orthogonal Frequency Division Multiplexing* (OFDM) flexible comme solution de modulation pour les transmissions sur le SL permet de sélectionner dynamiquement l'espacement inter-sous-porteuse jusqu'à 60 kHz, afin d'amortir l'impact de l'effet Doppler. De plus, l'intégration de *Demodulation Reference Signals* (DMRS) adaptatifs optimise l'équilibre entre débit et fiabilité, en augmentant la quantité de signaux par sous-porteuse lorsque la vitesse du véhicule augmente, réduisant le *Bit Error Rate* (BER) malgré la forte mobilité. Enfin, la prise en charge de *Negative Acknowledgements* (NACK) sur une fréquence dédiée offre la possibilité de répondre rapidement aux erreurs détectées lors du décodage tout en conservant un overhead constant.

4.3.4 La prometteuse Release 17

Avec la Release 17, la prise en charge du mmWave et d'*enhanced Mobile Broadband* (eMBB) devint le point central de l'étude des applications de la 5G aux réseaux véhiculaires. Ainsi, la bande de fréquences FR2 a été introduite sur l'interface Uu, de 24,25 GHz à 52,6 GHz, permettant la transmission de données sur des largeurs de bande de 100 à 400 MHz [141]. Par conséquent, en tirant parti de cette multiplicité de fréquences et d'autres avantages de la 5G-NR tels que l'espacement dynamique des sous-porteuses, le C-V2X est en mesure de répondre aux exigences de l'URLLC [142], [143]. Cependant, constatant la vitesse de déploiement des systèmes 5G *Non-Standalone* (NSA), et notant que ces nouvelles bandes de fréquence dépendent intégralement du déploiement massif de systèmes 5G *Standalone* (SA), la mise en œuvre réelle et la commercialisation à grande échelle d'équipements automobiles prenant en charge la Release 17 n'est pas à prévoir pour un futur proche. À ce jour, les équipements V2X de pointe mis en vente dans l'industrie intègrent les protocoles de communication allant jusqu'à la Release 15 [39], [40].

En résumé, les technologies de communication V2X telles que DSRC et le C-V2X Mode 4 réduisent considérablement le temps de latence en permettant une transmission directe sans nécessiter de BS. Le tableau 4.1 présente une comparaison des caractéristiques des technologies de communication abordées dans ce chapitre, et la figure 4.5 en résume la chronologie. Les communications dépendantes du D2D sont généralement requises par les applications MEC, qui ont besoin d'atteindre les serveurs périphériques de manière rapide et fiable pour décharger des données critiques et souvent volumineuses.

Cependant, avec des portées de transmission directe annoncées ne dépassant pas le kilomètre [144], il est attendu que les véhicules en mouvement basculent fréquemment d'un RAT à l'autre, compte tenu des conditions de communication souvent difficiles sur la route, pour décharger les tâches complexes et les données à stocker. Nous nous intéressons donc aux différentes solutions proposées dans la littérature, allant des architectures de réseau aux mécanismes de sélection de RAT, afin d'optimiser les communications véhiculaires dans des conditions fortement variables.

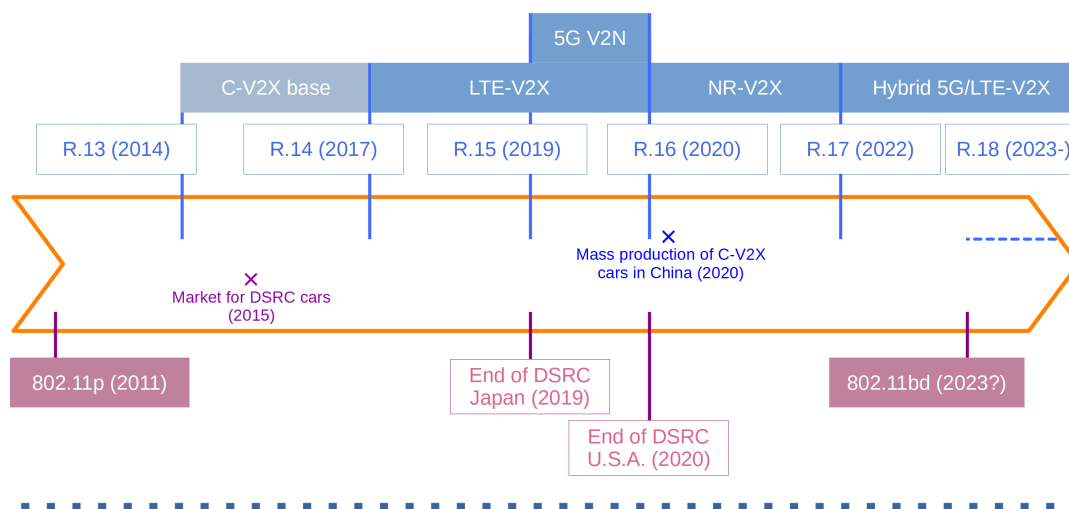


FIGURE 4.5 Chronologie des protocoles de communication V2X.

4.4 L'arrivée du Cloud-RAN

La comparaison entre les technologies de communication cellulaire et le Wi-Fi est ancienne, mais un nouveau concept a été établi, visant à intégrer les capacités d'informatique Cloud dans le réseau radio : *Cloud-RAN (C-RAN)* [146]. Cette idée a vu le jour lorsqu'il devint évident que le choix d'un RAT ne suffirait pas à garantir un certain niveau de qualité de service pour les contextes fortement mobiles. En effet, la QoS dépend de la capacité à programmer les transmissions, à catégoriser les paquets et à adapter le service en fonction de l'état du réseau, donc notamment de sa charge. Ces opérations sont rendues bien plus complexes lorsque les entités impliquées dans le calcul sont dissociées de celles impliquées dans la communication. Le C-RAN réunit ces deux plans.

En combinant la puissance de calcul disponible au niveau de chaque antenne, nous obtenons un pool de ressources pouvant être utilisé par n'importe laquelle des stations de base chaque fois que des transmissions sont reçues au niveau du *Remote Radio Head (RRH)*. Plus précisément, en permettant au signal en bande de base d'être traité au sein d'un pool de *Base Band Unit (BBU)* au lieu d'une unité unique allouée à chaque station de base, le traitement devient moins coûteux, et le lien fronthaul entre le pool BBU et les RRH permet des communications rapides et de meilleure fiabilité [147]. En gardant ces caractéristiques à l'esprit, [148] dévoile un exemple de déploiement MEC V2X exploitant le C-RAN et un réseau central LTE. En tirant profit de la robustesse du pool BBU et des *Virtualized Network Function (VNF)* dans des conditions de forte dynamique, l'architecture répond aux scénarios de forte mobilité avec du matériel reconfigurable et une gestion améliorée des liens d'accès. Cette architecture peut être mise à jour avec un réseau central 5G pour satisfaire des services V2X encore plus exigeants tout en continuant à s'appuyer sur une

structure MEC. La contribution du C-RAN dans ce type d'architecture réside dans la capacité du pool BBU à fournir des quantités de ressources flexibles et dynamiques, en fonction des conditions de mobilité et des services applicatifs demandés.

Nous avons étudié les principales technologies radio impliquées dans les architectures de communications véhiculaires. La sélection de l'un ou l'autre des ces RAT dépend d'une multitude de facteurs, corrélés avec les spécificités de chacun. En outre, le comportement des véhicules eux-mêmes a un impact sur les performances du réseau. Suite à notre étude, certains facteurs de variation viennent à l'esprit, comme la forte mobilité des appareils, les courtes portées de communication des protocoles V2X, ou encore les zones de couvertures restreintes de la plupart des points d'accès. Nous explorons désormais les solutions déjà abordées en réponse à ces difficultés de communication, de sorte à positionner notre travail.

4.5 Les difficultés des communications V2X

4.5.1 Zones de couverture

Les applications mobiles ont une utilité réduite si elles ne peuvent pas fonctionner dans des environnements dynamiques. Cette motivation est d'autant plus prégnante pour les équipements se déplaçant rapidement et sur de longues distances. Par conséquent, nous nous attendons à ce que les réseaux véhiculaires soient conçus de manière à assurer la couverture d'une zone étendue. La présente sous-section explique les différentes approches adoptées à cette fin.

Le multi-RAT

Compte tenu de la diversité des RAT disponibles dont les capacités conviendraient partiellement aux applications à forte mobilité, la recherche sur le sujet des combinaisons de technologies radio s'est développée. Avec le développement incessant de nouvelles générations de réseaux cellulaires, chacune possédant des caractéristiques uniques, la possibilité de passer d'un canal à l'autre de manière fluide assure la rétrocompatibilité ainsi qu'une cohérence améliorée dans la gestion de QoS. Ainsi naquirent les architectures multi-RAT et hybrides [149], [150], ensuite étendues aux réseaux véhiculaires [151]. En effet, comme nous l'avons remarqué dans la partie précédente, les véhicules peuvent utiliser plusieurs liaisons de communication pour exploiter simultanément un lien sur un RAT pour la transmission et d'autres liens ou RAT pour la réception de paquets provenant d'équipements voisins. Cette structure robuste est plus évolutive que les structures single-RAT, car le taux d'occupation des canaux peut être harmonisé sur un ensemble étendu de canaux radio, en fonction des besoins de chaque véhicule. Les technologies V2V et V2I pourraient même être utilisées simultanément, étant donné que les infrastructures et les véhicules sont

capables de gérer les mêmes technologies radio. En ce qui concerne exclusivement le V2V, en 2018, [152] a mis en œuvre une architecture à overhead réduit, en mettant l'accent sur la prise en compte du contexte au profit de la stabilité des performances, quelle que soit la densité de véhicules. L'année suivante, [153] a visé une application centrée sur la fiabilité, en exploitant différents RAT et une redondance au niveau des paquets pour garantir le moins d'échecs de transmissions possibles. Plus récemment, [154] établit une application de mise en correspondance pour les réseaux hétérogènes, qui facilite la sélection d'un AP en fonction des besoins de chaque utilisateur, même si le point d'accès le plus approprié n'est pas le plus proche de l'équipement. En exploitant une variété de RAT, chaque élément de contenu peut bénéficier d'un service adapté à ses spécificités. Cependant, l'inconvénient de la multiplication des canaux radio est que la sélection du RAT entraîne un délai de traitement supplémentaire et que les appareils peuvent être surchargés d'informations : l'utilisation simultanée de plusieurs canaux permet la réception simultanée de tâches et de paquets, contrairement aux architectures classiques où un seul canal ne peut être utilisé à la fois pour recevoir des données.

La virtualisation

Le principal problème conceptuel du multi-RAT est la quantité de gestion supplémentaire de la couche MAC nécessaire pour chaque équipement, des serveurs MEC jusqu'aux véhicules eux-mêmes, afin de leur permettre de basculer d'un canal radio à l'autre en fonction du contexte. La virtualisation des fonctions réseau, ou *Network Function Virtualization (NFV)*, est un autre moyen de mieux contrôler le comportement des appareils en matière de communication. Ce paradigme se définit comme la virtualisation de toute fonction réseau intégrée dans le plan de données ou de contrôle [155] [156]. Il s'agit essentiellement d'un moyen de consolider différents types d'équipements de réseau potentiellement très variés dans une architecture, ce qui rend son déploiement approprié pour les structures V2X. Ainsi, [157] établit une architecture de réseau véhiculaire fondée sur le NFV, en utilisant le partage d'instances de fonctions de réseau virtuel, ou VNF, pour fournir des services appropriés pour de multiples tâches similaires. En outre, le problème connu du routage multicast peut maintenant être abordé grâce à la connaissance globale des dispositifs impliqués apportée par le concept de virtualisation, étant donné que toutes les instances VNF actives et la position des cloudlets qui les hébergent sont connues. Ces deux nouveautés contribuent à réduire le coût d'admission des demandes de délestage vers le MEC et le temps d'établissement du chemin le plus court. De la même manière, le recours à du logiciel et de l'algorithmique sophistiqués dans les mécanismes d'allocation de ressources facilite la gestion des situations de trafic réseau fluctuant, en assurant un équilibrage des canaux alloués entre chaque RAT [158]. Néanmoins, la virtualisation pèse sur les ressources informatiques, qui sont déjà limitées dans les serveurs MEC. Par conséquent, les performances et la qualité de

service peuvent se détériorer dans les environnements à forte densité de véhicules, où les instances VNF peuvent être saturées et les ressources des cloudlets peuvent être allouées en intégralité.

Toutefois, pour certains déploiements de réseaux véhiculaires MEC, éviter la saturation des serveurs locaux peut être la principale exigence en matière de performances. Dans ce cas, l'objectif est de fournir une couverture large et fiable à un nombre variable de véhicules, quelle que soit leur densité, en encourageant le délestage vers des serveurs MEC distants lorsque le nombre de véhicules par unité de distance augmente. [88] présente une stratégie de délestage multi-saut, améliorée par la théorie de la corrélation des liens pour accroître la fiabilité des transmissions V2V. Le principal avantage de cette technologie est l'augmentation des possibilités de délestage, ce qui élargit la zone de prise en charge des véhicules par les serveurs périphériques des RSU et BS. L'accent mis sur la fiabilité est crucial dans un système multi-saut, car toute retransmission implique une perte substantielle des ressources du réseau pour chaque lien impliqué. Néanmoins, une stratégie de délestage assistée par véhicule nécessite que les véhicules stockent et mettent à jour leurs connaissances sur leurs voisins. Ainsi, une capacité informatique embarquée accrue est nécessaire pour stocker et mettre à jour les données localement, de même qu'une consommation supplémentaire de ressources réseau émerge du balisage des informations de contexte.

Ces coûts ont ensuite été réduits dans le cadre d'un autre déploiement edge computing multi-sauts, conceptualisé par [159] et fondé sur des services de réservation de relais. Un système de popularité facilite la sélection des relais véhiculaires en interdisant systématiquement aux dispositifs non fiables de participer au processus de relais de délestage. En ne conservant que les véhicules dignes de confiance comme candidats à la stratégie multi-saut, le risque de perte de paquets est considérablement réduit ainsi que la consommation de ressources. Dans le même temps, une étape de la sélection du saut suivant est déjà effectuée avant la transmission, ce qui réduit la durée globale du processus, et donc le délai de transmission des données délestées.

4.5.2 Forte mobilité et bascule horizontale

Maintenant que les questions d'optimisation des zones de couverture pour les architectures véhiculaires ont été abordées, nous allons nous focaliser sur la mobilité. Compte tenu de la vitesse élevée des UE dans les réseaux véhiculaires et de la portée souvent restreinte des points d'accès, nous étudions les solutions proposées afin de réduire les délais liés aux processus de gestion de mobilité.

Différenciation entre bascule et migration

Tout d'abord, il est nécessaire de différencier les processus de bascule (ou *handover*) et de migration. Bien qu'ils soient tous deux concernés par le même phénomène, c'est-à-dire la mobilité de l'UE émettrice, ils ne sont pas synonymes. En effet, le *handover* est un processus lié au réseau de communication, qui se déclenche lorsqu'un utilisateur mobile quitte sa cellule ou perd la couverture du réseau de son point d'accès courant (*Horizontal Handover (HHO)*), ou bien afin d'améliorer ses performances en cas de couverture simultanée dans une structure hétérogène (*Vertical Handover (VHO)*) [160]. Afin d'assurer la continuité de la connectivité sans fil lorsqu'un appareil change de zone de couverture cellulaire, un ensemble de messages de signalisation circule sur le canal d'accès aléatoire *Random Access Channel (RACH)*, alors que la station de base courante et la station de base suivante partagent des données de contexte pour enregistrer la nouvelle position de l'UE dans une nouvelle cellule.

Par ailleurs, la migration est un processus concernant l'utilisation des ressources informatiques par les utilisateurs mobiles [161]. Dans le cas du MEC, les ressources de calcul (machines virtuelles et conteneurs) sont gérées au niveau des serveurs périphériques. Lorsqu'une tâche est délestée par un véhicule qui quitte la couverture du point d'accès associé au serveur MEC sollicité, il peut y avoir migration de ressources afin d'assurer la continuité du service de délestage. Ainsi, le *handover* peut être perçu principalement comme une thématique liée au réseau, et la migration comme une question orientée vers le système. Certains travaux ont visé à examiner les deux questions simultanément [162], mais leur nature différente tend à motiver des projets distincts.

Le *handover* multi-RAT

En raison des limites des réseaux cellulaires existants dans les architectures véhiculaires, nous nous attendons à ce que les équipements basculent d'une technologie radio à l'autre et d'un point d'accès à l'autre. Le principal problème posé par les bascules fréquentes est le temps passé sans connectivité après la rupture d'un lien, par exemple lors de la sortie d'une cellule. Afin d'y remédier, [163] étudie un mécanisme de transfert proactif, qui s'affranchit également de la signalisation sur le RACH pour réduire les temps de connexion. La limite de cette procédure est sa centralisation auprès des BS cellulaires, qui est peu appropriée aux exigences des applications véhiculaires.

Dans [151], la sélection du RAT dans un réseau hybride 802.11p et LTE est étudiée, en prenant le postulat d'un VHO déficitaire lors du retour au cellulaire. Tout en assurant la continuité de la connectivité, la recherche de la limitation d'exploitation des canaux LTE a conduit à l'élaboration d'un système distribué de gestion des ressources radio fondé sur des données de position et de QoS comme le débit, la latence et le PDR. Ce mécanisme doit être actualisé, ce pourquoi nous souhaitons introduire la 5G et le C-V2X afin d'étudier des réseaux hybrides plus diversifiés et plus modernes.

Dans [164], la performance du handover LTE est étudiée dans le contexte d'un réseau hétérogène avec des UE fortement mobiles réalisant la transition entre macrocellules et picocellules. Un modèle est établi pour surveiller les emplacements des échecs de bascule, de manière à déduire une probabilité d'échec en fonction de la mobilité des UEs. La prise en compte du *fading* et du *shadowing* est ensuite ajoutée afin d'étudier la robustesse du mécanisme. La notion de cellules hétérogènes est également étudiée dans [165], par le biais de cellules virtuelles constituées de plusieurs APs desservant les mêmes canaux pour un même groupement de véhicules (ou *cluster*), par *beamforming* 5G [134]. Cette solution limite le nombre de bascules dans les situations de forte densité et de grande mobilité, à condition que le milieu héberge un très grand nombre de points d'accès 5G, ce qui ne semble approprié que dans le cadre d'expériences par simulation.

L'apport du SDN

Le *Software-Defined Networking* (SDN) caractérise l'ensemble des architectures réseau dont les ressources habituellement physiques sont virtualisées et gérées de manière centralisée au niveau d'un contrôleur dédié. Une telle structure permet une séparation claire des plans de données et de contrôle, ainsi qu'une administration facilitée car uniformisée. Un lien clair apparaît avec le NFV, en raison de la gestion informatisée des fonctions réseau, même si le NFV peut être déployé de manière morcelée sans nécessairement être accompagné des mécanismes liés au SDN, là où le SDN est présenté comme un tout indissociable. Au vu de la complexité de certaines architectures réseau pour la gestion de mobilité, nous étudions l'apport du SDN dans les solutions V2X.

Nous pouvons imaginer une structure intégralement gérée par programmation, de la communication à l'ordonnancement, afin d'optimiser l'utilisation de la bande passante pour les usagers de la route. Ainsi, l'ordonnancement par codage, ou *Coding-Based Scheduling* (CBS), en combinaison avec le SDN, bénéficie d'une connaissance globale des dispositifs actifs, ainsi que d'un contrôle centralisé. Dans les applications V2X reposant sur le réseau SDN, tous les dispositifs sans fil sont considérés comme des commutateurs, se comportant selon la programmation du plan de contrôle abstrait. De cette manière, chaque exigence en matière d'ordonnancement, de performance et même de handover peut être gérée par un contrôleur central pour cibler les véhicules et les données sur la base des informations de contexte constamment diffusées [166]. Ces informations peuvent aller de la connaissance du trafic à la popularité du contenu demandé, toutefois la quantité de données propagées vers les RSU pourrait surcharger le réseau dans des environnements plus denses. Ainsi, [167] ajoute une considération pour le broadcast dans son mécanisme d'ordonnancement, de sorte à exploiter efficacement les quantités limitées de bande passante au niveau de la BS.

