



Study about interactions between Clouds and Vehicular Networks

Geoffrey Wilhelm

► To cite this version:

Geoffrey Wilhelm. Study about interactions between Clouds and Vehicular Networks. Informatique [cs]. Université de Reims Champagne-Ardenne, 2018. Français. NNT: . tel-02883236

HAL Id: tel-02883236

<https://hal.science/tel-02883236>

Submitted on 28 Jun 2020

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

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Discipline : INFORMATIQUE

Spécialité : Informatique

Présentée et soutenue publiquement par

GEOFFREY WILHELM

Le 6 décembre 2018

Etudes des interactions entre réseaux véhiculaires et Clouds

Thèse dirigée par **HACENE FOUCHAL**

JURY

M. Charles Tatkeu,	Directeur de Recherche,	IFSTTAR,	Président
M. Hacène Fouchal,	Professeur,	Université de Reims,	Directeur de thèse
M. Marwane Ayaida,	Maître de Conférences,	Université de Reims,	Co-Directeur de thèse
Mme Paola Flocchini,	Professeur,	Université d'Ottawa,	Rapporteur
Mme Selma Boumerdassi,	Maître de Conférences HDR,	CNAM de Paris,	Rapporteur
Mme Sidi Mohammed Senouci,	Professeur,	Université de Bourgogne,	Rapporteur
M. Jalel Ben Othman,	Professeur,	Université de Paris 13,	Examineur
M. Eric Ollinger,	Sous-Directeur Adjoint,	Ministère des transports,	Examineur

Cette thèse est financée par le projet

Scoop@f



« La science sans conscience n'est que ruine de l'âme »

Rabelais

Résumé

Cette thèse fait un état des lieux des solutions Cloud qui visent à améliorer certains aspects des réseaux véhiculaires comme la latence et l'utilisation de la bande passante. L'état de l'art présente les principaux sujets qui nous concernent, à savoir, le Vehicular Cloud Computing (VCC), l'Information Centric Networking (ICN), les villes intelligentes ainsi que les standards ETSI (European Telecommunications Standards Institute) pour les ITS (Intelligent Transport System).

Les véhicules intelligents ouvrent de nouveaux horizons pour les applications de sécurité routière ou l'optimisation de trafic. Le canal de communication G5 (Wifi 802.11p), utilisé par les ITS qui sont actuellement déployées en Europe et aux États-Unis notamment, ne permet pas toujours d'assurer la connectivité continue d'un véhicule au réseau. A ceci s'ajoute une bande passante limitée. Ces deux phénomènes ont pour conséquence d'allonger les délais de communication d'une station ITS à une autre.

A travers cette thèse, nous apportons des solutions qui bénéficient du large déploiement du réseau cellulaire. Ces solutions, de type SaaS ou NaaS, collectent tout ou une partie des messages générés par les véhicules, pour les redistribuer, ou en déduire des événements routiers. Elles ont été expérimentées sur route à l'aide d'applications embarquées. Ceci a permis de mesurer les délais moyens de communication sur le réseau cellulaire, et le temps nécessaire à la distribution d'un événement. Afin de comparer ces résultats avec les VANET, nous présentons un simulateur qui reproduit le comportement des véhicules sur route et les communications G5 en implémentant la pile ETSI des ITS.

Nous avons abordé la ville intelligente que nous modélisons sous la forme d'un Cloud de contexte, composé d'un serveur, de multiples capteurs et des véhicules. Et nous présentons notre architecture ICN qui consiste, entre autres, à intégrer les échanges dans la couche « facilities » de l'ETSI après avoir condensé un ensemble de données en une seule au niveau d'un serveur central, et a pour but de limiter l'impact des échanges, des données de capteurs, sur la bande passante du G5. De plus grâce aux échanges de véhicule à véhicule, nous cherchons à étendre la zone du réseau dans laquelle l'information est distribuée. La présentation de notre approche passe par une modélisation analytique prenant en compte la mobilité et le placement aléatoire des UBR dans une ville de type Manhattan. Cette modélisation permet de mesurer la charge réseau engendrée par les échanges et du pourcentage de véhicules qui reçoit l'information. Nous comparons notre approche avec deux autres : *Named Data Network (NDN)* et *Rendez-Vous Network (RENE)*, après avoir donné les modèles analytiques de celles-ci. Pour valider ces modèles, nous présentons un simulateur qui reproduit la ville, la mobilité des véhicules et les communications liées à chaque approche. Enfin, nous comparons les performances de ces approches dans la recherche du plus court chemin dans une ville intelligente.

Mots-clés : VANETs ; systèmes de transport intelligents ; réseau orienté donnée ; ICN ; Cloud ; simulation ; modèle analytique ; ITS-G5 ; SaaS ; NaaS ; simulateur ITS ; performances ITS.

Abstract

This thesis deals with Cloud solutions which aim to enhance some aspects of the vehicular networks such as the latency and the use of the (bandwidth). The state of the art presents the main subjects that concern us, namely the vehicular Cloud Computing (VCC), the Information Centric Networking (ICN), the smart cities and at last, the ETSI (European Telecommunications Standards Institute) standards for the ITS (Intelligent Transport System).

The intelligent vehicles increase the scope of the road safety apps and the traffic optimization. The communication channel G5 (Wi-Fi 802.11p), used by the ITS currently deployed in Europe and in the USA, doesn't always enable to secure the non-stop connectedness of a vehicle to the network. Moreover, a limited bandwidth creates a growth of crashes. The consequence of these 2 phenomena is to extend the communication period from a ITS station to another.

This thesis offers solutions which work thanks to the wide deployment of the cellular network. These solutions have been tested on the road, with the help of the Android app. Through this procedure, we can now demonstrate all the benefits offered by these solutions to improve connectedness as well as the time of the receipt of what can happen on the road. In addition, we support the problems of connectedness of the G5 using a simulator that we have designed; this device implements, in a very precise way, the ETSI stack of the ITS. Thus, we can now analyse the strengths and the weaknesses of the different approaches as well as those from the scientific publications. For instance, within the framework of a smart city that we have modelled into a context Cloud, composed of a server, many sensors and vehicles, we partly solved the problem of the bandwidth linked to the sending of the data collected by the sensors towards the vehicles.

One of our contributions is an ICN architecture which uses the ETSI facilities among other things. It both enhances the quality of the service (which is providing the data to the largest number of vehicles) and the network overhead applied on the G5 channel, compared with the existing approaches. We have created an analytic model for each of these ones, taking into account the mobility of the vehicles into a city and the random positions of the UBR. In order to verify and approve the results, we have developed the simulator to take into account each aspects of the model and the communication of the different approaches.

Keywords: VANETs; Intelligent Transport Systems; Information Centric Networking; Cloud; Simulation; Analytic model; ITS-G5; SaaS; NaaS; ITS Simulator; ITS performances.

Remerciements

Je voudrais adresser des remerciements tout particuliers à Hacène Fouchal qui, après m'avoir supporté en tant qu'élève, m'a offert un poste d'ingénieur passionnant et m'a permis d'entreprendre cette thèse. Je voudrais également lui exprimer ma gratitude pour ses conseils avisés et pour m'avoir guidé, avec expérience, en tant que directeur de thèse, au cours de ce travail. Enfin, je voudrais le remercier pour toutes les qualités humaines dont il fait preuve.

Je voudrais aussi exprimer toute ma reconnaissance à mon co-encadrant de thèse, Marwane Ayaida, pour sa disponibilité et pour sa fiabilité. D'autre part, je voudrais le remercier d'avoir participé à faire de chaque journée de travail un aussi agréable.

Je voudrais également remercier le Pr. Sidi Mohammed Senouci, Dr. Selma Bourmedassi et Pr. Paola Flocchini pour avoir accepté d'être mes rapporteurs de thèse. De plus, je remercie Pr. Jalel Ben Otham, Mr Eric Ollinger et Mr Charles Tatkeu pour leur participation en tant qu'examineurs de ma thèse.

Je ne saurais omettre de mentionner Emilien Bourdy, avec qui j'ai le plaisir de partager le café depuis 4 ans.

Je voudrais aussi exprimer ma gratitude à Dimitri Marcheras, pour avoir tenté de m'aider... en vain.

Je souhaiterais également remercier les membres du laboratoire que j'ai côtoyé, presque quotidiennement, pendant ces années : Ida pour sa simplicité, sa bonté, sa bienveillance, sa gentillesse et son amabilité et Brice et Kevin pour leur simplicité. Je remercie également Bernard Riera, directeur du laboratoire, pour m'y avoir accueilli.

Je voudrais aussi adresser l'euphémisme de mes remerciements à Sybille Lequertier pour le soutien, les encouragements et l'amour qu'elle m'a apporté au quotidien.

Je voudrais également exprimer toute ma gratitude à ma mère, Nicole, qui m'a mis sur la voie et à mon père, Reynald, pour m'avoir attendu au tournant. Je voudrais enfin remercier mon frère et mes sœurs pour leur amour.

Geoffrey WILHELM

Sommaire

SOMMAIRE	6
LISTE DES FIGURES	8
LISTE DES TABLEAUX	11
LISTE DES ALGORITHMES.....	12
INTRODUCTION.....	13
CONTEXTE	13
MOTIVATIONS ET DEFIS	13
CONTRIBUTIONS	13
CHAPITRE 1 : ETAT DE L'ART	16
I – RESEAUX ORIENTES SUR LES DONNEES (ICN).....	17
II – VILLES INTELLIGENTES.....	26
III – VEHICULAR CLOUD COMPUTING	30
IV – STANDARDS ETSI.....	33
V – DISCUSSIONS.....	36
CHAPITRE 2 : CONTRIBUTIONS A LA MESURE ET A L'AMELIORATION DE LA CONNECTIVITE DES ITS	37
I – CONTRIBUTIONS A L'OBSERVATION ET LA MESURE DES ITS.....	39
1. <i>Indicateurs de performance</i>	39
2. <i>Les mesures</i>	40
3. <i>Les limites des standards ETSI et l'ITS-G5</i>	43
4. <i>Proposition d'un simulateur pour autoroute</i>	48
II – CONTRIBUTIONS A BASE DE CLOUD	51
1. <i>Solution Cloud SaaS</i>	51
2. <i>Solution Cloud NaaS</i>	58
III – DISCUSSIONS.....	64
3.1 <i>SaaS</i>	64
3.2 <i>NaaS</i>	64
3.3 <i>Global</i>	65
CHAPITRE 3 – PROPOSITION D'UNE APPROCHE ICN DANS UNE VILLE INTELLIGENTE.....	67
I – DEFINITIONS.....	68
II – MOTIVATIONS ET PRESENTATION DE NOTRE VILLE INTELLIGENTE	70
III – PRESENTATION D'APPROCHES ICN EXISTANTES	74
1. <i>NDN (pull)</i>	74
2. <i>RENE (push)</i>	76
IV – LE MODELE PROPOSE: GEOVEHICULAR CENTRAL DATA NETWORKING (GEOVCDN)	77
1. <i>Métadonnée avec version</i>	77
2. <i>GeoNetworking</i>	79
3. <i>Application</i>	80

V – MODELISATION	80
1. Définitions communes	83
2. Définitions spécifiques à l'approche NDN.....	85
3. Définitions spécifique à l'approche RENE	86
4. Définitions spécifiques à nos approches VCDN & VCDNoCAM	87
5. Définitions spécifiques à notre approche GeoVCDN.....	91
VI – DISCUSSIONS.....	100
CHAPITRE 4 : EVALUATION DES PERFORMANCES ICN.....	102
I – COMPARAISON DES MODELES	103
1. Charge réseau.....	103
2. Fraîcheur.....	112
3. Résumé	117
II - SIMULATION	118
1. Charge	119
2. Fraîcheur.....	126
3. Contribution à la recherche du meilleur chemin.....	131
III – DISCUSSIONS.....	134
CHAPITRE 5 : CONCLUSION ET TRAVAUX FUTURS	136
CONCLUSION	137
TRAVAUX FUTURES.....	139
BIBLIOGRAPHIE	140

Liste des figures

Figure 1 - Station STI architecture de reference	34
Figure 2 - Architecture de notre prototype.....	39
Figure 3 – Couches ETSI.....	42
Figure 4 - Vérification de signature logiciel.....	43
Figure 5 – Signature logiciel	43
Figure 6 - Communications des ITS	44
Figure 7 - Problème de connectivité en l'absence d'UBR	45
Figure 8 - Problème de connectivite avec des ubrs	46
Figure 9 - Simulateur ITS sur autoroute (IHS).....	48
Figure 10 - Résultats de la simulation n°1.....	49
Figure 11 - Résultats de la simulation n°2.....	50
Figure 12 - Résulats de la simulation n°3	50
Figure 13 - Communication ITS avec SaaS.....	52
Figure 14 - Trajet 1 - Durée d'un échange.....	56
Figure 15 - Trajet 2 - Durée d'un échange.....	56
Figure 16 - Trajet 4 - Durée d'un échange.....	56
Figure 17 - Trajet 3 - Durée d'un échange.....	56
Figure 18 - Trajet 5 - Durée d'un échange.....	56
Figure 19 - Trajet 2 - Délai d'envoi DENM	57
Figure 20 - Trajet 1 - Délai d'envoi DENM	57
Figure 21 – Trajet 4 - Délai d'envoi DENM	57
Figure 22 - Trajet 3 - Délai d'envoi denm	57
Figure 23 - Communications NaaS	59
Figure 24 - Routine de réception d'un message sur la station centrale.....	60
Figure 25 - Routine d'envoi des denms	60
Figure 26 - Extrait open street map	61
Figure 27 - Zone de destination d'un denm	61
Figure 28 - Trajet 2 - Experimentation NaaS	63
Figure 29 - Trajet 3 - Exprimmentation NaaS	63
Figure 30 – Trajet 1 – Experimentation NaaS.....	63
Figure 31 - Echanges ICN	69
Figure 32 - Choix de chemin 1	71
Figure 33 - Choix de chemin 2	72
Figure 34 - Communication dans notre modèle de ville intelligente.....	73
Figure 35 – Echanges de message en NDN	75
Figure 36 – Publication d'un message dans une architecture rendez-vous.....	76
Figure 37 – Paquet GeoVehicular Central Data Networking.....	77
Figure 38 – Echanges de messages en GeoVehicular Central Data Networking.....	80

Figure 39 - Environnement de l'étude.....	81
Figure 40 - Exemple du pire des cas.....	91
Figure 41 - Exemple de dissémination d'un message	91
Figure 42 - illustration d'un segment	92
Figure 43 - Configuration 2.....	92
Figure 44 - Configuration 1.....	92
Figure 45 - Configuration 4.....	92
Figure 46 - Configuration 3.....	92
Figure 47 - Loi de probabilité des feux.....	93
Figure 48 - Segment à 4 routes	96
Figure 49 - Segment à 3 routes	96
Figure 50 - Segment à 5 routes	96
Figure 51 - Segment à 6 routes	96
Figure 52 - Charge réseau analytique - nombre de véhicules.....	104
Figure 53 - Charge réseau analytique - longueur d'un segment	105
Figure 54 - Charge réseau analytique - longueur d'un segment – GeoVCDN	106
Figure 55 - Charge réseau analytique - portée des ubr.....	106
Figure 56 - Charge réseau analytique - couverture.....	107
Figure 57 - Charge réseau analytique - portée des véhicules	107
Figure 58 - Charge réseau analytique - portée des véhicules - GeoVCDN	108
Figure 59 - Charge réseau analytique - nombre d'ubrs.....	109
Figure 60 - Charge réseau analytique - nombre d'ubrs.....	109
Figure 61 - Charge réseau analytique - fréquence de la donnée	110
Figure 62 - Charge réseau analytique - fréquence de la donnée - GeoVCDN & VCDNoCAM	110
Figure 63 - Charge réseau analytique - nombre de capteurs.....	111
Figure 64 - Charge réseau analytique - nombre de véhicules/Longueur d'un segment.....	112
Figure 65 - Fraicheur réseau analytique - portée des véhicules	113
Figure 66 - Fraicheur réseau analytique - nombre de véhicules.....	114
Figure 67 - Charge réseau analytique - nombre d'ubrs.....	115
Figure 68 - Fraicheur réseau analytique - portée des ubrs	116
Figure 69 - Fraicheur réseau analytique - fréquence de la donnée	117
Figure 70 - Extrait d'une simulation	118
Figure 71 - Charge réseau simulation - nombre de véhicules.....	120
Figure 72 - Charge réseau simulation - RENE % GeoVCDN	120
Figure 73 - Charge réseau simulation - nombre d'ubrs.....	122
Figure 74 - Charge réseau simulation - portée des ubrs	123
Figure 75 - Charge réseau simulation - portée des véhicules - GeoVCDN	124
Figure 76 - Charge réseau simulation - fréquence de la donnée	125
Figure 77 - Fraicheur réseau simulation - portée des ubrs	126
Figure 78 - Fraicheur réseau simulation - portée des véhicules	127

Figure 79 - Fraicheur réseau simulation - nombre d'ubrs.....	128
Figure 80 - Fraicheur réseau simulation - nombre de véhicules.....	130
Figure 81 - Fraicheur réseau simulation - fréquence de la donnée	131
Figure 82 - Configuration d'un carrefour	132
Figure 83 - Path finding avec 1 UBR aléatoire.....	133
Figure 84 - Path finding avec 8 UBRs.....	133
Figure 85 - Path finding avec 2 UBRs.....	133
Figure 86 - Path finding avec 11 UBRs.....	133

Liste des tableaux

Tableau 1 - Véhicule en émission - sur piste	40
Tableau 2 - UBR en émission - sur route	41
Tableau 3 - Véhicule en émission - sur route	41
Tableau 4 - LATENCE HSM	42
Tableau 5 - Configuration de l'ordinateur pour la simulation.....	49
Tableau 6 - Variables communes	82
Tableau 7 - Variables spécifiques à chaque approche	83
Tableau 8 - Description des variables et valeurs par défaut	103
Tableau 9 - Fraicheur réseau analytique - portée des véhicules.....	112
Tableau 10 - Fraicheur réseau analytique - nombre de véhicules	113
Tableau 11 - Fraicheur réseau analytique - nombre d'ubrs	114
Tableau 12 - Fraicheur réseau analytique - portée des ubrs	115
Tableau 13 - Fraicheur réseau analytique - fréquence de la donnée	116
Tableau 14 – Configuration de l'ordinateur pour la simulation.....	119
Tableau 15 - Charge réseau simulation - nombre de véhicules	120
Tableau 16 - Comparaison charge réseau (nombre de véhicules).....	121
Tableau 17 - Charge réseau simulation - nombre d'ubrs	121
Tableau 18 - Comparaison charge réseau (nombre d'ubrs).....	122
Tableau 19 - Charge réseau simulation - portée des ubrs	122
Tableau 20 - Comparaison charge réseau (portée des UBRs).....	123
Tableau 21 - Charge réseau simulation - portée des véhicules.....	124
Tableau 22 - Comparaison charge réseau (portée des véhicules)	124
Tableau 23 - Fraicheur réseau simulation – portée des ubrs.....	126
Tableau 24 - Comparaison fraicheur réseau (portée des UBR).....	127
Tableau 25 - Fraicheur réseau simulation - portée des véhicules.....	127
Tableau 26 - Comparaison fraicheur réseau (portée des véhicules).....	128
Tableau 27 - Fraicheur réseau simulation - nombre d'ubrs	128
Tableau 28 - Comparaison fraicheur réseau (nombre d'UBR)	129
Tableau 29 - Fraicheur réseau simulation - nombre de véhicules	129
Tableau 30 - Comparaison fraicheur réseau (nombre de véhicules)	130
Tableau 31 - Résultats des approches.....	134

Liste des algorithmes

Algorithme 1 - Client HTTP – Traitement des cams	53
Algorithme 2 - Client HTTP – Traitement des denms.....	53
Algorithme 3 - Serveur SaaS – Routine de réception d'un CAM	54
Algorithme 4 - Serveur SaaS – Routine de recherche de DENM	54
Algorithme 5 - Détection d'un événement.....	55
Algorithme 6 - Server NaaS - Reception d'un DENM.....	62

INTRODUCTION

Contexte

Le véhicule est au cœur de notre société. Nous l'utilisons en toutes circonstances, que ce soit pour partir en vacances ou pour aller travailler. Le véhicule le plus basique est de plus en plus sophistiqué et les véhicules plus haut de gamme sont bardés de technologies et de capteurs.

Les ressources produites par un véhicule sont, depuis toujours, sous exploitées. Cette tendance s'inverse, par exemple le co-voiturage cherche à optimiser l'occupation des sièges d'un véhicule. Une autre ressource en excès est l'énergie électrique produite par le moteur (on peut par exemple recharger nos téléphones portable en roulant). En plaçant un ordinateur à bord des véhicules, leur permettant de communiquer avec le réseau routier, nous permettons aux véhicules d'accomplir de nouvelles tâches sans avoir besoin d'ajouter une nouvelle source d'énergie. Cet ordinateur devient à son tour une ressource insuffisamment utilisée.

Ces dernières années ont été marquées par une montée des enjeux écologiques, et le trafic routier est un sujet majeur dans ce domaine. Nous pensons que le véhicule intelligent et certains services qui pourraient en découler, permettraient au plus gros vecteur de pollution atmosphérique de faire un pas dans le sens de l'écologie.

Motivations et défis

Dans le monde de l'informatique et des ressources sous-exploitées, une réponse est le Cloud. En effet, les méthodes de Cloud permettent aux ordinateurs de partager leurs capacités de calcul ou de stockage en veille, ou de profiter de celles des autres. Dans le domaine véhiculaire, le Cloud peut prendre cet aspect, mais peut aussi se décliner d'autres façons, comme la collaboration entre plusieurs véhicules pour produire une donnée plus complexe, ou encore le monitoring par un Cloud distant permettant d'observer et d'informer les véhicules.

Si le paradigme du Cloud est une solution, il est entravé par les technologies de communication, telles que les ondes. Intrinsèquement le Cloud requiert des échanges, pour se découvrir, partager des tâches et récupérer les résultats. Le Wifi véhiculaire est un réseau à faible bande passante et la quantité de véhicules et de capteurs pose un problème. Le frein du Wifi, ce sont les collisions, plus nous échangeons de paquets, plus les collisions sont nombreuses. Elles ont pour conséquence une augmentation du délai nécessaire pour délivrer un message (à noter que le délai est un paramètre critique dans les réseaux véhiculaires).

Contributions

Dans cette thèse, nous proposons deux solutions pour deux problèmes différents. La première vise à améliorer la connectivité du réseau véhiculaire grâce à un Cloud distant, tandis que la deuxième

se concentre sur l'optimisation, en termes de taille et de quantité, des échanges qui ont lieu sur le canal de communication G5.

Pour la première solution, nous explorons deux approches de type Cloud :

1) Soit en tant que moniteur, ce qui veut dire que le Cloud reçoit toutes les informations liées au trafic routier (de la part des véhicules ou des infrastructures), lui permettant de créer une représentation de la topologie du trafic et d'en déduire des événements. Dans notre étude, nous nous sommes concentrés sur les autoroutes, car c'est la partie du réseau routier où la connectivité sur le canal G5 est la moins fiable. Cependant, la méthode peut s'étendre à d'autres environnements et la diversité des acteurs n'est pas limitée (i.e. les vélos, et les piétons sont aussi des usagers de la route qui peuvent fournir des informations grâce à leur smartphone).

2) Soit en tant que routeur. Dans le schéma de base des ITS, les véhicules sont supposés communiquer entre eux et avec les infrastructures routières via le canal G5. Cette méthode pose différents problèmes, parmi lesquels nous pouvons citer les problèmes technologiques, liées à la portée de communication des véhicules et les problèmes logistiques liées au taux de pénétration de véhicules capables de communiquer sur ce canal et à la quantité d'infrastructures déployées capables de collecter les informations fournies par les véhicules. La transition (déploiement d'UBR et vente de véhicules connectés) va prendre un certain temps et nous proposons ainsi d'utiliser la connexion cellulaire pour accéder au Cloud de type ITS centrale qui serait capable de router un événement routier détecté par un véhicule, sur de grandes distances.

La proposition de ces solutions fait suite aux observations, relevés et mesures que nous avons eu l'occasion de faire dans le cadre de projets européens. Ces relevés, ont été faits en situation réelle, sur piste ou en laboratoire avec divers partenaires, tels que des gestionnaires d'autoroutes, des gestionnaires de routes, ou des constructeurs de véhicules. Ces différentes expérimentations nous ont permis d'apporter de nouvelles valeurs sur des indicateurs de performance (délai de communication et portée de communication) de la technologie ITS-G5 en situation réelle.

Pour comparer les deux approches de type Cloud avec le réseau G5, nous avons développé un simulateur qui reproduit le comportement des véhicules sur un tronçon d'autoroute. Ce simulateur implémente les différentes couches de la pile protocolaire de l'ETSI [1] pour les ITS et simule la couche physique d'après les relevés que nous venons d'évoquer. Cela nous a permis de produire des résultats sur les délais moyens de disséminations d'un message pour un scénario établi et de les comparer avec les résultats liés aux solutions Clouds proposées.

L'autre solution, basée sur le paradigme de l'Information Centric Networking (ICN), a pour but de réduire la quantité et la taille des messages échangés entre véhicules, sans perdre en qualité de service. Le 802.11p [2] est un protocole Wifi dont l'une des caractéristiques est l'absence de connexion. En effet, en véhiculaire la mobilité est telle que nous ne pouvons nous permettre de perdre du temps avec des connexions. Pour rester dans cette logique, il semble indispensable que les échanges de données se fassent de la même façon, et l'ICN est basé sur ce principe. Cependant, les échanges ICN sur le canal G5 auraient pour conséquence une surcharge de celui-ci. Dans le Chapitre 3, nous

proposons donc une solution adaptée aux données de petites tailles, générées par un grand nombre de capteurs et permettant aux véhicules de sonder leur environnement sans surcharger le réseau avec ce qui s'apparente à du BigData véhiculaire [3].

Pour cette proposition, nous avons dirigé notre étude sur le milieu urbain, où la connectivité G5 est plus fiable que sur autoroute. Par la fiabilité, nous exprimons les chances plus grandes pour un véhicule d'être connecté directement, ou indirectement, à une infrastructure intelligente. Cela s'explique, entre autres, par des inter-distances plus réduits. Nous choisissons de considérer la ville intelligente comme un Cloud de contexte, puis nous présentons notre solution d'échange des données en son sein.

CHAPITRE 1 : ETAT DE L'ART

Le champs d'étude sur les réseaux véhiculaires est de plus en plus vaste. Les problèmes de recherche se multiplient, et se déclinent sous différentes formes. Nous pouvons tout de même faire ressortir quelques grands axes. Parmi ceux-ci, nous trouvons le Cloud Computing, une méthode bien connue en informatique, que les chercheurs essaient d'adapter aux environnements mobiles de façon générale, et aux réseaux véhiculaires plus particulièrement. Le Cloud Computing véhiculaire réutilise des formes déjà connues, ou en introduit de nouvelles. Il peut concerner de nombreux domaines et l'un d'entre eux n'est autre que la ville intelligente. Ces villes théoriques ou à l'état embryonnaire, vont générer des quantités impressionnantes de données, liant irrémédiablement la recherche sur les villes intelligentes à celle sur les Big Data. Les véhicules font, ou vont faire partis de ces créateurs et consommateurs de données. Il convient de trouver des méthodes de communication qui collent avec les restrictions que les technologies physiques tels que l'ITS-G5 impose et le flot de données propre aux Big Data. L'Information Centric Networking (ICN) est l'une de ces méthodes qu'il convient d'explorer. L'ITS-G5 sert de base aux standards ETSI (European Telecommunications Standards Institute), standards adoptés pour le déploiement des véhicules connectés en Europe. Ceux-ci ne peuvent être ignorés et doivent être pris en compte dans la recherche. Une autre forme du Cloud Computing des véhicules, plus classique, est l'utilisation d'un Cloud à distance capable de fournir des services et que nous jugeons indispensable, au moins le temps que les véhicules connectés se propagent. Dans ce chapitre, nous proposons de faire un état de l'art des domaines que nous venons de citer, tous liés et tous indispensables pour apporter une solution complète, et pérenne, aux véhicules connectés.

I – Réseaux orientés sur les données (ICN)

Les articles suivants sont traités dans l'ordre chronologique, ce qui permet de faire apparaître les évolutions des sujets de recherche concernant l'ICN. En 2012-2014, les articles sont très généralistes et posent les bases d'un nouveau paradigme. Puis au fur et à mesure, les différents travaux cherchent à appliquer cette architecture à des domaines particuliers. Dans notre cas, les réseaux véhiculaires. Nous trouvons, dans la littérature, de nombreuses propositions concernant les méthodes de mise en cache ou de routage, mais très peu prennent en compte le lien final vers le client, à savoir le G5 et les standards ETSI. Or, ceux-ci présentent une opportunité pour améliorer, à la fois la qualité de service, et la charge réseau. C'est la pierre que nous proposerons dans le chapitre 3.

L'article [4] présente les grandes lignes de l'ICN. A cette date, ce paradigme est nouveau. Les auteurs introduisent les notions fondatrices de l'ICN et comparent quatre approches en phase de développement. Certaines ont été abandonnées mais d'autres sont encore d'actualité dans la recherche. Dans un premier temps, sont présentées, les caractéristiques de l'ICN.

La première caractéristique est le nommage des objets (named data objects – NDO). Le nommage est la pierre angulaire des échanges en ICN, il permet d'identifier une donnée (de la même manière que nous identifions un serveur par son IP), de router un intérêt et potentiellement d'assurer l'intégrité de la donnée. Chaque approche diffère dans sa façon d'utiliser le nommage et également dans la façon de nommer l'objet. Deux méthodes de nommage se détachent du lot : l'une est hiérarchique, à la manière des URL et l'autre est dite « flat », moins instinctive, elle peut contenir dans son nom les données nécessaires pour s'assurer de l'authenticité et de l'intégrité du message.

La deuxième caractéristique est le modèle d'échange des données. Les échanges se font selon un modèle requête/réponse, qui en ICN, se dénote sous la forme d'**intérêt/donnée**. Le client/consumer soumet au réseau le nom de la donnée qui l'intéresse puis reçoit la donnée en retour. La troisième caractéristique, est le routage des données et des intérêts dans le réseau. Il a deux phases de routage, celle qui consiste à router l'intérêt et celle du retour de la donnée vers le client. Pour le routage des intérêts, nous pouvons soit nous baser sur une résolution de nom, de la même manière que nous le faisons actuellement sur Internet avec les serveurs DNS, soit le routage se fait grâce au nom de l'objet. La donnée emprunte ensuite le chemin retour pour arriver jusqu'au client.