Le CBS utilise le codage réseau dans des environnements hétérogènes pour faciliter et optimiser la mise en cache des données dans les véhicules. Face aux difficultés conceptuelles liées au routage de messages broadcastés, un déploiement plus robuste et efficient a été

imaginé dans [89]. Dans cette architecture dénommée CVECN, les véhicules suivent une stratégie de collaboration définie par le plan de contrôle et se regroupent au sein de clusters. Ces petits groupes de véhicules facilitent le délestage et le partage des ressources entre appareils proches. Cependant, la mobilité doit être gérée à deux échelles à la fois : le véhicule et le cluster. Combinée aux migrations de VM, nous déduisons que la gestion de la mobilité de CVECN consomme de grandes quantités de ressources de réseau et de calcul. Cependant, sa contribution aux processus de migration de calcul des tâches en réponse à la forte mobilité véhiculaire ne peut pas être négligée, tirant profit du SDN afin d'améliorer la proactivité des migrations telle qu'étudiée dans [168]. Plus précisément, lorsqu'un véhicule quitte progressivement la zone de couverture de sa RSU courante, l'augmentation du délai de communication est détectée, ce qui déclenche un mécanisme de prédiction de l'itinéraire. Ensuite, en fonction de l'état de la file d'attente des tâches du véhicule, le calcul des tâches peut se poursuivre avec la RSU courante ou être transféré à la RSU suivante, moyennant transfert sur le lien backhaul.

Enfin, nous avons abordé des solutions spécifiques au problème du multi-RAT, sans nous concentrer sur des architectures plus classiques, à un seul RAT et non-SDN. Pour les zones déjà couvertes par une technologie radio donnée, avec des équipements de réseau et de cœur prêts à être déployés, ni le SDN ni le NFV ne sont envisageables, de sorte que la connaissance globale des dispositifs actifs doit être acquise différemment. Ainsi, les réseaux véhiculaires de partage de renseignements ont été étudiés [78], où les serveurs MEC hébergent un algorithme d'apprentissage profond de haut niveau qui extrait des connaissances spatio-temporelles des données contextuelles contenues dans les paquets. Cette intelligence peut ensuite être partagée entre les serveurs pour maintenir une connaissance mutuelle des dispositifs et des contextes dans une zone étendue. Par conséquent, la mise en œuvre d'algorithmes d'apprentissage profond au-dessus des mécanismes de transmission et de calcul des serveurs périphériques induit une charge supplémentaire sur le processeur. Toutefois, cet investissement confère au système une certaine robustesse face à la dynamique du contexte routier, ainsi qu'un traitement rapide des données et une réduction de la latence, en particulier lors de l'accès aux services Cloud.

Les algorithmes d'allocation de ressources réseau

En complément des technologies de communication utilisées, les algorithmes d'allocation de ressources sont à l'étude en réponse au contexte de forte mobilité véhiculaire. En effet, dans les réseaux V2X, les indicateurs de performance sont nombreux (délai, priorité, PDR, interférences), et leur importance parfois dépendante des applications déployées. Par conséquent, la prise en compte de multiples facteurs nécessite un mécanisme d'allocation de ressources dynamique et flexible. La littérature est foisonnante, avec des exemples de mise en œuvre proposés pour des contextes allant de l'amélioration du rapport consommation énergétique / performance [169] à la maximisation de l'utilisation des ressources physiques

disponibles [170]. D'autres modèles choisissent une approche prédictive de sorte à maximiser la QoE y compris en cas de forte mobilité des utilisateurs, par la réallocation dynamique de ressources à l'issue d'un processus de décision markovien [171].

Dans le cadre de transmissions D2D entre un grand nombre d'équipements, le partage des ressources de communication peut être amélioré à partir de mécanismes personnalisés, étudiés dès la naissance de protocoles de communication véhiculaire [172]. Les travaux présentés dans [173] étudient un ensemble de modèles d'allocation de puissance et de partage de ressources radio pour des communications V2V et V2I simultanées. Ces études aboutissent à divers algorithmes de *clustering* des liens D2D, différenciés par le principal critère pris en compte dans le modèle. Ces critères concernent la taille maximale des *clusters*, l'équité entre équipements, ou la régularité des intervalles d'accès au lien, à la manière d'un mécanisme *round-robin* D2D tel qu'étudié dans [174].

En prenant en compte une multitude de critères concomitants, différentes tâches de calcul peuvent être délestées simultanément au travers de liaisons multiples de manière équitable, au prix d'une charge de calcul instantanée plus élevée sur les serveurs périphériques. [175] élabore un algorithme de garantie d'équité à long terme qui optimise conjointement l'allocation des ressources de communication et de calcul, ainsi que la subdivision des tâches pour leur transmission simultanée. Ce mécanisme tient compte des multiples facettes du délestage MEC, avec une réaffectation rapide de la puissance d'émission sur le lien UL, de la sous-porteuse du RAT choisi et des ressources du processeur du serveur MEC, en fonction de l'état du système. La congestion du côté du serveur et de l'équipement peut alors être évitée, grâce à l'intégration d'informations contextuelles dans le paradigme de transmission/distribution des sous-tâches.

4.6 Les expérimentations réelles

Malgré un intérêt grandissant ces dernières années, les déploiements matériels V2X MEC sont relativement rares. Les raisons de cette rareté sont multiples, allant de la montée en flèche des coûts du matériel, ou à la nécessité de disposer d'une vaste zone urbaine pour effectuer des essais expérimentaux dans des conditions proches du réel. Néanmoins, le domaine du edge computing est en plein essor, comme en témoignent les efforts déployés pour tenter d'harmoniser les standards communautaires et les contextes de déploiement, compte tenu de l'ampleur des cas d'usage envisageables [176]. Ainsi, en raison de la proximité clairement établie entre le MEC et les architectures véhiculaires, ces dernières nécessitent de vraies études concrètes. Nous avons constaté que des modèles mathématiques ont été élaborés et testés, des problèmes ont été formulés, des algorithmes ont été écrits, des simulations ont été effectuées ; la prochaine étape est donc de soumettre ces avancées à des conditions expérimentales réelles. Nous proposons donc d'étudier les principaux projets historiques et fondateurs des architectures V2X.

À l'heure de la rédaction de ce travail, les déploiements d'architectures de communication véhiculaire sont relativement peu nombreux. Malgré cette situation, une quantité grandissante de solutions V2X sont à l'étude, l'étape finale étant une évaluation en conditions réelles [177]. Les premières expérimentations ont eu lieu à l'échelle européenne, avec deux objectifs : mettre en application les technologies décrites dans les standards, et étudier les comportements des véhicules connectés dans des situations d'itinérance. Le premier donna naissance aux projets 5GCAR [178] et 5G-CARMEN [179], [180], le second fut à l'origine de 5GCroCo [181], [182] et 5G-MOBIX [183]. Ces essais demeurent cependant concentrés sur l'utilisation des réseaux 5G dans le cadre d'applications véhiculaires, tel qu'abordée dans la sous-section 4.3.2.

Les essais sur le terrain V2X multi-RAT disponibles ont été menés dans des conditions routières favorables, comme les autoroutes [184] ou les routes peu fréquentées [185]. Avant ces travaux, les essais sur le terrain ont consisté à mettre en œuvre et à évaluer les performances des normes de communication entre véhicules [130], [178]. Un ensemble de travaux comparatifs a également été mené pour combler l'écart entre les performances des simulations et celles des essais sur le terrain. [144] se concentre sur les performances du standard 802.11p et met en évidence le fait que le DSRC tend à être moins sensible à la vitesse qu'estimé à la suite d'évaluations réalisées dans un environnement de simulation. Bien que cruciales pour nos hypothèses, les découvertes faites dans l'ensemble de ces travaux ont tendance à négliger l'impact potentiel des aménagements urbains sur les performances du réseau, ce qui nous incite à approfondir l'évaluation des performances des protocoles de communication V2X dans des conditions difficiles et soumises à de multiples obstacles.

Le consensus émergent de ces essais sur le terrain est que les perturbations physiques n'ont pas fait l'objet d'études expérimentales approfondies pour les réseaux véhiculaires. Par exemple, l'impact du *shadowing* temporaire sur la route a été analysé dans [186] pour les transmissions C-V2X PC5, mais aucune évaluation hors simulation n'a pas encore été menée à bien. En outre, en prévision de l'arrivée de la 5G, [187] étudie les performances du C-V2X dans diverses conditions de canaux et d'interférences pour une variété d'applications véhiculaires potentielles sur une autoroute. Quoi qu'il en soit, étant donné que la Release 17 a fait son entrée en 2022, de multiples mises en œuvre de la 5G-V2X sur le terrain doivent être effectuées.

4.7 Notre positionnement

Avec des portées de transmission directe annoncées allant jusqu'à un kilomètre [144], les véhicules en mouvement basculent fréquemment d'un RATs à l'autre, compte tenu des conditions variables du réseau sur la route, afin de continuer à déléster leurs tâches complexes et leurs données à stocker. Nous présentons donc un mécanisme de sélection de RAT fondé sur une connaissance géographique des métriques de performances afin

d'optimiser le déclenchement de **VHO**, afin de réduire les temps de latence. Notre travail se démarque de la littérature existante sur le multi-RAT véhiculaire pour les raisons suivantes :

1. **Messages de position.** En complément des messages de demande de délestage, nous exploitons les messages de position émis par les véhicules afin de mesurer la **QoS** sur l'ensemble des interfaces de communication **D2D** à disposition, afin d'affiner la connaissance des performances réseau sur le territoire parcouru.
2. **Proactivité.** Contrairement aux travaux mis en avant dans notre étude, la solution de handover proactif que nous proposons est distribuée au niveau des véhicules, s'affranchissant de l'orchestration d'un contrôleur SDN central ou d'une station de base.
3. **Repli cellulaire.** Nous avons remarqué qu'une grande quantité des systèmes conçus exploitent peu les interfaces à longue portée, privilégiant une comparaison de performances entre interfaces à courte portée. Nous proposons un fonctionnement conjoint, intégrant l'interface cellulaire mobile comme une solution de repli par laquelle délester les tâches dans les situations où la qualité des liens **SL** est dégradée au-delà des limites.
4. **Déploiement expérimental.** À chaque étape de la conception de notre mécanisme, nous prenons en considération la finalité d'un déploiement en conditions réelles, apportant une grande variabilité dans les comportements observés, due à de nombreux phénomènes physiques complexes et inattendus qui le démarquent d'une mise en oeuvre par simulation.
5. **Matériel utilisé.** Afin d'actualiser les travaux les plus anciens, nous menons notre campagne expérimentale avec du matériel commercial de pointe, nous permettant d'être aussi technologiquement à jour que possible.

TABLEAU 4.1 Comparaison des caractéristiques 802.11p et C-V2X

Feature	DSRC	LTE-V2X (PC5 M4)	LTE-V2X (Uu)	5G-V2X (PC5 M2)	5G-V2X (Uu)
<i>Base technology</i>	802.11a	LTE	LTE	5G NR	5G NR
<i>Multiplexing</i>	OFDM	SC-FDMA	SC-FDMA	Scalable OFDM	OFDM + SC-FDMA
<i>Frequency band</i>	5.9 GHz	5.9 GHz	Multiple [145]	FR1 & FR2	FR1 & FR2
<i>Channel bandwidth</i>	10 MHz	10 - 20 MHz	10 - 15 - 20 MHz	10 - 20 - 30 - 40 MHz	Up to 100 or 400 MHz
<i>Subcarrier spacing</i>	156.25 kHz	15 kHz	15 kHz	15 - 30 - 60 - 120 kHz	15 - 30 - 60 - 120 kHz
<i>Access method</i>	CSMA-CA	SB-SPS	Random	Dynamic or SB-SPS	Random
<i>Reliability</i>	None	HARQ	HARQ	HARQ-NACK	HARQ-NACK

CERES-V2X : STABILISER LA CONNECTIVITÉ V2X PAR HANDOVER PROACTIF MULTI-RAT

La validation de la 3GPP Release 17 ainsi que le déploiement massif des réseaux mobiles 5G ont donné lieu à un regain d'intérêt pour les technologies cellulaires, y compris dans le cadre d'applications véhiculaires. Certains constructeurs automobiles s'orientent d'ores et déjà vers des solutions 5G pour la connectivité de leurs futurs véhicules, de par leur simplicité d'implantation en comparaison de solutions D2D ne disposant pas d'une gestion de la mobilité suffisamment performante. Toutefois, les expériences menées par le passé ont mis en évidence des performances de communication dégradées par la nécessité d'un passage dans le cœur de réseau mobile.

Dans ce chapitre, nous présentons CERES-V2X, notre mécanisme d'optimisation de sélection de RAT fondé sur le partage de données de performance en vue de la stabilisation de connectivité V2X, que nous avons implanté en conditions réelles. Pour y parvenir, chaque élément d'infrastructure représente sa zone de couverture réseau sous la forme d'une grille; pour chaque case, des valeurs de QoS sont collectées lorsque des paquets sont reçus. Ces grilles sont ensuite mises en forme et distribuées aux véhicules entrant dans la zone, leur permettant d'adapter proactivement leur RAT en fonction de leur itinéraire, évitant des ruptures de lien inattendues. Notre campagne d'expérimentation à bord d'un véhicule autonome connecté déambulant sur un circuit urbain met en évidence une dépendance amoindrie des communications V2N, une réduction des latences, et de bonnes performances de l'interface DSRC en situation de ligne de vue directe malgré le désintérêt croissant pour cette technologie.

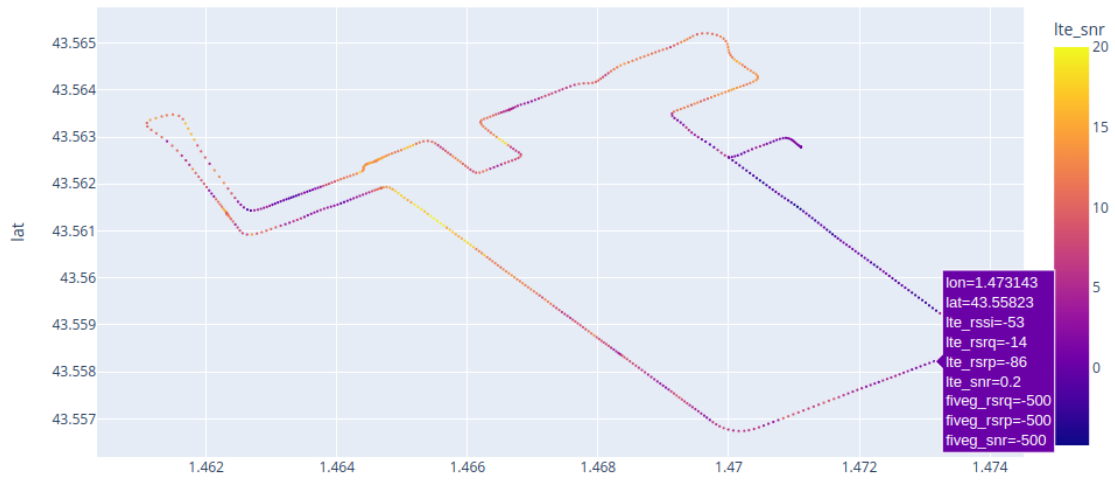


FIGURE 5.1 Nissan Micra K11E (2000) équipée d'une OBU.

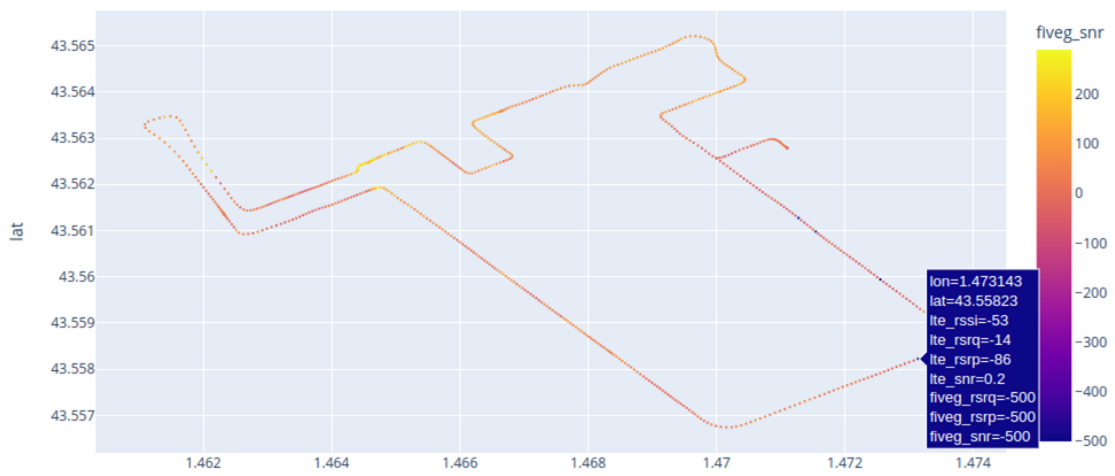
5.1 Travail préliminaire : couverture cellulaire

En amont de notre travail d'optimisation de bascule, nous estimons fondamental d'étudier le terrain de jeu afin de déterminer de potentielles zones sujettes à un signal cellulaire fortement bruité. Le campus de l'Université Toulouse III Paul Sabatier est couvert par un eNodeB et un gNodeB, lui fournissant un accès LTE et 5G NSA. Nous nous proposons donc de parcourir le circuit prévu pour une navette autonome connectée, avec un véhicule équipé d'une OBU non-commerciale prenant en charge ces deux RAT. La navette n'étant pas encore habilitée à déambuler sur le campus au moment de cette étude préalable, nous équipons une Nissan Micra K11E (2000) du matériel nécessaire, la rendant de fait connectée (voir Figure 5.1. le véhicule est piloté manuellement le long du circuit, pendant que l'unité à bord exécute un programme relevant les métriques de couche physique pour les deux technologies mobiles. Ces mesures répétées nous permettent de déterminer simplement d'éventuelles zones fortement perturbées, voire trop bruitées pour être exploitables de manière fiable dans le cadre de transmissions véhiculaires. La figure 5.2 présente les résultats de ces mesures, sous la forme de cartographies du rapport signal-sur-bruit, ou *Signal-to-Noise Ratio* (SNR), en LTE et en 5G.

Les cartographies affichées mettent en évidence un ensemble de phénomènes. Tout d'abord, il apparaît de manière intuitive qu'une grande partie du circuit bénéficie de



(a) SNR – LTE.



(b) SNR – 5G.

FIGURE 5.2 Cartographie du SNR sur le circuit

performances de couche physique 5G supérieures à celle de LTE. Dans certaines situations, mises en évidence par les segments bleus sur la figure 5.2a, la couverture LTE n'est même pas disponible, ce qui permet de privilégier la 5G par défaut. Cependant, il apparaît que les performances de couche physique en 5G ne sont pas systématiquement supérieures à celles du LTE. Certains points du circuit, mis en évidence par la couleur indigo sur la figure 5.2b, présentent un signal infiniment bruité en 5G (la valeur $-\infty$ est affichée comme -500 pour des raisons de lisibilité), mais un SNR positif en LTE. Cette observation justifie l'utilisation d'un mécanisme de bascule entre technologies cellulaires, de sorte à utiliser le LTE comme solution de repli. Par ailleurs, il existe également des zones du circuit où aucune des deux couvertures cellulaires ne propose de SNR positif, ce qui dans notre cas implique l'exploitation d'autres technologies d'accès radio, telles que les technologies D2D, afin de stabiliser la connectivité des utilisateurs.

5.2 Vue d'ensemble de CERES-V2X

En réponse aux besoins de latences extrêmement réduites, nous avons précédemment étudié des solutions de délestage optimisé dans un contexte MEC et multi-RAT, afin de parvenir à notre propre proposition d'architecture. Cette solution était concentrée sur la gestion des ressources de calcul mises à disposition par les serveurs périphériques, de sorte à réduire les temps d'attente des tâches une fois reçues. Ce mécanisme orienté vers le système ne tenait toutefois pas compte des protocoles d'accès au réseau, ni de l'impact des éventuelles variations du milieu, notamment dans le cadre de transmissions sur le SL, particulièrement sensibles à la fiabilité. La présente contribution aspire à étudier ces thématiques.

Au vu de la multiplicité des technologies radio prises en charge au niveau des UE, ainsi que de la diversité des couvertures réseau proposées dans un contexte urbain, l'ajout de cas d'usage rattachés aux véhicules connectés bénéficierait grandement d'une solution robuste capable de détecter les perturbations sur les canaux, de sorte à sélectionner dynamiquement l'interface la moins impactée afin de transmettre les requêtes de délestage. Une telle solution accélérerait l'accès au support, fiabiliserait les transmissions, et limiterait la dépendance aux technologies cellulaires mobiles. Nous proposons un mécanisme s'attaquant à ce besoin : CERES-V2X.

CERES-V2X est un mécanisme de bascule "make-before-break" [163] qui délègue aux OBU le processus de sélection de l'interface la plus appropriée entre DSRC, C-V2X PC5, 5G-V2X Uu et LTE-V2X Uu, en fonction de la performance mesurée en une position donnée. Notre solution est bilatérale, tirant parti de la vision à moyen et long-terme des RSU statiques ainsi que de la mobilité des OBU. Les contributions de CERES-V2X sont les suivantes :

1. La zone de couverture de la RSU est divisée en une grille, enregistrant progressivement

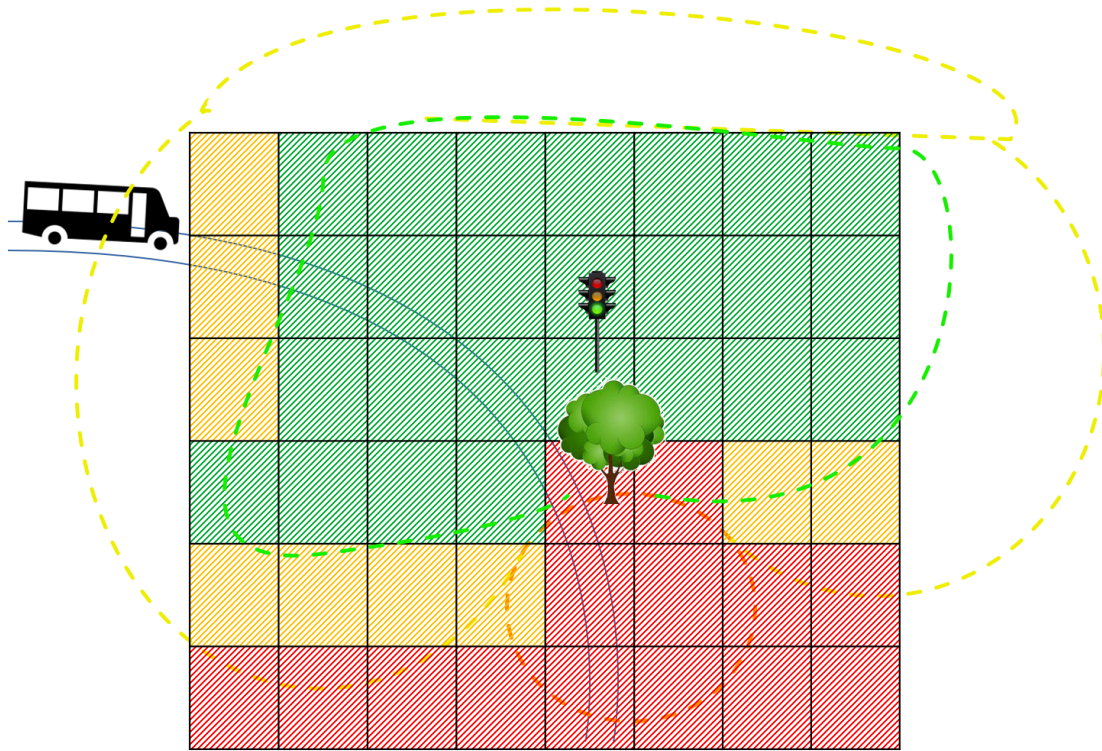


FIGURE 5.3 Principe de la cartographie de la RSU.

les données de performance sur chaque interface **D2D** disponible à partir des messages reçus afin de suivre les évolutions d'environnement.

2. La grille est partagée avec les véhicules à proximité, ce qui permet aux **OBU** de sélectionner le **RAT** le plus approprié en fonction des prochaines cases de la grille où le véhicule s'apprête à entrer. Un algorithme de prise de décision augmente la fiabilité des résultats et traite les cas limites.
3. Une différenciation des modes de communication entre le cellulaire et le **D2D** permet de gérer les situations où la couverture d'une **RSU** est compromise, ainsi que les situations très favorables comme le **LOS**.
4. L'évaluation des performances du **C-V2X** et du **DSRC** est réalisée avec un essai sur le terrain par le biais d'une navette autonome pour suivre l'évolution de la latence des communications sur un grand nombre de tours de circuit.

La figure 5.3 met en évidence le principe de cartographie intégré à la **RSU** par le biais d'une grille. À mesure que les **UE** circulent dans la zone de couverture et transmettent leur messages, la **RSU** relève les métriques de **QoS** associées au lieu d'émission des messages, de sorte à regrouper ces informations dans des cases de la grille. Par moyenne glissante, un indice de performance est associé à chaque cellule, allant d'excellent (ici représenté en vert) à très mauvais (ici représenté en rouge). L'opération est combinée pour chaque **RAT**

étudié, aboutissant à une unique grille mais à de multiples valeurs stockées dans chaque cellule. Du point de vue du véhicule recevant ensuite la grille, l'ensemble du processus de bascule lui est propre, sans nécessiter d'intervention tierce : tout véhicule reçoit la même grille, qu'il lui reste à interpréter en fonction de son propre itinéraire. Ainsi, sur la figure 5.3, si l'on suppose que les cellules colorées concernent les performances du DSRC, le véhicule représenté est donc conscient des apports du DSRC dans la première moitié de son passage dans la zone de couverture de la RSU, puis du besoin de changer de RAT (donc de basculer) une fois son entrée dans la case partagée par l'arbre, car l'indicateur de performance associé est "très mauvais" et le demeure sur le reste de la portion d'itinéraire.

5.3 Construction de la grille

Dans cette section, nous expliquons le mécanisme de bascule améliorée par la subdivision en grille. Dans les réseaux de véhicules urbains, il n'est pas rare de présenter tout une partie de la zone d'expérience, généralement couverte par un AP comme une RSU, sous la forme d'une grille dont chaque cellule contient ses propres valeurs significatives. Les caractéristiques de la grille concernent généralement des données applicatives telles que la position à une intersection [188] ou des données de perception de véhicules multi-capteurs [189]. Notre proposition s'en démarque grandement, car nous tirons parti de la subdivision en grille pour surveiller les valeurs de QoS, c'est-à-dire les indicateurs de performance du côté du réseau, et déclencher les bascules en conséquence, ce qui permet d'éviter les ruptures de liaison inattendues. Ce principe présente l'avantage d'être agnostique vis-à-vis des applications, et donc de bénéficier à tout système déployé sur le réseau.

Dans les réseaux véhiculaires, les véhicules diffusent leurs CAM périodiquement pour partager des informations contextuelles telles que leur position et leur vitesse, à une fréquence f_{cam} allant de 1 à 10 Hz [122]. Nous activons la propagation des CAM sur les interfaces DSRC et C-V2X PC5. À la réception, les RSU enregistrent les performances des deux interfaces et alimentent une base de données géographiques. Après avoir été alimentée par des données de performance provenant de différentes positions de véhicules et à différents moments, chaque RSU obtient une connaissance contextuelle de sa propre zone de couverture, incluant les positions où la latence augmente et sur quelle interface. Ces informations sont cruciales pour le processus de sélection de RAT du côté du véhicule.

Pour propager le contenu de la base de données de la RSU aux OBU, les données doivent être formatées d'une manière qui n'implique pas une charge excessive sur le réseau, ni une puissance de traitement excessive à exploiter une fois qu'elles atteignent les véhicules. À cette fin, nous modélisons la zone de couverture du RSU comme une grille, dont le contenu est propagé par un générateur de grille. La figure 5.4 illustre la structure d'une grille, d'une cellule et d'un générateur. Chaque RSU crée et gère une grille G carrée de K lignes et K colonnes, chaque cellule couvrant une latitude L et une longitude l . La RSU

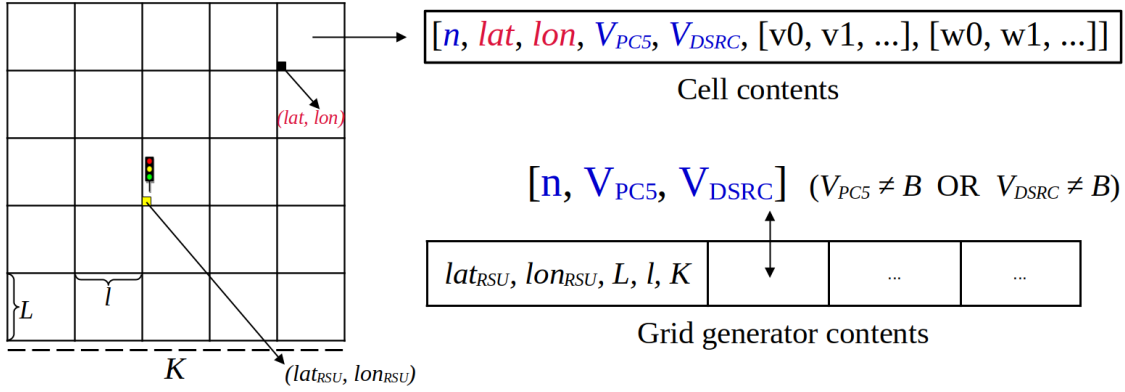


FIGURE 5.4 Structure du contenu de grille, case et générateur.

se situe au centre de sa propre grille. Chaque cellule de G , dénommée $G(n)$ est identifiée par son ID unique n , ainsi que par les coordonnées GPS de son coin inférieur gauche, et contient les données moyennes de **QoS** relevée sur les interfaces **C-V2X** et **DSRC**. Ces données de **QoS** sont dénommées $G(n)_R$, avec R le nom de l'interface concernée.

Si aucun message n'est reçu sur une interface en provenance d'une position associée à une cellule donnée, les données de qualité de service pour cette cellule et cette interface sont fixés à une valeur de borne supérieure B . Lorsque plusieurs **CAM** sont reçus en provenance de la même cellule et de la même interface au fil du temps, les données **QoS** associées sont mises à jour selon une moyenne glissante sur les derniers M paquets reçus, ce qui garantit des réactions promptes si des perturbations physiques apparaissent dans le réseau ou le quittent. Entre-temps, le tableau des générateurs de grille est mis à jour par l'ajout de triplets contenant l'ID de la cellule et les valeurs de **QoS** de la cellule pour les deux interfaces. Si un triplet existe déjà avec l'ID de la cellule recherchée, il est écrasé. La valeur de tête d'un générateur est toujours la liste suivante de données structurales : coordonnées GPS du centre de la grille, dimensions de la zone couverte par une cellule, nombre de cellules dans la grille. Afin de parvenir à cette représentation, nous avons étudié d'autres approches d'identification de cases, notamment Quadkeys [190]. Ces approches bénéficient d'une grande lisibilité et sont adaptables à de multiples scénarios applicatifs, ce qui entraîna leur utilisation dans de multiples projets liés à l'IoT [191], puis à la mobilité véhiculaire [192]. Toutefois, la granularité de la géolocalisation par la méthode Quadkeys, c'est-à-dire la taille du côté d'une case, ne peut évoluer que par facteurs 2 successifs. Dans notre scénario expérimental urbain, nous souhaitons bénéficier d'une plus grande finesse dans le paramétrage de la grille, ce qui nous a orienté vers une solution latitude-longitude. Le processus de mise à jour de la grille **RSU** est résumé dans l'algorithme 3.

Algorithm 3 Gestion de grille – RSU positionnée en (lat, lon)

Require: $K, L, l, M, lat, lon, B, f_{grid}$

- 1: Let T_G be the grid generator table
 - 2: Generate grid G of dimension $K \times K$, centered on (lat, lon) , each cell covering $L \times l$ areas
 - 3: Upon timer expiration
 - 4: **if** CAM packet received on RAT R **then**
 - 5: Retrieve CAM coordinates d
 - 6: Find id n of G 's cell covering coordinates d
 - 7: Append received QoS data in $G(n)_R$ window
 - 8: **if** $G(n)_R$ window size $> M$ **then**
 - 9: Pop first value in $G(n)_R$ window
 - 10: **end if**
 - 11: Let V_R be the average of values in $G(n)_R$ window
 - 12: Update $G(n)$ cell contents with V_R
 - 13: **if** T_G does not contain id n **then**
 - 14: Append values $[n, V_{PC5}, V_{DSRC}]$ extracted from $G(n)$, as a triplet, to T_G
 - 15: **else**
 - 16: Update T_G 's triplet of id n with new $[n, V_{PC5}, V_{DSRC}]$ triplet
 - 17: **end if**
 - 18: **end if**
 - 19: **if** $1/f_{grid}$ elapsed **then**
 - 20: Transmit generator $[lat, lon, L, l, K, T_G]$
 - 21: **end if**
 - 22: Reset timer
-

5.4 Mécanisme de bascule

Les générateurs de grille sont envoyés par chaque **RSU** à intervalles réguliers définis par la fréquence f_{grid} , et sur l'interface *sidelink* **C-V2X** PC5. Nous choisissons ce **RAT** spécifiquement pour sa plus longue portée et sa meilleure robustesse face aux obstacles, et parce que nous souhaitons éviter que le réseau cellulaire ne soit sollicité par les **RSU**, créant de la charge excédentaire. En outre, si un véhicule sans grille ne reçoit pas de générateur de grille, il peut en déduire que ni couverture **DSRC** ni **C-V2X** ne sont disponibles à l'heure et à la position actuelles, conservant par conséquent son utilisation du réseau cellulaire. À la réception d'un générateur de grille, l'**OBU** vérifie les valeurs de tête pour s'assurer de l'existence préalable de cette grille dans sa base de données. Si aucune grille de la base de données de l'**OBU** ne correspond aux coordonnées du générateur, une nouvelle grille doit être construite, remplie dans un premier temps avec des valeurs de borne supérieure. Ensuite, les valeurs des cases sont mises à jour en fonction du contenu du générateur. Avec les grilles mises à jour, l'**OBU** peut maintenant faire correspondre sa position avec les cases des grilles et trouver la valeur minimale de **QoS**, nommée V_{Rsel} , pour déduire le **RAT** R_{sel} approprié pour l'envoi de requêtes de délestage. Ces derniers sont dénommés REQ à des fins de synthèse. Ce mécanisme permet également de gérer les situations où plusieurs grilles couvrent la position actuelle de l'**OBU** : la valeur minimale s'applique toujours et les paquets suivants sont envoyés à la **RSU** correspondante. Le processus de **VHO** au niveau des **OBU** est formulé dans l'algorithme 4.

5.5 Évaluation des performances

Après les travaux de simulation évoqués en première partie, nous conduisons pour cette contribution une évaluation par déploiement en conditions réelles. Les performances de notre mécanisme de sélection de **RAT** sont évaluées par rapport à deux systèmes de référence. Le premier schéma de référence réalise un **VHO** opportuniste lors de la détection d'une couverture **D2D** et bascule à nouveau sur l'interface cellulaire lorsque la connectivité **D2D** est perdue. Le second schéma de référence ne transmet que sur l'interface cellulaire, ce qui constitue une limite supérieure dans notre scénario expérimental. Nous distinguons les paquets contextuels comme les paquets **CAM** et REQ : les premiers sont exploités pour mettre à jour les grilles et détecter la couverture, tandis que les seconds sont envoyés périodiquement à une fréquence f_{req} sur l'interface la mieux perçue. Dans le processus d'évaluation, les performances sont mesurées sur la transmission de paquets REQ, car nous maîtrisons chaque aspect de ces paquets, leur contenu et leur instant d'émission, ce qui en facilite le suivi.

Algorithm 4 Mécanisme de bascule – OBU**Require:** f_{req}, B, R

```

1: Upon timer expiration
2: if Generator [ $lat, lon, L, l, K, T_G$ ] received then
3:   if ( $lat, lon$ ) do not match any existing grid then
4:     Generate grid of dimension  $K \times K$ , centered on ( $lat, lon$ ), each cell covering
        $L \times l$  areas
5:   end if
6:   Update cells of ( $lat, lon$ ) grid with  $T_G$  contents
7: end if
8: Update GPS position ( $x, y$ )
9: if One or more grids contain cells covering ( $x, y$ ) then
10:  Fetch minimal  $V_{R_{sel}}$  value
11:  if  $V_{R_{sel}} < B$  then
12:    Deduct best RAT  $R_{sel}$  and RSU  $d_{(lat, lon)}$ 
13:  end if
14: else
15:   $R_{sel} = Cellular$ 
16: end if
17: if  $R_{sel} \neq R$  then
18:  VHO from  $R$  to  $R_{sel}$ 
19:   $R \leftarrow R_{sel}$ 
20: end if
21: if  $1/f_{req}$  elapsed then
22:  Send REQ packet over RAT  $R$  to RSU  $d_{(lat, lon)}$ 
23: end if
24: Reset timer

```

TABLEAU 5.1 Paramètres de la campagne d'expérimentation

Item	Value
Vitesse max du véhicule	20 km/h
Périmètre du circuit	2.8 km
Hauteur de la RSU par rapport au sol	2.5 m
RAT utilisables	DSRC, LTE, 5G NSA, C-V2X
Valeur de L	35 m
Valeur de l	20 m
Valeur de K	500
Valeur de M	5 derniers paquets reçus
Valeur de B	50 ms
Valeur de f_{grid}	2 Hz
Valeur de f_{cam}	1 Hz
Valeur de f_{req}	5 Hz

5.5.1 Conditions expérimentales

Les dispositifs utilisés pour créer le trafic véhiculaire et infrastructurel sont des **OBU** et **RSU** Cohda Wireless MK6, qui sont conformes à la version 15 du 3GPP et équipés de 5G **C-V2X**. Les limitations matérielles du constructeur nous interdisent de mesurer les indicateurs de qualité de service de la couche physique (SINR, RSSI, RSRP, RSRQ) avec les **RAT D2D**. De plus, certaines de ces métriques peuvent également varier en fonction du matériel utilisé, ce qui nous motive à surveiller les performances au niveau des paquets. Nous souhaitons comparer les performances des **RAT** dans les communications **V2I** et **V2N** qui, dans notre contexte expérimental, couvrent la robustesse physique et la stratégie d'accès. Les destinations ne changent pas entre les deux technologies **D2D** et le transport est supposé être identique entre les **RAT** : *User Datagram Protocol* (**UDP**) comme proposé dans les protocoles de communication **SL** afin de simplifier la quantité de transmissions, donc de réduire les délais. En outre, notre analyse de l'état de l'art a conclu que les principaux points de différence entre les **RAT V2X** étudiés (**DSRC** et **C-V2X Mode 4**) se situent dans les couches **PHY** et **MAC**. Par conséquent, il est cohérent de s'appuyer sur la latence de couche **MAC** à couche **MAC**, ainsi que sur le taux de perte de paquets, comme indicateurs de qualité de service. Les grilles sont donc remplies avec des valeurs de latence.