La quatrième caractéristique est la mise en cache. La mise en cache se fait pendant le processus de retour de la donnée. Les équipements qui en sont capables stockent la donnée, ce qui permet de la fournir plus rapidement par la suite, plutôt que de router à nouveau la requête jusqu'à la source. L'article présente ensuite les avantages de l'approche ICN. Dans un premier temps, le caching permet de réduire la quantité d'informations échangées sur le réseau, puisqu'une partie des requêtes n'a plus besoin d'aller jusqu'à la source. Ensuite, le nommage permet de stocker une donnée sur plusieurs serveurs en évitant que chacune des copies ne soit considérée comme des objets différents. D'autre part, pour les aspects de sécurité autre que le chiffage, il n'est plus nécessaire de faire confiance au serveur, puisque la donnée assure son intégrité et son authenticité par elle-même. L'article présente et compare enfin les quatre principales approches contemporaines à l'article. La première approche DONA (Data Oriented Network Architecture) consiste à router un paquet en fonction de son nom et faire revenir la donnée sur le même chemin, ou un chemin plus approprié. La deuxième est le CCN (Content Centric Networking) route un intérêt de façon hiérarchique et envoie la donnée sur le chemin inverse. La troisième est le PSIRP (Publish-Subscribe Internet Routing Paradigm), qui se base sur un système de rendez-vous [5] : quand un nœud publie une donnée, il consulte le serveur de rendez-vous pour savoir qui est abonné et lui fait parvenir sa donnée. Enfin, la quatrième approche est le NetInf [6](Network of Information), qui se base sur un système de résolution de nom à la manière du DNS.

Cet article [4] présente les grandes lignes de l'approche ICN, son fonctionnement et ses qualités. Mais tout n'est pas traité. Notamment la façon dont les clients sont liés au réseau, comment ils sont identifiés et leur lien de communication avec le réseau. Le système actuel, client/serveur a fait ses preuves pour la navigation web. L'ICN semble adapté à la distribution de petits paquets qui ont individuellement du sens. Pour les objets plus complexes (composés de plusieurs petits objets) ce n'est pas forcément le cas, un sujet qui aurait mérité d'être abordé dans l'article.

Les travaux de recherche dans [7] présentent en détail une approche ICN particulière, appelée DONA, pour Data Oriented Network Architecture.

Dans un premier temps, l'article pose les motivations, qui sont, principalement, la persistance, la disponibilité et l'authentification des données. Dans cette approche, le nommage permet d'assurer ces trois caractéristiques.

Le nom est composé du hash de la clé publique du publisher, de son identifiant et du hash de la donnée, et qui permet à l'utilisateur de s'assurer de l'intégrité et de l'authenticité de la donnée. Ce nommage qui n'est pas interprétable à l'œil nu, comme peut l'être le nom d'un site web, implique l'intervention de système externe de confiance pour trouver le nom d'une donnée.

Ensuite, le nommage permet de router l'intérêt. DONA introduit un nouvel acteur, appelé Resolution Handler qui va être capable de trouver et d'enregistrer la source d'une donnée grâce à son nom. Une fois la donnée trouvée, son acheminement vers le client peut se faire, soit via les même Resolution Handler, soit via le réseau IP standard.

Enfin, les auteurs soulèvent quelques problèmes de sécurité et apportent des solutions, soit par des mécanismes déjà existants dans le réseau IP, soit au niveau des Resolution Handlers. Pour finir, l'article présente une implémentation de la méthode ICN et discute la faisabilité de celle-ci. En se basant sur le flux connu des requêtes HTTP sur les routeurs actuels, ils estiment la quantité de mémoire et de calcul nécessaire à l'application de leur approche, qui apparait de l'ordre du réalisable.

L'un des aspects séduisants de la méthode est qu'elle ne cherche pas à remplacer le réseau déjà en place, mais à s'y greffer pour améliorer la qualité de service pour un certain type de donnée ou toutes les données. La méthode de nommage permet de décider de la granularité de nommage d'un objet : page web entière ou chaque élément de la page web séparé. Si la réalisation est théoriquement possible, il reste à démontrer la réalisation pratique. D'autre part, une comparaison avec les méthodes actuelles en termes de charge du réseau ou de qualité de service est nécessaire et n'apparaît pas dans cet article. Enfin, la nature du lien final vers le client n'est pas considérée.

L'article [8] présente le Named Data Networking qui regroupe un ensemble d'approches ICN. Dans celui-ci, le paquet ICN est représenté comme une alternative au paquet IP standard, bien que le paquet ICN s'appuie sur la couche IP.

Dans un premier temps l'article introduit des grands principes, tels que les deux paquets ICN : l'intérêt et la donnée. La table des intérêts en cours, qui se trouve au niveau des nœuds de routage, leur permettant de garder une trace des intérêts en cours (Pending interest table – PIT). Le stock de contenu (Content store – CS), où les données sont mises en cache pour fournir la donnée plus rapidement et ne pas répéter plusieurs fois des requêtes. De plus, il existe une table où sont stockées les informations nécessaires à router le retour de la donnée (Forwarding Information Base – FIB). Dans cette approche, le nommage est hiérarchique, compréhensible par l'humain, il assure l'identification unique et la localisation d'une donnée.

La sécurité s'effectue dans le paquet lui-même, le fournisseur de la donnée, fournit sa clef et un hash de la donnée permettant d'authentifier et de vérifier l'intégrité de celle-ci. Pour le chiffrement, le client fournit sa clef publique dans son paquet d'intérêt.

Le routage se fait grâce aux différentes tables évoquées précédemment et au nommage de la donnée. Enfin, cette étude explore les différents domaines d'application, tels que les flux vidéo, les ensembles de capteurs, la distribution de fichiers ou les réseaux véhiculaires.

Cet article n'apporte rien de nouveau à ce qui était déjà fait les années d'avant au niveau de l'architecture, mais précise un peu mieux les domaines d'application. L'étude reste, comme les précédentes, de haut niveau et ne fournit aucune information permettant d'affirmer l'efficacité de cette méthode ou de la comparer avec celles déjà en place. L'utilisation de l'approche pour les réseaux véhiculaires est évoquée, sans prendre en compte que ceux-ci ne communiquent déjà plus en IP.

Le travail de recherche présenté dans [9] propose une forme d'ICN adaptée aux réseaux véhiculaires. Dans un premier temps, il rappelle les caractéristiques du NDN. Puis, il identifie 4 rôles que peut prendre un véhicule dans les réseaux véhiculaires : consumer, producer, forwarder ou mule. Par la suite, il adapte le NDN pour prendre en compte le lien de communication Wifi, notamment en stockant toutes les données reçues pour pouvoir jouer le rôle de mule.

Ensuite, il propose un algorithme de forwarding pour lutter contre les collisions qui existent dans les réseaux véhiculaires, basé sur la position, le temps et une limite de sauts.

Enfin, les auteurs proposent une démonstration grâce à un prototype qu'ils ont développé pour faire de l'info trafic, et exposent leurs résultats en termes de latence, de pourcentage de succès (obtention de la donnée après l'envoi d'un intérêt) et de quantité de messages échangés.

L'idée des auteurs est d'utiliser les caractéristiques de broadcast du réseau Wifi dont se servent les réseaux véhiculaires. L'adaptation est concluante et les résultats sont bons. Cependant, les paquets échangés se doivent d'être encapsulés dans une couche sécurité et s'ajoutent à d'autres messages déjà échangés sur le canal, si les véhicules ont plusieurs intérêts, la bande passante sera vite saturée.

Dans le survey [10], les auteurs apportent des informations supplémentaires sur les aspects fondamentaux de l'architecture ICN, déjà évoqués dans l'article [4].

Le premier d'entre eux est le nommage des données. L'utilisateur étant plus intéressé par la donnée, que par sa provenance, le réseau ICN permet d'obtenir la copie de la donnée la plus proche sans avoir à spécifier l'endroit où elle se trouve, contrairement aux réseaux orientés sur la connaissance de serveur hôte.

Le deuxième point est la livraison de l'information. L'architecture actuelle est pensée pour délivrer des informations d'un point A à un point B. L'augmentation des quantités de données qui circulent, implique la mise en place de mécanismes au niveau applicatif. Ces derniers mettent en lumière la nécessité d'améliorer la mise en cache au niveau réseau, ce à quoi, l'ICN s'emploie.

Le troisième point est la robustesse face à la mobilité. Le réseau actuel a été pensé pour satisfaire des communications entre des points fixes. L'arrivée des réseaux sans fil et mobiles, entraîne, dans le même temps, le changement de nature des connexions, qui deviennent éphémères, voire opportunistes. L'ICN est adapté à ce type de connexion grâce au cache réseau.

Le quatrième et dernier point est la sécurité. Alors que nativement l'Internet n'était pas pensé pour

contrer des utilisateurs malicieux, il a fallu mettre en place des systèmes pour contrer ceux-ci. Bien souvent les mécanismes de sécurité existent au niveau des serveurs et ne pénètrent pas profondément dans le réseau. Il existe donc en permanence une quantité de données malicieuses qui circulent dans le réseau. En appliquant les principes de sécurité sur la donnée elle-même, l'ICN apporte une solution efficace à ce problème, chaque nœud du réseau est capable d'identifier la malhonnêteté d'un paquet et d'arrêter de le traiter. Les paquets sont donc jetés à l'entrée du réseau, plutôt qu'à la sortie. Ensuite, les auteurs comparent 9 des projets en développement à cette époque, en gardant comme fil rouge ces quatre aspects.

Enfin, ils présentent une liste de verrous technologiques encore non résolus. Parmi ceux-ci, nous trouvons le nommage. La méthode de nommage a une importance primordiale pour la sécurité et l'overhead. Nous trouvons également les méthodes de routage et de cache, comment améliorer leurs performances ? Il y a aussi la problématique du transport, comment exploiter au mieux chaque lien physique. Enfin, la qualité de service : améliorer les performances sans augmenter les coûts.

Cette étude fait le point sur les avancées de l'architecture ICN, dans le but de remplacer ou de compléter les mécanismes déjà mis en place sur Internet. Il n'y est pas fait mention des autres formes de réseaux dans lesquels l'ICN trouverait sa place de façon native. Dans les problèmes ouverts, il est évoqué indirectement la possibilité de faire de l'ICN sur différentes couches, ce que nous proposons par la suite dans cette étude.

Dans l'article [11], l'ICN n'est pas présenté comme une alternative à l'Internet mais plutôt comme une architecture pour l'Internet Of Things (IOT). En effet, le modèle ICN colle parfaitement à l'avènement des objets connectés. Les objets connectés génèrent des données dans un contexte, sans que l'objet en lui-même n'ait d'importance. L'ICN est alors vu comme un outil permettant d'améliorer les délais de livraison d'une donnée et la charge réseau.

Les auteurs rappellent, ensuite, les bases de l'ICN (nommage, cache, sécurité, intérêt/donnée).

Enfin, les auteurs consacrent le reste de la discussion à expliquer le choix de l'ICN pour l'IOT. Parmi ces raisons, nous trouvons le passage à l'échelle, la qualité de service, la sécurité, l'efficacité énergétique, la mobilité et l'hétérogénéité. L'Internet des Objets pose un réel défi de passage à l'échelle auquel, l'ICN répond grâce au nommage hiérarchique.

Concernant la qualité de service, elle concerne la capacité de délivrer une information dans un délai restreint. Les objets connectés peuvent générer deux types de données, celles auxquelles nous accédons à la demande, et celles qui génèrent des avertissements. Les mécanismes de transfert de messages et de mise en cache qui existent au niveau réseau en ICN, aident dans ce sens.

La sécurité, comme cela a déjà été évoqué, ne se fait plus sur le lien de la communication, il n'est plus nécessaire d'avoir confiance en son interlocuteur, puisque chaque nœud du réseau est capable de s'assurer de l'intégrité et de l'authenticité des informations envoyées.

L'un des points abordés dans l'étude, qui ne l'a pas été avant, concerne l'économie d'énergie. En effet, en minimisant les requêtes et en faisant appel au cache plutôt qu'au transfert de message, nous économisons en opérations et donc en énergie, un sujet d'actualité.

Un autre point qui pouvait se déduire mais qui n'avait pas été formulé est l'hétérogénéité. L'ICN fournit

une nouvelle couche de communication sur laquelle tous les objets, quelle que soit leur nature, peuvent se connecter.

Cet article présente des principes intéressants sur les bienfaits et la relation étroite qu'il existe entre les deux (l'ICN et l'IOT), sans détailler la modélisation théorique ou l'implémentation pratique.

L'article [12] traite d'une branche plus précise des objets connectés : les véhicules connectés. Dans un premier temps, les auteurs rappellent les caractéristiques de base de l'ICN, pour ensuite les appliquer aux véhicules connectés. L'étude rappelle que les réseaux véhiculaires requièrent la distribution d'une grande quantité de données provenant d'acteurs de natures hétérogènes, connectés entre eux de façon intermittente.

Les applications dédiées aux réseaux véhiculaires consomment, par nature, des données limitées dans l'espace et dans le temps. L'ICN va permettre de ramener la donnée au plus proche de sa zone de pertinence.

Les réseaux véhiculaires sont déjà basés sur un lien de communication sans session, tout comme est censé fonctionner l'ICN. Cela permet de gérer la mobilité et l'intermittence de la connectivité. Dans l'article, les auteurs évoquent un consensus sur la méthode de nommage hiérarchique, mais la règle de nommage reste à définir.

Les auteurs discutent ensuite des manières de maintenir le réseau à jour, soit en inondant le réseau d'intérêts ICN, soit à l'initiative du créateur de la donnée. Chacune présentant des défauts. L'annonce récurrente de paquets de type intérêt semble, pour eux, la méthode la plus adaptée aux VANETs mais elle nécessiterait de mettre en place des mécanismes pour ne pas surcharger le réseau.

En termes de perspectives, ils évoquent deux idées importantes : la possibilité de mettre en place les échanges ICN à un autre niveau en prenant en compte les standards de communication ITS-G5 et la manière d'optimiser les échanges ICN grâce au Cloud Computing.

Cet article intègre l'ICN aux VANETs en prenant en compte les développements actuels, ce que nous n'avons retrouvé dans les articles précédents. Les auteurs évoquent des idées intéressantes qui ne sont pas mises en œuvre, telle que l'intégration des échanges ICN à un niveau plus élevé que le réseau, ce que nous proposons dans le chapitre 3.

Pour les auteurs, les réseaux ICN doivent coopérer avec le réseau Internet actuel. Et cela semble assez juste, l'ICN pour l'Internet des Objets et l'HCN (Host Centric Networking) pour accéder aux mécanismes déjà en place, tels que les sites web ou les services Clouds.

Dans l'article [13], les auteurs confirment la tendance actuelle de la recherche qui vise à appliquer l'architecture ICN au réseau ad hoc en s'éloignant de l'idée d'utiliser cette architecture pour remplacer le réseau Internet IP actuel.

L'article résume ensuite l'état actuel des recherches sur l'ICN, l'avancé des projets et l'émergence de nouveaux. Il met aussi en avant que les recherches de nommage, routage et de sécurité constituent la première étape et que l'étape actuelle consiste à trouver des applications de l'architecture ICN. C'est

à cette étape qu'apparaissent les « ICMANET », l'ICN appliqué au MANET (Mobile AdHoc Networking). L'article rappelle les défauts que présentent les communications IP dans des réseaux mobiles pour justifier l'utilisation de l'ICN. Nous y retrouvons tous les axes de recherche sur l'ICN : nommage, routage, mise en cache, transport et sécurité.

Les auteurs proposent ensuite une classification des projets en fonction des méthodes de routage. Nous y retrouvons les deux sections évoquées dans l'article [12], à savoir, le routage réactif et le routage proactif, auxquelles s'ajoute le routage opportuniste.

Le routage proactif, consiste à émettre régulièrement sa route, pour que les nœuds du réseau puissent maintenir leurs tables de routage à jour. Le routage réactif consiste à inonder le réseau d'intérêts ICN jusqu'à trouver une copie de la donnée cherchée. L'opportuniste consiste à effectuer une découverte du réseau quand cela devient nécessaire.

Enfin, dans [13] nous trouvons les axes de recherche futurs, dont le principal concerne le lien physique de communication basé sur du broadcast : comment l'utiliser au mieux, éviter les collisions et la surcharge du réseau.

Parmi les méthodes de routage réactives, nous trouvons le « GeoAware », c'est celui qui nous intéresse dans les réseaux véhiculaires, puisque les standards ITS-G5 utilisent cette méthode de routage. Cependant, dans les méthodes évoquées, c'est l'ICN qui s'occupe du routage, au lieu de se greffer aux méthodes déjà en place, ce que nous proposons dans le chapitre 3.

Dans l'article [14], les auteurs proposent leur architecture adaptée du NDN pour les villes intelligentes. Ils introduisent les champs d'application des ITS dans les villes intelligentes, tels que l'information trafic en temps réel, l'assistance aux véhicules d'urgence, le parage intelligent, les changements dynamiques des limites de vitesse ou la recharge intelligente des véhicules électriques. L'article propose une architecture en détaillant pour chaque aspect de l'ICN une solution (nommage, routage, mise en cache...).

L'architecture proposée, se présente comme une interface entre les applications et les interfaces physiques d'un système, sans prendre en compte ce qui existe déjà à ce niveau (Les standards ETSI). De plus aucune comparaison analytique ou expérimentale n'est faite pour justifier l'utilisation de leur architecture. Aussi, aucune mesure n'est proposée, concernant l'utilisation du réseau ou la qualité de service. Il mentionne, en outre, un point essentiel : la nécessité d'apporter quelques modifications aux ITS pour qu'ils puissent utiliser l'architecture ICN.

Le but de l'étude [15] est de présenter et discuter les différentes approches de mise en cache et de leur possibilité de réalisation dans un réseau véhiculaire. D'après les auteurs, il faut s'appuyer sur la pertinence spatiale ou temporelle des données. Ils présentent une liste de méthodes et apportent leur point de vue pour chacune d'elle.

La première méthode de mise en cache se base sur la mesure d'utilisation du flux de données, plus une donnée ou un type de donnée est utilisé à un endroit, ou à un moment, plus nous aurons tendance à mettre en cache les informations de ce type à l'endroit concerné. C'est une méthode qui nécessite de l'apprentissage. D'après les auteurs, une telle méthode est compliquée à mettre en place

à cause de la mobilité des nœuds. Cependant, il ne faut pas oublier l'existence d'ITS de type infrastructure.

La deuxième méthode consiste à répartir les données à certains points du réseau. Cette méthode se base sur la coordination « explicite » des nœuds. Encore une fois, cette méthode est applicable aux infrastructures de routes, mais difficilement aux véhicules.

La troisième méthode consiste à mettre en cache de façon préventive des morceaux de données. Cela implique un certain niveau de granularité concernant le nommage des données, ce qui n'est pas acquis.

La quatrième méthode est la mise en cache de prédiction, une méthode largement utilisée qui consiste à prévoir l'utilisation d'une donnée après une autre. Comme par exemple la mise en cache de la case 1 d'un tableau, dans la mémoire d'un ordinateur, si un programme accède à la case 0. Cette méthode a fait ses preuves et peut être complétée par la cinquième qu'il expose.

La cinquième consiste en effet à prédire la prochaine position du véhicule pour amener les informations au plus près de lui.

La dernière méthode implique de connaître la pertinence spatiale d'une donnée pour la placer au plus proche de cette zone, avant même son utilisation.

Enfin, les auteurs proposent de comparer leur méthode sur un scénario en ville en mesurant le taux de paquets délivrés par un serveur de proximité contre celui d'un serveur central. Le résultat qu'ils mettent en avant est l'importance de la mise en cache et l'importance de la taille du cache, sur le serveur de proximité.

Les méthodes sont exposées de manière généraliste, elles présentent toutes des avantages et des inconvénients, ce qui laisse penser que chacune d'entre elles pourrait convenir à un cas d'usage. L'article semble plutôt mettre en avant, sans le mentionner, la nécessité de pouvoir manipuler le cache réseau au niveau applicatif, pour permettre d'adopter la méthode la plus adaptée à chaque application.

Les auteurs de [16] rappellent l'un des défis de la recherche dans l'ICN, à savoir, la quantité d'intérêts ICN envoyés, qui inondent le réseau. Ce défi est d'autant plus vaste dans le domaine des VANETs et les auteurs proposent une solution dans leur article. La piste principale est d'empêcher que tous les voisins ne transfèrent inutilement des intérêts.

Leur solution se base sur l'élection d'un voisin du véhicule qui génère le paquet de type « intérêt ». Un véhicule qui prédit une longue durée de connectivité avec le véhicule source devient prioritaire pour transférer l'intérêt ICN.

Dans la suite de l'article, les auteurs proposent une implémentation de leur méthode et des simulations, pour la comparer avec une solution NDN standard, selon deux critères : le taux de réception d'une donnée par rapport au nombre d'intérêts ICN transférés et le délai de réception de la donnée. Leur méthode permet d'améliorer ces deux critères.

Leur proposition, bien qu'efficace, fait l'impasse sur certains aspects. En effet, ils minimisent le nombre de transferts de paquet de type « intérêt » pour accéder ponctuellement à une donnée, mais aucune valeur sur la bande passante qu'occupe leur méthode sur le canal G5 n'est donnée. Or, si nous voulons maintenir une donnée à jour avec la méthode NDN, nous devons générer des intérêts

ICN à faible intervalle. Tout laisse à penser que leur méthode aurait pour conséquence de surcharger le canal G5. De plus, la manière d'élire un voisin ne garantit pas que nous sélectionnions le voisin le plus proche de la donnée.

Dans l'article [17], nous retrouvons l'idée qui justifie l'usage de l'ICN dans les VANETs : le support de la mobilité par rapport au modèle IP. Les auteurs annoncent que le but de leur méthode est de réduire les collisions sur le canal Wifi afin d'améliorer la qualité de service. Ils présentent une méthode qui existe déjà, le R-NDN, qui consiste, pour un client qui émet un intérêt, à sélectionner le nœud le plus éloigné, parmi ses voisins, pour retransmettre son intérêt. Ceci est fait afin d'éviter que tout le monde ne retransmette les intérêts et que nous ayons une « tempête » d'intérêts. Les auteurs soulèvent le problème de la qualité du signal qui arrive en bord de la zone de transmission et propose de sélectionner un nœud un peu moins lointain pour garantir une meilleure qualité de signal.

Les auteurs proposent ensuite une évaluation par simulation sur NS-3. Ils comparent les résultats selon trois scénarios avec la méthode R-NDN. Le premier scénario se déroule en milieu urbain, où 20 véhicules font la requête pour une donnée. La densité varie entre haute (900 véhicules), moyenne (450 véhicules) et basse (200 véhicules). Le deuxième scénario se déroule de la même façon, mais sur autoroute cette fois-ci. Les véhicules ayant une vitesse variable entre 70 et 120km/h. Le troisième scénario sert à évaluer les effets du cache, en environnement urbain, en faisant varier cette fois-ci le nombre de clients requérant la donnée.

Ils proposent enfin des courbes permettant de comparer les deux méthodes, selon le taux de retransmission d'un paquet et le taux de satisfaction d'un intérêt.

L'idée des auteurs est de prendre un nœud un peu moins loin, pour s'assurer d'une meilleure qualité de signal. Les graphes issus des simulations et exprimant la comparaison entre l'approche R-NDN et EVNDN (celle proposée dans l'article) ne permettent pas d'extraire un réel progrès avec leur méthode. De plus, les graphes sur le temps de réponse laissent apparaître une forme de cycle dans la courbe de montée/descente, ce qui est inattendu considérant la nature des axes.

D'autre part, la stratégie consistant à sélectionner un voisin pose toujours le même problème, le voisin sélectionné est-il vraiment le meilleur et le plus proche de la donnée (par exemple, je le capte, mais lui me capte-t-il ?).

Les auteurs de l'étude [18] mettent en avant un domaine d'application particulier : les véhicules autonomes. En effet, tout laisse à penser que ces véhicules représentent notre futur sur la route. Pour fonctionner, ils ont besoin d'informations, pour construire l'horizon électronique. Cet horizon est construit grâce à toutes sortes de capteurs ou d'équipements, de nature hétérogène qui fournissent chacun une partie du puzzle. Pour fournir ces informations, l'architecture ICN est toute désignée, pour les raisons déjà évoquées dans les articles précédents. De plus, l'article aborde le fait que les véhicules connectés ont tendance à faire usage, simultanément, de plusieurs réseaux sans fil.

Chacun de ces réseaux ayant un avantage, le cellulaire pour la large bande passante, le 802.11p pour ses faibles délais etc.

Les auteurs font ensuite un état de l'art de l'ICN, rappelant les différents projets en cours de

développement, et rappellent l'historique de la recherche dans ce domaine, à savoir, dans un premier temps, la définition de caractéristiques générales : routage, nommage, transport, mise en cache réseau, puis le défi de la sécurité et enfin l'application de l'ICN aux réseaux mobiles. En troisième partie, les auteurs rappellent les grandes lignes de l'ICN. Par la suite, ils les appliquent aux réseaux véhiculaires. Ils présentent ainsi leur vision d'un écosystème d'objets connectés, communiquant grâce au paradigme ICN.

En cinquième partie, l'article développe les exigences ICN pour les véhicules connectés. La mobilité est couverte par les différents canaux de communication, le routage se doit d'être réactif, à cause notamment de la mobilité, le cache est très important, encore une fois pour supporter la mobilité et améliorer la qualité de service. Pour la sécurité, les communications véhiculaires privilégiant les échanges sans connexion, l'ICN, en sécurisant la donnée, calque parfaitement à cette manière de faire. L'interopérabilité sera un défi majeur, car il y a beaucoup d'acteurs, apportant chacun sa vision de l'ICN.

Enfin, les auteurs proposent une comparaison, sur chacun des aspects ICN, des projets en cours de développement.

L'article propose un bon résumé de ce qui se fait déjà et rappelle les enjeux et les raisons d'utiliser le paradigme ICN dans les véhicules connectés. Le survey étant orienté véhicules connectés, il est surprenant qu'il ne soit fait aucune mention des standards ETSI et du routage GeoNetworking, qui sont les technologies actuellement utilisées en Europe pour les véhicules connectés.

Pour conclure, le routage, le nommage, la mise en cache et la sécurité sont autant de sujets redondants qui sont abordés dans les travaux de recherche. Des entreprises proposent leur approche de l'architecture ICN en faisant des choix sur ces 4 sujets.

L'ICN est vu par certains, comme un remplaçant de l'architecture Internet par IP, par d'autres comme un complément pour des domaines plus précis, tel que l'Internet des Objets. En ce qui concerne l'Internet des Véhicules, aucune des propositions faites pour le moment ne prend en compte les standards ETSI. Actuellement utilisés pour les véhicules connectés, les standards ETSI et la nature du lien final du véhicule (broadcast) ne peuvent être ignorés [12] dans l'architecture ICN proposée. Puisque le canal de communication G5 et sa relative faible bande passante apporte une contrainte supplémentaire.

II – Villes intelligentes

Le terme désigne une ville équipée d'objets connectés. Les villes intelligentes sont plutôt des villes connectées, capables de percevoir leur environnement grâce aux nombreux capteurs qui y sont déployés. Les termes Internet des Objets et Big Data, sont des termes qui ressortent souvent dans les travaux de recherche sur les villes intelligentes. Le Cloud Computing est bien souvent la réponse pour les gérer. La section suivante fait le résumé et la critique de différents articles concernant les villes intelligentes, en essayant d'apporter brièvement de nouvelles pistes.

Dans les travaux de recherche [19], les auteurs présentent comment le Cloud Computing, l'Internet des Objets et les réseaux de capteurs se complètent les uns les autres. Le Cloud Computing permet d'exploiter les ressources de manière optimale pour des opérations à la demande. Les réseaux de capteurs et les objets connectés, sont une source de données considérable qui doivent être traitées et le Cloud Computing peut y aider. Ce dernier, a besoin de ressources, que des objets connectés sont capable de partager quand ils ne les utilisent pas. Ils font aussi le parallèle avec les Big Data, pour les capteurs en particulier, qui collent parfaitement aux trois V (volume, variété, vélocité). Puis, ils présentent ce que sont des capteurs, ce qu'ils produisent et comment nous les gérons, via une interface web par exemple. Ils présentent également des domaines d'applications, par exemple dans des services d'environnement ou de santé.

Ensuite, les auteurs introduisent et présentent le concept de Sensing as a Service (SnaaS) et de Sensor Event as a Service (SEaaS), également abordé dans [20], qui sont les processus permettant de passer des données générées par des réseaux de capteurs à un événement pour l'utilisateur.

L'article liste ensuite quelques défis qui persistent dans le domaine du Sensor Cloud : la calcul d'événement complexe grâce aux capteurs, le passage à l'échelle et le traitement en temps réel et la nécessité de fournir un framework qui permet d'exploiter tous ces capteurs et leurs données. Enfin, les auteurs listent quelques domaines d'application (réalité augmentée, contrôle agricoles...) et enchainent sur la nécessité d'avoir une plateforme de gestion des réseaux de capteurs, faisant le lien entre les données collectées, le calcul d'événement et la distribution des résultats

En premier lieu, les auteurs considèrent que chaque objet connecté aura une adresse IP. Ils ne prennent pas en compte les autres architectures réseaux tel que l'ICN. Ils oublient d'apporter une précision assez cruciale entre le SnaaS et le SEaaS, une différence que nous retrouvons dans notre Chapitre 2 sur l'hybridation, l'un des modèles est orienté pull (SnaaS) et l'autre push. Ce sont aussi deux modèles qui s'appliquent à n'importe quelle échelle : une ville, un pays, un tronçon de route.

L'étude [21] présente la relation nécessaire entre l'Internet des Objets et le Cloud Computing. Les auteurs rappellent également la nature hétérogène et la multitude d'acteurs qui interviennent dans l'Internet des Objets et présentent ensuite leur contribution, qui est un framework pour un Cloud de capteurs.

Ensuite, les auteurs présentent le travail déjà effectué et comment les deux technologies (Cloud et Internet des Objets) convergent. Ils rappellent cependant, que cette convergence nécessite que des adaptations soient faites et que rien n'a été proposé dans ce sens pour le moment. Ils discutent ensuite d'un des défis majeurs du domaine, comme à de nombreux domaines en informatique : la sécurité. Sont explorés plusieurs aspects : la virtualisation, le contrôle des ressources ou le contrôle de la vie privée.

Le concept que les auteurs proposent est une plateforme basée sur la virtualisation des infrastructures. Elle combine plusieurs services : Network as a Service, Plateforme as a Service et Sensing as a Service.

Les auteurs proposent une implémentation et concluent sur le fait qu'ils l'ont évaluée, mais aucun résultat n'apparaît dans l'article. D'après nous, ce qu'il faut retenir de cette étude, est la symbiose qu'il

existe entre le Cloud et les IoT, une association que nous cherchons à conserver dans les propositions faites dans cette thèse.

Dans l'article [22], les auteurs introduisent leurs travaux en expliquant que l'IoT est en plein essor. Entre 2010 et 2013, la quantité d'informations stockées a été multipliée par sept et la quantité qui circulent sur le réseau par trois ; et cette croissance est exponentielle.

Comme dans d'autres articles, ils font ensuite l'association de l'Internet des Objets avec le Big Data, toujours grâce au 3 V (vélocité, variété et volume). D'autre part, ils fournissent quelques exemples concrets, et déjà en place, de solution basée sur l'IoT (maison verte, contrôle qualité d'un pont, contrôle d'une centrale nucléaire, etc.).