L'essai sur le terrain est mené sur le campus de l'Université Toulouse III Paul Sabatier, où une navette autonome EasyMile EZ10 [193] suit une voie préprogrammée et dessert les passagers à des arrêts de bus prédéfinis. La navette dispose d'un récepteur GPS-RTK, de caméras et de **LiDAR**, lui permettant de se positionner dans son environnement tout en restant attentive aux autres usagers de la route. Un élément extérieur pénétrant dans le périmètre de sécurité de la navette provoque un ralentissement, ou son arrêt pur et simple en attente de la validation de l'opérateur à bord. En dehors de ces opérations de validation manuelle, l'ensemble de la direction et de la vitesse sont gérés de manière autonome. Tous les tours sont donc strictement identiques d'un point de vue géographique, ce qui accroît la similitude entre les parcours. Nous avons intégré à la navette deux **OBU** en fonctionnement simultané afin de lui fournir une connectivité tout en évitant les variations de performance entre les parcours dues aux fluctuations du trafic routier : une unité héberge notre solution, l'autre héberge les schémas de référence. Les **RSU** MK6 se trouvent à proximité de l'aire de jeu. Les transmissions **OBU-RSU** peuvent se faire en mode **D2D** ou via **V2N**, la 5G étant privilégiée et le LTE étant une solution de repli. La figure 5.5 présente une vue aérienne du système. Deux **RSU** sont positionnées en bordure de route sur de longues lignes droites, ce qui garantit une couverture importante tout en maintenant des situations de *Non Line of Sight* (**NLOS**) en raison de leur proximité avec des obstacles fixes tels que des bâtiments et des arbres. Le matériel est présenté sur la figure 5.6. Les principaux paramètres de l'essai sur le terrain sont résumés dans le tableau 5.1.

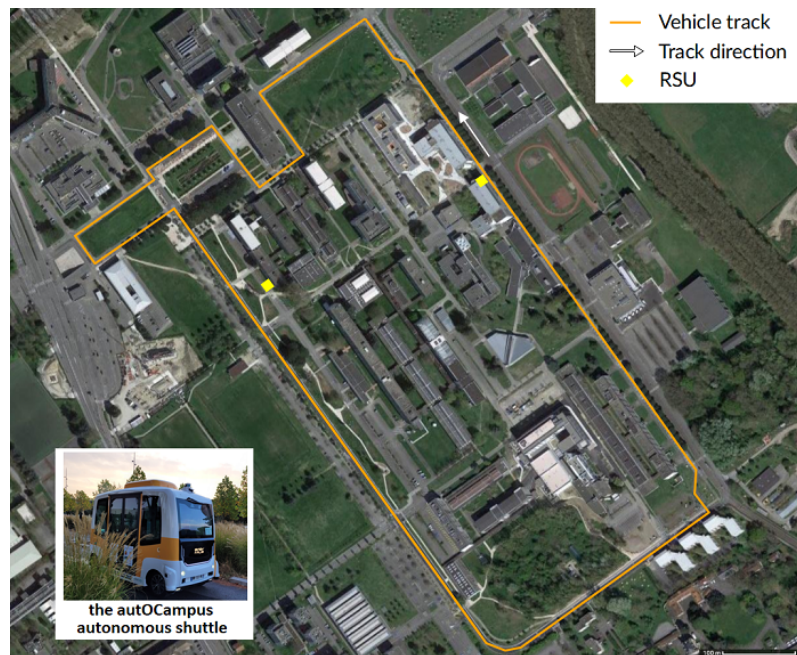
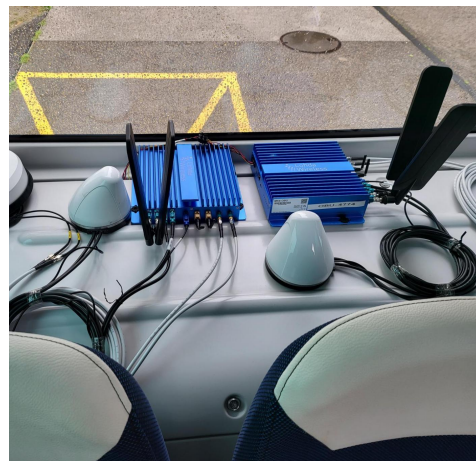


FIGURE 5.5 Vue du dessus



(a) Navette autonome



(b) Deux OBU dans la navette



(c) RSU

FIGURE 5.6 Matériel d'expérimentation



FIGURE 5.7 Comparaison des zones de couverture

5.5.2 Comparaison DSRC / C-V2X

Avant de procéder à l'évaluation des performances tout au long du trajet de la navette, nous souhaitons réaliser une observation initiale du comportement des technologies **C-V2X** et **DSRC**. À cette fin, nous générons du trafic **V2X** sur la piste et laissons les **RSU** fournir à leurs grilles les données de performance qu'elles recueillent à partir des messages **CAM**. La figure 5.7 présente la visualisation de ces grilles, projetée sur une vue satellite du terrain d'expérimentation sous forme de carte thermique. Les figures 5.7(a) et 5.7(b) affichent respectivement les valeurs de **QoS** des grilles **C-V2X** et **DSRC**. Les emplacements de chaque **RSU** sont indiqués en vert. Plus la couleur d'une cellule est chaude, plus la latence est élevée.

Tout d'abord, la quantité de cellules colorées est beaucoup plus importante pour le **C-V2X** PC5 que pour le **DSRC**, ce qui traduit une portée de communication supérieure d'environ 60 %. La robustesse vis-à-vis des obstacles physiques est également plus grande pour le **C-V2X**, comme le montrent les cellules couvrant les positions **NLOS** (mises en évidence aux points 1, 2 et 3) où la latence est beaucoup plus élevée pour le **DSRC** que pour le **C-V2X**. La grande variété des bandes de fréquences exploitables, de même qu'un mécanisme d'accès plus robuste, expliquent cet avantage en termes de performances dans les situations difficiles. Enfin, nous observons que, dans des conditions favorables telles qu'une proximité **LOS** avec la **RSU**, le **DSRC** obtient des temps de latence plus courts que le **C-V2X**, grâce à ses structures simplifiées des couches **PHY** et **MAC**.

Nous pouvons donc prévoir différents facteurs de déclenchement de **VHO** : tout d'abord, la sortie de la zone de couverture des **RSU** déclenchera un repli vers le cellulaire mobile. Deuxièmement, l'entrée dans la couverture d'une **RSU** depuis le bord de la grille peut déclencher un **VHO** du cellulaire vers le **C-V2X**. Troisièmement, l'approche de la **RSU**

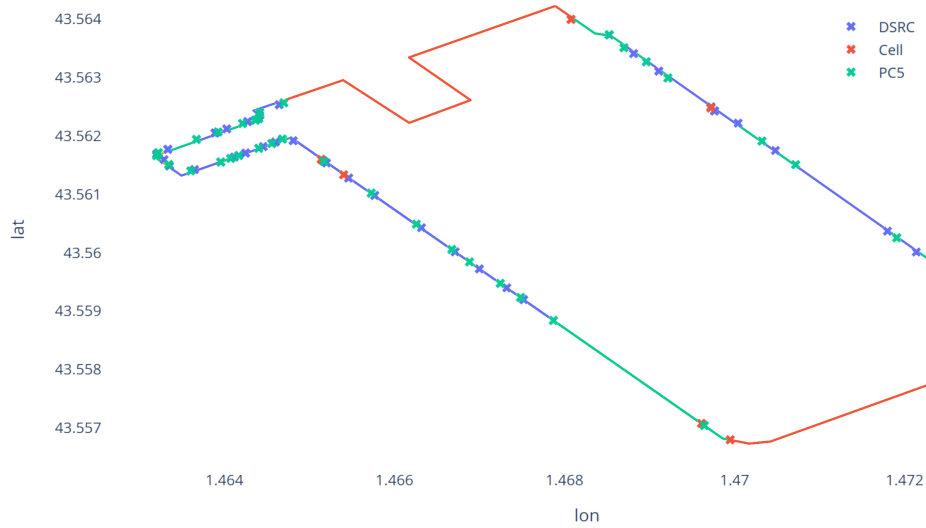
TABLEAU 5.2 Évaluation de performances, par RAT, par solution

	CERES-V2X	Opportuniste	Cellulaire
Latence DSRC (ms)	8.37 ± 0.00	<i>n/a</i>	<i>n/a</i>
Latence C-V2X (ms)	16.26 ± 0.03	15.41 ± 0.02	<i>n/a</i>
Latence Cellular (ms)	120.7 ± 0.3	130.9 ± 0.3	263.4 ± 0.8
Latence globale (ms)	51.4 ± 5.60	75.9 ± 0.2	263.4 ± 0.8
Taux de perte paquet D2D (%)	6.43 ± 0.05	4.21 ± 0.23	<i>n/a</i>

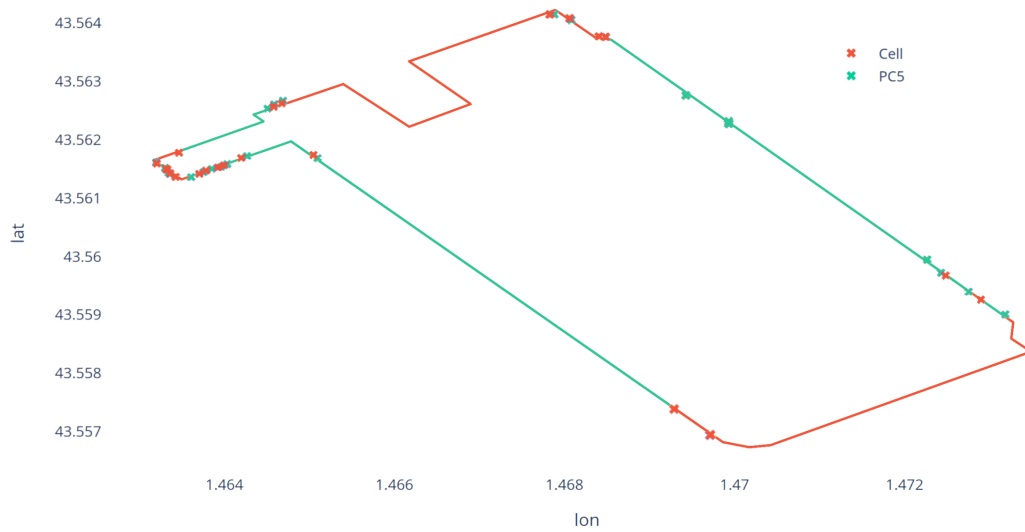
peut se traduire par l'entrée dans une cellule où les indicateurs de QoS favorisent le DSRC, ce qui déclencherait un VHO du C-V2X vers le DSRC. L'inverse est également vrai : le fait de s'éloigner de la RSU ou de passer derrière un obstacle tout en restant sous couverture D2D peut déclencher un VHO du DSRC vers le C-V2X. La sous-section suivante s'emploie à vérifier ces hypothèses.

5.5.3 Optimisation de bascule

Nous évaluons maintenant la performance de CERES-V2X (parfois dénommée "Grid" sur les figures) sur la fréquence de VHO, la latence et la perte de paquets, en utilisant le schéma de référence opportuniste comme sujet de comparaison et le schéma de référence cellulaire comme limite supérieure. Plusieurs tours sont effectués sur la même piste au moyen de la navette autonome afin d'obtenir des intervalles de confiance satisfaisants. La figure 5.8 présente une projection des emplacements où les VHO ont lieu, ainsi que la portion de piste parcourue avec chaque RAT. L'exploitation du DSRC, comme attendu d'après ce qui a été suggéré précédemment, implique de multiples VHO entre les technologies D2D, comme mis en évidence sur la figure 5.8a le long de longues lignes droites en ligne de vue directe avec la RSU. En effet, alors que le scénario opportuniste se concentre sur le passage au D2D dès lors que la couverture C-V2X est disponible, le scénario fondé sur la grille déclenche également des VHO supplémentaires pour améliorer les performances, par exemple lorsque le DSRC bénéficie du LOS. En outre, le coin supérieur gauche de la figure 5.8b présente un grand nombre de bascules entre C-V2X et cellulaire, en raison de la présence d'arbres qui perturbent la propagation (figure 5.5). L'OBU opportuniste interprète ces positions comme étant uniquement cellulaires, mais l'OBU équipée de la grille les associe au RAT le plus performant, pour lequel les performances sont encore meilleures, bien que dégradées, que celles d'un système purement cellulaire. Notre solution accroît la résistance aux longues distances et aux obstacles, alors que le *shadowing* et le *pathloss* détériorent considérablement la réception et les performances du scénario opportuniste. Nous souhaitons maintenant établir un lien entre ce comportement et les performances mesurées au niveau des paquets.



(a) VHO avec CERES-V2X



(b) VHO Opportuniste

FIGURE 5.8 Cartographie des lieux de VHO

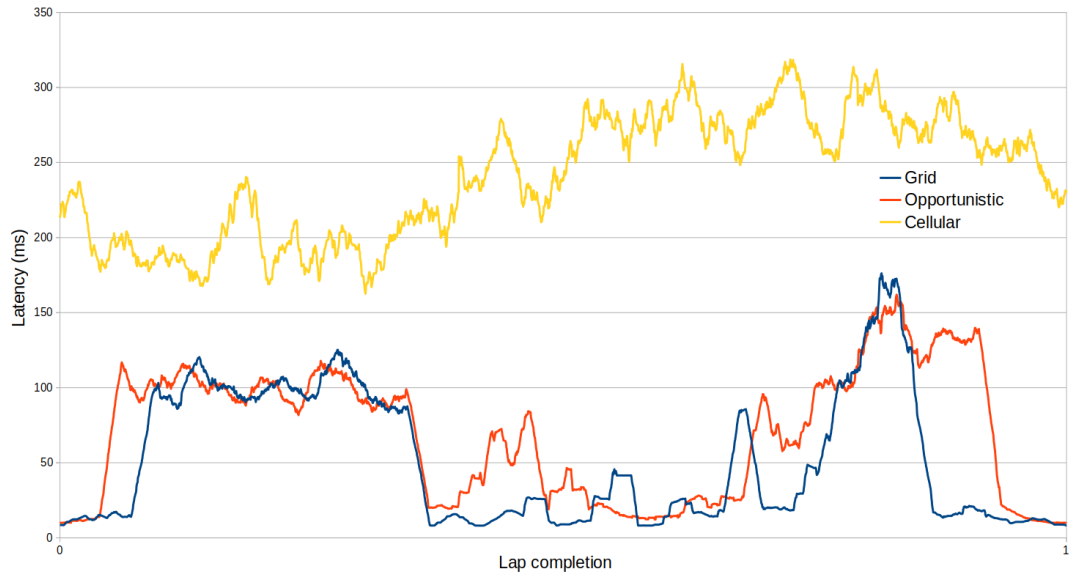


FIGURE 5.9 Évolution des latences paquet sur la durée d'un tour de circuit

Le tableau 5.2 présente un résumé de notre évaluation des performances, en comparant la latence moyenne et le taux de perte de paquets pour chaque scénario. Les temps de latence présentés sont mesurés pour les paquets REQ, qui sont envoyés par le biais du RAT perçu comme le meilleur. Nous observons un net avantage pour le DSRC, qui obtient de faibles délais avec de très faibles fluctuations malgré sa grande sensibilité aux perturbations physiques. CERES-V2X permet donc d'obtenir une latence globale plus faible : elle étend les plages d'exploitation du C-V2X, puis s'appuie sur le DSRC pour gagner quelques millisecondes supplémentaires. Cependant, la grille est sensible aux obstacles dynamiques : si un obstacle entre dans le champ et crée du *shadowing* lorsque les grilles ont été mises à jour, certains paquets sont perdus jusqu'à ce que les grilles suivantes remontent l'information et empêchent les OBU de passer à des technologies D2D perturbées. Ce problème ne concerne pas le scénario opportuniste car il n'intègre pas de comportement prédictif. Nous répondons à cette observation que les expériences ont été menées en utilisant un unique véhicule et deux OBU, et qu'il est donc probable que ce problème s'atténue avec l'ajout d'autres véhicules, ce qui accélérerait le remplissage de la fenêtre glissante et donc le processus de mise à jour de la grille. Quoi qu'il en soit, le taux de perte D2D reste acceptable, comme l'ont montré des travaux antérieurs [185].

La figure 5.9 montre l'évolution de la latence des paquets tout au long d'un tour. Nous remarquons que le système amélioré par la grille bénéficie de phases de faible latence plus longues que le système opportuniste, avec des latences basses atteignant des valeurs plus faibles et des latences élevées atteignant des valeurs similaires. Comme remarqué précédemment sur la figure 5.8, la présence de la grille prolonge les situations où la couverture

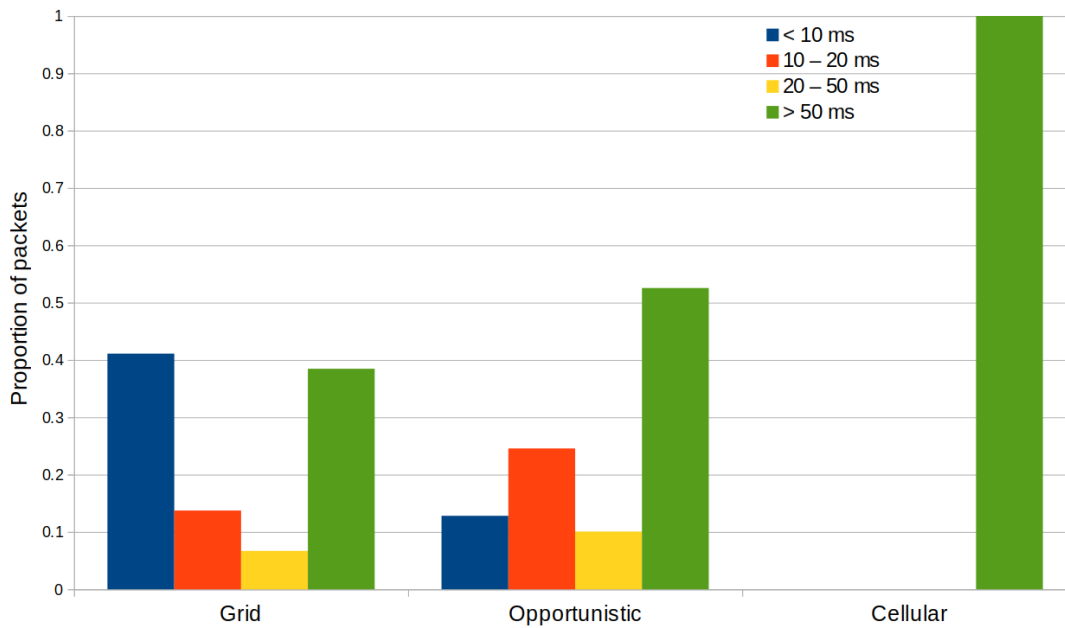


FIGURE 5.10 Distribution empirique de la latence des paquets, par solution

D2D est exploitée et permet d'obtenir des temps de latence plus faibles lorsqu'elle est disponible via DSRC. Le VHO du cellulaire vers le C-V2X peut également être anticipé, par rapport au schéma opportuniste qui doit d'abord recevoir le message CAM d'une RSU pour déduire la présence d'une couverture C-V2X. La réactivité de notre solution peut s'avérer utile lors de l'ordonnancement de transmissions prolongées pour des applications exigeantes.

La figure 5.10 représente la distribution empirique des latences des paquets, selon les quatre intervalles de latence les plus significatifs. 10 ms est la limite acceptable pour les applications V2X URLLC ; 20 ms est la latence maximale pour les cas d'utilisation V2X avancés tels que la conduite à distance, comme établi par 5GAA [9] ; 50 ms est la latence maximale attendue pour DSRC et C-V2X Mode-4 D2D, ainsi que la latence limite pour les cas d'utilisation véhiculaires moins exigeants tels que l'assistance au dépassement et le streaming vidéo embarqué [9]. Premièrement, l'hypothèse théorique selon laquelle le D2D doit être exploité dès qu'il est disponible peut être vérifiée, en observant que tous les paquets envoyés avec le système purement cellulaire dépassent la limite de latence de 50 ms en raison du saut supplémentaire et du temps passé dans le cœur de réseau. Deuxièmement, notre grille influe sur la sélection de RAT de sorte que la dépendance à l'égard des transmissions cellulaires est moindre (comme le montre la figure 5.8a avec ses segments cellulaires plus courts), ce qui réduit la proportion de paquets aux latences de plus de 50 ms. En outre, l'utilisation du DSRC produit une performance régulière avec une grande majorité de paquets atteignant l'intervalle de moins de 10 ms, comme cela a été déduit du tableau 5.2, ce qui réduit la proportion de paquets aux latences allant de 10 à 20 ms et augmente la

proportion de paquets aux latences de moins de 10 ms. Notre solution fait preuve d'une robustesse qui s'avère utile lorsqu'il s'agit d'atteindre des latences très faibles pendant des périodes prolongées, dans le cadre d'une couverture D2D, ce qui entraîne une réduction de 27 % de la proportion de paquets avec des latences supérieures à 50 ms par rapport au schéma opportuniste (52 %) et au schéma en grille (38 %). Le positionnement d'un plus grand nombre de RSU autour de la piste permettrait de réduire le nombre de paquets exclusivement cellulaires, ce qui réduirait d'autant la proportion de paquets de latence excédant les 50 ms et nous permettrait d'évaluer chaque solution plus précisément.

5.6 Limites et discussion

Dans cette section, nous discutons des limites de l'implantation actuelle de CERES-V2X et des pistes d'amélioration à étudier dans le cadre de travaux futurs.

Précision des informations. Les limitations techniques imposées par le constructeur du matériel utilisé nous ont poussé à mesurer les données de QoS au niveau paquet uniquement. Nous estimons qu'une plus grande variété de données permettrait d'affiner la perception du milieu par les RSU. Ces données pourraient inclure des indicateurs de performance de couche physique tels que ceux utilisés dans le travail préliminaire, notamment le SINR qui est un indicateur grandement utilisé dans le cadre de travaux portant sur le handover [184], [194]. Un suivi de l'état du signal augmenterait la robustesse des grilles face à l'arrivée d'un obstacle temporaire, et ce avant que la transmission de trop nombreux paquets ne survienne, accélérant donc le processus de bascule proactive vers une technologie radio moins impactée afin de réduire les pertes de paquets.

Passage à l'échelle : véhicules. L'apport du véhicule autonome fut important pour la rigueur de notre étude expérimentale. Cependant, deux limites en émergent. Premièrement, le fait de ne parcourir que le circuit prédéfini, sur les routes périphériques du quartier étudié, limite la portée de l'analyse du comportement du système. Deuxièmement, la petite quantité d'équipements ne permet pas de valider finement les apports de CERES-V2X à grande échelle. Des véhicules supplémentaires parcourant d'autres circuits traversant par exemple le cœur du quartier, entreraient dans des cases inexplorées lors de cette première étude, ce qui alimenterait d'autant les grilles des RSU, ainsi que les générateurs de grille. Par ailleurs, une plus grande quantité d'OBUs traversant les mêmes cases engendreraient des données de QoS bien plus fréquemment, ce qui augmenterait la réactivité du système en cas d'observation d'un nouveau phénomène. Nous formulons l'hypothèse que le passage à l'échelle au niveau véhiculaire réduirait le taux de pertes de paquet observé lors de la précédente évaluation de performances. Nous planifions de conduire une nouvelle campagne expérimentale intégrant ces ajouts de véhicules connectés, afin de confirmer ou d'infirmer cette hypothèse.

Passage à l'échelle : infrastructure. Nous avons eu la chance de mener une campagne

d'expérimentation en conditions réelles grâce au déploiement d'une navette autonome, ainsi qu'à l'acquisition de matériel dédié aux applications orientées *smart cities* : les OBU et RSU. Cependant, nous considérons les travaux ici présentés comme la première étape de la vie de CERES-V2X, en vue d'un affinement futur. Cet apport de précisions ne pourra avoir lieu que suite au déploiement d'une infrastructure plus complète, allant de RSU supplémentaires à un réseau 5G SA. En effet, les conditions expérimentales ici présentées font état d'un circuit à la couverture D2D fractionnée (voir Figure 5.7), ce qui implique de fait une dépendance au réseau cellulaire. Afin d'étudier les éventuelles situations où une bascule cellulaire interviendrait suite à ces causes autres qu'une simple absence d'AP à proximité, l'élargissement des zones de couverture D2D par ajout de nouveaux éléments est essentiel. De même, il ne nous est pas possible d'étudier l'implantation réelle d'une architecture de communication V2X multi-RAT, ses capacités, ses performances, et son comportement à l'échelle, sans la présence d'un réseau mobile 5G disposant du cœur de réseau associé, donc de 5G SA. Ces différents ajouts nous permettraient de mener de nouvelles études comparatives sur les zones de couverture des AP, la stabilité des communications V2X et les performances en matière de latence et de pertes.

5.7 Conclusion

Dans ce chapitre, nous avons présenté CERES-V2X, un mécanisme de stabilisation de connectivité V2X par optimisation de sélection de RAT fondé sur le partage de données de performance réseau. Cette solution aborde les thématiques de rapidité et de fiabilité essentielles aux futures applications véhiculaires, en offrant aux véhicules une capacité de bascule proactive, afin d'exploiter les communications D2D aussi souvent que possible. En subdivisant les zones de couverture des RSU en grilles, et en s'appuyant sur un partage régulier d'informations de contexte, notre solution s'intègre dans des contextes applicatifs urbains et sujets au *shadowing*. Nous avons conçu l'architecture hébergeant CERES-V2X, ainsi que ses mécanismes et ses algorithmes, puis nous avons évalué ses performances en conditions réelles par le biais d'un véhicule autonome et de matériel de communication V2X de pointe. Les résultats obtenus montrent que le DSRC reste un RAT fiable dans son maintien de faibles latences en LOS, tandis que le C-V2X prend l'avantage en présence d'obstacles et couvre de plus longues distances. Deuxièmement, comparées aux technologies cellulaires mobiles, les communications D2D s'avèrent systématiquement plus compétitives pour les applications véhiculaires. Troisièmement, CERES-V2X lui-même permet d'obtenir des temps de latence plus faibles et une dépendance moins fréquente à la couverture cellulaire que les schémas de bascule opportuniste habituels.

CONCLUSIONS ET PERSPECTIVES

Conclusions

Les véhicules autonomes représentent une part grandissante du marché automobile. Le développement de systèmes de conduite autonome ouvre la voie à de nombreux services visant à faciliter et sécuriser la circulation des biens et des personnes. Avec l'ajout de capteurs embarqués de plus en plus précis comme les caméras à très haute définition ou les [LiDAR](#), les véhicules sont capables de se positionner et d'évoluer de manière fiable dans leur environnement.

Toutefois, en raison des grandes quantités de données engendrées par lesdits capteurs, ainsi que de la courte portée de perception, les informations extraites par l'unité embarquée sont trop limitées pour parvenir à des usages complexes. De plus, l'absence d'interactions entre véhicules empêche une réelle coordination des éléments de l'[IoV](#), inscrite dans la thématique de la *smart city*. Ainsi, la conception de technologies de communication véhiculaire devient une priorité, avec leurs propres protocoles, leurs propres messages, et leurs propres exigences en matière de performance.

Dans cette thèse, nous nous sommes proposés d'étudier ces mécanismes de communication, assemblés et coordonnés sous la forme d'architectures, en vue de la satisfaction des besoins de performance des futurs cas d'usage des véhicules autonomes connectés. En tirant profit d'une bibliographie jeune mais florissante, de même que de multiples méthodes de validation par l'expérience, nous avons abordé des thématiques issues à la fois des domaines des réseaux et des systèmes distribués. Nous avons abouti à deux systèmes visant à réduire les délais impliqués dans le cadre d'applications véhiculaires, dans un contexte réaliste : une zone fortement urbanisée, à la couverture réseau hétérogène.

Nous avons dans un premier temps présenté CAVTOMECC, un mécanisme d'optimisation de sélection de destination dans le cadre de délestage de tâches critiques. CAVTOMECC s'appuie sur la multiplicité des interfaces réseau exploitables afin de tirer profit à la fois de transmissions courte et longue portée en fonction du contexte système et de la quantité de ressources disponibles. Les serveurs périphériques participent à la fiabilité de la solution, en informant régulièrement les autres participants de leur consommation de ressources, ainsi que de l'état de leurs files d'attente de tâches. Notre évaluation par simulation a montré

que CAVTOMECC fournit une plus grande stabilité aux quantités de ressources utilisées entre les serveurs, ce qui aboutit à une proportion de tâches critiques traitées dans les temps similaire ou supérieure à celle des systèmes de délestage par proximité.

La solution proposée par cette première contribution s'appuyait sur la multiplicité des interfaces réseau disponibles comme un moyen d'atteindre des destinations plus ou moins lointaines, sans s'étendre sur les autres apports potentiels des technologies utilisées. Nous avons donc poursuivi notre étude en élaborant CERES-V2X, notre solution de stabilisation de connectivité véhiculaire dans des situations de forte perturbation des liens physiques. Avec CERES-V2X, les éléments d'infrastructure partagent avec les véhicules leurs connaissances géographiques sur la qualité des communications **multi-RAT**, leur laissant prendre les décisions de basculer vers l'une ou l'autre des technologies tout au long de leur trajet, de manière anticipée afin d'éviter les ruptures de communication. Les résultats de notre étude expérimentale en conditions réelles sur un campus universitaire ont mis en évidence une forte réduction de la dépendance vis-à-vis des réseaux cellulaires mobiles, ce qui réduit grandement les latences des paquets applicatifs transmis.

Perspectives

Plusieurs perspectives sont envisageables dans la suite de nos travaux. Au-delà des limites et des solutions discutées à la fin de chacune de nos contributions (sections 3.8 et 5.6), nous abordons ici les perspectives les plus prometteuses ainsi que notre vision sur l'évolution des architectures de communication véhiculaire.

1) CAVTOMECC : évaluation à large échelle.

Dans le chapitre 3, nous avons procédé à une évaluation du principe de notre solution par simulation sur un nombre variable de véhicules et de **RSU**. Cela dit, même si nous avons pu surcharger les serveurs périphériques en augmentant la fréquence de génération de requêtes de délestage en provenance des véhicules, il serait intéressant de procéder à des évaluations à très large échelle, dans un premier temps par simulation. Une très forte augmentation du nombre de véhicules permettrait de provoquer des phénomènes non observés lors de nos expérimentations, tels que des saturations d'axes routiers, une utilisation maximale de la bande passante, ou une surcharge de l'ensemble des serveurs périphériques. Se poserait alors la question du positionnement optimal des **RSU**, de sorte à limiter la proportion de tâches en échec.

2) a. CERES-V2X : diversification des usages.

Dans le chapitre 5, nous avons concentré notre évaluation sur l'envoi de requêtes de délestage normalisées (REQ). Afin de fournir un service adapté à chaque besoin, CERES-V2X pourrait prendre en compte la nature des messages à transmettre dans son mécanisme de déclenchement de **VHO**. Ce critère supplémentaire permettrait d'ajouter un critère de

performances réseau nécessaires, en complément de la stabilité de connectivité. Ainsi, en cas de chevauchement de couvertures [multi-RAT](#), certains véhicules pourraient faire le choix de transmettre leurs messages sur une interface moins performante afin de libérer la bande passante sur l'interface optimale pour des usagers nécessitant l'envoi de messages critiques.

2) b. CERES-V2X : considérations algorithmiques.

CERES-V2X met en œuvre un algorithme de bascule verticale dont les décisions reposent sur les performances mesurées au niveau des paquets, privilégiant la technologie radio dont les mesures sont les plus satisfaisantes en un lieu donné. En complément de la littérature foisonnante sur le sujet des algorithmes d'allocation de ressources, nous pouvons imaginer plusieurs moyens de renforcer le processus de décision, en intégrant un ensemble de critères de bascule supplémentaires en fonction des usages souhaités. Par exemple dans le cadre de notre étude, la réduction de la consommation énergétique, le respect de la haute priorité de certains véhicules, ou encore la gestion de multiples applications, pourraient justifier l'ajout de variables intervenant dans le processus de sélection d'interface, ou provoquer une réallocation de ressources au sein d'une même technologie radio.

3) Perspectives à long terme : vers une ubiquité de la connectivité véhiculaire

Les solutions que nous avons présentées dans cette thèse sont pensées comme des ajouts aux architectures de réseaux urbains actuellement répandues. Ces structures, hétérogènes et articulées autour de technologies cellulaires mobiles, subissent de fortes perturbations sur leurs canaux en raison d'obstacles et de la sensibilité à la densité variable d'utilisateurs. En ajoutant des éléments d'infrastructure au niveau des zones surchargées ou perturbées, nous avons apporté une première réponse à ces problèmes par la modification des zones de couverture réseau, la mise à disposition de ressources de calcul dédiées, et la prise en charge d'un plus grand nombre de technologies de communication. Cependant, ces nouvelles architectures se heurtent à des limites structurelles liées à la staticité des points d'accès *edge*, à la sensibilité aux obstacles physiques des [RAT](#) utilisés, et aux besoins applicatifs grandissants.

Nous pensons que les architectures à venir dans les prochaines années s'attarderont à rendre la connectivité [V2X](#) encore plus disponible pour les futures applications véhiculaires, en s'attaquant aux limites précédentes. Les motivations derrière ces futurs travaux comprennent le besoin de stabilité pour des transmissions prolongées, la libération de bande passante pour les applications fortement demandeuses de ressources, et la robustesse face aux périodes de forte affluence. Pour ce faire, les thématiques à traiter concernent tant le domaine des technologies de communication véhiculaires que celui des réseaux ad-hoc et 6G (par les études prenant en compte une éventuelle mobilité des [RSU](#) [195]-[197]) ou des télécommunications (par les questionnements sur le suivi physique des utilisateurs via le beamforming [198], [199]).

Quoi qu'il en soit, le partage de données de contexte peut grandement améliorer les performances des réseaux véhiculaires, en fournissant une capacité d'adaptation de service aux éléments d'architecture. L'intégration de ces systèmes dans de futures architectures nécessite cependant d'étudier les nombreuses problématiques relatives au [multi-RAT](#), à la gestion d'applications multiples dans le [MEC](#), ou encore aux technologies radio à venir dédiées aux communications véhiculaires comme le 802.11bd et la 5G-V2X de la Release 17. Ces futures évolutions des [RAT](#) véhiculaires pourraient ouvrir la voie à une nouvelle ère de diversification des usages automobiles, pour une quantité d'utilisateurs toujours plus vaste.

PUBLICATIONS

Reuves internationales

- L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Mobile edge computing for V2X architectures and applications: A survey », *Computer Networks*, t. 206, p. 108 797, 2022. DOI : [10.1016/j.comnet.2022.108797](https://doi.org/10.1016/j.comnet.2022.108797)

Conférences internationales

- L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Context-aware task offloading with QoS-provisioning for MEC multi-RAT vehicular networks », in *2022 International Conference on Computer Communications and Networks (ICCCN)*, 2022, p. 1-9. DOI : [10.1109/ICCCN54977.2022.9868873](https://doi.org/10.1109/ICCCN54977.2022.9868873)
- L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Field Trial for Enhanced V2X Multi-RAT Handover in Autonomous Vehicle Networks », in *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, 2023, p. 1-8. DOI : [10.1109/LCN58197.2023.10223342](https://doi.org/10.1109/LCN58197.2023.10223342)

Soumis

- L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Multi-RAT-Enabled Edge Computing for Vehicle-to-Everything Architectures », *Version étendue — Révision mineure*.

BIBLIOGRAPHIE

- [1] WIKIPEDIA. « Ford T ». (mai 2023), adresse : https://en.wikipedia.org/wiki/Ford_T.
- [2] « Équipement des ménages », INSEE, rapp. tech., 2020.
- [3] C. ENGELKING. « The "Driverless" Car Era Began More Than 90 Years Ago ». (déc. 2017), adresse : <https://www.discovermagazine.com/technology/the-driverless-car-era-began-more-than-90-years-ago>.
- [4] F. KRÖGER, « Automated Driving in Its Social, Historical and Cultural Contexts », in mai 2016, p. 41-68. DOI : [10.1007/978-3-662-48847-8_3](https://doi.org/10.1007/978-3-662-48847-8_3).
- [5] CEREMA. « Les facteurs d'accidents mortels de la circulation en France en 2015 ». (août 2021), adresse : <https://www.cerema.fr/fr/actualites/facteurs-accidents-mortels-circulation-france-2015>.
- [6] N. H. T. S. ADMINISTRATION, « Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey », U.S. Department of Transportation, rapp. tech., fév. 2015.
- [7] Ó. SILVA, R. CORDERA, E. GONZÁLEZ-GONZÁLEZ et S. NOGUÉS, « Environmental impacts of autonomous vehicles: A review of the scientific literature », *Science of The Total Environment*, t. 830, 2022. DOI : <https://doi.org/10.1016/j.scitotenv.2022.154615>.
- [8] 5GAA, « C-V2X Use Cases: Methodology, Examples and Service Level Requirements », 5G Automotive Association, rapp. tech., juin 2019.
- [9] 5GAA, « C-V2X Use Cases Volume II: Examples and Service Level Requirements », 5G Automotive Association, rapp. tech., oct. 2020.
- [10] T. GROUP. « Moving towards keyless car access with Digital Car Keys ». (oct. 2021), adresse : <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/industries/automotive/use-cases/virtual-car-key>.
- [11] G. GRIM. « Ford Pass : Profitez de toute l'intelligence de votre véhicule connecté ». (juin 2022), adresse : <https://www.groupe-grim.com/ford/ford-pass-vehicule-connectee/>.
- [12] A. INNOVATION. « Audi connect Remote & Control ». (juin 2021), adresse : <https://www.audi.fr/fr/web/fr/innovation/connect/services-et-fonctionnalites/remote-et-control.html>.
- [13] CITROËN. « Commandes à distance ». (mai 2022), adresse : <https://services-store.citroen.fr/store/remote-control>.
- [14] STELLANTIS. « 18 million de véhicules Stellantis équipés du système d'alerte de véhicules d'urgence grâce à la plateforme V2X ». (mai 2023), adresse : <https://www.stellantis.com/fr/actualite/communiques-de-presse/2023/may/1-8-million-de-vehicules-stellantis-equipés-du-système-d-alerte-de-vehicules-d-urgence-grâce-a-la-plateforme-v2x>.