Les auteurs évoquent ensuite les intérêts que procure le Big Data, économique notamment, et discutent ensuite des technologies qui permettent de traiter ces grandes quantités de données (NoSQL par exemple).

Sont ensuite énoncés, les défis autour de la gestion du BigData, telles que l'optimisation des requêtes ou la diminution de la consommation d'énergie.

L'article ressemble à un mini tutoriel du Big Data. A la fin, les auteurs évoquent et discutent l'association des BigData avec le Cloud Computing, transformant l'Internet des Objets en Sensing as a Service ou plutôt en XaaS : Quoique ce soit as a service. Nous savons que ces mêmes objets/capteurs disposent de capacités de calcul ou de stockage inutilisées, c'est pourquoi dans notre chapitre 2 nous proposons de faire du calcul entre capteurs (véhicules) si possible et de passer par des infrastructures de type Cloud sinon.

Les auteurs rappellent dans leur étude [23] que la population mondiale se concentre de plus en plus sur les espaces urbains, participant à l'augmentation du trafic des données numériques. Pour gérer celles-ci (collecte et exploitation), deux paradigmes sont associés : le Cloud Computing et l'Internet des Objets, ce qui donne le Cloud des objets (Cloud of Things – CoT). Au début de l'article, sont présentées différentes plateformes permettant de manipuler les objets. La tendance est de développer des middlewares spécialisés dans un domaine ; pour faire le pont entre un réseau d'un type d'objets et une plateforme plus globale.

Les auteurs identifient 4 couches : la couche des capteurs, la couche des publieurs (ceux qui possèdent les capteurs et qui génèrent des données complexes grâce à ceux-ci), une couche d'accès aux données complexes et la couche client (qui consomme la donnée).

Les auteurs expliquent que le CoT est la combinaison de plusieurs modèles : Software as a service pour l'application utilisateur, Platform as a Service pour permettre aux publieurs de gérer leurs réseaux de capteurs et Infrastructure as a Service pour gérer les données des capteurs. Ils expliquent que dans la littérature 3 autres aspects ont été ajoutés : Network as a Service, Policy Management as a Service et Data as a Service. Toutes ces manières d'exploiter le Cloud mène au XaaS : tout est un service et mène au « pay as you go ». Les objets ont des ressources que nous pouvons acheter pour une opération, ou plusieurs.

Les auteurs exposent des exigences dans le cadre du CoT qui concernent les services et la disponibilité. La disponibilité des infrastructures nécessaires au modèle. Les services quant à eux doivent être

orientés pour l'utilisateur, accessibles de partout, avoir un faible coût énergétique et financier et être interopérables.

D'autre part il découpe l'architecture IoT en 5 couches : Device layer, Network Layer, Middleware Layer, Application Layer et Business Layer. Et enfin, les auteurs présentent VITAL (Virtualized programmable InTerfAces for innovative cost-effective IoT deployments in smart cities) ; une architecture qui cherche à intégrer des objets connectés sur plusieurs plateformes IoT en minimisant les coûts financiers.

Il ressort souvent, dans cette étude, l'aspect Big Data et le défi dans la sécurité dans les réseaux de capteurs. L'architecture ICN offre des réponses concrètes aux aspects de sécurité, et peut être abordée comme la couche middleware apportant des solutions d'interopérabilité à moindre coût.

L'étude [24] présente une forme de réseau de capteurs appelée « participatory sensing » (PS), que nous pourrions traduire par « perception participative ». Cette forme de réseau fait intervenir l'individu et les capteurs du smartphone pour relever des paramètres environnementaux.

Les auteurs présentent une plateforme appelée TQMS (Trip Quality Measurement System), qui a pour but de mesurer la qualité du trajet d'un individu. Ils listent un ensemble de domaines d'applications aux PS : transport, relevé numérique de l'environnement, santé publique, sécurité urbaine, économie, éducation/recherche. Juste après sont introduits les défis autour du PS : sécurité (au sens informatique du terme), motivation (motivation à participer), qualité des données.

Ensuite, l'article énumère les défis du PS pour les villes intelligentes : l'interopérabilité (à cause de la diversité des vendeurs), la gestion des données (Big Data), la mise en place d'API pour faciliter le développement.

Il présente ensuite leur plateforme TQMS sur 4 points : la conception architecturale, les approches pour garantir la qualité des données, des modèles analytiques pour vérifier les résultats et une expérience pratique du PS. Une application smartphone collecte et enregistre ses vibrations, ce qui permet à la plateforme de dessiner une carte des vibrations dans la ville.

Les auteurs introduisent la perception participative comme étant basée sur les smartphones, mais n'importe quel objet connecté d'un individu peut devenir un capteur participatif et notamment un véhicule comme nous le montrons dans le chapitre 2.

L'étude fait une démonstration mathématique et pratique d'utilisation de capteurs dans une ville intelligente pour améliorer la qualité de trajet des individus. Une méthode qui pourrait s'appliquer dans l'architecture que nous proposons au Chapitre 3, les véhicules relevant les temps de trajets, un serveur central serait capable de dessiner une carte des temps de trajet en fonction des heures.

Dans l'article [25], les auteurs expliquent d'une part que les villes intelligentes et l'Internet des Véhicules peuvent améliorer la qualité et la sécurité du transport. Ils partent également du principe que si un nœud du réseau n'est pas fiable individuellement, le réseau entier est fiable grâce à la redondance des nœuds. Enfin, ils annoncent la méthode de propagation des données qu'ils proposent : combiner smartphone et véhicule, en utilisant le smartphone pour une communication longue distance.

Ils introduisent, dans leur modèle, la notion de small world. Ceux-ci sont reliés entre eux grâce aux smartphones et sont capables de construire des routes pour la communication inter-véhicules sur de longues distances.

Dans leur proposition, ils partent du principe que le téléphone permet d'assurer la communication longue distance, en réalité c'est la connexion au réseau cellulaire qui le permet. Dans les projets européens, les boîtiers embarqués dans les véhicules sont capables de communiquer sur ce canal, et le smartphone n'est plus nécessaire. Dans la pratique, le téléphone dans un véhicule ne sert que d'interface graphique.

En réalité, cette étude présente ce qui se fait déjà, en utilisant d'autres mots : small world pour le voisinage, smartphone pour connexion cellulaire, social network pour le Cloud distant, etc. Si leur modèle n'est pas faux, il n'est pas nouveau.

Pour conclure, si nous devons donner une définition de la ville intelligente, ce serait : l'organisation des capteurs d'une ville, afin de proposer des services aux usagers. La nature des services proposés est multiple, voire sans limite, c'est pourquoi nous pouvons parfois trouver le terme XaaS, qui signifie que tout peut être un service. L'un de ceux-ci est le Sensing as a Service, qui vise à exploiter les données brutes générées par les différents équipements de la ville, pour prendre connaissance de l'environnement. L'orientation majeure de la recherche vise à proposer un framework permettant de gérer un réseau de capteurs et d'accéder aux données produites. Dans le chapitre 2, nous constaterons que le Sensing as a Service n'est pas réservé aux villes intelligentes, mais peut s'appliquer à tout le réseau routier, chaque véhicule étant un « super » capteur.

III – Vehicular Cloud Computing

Le Cloud Computing Véhiculaire est un type de Cloud Computing Mobile. Il peut être vu de deux façons. La première façon est la mise à disposition de services dédiés aux véhicules, sur un Cloud distant, ce que nous proposons dans le Chapitre 2. La deuxième façon d'aborder le Cloud Computing, c'est de transformer les véhicules en fournisseurs et consommateurs de services. Ainsi, le véhicule partage ses ressources ou s'associe avec d'autre véhicules pour produire une donnée, cela représente nos travaux futurs du Chapitre 3.

L'auteur de [26] parle d'abord du mobile Cloud Computing avant de dériver sur une spécialisation de celui-ci pour les véhicules. Dans le mobile Cloud Computing, les nœuds peuvent être, à la fois, utilisateurs et fournisseurs de services. L'avènement des objets connectés et autonomes aura pour conséquence d'accroître le nombre de requêtes dédiées à la découverte du monde qui nous entoure. Pour accomplir cette tâche, les véhicules et les gens sont les meilleurs senseurs qui existent. Afin de ne pas inonder Internet de ces requêtes et données, il est nécessaire de trouver des systèmes qui permettent d'alléger la charge que nous envoyons à des Cloud distants. Dans cette optique, l'amélioration considérable des capacités de calcul et de stockage des appareils mobiles ou embarqués, permet d'envisager de faire une partie des calculs localement, et de n'envoyer sur Internet que des données avec une longue durée de vie, ou de ne faire des requêtes que pour des calculs

complexes.

Les trois défis que l’auteur pose sont : Comment accéder à Internet ? Comment former un Cloud dynamique ? Comment assurer la sécurité et la vie privée dans cette architecture ? L’auteur cite ensuite une liste d’approches qui correspond aux caractéristiques attendues : Cloud mobile véhiculaire (collaboration pour l’analyse de l’entourage), Cloud mobile personnel (limitée par la batterie), Cloud mobile orienté sur une mission (association de l’humain et du robot pour accomplir une tâche).

L’un des défis du calcul par Cloud mobile est l’agrégation des données collectées et calculées localement. L’autre est l’exploration du réseau orienté sur les données, que ce soit pour la communication ou la sécurité. La force du Cloud véhiculaire, n’est pas seulement la puissance de calcul dont il dispose, mais aussi des capteurs dont les véhicules sont équipés.

L’article met, ensuite, en avant deux cas d’usage : la surveillance urbaine et la gestion de trafic. De plus, deux points importants sont abordés dans l’article mais ne sont pas approfondis : le premier est l’orientation du réseau sur la donnée, c’est la méthode ICN. Le deuxième est l’aspect local du calcul et du stockage qui n’est autre qu’une perspective notable pour l’Edge Computing et aussi pour l’ICN, chaque véhicule pouvant jouer le rôle de routeur, permettant de garder une donnée au plus proche de sa zone de pertinence.

Dans l’étude [27], les auteurs énoncent que la part de la population en milieu urbain est de plus en plus importante et la population totale croît également. Ce fait engendre des congestions de plus en plus importantes dans les villes qui n’ont pas été pensées pour accueillir autant de véhicules.

Par conséquent, le passage aux véhicules connectés est très coûteux pour le client et l’entreprise. De plus, les auteurs mettent en avant le besoin d’un « framwekork » permettant de faciliter l’usage des ITS en tant que Cloud.

Une idée principale ressort : celle d’utiliser les ressources, sous-exploitées, des véhicules (ressources de calcul et de stockage).

Les auteurs proposent ensuite leur architecture de type ITS-Cloud. Celle-ci se décompose en 3 couches. La première couche, utilisateur final, désigne soit un utilisateur, soit un server/data-center. L’acteur effectue des requêtes et reçoit des réponses via la deuxième couche, communication. Le rôle de la couche communication est de déterminer le canal à utiliser entre deux acteurs en fonction du type de chacun. La couche communication permet d’accéder à la troisième, le Cloud. La couche Cloud désigne soit un data-center sur Internet, soit un Cloud dynamique formé de composants ITS.

Pour valider cette proposition, les auteurs proposent d’évaluer la charge qui a été absorbée par les véhicules. Le Cloud de véhicule est formé au déclenchement d’un serveur, qui sélectionne un nœud, désigné sous le terme d’abeille dans l’article. Cette première abeille sélectionne d’autres abeilles, qui, à leur tour, en sélectionnent d’autres etc.

Concernant le coût de déploiement pour les véhicules connectés, il est, pour le moment, et en grande partie, financé par la commission européenne ou des aides de collectivité. De plus nous ne savons pas encore le business modèle qui va être adopté par les fournisseurs de ces services dans les

véhicules. Il est probable que l'utilisateur final ne voit aucun changement sur sa facture. Dans leur architecture, les véhicules n'ont accès au Cloud conventionnel que par le biais d'UBR. Nous savons, et nous le montrons dans notre Chapitre 2, que la connectivité peut être très limitée si nous nous basons exclusivement sur des communications à courte distance.

Un autre aspect, qui n'est pas abordé dans l'article, est la charge de communication qu'engendre une telle méthode sur le canal G5. Les communications sans fil sur des canaux à ondes courtes ont des bandes passantes très limitées.

Les principales idées qui sont énoncées dans [28] sont les suivantes :

- 1) l'utilisation des réseaux véhiculaires pour aider à la gestion de trafic, à la sécurité routière et aux transferts d'information.
- 2) la sous-exploitation des ressources informatiques des véhicules.
- 3) « pay as you go », qui est le fait de vendre ses ressources.

Le but de cet article est de fournir une vue claire des fondamentaux du VCC et une étude comparative entre le calcul par Cloud et le calcul par Cloud véhiculaire. Pour les auteurs, le principal atout des véhicules, réside dans la quantité de capteurs qu'ils embarquent.

Dans un premier temps, les auteurs proposent une taxonomie des applications et des exigences des réseaux véhiculaires. Ensuite, en définissant les termes, ils présentent la différence entre le Cloud Computing et le Mobile Cloud Computing. Le premier a pour vocation de fournir des ressources à la demande avec une gestion minimale et dans de courts délais, tandis que le deuxième consiste, pour des équipements mobiles, à utiliser des Clouds distants pour accomplir des tâches. D'après l'article, l'architecture mobile Cloud Computing se découpe en trois couches : couche application (SaaS), couche plateforme (PaaS), couche infrastructure (IaaS). Pour les Clouds véhiculaires, ils peuvent être, soit des Clouds mobiles avec la définition donnée précédemment, soit devenir des Clouds formés dynamiquement et fournissant les services que peuvent produire des Clouds classiques sur Internet. Les auteurs présentent ensuite une taxonomie du vehicular Cloud Computing qui se décompose en 5 grandes catégories : nature, application, services, infrastructure, sécurité ; et expliquent également les services que peut fournir le VCC : Storage as a Service, Cooperation as a service, Computing as a service, Pictures on a wheel as a service, Information as a Service, Entertainment as a service. Enfin, avant de fournir une liste non exhaustive de cas d'usage pour le VCC, les auteurs présentent trois manières de former un Cloud : de façon statique (parking), grâce à une infrastructure chef d'orchestre, ou de façon dynamique (à la demande, avec des véhicules en mouvement).

Cette étude fournit une définition claire des taxonomies, et précise les différents aspects du calcul par Cloud véhiculaire. Cependant, si les aspects généraux sont traités, aucun détail technique n'est abordé sur les différentes possibilités de résoudre un problème.

Concernant l'aspect sécurité, ils s'interrogent sur la manière de l'implémenter et proposent une taxonomie de la sécurité et de la vie privée. Il existe déjà des solutions, par exemple dans le projet SCOOP@F la sécurité est basée sur un système de PKI [29] (Public Key Infrastructure). Le défi est de réduire la place, en termes de taille, qu'occupe la sécurité dans les échanges.

Dans l'article [30], les auteurs proposent une architecture combinant Cloud distant et Cloud véhiculaire qui s'associent pour former un seul Cloud. Dans cette architecture la nature du Cloud (statique, dynamique ou d'infrastructure) ne compte plus, puisque qu'elle est basée sur la virtualisation pour palier à l'hétérogénéité des ressources des véhicules. Ce travail de recherche vise à proposer une solution d'allocation des ressources.

L'article propose, ensuite, un modèle analytique de la solution et formule le problème selon un processus de markov. Les auteurs exposent enfin les courbes issues des résultats numériques de leur modèle et affichent un gain de 7% par rapport aux deux autres modèles qu'ils présentent, nommés GA scheme et SA scheme, et qui sont des modèles heuristiques.

L'article [31] rappelle que le VCC a pour but de distribuer des tâches pour s'alléger, mais aussi de partager de l'information. Les auteurs introduisent deux notions pour mieux illustrer leur méthode. La première est la notion de contexte. Le contexte est la vision logique d'un ou plusieurs éléments physiques, un parking ou un centre commercial par exemple. L'écosystème est la seconde notion présentée qui se définit comme un ensemble de contextes, liés entre eux par un modèle (modèle de mobilité par exemple). Le but de l'écosystème est de rapprocher de l'utilisateur les données d'un autre contexte que celui dans lequel il se trouve, et qu'il est susceptible de vouloir utiliser. Ils proposent ensuite une méthode de communication à l'intérieur de cet écosystème.

Dans l'article, nous pouvons ainsi lire que l'échange des messages n'est pas possible en multi-sauts à cause de la nature du 802.11p qui émet les messages en broadcast. Ceci est inexact. En effet, les standards ETSI et la couche mac de l'ITS-G5 permet de faire des communications unicast, même si le message est envoyé physiquement à tout le monde. La notion d'écosystème est intéressante, surtout dans le domaine des véhicules, où nous pouvons mettre en place un apprentissage pour prévoir les déplacements d'un utilisateur. Elle permet aussi de découper les zones géographiques en écosystèmes hiérarchiques, l'écosystème de la France serait l'ensemble des écosystèmes qui le compose (les régions par exemple) et les régions sont composées de départements et les départements de villes. Cette vision hiérarchique rejoint la méthode de nommage du même nom, explorée pour l'ICN. Dans le Chapitre 3, nous proposons une méthode de communication dans une architecture qui s'apparente à la définition de l'écosystème présentée dans cet article.

Pour conclure, sur les travaux de recherche concernant le Cloud Computing, nous pouvons dire qu'ils sont principalement orientés sur la théorie ; il existe peu de propositions d'implémentation. Les techniques de communication au sein d'un Cloud ne sont même pas abordées, et les technologies déjà en place (standards ETSI) ne sont que très rarement mentionnées. Le calcul en Cloud dans les réseaux de véhicules semble être une pierre indispensable pour agréger les données et exploiter les ressources inutilisées des véhicules. Au niveau Internet, ils permettent collecter, stocker et fournir des données dont la pertinence n'est pas seulement locale.

IV – Standards ETSI

L'ETSI a défini les différentes couches utilisées par les ITS (Intelligent Transport System) pour communiquer entre elles [32]. Nous trouvons deux sous couches dans la couche accès : la couche physique, basée sur le canal Wifi 802.11p et que nous appelons G5, et la couche MAC [33] [34] [35] [36] [37]. La couche Networking & Transport est composée du protocole GeoNetworking pour le routage et du BTP (Basic Transport Protocol) pour le transport [38]. Ensuite, il y a la couche « facilities », qui spécifie les messages échangés sur le canal, de types : CAM (Cooperative Awareness Message) [39], DENM (Decentralized Environmental Notification Message) [40], SPAT (Signal Phase and Time), MAP (Map Data).... Et enfin la couche application. En couche verticale, nous trouvons également la sécurité [41] et la couche management. Cette section a pour but d'introduire ces couches et de présenter les éléments nécessaires à la compréhension du reste du document.

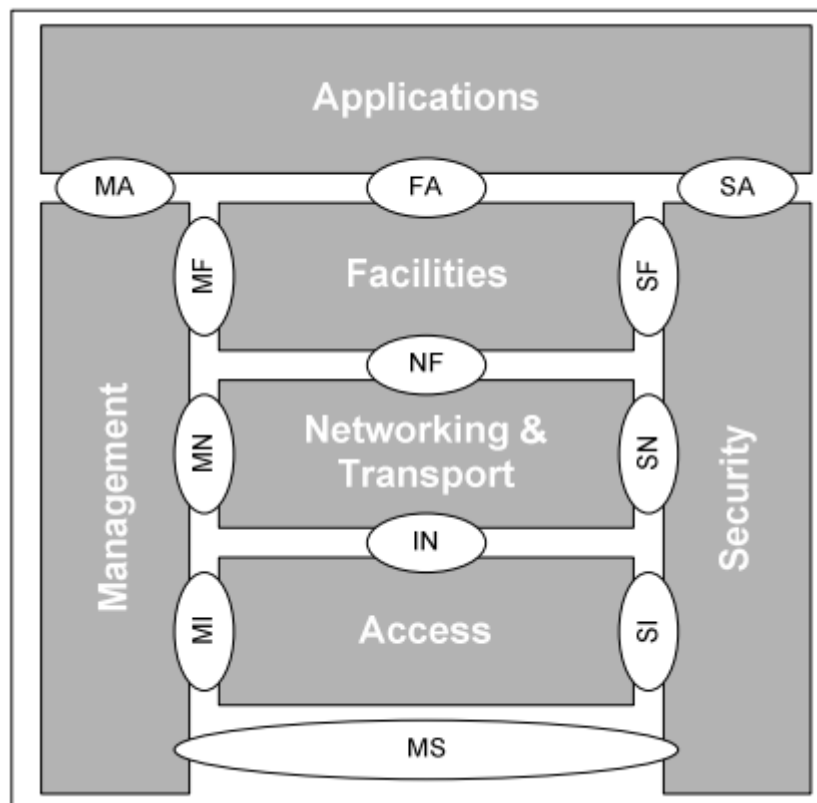


FIGURE 1 - STATION STI ARCHITECTURE DE REFERENCE

La Figure 1, extraite du document [37] présente l'architecture des couches ETSI expliquées au-dessus. Les bulles MA, FA, SA, etc. désigne les protocoles ou les points d'accès aux services associés.

Le canal G5 est basé sur le standard IEEE 802.11p, une forme de Wifi (IEEE 802.11) dédiée aux communications à courte distance et servant de support aux systèmes de transport intelligents. Nous retrouverons ces communications sous la forme de V2V (Vehicle to Vehicle), V2I (Vehicle to Infrastructure), I2V (Infrastructure to Vehicle), pour désigner quel type d'acteur communique avec quel type d'acteur (véhicule, infrastructure, etc.). L'ITS-G5 désigne la couche accès des standards ETSI, composé de la couche Physique 802.11p et de la couche MAC standard. La couche MAC (Media Access Control) est un élément crucial pour la couche network

(GeoNetworking), puisqu'elle permet de faire la différence entre broadcast et unicast. La couche physique, par nature, envoie les informations en broadcast. Or, il est important pour un véhicule de savoir le plus tôt possible si le message capté lui est destiné ou non.

Le GeoNetworking, associé au BTP, permet de router la donnée et de la transporter. Le BTP renseigne un numéro de port et un numéro de sous-port qui permettent de désigner la nature du message contenu dans le paquet. La couche GeoNetworking, quant à elle, contient les informations nécessaires au routage du paquet : sa provenance, son nombre de sauts, sa destination, etc. Il existe trois formes de routage : Simple, CBF (Contention Based Forwarding) et Advanced. Le simple consiste, pour un véhicule, à réémettre en broadcast un message reçu, s'il se trouve dans la zone de destination ou, s'il ne s'y trouve pas, l'émettre à son voisin le plus proche de la zone de destination. Avec le CBF, quand un véhicule reçoit un message, il le met dans un buffer et attend pendant un certain temps. S'il reçoit de nouveau le message, il le supprime de son buffer. L'Advanced rajoute un contrôle supplémentaire sur l'angle entre l'émetteur et le récepteur.

La sécurité est traitée à ce niveau, ainsi dans la couche GeoNetworking nous trouvons une en-tête contenant les informations nécessaires à vérifier l'intégrité et l'authenticité du message. Les messages sont signés par un certificat, obtenu par un système de PKI. Ce certificat contient, entre autres, la clef publique de l'émetteur et le hash du certificat de l'autorité de confiance, accompagnée de la signature du message.

Les principaux messages échangés sont les CAM (Cooperative Awareness Message) et les DENM (Decentralized Event Notification Message). Les CAM sont une sorte de super beacons, émis par les véhicules et indiquant son état à l'instant T : vitesse, position, direction, historique de déplacement, etc. La plupart de ces informations proviennent du bus CAN. Ce message est émis à une fréquence variant de 1 à 10Hz (10Hz dans la majorité des cas). Sa définition possède une partie extensible permettant d'inclure du contenu « propriétaire » dans le CAM, ne changeant pas la nature même du message, mais ajoutant des informations supplémentaires dont il faut connaître la définition pour les comprendre. Ils sont émis en saut unique, c'est-à-dire seulement à ses voisins. Quant au DENM, il est généré ponctuellement pour signaler un événement détecté sur la route, par le conducteur ou de façon automatique par les capteurs du véhicule. Les DENM ont une zone de pertinence, qui est déterminée au cas par cas. Les véhicules se passent le message en unicast jusqu'à atteindre un véhicule qui se trouve dans la zone de destination et qui va diffuser le message à tout le monde autour de lui (ceci est assuré par le GeoNetworking)

Pour résumer les standards ETSI, nous pouvons dire qu'ils mettent en place un ensemble de couches assez classiques (accès, réseau, application). Ils sont principalement orientés sur la sécurité et l'information routière. Le GeoNetworking est un protocole adaptatif avec ses 3 déclinaisons (Simple, CBF, Advanced) et ses nombreux paramètres. L'architecture du CAM, avec sa partie extensible, peut être remaniée pour créer de nouvelles fonctionnalités, ce que nous proposons au chapitre 3.

V – Discussions

Dans cette section, nous avons présenté différents paradigmes, modèles ou standards et leurs avancements dans la recherche. Les ITS en Europe sont actuellement déployés en se basant sur les standards ETSI, c'est pourquoi nos propositions les intègrent, soit en essayant de les compléter, soit en les exploitant pour communiquer des informations non natives. Nous avons aussi présenté le concept de villes intelligentes qui pose des problèmes par rapport aux volumes des données qui vont être générées. Le Cloud Computing fournit plusieurs approches pour tenter d'alléger le transit de données liées aux villes intelligentes. L'ICN de son côté apporte une architecture de communication qui peut à la fois intervenir dans le Cloud Computing et dans les villes intelligentes. Dans le chapitre suivant, nous présentons un modèle de communication hybride des ITS, avec un Cloud distant, permettant de compléter les lacunes liées à l'ITS-G5.

CHAPITRE 2 : Contributions à la mesure et à l'amélioration de la connectivité des ITS

La communication inter-véhicules, peut être dédiée à de nombreuses applications : confort, sécurité routière, gestion du trafic, information trafic, etc. Pour le domaine de la sécurité routière, la plus critique ici, le délai recommandé est de 300 ms point-à-point [42] [43].

Nos observations, concernant les communications sur le canal G5, au cours des dernières années, dans le cadre des projets européens sur le déploiement des ITS (SCOOP@F, InterCor et C-Roads), nous amènent à penser que cette technologie, bien qu'efficace, ne se suffit pas à elle-même dans certains cas.

Si nous regardons en arrière et que nous observons le déploiement de nouvelles technologies à grande échelle, nous pouvons distinguer 3 étapes :

- 1) Nous qualifierions la première étape, de phase « prototype ». Elle permet de confirmer l'efficacité de cette nouvelle technologie et ses bénéfices.
- 2) La deuxième étape est la phase de déploiement ; la nouvelle technologie est mise à disposition du client.
- 3) Enfin la dernière étape est atteinte quand, la nouvelle technologie n'est plus nouvelle et qu'elle fait partie du quotidien.

Prenons pour exemple le téléphone portable. En 1973, apparaît le premier prototype. En 1983, le produit est commercialisé pour la première fois. En 1990, 6% des habitants ont un téléphone portable. En 2002, le nombre de téléphones portables dépasse le nombre de téléphones fixes. En 2014, le nombre d'abonnements de téléphone mobile atteint 96% de la population humaine (sachant que certains possèdent plusieurs abonnements). Cette évolution du nombre de téléphones mobiles est accompagnée de l'augmentation de la couverture du réseau mobile.

En l'absence de réseau, il est impossible de téléphoner. C'est pourquoi le téléphone fixe coexiste avec le téléphone portable. Le premier ayant un réseau déjà en place et fiable, et le deuxième étant plus pratique, moins fiable et plus cher.

L'analogie avec les communications sur le canal G5 est assez évidente. Nous nous situons actuellement à la fin de la phase « prototype » et au début de la phase de commercialisation. Avec des constructeurs, comme Renault et PSA, qui proposent à des consommateurs des produits d'essai. Le taux de pénétration actuel est quasiment nul : il est de l'ordre de 3000 véhicules, qui représentent 0,007% des véhicules en circulation en France, et de 2000km de routes connectées, ce qui représente 0,18% du réseau routier Français. Il semble donc nécessaire de faire cohabiter cette nouvelle technologie avec une ancienne, plus en place et plus fiable. Contrairement au téléphone, nous ne pouvons pas nous baser sur un réseau filaire, il ne reste donc qu'une seule solution : Le réseau cellulaire.

Au cours des dernières années, dans le cadre des projets européens de déploiement des ITS, nous avons pu effectuer des relevés de performances, sur différents critères, des communications sur le canal G5. Ces observations qui sont présentées ci-dessous, nous ont amenés à nous poser des questions sur la connectivité du réseau quand le nombre d'utilisateurs possédant un véhicule connecté

est faible. Ces questions sont détaillées dans la deuxième partie du chapitre. Pour apporter des solutions aux problèmes soulevés, nous proposons, en dernière partie, deux modèles basés sur les communications cellulaires permettant de compléter le réseau G5, en attendant que le taux de pénétration des véhicules connectés soit suffisant.

I – Contributions à l’observation et la mesure des ITS

Afin de mettre en évidence ou de confirmer des problèmes liés à l’utilisation du canal G5, dans un environnement où le taux de pénétration de la technologie est faible, nous proposons de corréler les standards et les mesures effectuées dans le cadre de tests réels. Dans les sections qui suivent, nous présentons, dans un premier temps, les différents acteurs sur lesquels ont été faites les mesures, et dans un deuxième temps, les mesures relevées pendant des tests en laboratoire, sur piste ou sur route ouverte.

1. Indicateurs de performance

Les mesures de performances des communications G5, ont été faites en laboratoire, sur piste fermée et sur routes, dans le cadre des projets SCOOP@F [44], InterCor [45] et C-Roads [46]. Les résultats sont tirés de différentes ITS provenant d’une dizaine de partenaires (que nous nommerons fournisseur x par la suite par souci d’anonymat). La nature de ces partenaires est assez hétérogène : société d’autoroute, constructeurs et gestionnaires routiers, chacun ayant sous-traité le développement de leurs prototypes à différentes entreprises. Parmi celles-ci, figurent des start-up aussi bien que des entreprises du CAC40.

De plus, nous avons développé notre propre prototype, implémentant toutes les couches ETSI (voir Figure 2) [47].

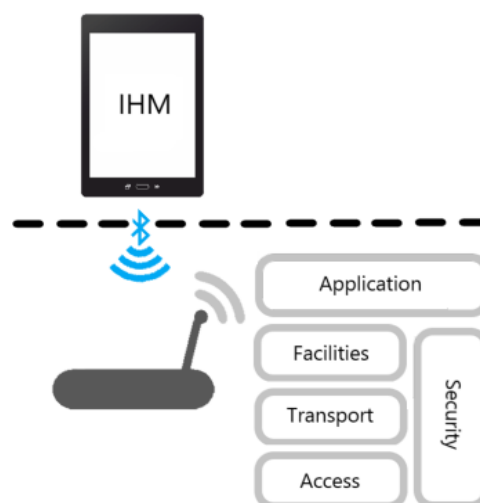


FIGURE 2 - ARCHITECTURE DE NOTRE PROTOTYPE

La multitude de sources de données est un facteur positif dans les conclusions que nous pouvons tirer des mesures. Si une mesure est la même chez tous les fournisseurs de prototype, nous pouvons supputer qu'elle représente une limite liée à la technologie (logique ou physique). Si une mesure présente des différences significatives, d'un fournisseur à l'autre, nous pouvons en déduire que le matériel ou l'implémentation logicielle joue un rôle sur les performances.