- [15] O. COHEN et M.-T. KOEHLER. « En voiture, un boîtier connecté pour alerter les secours en cas d'accident ». (mai 2016), adresse : <https://www.francebleu.fr/infos/insolite/innovation-sur-une-voiture-un-boitier-connecte-alerte-les-secours-en-cas-d-accident-1463914114>.
- [16] M. ZABAT, N. STABILE, S. FARASCAROLI et F. BROWAND, « The Aerodynamic Performance Of Platoons: A Final Report », Institute of Transportation Studies, UC Berkeley, rapp. tech., jan. 1995.
- [17] M. MARTÍNEZ-DÍAZ, C. AL-HADDAD, F. SORIGUERA et C. ANTONIOU, « Platooning of connected automated vehicles on freeways: a bird's eye view », *Transportation Research Procedia*, t. 58, p. 479-486, 2021, XIV Conference on Transport Engineering, CIT2021. DOI : <https://doi.org/10.1016/j.trpro.2021.11.064>.
- [18] M.-B. MEDIA. « Daimler Trucks invests half a billion Euros in highly automated trucks ». (jan. 2019), adresse : <https://media.mercedes-benz.com/article/570cd7d9-08a4-4468-8dcd-136e2438f81e>.
- [19] H. BAGHERI, M. NOOR-A-RAHIM, Z. LIU et al., « 5G NR-V2X: Toward Connected and Cooperative Autonomous Driving », *IEEE Communications Standards Magazine*, t. 5, n° 1, p. 48-54, 2021. DOI : [10.1109/MCOMSTD.001.2000069](https://doi.org/10.1109/MCOMSTD.001.2000069).
- [20] F. ALANAZI, « A Systematic Literature Review of Autonomous and Connected Vehicles in Traffic Management », *Applied Sciences*, t. 13, n° 3, 2023. DOI : [10.3390/app13031789](https://doi.org/10.3390/app13031789).
- [21] « Indra to implement the next-generation traffic management platform for connected highways using data from vehicles and advanced sensors ». (sept. 2022), adresse : <https://www.indracompany.com/en/noticia/indra-implement-next-generation-traffic-management-platform-connected-highways-using-data>.
- [22] J. SHELTON, J. WAGNER, S. SAMANT, G. GOODIN, T. LOMAX et E. SEYMOUR, « Impacts of connected vehicles in a complex, congested urban freeway setting using multi-resolution modeling methods », *International Journal of Transportation Science and Technology*, t. 8, n° 1, p. 25-34, 2019. DOI : <https://doi.org/10.1016/j.ijstst.2018.06.004>.
- [23] J. L. ROVER. « Jaguar Land Rover gets the green light to solve 150-year-old problem ». (nov. 2018), adresse : <https://media.jaguarlandrover.com/news/2018/11/jaguar-land-rover-gets-green-light-solve-150-year-old-problem>.
- [24] T. MOHN. « Can Controlling Vehicles Make Streets Safer and More Climate Friendly ». (mars 2022), adresse : <https://www.nytimes.com/2022/03/28/world/europe/geofencing-sweden.html>.
- [25] H. FIGG. « Geofencing: a new tool to make urban transport safer and more sustainable ». (oct. 2020), adresse : <https://www.eltis.org/resources/case-studies/geofencing-new-tool-make-urban-transport-safer-and-more-sustainable>.
- [26] M. CHAFKIN. « Even After \$100 Billion, Self-Driving Cars Are Going Nowhere ». (oct. 2022), adresse : <https://www.bloomberg.com/news/features/2022-10-06/even-after-100-billion-self-driving-cars-are-going-nowhere>.
- [27] S. RAZA, S. WANG, M. AHMED et M. ANWAR, « A Survey on Vehicular Edge Computing: Architecture, Applications, Technical Issues, and Future Directions », *Wireless Communications and Mobile Computing*, p. 1-19, 2019. DOI : [10.1155/2019/3159762](https://doi.org/10.1155/2019/3159762).

- [28] E. CINQUE, F. VALENTINI, A. PERSIA, S. CHIOCCHIO, F. SANTUCCI et M. PRATESI, « V2X Communication Technologies and Service Requirements for Connected and Autonomous Driving », in *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, 2020, p. 1-6. DOI : [10.23919/AEITAUTOMOTIVE50086.2020.9307388](https://doi.org/10.23919/AEITAUTOMOTIVE50086.2020.9307388).
- [29] C. LI, L. TONI, J. ZOU, H. XIONG et P. FROSSARD, « QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming », *IEEE Transactions on Multimedia*, t. 20, n° 4, p. 965-984, 2018. DOI : [10.1109/TMM.2017.2757761](https://doi.org/10.1109/TMM.2017.2757761).
- [30] O.-R. A. D. (COMMITTEE, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Society of Automotive Engineers (SAE), sept. 2016. DOI : https://doi.org/10.4271/J3016_201609.
- [31] J. LACONTE, A. KASMI, R. AUFRÈRE, M. VAIDIS et R. CHAPUIS, « A Survey of Localization Methods for Autonomous Vehicles in Highway Scenarios », *Sensors*, t. 22, n° 1, 2022. DOI : [10.3390/s22010247](https://doi.org/10.3390/s22010247).
- [32] K. SERR, T. WINDHOLZ et K. WEBER, « Comparing GPS receivers: A field study », t. 18, jan. 2006.
- [33] N. ACOSTA et J. TOLOZA, « Techniques to improve the GPS precision », *International Journal of Advanced Computer Science and Applications*, t. 3, n° 8, 2012.
- [34] Z. LIAO, B. YU, C. KANG, Y. GE et Z. GONG, « Research on GNSS and RTK Performance for V2X Application Scenarios », in *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, 2022, p. 499-504. DOI : [10.1109/ICCC56324.2022.10065945](https://doi.org/10.1109/ICCC56324.2022.10065945).
- [35] M. KOK, J. D. HOL et T. B. SCHÖN. *Foundations et Trends in Signal Processing*, 2017, t. 11, p. 1-153. DOI : [10.1561/20000000094](https://doi.org/10.1561/20000000094).
- [36] F. SAHIN, « Long-Range, High-Resolution Camera Optical Design for Assisted and Autonomous Driving », *Photonics*, t. 6, p. 73, juin 2019. DOI : [10.3390/photonics6020073](https://doi.org/10.3390/photonics6020073).
- [37] S.-W. KO, H. CHAE, K. HAN, S. LEE, D.-W. SEO et K. HUANG, « V2X-Based Vehicular Positioning: Opportunities, Challenges, and Future Directions », *IEEE Wireless Communications*, t. 28, n° 2, p. 144-151, 2021. DOI : [10.1109/MWC.001.2000259](https://doi.org/10.1109/MWC.001.2000259).
- [38] M. CUI, S. ZHONG, B. LI, X. CHEN et K. HUANG, « Offloading Autonomous Driving Services via Edge Computing », *IEEE Internet of Things Journal*, t. 7, n° 10, p. 10 535-10 547, 2020. DOI : [10.1109/JIOT.2020.3001218](https://doi.org/10.1109/JIOT.2020.3001218).
- [39] *ITS-OB4, Powerful V2X Onboard Unit*, Commsignia, nov. 2020. adresse : <https://www.commsignia.com/products/obu/>.
- [40] *MK6-OBU*, Cohda Wireless, jan. 2023. adresse : <https://www.cohdawireless.com/solutions/hardware/mk6-obu/>.
- [41] *V2X On-Board Unit*, iWave, juin 2021. adresse : <https://www.iwavesystems.com/product/v2x-connectivity-hub/>.
- [42] A. AL-FUQAHA, M. GUIZANI, M. MOHAMMADI, M. ALEDHARI et M. AYYASH, « Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications », *IEEE Communications Surveys & Tutorials*, t. 17, n° 4, p. 2347-2376, 2015. DOI : [10.1109/COMST.2015.2444095](https://doi.org/10.1109/COMST.2015.2444095).
- [43] A. ZANELLA, N. BUI, A. CASTELLANI, L. VANGELISTA et M. ZORZI, « Internet of Things for Smart Cities », *IEEE Internet of Things Journal*, t. 1, n° 1, p. 22-32, 2014. DOI : [10.1109/JIOT.2014.2306328](https://doi.org/10.1109/JIOT.2014.2306328).

- [44] N. LU, N. CHENG, N. ZHANG, X. SHEN et J. MARK, « Connected Vehicles: Solutions and Challenges », *IEEE Internet of Things Journal*, t. 1, n° 4, p. 289-299, 2014. DOI : [10.1109/JIOT.2014.2327587](https://doi.org/10.1109/JIOT.2014.2327587).
- [45] K. NASEER QURESHI, F. BASHIR et S. IQBAL, « Cloud Computing Model for Vehicular Ad hoc Networks », in *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, 2018, p. 1-3. DOI : [10.1109/CloudNet.2018.8549536](https://doi.org/10.1109/CloudNet.2018.8549536).
- [46] R. SHARMA, S. KUMAR et M. C. TRIVEDI, « Mobile Cloud Computing: A Needed Shift from Cloud to Mobile Cloud », in *2013 5th International Conference and Computational Intelligence and Communication Networks*, 2013, p. 536-539. DOI : [10.1109/CICN.2013.116](https://doi.org/10.1109/CICN.2013.116).
- [47] N. ABBAS, Y. ZHANG, A. TAHERKORDI et T. SKEIE, « Mobile Edge Computing: A Survey », *IEEE Internet of Things Journal*, t. 5, n° 1, p. 450-465, 2018. DOI : [10.1109/JIOT.2017.2750180](https://doi.org/10.1109/JIOT.2017.2750180).
- [48] A. BOUKERCHE et R. DE GRANDE, « Vehicular cloud computing: Architectures, applications, and mobility », *Computer Networks*, t. 135, p. 171-189, 2018. DOI : <https://doi.org/10.1016/j.comnet.2018.01.004>.
- [49] M. GERLA, « Vehicular Cloud Computing », in *2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2012, p. 152-155. DOI : [10.1109/MedHocNet.2012.6257116](https://doi.org/10.1109/MedHocNet.2012.6257116).
- [50] M. BAHRAMI, « Cloud Computing for Emerging Mobile Cloud Apps », in *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, 2015, p. 4-5. DOI : [10.1109/MobileCloud.2015.40](https://doi.org/10.1109/MobileCloud.2015.40).
- [51] A. REZNIK. « What is Edge? » (Mai 2018), adresse : <https://www.etsi.org/newsroom/blogs/entry/what-is-edge>.
- [52] A. REZNIK, *ETSI - Multi-access Edge Computing - Standards for MEC*, Accessed on 03.11.2021, 2021. adresse : <https://www.etsi.org/technologies/multi-access-edge-computing>.
- [53] C. LIN, D. DENG et C. YAO, « Resource Allocation in Vehicular Cloud Computing Systems With Heterogeneous Vehicles and Roadside Units », *IEEE Internet of Things Journal*, t. 5, n° 5, p. 3692-3700, 2018. DOI : [10.1109/JIOT.2017.2690961](https://doi.org/10.1109/JIOT.2017.2690961).
- [54] R. DZIYAUDDIN, D. NIYATO, N. CONG LUONG, M. MOHD IZHAR, M. HADHARI et S. DAUD, *Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey*, 2019. arXiv : [1912.07803 \[cs.NI\]](https://arxiv.org/abs/1912.07803).
- [55] ETSI, *Multi-access Edge Computing (MEC); Proof of Concept Framework*, GS MEC-IEG 005 V1.1.1 (2015-08), août 2015.
- [56] F. SPINELLI et V. MANCUSO, « Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility », *IEEE Communications Surveys & Tutorials*, t. 23, n° 1, p. 596-630, 2021. DOI : [10.1109/COMST.2020.3037674](https://doi.org/10.1109/COMST.2020.3037674).
- [57] P. MACH et Z. BECVAR, « Mobile Edge Computing: A Survey on Architecture and Computation Offloading », *IEEE Communications Surveys & Tutorials*, t. 19, n° 3, p. 1628-1656, 2017. DOI : [10.1109/COMST.2017.2682318](https://doi.org/10.1109/COMST.2017.2682318).
- [58] X. CHEN, L. JIAO, W. LI et X. FU, « Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing », *IEEE/ACM Transactions on Networking*, t. 24, n° 5, p. 2795-2808, 2016. DOI : [10.1109/TNET.2015.2487344](https://doi.org/10.1109/TNET.2015.2487344).
- [59] ETSI, *Multi-access Edge Computing (MEC); V2X Information Service API*, GS MEC 030 V2.1.1 (2020-04), avr. 2020.

- [60] ETSI, *Multi-access Edge Computing (MEC); Framework and Reference Architecture*, GS MEC 003 V2.2.1 (2020-12), déc. 2020.
- [61] ETSI, *Multi-access Edge Computing (MEC); MEC 5G Integration*, GR MEC 031 - V2.1.1, oct. 2020.
- [62] S. ABDEL HAKEEM, A. HADY et H. KIM, « 5G-V2X: standardization, architecture, use cases, network-slicing, and edge-computing », *Wireless Networks*, t. 26, p. 6015-6041, 2020. DOI : [10.1007/s11276-020-02419-8](https://doi.org/10.1007/s11276-020-02419-8).
- [63] M. S. ASLANPOUR, S. S. GILL et A. TOOSI, « Performance Evaluation Metrics for Cloud, Fog and Edge Computing: A Review, Taxonomy, Benchmarks and Standards for Future Research », *Internet of Things*, t. 12, p. 100 273, août 2020. DOI : [10.1016/j.iot.2020.100273](https://doi.org/10.1016/j.iot.2020.100273).
- [64] OPENFOG CONSORTIUM ARCHITECTURE WORKING GROUP, « OpenFog Reference Architecture for Fog Computing », OpenFog Consortium, rapp. tech., 2017.
- [65] B. BARON, P. SPATHIS, H. RIVANO et M. D. DE AMORIM, « Offloading Massive Data Onto Passenger Vehicles: Topology Simplification and Traffic Assignment », *IEEE/ACM Transactions on Networking*, t. 24, n° 6, p. 3248-3261, 2016. DOI : [10.1109/TNET.2016.2518926](https://doi.org/10.1109/TNET.2016.2518926).
- [66] M. AAZAM et E. HUH, « Dynamic resource provisioning through Fog micro datacenter », in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, p. 105-110. DOI : [10.1109/PERCOMW.2015.7134002](https://doi.org/10.1109/PERCOMW.2015.7134002).
- [67] X. HOU, Y. LI, M. CHEN, D. WU, D. JIN et S. CHEN, « Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures », *IEEE Transactions on Vehicular Technology*, t. 65, n° 6, p. 3860-3873, 2016. DOI : [10.1109/TVT.2016.2532863](https://doi.org/10.1109/TVT.2016.2532863).
- [68] F. DRESSLER, G. PANNU, F. HAGENAUER, M. GERLA, T. HIGUCHI et O. ALTINTAS, « Virtual Edge Computing Using Vehicular Micro Clouds », in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, p. 537-541. DOI : [10.1109/ICCNC.2019.8685481](https://doi.org/10.1109/ICCNC.2019.8685481).
- [69] ETSI, *Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases*, GR MEC 022 - V2.1.1, sept. 2018.
- [70] C. MELLOR. « Autonomous vehicle data storage: We grill self-driving car experts about sensors, clouds ... and robo taxis ». (2021), adresse : <https://blocksandfiles.com/2020/02/03/autonomous-vehicle-data-storage-is-a-game-of-guesses/>.
- [71] L. LIU, C. CHEN, P. QINGQI, M. SABITA et Z. YAN, « Vehicular Edge Computing and Networking: A Survey », *Mobile Networks and Applications*, 2019. DOI : [10.1007/s11036-020-01624-1](https://doi.org/10.1007/s11036-020-01624-1).
- [72] N. XIA et C.-S. YANG, « Vehicular Communications: Standards and Challenges », in *Internet of Vehicles. Technologies and Services for Smart Cities*, Cham : Springer International Publishing, 2017, p. 1-12, ISBN : 978-3-319-72329-7.
- [73] S. T. MAGULURI, R. SRIKANT et L. YING, « Stochastic models of load balancing and scheduling in cloud computing clusters », in *2012 Proceedings IEEE INFOCOM*, Orlando, FL, USA : IEEE, mars 2012, p. 702-710. DOI : [10.1109/INFCOM.2012.6195815](https://doi.org/10.1109/INFCOM.2012.6195815). adresse : <http://ieeexplore.ieee.org/document/6195815/>.
- [74] M. MITZENMACHER, « The power of two choices in randomized load balancing », *IEEE Transactions on Parallel and Distributed Systems*, t. 12, n° 10, p. 1094-1104, oct. 2001. DOI : [10.1109/71.963420](https://doi.org/10.1109/71.963420). adresse : <http://ieeexplore.ieee.org/document/963420/>.

- [75] S. R. TURNER, « The Effect of Increasing Routing Choice on Resource Pooling », *Probability in the Engineering and Informational Sciences*, t. 12, n° 1, p. 109-124, jan. 1998. DOI : [10.1017/S0269964800005088](https://doi.org/10.1017/S0269964800005088).
- [76] A. MUKHOPADHYAY, R. R. MAZUMDAR et F. GUILLEMIN, « The Power of Randomized Routing in Heterogeneous Loss Systems », in *2015 27th International Teletraffic Congress*, Ghent, Belgium : IEEE, sept. 2015, p. 125-133. DOI : [10.1109/ITC.2015.22](https://doi.org/10.1109/ITC.2015.22).
- [77] Q. YUAN, H. ZHOU, J. LI, Z. LIU, F. YANG et X. S. SHEN, « Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution », *IEEE Network*, t. 32, n° 1, p. 80-86, 2018. DOI : [10.1109/MNET.2018.1700105](https://doi.org/10.1109/MNET.2018.1700105).
- [78] J. GUO, W. LUO, B. SONG, F. R. YU et X. DU, « Intelligence-Sharing Vehicular Networks with Mobile Edge Computing and Spatiotemporal Knowledge Transfer », *IEEE Network*, t. 34, n° 4, p. 256-262, 2020. DOI : [10.1109/MNET.001.1900512](https://doi.org/10.1109/MNET.001.1900512).
- [79] M. JIANG, T. WU, Z. WANG, Y. GONG, L. ZHANG et R. P. LIU, « A Multi-Intersection Vehicular Cooperative Control Based on End-Edge-Cloud Computing », *IEEE Transactions on Vehicular Technology*, t. 71, n° 3, p. 2459-2471, mars 2022. DOI : [10.1109/TVT.2022.3143828](https://doi.org/10.1109/TVT.2022.3143828).
- [80] Z. QIN, S. LENG, J. ZHOU et S. MAO, « Collaborative Edge Computing and Caching in Vehicular Networks », in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, p. 1-6. DOI : [10.1109/WCNC45663.2020.9120600](https://doi.org/10.1109/WCNC45663.2020.9120600).
- [81] Q. LUO, C. LI, T. H. LUAN et W. SHI, « Collaborative Data Scheduling for Vehicular Edge Computing via Deep Reinforcement Learning », *IEEE Internet of Things Journal*, t. 7, n° 10, p. 9637-9650, 2020. DOI : [10.1109/JIOT.2020.2983660](https://doi.org/10.1109/JIOT.2020.2983660).
- [82] B. WU, T. CHEN, K. YANG et X. WANG, « Edge-Centric Bandit Learning for Task-Offloading Allocations in Multi-RAT Heterogeneous Networks », *IEEE Transactions on Vehicular Technology*, t. 70, n° 4, p. 3702-3714, 2021. DOI : [10.1109/TVT.2021.3062634](https://doi.org/10.1109/TVT.2021.3062634).
- [83] L. LIU, C. CHEN, P. QINGQI, M. SABITA et Z. YAN, « Vehicular Edge Computing and Networking: A Survey », *Mobile Networks and Applications*, 2019. DOI : [10.1007/s11036-020-01624-1](https://doi.org/10.1007/s11036-020-01624-1).
- [84] H. SAMI, A. MOURAD et W. EL-HAJJ, « Vehicular-OBUs-As-On-Demand-Fogs: Resource and Context Aware Deployment of Containerized Micro-Services », en, *IEEE/ACM Transactions on Networking*, t. 28, n° 2, p. 778-790, avr. 2020. DOI : [10.1109/TNET.2020.2973800](https://doi.org/10.1109/TNET.2020.2973800).
- [85] J. WANG, K. ZHU, B. CHEN et Z. HAN, « Distributed Clustering-Based Cooperative Vehicular Edge Computing for Real-Time Offloading Requests », *IEEE Transactions on Vehicular Technology*, t. 71, n° 1, p. 653-669, jan. 2022. DOI : [10.1109/TVT.2021.3122001](https://doi.org/10.1109/TVT.2021.3122001).
- [86] H. LI, *A Novel Solution for Uu Interface Based C-V2X*, 2021. DOI : [10.48550/ARXIV.2109.05684](https://doi.org/10.48550/ARXIV.2109.05684). adresse : <https://arxiv.org/abs/2109.05684>.
- [87] G. S. PANNU, S. UCAR, T. HIGUCHI, O. ALTINTAS et F. DRESSLER, « Vehicular Virtual Edge Computing using Heterogeneous V2V and V2C Communication », en, in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, mai 2022, p. 1-2. DOI : [10.1109/INFOCOMWKSHPS54753.2022.9798362](https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798362).
- [88] Z. DENG, Z. CAI et M. LIANG, « A Multi-Hop VANETs-Assisted Offloading Strategy in Vehicular Mobile Edge Computing », *IEEE Access*, t. 8, p. 53 062-53 071, 2020. DOI : [10.1109/ACCESS.2020.2981501](https://doi.org/10.1109/ACCESS.2020.2981501).

- [89] R. XIE, Q. TANG, Q. WANG, X. LIU, F. R. YU et T. HUANG, « Collaborative Vehicular Edge Computing Networks: Architecture Design and Research Challenges », *IEEE Access*, t. 7, p. 178 942-178 952, 2019. DOI : [10.1109/ACCESS.2019.2957749](https://doi.org/10.1109/ACCESS.2019.2957749).
- [90] E. TAYLOR. « VW says rollout of driverless vehicles will be limited by high costs ». (2021), adresse : <https://europe.autonews.com/automakers/vw-says-rollout-driverless-vehicles-will-be-limited-high-costs>.
- [91] J. ZHAO, Q. LI, Y. GONG et K. ZHANG, « Computation Offloading and Resource Allocation For Cloud-Assisted Mobile Edge Computing in Vehicular Networks », *IEEE Transactions on Vehicular Technology*, t. 68, n° 8, p. 7944-7956, 2019. DOI : [10.1109/TVT.2019.2917890](https://doi.org/10.1109/TVT.2019.2917890).
- [92] X. CHEN, N. THOMAS, T. ZHAN et J. DING, « A Hybrid Task Scheduling Scheme for Heterogeneous Vehicular Edge Systems », *IEEE Access*, t. 7, p. 117 088-117 099, 2019. DOI : [10.1109/ACCESS.2019.2934890](https://doi.org/10.1109/ACCESS.2019.2934890).
- [93] Z. WANG, S. ZHENG, Q. GE et K. LI, « Online Offloading Scheduling and Resource Allocation Algorithms for Vehicular Edge Computing System », *IEEE Access*, t. 8, p. 52 428-52 442, 2020. DOI : [10.1109/ACCESS.2020.2981045](https://doi.org/10.1109/ACCESS.2020.2981045).
- [94] J. ZHAO, X. SUN, Q. LI et X. MA, « Edge Caching and Computation Management for Real-Time Internet of Vehicles: An Online and Distributed Approach », *IEEE Transactions on Intelligent Transportation Systems*, t. 22, n° 4, p. 2183-2197, 2021. DOI : [10.1109/TITS.2020.3012966](https://doi.org/10.1109/TITS.2020.3012966).
- [95] E. KASSENS-NOOR, Z. KOTVAL-KARAMCHANDANI et M. CAI, « Willingness to ride and perceptions of autonomous public transit », *Transportation Research Part A: Policy and Practice*, t. 138, p. 92-104, 2020. DOI : [10.1016/j.tra.2020.05.010](https://doi.org/10.1016/j.tra.2020.05.010).
- [96] L. MU, B. GE, C. XIA et C. WU, « Deep reinforcement learning based adaptive threshold multi-tasks offloading approach in MEC », *Comput. Networks*, t. 232, p. 109 803, 2023. DOI : [10.1016/j.comnet.2023.109803](https://doi.org/10.1016/j.comnet.2023.109803).
- [97] 3GPP, « 36.885 - Study on LTE-based V2X Services (Release 14) », 3rd Generation Partnership Project, rapp. tech., 2016, Rel.14 v14.0.0.
- [98] 5GAA, « Cellular-Vehicle-to-Everything (C-V2X): today and next steps », 5G Automotive Association, rapp. tech., juill. 2020.
- [99] L. CITY. « Smart City à Sète : 5 années d'une stratégie éprouvée ». (jan. 2023), adresse : <https://www.lacroix-city.fr/smart-stories/strategie-fr/smart-city-a-sete-5-annees-dune-strategie-eprouvee/>.
- [100] J. MAROUANI. « Lacroix City retenu par la RATP pour déployer son dispositif de communication PrioV2X ». (mai 2023), adresse : <https://www.pei-france.com/article/lacroix-city-retenu-par-la-ratp-pour-deployer-son-dispositif-de-communication-priov2x/>.
- [101] X. HE, R. JIN et H. DAI, « Multi-Hop Task Offloading With On-the-Fly Computation for Multi-UAV Remote Edge Computing », *IEEE Transactions on Communications*, t. 70, n° 2, p. 1332-1344, 2022. DOI : [10.1109/TCOMM.2021.3129902](https://doi.org/10.1109/TCOMM.2021.3129902).
- [102] M. ADAMU et A. ADEWUMI, « Minimizing the weighted number of tardy jobs on multiple machines: A review », *Journal of Industrial and Management Optimization*, t. 12, n° 4, p. 1465-1493, 2016.
- [103] A. VARGA, « OMNeT++ », in *Modeling and Tools for Network Simulation*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, p. 35-59, ISBN : 978-3-642-12331-3. DOI : [10.1007/978-3-642-12331-3_3](https://doi.org/10.1007/978-3-642-12331-3_3). adresse : https://doi.org/10.1007/978-3-642-12331-3_3.

- [104] A. VIRDIS, G. STEA et G. NARDINI, « Simulating LTE/LTE-Advanced Networks with SimuLTE », in jan. 2016, ISBN : 978-3-319-26470-7. DOI : [10.1007/978-3-319-26470-7](https://doi.org/10.1007/978-3-319-26470-7).
- [105] C. SOMMER, R. GERMAN et F. DRESSLER, « Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis », *IEEE Transactions on Mobile Computing (TMC)*, t. 10, n° 1, p. 3-15, jan. 2011. DOI : [10.1109/TMC.2010.133](https://doi.org/10.1109/TMC.2010.133).
- [106] P. A. LOPEZ, M. BEHRISCH, L. BIEKER-WALZ et al., « Microscopic Traffic Simulation using SUMO », in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, p. 2575-2582. DOI : [10.1109/ITSC.2018.8569938](https://doi.org/10.1109/ITSC.2018.8569938).
- [107] A. MORATTAB, Z. DZIONG et K. SOHRABY, « Mode selection map-based vertical handover in D2D enabled 5G networks », *IET Communications*, t. 13, n° 14, p. 2173-2185, août 2019. DOI : [10.1049/iet-com.2018.6133](https://doi.org/10.1049/iet-com.2018.6133).
- [108] J. ZHANG, H. GUO, J. LIU et Y. ZHANG, « Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution », *IEEE Transactions on Vehicular Technology*, t. 69, n° 2, p. 2092-2104, 2020. DOI : [10.1109/TVT.2019.2959410](https://doi.org/10.1109/TVT.2019.2959410).
- [109] L. TANG, B. TANG, L. ZHANG, F. GUO et H. HE, « Joint optimization of network selection and task offloading for vehicular edge computing », *Journal of Cloud Computing*, t. 10, 2021. DOI : [10.1186/s13677-021-00240-y](https://doi.org/10.1186/s13677-021-00240-y).
- [110] B. LI, F. CHEN, Z. PENG, P. HOU et H. DING, « Mobility-aware dynamic offloading strategy for C-V2X under multi-access edge computing », *Physical Communication*, t. 49, p. 101 446, 2021. DOI : <https://doi.org/10.1016/j.phycom.2021.101446>.
- [111] OPENSTREETMAP CONTRIBUTORS, *Planet dump retrieved from https://planet.osm.org, https://www.openstreetmap.org*, 2017.
- [112] M. DE PAOLA, R. FANTINI, R. GAVAZZI, M. MADELLA, P. SCALAMBRO et S. SCARPINA, « Initial report on 5G Technological Enablers for CCAM », The 5G Infrastructure Public Private Partnership, rapp. tech., 2019.
- [113] A. AL-FUQAHA, M. GUIZANI, M. MOHAMMADI, M. ALEDHARI et M. AYYASH, « Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications », *IEEE Communications Surveys & Tutorials*, t. 17, n° 4, p. 2347-2376, 2015. DOI : [10.1109/COMST.2015.2444095](https://doi.org/10.1109/COMST.2015.2444095).
- [114] A. ZANELLA, N. BUI, A. CASTELLANI, L. VANGELISTA et M. ZORZI, « Internet of Things for Smart Cities », *IEEE Internet of Things Journal*, t. 1, n° 1, p. 22-32, 2014. DOI : [10.1109/JIOT.2014.2306328](https://doi.org/10.1109/JIOT.2014.2306328).
- [115] N. LU, N. CHENG, N. ZHANG, X. SHEN et J. MARK, « Connected Vehicles: Solutions and Challenges », *IEEE Internet of Things Journal*, t. 1, n° 4, p. 289-299, 2014. DOI : [10.1109/JIOT.2014.2327587](https://doi.org/10.1109/JIOT.2014.2327587).
- [116] C. BERNSTEIN. « What is radio access network (RAN)? » (2021), adresse : <https://searchnetworking.techtarget.com/definition/%20radio-access-network-RAN>.
- [117] A. JAFARI, S. AL-KHAYATT et A. DOGMAN, « Performance Evaluation of IEEE 802.11p for Vehicular Communication Networks », juill. 2012. DOI : [10.1109/CSNDSP.2012.6292712](https://doi.org/10.1109/CSNDSP.2012.6292712).
- [118] L. MIAO, K. DJOUANI, B. VAN WYK et Y. HAMAM, « A Survey of IEEE 802.11p MAC Protocol », *Journal of Selected Areas in Telecommunications*, sept. 2011.
- [119] D. MANIVANNAN, S. S. MONI et S. ZEADALLY, « Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETWORKS (VANETs) », *Vehicular Communications*, t. 25, p. 100 247, 2020. DOI : <https://doi.org/10.1016/j.vehcom.2020.100247>.

- [120] « IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture », *IEEE Std 1609.0-2013*, p. 1-78, 2014. DOI : [10.1109/IEEESTD.2014.6755433](https://doi.org/10.1109/IEEESTD.2014.6755433).
- [121] B. LONC et P. CINCILLA, « Cooperative ITS security framework: Standards and implementations progress in Europe », in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, p. 1-6. DOI : [10.1109/WoWMoM.2016.7523576](https://doi.org/10.1109/WoWMoM.2016.7523576).
- [122] ETSI, *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, EN 302 637-2 V1.3.0, sept. 2014.
- [123] S. LEE et A. LIM, « An Empirical Study on Ad Hoc Performance of DSRC and Wi-Fi Vehicular Communications », *International Journal of Distributed Sensor Networks*, t. 9, n° 11, p. 482-695, 2013. DOI : [10.1155/2013/482695](https://doi.org/10.1155/2013/482695).
- [124] D. SHEPARDSON. « U.S. court upholds FCC reallocation of auto safety spectrum ». (août 2022), adresse : <https://www.reuters.com/legal/us-court-upholds-fcc-reallocation-auto-safety-spectrum-2022-08-12/>.
- [125] G. NAIK, B. CHOUDHURY et J.-M. PARK, « IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications », *IEEE Access*, t. 7, p. 70 169-70 184, 2019. DOI : [10.1109/ACCESS.2019.2919489](https://doi.org/10.1109/ACCESS.2019.2919489).
- [126] B. YACHEUR, T. AHMED et M. MOSBAH, « Analysis and Comparison of IEEE 802.11p and IEEE 802.11bd », in *Communication Technologies for Vehicles*, Cham : Springer International Publishing, 2020, p. 55-65, ISBN : 978-3-030-66030-7.
- [127] B. Y. YACHEUR, T. AHMED et M. MOSBAH, « Implementation and Assessment of IEEE 802.11BD for Improved Road Safety », in *2021 IEEE 18th Annual Consumer Communications and Networking Conference (CCNC)*, 2021, p. 1-6. DOI : [10.1109/CCNC49032.2021.9369649](https://doi.org/10.1109/CCNC49032.2021.9369649).
- [128] R. JACOB, N. SCHWARZENBERG, F. BURMEISTER et G. FETTWEIS, « Congestion-aware Packet Repetitions for IEEE 802.11bd-based Safety-critical V2V Communications », in *ICC 2022 - IEEE International Conference on Communications*, 2022, p. 315-321. DOI : [10.1109/ICC45855.2022.9838576](https://doi.org/10.1109/ICC45855.2022.9838576).
- [129] M. PATLAYENKO, O. OSHAROVSKA et V. SOLODKA, « Comparison of LTE Coverage Areas in Three Frequency Bands », in *2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT)*, 2021, p. 212-215. DOI : [10.1109/AICT52120.2021.9628960](https://doi.org/10.1109/AICT52120.2021.9628960).
- [130] Z. XU, X. LI, X. ZHAO, M. ZHANG et Z. WANG, « DSRC versus 4G-LTE for Connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance », *Journal of advanced transportation*, t. 435, août 2017. DOI : [10.1155/2017/2750452](https://doi.org/10.1155/2017/2750452).
- [131] 3GPP, « Vehicle-to-everything (V2X); Media handling and interaction », 3rd Generation Partnership Project, rapp. tech., nov. 2020, TR 26.985 V16.0.0 Rel. 16.
- [132] R. MOLINA-MASEGOSA, J. GOZALVEZ et M. SEPULCRE, « Configuration of the C-V2X Mode 4 Sidelink PC5 Interface for Vehicular Communication », in *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2018, p. 43-48. DOI : [10.1109/MSN.2018.00014](https://doi.org/10.1109/MSN.2018.00014).
- [133] G. P. WIJESIRI et J. HAAPOLA, « A Discrete-Time Markov Chain Based Comparison of the MAC Layer Performance of C-V2X Mode 4 and IEEE 802.11p », *IEEE Transactions on Communications*, avr. 2021. DOI : [10.1109/TCOMM.2020.3044340](https://doi.org/10.1109/TCOMM.2020.3044340).

- [134] Q. PHAM, F. FANG, V. N. HA et al., « A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art », *IEEE Access*, t. 8, p. 116974-117017, 2020. DOI : [10.1109/ACCESS.2020.3001277](https://doi.org/10.1109/ACCESS.2020.3001277).
- [135] J. WANG, J. LIU et N. KATO, « Networking and Communications in Autonomous Driving: A Survey », *IEEE Communications Surveys & Tutorials*, t. 21, n° 2, p. 1243-1274, 2019. DOI : [10.1109/COMST.2018.2888904](https://doi.org/10.1109/COMST.2018.2888904).
- [136] C. ZHAO, Y. CAI, A. LIU, M. ZHAO et L. HANZO, « Mobile Edge Computing Meets mmWave Communications: Joint Beamforming and Resource Allocation for System Delay Minimization », *IEEE Transactions on Wireless Communications*, t. 19, n° 4, p. 2382-2396, 2020. DOI : [10.1109/TWC.2020.2964543](https://doi.org/10.1109/TWC.2020.2964543).
- [137] M. FALLGREN, T. ABBAS, S. ALLIO et al., « Multicast and Broadcast Enablers for High-Performing Cellular V2X Systems », *IEEE Transactions on Broadcasting*, t. 65, n° 2, p. 454-463, 2019. DOI : [10.1109/TBC.2019.2912619](https://doi.org/10.1109/TBC.2019.2912619).
- [138] M. M. ELSAYED, K. M. HOSNY et M. M. FOUDA, « Vehicles communications handover in 5G: A survey », *ICT Express*, jan. 2022. DOI : [10.1016/j.icte.2022.01.005](https://doi.org/10.1016/j.icte.2022.01.005).
- [139] ETSI, *5G; Service requirements for enhanced V2X scenarios*, TS 22.186 V15.4.0 Rel. 15, oct. 2018.
- [140] M. H. C. GARCIA, A. MOLINA-GALAN, M. BOBAN et al., « A Tutorial on 5G NR V2X Communications », *IEEE Communications Surveys & Tutorials*, t. 23, n° 3, p. 1972-2026, 2021. DOI : [10.1109/COMST.2021.3057017](https://doi.org/10.1109/COMST.2021.3057017).
- [141] K. GANESAN, P. B. MALLICK, J. LÖHR, D. KARAMPATSI et A. KUNZ, « 5G V2X Architecture and Radio Aspects », in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2019, p. 1-6. DOI : [10.1109/CSCN.2019.8931319](https://doi.org/10.1109/CSCN.2019.8931319).
- [142] J. YAN et J. HÄRRI, « On the Feasibility of URLLC for 5G-NR V2X Sidelink Communication at 5.9 GHz », in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, p. 3599-3604. DOI : [10.1109/GLOBECOM48099.2022.10000606](https://doi.org/10.1109/GLOBECOM48099.2022.10000606).
- [143] M. HAROUNABADI, D. M. SOLEYMANI et S. BHADAURIA, « V2X in 3GPP Standardization: NR Sidelink in Release-16 and Beyond », *IEEE Communications Standards Magazine*, t. 5, n° 1, p. 12-21, 2021. DOI : [10.1109/MCOMSTD.001.2000070](https://doi.org/10.1109/MCOMSTD.001.2000070).
- [144] T. T. ALMEIDA, L. de C. GOMES, F. M. ORTIZ, J. R. JÚNIOR et L. H. M. K. COSTA, « IEEE 802.11p Performance Evaluation: Simulations vs. Real Experiments », in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, p. 3840-3845. DOI : [10.1109/ITSC.2018.8569676](https://doi.org/10.1109/ITSC.2018.8569676).
- [145] K. KIELA et V. BARZDENAS, « Review of V2X-IoT Standards and Frameworks for ITS Applications », *Applied Sciences*, t. 10, juin 2020. DOI : [10.3390/app10124314](https://doi.org/10.3390/app10124314).
- [146] J. WU, Z. ZHANG, Y. HONG et Y. WEN, « Cloud radio access network (C-RAN): a primer », *IEEE Network*, t. 29, n° 1, p. 35-41, 2015. DOI : [10.1109/MNET.2015.7018201](https://doi.org/10.1109/MNET.2015.7018201).
- [147] J. CHENG, Y. SHI, B. BAI et W. CHEN, « Computation offloading in Cloud-RAN based mobile cloud computing system », in *2016 IEEE International Conference on Communications (ICC)*, 2016, p. 1-6. DOI : [10.1109/ICC.2016.7511367](https://doi.org/10.1109/ICC.2016.7511367).
- [148] F. GIUST, V. SCIANCALEPORE, D. SABELLA et M. FILIPPOU, « Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars », *IEEE Communications Standards Magazine*, t. 2, mars 2018. DOI : [10.1109/MCOMSTD.2018.1800013](https://doi.org/10.1109/MCOMSTD.2018.1800013).