2. Les mesures

L'étude se concentre plus précisément sur la portée et la latence. La littérature et nos observations ont montré que ces deux paramètres peuvent varier négativement en fonction de la densité de véhicules dans le réseau [48], la charge réseau [49], de la topologie de la route et du bord de route ou même de l'orientation de l'antenne.

2.1 Portée

La portée de transmission est un paramètre crucial dans les communications sans fil. Dans le cas du réseau véhiculaire et du routage géographique, c'est ce qui permet ou empêche d'avoir un réseau connecté.

Si nous trouvons des défaillances de connectivité dans le meilleur des cas, cela vaudra pour tous les cas.

Pour relever les distances de communication, nous utilisons ce que nous appelons les Tlogs, qui sont un ensemble de données techniques, enregistrées par les véhicules dans le projet SCOOP@F. En comparant les données de l'émetteur et du récepteur, nous sommes capables, grâce aux positions, de déterminer la distance d'émission/réception.

Nous avons effectué des tests sur une piste fermée, au centre d'essai de l'IFFSTAR à Satory, entre différents protagonistes, dont l'anonymat est respecté en les nommant Fournisseur x. Nous avons retranscrit les résultats dans le Tableau 1, Ces tests ont été effectués à différentes vitesses, sans constater de changement dans les résultats. Les émetteurs sont des UEV (Unité Embarquée Véhicule) et les récepteurs des véhicules ou des UBR (Unité Bord de Route). Les portées observées sont de 300 ou 500 mètres selon les protagonistes.

TABLEAU 1 - VEHICULE EN EMISSION - SUR PISTE

Émetteur	Récepteur	Portée retenue
Fournisseur 3	Fournisseur 4	300 m
Fournisseur 1	Fournisseur 1	500 m
Fournisseur 4	Fournisseur 3	500 m
Fournisseur 3	Fournisseur 3	350 m

Sur autoroute, à Dordrecht (Pays-Bas), et à Reims (France), dans le cadre du projet InterCor, des UBRs ont été placées en situation réelle. Les partenaires du projet InterCor, ne bénéficiant pas tous des Tlogs, la mesure des distances a parfois été faite grâce aux enregistrements du logiciel Wireshark [50] ou une application Android spécialement développée pour l’événement.

Le Tableau 2, présente les relevés de portée obtenus avec des UBRs en émission, placées en hauteur, sur des mâts, et des UEVs en réception. La valeur retenue est un arrondi à 50 mètres près. La différence de portée observée sur le fournisseur 9 est dûe à la hauteur à laquelle était placée l’UBR, qui culminait à 4 mètres.

TABLEAU 2 - UBR EN EMISSION - SUR ROUTE

Émetteur	Récepteur	Portée retenue
Fournisseur 5	Fournisseur 2	1000 m
Fournisseur 5	Fournisseur 1	1600 m
Fournisseur 5	Fournisseur 6	1600 m
Fournisseur 5	Fournisseur 7	1700 m
Fournisseur 9	Fournisseur 8	3000 m

Dans le Tableau 3, les émetteurs sont des véhicules Le fournisseur 7 apparaît avec deux types d’équipements : UEVs et UBRs. Quand il s’agit de l’UBR, la précision est apportée entre parenthèse. Les portées des véhicules oscillent presque toujours autour des 300 mètres. Les valeurs observées sur route ne diffèrent pas de celles enregistrées sur piste.

TABLEAU 3 - VEHICULE EN EMISSION - SUR ROUTE

Émetteur	Récepteur	Portée retenue
Fournisseur 6	Fournisseur 8	300 m
Fournisseur 2	Fournisseur 6	250 m
Fournisseur 2	Fournisseur 7 (UBR)	230 m
Fournisseur 6	Fournisseur 7 (UBR)	300 m
Fournisseur 7	Fournisseur 7 (UBR)	450 m
Fournisseur 1	Fournisseur 7 (UBR)	350 m
Fournisseur 1	Fournisseur 7	130 m

2.2 Latence

Le traitement de la sécurité est une étape importante dans les échanges entre véhicules [41] [51]. Elle a un impact non négligeable sur les latences de communication entre 2 véhicules. Elle peut se faire selon deux approches : physique ou logicielle. Dans le projet SCOOP@F, le HSM est un requis

tandis que pour notre prototype nous avons adopté une approche logicielle. Les deux sections qui suivent présentent les mesures enregistrées pour chacune des deux approches. Le HSM est un module de sécurité physique, dans lequel un jeu de clef publique/privé est enregistré. Il prend une chaîne d'octets en entrée et donne la signature associée, en utilisant son jeu de clef et un algorithme de signature, en sortie [52].

Nous avons donc mesuré la latence, qui est le temps nécessaire à une entité pour arriver d'un point d'entrée à un point de sortie. Dans cette étude, nous mesurons 5 latences : de la couche accès vers la couche accès, la signature par un HSM (Hardware Security Module), la vérification par un HSM, la signature par logiciel, la vérification de la signature par logiciel.

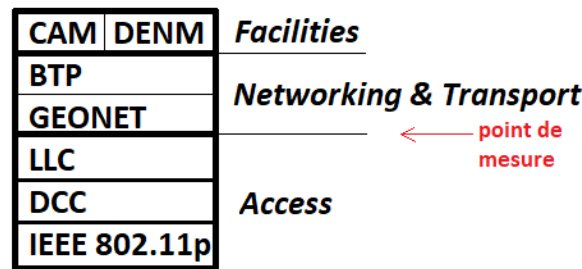


FIGURE 3 – COUCHES ETSI

En laboratoire, nous observons en G5 une latence de boîtier à boîtier (au niveau de la couche accès) d'environ 3 ms, pour un paquet de 400 octets. Ce qui diffère des 60 ms annoncées dans la littérature [49]. La Figure 3 précise à quel endroit de la pile protocolaire nous plaçons notre sonde de mesure.

Chaque marque d'HSM a des performances différentes et nous pouvons parfois monter à plusieurs dizaines de millisecondes. Le Tableau 4 regroupe quelques-unes des latences observées dans le projet SCOOP@F, pour la génération ou la vérification d'une signature.

TABLEAU 4 - LATENCE HSM

	Signature	Vérification
Fournisseur 1	80 ms	8 ms
Fournisseur 2	35 ms	4 ms
Fournisseur 6	8 ms	Pas de donnée
Fournisseur 5	5 ms	Pas de donnée

Sur notre prototype, nous gérons l’aspect sécurité côté logiciel. Les entrées et les sorties sont les mêmes, nous perdons les avantages liés au HSM, notamment celui de protéger la clef privée, mais nous gagnons en latence sur les opérations, comme l’illustre la Figure 5 et Figure 4.

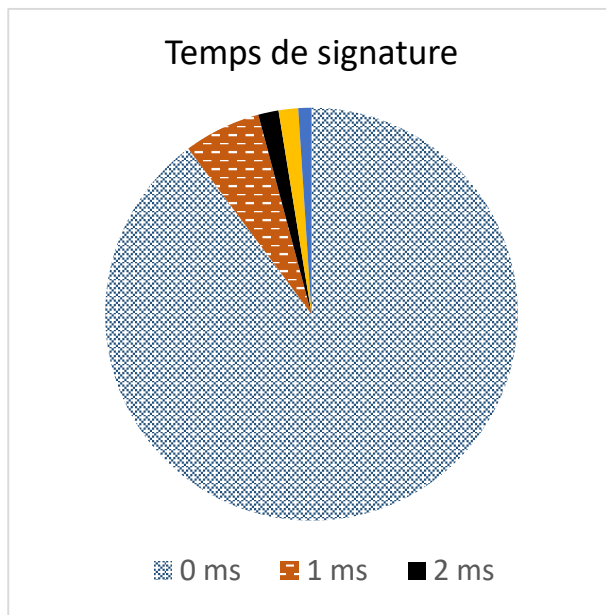


FIGURE 5 – SIGNATURE LOGICIEL

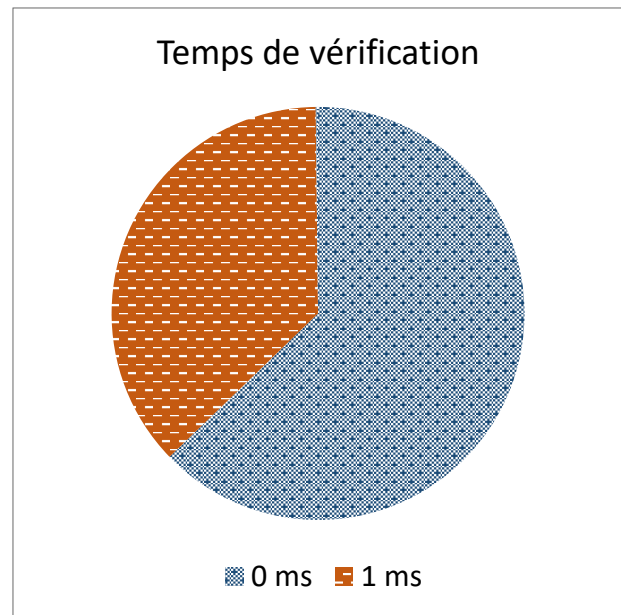


FIGURE 4 - VERIFICATION DE SIGNATURE LOGICIEL

Dans la majorité des cas, la génération de la signature, représentée en Figure 5, prend moins d’une milliseconde et très rarement plus de deux. Dans la Figure 4 qui représente le temps nécessaire à la vérification de la signature, nous ne dépassons jamais la milliseconde et le plus souvent nous sommes en deçà.

Pour résumer, nous avons, pour l’instant, présenté les différentes mesures que nous avons pu réaliser, à diverses occasions, dans les projets européens de déploiement des ITS. Ces mesures concernent la latence de communication d’un véhicule à l’autre, et les portées de transmission. Elles ont été faites en laboratoire, sur piste fermée, ou sur route ouverte, et elles ont fait intervenir les différents acteurs des projets. Nous allons maintenant croiser ces mesures avec le fonctionnement théorique des standards ETSI et de l’ITS-G5 pour en faire apparaître les limites technologiques.

3. Les limites des standards ETSI et l’ITS-G5

Les standards ETSI, présentés en Chapitre 1, ont été pensés en considérant que tous les véhicules sont équipés de la technologie G5. Certains mécanismes des standards, permettent de transporter un paquet lorsque nous sommes temporairement déconnectés du réseau. Par déconnecté, nous entendons l’incapacité d’avoir un accès direct ou indirect à une UBR, ou à un autre véhicule. Certains de ces mécanismes n’ont pas été retenus dans le déploiement actuel des ITS en Europe et d’autres ont pour conséquence d’allonger les temps de communication. Dans cette section, nous nous attelons à présenter les situations réalistes dans lesquelles l’ITS-G5 et les standards ETSI ne permettent pas d’assurer une diffusion efficace des événements.

3.1 Modèle

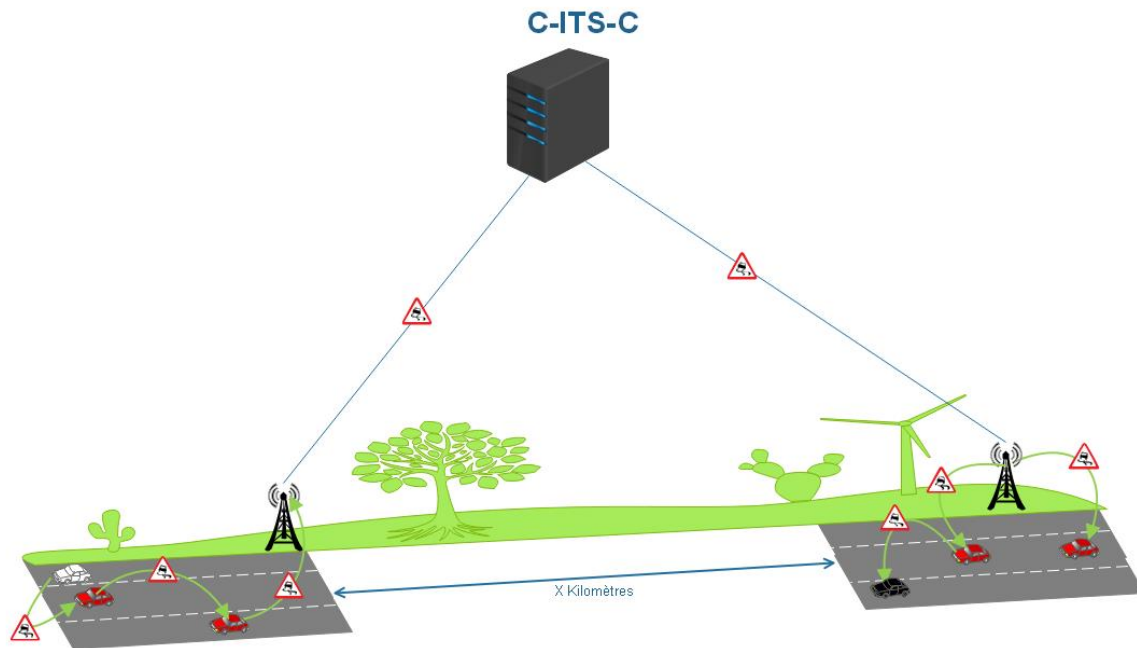


FIGURE 6 - COMMUNICATIONS DES ITS

La Figure 6 montre différents échanges entre des stations ITS. Nous avons le V2V (véhicule à véhicule), le V2I (véhicule vers infrastructure), l'I2I (infrastructure vers infrastructure), et enfin l'I2V (infrastructure vers véhicule).

C'est le modèle théorique des communications. Un véhicule détecte automatiquement, ou au déclenchement du conducteur ou de l'agent routier, un événement. Celui-ci est instantanément transmis aux véhicules qui suivent et si possible à une UBR, qui remonte l'information à la plateforme centrale et permet au gestionnaire de réagir, en envoyant du personnel et/ou en fournissant l'information à d'autres UBRs pour informer les véhicules susceptibles d'être concernés. Un événement peut aussi être créé directement depuis la plateforme centrale, comme des travaux programmés par exemple.

3.2 Configuration

Certains paramètres ont une incidence directe sur la distance qu'un paquet peut parcourir et sa capacité à persister en l'absence de voisin. Les normes ETSI proposent différents mécanismes permettant d'améliorer ces caractéristiques, mais ils n'ont pas forcément été implémentés dans le déploiement actuel.

Parmi ces mécanismes nous trouvons l'algorithme de routage. Comme présenté précédemment, l'ETSI en propose trois : Simple, CBF et Advanced. Le projet SCOOP@F ne prévoit que l'implémentation du simple GeoNetworking, que nous pouvons résumer ainsi : quand un véhicule reçoit un paquet de type GeoNetworking, soit ce véhicule est dans la zone de destination du paquet

et il le diffuse alors à tout le monde autour de lui, soit le véhicule est en dehors de la zone de destination et il envoie le paquet au voisin le plus proche de la zone de destination, par le biais de la diffusion unicast au niveau MAC.

On trouve également le Keep Alive Forwarding (KAF), qui est un mécanisme qui permet d'attendre de trouver un voisin intéressant avant de transmettre un paquet [40]. Il permet notamment au véhicule allant en sens inverse de transmettre l'information à un véhicule se trouvant derrière mais en dehors de la portée de transmission. L'absence de ce mécanisme va limiter la durée de vie d'un paquet et par conséquent la distance que celui-ci peut parcourir. Le KAF n'est pas requis pour être conforme aux spécifications SCOOP@F.

Le Maximum Hop Limit (MHL) est le nombre de fois qu'un même paquet GeoNetworking peut-être retransmis. C'est un paramètre variable, mis à défaut à 10 [38] et conservé dans les projets de déploiement précédemment cités. La distance qu'un paquet peut parcourir est lié à ce paramètre.

Enfin, dans SCOOP@F, pour le traitement de signatures, il est requis, pour des raisons de sécurité, de faire usage d'un HSM. Si nous faisons une moyenne des différentes performances obtenues avec ce module, nous arrivons aux résultats suivants :

En moyenne, une signature est générée en 32ms et la vérification de la signature est faite en 6ms. Ces délais vont avoir un impact notable sur la latence, puisque 32 ms c'est déjà un dixième de l'objectif des 300 millisecondes et la vérification de la signature, qui se répète à chaque saut, multiplie par 3 le délai nécessaire à l'échange d'un paquet entre 2 véhicules.

3.3 Problème de connectivité et de latence

Pour l'ITS-G5, la latence est directement liée à la connectivité. Le faible temps de communication d'un nœud à l'autre est l'argument principale pour utiliser cette technologie. Cependant, dès lors que la connectivité n'est pas assurée, la faible latence ne l'est pas non plus. D'autre part, ces faibles temps de communication ne sont valables que sur une distance limitée. Dans cette section, nous présentons plusieurs scénarios où la connectivité n'est pas effective. Ceux-ci sont réalistes d'un point de vue des infrastructures ou de la topologie du trafic.

En l'absence d'UBR, pour que chaque véhicule soit connecté au réseau, il faut au minimum une voiture équipée tous les 500m (valeur très optimiste). Or, même en considérant que tous les véhicules sont des stations ITS, cette condition ne peut être vérifiée. De plus, le temps nécessaire pour combler un blanc peut être très long.

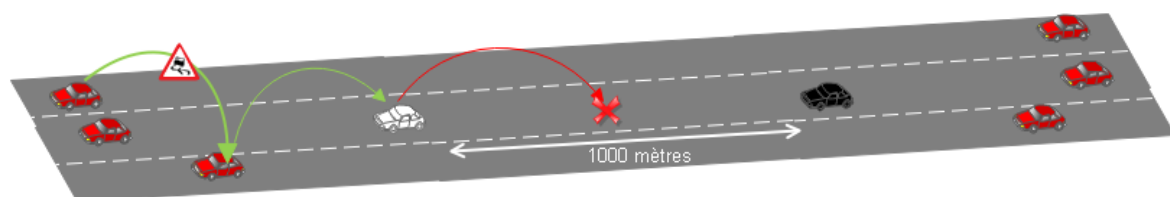


FIGURE 7 - PROBLEME DE CONNECTIVITE EN L'ABSENCE D'UBR

Dans la Figure 7 ci-dessus, une voiture génère un événement routier qui parvient par rebond jusqu'au véhicule blanc. Ce dernier n'est pas à portée du véhicule noir et ne peut donc pas faire parvenir le paquet plus en aval. Si par chance la voiture blanche roule à 110km/h et la voiture noire à 130km/h, alors :

- Le véhicule noir parcourt 5,55m de plus que la blanche chaque seconde et il faut rattraper 500m pour arriver à portée radio de la voiture blanche, soit en temps : $500 / 5,55 = 90s$
- Pendant ces 90s, le véhicule noir aura déjà dépassé l'événement sans en avoir été alerté.

Pour éviter ce scénario, nous avons recours à des UBRs qui, placées en hauteur, émettent bien plus loin. Ici, nous considérons une UBR avec une portée de 3km. Pour que toute la route soit couverte par une UBR, il faudrait en placer une tous les 6km. De cette manière, nous nous assurons que si un événement est généré au niveau du ITS-C (Intelligent Transport System Central), chaque véhicule sera en mesure de le recevoir, sans délai particulier. Cependant, côté véhicule, la portée d'émission est toujours la même, il existe donc une zone blanche longue de 5km entre chaque UBR.

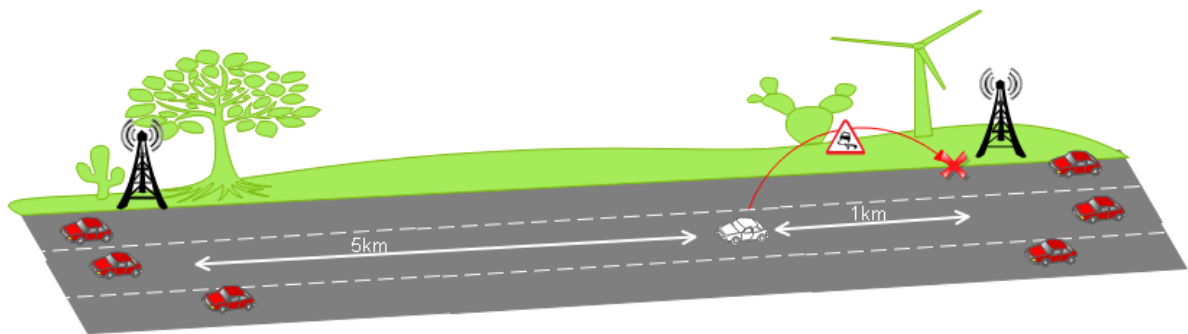


FIGURE 8 - PROBLEME DE CONNECTIVITE AVEC DES UBRs

Dans la Figure 8 ci-dessus la voiture doit parcourir 3500m avant d'être en mesure de faire parvenir son événement au réseau, à 130km/h c'est 100 secondes qui seraient nécessaires pour parcourir cette distance.

On se rend compte qu'à moins de placer une UBR tous les kilomètres (ce n'est pas envisageable, pour des raisons économiques) nous ne pouvons nous assurer d'une connectivité continue.

3.4 Les problèmes de distance

On peut distinguer deux catégories d'événements : celle des alertes et celle des informations. Dans les deux cas, tous les événements sont importants pour la sécurité routière. Les alertes doivent être reçues par les véhicules adjacents le plus rapidement possible, cette tâche est très bien assurée par l'ITS-G5. Les événements d'information, de type bouchon, véhicule à l'arrêt ou travaux, sont importants pour accroître la vigilance du conducteur et c'est entre autres pour cela qu'il est important d'avertir le conducteur au plus tôt, pour qu'il ralentisse ou même qu'il décide de changer de route.

Cette section vise à mettre en évidence les lacunes de l’ITS-G5 pour faire parvenir un événement à bonne distance de son lieu d’origine.

Sans UBR, la distance maximale qu’un paquet peut parcourir est limité par le nombre de sauts qu’il peut effectuer (MHL). Et dans le cas optimal où un véhicule est placé exactement tous les 500m, cette distance est de 5km. Une valeur suffisante pour certains cas d’usage mais insuffisante pour d’autres. Par exemple, signaler la présence d’un bouchon sur autoroute aux véhicules se trouvant à 5km, permet aux usagers d’être prudents et de ne pas arriver trop vite sur celui-ci, mais ne permet pas à ceux qui se trouvent plus loin de s’adapter en quittant ce tronçon de route pour en emprunter un autre.

Maintenant, si nous considérons qu’une UBR est placée tous les 6km, alors un véhicule aura toujours une UBR placée au maximum à 3km (valeur supposée optimiste). Une distance atteignable avec un MHL à 10, si le placement des véhicules le permet. A partir du moment où un paquet arrive à une UBR, il n’a plus de limite de distance. Cette répartition des UBR est valable sur des routes droites et dégagées. Sur des routes sinueuses, la densité d’UBR se doit d’être plus importante pour couvrir la totalité de la route. De plus, si les autoroutes bénéficient déjà d’infrastructures sur lesquelles placer les UBR, ce n’est, par exemple, pas le cas des routes régionales.

D’autre part, nous rappelons les latences suivantes :

Signature du paquet (α) – une fois : 32ms

Vérification de la signature (β) – chaque saut : 6ms

Algorithme de routage (γ) – chaque saut : 1ms

Communication G5 (δ) – chaque saut : 3 ms

Chaque saut prend donc 10ms ($\sigma = \beta + \gamma + \delta$) et nous cherchons à avoir une latence maximale de 300 ms (ω).

$$(\omega - \alpha) / \sigma = 27$$

En considérant cette fois-ci qu’il n’y pas de limite de saut, et en supposant qu’il existe un chemin dans le trafic de telle sorte que le paquet ne s’arrête jamais et ne fasse que des bonds de 500m, celui-ci ne pourra pas aller au-delà de 13,5 kilomètres avant d’atteindre la barrière des 300ms.

3.5 Discussions

L’ITS-G5 est une technologie prometteuse, avec des temps de transmission très faibles. Néanmoins, nous nous rendons compte que dans cet environnement, les véhicules ne peuvent se satisfaire à eux-mêmes. Nous sommes obligés d’avoir recours à des UBRs pour assurer la connectivité de toutes les routes ; ce qui, pour une couverture totale, représente plusieurs centaines de millions d’euros d’investissement et des années de déploiement. Il semble donc, si ce n’est nécessaire, utile d’avoir un mécanisme qui permette d’acheminer les informations, dans un sens ou dans un autre, pendant ces années de transition, où le taux de pénétration des stations équipées de cette nouvelle technologie n’est pas suffisant. Afin de valider et de quantifier les problèmes de connectivité dans les réseaux ITS

sur autoroute, nous proposons dans la section suivante un simulateur qui reproduit la circulation sur ce type de route.

4. Proposition d'un simulateur pour autoroute

Les mesures de portée de communication des véhicules sur le canal G5, nous ont permis, en l'absence d'une forte densité d'UBR, de mettre en évidence d'éventuelles absences de connectivité des réseaux véhiculaires. La simulation a pour but de mesurer l'impact de ces absences.

4.1 Simulation d'un réseau ITS sur autoroute

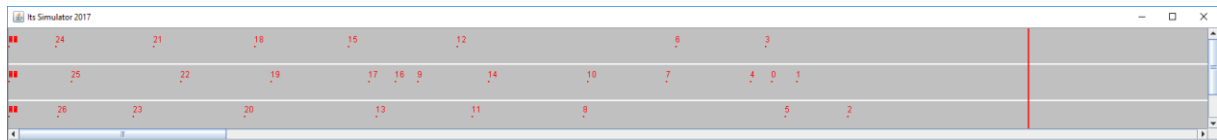


FIGURE 9 - SIMULATEUR ITS SUR AUTOROUTE (IHS)

Nous avons conçu un simulateur (illustré en Figure 9), que nous appellerons par la suite IHS (ITS Highway Simulator), qui reproduit la circulation sur l'autoroute. Les véhicules ont été implémentés de sorte qu'ils reproduisent les règles qu'un conducteur est censé adopter : à savoir de doubler un véhicule plus lent en prenant la voie de gauche et de se rabattre sur la droite quand le dépassement a eu lieu. Dans le cas où un véhicule aurait entamé un dépassement mais qu'un autre véhicule arrive derrière lui à vive allure, alors il abandonne son dépassement si cela est possible. Comme dans la réalité, les véhicules respectent plus ou moins les distances de sécurité, à savoir que certains ne la respectent pas du tout et d'autres la respectent rigoureusement.

Le simulateur IHS représente un segment d'autoroute de trois voies, et un événement de type bouchon est généré au bout de 5km. Quand le premier véhicule arrive sur le bouchon il génère un DENM, qu'il transmet aux véhicules qui le suivent selon la logique de la version simple du protocole de routage geoNetworking. Comme dans la réalité, chaque véhicule qui reçoit le DENM vérifie la validité de sa signature avant de poursuivre le routage du paquet.

Les véhicules démarrent au kilomètre 0, avec une vitesse comprise entre 100 et 130kmh, sur chaque voie, à un intervalle régulier, plus ou moins un epsilon de 2secondes.

On mesure à chaque itération de la simulation le temps nécessaire pour qu'un DENM généré au kilomètre 5, atteigne la zone de destination dont la position se trouve au kilomètre 0 avec un rayon de 500 mètres.

4.2 Configuration et résultats

TABEAU 5 - CONFIGURATION DE L'ORDINATEUR POUR LA SIMULATION

Processeur	i7 @3.6GHZ
Mémoire vive	32Go
Système d'exploitation	Windows 10 x64
Carte graphique	Intel HD Graphics 630
IDE	Eclipse neon 3
Langage de programmation	Java 8

Eclipse néon est un logiciel développé par Eclipse Foundation [53] qui offre un environnement de développement java notamment.

Nous avons exécuté la simulation avec 3 configurations différentes, en faisant varier la fréquence d’insertion du véhicule et sa portée de communication.

Dans la première configuration, nous insérons un véhicule sur chaque voie toutes les 15 secondes, plus ou moins 2 secondes et la portée de communication est de 500m. A raison d’une vitesse comprise entre 26 et 36 mètres par seconde, quand un véhicule est inséré, il est, la plupart du temps, encore à portée des véhicules insérés dans la vague précédente. La courbe ci-dessous (Figure 10) présente les temps enregistrés pour 100 simulations.

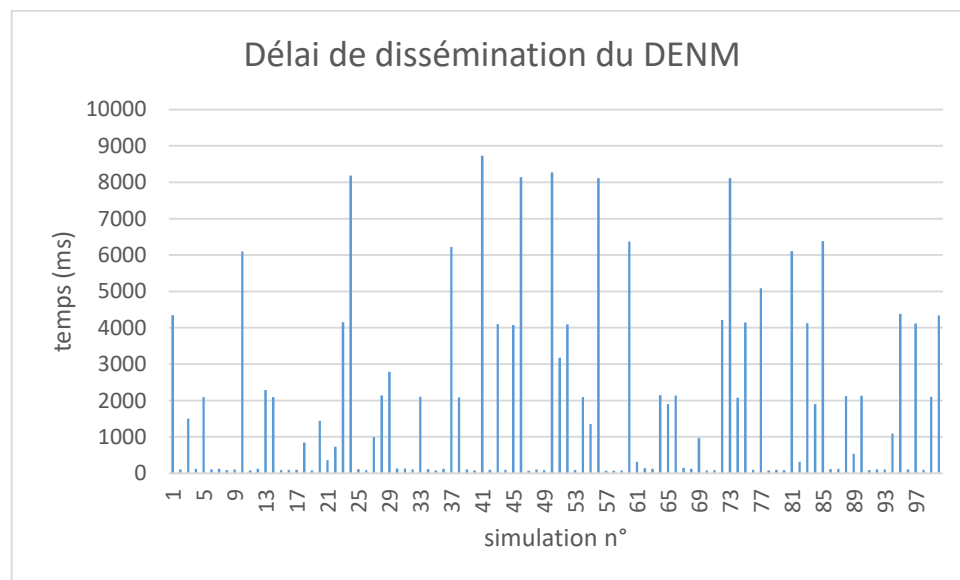


FIGURE 10 - RESULTATS DE LA SIMULATION N°1

Dans 44% des cas, le paquet a pris plus d’une seconde à atteindre la zone de destination et dans le pire des cas il a fallu 8 secondes, tout cela pour un délai moyen de 1,8 seconde. En réduisant l’intervalle d’insertion des véhicules à 10 secondes, plus ou moins 2 secondes, nous obtenons le résultat illustré par la Figure 11.

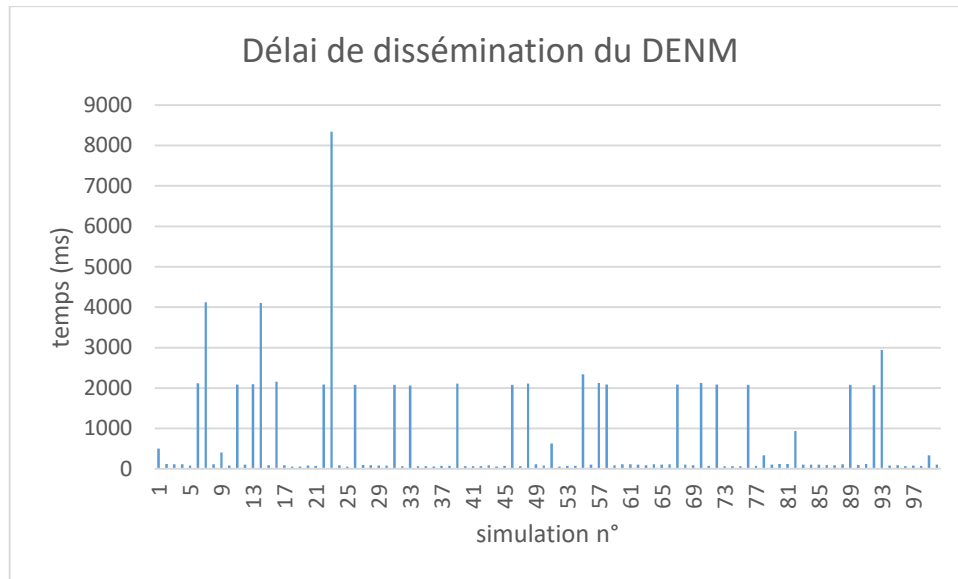


FIGURE 11 - RESULTATS DE LA SIMULATION N°2

Cette fois, dans seulement 24% des cas nous dépassons une seconde de délai. De plus, dans seulement 5,5% nous dépassons les deux secondes de délai, contre 26% pour la configuration précédente. Dans cette configuration, le délai moyen de dissémination est de 0,7 ms.

Dans la configuration suivante, nous réduisons la portée de transmission à 300m, une valeur tout à fait réaliste compte tenu de nos observations et cela se fait ressentir dans les résultats. En insérant les véhicules à la même fréquence que précédemment (10 secondes en moyenne), nous obtenons les résultats suivants (Figure 12).

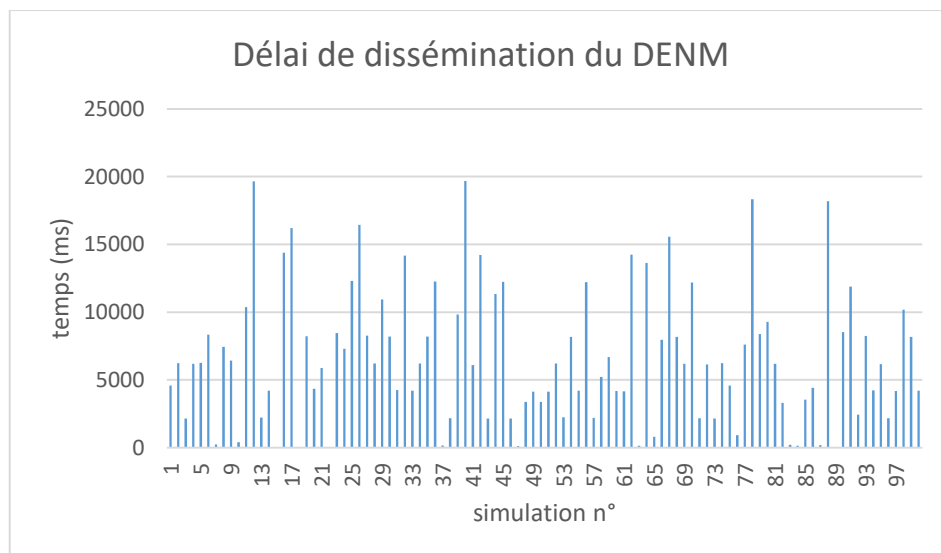


FIGURE 12 - RESULTATS DE LA SIMULATION N°3

Cette fois, dans 86% des cas la dissémination a pris plus d'une seconde, elle est même supérieure à 5 secondes pour plus de la moitié des simulations (56%), pour un délai moyen de 6,9 secondes, avec un délai maximal de 19,6 secondes.

4.3 Discussions

Avec le G5, les délais de communication sont très bons lorsque nous avons de la connectivité, les résultats oscillent entre 70 et 100 millisecondes pour remonter l'événement sur 5 Km. Cependant, plus nous augmentons la distance, plus nous augmentons le nombre de sauts nécessaires, et plus nous augmentons également la chance de tomber sur une zone blanche, où l'espace séparant deux véhicules qui se suivent est plus grand que la portée de communication. Le temps nécessaire pour combler ces zones blanches est souvent de l'ordre de plusieurs secondes et c'est pourquoi, dans certaines configurations, nous avons un délai moyen du même ordre.

Ces simulations ont été faites avec un taux de pénétration de 100% de véhicules communiquant en G5. La portée de 500m pour tout le monde est une valeur optimiste étant données les observations faites sur route et sur piste. En outre, nous ne bénéficions pas, encore tout du moins, d'UBR tous les 5km. Et nous n'avons pas non plus pris en compte les éventuelles collisions qui aurait encore dues rajouter un peu au délai.

II – Contributions à base de Cloud

La solution que nous proposons, repose sur l'utilisation du réseau cellulaire. Nous faisons ce choix en raison de l'omniprésence d'équipements capables de communiquer sur ce réseau, de sa couverture importante sur le territoire – qui tend, d'autre part, à s'agrandir – et des délais de communications qui correspondent théoriquement aux besoins.

Ainsi, les gestionnaires de routes n'ont plus qu'à identifier les parties de leur réseau ne bénéficiant pas d'une couverture cellulaire optimale et les zones à danger, et peuvent placer des UBR de façon précise en fonction de ces deux critères.

Cette solution n'est pas une alternative, c'est un complément au G5, qui permet dans un premier temps de palier le faible taux de pénétration de celui-ci et qui pourra persister par la suite pour, par exemple, alléger sa bande passante limitée.

Nous avons étudié deux solutions de type Cloud, l'une sous la forme de Software as a Service (SaaS) et l'autre sous la forme de Network as a Service (NaaS).

1. Solution Cloud SaaS

Dans cette section nous présentons l'architecture de notre solution, les acteurs et le déroulement d'un échange. Pour cela, nous proposons d'utiliser le Cloud distant en tant que service logiciel, dans le sens où nous l'utilisons comme une API. La communication avec ce service se fait selon un protocole requête/réponse. Nous présentons ensuite une expérimentation sur route ouverte pour mesurer le temps d'échange requête et réponse entre un client et le Cloud, ainsi que le temps qui s'écoule entre la génération d'un événement et l'obtention de celui-ci par un véhicule.

1.1 Description

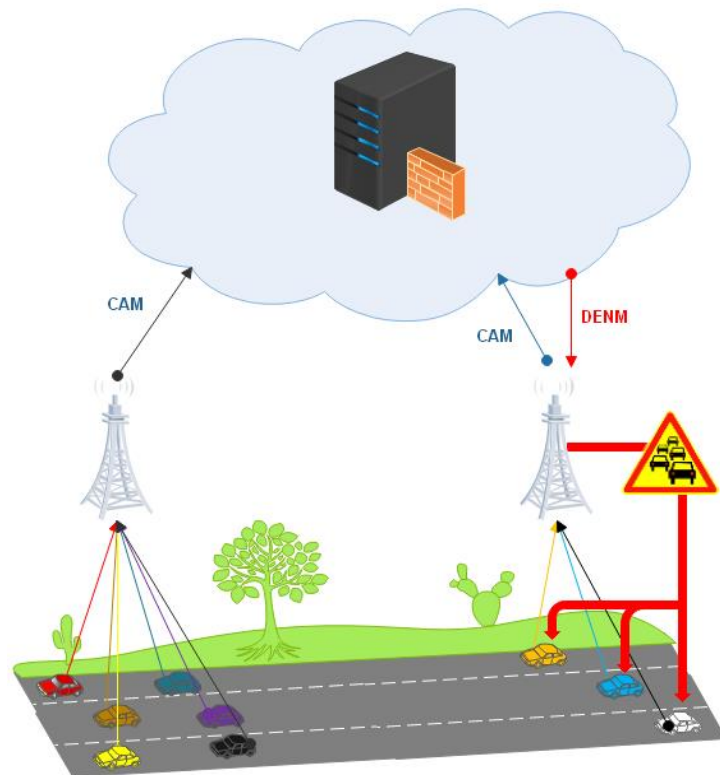


FIGURE 13 - COMMUNICATION ITS AVEC SAAS

La Figure 13 présente les communications entre les différentes stations ITS dans une architecture contenant un Cloud de type SaaS. Les véhicules communiquent via le réseau cellulaire en envoyant des CAMs et reçoivent sur le même réseau les DENMs pertinents dans leur zone. Sur cette figure, le serveur est présenté derrière un proxy, puisque c'est généralement le cas en situation réel, et c'était le cas dans nos expérimentations.

Le client peut se manifester sous deux formes : soit l'UEV est équipée d'un module permettant de communiquer en cellulaire, soit le client est exécuté directement sur un smartphone. Il émet des CAMs à une fréquence de 10Hz et affiche, s'il est valide, chaque événement qu'il reçoit. Par valide, nous entendons non expiré et bien localisé.

Le client se résume à deux clients légers très simples et ne nécessite donc aucune puissance de calcul, tout est déporté dans le Cloud.

```

1: WHILE (true) DO
2:   IF (DIFFERENCE(lastPos, currentPos) > TRESHOLD) THEN
3:     SEND(cam)
4:   ELSE IF (lastCamSent > 1000ms) THEN
5:     SEND(cam)
6:   END IF
7:   RECEIVE(httpAnswer)
8:   IF (httpAnswer = denm) THEN
9:     STORE(denm)
10:  END IF
11:  SLEEP(100)
12: END WHILE

```

ALGORITHME 1 - CLIENT HTTP – TRAITEMENT DES CAMS

L'Algorithme 1 décrit les différentes opérations de traitement des CAM pour le client, que ce soit pour l'envoi ou pour la réception. Le CAM est envoyé soit toutes les secondes (instruction 4 et 5), soit si la différence entre mon ancienne et ma nouvelle position a dépassé un seuil (instructions 2 et 3). Après l'envoi du CAM, il attend la réponse du serveur (instruction 7) et si celle-ci contient un ou plusieurs DENMS, ceux-ci sont stockés afin d'être traités par une seconde partie du client.

```

1: WHILE (true) DO
2:   FOREACH (denm ∈ denmStored) DO
3:     IF (ISvalid(denm) AND ISnotDISPLAYED(denm)) THEN
4:       DISPLAY(denm)
5:     ELSE IF (ISnotVALID(denm)) THEN
6:       REMOVEorHIDE(denm)
7:     END IF
8:   END FOREACH
9:   SLEEP(100)
10: END WHILE

```

ALGORITHME 2 - CLIENT HTTP – TRAITEMENT DES DENMS

L'Algorithme 2 présente les opérations qui s'appliquent aux DENMs stockés dans la mémoire du client. Si le DENM est valide et n'est pas encore affiché sur l'interface homme-machine, alors nous procédons à l'affichage (instructions 3 et 4). A l'opposé si un DENM n'est pas valide, nous l'effaçons de l'interface homme-machine et nous le supprimons de la mémoire si le temps de validité est dépassé (instructions 5 et 6).

Le serveur est déployé au sein d'un Cloud. Nous supposons qu'il dispose d'une capacité de calcul et de stockage sans limite. De plus, Il a connaissance en temps réel de la topographie exacte de la circulation (véhicule arrêté, changement brutal de conduite, bouchon). En couplant les données de chaque véhicule, il est capable de détecter et de générer des événements et d'identifier les clients à qui ils sont destinés.

A la réception d'un CAM, trois routines sont exécutées successivement sur le serveur : la mise à jour du véhicule dans la base de données (position, vitesse, orientation, etc.), une routine de calcul d'événement dans la zone du véhicule, une routine de recherche d'événements déjà existants dans la zone (Algorithme 4). Le résultat de ces opérations est une liste d'événements de taille 0..n.

```

1: RECEIVE(cam)
2: UPDATEdatabase(cam)
3: denms : Liste
4: tce : ThreadComputeEvent(denms)
5: tre : ThreadResearchEvent(denms)
6: START(tre)
7: START(tce)
8: JOIN(tre)
9: JOIN(tce)
10: SENDdenmTO(cam.sender, denms)
    
```

ALGORITHME 3 - SERVEUR SAAS – ROUTINE DE RECEPTION D'UN CAM

L'Algorithme 3 décrit le processus qui a lieu après la réception d'un CAM, les routines de calcul ou de recherche d'événements se font en parallèle (instructions 4 à 7) et vont remplir la liste déclarée à l'instruction 3, avant que la routine principale ne l'envoie au client (instruction 10).

```

1: FOREACH (denm ∈ denmsTriggerd) DO
2:     IF(cam.pos ∈ denm.destinationArea) THEN
3:         ADDto(denms, denm)
4:     END IF
5: END FOREACH
    
```

ALGORITHME 4 - SERVEUR SAAS – ROUTINE DE RECHERCHE DE DENM

Les opérations au sein du serveur sont de l'ordre de l'instantané. L'Algorithme 4 présente une simple boucle qui compare la position d'un client avec la position de chacun des DENMs. S'il y a correspondance, le DENM en question est ajouté à la liste qui sera renvoyée au client.

On ne présente pas les algorithmes de détection, qui varient pour chaque type d'événements et qui seraient trop nombreux. Nous pouvons tout de même préciser que dans les spécifications de SCOOP@F, les conditions de génération de chaque événement sont décrites et qu'il suffit de les appliquer au niveau du serveur, plutôt qu'au niveau des véhicules. Dans l'Algorithme 5 ci-dessous, nous proposons une forme générique d'un algorithme de détection d'un événement.

```
1: data : liste
2: denms : liste
3: FOREACH (cam ∈ DATABASE) DO
4:     collectSpecificData(cam, data)
5: END FOREACH
6: denms = analyseData(datas)
7: FOREACH (denm ∈ denms)
8:     denmsTriggered.add(denm)
9: END FOREACH
```

ALGORITHME 5 - DETECTION D'UN EVENEMENT

Pour résumer, la première étape des communications est la transmission au serveur, par le client, d'une requête, laquelle contient son CAM. Après réception de la requête, le serveur extrait la position du véhicule pour effectuer une recherche des événements dans la zone du client. En parallèle, il enregistre les nouvelles informations contenues dans le CAM avant de calculer si celles-ci permettent de conclure sur la présence d'un événement sur la route. Enfin, le serveur transmet au client la liste des DENMs le concernant.

1.2 Expérimentation

Afin de pouvoir évaluer notre solution et la comparer avec d'autres propositions, nous avons effectué, sur routes, avec plusieurs véhicules, pendant plusieurs heures, en émettant un CAM toutes les 100ms, des relevés sur 3 indicateurs :

- Le délai de requête/réponse avec un CAM
- Le délai pour recevoir un DENM après sa génération
- Les pertes de connexion et leur durée

Notre expérimentation consiste à effectuer un trajet, d'un point A à un point B, pour pouvoir faire des relevés sur nos 3 indicateurs. Nous avons effectué 1 trajet dans la ville de Reims et 4 sur des autoroutes distinctes (A4, A31, A20 et A71). Ces trajets nous ont permis de mettre en évidence une disparité entre autoroute et ville, au niveau des pertes de connexion d'abord, mais également une disparité sur les latences moyennes et générales. Le détail des trajets est le suivant :

- Le trajet 1 a eu lieu sur la ville de Reims et sa périphérie. Il a duré 49 minutes.
- Le trajet 2 a eu lieu sur l'A4 et l'A31 allant de Reims à Metz. Le trajet a duré 120 minutes.
- Le trajet 3 a eu lieu sur l'A20 entre Toulouse et Limoges. Il a duré 111 minutes.

- Le trajet 4 a eu lieu sur l'A20 après Limoges. Il a duré 85 minutes.
- Le trajet 5 a eu lieu sur l'A71 en direction de Paris. Il a duré 223 minutes.

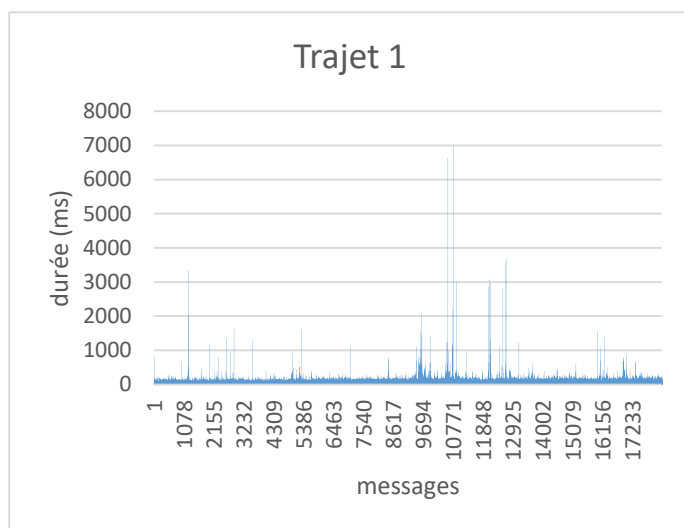


FIGURE 14 - TRAJET 1 - DUREE D'UN ECHANGE

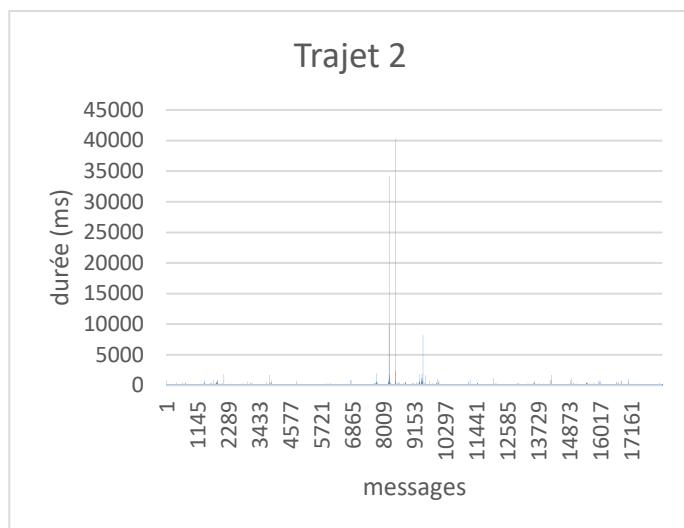


FIGURE 15 - TRAJET 2 - DUREE D'UN ECHANGE

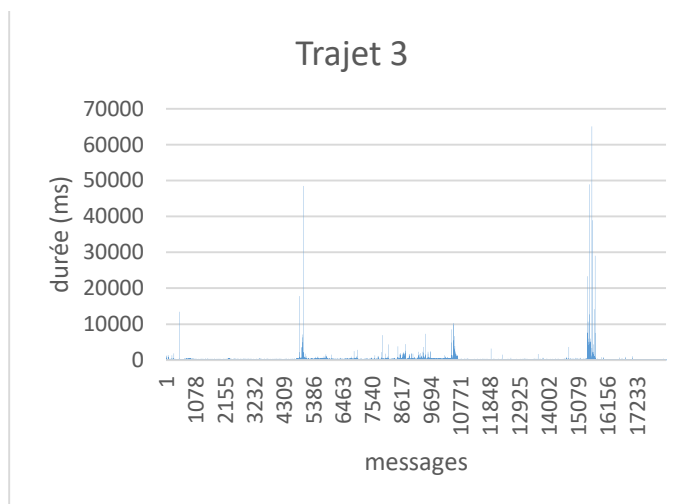


FIGURE 17 - TRAJET 3 - DUREE D'UN ECHANGE

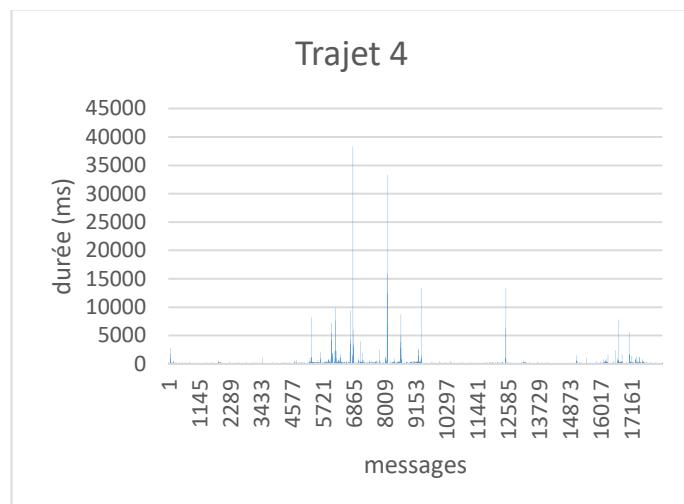


FIGURE 16 - TRAJET 4 - DUREE D'UN ECHANGE

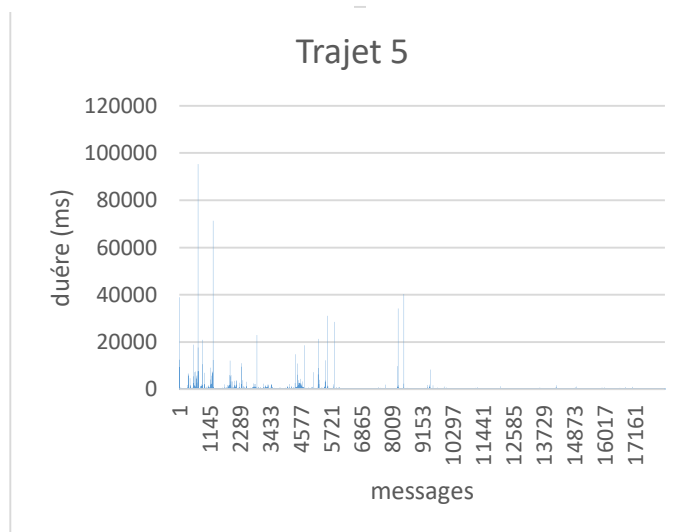


FIGURE 18 - TRAJET 5 - DUREE D'UN ECHANGE

Dans la Figure 14, l'échelle est bien plus réduite, car en milieu urbain les communications cellulaires sont plus stables et les délais meilleurs. Pour la Figure 16, la Figure 17 et la Figure 18, nous observons le même comportement : de rares pics de délais de plusieurs dizaines de secondes, la majorité des échanges ont lieu en moins de 300ms et une petite part a pris entre 1 et 10 secondes à se conclure.

En moyenne, sur tous les trajets extra-urbains confondus (193 016 messages), les échanges durent en moyenne 169ms. 16,3675% d'entre eux ont dépassé les 200ms. Nous nous intéressons aux échanges de 200ms, car si nous rajoutons les 100ms d'intervalle entre les CAMs, nous arrivons aux 300ms correspondant à la limite que nous ne voulons pas dépasser. En milieu urbain la moyenne tombe à 130 et seulement 4,9% des échanges dépassent les 200 ms. Les figures suivantes présentent le délai nécessaire pour qu'un DENM soit envoyé après sa génération.

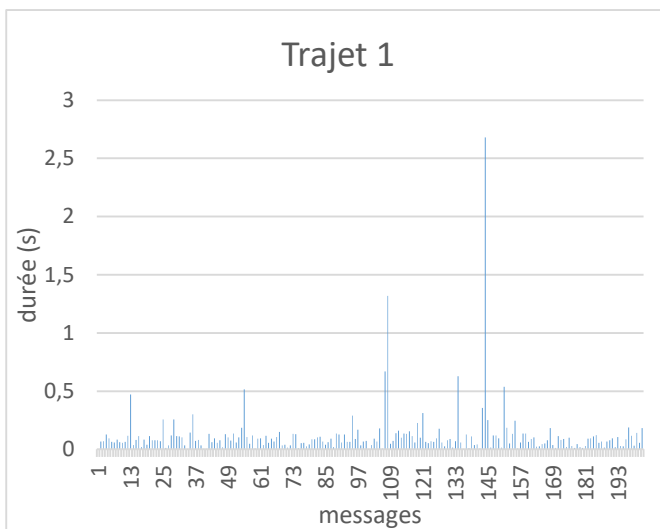


FIGURE 20 - TRAJET 1 - DELAI D'ENVOI DENM

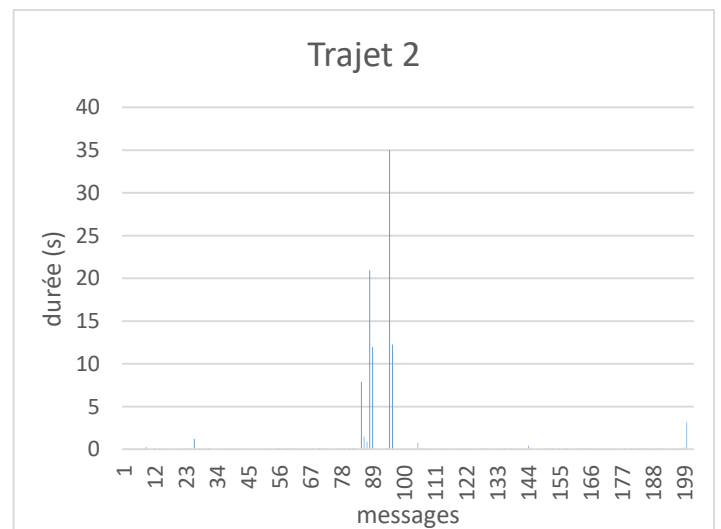


FIGURE 19 - TRAJET 2 - DELAI D'ENVOI DENM

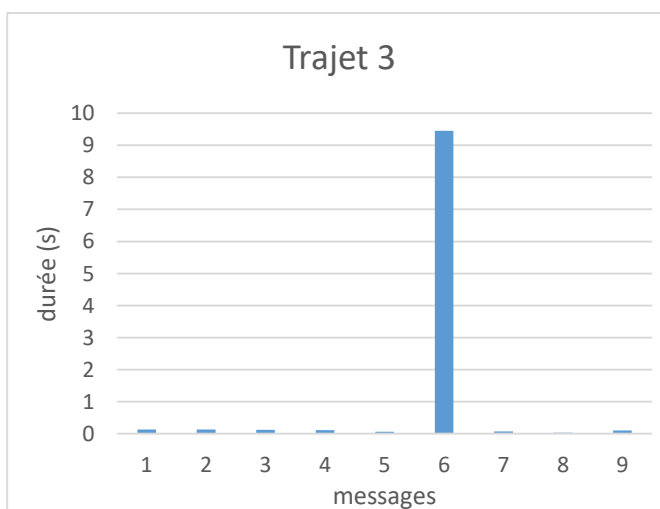


FIGURE 22 - TRAJET 3 - DELAI D'ENVOI DENM

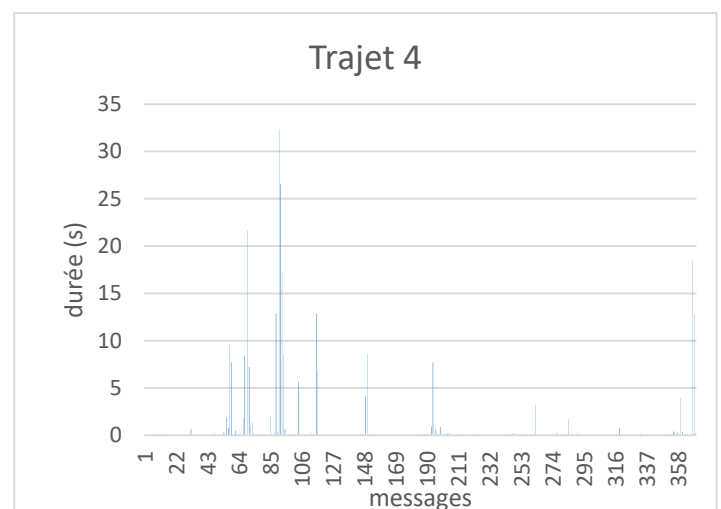


FIGURE 21 – TRAJET 4 - DELAI D'ENVOI DENM

Le constat est le même : la Figure 20 représente, en milieu urbain, le temps écoulé entre le moment où un DENM est généré et celui où il est envoyé au client. L'échelle est plus réduite que pour la Figure 19, la Figure 22 ou la Figure 21 qui représentent des échanges qui ont eu lieu sur routes extra-urbaines, car quelques DENMs prennent plusieurs dizaines de secondes à être envoyés.

En tout, tous trajets extra-urbains confondus, 863 DENMs ont été envoyés. Ils l'ont été, en moyenne, 548ms après leur détection. Cependant, 91,16% d'entre eux ont été envoyés en moins de 250ms après leur détection. En interprétant ces chiffres, nous comprenons que la plupart des DENMs ont été envoyés dans un délai permettant de remplir la condition des 300ms, mais que quelques-uns font chuter drastiquement la moyenne. Par exemple le DENM qui a pris le plus de temps à être envoyé, l'a été après 32 secondes. En milieu urbain, la moyenne du temps écoulé avant l'envoi d'un DENM après sa détection tombe à 116 ms et 93% d'entre eux sont envoyés en moins de 250 ms. L'échantillon urbain est plus réduit (1 seul trajet), mais notons tout de même que le DENM le plus tardif a été envoyé après 2,6 secondes.

1.3 Discussion

La solution de type SaaS auquel nous accédons par réseau cellulaire ne permet pas de délivrer, de façon sûre et dans les délais requis, un message critique (freinage d'urgence, queue de bouchon). Cependant, avec 90% des messages qui arrivent en moins de 250ms et une moyenne de 548ms, la solution est viable pour les cas d'usage non critiques (travaux, bouchons etc.).

2. Solution Cloud NaaS

Notre deuxième solution est plus passive et son comportement se rapproche d'une ITS accessible à longue distance en IP. Elle apporte un service de routage aux usagers, que nous nommons Network as a Service. Dans cette architecture le serveur reçoit des messages des ITS-V, CAM ou DENM encapsulés dans une en-tête GeoNetworking. Grâce au CAM, l'ITS-C (le serveur) se construit une table de voisinage et route les DENMs qu'il reçoit selon les règles du GeoNetworking mais via une connexion cellulaire cette fois.