- [149] A. ALSOHAILY et E. SOUSA, « Unified radio access network operation for multi-radio access technology cellular systems », in *2014 21st international conference on telecommunications (ICT)*, IEEE, 2014, p. 32-36.
- [150] 3. T. 3. T. RAN, *Study on Multiple Radio Access Technology (Multi-RAT) joint coordination (Release 13)*, Rel.13 v13.0.0, 2015.
- [151] Z. H. MIR, J. TOUTOUH, F. FILALI et E. ALBA, « QoS-Aware Radio Access Technology (RAT) Selection in Hybrid Vehicular Networks », in *Communication Technologies for Vehicles*, 2015, p. 117-128.
- [152] M. SEPULCRE et J. GOZALVEZ, « Heterogeneous V2V Communications in Multi-Link and Multi-RAT Vehicular Networks », *IEEE Transactions on Mobile Computing*, t. 20, n° 1, p. 162-173, 2021. DOI : [10.1109/TMC.2019.2939803](https://doi.org/10.1109/TMC.2019.2939803).
- [153] R. JACOB, W. ANWAR, G. FETTWEIS et J. POHLMANN, « Exploiting Multi-RAT Diversity in Vehicular Ad-Hoc Networks to Improve Reliability of Cooperative Automated Driving Applications », in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, p. 1-7. DOI : [10.1109/VTCFall.2019.8891072](https://doi.org/10.1109/VTCFall.2019.8891072).
- [154] H. WU, J. CHEN, W. XU et al., « Delay-Minimized Edge Caching in Heterogeneous Vehicular Networks: A Matching-Based Approach », *IEEE Transactions on Wireless Communications*, t. 19, n° 10, p. 6409-6424, 2020. DOI : [10.1109/TWC.2020.3003339](https://doi.org/10.1109/TWC.2020.3003339).
- [155] M. CHIOSI, D. CLARKE et P. WILLIS, « Network Functions Virtualisation - Introductory White Paper », European Telecommunications Standards Institute, rapp. tech., 2012.
- [156] M. CHIOSI, S. WRIGHT, D. CLARKE et P. WILLIS, « Network Functions Virtualisation - Update White Paper », European Telecommunications Standards Institute, rapp. tech., 2013.
- [157] Y. MA, W. LIANG, J. WU et Z. XU, « Throughput Maximization of NFV-Enabled Multicasting in Mobile Edge Cloud Networks », *IEEE Transactions on Parallel and Distributed Systems*, t. 31, n° 2, p. 393-407, 2020. DOI : [10.1109/TPDS.2019.2937524](https://doi.org/10.1109/TPDS.2019.2937524).
- [158] J. XIAO et J. ZHENG, « A Delay Balanced Adaptive Channel Allocation Mechanism for LTE-U and WiFi Coexistence Systems », *Mobile Networks and Applications*, t. 27, n° 2, p. 457-468, avr. 2022. DOI : [10.1007/s11036-020-01690-5](https://doi.org/10.1007/s11036-020-01690-5).
- [159] Y. HUI, Z. SU, T. H. LUAN et C. LI, « Reservation Service: Trusted Relay Selection for Edge Computing Services in Vehicular Networks », *IEEE Journal on Selected Areas in Communications*, t. 38, n° 12, p. 2734-2746, 2020. DOI : [10.1109/JSAC.2020.3005468](https://doi.org/10.1109/JSAC.2020.3005468).
- [160] S.-J. YOO, D. CYPHER et N. GOLMIE, « Predictive link trigger mechanism for seamless handovers in heterogeneous wireless networks », *Wireless Communications and Mobile Computing*, t. 9, p. 685-703, mai 2009. DOI : [10.1002/wcm.620](https://doi.org/10.1002/wcm.620).
- [161] L. BITTENCOURT, M. LOPES, I. PETRI et O. RANA, « Towards Virtual Machine Migration in Fog Computing », in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, nov. 2015, p. 1-8. DOI : [10.1109/3PGCIC.2015.85](https://doi.org/10.1109/3PGCIC.2015.85).
- [162] M. NGO, T. LUO, H. HOANG et T. QUEK, « Coordinated Container Migration and Base Station Handover in Mobile Edge Computing », in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, déc. 2020. DOI : [10.1109/GLOBECOM42002.2020.9322368](https://doi.org/10.1109/GLOBECOM42002.2020.9322368).

- [163] D. O. RODRIGUES, T. BRAUN, G. MAIA et L. VILLAS, « Towards SDN-enabled RACH-less Make-before-break Handover in C-V2X Scenarios », in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2021, p. 337-344. DOI : [10.1109/WiMob52687.2021.9606283](https://doi.org/10.1109/WiMob52687.2021.9606283).
- [164] K. VASUDEVA, M. SIMSEK et D. LOPEZ-PEREZ, « Analysis of Handover Failures in Heterogeneous Networks With Fading », *IEEE Transactions on Vehicular Technology*, t. 66, p. 6060-6074, juill. 2017. DOI : [10.1109/TVT.2016.2640310](https://doi.org/10.1109/TVT.2016.2640310).
- [165] H. XIAO, X. ZHANG, A. T. CHRONOPOULOS et Z. ZHANG, « Resource Management for Multi-User-Centric V2X Communication in Dynamic Virtual-Cell-Based Ultra-Dense Networks », *IEEE Transactions on Communications*, t. 68, p. 6346-6358, oct. 2020. DOI : [10.1109/TCOMM.2020.3007612](https://doi.org/10.1109/TCOMM.2020.3007612).
- [166] K. LIU, J. K. Y. NG, V. C. S. LEE, S. H. SON et I. STOJMENOVIC, « Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network », *IEEE/ACM Transactions on Networking*, t. 24, n° 3, p. 1759-1773, 2016. DOI : [10.1109/TNET.2015.2432804](https://doi.org/10.1109/TNET.2015.2432804).
- [167] K. LIU, L. FENG, P. DAI, V. C. S. LEE, S. H. SON et J. CAO, « Coding-Assisted Broadcast Scheduling via Memetic Computing in SDN-Based Vehicular Networks », *IEEE Transactions on Intelligent Transportation Systems*, t. 19, n° 8, p. 2420-2431, 2018. DOI : [10.1109/TITS.2017.2748381](https://doi.org/10.1109/TITS.2017.2748381).
- [168] P. BELLAVISTA, A. ZANNI et M. SOLIMANDO, « A migration-enhanced edge computing support for mobile devices in hostile environments », in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, p. 957-962. DOI : [10.1109/IWCMC.2017.7986415](https://doi.org/10.1109/IWCMC.2017.7986415).
- [169] X. WANG, K. WANG, S. WU et al., « Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network », *IEEE Transactions on Parallel and Distributed Systems*, t. 29, n° 11, p. 2429-2445, 2018. DOI : [10.1109/TPDS.2018.2832124](https://doi.org/10.1109/TPDS.2018.2832124).
- [170] M. ZHANG, Y. DOU, P. H. J. CHONG, H. C. B. CHAN et B.-C. SEET, « Fuzzy Logic-Based Resource Allocation Algorithm for V2X Communications in 5G Cellular Networks », *IEEE Journal on Selected Areas in Communications*, t. 39, n° 8, p. 2501-2513, 2021. DOI : [10.1109/JSAC.2021.3087244](https://doi.org/10.1109/JSAC.2021.3087244).
- [171] T. TALEB et A. KSENTINI, « VECOS: A Vehicular Connection Steering Protocol », *IEEE Transactions on Vehicular Technology*, t. 64, n° 3, p. 1171-1187, 2015. DOI : [10.1109/TVT.2014.2327241](https://doi.org/10.1109/TVT.2014.2327241).
- [172] M. SEPULCRE, J. GOZALVEZ, J. HÄRRI et H. HARTENSTEIN, « Contextual Communications Congestion Control for Cooperative Vehicular Networks », *IEEE Transactions on Wireless Communications*, t. 10, n° 2, p. 385-389, 2011. DOI : [10.1109/TWC.2010.120610.100079](https://doi.org/10.1109/TWC.2010.120610.100079).
- [173] H. H. HUSSEIN, M. H. RADWAN, H. A. ELSAYED et S. M. ABD EL-KADER, « Multi V2X channels resource allocation algorithms for D2D 5G network performance enhancement », *Vehicular Communications*, t. 31, p. 100371, 2021. DOI : [10.1016/j.vehcom.2021.100371](https://doi.org/10.1016/j.vehcom.2021.100371).
- [174] W. GONG, X. WANG, M. LI et Z. HUANG, « Round-robin resource sharing algorithm for device-to-device multicast communications underlying single frequency networks », in *2014 21st International Conference on Telecommunications (ICT)*, 2014, p. 191-195. DOI : [10.1109/ICT.2014.6845106](https://doi.org/10.1109/ICT.2014.6845106).

- [175] Z. JING, Q. YANG, M. QIN, J. LI et K. KWAK, « Long-Term Max-Min Fairness Guarantee Mechanism for Integrated Multi-RAT and MEC Networks », *IEEE Transactions on Vehicular Technology*, t. 70, n° 3, p. 2478-2492, 2021. DOI : [10.1109/TVT.2021.3059944](https://doi.org/10.1109/TVT.2021.3059944).
- [176] ETSI, « Harmonizing standards for edge computing - A synergized architecture leveraging ETSI ISG MEC and 3GPP specifications », European Telecommunications Standards Institute, rapp. tech., juill. 2020.
- [177] B. DUCOURTHIAL, V. CHERFAOUI, T. FUHRMANN et S. BONNET, « Experimentation of a road hazard anticipation system based on vehicle cooperation », *Vehicular Communications*, t. 36, p. 100486, 2022. DOI : [10.1016/j.vehcom.2022.100486](https://doi.org/10.1016/j.vehcom.2022.100486).
- [178] M. FALLGREN, B. CELLARIUS, M. DILLINGER, A. FERNANDEZ, Z. LI et S. ALLIO, « White Paper - 5GCAR: Executive Summary », The 5G Infrastructure Public Private Partnership, rapp. tech., 2019.
- [179] E. CORONADO, D. MARANDIN, M. FEMMINELLA, G. REALI et C. CASETTI, « 5G-CARMEN Preliminary System Architecture and Interfaces Specifications », 5G Infrastructure Public Private Partnership, rapp. tech., nov. 2019.
- [180] M. DE PAOLA, R. FANTINI, R. GAVAZZI et M. MADELLA, « 5G-CARMEN: Initial report on 5G Technological Enablers for CCAM », 5G Infrastructure Public Private Partnership, rapp. tech., oct. 2019.
- [181] A. SCHIMPE, I. GONZALEZ VAZQUEZ, A. PFALDER et K. FLOESS, « 5GCroCo: Test Case Definition and Trial Site Description Part 1 », 5G Infrastructure Public Private Partnership, rapp. tech., juill. 2020.
- [182] S. VÍA, M. PAYARÓ, M. MUEHLEISEN, M. FALLGREN(ERI) et K. ECKERT, « 5GCroCo Final Project Report », 5G Infrastructure Public Private Partnership, rapp. tech., août 2022.
- [183] K. TRICHIAS, A. FERNANDEZ BARCIELA et A. HEIDER-AVIET, « White Paper: 5G Trials for Cooperative, Connected and Automated Mobility (CCAM) along European Cross-Border Corridors », The 5G Infrastructure Public Private Partnership, rapp. tech., 2020.
- [184] M. MIKAMI, Y. ISHIDA et K. SERIZAWA, « Field Experimental Trial of Dynamic Mode Switching for 5G NR-V2X Sidelink Communications towards Application to Truck Platooning », in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, mai 2020, p. 1-5. DOI : [10.1109/VTC2020-Spring48590.2020.9128729](https://doi.org/10.1109/VTC2020-Spring48590.2020.9128729).
- [185] M. N. TAHIR, M. KATZ et A. POUTTU, « Analysis of Heterogeneous Wireless Vehicular Communication During Handover Process », in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, Dubrovnik, Croatia : IEEE, mai 2022, p. 754-759. DOI : [10.1109/IWCMC55113.2022.9825206](https://doi.org/10.1109/IWCMC55113.2022.9825206).
- [186] H. T. NGUYEN, M. NOOR-A-RAHIM, Y. L. GUAN et D. PESCH, « Cellular V2X Communications in the Presence of Big Vehicle Shadowing: Performance Analysis and Mitigation », *IEEE Transactions on Vehicular Technology*, t. 72, n° 3, p. 3764-3776, 2023. DOI : [10.1109/TVT.2022.3212704](https://doi.org/10.1109/TVT.2022.3212704).
- [187] D. WANG, R. R. SATTIRAJU et H. D. SCHOTTEN, « Performances of C-V2X Communication on Highway under Varying Channel Propagation Models », in *2018 10th International Conference on Communications, Circuits and Systems (ICCCAS)*, 2018, p. 305-309. DOI : [10.1109/ICCCAS.2018.8768912](https://doi.org/10.1109/ICCCAS.2018.8768912).
- [188] H. J. LAK, A. GHOLAMHOSSEINIAN et J. SEITZ, « Distributed Vehicular Communication Protocols for Autonomous Intersection Management », *Procedia Computer Science*, t. 201, p. 150-157, 2022. DOI : [10.1016/j.procs.2022.03.022](https://doi.org/10.1016/j.procs.2022.03.022).

- [189] X. MENG et D. DUAN, « Multi-vehicle multi-sensor occupancy grid map fusion in vehicular networks », *IET Communications*, t. 16, 2022. DOI : <https://doi.org/10.1049/cmu2.12314>.
- [190] M. LEARN. « Bing Maps Tile System ». (août 2022), adresse : <https://learn.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>.
- [191] D. LIAN, Y. WU, Y. GE, X. XIE et E. CHEN, « Geography-Aware Sequential Location Recommendation », août 2020, p. 2009-2019. DOI : [10.1145/3394486.3403252](https://doi.org/10.1145/3394486.3403252).
- [192] R. SEDAR, F. VÁZQUEZ-GALLEGO, R. CASELLAS et al., « Standards-Compliant Multi-Protocol On-Board Unit for the Evaluation of Connected and Automated Mobility Services in Multi-Vendor Environments », *Sensors*, t. 21, n° 6, 2021. DOI : [10.3390/s21062090](https://doi.org/10.3390/s21062090).
- [193] EASYMILE. « EZ10 passenger shuttle ». (sept. 2020), adresse : <https://easymile.com/vehicle-solutions/ez10-passenger-shuttle>.
- [194] F. ABBAS, G. LIU, P. FAN et Z. KHAN, « An Efficient Cluster-Based Resource Management Scheme and its Performance Analysis for V2X Networks », *IEEE Access*, t. 8, p. 87 071-87 082, 2020. DOI : [10.1109/ACCESS.2020.2992591](https://doi.org/10.1109/ACCESS.2020.2992591). (visité le 28/03/2023).
- [195] S. ERCAN, M. AYAYIDA et N. MESSAI, « How mobile RSUs can enhance communications in VANETs? », in *6th International Conference on Wireless Networks and Mobile Communications, WINCOM 2018, Marrakesh, Morocco, October 16-19, 2018*, IEEE, 2018, p. 1-5. DOI : [10.1109/WINCOM.2018.8629641](https://doi.org/10.1109/WINCOM.2018.8629641).
- [196] F. JAVED, Z. A. KHAN, S. RIZWAN, S. SHAHZADI, N. R. CHAUDHRY et M. IQBAL, « A Novel Energy-Efficient Reservation System for Edge Computing in 6G Vehicular Ad Hoc Network », *Sensors*, t. 23, n° 13, 2023. DOI : [10.3390/s23135817](https://doi.org/10.3390/s23135817).
- [197] B. SATHYAPRAKASH, P. NILANGE, S. PRATHAB, P. PRAJWAL et S. MANOJ, « StaVen: A Secure and Lightweight PWA for Traffic Facilitation », in *2022 4th International Conference on Circuits, Control, Communication and Computing (I4C)*, IEEE, 2022, p. 59-63.
- [198] T. HARDES, F. KLINGLER et C. SOMMER, « Modern WLAN for V2X Applications: Exploiting Beamforming for Platooning », in *IEEE Vehicular Networking Conference, VNC 2020, New York, NY, USA, December 16-18, 2020*, IEEE, 2020, p. 1-8. DOI : [10.1109/VNC51378.2020.9318329](https://doi.org/10.1109/VNC51378.2020.9318329).
- [199] V. ELANGOVAN, W. XIANG et S. LIU, « A Real-Time C-V2X Beamforming Selector Based on Effective Sequence to Sequence Prediction Model Using Transitional Matrix Hard Attention », *IEEE Access*, t. 11, p. 10 954-10 965, 2023. DOI : [10.1109/ACCESS.2023.3241130](https://doi.org/10.1109/ACCESS.2023.3241130).
- [200] L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Mobile edge computing for V2X architectures and applications: A survey », *Computer Networks*, t. 206, p. 108 797, 2022. DOI : [10.1016/j.comnet.2022.108797](https://doi.org/10.1016/j.comnet.2022.108797).
- [201] L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Context-aware task offloading with QoS-provisioning for MEC multi-RAT vehicular networks », in *2022 International Conference on Computer Communications and Networks (ICCCN)*, 2022, p. 1-9. DOI : [10.1109/ICCCN54977.2022.9868873](https://doi.org/10.1109/ICCCN54977.2022.9868873).
- [202] L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Field Trial for Enhanced V2X Multi-RAT Handover in Autonomous Vehicle Networks », in *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, 2023, p. 1-8. DOI : [10.1109/LCN58197.2023.10223342](https://doi.org/10.1109/LCN58197.2023.10223342).

-
- [203] L. BRÉHON--GRATALOUP, R. KACIMI et A.-L. BEYLOT, « Multi-RAT-Enabled Edge Computing for Vehicle-to-Everything Architectures », *Version étendue* — Révision mineure.

Résumé

Cette thèse s'intéresse au déploiement d'une architecture de communication V2X afin de faciliter l'insertion de véhicules autonomes dans une infrastructure réseau hétérogène. L'ajout de connectivité permettrait à une grande diversité d'applications de satisfaire leurs contraintes strictes de performance, en allégeant la charge induite sur les unités de calcul embarquées. Nous nous intéressons en particulier à des conditions de déploiement complexes : celles d'un milieu fortement urbanisé, pouvant grandement bénéficier de services de conduite autonome. Nous proposons alors deux systèmes répondant à des besoins concrets et d'actualité. Le premier est un système d'optimisation de délestage de tâches fondé sur la dissémination fréquente d'informations de contexte. Le second est un système de stabilisation de connectivité réseau pour des véhicules évoluant dans un milieu à fortes perturbations physiques. Au coeur de notre approche, nous tirons profit de la multiplicité des technologies d'accès radio ainsi que des spécificités des communications up, down et sidelink. Ces solutions ont soit été validées par simulation sur de multiples échelles, soit implantées sur du matériel de pointe dans le cadre d'expérimentations en conditions réelles.

Mots clés : Réseaux véhiculaires, Communication V2X, Calcul en bordure, Allocation de ressources, Mobilité

Abstract

This thesis focuses on the deployment of a V2X communication architecture to facilitate the insertion of autonomous vehicles into a heterogeneous network infrastructure. The presence of connectivity would enable a wide variety of applications to satisfy their strict performance constraints, while alleviating the load induced on on-board computing units. We are particularly interested in complex deployment conditions : those of a highly urbanized environment, which can greatly benefit from autonomous driving services. We are proposing two systems to meet concrete, topical needs. The first is a task offloading optimization scheme based on the frequent dissemination of context information. The second is a network connectivity stabilization solution for vehicles operating in an environment with strong network disturbances on the physical layer. At the core of our approach, we take advantage of the multiplicity of radio access technologies, as well as the specificities of up, down and sidelink communications. These solutions have either been validated by simulation on multiple scales, or implemented on state-of-the-art equipment in real-life experiments.

Keywords : Vehicular networks, V2X Communication, Edge computing, Network resource allocation, Mobility