2.1 Description

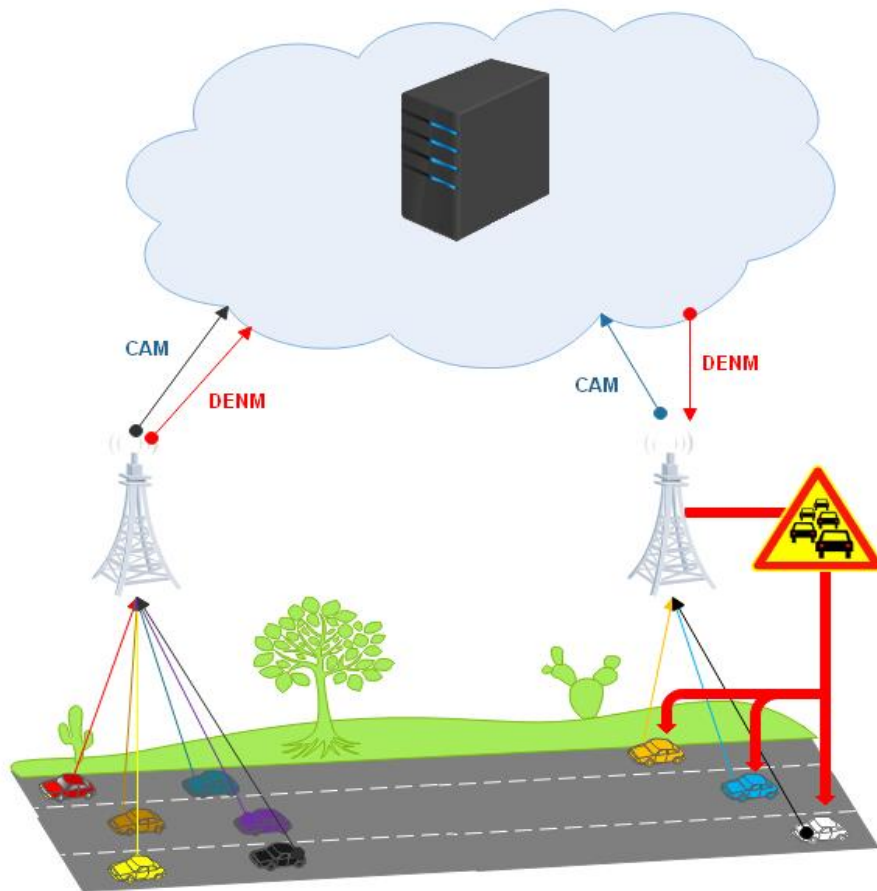


FIGURE 23 - COMMUNICATIONS NAAS

La Figure 23 présente une architecture où les échanges se font, avec le serveur, via une connexion persistante.

Les clients envoient leur CAMs à l'ITS-C, l'informant ainsi de leur position. Les DENMs envoyés à l'ITS-C sont toujours encapsulés dans leur couche GeoNetworking. A la réception de ceux-ci, le serveur les transfère aux clients concernés.

Le protocole GeoNetworking fonctionne en partie, grâce à une table des voisins. Quand un nœud du réseau reçoit un message GeoNetworking, il inscrit dans sa table la position de celui-ci. Ce qui lui permet, par la suite, de sélectionner son voisin le plus pertinent pour envoyer un message GeoNetworking vers une zone. Dans cette logique, L'ITS-C apparaît comme un « super-voisin » accessible partout où le réseau cellulaire le permet. Dans ce modèle nous sommes donc dans un mode de push de l'information où l'ITS-C joue le rôle de routeur GeoNetworking accessible à tout le monde.

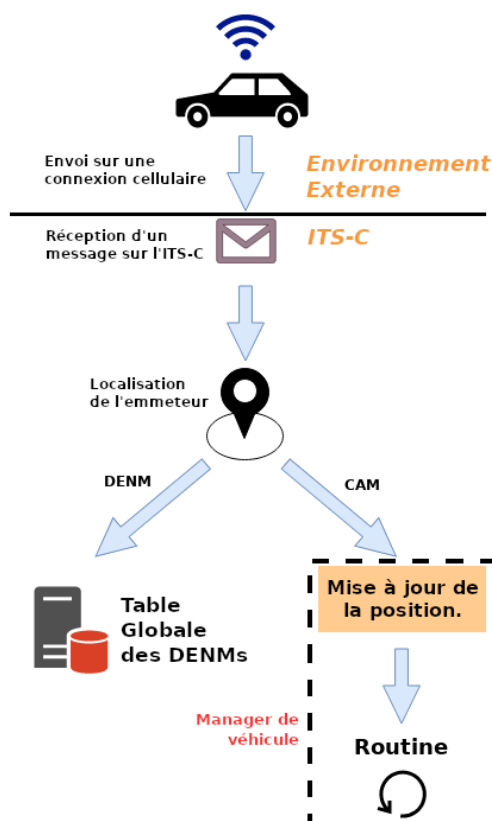


FIGURE 24 - ROUTINE DE RECEPTION D'UN MESSAGE SUR LA STATION CENTRALE

La station centrale reçoit des messages GeoNetworking. S'il contient des DENMs, ceux-ci sont stockés dans une base de données commune des DENMs, comme le montre la Figure 24. S'il contient un CAM, la nouvelle position du véhicule est extraite. L'ITS-C est composée d'un certain nombre de managers, chacun étant dédié à un véhicule.

Si la position du véhicule a changé ou si la base de données des DENMs locale diffère de la base de données commune, nous appelons une opération de mise à jour du manager.

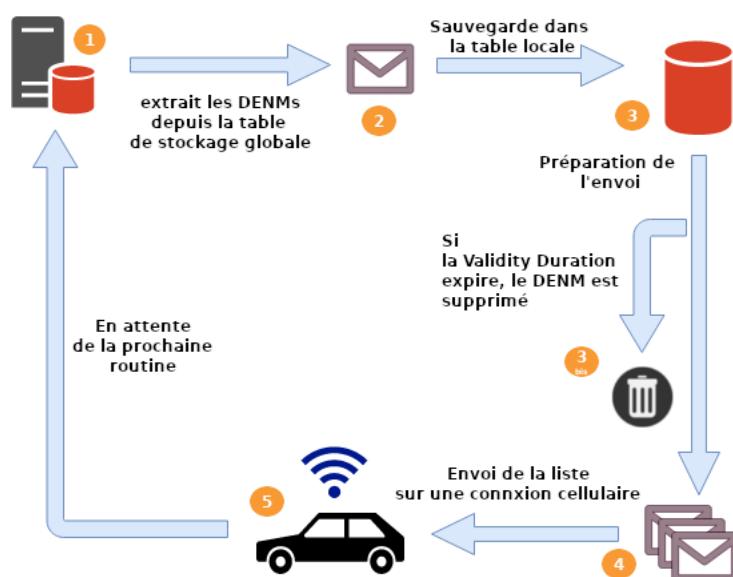


FIGURE 25 - ROUTINE D'ENVOI DES DENMS

La Figure 25 décrit les étapes suivantes :

- 1) 1-2-3 : La routine extrait de la base de données commune, les DENMs qui intéressent le client et qui n'ont pas encore été traités pour celui-ci, soit parce que le client a changé de position, soit parce qu'un DENM a été reçu dans la zone de celui-ci.
- 2) 3bis-4 : Si de nouveaux DENMs ont été trouvés, et après un test de validité, ils sont envoyés aux clients et marqués comme « envoyé » au niveau du manager.

Avant de stocker un événement dans la base de données, nous déterminons sa position OSM (OpenStreetMap), dérivée de la *destinationArea* indiquée dans l'en-tête GeoNetworking. Nous appelons position OSM, l'ensemble des tuiles calculées en suivant la formule indiquée par OpenStreetMap.

La position OpenStreetMap est déterminée par trois coordonnées qui définissent une tuile :

- 1) X, Y : Coordonnées cartésiennes classiques
- 2) Z : Le niveau de zoom (de 1 à 18)

On cherche la tuile, la plus petite possible, contenant toute la zone de destination du DENM.

Par exemple, pour un DENM situé aux coordonnées gps suivant : 492508490 ; 40216230 et une zone en forme de cercle de rayon 1km, nous obtenons la tuile suivante : 13/4187/2804 (z/x/y)

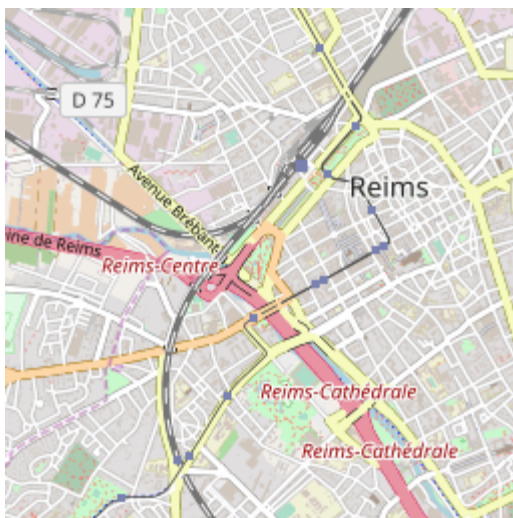


FIGURE 26 - EXTRAIT OPEN STREET MAP



FIGURE 27 - ZONE DE DESTINATION D'UN DENM

La Figure 26 est un extrait de la cartographie Open Street Map représentant la tuile citée précédemment. Dans la Figure 27, nous avons schématisé la zone de destination d'un DENM à l'intérieur de cette même tuile.

Tout véhicule qui entrera dans cette tuile recevra les DENMS qui s'y trouvent même si le véhicule ne se trouve pas dans la *destinationArea*.

L'algorithme suivant d'écrit le traitement d'un DENM entrant :

```

1: WHILE (RECEIVE(denm)) DO
2:   northPoint ← getNorthOfDestinationArea(denm);
3:   eastPoint ← getEastOfDestinationArea(denm);
4:   southPoint ← getSouthOfDestinationArea(denm);
5:   westPoint ← getWestOfDestinationArea(denm);
6:   FOR (zoom = 18; zoom < 0; zoom - -) DO
7:     northOSM ← OsmPosFromGPSPoint(northPoint, zoom);
9:     eastOSM ← OsmPosFromGPSPoint(eastPoint, zoom);
11:    southOSM ← OsmPosFromGPSPoint(southPoint, zoom);
13:    westOSM ← OsmPosFromGPSPoint(westPoint, zoom);
15:    IF (northOSM == eastOSM == southOSM == westOSM) THEN
16:      Store(zoom, northOSM, denm);
17:      UpdateServerVersionNumber();
18:    END IF
19:  END FOR
20: END WHILE

```

ALGORITHME 6 - SERVER NAAS - RECEPTION D'UN DENM

Dans l'Algorithme 6, les instructions 2 à 5 permettent d'extraire les coordonnées GPS de la zone de destination, à chaque point cardinal. Ensuite, nous transformons chacune de ces coordonnées GPS en tuiles OSM (instructions 7 à 13) et nous répétons l'opération pour les 18 niveaux de zoom (instruction 6). Quand nous trouvons un niveau de zoom où la transformation des 4 points aboutit à la même tuile (instruction 15), alors nous stockons le DENM en l'associant à celle-ci (instruction 16).

Ce système permet au client de recevoir l'information avant d'arriver dans la zone de destination et de l'afficher dans un délai plus court. D'autre part, cela permet d'avoir une méthode de positionnement indépendante du GeoNetworking et donc d'assurer une interopérabilité entre différentes approches. Par exemple, différents ITS-C peuvent s'échanger leurs DENMs en fonction de ces zones.

Dans cette architecture, le client ne diffère du client présenté dans la solution ITS-G5, que par l'existence d'une interface réseau supplémentaire. Ses traitements sont les mêmes, et ses messages sont les mêmes, mais il envoie et reçoit les messages sur le réseau cellulaire, en plus du canal G5. Il transmet des messages de type GeoNetworking à la station centrale. Cette dernière détermine s'il s'agit d'un CAM ou d'un DENM, chaque type de message bénéficiant d'un traitement différent. A la réception d'un CAM, la nouvelle position du client est sauvegardée, puis comparée avec l'ancienne. Si celle-ci a changé, l'opération d'envoi des DENMs est exécutée.

Si le message reçu est un DENM, alors le cloud calcule sa position OSM, puis le sauvegarde dans une base de données commune à tous les managers. Ces derniers sont alors avertis d'un changement, par le serveur, et exécutent l'opération d'envoi des DENMs.

2.2 Expérimentation

L'expérimentation s'est déroulée sur des routes de la périphérie de Reims. Pendant les trajets, des événements sont générés aléatoirement. Nous mesurons le temps nécessaire à un DENM pour partir du véhicule source et arriver au véhicule concerné.

Lorsqu'un DENM est reçu par le serveur il est immédiatement envoyé aux véhicules concernés, le temps moyen nécessaire à ce processus est d'un peu moins de 2ms. Notons tout de même que ce délai peut être allongé si la capacité de calcul du serveur n'est pas adaptée au nombre de véhicules.

La Figure 29, la Figure 28 et la Figure 30 présentent les relevés de la durée des échanges de DENM sur les 3 trajets.

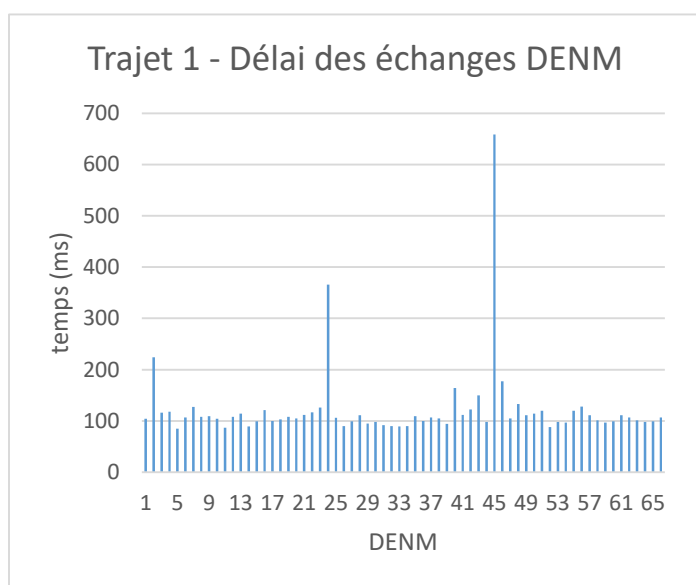


FIGURE 30 – TRAJET 1 – EXPERIMENTATION NAAS

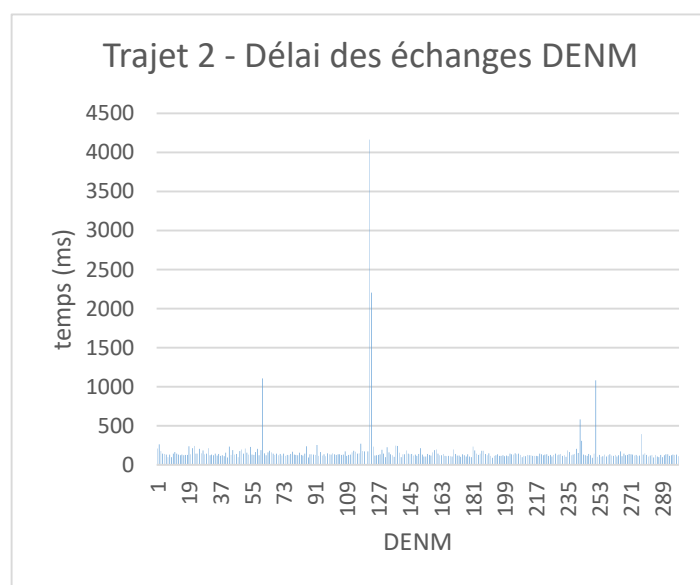


FIGURE 28 - TRAJET 2 - EXPERIMENTATION NAAS

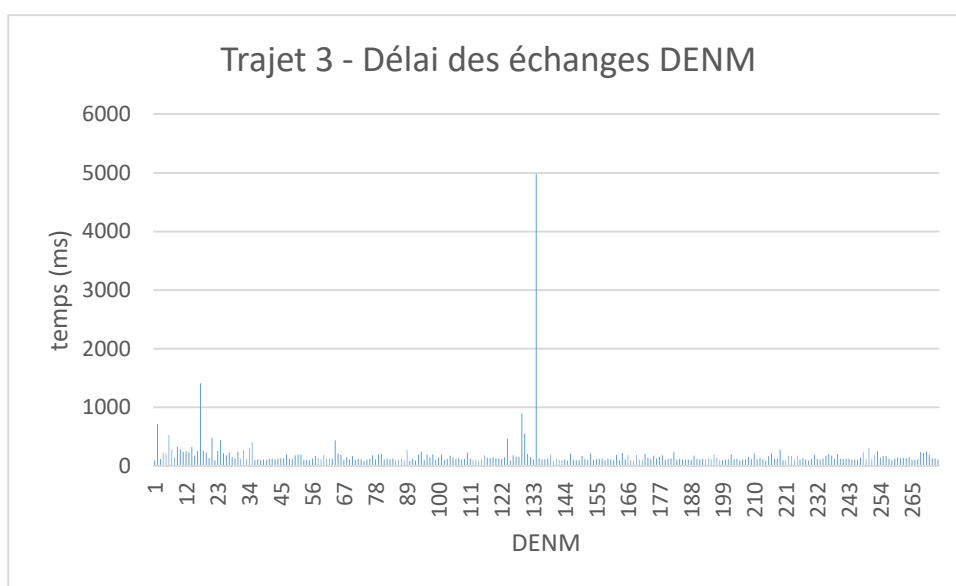


FIGURE 29 - TRAJET 3 - EXPRIMENTATION NAAS

A quelques exceptions près, dues aux pertes de connexions, la plupart des échanges se font en moins de 200 ms. Sur l'ensemble des trois trajets, nous obtenons un délai moyen de 169 ms. Et seulement 3.76% des échanges ont dépassé le seuil de 300 millisecondes.

III – Discussions

Dans un premier temps nous avons mis en évidence les problèmes de connectivité qu'il peut exister pour les ITS si nous nous limitons à la seule utilisation du canal G5. Puis nous avons proposé deux approches différentes, mais toutes deux basées sur le réseau cellulaire pour contrecarrer ce problème. Nous avons présenté des relevés de latence fait sur route et effectué des simulations basées, en partie, sur ces relevés. Nous allons maintenant résumer ces approches et apporter des points de comparaison avec l'ITS-G5.

3.1 SaaS

Dans cette approche, tous les calculs sont effectués sur le serveur. Le client ne fait qu'envoyer ses données et afficher les informations que le serveur génère. Les avantages de cette méthode sont nombreux. Le serveur possède une vision globale du trafic, tandis que le véhicule n'en a qu'une vision locale. Le serveur est donc capable de détecter des événements avec plus de précision que les véhicules. Si nous mettons en place des mécanismes d'apprentissage liés à l'intelligence artificielle, le serveur a à disposition une base de données bien plus complète et une puissance de calcul illimitée qui pourrait lui permettre de détecter, par exemple, des bouchons plusieurs dizaines de minutes avant même qu'ils n'aient commencé à se former.

Dans notre implémentation, les requêtes sont faites à travers l'envoi de CAM, mais il pourrait en être autrement. Nous pourrions faire des requêtes plus élaborées qui ne concernent qu'un type d'application, ou une zone plus étendue que celle dans laquelle nous nous trouvons, etc. Nous avons là une solution qui permet de fournir les mêmes services que les ITS, et d'autres, sans affecter le fonctionnement nominal. La solution que nous proposons dans le chapitre suivant, s'adapterait parfaitement à cette architecture. Nous introduisons dans le CAM, l'identifiant d'une donnée souhaitée et le serveur renvoie cette donnée ou un ensemble de données.

L'un des désavantages de cette architecture n'est pas spécifique aux réseaux véhiculaires, mais plutôt lié à sa nature. Comme tout système centralisé, une panne au niveau du Cloud entraîne une interruption totale du service. L'autre désavantage est la façon de récupérer la donnée, par requête, sur Internet, ce qui implique un délai supplémentaire.

3.2 NaaS

Le serveur se comporte comme une station ITS, mais il est capable de s'adapter à plusieurs types d'implémentation. Cette solution propose plus de souplesse au client et ne nécessite qu'une seule adaptation : l'ajout d'une interface réseau. Le client fournit des données de position et des

événements. Le serveur traduit ces données dans son format et transmet les événements aux véhicules concernés.

Avec cette solution, nous améliorons les délais de communication en nous arrêtant à la couche transport. Les messages sont envoyés sans délai, dès la réception de ceux-ci. Le serveur se comporte comme un nœud du réseau véhiculaire, mais accessible en cellulaire plutôt qu'en G5.

Comme pour la solution SaaS, une panne au niveau du serveur entraînerait une interruption de service.

3.3 Global

3.3.1 Résumé

Pour résumer les différentes approches, nous rappelons les comportements de chacune à chaque étape du traitement d'un message :

Fonction	SaaS	NaaS	ITS-G5
Collecte des messages	Le serveur collecte tous les CAMs fournis directement par les véhicules	Le serveur collecte CAMs et DENMs fournis par les véhicules et peut collecter les messages d'autres serveurs, tels que ses homologues étrangers	Les UBR collectent les DENMs et les CAMs des véhicules et les remontent au CIT-S, sous forme agrégée pour les CAMs. Les véhicules reçoivent les messages des autres véhicules
Détection d'un événement	L'événement est détecté et généré par le serveur grâce aux informations collectées.	L'événement est généré par les véhicules, manuellement ou automatiquement	L'événement est généré par le véhicule grâce aux CAMs environnants et aux données de son bus CAN ou par un opérateur routier
Distribution d'un événement	L'événement est envoyé aux véhicules dans la zone de destination en réponse à une requête, selon une logique de pull.	L'événement est envoyé aux véhicules dans la zone de destination, dès réception de celui-ci, selon une logique de push.	L'événement est envoyé à l'UBR la plus proche de la zone de destination qui le diffuse ensuite à intervalle régulier. Les véhicules s'envoient les événements, en essayant d'atteindre la zone de destination, saut par saut.

3.3.2 Comparaison

Le format **SaaS** est celui qui répond le moins bien aux besoins. Il en résulte, un délai plus important, dû à la méthode de distribution et aux protocoles choisis pour communiquer. Et pourtant, son potentiel de calcul et de stockage ouvre la porte à certains services qui ne peuvent être fournis par les véhicules, notamment dans le domaine de l'intelligence artificielle.

Le format **NaaS** colle parfaitement aux besoins : combler les zones où les véhicules ne peuvent communiquer en G5 et permettre aux messages de voyager plus loin, sans aucune modification d'implémentation dans les véhicules. Des délais corrects, qui rentrent largement dans l'objectif des 300ms.

Pour les deux méthodes, elles permettent d'améliorer, de façon conséquente, la connectivité du réseau, mais elles restent faillibles aux aléas de la couverture cellulaire.

L'ITS-G5 reste le meilleur canal pour les messages d'urgences, des délais de l'ordre de la dizaine de millisecondes et une connectivité de 100% dans un rayon de 300 à 500 mètres, qui font de lui, le seul apte à être utilisé pour les cas critiques de sécurité routière (freinage d'urgence par exemple). Pour les cas non-critiques, de trafic routier (routage dynamique) ou de sécurité routière (véhicule à l'arrêt sur le bord de la route), les solutions Clouds assurent le travail, en moyenne plus efficacement que l'ITS-G5 : une moyenne de 169ms pour les solutions NaaS et SaaS contre une moyenne comprise entre 700 ms et 6,9 secondes pour la solution G5. Une hybridation de nos deux solutions (SaaS et NaaS) permettrait d'accomplir le travail de transfert de messages dans les meilleurs délais, tout en ajoutant les bénéfices que nous pouvons tirer de la puissance de calcul d'un data center (apprentissage par exemple).

Alors que sur le réseau routier extra urbain la problématique est la connectivité, sauf dans les bouchons, en ville le problème est inversé : la connectivité est assurée par la forte densité de véhicules mais elle entraîne une augmentation considérable de la charge réseau appliquée au canal G5. Dans le chapitre suivant, nous proposons une architecture ICN permettant de fournir de nouveaux services sans trop alourdir la charge sur ce canal. Dans cette architecture, le serveur central aura encore un tout autre rôle.

CHAPITRE 3 – Proposition d'une approche ICN dans une ville intelligente.

L’évolution des infrastructures routières n’arrive pas à suivre celle de la population et du nombre de véhicules sur la route, il faut donc fournir de nouvelles solutions : les ITS, le Cloud Computing, l’Internet des Objets et l’ICN sont autant de modèles, d’architectures, de paradigmes qui offrent des solutions pour améliorer la vie des usagers de la route. Dans ce chapitre, nous proposons un modèle de ville intelligente qui permet de distribuer des informations de tailles réduites, issues d’une grande quantité de capteurs qui se mettent à jour régulièrement, et qui associe les technologies déjà en place (ITS-G5, standards ETSI) avec ces nouvelles approches.

La circulation en ville est de plus en plus difficile car les routes sont surchargées, concentrée sur les axes principaux, et parfois les axes secondaires peuvent être sous-exploités. Les villes tentent de mettre place depuis longtemps des solutions pour remédier à ces problèmes, bien avant l’avènement des ITS, tels que les systèmes de feux intelligents ou la détection de congestion, par caméras ou boucles magnétiques, au niveau des carrefours. Ces solutions se basent sur une centralisation des données et tentent d’adapter dynamiquement (mise à jour des durées de feu), ou statiquement (remplacer un carrefour à feux, par un carrefour giratoire), les infrastructures pour fluidifier le trafic.

Le paradigme des Clouds véhiculaires, associé aux ITS semble une piste prometteuse. Une solution de ce type permettrait de collecter plus facilement les données de trafic et d’interagir, aussi, avec les conducteurs, plutôt que d’être limitée aux seules infrastructures.

Dans le chapitre précédent, nous avons mis en évidence la viabilité d’une solution Cloud localisée sur Internet pour collecter et transmettre des informations routières aux véhicules intelligents. Il apparait aussi que, même si les délais sont très bons, ils ne le sont pas autant qu’en G5. Le problème de la connectivité du G5 qui existe sur autoroute semble beaucoup moins évident en ville et beaucoup moins critique. Par exemple, sur autoroute, la distance séparant deux sorties peut être très grande, si l’information ne nous arrive que quelques kilomètres avant un événement, il est impossible de changer de route. En ville, ce n’est pas le cas, si nous accédons à une information après avoir été isolé du réseau, nous pouvons nous réadapter facilement aux informations reçues.

Il existe différents types de Cloud, certains sont dynamiques, d’autres statiques. Certains sont fait pour les ITS, d’autres sont fait par les ITS. L’architecture que nous proposons est une association des différentes visions des Clouds véhiculaires, chacun ayant une fonctionnalité précise dans le schéma global. Avant d’introduire notre modèle, il convient de rappeler la définition des différents paradigmes que nous allons utiliser, l’architecture considérée pour une ville intelligente, puis de présenter deux approches représentatives des différentes propositions et leur comportement au sein de cette ville.

I – Définitions

Le premier paradigme a pour nom « Information Centric Networking » et est une architecture réseau dans laquelle les échanges sont tournés vers la donnée, plutôt que son créateur. Dans le réseau IP, nous cherchons à atteindre un hôte, dans le réseau ICN, nous cherchons à atteindre une donnée. Cette donnée est accessible grâce à un nommage, dont la méthode peut différer d’une implémentation à l’autre.

Les échanges se font en deux temps, la propagation d’un intérêt d’abord, qui se résume par l’envoi du nom de la donnée souhaitée dans le réseau, et le retour de la donnée ensuite. Il existe plusieurs façons de router un intérêt et une donnée (voir Chapitre 1).

A la différence des réseaux IP, le lien de communication entre deux nœuds n’est plus sécurisé. Dans l’architecture ICN, c’est la donnée elle-même qui est sécurisée (en authenticité et intégrité). Cette fois encore, il existe différentes méthodes présentées dans le Chapitre 1.

La mise en cache réseau est la principale fonctionnalité de l’ICN. Celle-ci permet aux équipements de conserver les données qu’ils ont routées. Ce mécanisme permet de fournir plus rapidement la donnée à une éventuelle deuxième requête. Il est aussi particulièrement efficace pour les réseaux mobiles, où les pertes de connexion sont plus fréquentes, et d’autant plus indiqué dans les réseaux véhiculaires, où les données ont une pertinence spatio-temporelle.

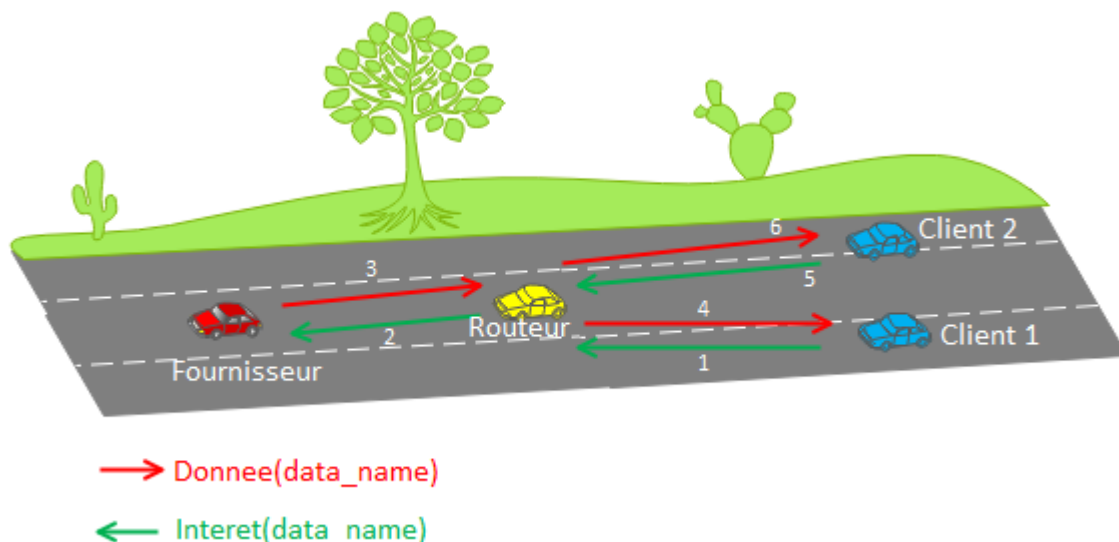


FIGURE 31 - ECHANGES ICN

Dans la Figure 31, le client cherche une donnée en envoyant un intérêt avec le nom de la donnée souhaitée (1), le routeur reçoit l’intérêt et le route vers le fournisseur (2). Le fournisseur répond avec la donnée au routeur (3), qui la met en cache et l’envoie au client 1 (4). Le client 2 envoie ensuite un intérêt pour la même donnée (5) et le routeur lui répond directement puisqu’il la possède en cache (5). Ainsi il ne perd pas de temps et de ressource à transmettre et répondre à la nouvelle demande.

Ensuite, rappelons que le Vehicular Cloud Computing consiste à exploiter les ressources sous-utilisées des véhicules (capteurs, calcul, stockage), ou, pour un véhicule, à utiliser un Cloud distant. C’est une forme dérivée du mobile Cloud Computing. La nouveauté réside dans la capacité pour un véhicule de produire un petit Cloud (parfois appelé Cloudlet [54]) avec ses voisins, afin de produire une donnée plus complexe qu’il n’est capable de faire tout seul, ou plus rapidement, ou de s’associer pour

alléger la quantité de données échangées avec les réseaux extérieurs. Ces Clouds sont rangés dans la catégorie dynamique en opposition au Cloud statique [55] qui désigne le processus de formation d’un Cloud avec des véhicules en stationnement, dans un parking par exemple, le transformant ainsi en mini data-center, accessible grâce à la virtualisation et une plateforme à distance [56] [57] [58]. Le cloud statique peut également désigner un cloud formé par des RSU [59].

L’association de l’ICN et du VCC nous amène au troisième paradigme : le Vehicular Cloud Networking. Dans celui-ci, les véhicules s’associent afin de produire une donnée riche, laquelle sera échangée, par la suite, selon les principes de l’ICN.

On distingue 6 étapes :

1) La découverte des voisins : elle peut se faire grâce à des beacons ou à la demande. Nous sélectionnons alors les voisins adaptés pour remplir notre tâche. Plus les beacons sont complexes, plus l’initiateur a d’informations pour sélectionner les voisins adéquats.

2) La formation du Cloud [60] : Une fois les voisins choisis, nous leur demandons de participer à notre Cloud. Ils peuvent refuser ou accepter.

3) La distribution des tâches et la collecte des résultats : une fois le Cloud formé, nous distribuons les tâches qui peuvent prendre de nombreuses formes, comme par exemple du calcul ou du partage de capteur... Ces tâches génèrent des résultats qui sont collectés par l’initiateur du Cloud.

4) La publication/partage des résultats : une fois les résultats collectés, l’initiateur peut décider de publier (au sens ICN) sa nouvelle donnée pour la mettre à disposition des autres utilisateurs qui gagneraient du temps, ou même qui n’auraient pas les moyens de produire cette donnée.

5) La maintenance du Cloud : étant donné le caractère mobile des nœuds qui forment le Cloud, certains peuvent disparaître, l’initiateur doit alors se charger de les remplacer, ou de redistribuer leurs tâches.

6) La libération des voisins : quand l’initiateur n’a plus besoin des ressources de ses voisins, il les libère, afin qu’ils puissent mettre leurs ressources à disposition d’autres nœuds.

Nous venons de rappeler les caractéristiques de l’ICN et du VCC. Puis nous avons présenté le paradigme VCN qui découle de l’association des deux. L’architecture de la ville intelligente que nous présentons dans la section suivante est représentée sous la forme d’un cloud de contexte, à l’intérieur duquel les échanges se font en ICN. Dans ce sens notre architecture s’articule comme étant un VCN.

II – Motivations et présentation de notre ville intelligente

Instinctivement, nous associons la gestion de trafic en ville aux carrefours. L’idée principale est de voir une ville comme un Cloud de contexte. C’est-à-dire que n’importe quel ITS rejoignant ce Cloud va automatiquement bénéficier du partage d’informations de celui-ci.

Dans certaines villes, la dynamique des infrastructures routières existe déjà, mais elle est opaque pour les conducteurs. Le Cloud véhiculaire va permettre aux véhicules d’être informés des changements en temps réel mais aussi d’anticiper quelques dizaines de secondes à l’avance une éventuelle congestion à un feu.

Nous nous intéressons aux carrefours à feux parce qu'ils régulent entièrement le trafic d'une ville, ils sont placés sur les axes principaux et voient passer la majorité du trafic. Pour que les véhicules puissent améliorer leur choix de route, ils ont besoin des informations en temps réel issues de ces carrefours.

Pour l'instant, il existe deux types de messages, au niveau européen, pour informer un véhicule intelligent de la topologie du carrefour (message MAP) et de l'état des feux du carrefour (message SPAT). Ces messages présentent deux défauts majeurs :

- Le fait d'exister en tant que message indépendant
- Leur incapacité à parcourir de grandes distances

Les messages MAP et SPAT [61] sont encapsulés au niveau transport en GeoNetworking de type SHB (Simple Hop Broadcast). Cela signifie que ces messages sont générés par le carrefour puis envoyés aux seuls véhicules à portée. En amont, si un véhicule veut réagir, comprendre ici, éviter un carrefour, il a besoin de l'information le plus tôt possible.

Pour exemple voici la configuration suivante :

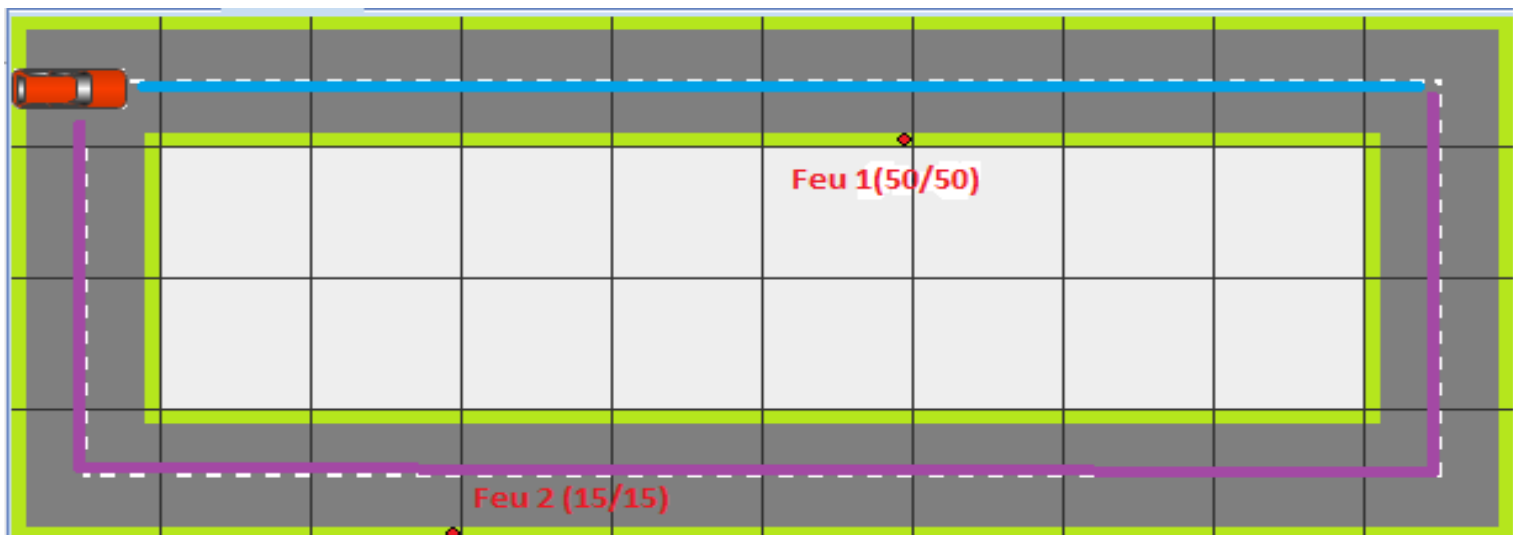


FIGURE 32 - CHOIX DE CHEMIN 1

Pour illustrer nos propos, prenons l'exemple d'une voiture qui a le choix entre deux chemins, comme le montre la Figure 32. Sur cette figure, le Feu1 passe 50 secondes au vert et 50 secondes au rouge, tandis que le Feu2 y passe 15 secondes.

Sur 300 secondes, le feu 1 a passé 150s au vert et 150s au rouge. Le feu 2 présente les mêmes temps. En revanche, en considérant une circulation fluide sur les deux routes, il vaut mieux choisir le chemin violet.

En effet, les chances d'arriver au vert à un feu sont les mêmes que d'y arriver au rouge, puisque le temps que passe un feu à chaque couleur est le même. Quand un véhicule arrive au feu et que celui-ci est au rouge, chaque durée d'attente comprise entre 1 et la durée maximale est équiprobable. Ainsi

si la durée maximum du Feu1 est de 50 secondes, un véhicule à l’arrêt à ce feu a plus de chance d’y rester plus de 15 secondes que moins de 15 secondes. Au contraire, un véhicule à l’arrêt au Feu2 est sûr d’y passer moins de 15 secondes, puisque c’est sa durée maximale.

On cherche $A(F)$ qui est le temps moyen qu’un véhicule passe à l’arrêt au feu F . Le temps que F passe au rouge est donné par D_R et celui qu’il passe au vert par D_V . Puisque les chances sont les mêmes pour chaque temps d’attente $A(F) = \frac{1+2+\dots+D_R}{D_V+D_R}$

Nous définissons α comme étant le temps supplémentaire qu’engendre un chemin par rapport à un autre à cause de la différence de distance (donc sans prendre en compte des éventuels feux tricolores).

Dans la Figure 32, Si $A(F1) - A(F2) > \alpha$, alors il vaut mieux prendre le chemin violet.

Afin de vérifier cette assertion, nous avons simulé deux véhicules qui prennent chacun l’un des deux chemins.

Au début de chaque simulation, la couleur du feu est initialisée aléatoirement à chance égale. Au début de la simulation, nous considérons que le feu est dans son état depuis un certain temps, le temps que passe un feu dans sa couleur initiale est donc compris entre 0 et sa durée maximum. Par exemple, le Feu1 est initialisé à 50 secondes auquel nous soustrayons une valeur aléatoire entre 0 et 50.

Sur 100 exécutions, le véhicule empruntant le chemin bleu est arrivé avant celui empruntant le chemin violet, un total de 61 fois. Cependant, sur les 100 exécutions, il a passé 21% de temps de plus sur la route que le véhicule ayant emprunté le chemin violet. Il est donc important pour qu’un véhicule puisse choisir son chemin d’avoir plus d’informations que la seule longueur de la route. Ici, un feu transforme un chemin plus long en termes de distance, en un chemin plus court en termes de temps.

Les caractéristiques d’un feu peuvent évoluer. Par exemple ici (Figure 33), le Feu 1 reste au rouge 15 secondes au lieu des 50 initiales.

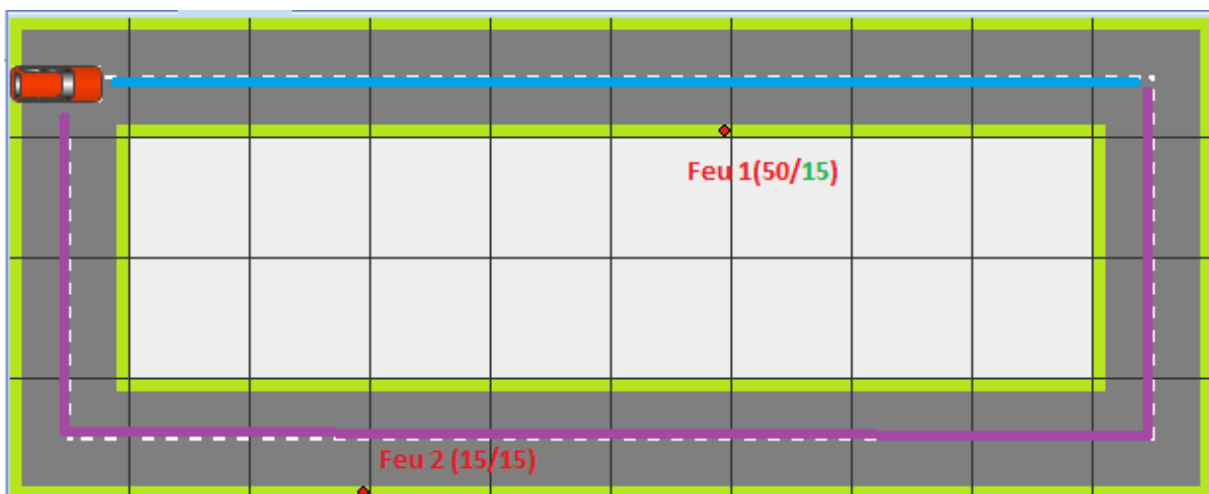


FIGURE 33 - CHOIX DE CHEMIN 2

II – Motivations et présentation de notre ville intelligente

Dans cette configuration (Figure 33), il est plus intéressant de passer par le chemin bleu que celui violet. Il est donc opportun de prévenir le véhicule de ces mises à jour pour qu'il puisse adapter sa conduite.

Pour faire parvenir l'information de ces changements à des véhicules lointains, nous choisissons l'approche ICN en intégrant les données des carrefours.

On généralise ainsi, la motivation de faire parvenir les informations d'un SPAT ou d'un MAP à des véhicules hors de portée du carrefour, par le fait de faire parvenir n'importe quelle donnée aux plus grand nombre de véhicules possibles. Cela, en ayant le plus petit impact possible sur le volume d'informations qui transitent sur le réseau.

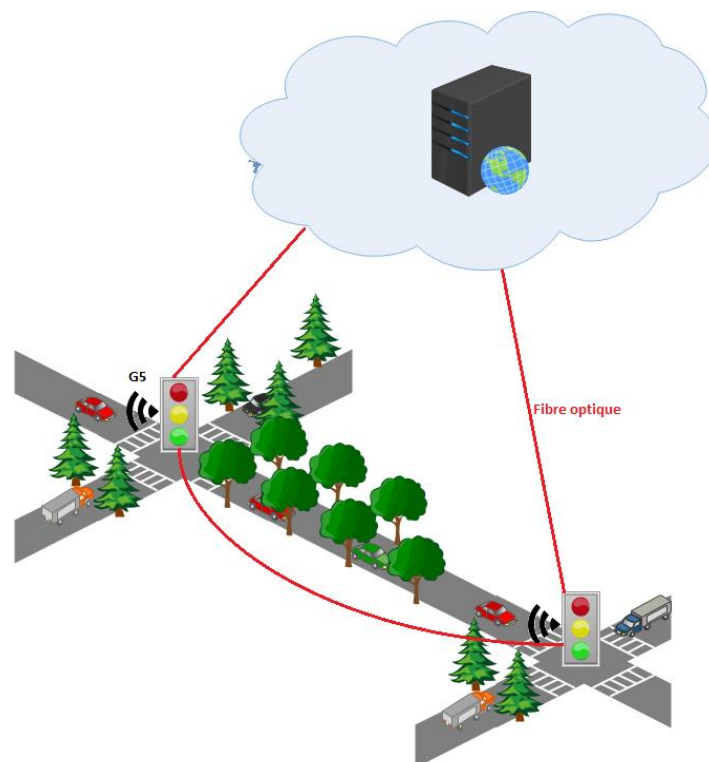


FIGURE 34 - COMMUNICATION DANS NOTRE MODELE DE VILLE INTELLIGENTE

La Figure 34 ci-dessus, représente les différents liens de communication qui existent dans notre vision de la ville intelligente. Ce modèle est un Cloud de contexte, le contexte étant la ville elle-même. Toute personne présente dans la ville, partage et consomme des informations la concernant.

Les véhicules peuvent communiquer avec les feux (V2I) et les autres véhicules (V2V) grâce à des modules G5. Les feux sont interconnectés entre eux en filaire. Ils sont, également, reliés au ITS-C (Intelligent Transport System-Central) en filaire.

Dans la section suivante nous allons présenter deux approches ICN qui font office de représentantes de deux catégories qui se distinguent par leur méthode de récupération de la donnée, à savoir « push » ou « pull ». Pour chacune d'elle, nous présenterons également leur implémentation dans le modèle de ville intelligente que nous venons d'énoncer.

III – Présentation d’approches ICN existantes

Dans cette architecture de ville intelligente, la donnée produite par le « publieur » (un capteur) est automatiquement envoyée aux routeurs que sont les nœuds fixes du Cloud.

Cette donnée est ensuite récupérée par les nœuds mobiles et la stratégie d’accès à la donnée diffère d’un algorithme à l’autre.

Il existe beaucoup de propositions d’implémentation de l’ICN qui se distinguent principalement selon leurs techniques de routage ou de nommage. Dans notre cas, le routage est simplifié. Ce qui nous intéresse c’est la méthode de requête. Pour cela, nous nous concentrons sur deux approches, ou plutôt deux familles d’approches : NDN (Named Data Network [62]) comme méthode de pull et RENE (REndez-vous NEtwork [63]) comme méthode de push.

La fréquence aléatoire de changement de caractéristiques d’un feu, et intrinsèquement de la production d’une donnée nouvelle, couplée à l’utilisation non ponctuelle de cette dernière, semble instinctivement causer des problèmes.

En méthode pull, nous devons régulièrement vérifier que la donnée n’a pas été mise à jour, assez souvent pour être à jour et pas trop souvent pour ne pas surcharger le réseau.

En méthode push, nous diminuons drastiquement le nombre de messages inutiles. D’un autre côté, nous perdons la faculté de faire d’un client un fournisseur de la donnée, et nous devons, soit avoir une connexion cellulaire vers un serveur central, soit être à la portée d’une UBR pour qu’elle puisse indiquer notre position à un serveur central.

1. NDN (pull)

Dans le modèle NDN [64] [65], une information est générée par un capteur. Certains nœuds du réseau appelé RH, pour Resolution Handler, sont aptes à router une requête ou une donnée dans le réseau.

La requête est routée grâce à son nom et la source répond sur le chemin inverse ou directement au concerné par un autre chemin ou un autre canal. La donnée est mise en cache sur chacun des routeurs par lesquels la réponse passe.

Une implémentation de l’approche NDN dans notre Cloud de contexte aura les caractéristiques suivantes

- **Nommage :**

CC:/Capteur/Donnée, par exemple : CC:/Feux46/État

- **Publieur :**

Chaque capteur (ici les feux), produit et publie des données.

- **Resolution handler :**

Les Resolution Handler, qui sont des routeurs, sont les infrastructures de routes (ici les feux aussi).

- **Sécurité**

La méthode de Certificat/Signature est utilisée.

- **Requête**

Un véhicule effectue une requête pour chaque donnée qu’il désire et devra recommencer à intervalle régulier pour s’assurer que celle-ci est à jour. Chaque requête et chaque réponse est un message qui doit être signé et accompagné d’un certificat.

Dans cette approche nous occupons la bande passante avec des requêtes lourdes et nous récupérons la donnée à chaque requête, qu’elle ait été mise à jour ou non. Le nombre de paquets inutiles (dont personne n’aura l’usage) sur le réseau est élevé.

La Figure 35, illustre les différents échanges qui ont lieu en utilisant l’approche NDN, dans notre modèle de ville intelligente.

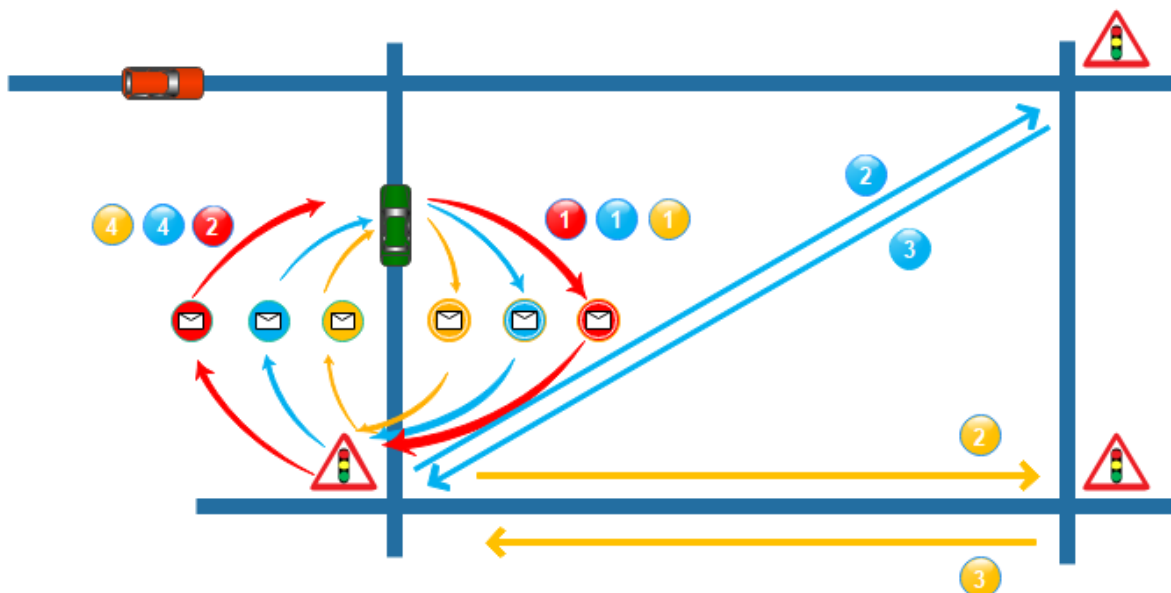


FIGURE 35 – ECHANGES DE MESSAGE EN NDN

Dans un premier temps, un véhicule envoie un intérêt pour chacun des capteurs dont il a besoin (1). Puis, le feu qui reçoit les intérêts répond avec son information, en tant que capteur lui-même, et route les autres intérêts vers les feux concernés, cette fois en tant que Resolution Handler (2). Chaque capteur sollicité, répond au RH avec sa donnée (3). Le feu qui fait office de RH peut alors transmettre les données des autres feux au véhicule (4). Ainsi, ces opérations sont répétées à fréquence régulière et assez élevée.

2. RENE (push)

RENE, pour REndez-vous NEtwork, apporte une vision intéressante de l’ICN pour les réseaux véhiculaires, principalement pour le mode push. Contrairement au modèle NDN, le véhicule ne va pas chercher l’information, nous la lui apportons quand elle est disponible. Sans les messages d’intérêts régulièrement envoyés, nous gagnons en charge réseau sur le canal G5. Avec le push de l’information quand elle est disponible, nous gagnons en termes de latence. Cependant, le véhicule doit être à la portée d’un routeur au moment où l’information est générée, sinon il n’en bénéficiera pas

La Figure 36 illustre les échanges qui ont lieu en utilisant une approche de type RENE, dans notre modèle de ville intelligente.

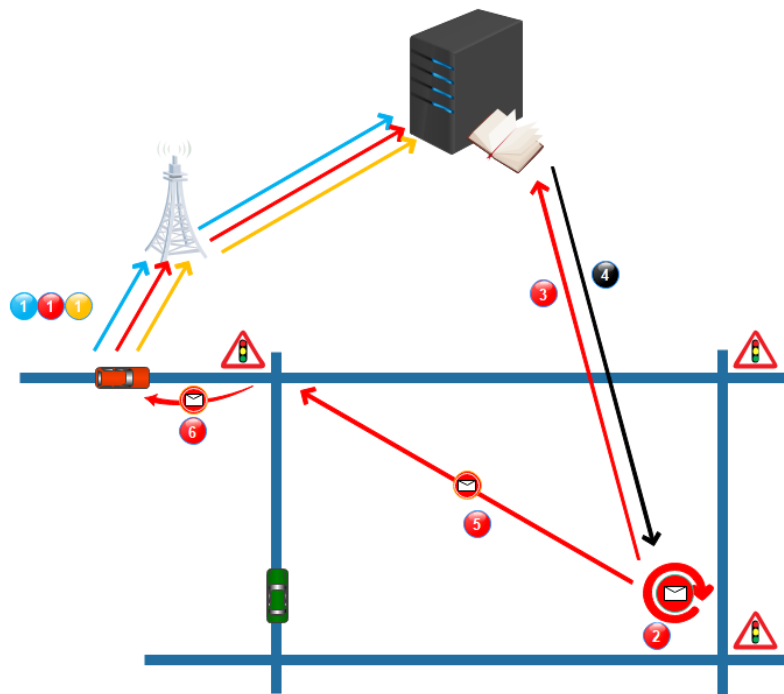


FIGURE 36 – PUBLICATION D’UN MESSAGE DANS UNE ARCHITECTURE RENDEZ-VOUS

Dans un premier temps (1), un véhicule s’abonne aux capteurs qui l’intéressent, auprès d’un serveur. Ensuite, si un des capteurs génère une nouvelle donnée (2), ce dernier fait ensuite une requête pour obtenir la liste des routes dont il a besoin pour atteindre les véhicules qui lui sont abonnés (3). Par la suite, le serveur lui répond avec ces informations (4). Le capteur peut alors transmettre sa donnée aux routeurs adéquats (5), qui, à leur tour, peuvent transmettre la donnée aux véhicules concernés (6).

IV – Le modèle proposé: GeoVehicular Central Data Networking (GeoVCDN)

Les approches ICN ont principalement été pensées pour améliorer le trafic à l'intérieur d'un réseau, grâce au cache réseau en grande partie. Le lien final vers l'utilisateur a souvent été ignoré et dans le cas des réseaux véhiculaires, les articles de recherche se concentrent en grande partie sur la mobilité et non pas sur les technologies de communication en place. Notre approche est une proposition qui tend à exploiter les standards ETSI pour les ITS afin d'optimiser la bande passante occupée sur le canal G5.

L'approche que nous proposons se différencie des deux autres, de trois façons. D'abord, elle regroupe un ensemble de données en une seule. Ensuite, elle donne une version à la donnée. Pour finir, elle inclut les échanges (intérêts et données) dans une version étendue du CAM. Comme nous l'illustrons dans la Figure 37. Celle-ci, représente les couches ETSI pour les ITS qui ont été modifiées pour intégrer notre brique au niveau des messages CAMs.

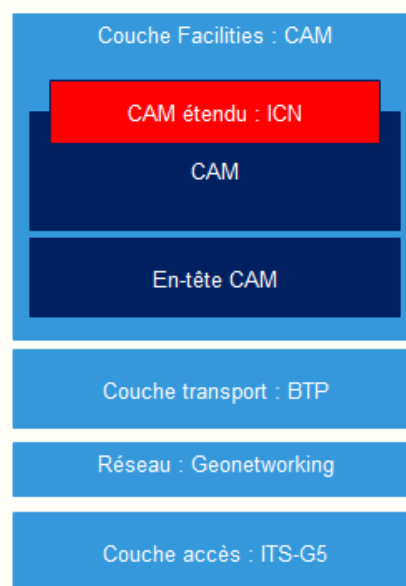


FIGURE 37 – PAQUET GEOVEHICULAR CENTRAL DATA NETWORKING

1. Métadonnée avec version

Dans notre approche, nous proposons de fusionner un ensemble de données en une seule. Les capteurs de même nature et d'un même contexte, envoient leurs relevés à une station centrale qui produit une donnée contenant l'identifiant et l'information de chaque capteur. Chaque fois que le relevé d'un capteur évolue, le serveur en est informé, ce qui produit une nouvelle métadonnée, toujours avec le même identifiant mais avec un nouveau numéro de version différent ; puis, le serveur l'envoie aux routeurs. L'identifiant de la donnée correspond dans l'architecture ICN au nom de la donnée.

Pour produire un intérêt, un client indiquera donc l'identifiant de la donnée qu'il souhaite ainsi que le

numéro de version qu’il possède. Ce couple permettra aux routeurs d’identifier si le client a une version à jour et s’ils doivent, ou non, lui envoyer la donnée dont ils disposent.

En regroupant les données en une seule, nous réduisons le nombre d’intérêts que doit envoyer un client pour se tenir informé de chaque capteur. Cette approche est valable dès lors que la donnée n’est pas générée trop souvent et que le nombre de capteurs est élevé. Il existe un rapport permettant d’identifier s’il est préférable de regrouper plusieurs données en une seule qui est donné par la formule suivante :

DESIGNATION	DESCRIPTION	Valeur par défaut
C	Nombre de capteurs	
P_i	Poids de l’identifiant d’une donnée (en octets)	1
P_v	Poids de la version d’une donnée (en octets)	1
P_d	Poids de la donnée (en octets)	4
F_{cam}	Fréquence d’un CAM (en Hertz)	10
F_d	Fréquence à laquelle la donnée est mise à jour (en Hertz)	1
N_v	Nombre de véhicules à portée d’une UBR	

Si $C * F_{cam} * (P_i + P_v) * N_v > C * (P_i + P_v + P_d) * F_d$, alors il est préférable de concentrer les données en une seule. A gauche de l’inéquation est représenté le poids des intérêts générés chaque seconde, par tous les véhicules à portée d’une UBR, si l’on ne regroupe pas les données des capteurs en une seule. A droite de l’inéquation, c’est la taille de la donnée, elle en regroupe plusieurs, qui est envoyée sur le canal G5 par une UBR à la fréquence F_d .

Pour faciliter la compréhension, prenons par exemple $N_v = 10$, $P_d = 1$ et $C = 10$, alors :

$$\begin{cases} C * F_{cam} * (P_i + P_v) * N_v = 2000 \\ C * (P_i + P_v + P_d) * F_d = 60 \end{cases}$$

Dans ce système, il est préférable de regrouper les données en une seule. Maintenant, si $P_d = 100$ et $F_d = 10$, alors :

$$C * (P_i + P_v + P_d) * F_d = 10\,200$$

Cette fois, il est préférable de ne pas regrouper les données.

Le numéro de version permet quant à lui de ne pas envoyer la donnée si celle-ci n’a pas été mise à jour. Cela rajoute une donnée supplémentaire, qui est bénéfique dès lors que la donnée n’est pas mise à jour trop souvent.

Si $P_v * N_v * F_{cam} < (P_d + P_i) * C * F_{cam}$, alors la numéro de version permet un gain sur l’occupation de la bande passant du canal G5.

Pour aider à la compréhension, voici un exemple avec $P_v = 4$, $N_v = 20$ et $C = 10$ alors :

$$\begin{cases} P_v * N_v * F_{cam} = 800 \\ (P_d + P_i) * C * F_{cam} = 500 \end{cases}$$

Dans ce cas, le numéro de version ne permet aucun gain sur la bande passante, au contraire. Maintenant si $C = 50$, alors $(P_d + P_i) * C * F_d = 2500$, dans ce cas le numéro de version est bénéfique. Ces prérequis sont à prendre en compte pour déterminer, pour chaque application, quelle stratégie adopter en considérant les cas où la bande passante est la plus sollicitée (c'est-à-dire quand N_v est élevé).

2. GeoNetworking

L'utilisation du CAM pour échanger en ICN est le pilier de notre proposition. Il y a une conséquence négative qui est liée à la fréquence d'envoi du CAM. Nous ne pouvons pas envoyer une donnée ICN à l'instant voulu, nous sommes obligés d'attendre que le prochain CAM soit envoyé. Cela dit à une fréquence de 10Hz (un CAM toutes les 100ms), cette contrainte nous paraît acceptable comparée aux bénéfices qu'elle apporte. Le canal G5 possède une faible bande passante, si nous multiplions les services qui transitent sur celui-ci nous arriverons vite à saturation, entraînant des délais de communication de plus en plus long. En intégrant les échanges dans les CAMs, nous profitons de sa couche sécurité, une couche indispensable mais très lourde (plus de 200 octets pour un CAM de 80 octets). Dans des réseaux de capteurs les données sont légères et multiples et nous ne pouvons nous permettre d'encapsuler chacune d'entre elles dans une couche sécurité 10 fois plus lourde que la donnée elle-même.

Au niveau GeoNetworking, le CAM est un message SHB (Single Hop Broadcast), c'est-à-dire qu'il est envoyé au voisin et il n'est pas retransmis pour d'autres véhicules. Or, nous voudrions que les véhicules puissent jouer le rôle de routeur et délivrer l'information lorsqu'ils l'ont. Il convient donc d'assurer le système de multi-sauts au niveau « facilities ». Pour cela, un CAM étendu pour fournir des informations ICN sera identifié sur un autre sous-port BTP (Basic Transport Protocol).

Une donnée est versionnée par le serveur. Quand un véhicule utilise une donnée, il la présente dans son message CAM avec la version de sa donnée. Si un de ses voisins possède cette donnée avec une version plus récente, il l'envoie en broadcast. Quand un routeur reçoit une donnée par le réseau filaire, et s'il possède dans sa table un voisin ayant cet intérêt, il la diffuse immédiatement.

Les principaux avantages de ce modèle sont le push de l'information, le gain théorique sur le délai, la baisse de la charge réseau et aucune connexion Internet n'est requise chez le client.

Afin d'éviter à une donnée d'être transférée d'un client à l'autre alors qu'elle n'est plus à jour, le serveur central estime un nombre limité de sauts qu'une donnée peut faire en fonction de la fréquence à laquelle elle est générée. Dans notre étude, nous supposons que nous cherchons les données de tous les capteurs. En revanche, il est également possible, de chercher un ensemble plus restreint de données, par exemple celles qui concernent une zone géographique. Le nommage laisse une grande liberté sur le choix des données.

3. Application

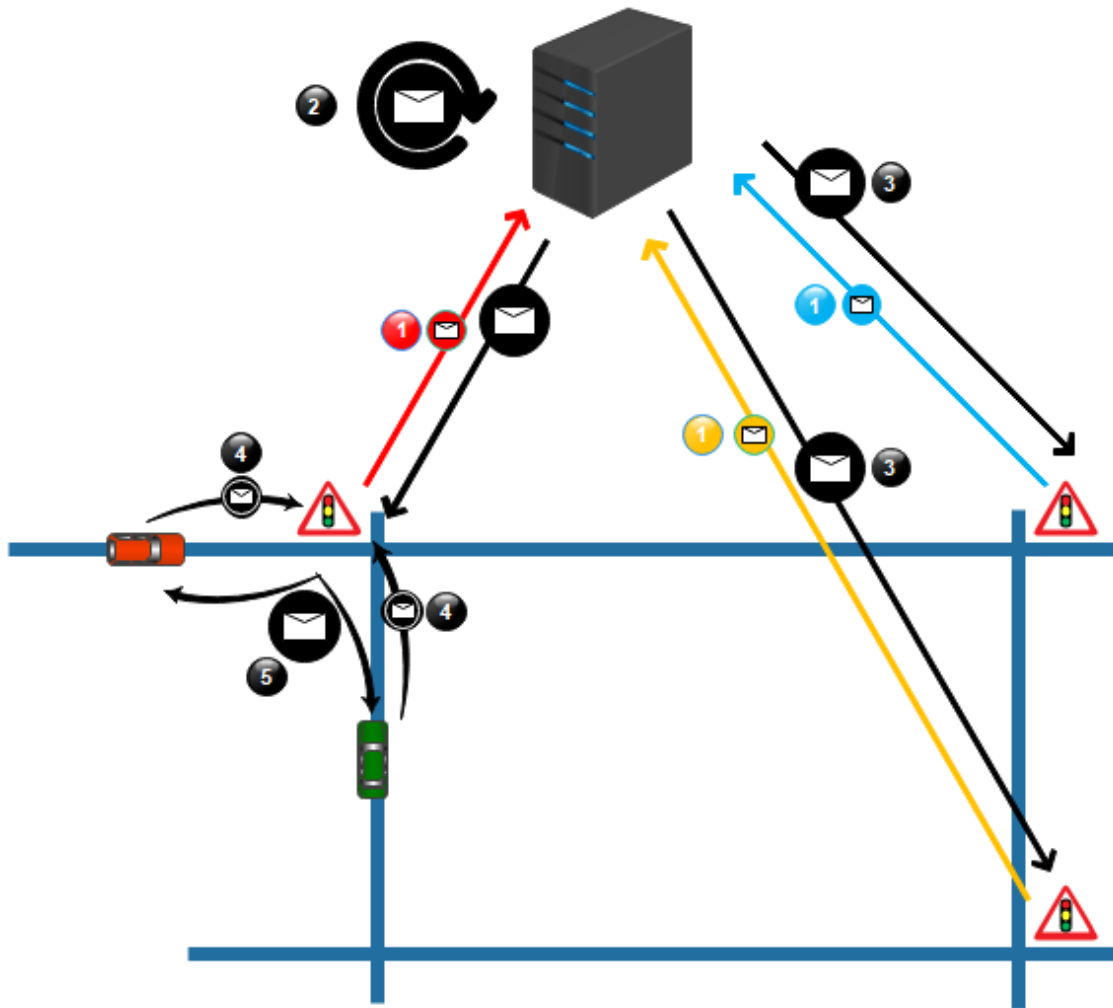


FIGURE 38 – ECHANGES DE MESSAGES EN GEOVEHICULAR CENTRAL DATA NETWORKING

La Figure 38 illustre les échanges qui ont lieu selon notre approche GeoVCDN dans notre modèle de ville intelligente. Dans un premier temps, chaque capteur transmet son état au serveur central (1). Le serveur central peut alors générer une métadonnée (2) qu’il transmet ensuite à tous les feux (3). Deux véhicules émettent un intérêt pour cette métadonnée (4) et le feu ne répond qu’une seule fois (5) pour l’envoyer aux deux véhicules.

V – Modélisation

Afin de comparer les différentes approches, il convient de donner une définition théorique de celles-ci. Ainsi, nous souhaitons comparer les différentes approches sur deux indicateurs de

performance : la charge et la fraîcheur du réseau. Nous entendons par fraîcheur, le taux de véhicules ayant obtenu l'information avant qu'elle ne devienne obsolète.

On définit différentes hypothèses de travail pour faciliter la modélisation mais qui n'auront théoriquement aucun impact sur la comparaison des modèles.

1. La portée des UBRs est inférieure à la taille des carrefours. Nous introduisons cette hypothèse pour ne pas avoir de zones de couverture qui se chevauchent.
2. On considère une ville qui respecte le modèle communément appelé « Manhattan ».
3. Les véhicules sont répartis de façon homogène et ils se déplacent à vitesse constante.

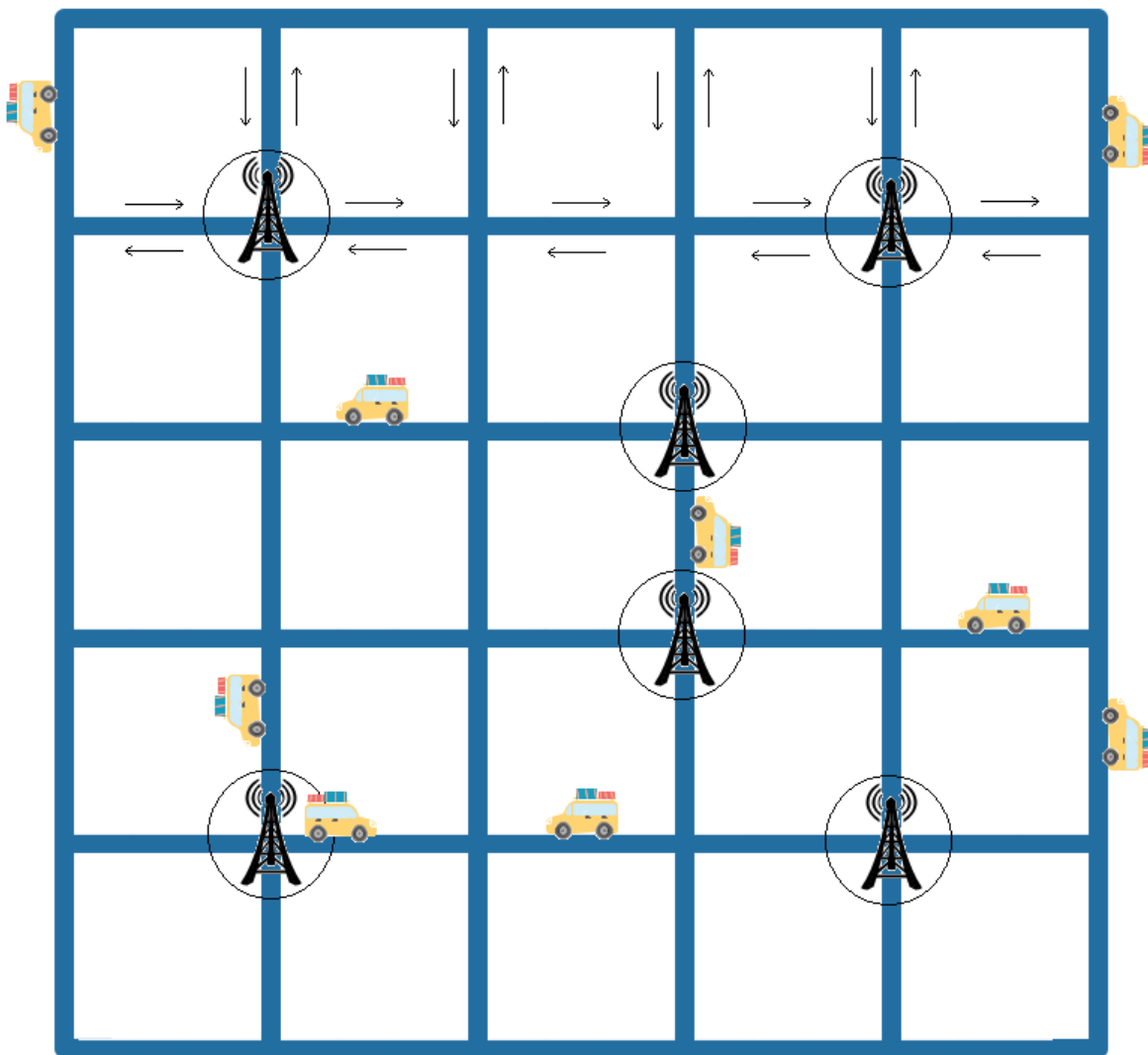


FIGURE 39 - ENVIRONNEMENT DE L'ETUDE

La Figure 39 ci-dessus représente un schéma de la ville. Y sont représentés : des UBRs placées de façon aléatoire et d'une portée inférieure à la taille d'un carrefour, ainsi que des véhicules se déplaçant dans les deux directions.

TABLEAU 6 - VARIABLES COMMUNES

DESIGNATION	DESCRIPTION	VALEUR
R_C	Rayon d’un carrefour	300
P_V	Portée de transmission d’un véhicule	200
P_{UBR}	Portée de transmission d’une infrastructure	250
V	Vitesse des véhicules en mètres par seconde	13
S	Taille dédiée à la sécurité d’un message	120
P_{IF}	Nombre d’octets dédiés à l’identification d’une donnée	4
P_{VF}	Nombre d’octets dédiés au versionnage d’une donnée	4
P_{DF}	Nombre d’octets dédiés à la donnée d’un feu	8
D	Dimension de la grille	
L_R	Longueur de route en mètres	
L_{UBR}	Longueur couverture UBR en mètres	
γ	Distance inter-véhicules en mètres	
N_S	Nombre de segments	
ρ	Densité des véhicules en veh/m	
ε	Nombre de carrefours	
δ	Pourcentage de couverture	
λ	Taux d’arrivée des véhicules en m/s	
F_{CAM}	Fréquence d’envoi d’un CAM en HZ	10
F_V	Fréquence d’arrivée d’un nouveau véhicule à portée d’une UBR	
α	Nombre de véhicules	
β	Nombre de feux	
σ	Fréquence de génération de la donnée en Hz	
φ	Nombre moyen d’entrées sur un carrefour	

TABLEAU 7 - VARIABLES SPECIFIQUES A CHAQUE APPROCHE

DESIGNATION	DESCRIPTION	NDN	RENE	GeoVCDN
PO _D	Poids d'une donnée			
PO _I	Poids d'un intérêt			
F _I	Fréquence d'envoi d'un intérêt en Hz	10		10
CR	Charge réseau en octets par seconde			
CR _D	Charge des données en octets par seconde			
CR _I	Charge des intérêts en octets par seconde			
Is	Intérêt par seconde sur tout le réseau			
α_D	Nombre de véhicules ayant reçu une donnée avant son expiration			

Pour comparer la charge réseau de chacune des approches, nous proposons une représentation théorique, faisant intervenir les différents paramètres du Tableau 6 et du Tableau 7.

Définition 1 : Pour mesurer le débit dédié aux échanges ICN, sur le canal G5, on exprime la charge réseau CR par la somme de la charge réseau dédiée à l'annonce des intérêts (CR_i) et de la charge réseau engendrée par l'envoi des données (CR_d)

$$CR = CR_i + CR_d$$

1. Définitions communes

Dans un modèle Manhattan, le nombre de carrefours ε est donné par la dimension D élevée au carré, on a donc l'expression suivante :

$$\varepsilon = D^2$$

D'abord, notons qu'il y a 4 coins qui ont deux routes. De plus, chaque bord de la ville (au nombre de 4) contient autant de cases que la dimension de la grille auquel nous retirons les coins, et chaque case possède 3 routes. Enfin, nous soustrayons 2 à la dimension de la grille (on retire les bords), qui, élevée au carré, nous permet d'obtenir le nombre de cases avec 4 routes.

Le nombre de kilomètres de route L_r est obtenu grâce au rayon des carrefours R_c . Il faut distinguer les carrefours ayant deux routes (les coins), ceux ayant trois routes (les bords), et les autres en ayant quatre.

$$L_r = 4 * R_c * 2 + 4(D - 2) * R_c * 3 + (D - 2)^2 * 4 * R_c$$

$$L_r = 4R_c(2 + 3(D - 2) + (D - 2)^2)$$

Nous identifions le pourcentage de chaque type de carrefour par les expressions suivantes :

$$\varphi_4 = \frac{(D-2)^2}{D^2} * 100, \text{ le nombre de carrefours à 4 entrées}$$

$$\varphi_3 = \frac{4(D-2)}{D^2} * 100, \text{ le nombre de carrefours à 3 entrées}$$

$$\varphi_2 = \frac{4}{D^2} * 100, \text{ le nombre de carrefours à 2 entrées}$$

Le nombre moyen φ de segments d’un carrefour s’exprime comme une moyenne du nombre de segments de chaque carrefour de la ville :

$$\varphi = \frac{(4 * 2) + (D - 2) * 4 * 3 + (D - 2)^2 * 4}{\varepsilon}$$

La longueur de route couverte par les UBRs s’exprime en fonction de la moyenne φ de segments d’une longueur P_{UBR} que nous multiplions par le nombre de d’UBR :

$$L_{RSU} = \varphi * P_{UBR} * \beta$$

La couverture des UBRs sur la carte correspond au pourcentage que représente la distance de route couverte sur la distance totale de la carte :

$$\delta = L_{UBR}/L_r * 100$$

Enfin, nous désignons par λ le taux d’arrivée de véhicules sur un segment.

Nous avons l’expression générale suivante qui permet de calculer la densité ρ de véhicules, avec la vitesse $V \in [a; b]$

$$\rho = \lambda \int_a^b \left(\frac{F(V)}{V} \right) dV$$

Ainsi nous voulons déterminer λ avec une vitesse V constante, donc :

$$\rho = \frac{\lambda}{V} \text{ et } \lambda = \rho * V$$

Définition 2 : On définit la fraîcheur du réseau (FR) comme étant le pourcentage de véhicules à avoir reçu une donnée avant que celle-ci n’expire et elle s’exprime de la façon suivante :

$$FR = \frac{\alpha_D}{\alpha} * 100$$

2. Définitions spécifiques à l'approche NDN

2.1 Charge des intérêts

Pour le NDN, la charge des intérêts par seconde se mesure par le nombre d'intérêts par seconde multiplié par le poids d'un intérêt, ce qui donne :

$$CR_i = PO_i * I_s$$

Dans le NDN, le poids d'un intérêt est la somme entre le nommage et la sécurité pour authentifier le paquet.

$$PO_i = S + P_{if}$$

Dans cette approche, chaque véhicule envoie ses intérêts pour chaque capteur, à une fréquence définie. Nous considérons qu'un véhicule n'envoie ses intérêts que s'il est à portée d'un RH. Ainsi, nous obtenons le nombre d'intérêts par seconde :

$$I_s = \beta \left(\frac{\alpha}{100} * \delta \right) * F_i$$

Enfin, nous aboutissons à l'expression définissant la charge réseau dédiée aux intérêts :

$$CR_i = (S + P_{if}) * \beta \left(\frac{\alpha}{100} * \delta \right) * F_i$$

2.2 Charge des données

A chaque intérêt reçu par un RH, celui-ci répond avec la donnée associée. Le nombre de données envoyées par seconde est donc égal à celui des intérêts envoyés, ce qui donne :

$$CR_d = I_s * PO_d$$

Le poids d'une donnée est la somme entre le poids de l'information d'un feu et la couche sécurité :

$$PO_d = S + P_{df} + P_{if}$$

On obtient donc l'expression suivante permettant d'évaluer la charge réseau dédiée au transit des données :

$$CR_d = \beta \left(\frac{\alpha}{100} * \delta \right) * F_i * (S + P_{df})$$

2.3 Charge réseau

L'expression générale suivante définit la charge réseau totale engendrée par l'approche NDN :

$$CR = (S + P_{if}) * \beta \left(\frac{\alpha}{100} * \delta \right) * F_i + \beta \left(\frac{\alpha}{100} * \delta \right) * F_i * (S + P_{df} + P_{if})$$

Qui nous donne :

$$CR = \beta \left(\frac{\alpha}{100} * \delta \right) * F_i * (2(S + P_{if}) + P_{df})$$

2.4 Fraîcheur réseau

Dans cette approche, le nombre de véhicule α_D représente le nombre de véhicules dans la zone de couverture, auxquels s’ajoutent les véhicules qui entrent à portée d’une UBR dans l’intervalle de génération de deux données. On a donc l’expression suivante :

$$\alpha_D = \left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\sigma}$$

Nous appliquons cette expression à la définition générale de la fraîcheur réseau, ce qui nous donne :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\sigma}}{\alpha} * 100$$

3. Définitions spécifique à l’approche RENE

3.1 Charge des intérêts

Le nombre d’intérêts par seconde envoyés dans une approche type RENE est relatif au taux d’arrivée des véhicules à portée d’une UBR, puisque ceux-ci ne sont envoyés qu’à l’entrée dans une zone couverte.

La charge des intérêts envoyés par seconde est donc liée au nombre de véhicules qui sont entrés dans une nouvelle zone couverte pendant cette seconde et au nombre de feux, puisque nous envoyons un intérêt par feu. Ainsi nous obtenons :

$$CR_i = PO_i * I_s$$

Avec le poids d’un intérêt qui est égal à la somme du poids de la couche sécurité et du poids de l’identifiant d’un feu : $PO_i = S + P_{if}$

Le nombre d’intérêts générés par seconde dans la toute la ville est obtenu en multipliant le nombre de feux par le nombre de véhicules qui arrivent à portée d’un feu à chaque seconde : $I_s = \beta * (\lambda * \varphi * \beta)$

La charge dédiée pour l’envoi des intérêts dans l’approche RENE est donc définie par l’expression suivante :

$$CR_i = (S + P_{if}) * \beta * (\lambda * \varphi * \beta)$$

3.2 Charge des données

La charge des données par seconde correspond au nombre de véhicules présents dans la zone au moment de la génération de la donnée, multipliée par la fréquence de génération d’une donnée et multipliée par le poids d’une donnée.

$$CR_d = \left(\frac{\alpha}{100} * \delta \right) * \sigma * PO_d$$

Avec le poids d'une donnée qui est égal à la somme du poids de la couche sécurité, du poids de la donnée d'un feu et du poids de l'identifiant d'un feu : $PO_d = S + P_{df} + P_{if}$

3.3 Charge réseau

D'après ce qui précède, la charge réseau dédiée aux échanges ICN avec une approche de type RENE est donnée par l'expression suivante :

$$CR = (S + P_{if}) * \beta * (\lambda * \varphi * \beta) + \left(\frac{\alpha}{100} * \delta\right) * \sigma * (S + P_{if} + P_{df})$$

3.4 Fraîcheur réseau

Dans cette approche, α_D est donnée par le nombre de véhicules présents dans la zone au moment de la génération de la donnée, soit l'expression suivante :

$$\alpha_D = \left(\frac{\alpha}{100} * \delta\right)$$

Que l'on applique à la définition générale de la fraîcheur réseau :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta\right)}{\alpha} * 100$$

4. Définitions spécifiques à nos approches VCDN & VCDNoCAM

Pour le moment, nous proposons de découper notre modèle en deux parties. La première étape consiste à fusionner les données de plusieurs capteurs de même type en une donnée commune et à lui appliquer une version, que nous nommons ici métadonnée. La deuxième étape est l'exploitation des standards ETSI, actuellement utilisés pour les communications véhiculaires en Europe, pour transmettre cette métadonnée et les intérêts liés à celle-ci.

4.1 Centralisation de la donnée (VCDN)

La centralisation de la donnée a deux effets positifs. Le premier étant de réduire la charge des intérêts, puisqu'au lieu que le nombre d'intérêts soit lié au nombre de capteurs, il est réduit à 1 pour un ensemble de capteur. Le deuxième permet de limiter l'utilisation de la couche sécurité sur les intérêts puisque nous ne sécurisons qu'un seul intérêt par ensemble de capteurs et une seule donnée par ensemble de capteur.

L'effet négatif est que désormais nous envoyons l'information de tous les capteurs même si seulement l'un d'entre eux a changé.

4.1.1 CHARGE DES INTERETS

On mesure donc la charge des intérêts en fonction du nombre de véhicules à portée d'UBR, de la fréquence d'envoi de ces intérêts et du poids de ceux-ci.

$$CR_i = \left(\frac{\alpha}{100} * \delta \right) * F_i * PO_i$$

Afin d’éviter de renvoyer inutilement une donnée, nous introduisons une notion de version dans l’intérêt.

Ainsi, l’intérêt est composé de l’identifiant de la donnée et de la version que le client possède actuellement.

$$PO_i = S + P_{if} + P_{vf}$$

4.1.2 CHARGE DES DONNEES

La charge de données échangées est mesurée en fonction du nombre de véhicules dans la zone de couverture ainsi que le nombre de véhicules qui rentrent dans la zone chaque seconde.

$$CR_d = \left(\frac{\alpha}{100} * \delta \right) * PO_D * \sigma + (\lambda * \varphi * \beta) * PO_d$$

La métadonnée est identifiée par un nom et une version, elle est composée de la valeur de l’ensemble des capteurs (ici, le nombre de feux) et encapsulée dans une couche sécurité, ce qui donne :

$$PO_D = P_{df} * \beta + S + P_{if} + P_{vf}$$

4.1.3 CHARGE RESEAU

Ainsi, si nous revenons à notre expression générale :

$$CR = \left(\frac{\alpha}{100} * \delta \right) * F_i * PO_i + \left(\frac{\alpha}{100} * \delta \right) * PO_d * \sigma + (\lambda * \varphi * \beta) * PO_d$$

4.1.4 FRAICHEUR RESEAU

Dans cette approche, α_D s’obtient en faisant la somme entre les véhicules présents dans la zone de couverture au moment où est générée la donnée et le nombre de véhicules rentrant dans la zone avant l’expiration de la donnée.

$$\alpha_D = \left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\sigma}$$

Que nous appliquons à la définition générale de la fraicheur réseau pour obtenir l’expression suivante :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\sigma}}{\alpha} * 100$$

4.2 Inclusion des échanges dans les CAMs (VCDN over CAM/GEONETWORKING)

En incluant les messages dans les CAMs, nous n’avons plus besoin d’une couche sécurité dédiée aux messages ICN et nous n’envoyons plus un message à un véhicule mais à l’ensemble des véhicules dans la zone, nous obtenons donc :

$$PO_i = P_{if} + P_{vf}$$

$$PO_D = P_{df} * \beta + P_{if} + P_{vf}$$

4.2.1 CHARGE DES INTERETS

L'expression de la charge d'intérêt évolue, la fréquence d'envoi d'un intérêt devient la fréquence d'envoi d'un CAM.

$$CR_i = \left(\frac{\alpha}{100} * \delta \right) * F_{CAM} * PO_i$$

4.2.2 CHARGE DES DONNEES

De plus, les données ne sont plus envoyées à un client mais à une zone, c'est le principe du routage géographique ; ici la zone autour d'une UBR. Par conséquent, la charge réseau consommée par les données, correspond à toutes les UBR (ici les feux) qui vont envoyer la donnée au moment de la génération.

$$CR_d = \beta * PO_D * \sigma$$

Tous les cycles de CAM (100 ms), si un véhicule (ou plus) est entré dans la zone, nous envoyons la donnée. En revanche, si aucun nouveau véhicule n'est arrivé dans la zone, nous n'envoyons pas de CAM. Nous calculons donc la fréquence d'entrée d'un véhicule dans une zone. Pour rappel, le renouvellement par seconde des véhicules dans la zone de couverture est obtenu grâce à λ , par l'opération suivante :

$$(\lambda * \varphi * \beta)$$

Nous la divisons ensuite par le nombre d'UBR pour savoir combien de véhicules entrent pour chacune d'elle.

$$F_v = \lambda * \varphi$$

On obtient ainsi le nombre de véhicules qui ne possèdent pas la donnée et qui rentrent dans une zone chaque seconde (F_v) ; que nous exprimons en fréquence (Hz). Si la fréquence est supérieure à 10, cela signifie que la donnée sera incluse dans chacun des CAM. Nous déterminons donc le nombre de CAMs qui contiendront la donnée de la façon suivante :

- 1) D'abord, nous évaluons le rapport fréquence CAM et fréquence d'entrée d'un nouveau véhicule.

$$v = F_{CAM} / F_v$$

Si v est inférieur à 1, c'est que les véhicules entrent dans la zone plus souvent que nous n'envoyons des CAMs. Par conséquent nous intégrons la donnée dans chacun des CAMs, mais nous ne devons pas dépasser 10.

- 2) En partant des constations suivantes : $\forall v < 1, \frac{[v]}{v} = 0, \left\lceil \frac{[v]}{v} \right\rceil = 0$
 et, $\forall v \geq 1, [v] \leq v, 0 < \frac{[v]}{v} \leq 1, \left\lceil \frac{[v]}{v} \right\rceil = 1$

C’est-à-dire que si v est inférieur à zéro, sa partie entière est égale à 0 et donc quel que soit son diviseur, le résultat donnera 0 et son arrondi supérieur conservera 0. Sinon nous obtiendrons 1.

- 3) Et nous généralisons par la fonction suivante, qui donne A si $A < B$ et B sinon :

$$\vartheta(A, B) = B - (B - A) \left\lceil \frac{[B/A]}{B/A} \right\rceil$$

Nous appliquons cette fonction pour déterminer le nombre de CAMs qui contiendront la donnée :

$$F_{CAM} - (F_{CAM} - F_v) \left\lceil \frac{[v]}{v} \right\rceil$$

Ainsi, nous obtenons :

$$CR_d = \beta * PO_D * \vartheta(\sigma, F_{CAM}) + \beta * \vartheta(F_v, F_{CAM}) * PO_D$$

4.2.3 CHARGE RESEAU

Pour une charge finale de :

$$CR = \left(\frac{\alpha}{100} * \delta \right) * F_{CAM} * (P_{if} + P_{vf}) + \beta * (P_{df} * \beta + P_{if} + P_{vf}) * (\vartheta(F_{CAM}, \sigma) + \vartheta(F_{CAM}, F_v))$$

4.2.4 FRAICHEUR RESEAU

Pour cette approche, α_D se calcule par la somme entre le nombre de véhicules présents dans la zone de couverture au moment où est générée la donnée et le nombre de véhicules rentrant dans la zone avant l’expiration de la donnée.

On a donc :

$$\alpha_D = \left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\vartheta(\sigma, F_{CAM})}$$

Nous appliquons cette expression à la définition générale de la fraîcheur réseau :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\vartheta(\sigma, F_{CAM})}}{\alpha} * 100$$

5. Définitions spécifiques à notre approche GeoVCDN

La prise en compte de la distribution des données par les clients, complexifie le modèle. D'abord, il faut considérer que cette fois TOUS les véhicules envoient un intérêt, à portée ou non d'une UBR. Ensuite, chaque seconde, des véhicules envoient la donnée à d'autres véhicules. Il faut donc estimer combien de véhicules vont envoyer la donnée chaque seconde. Pour cela, le placement des UBRs a une importance. Les figures suivantes (Figure 41 et Figure 40) illustrent la différence qu'il existe en fonction du placement des UBRs. Dans le cas de la deuxième figure (Figure 40) il faudra beaucoup plus d'échanges de véhicule à véhicule pour couvrir tout le réseau.

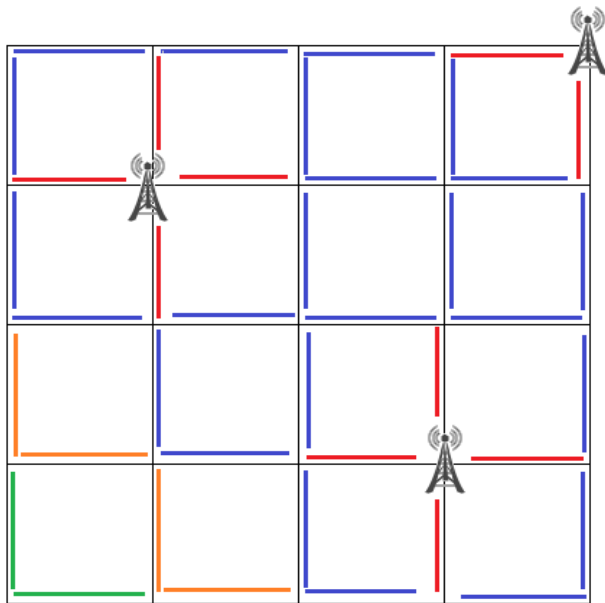


FIGURE 41 - EXEMPLE DE DISSEMINATION D'UN MESSAGE

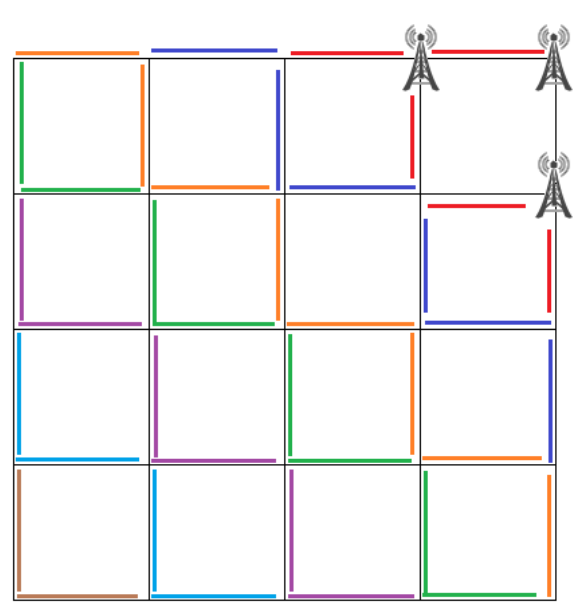


FIGURE 40 - EXEMPLE DU PIRE DES CAS

5.1 Segments

Le nombre de segments s'obtient en multipliant le nombre de carrefours par le nombre moyen d'entrées pour un carrefour, que nous divisons par deux car chaque segment est partagé entre 2 carrefours :

$$N_s = \varphi * \varepsilon / 2$$

5.1.1 CONFIGURATIONS

Dans un premier temps, pour simplifier la modélisation, nous sélectionnons un segment de la ville, composé de deux carrefours et de la route qui les joint. Puis, nous extrapolerons le résultat à la totalité de la ville.

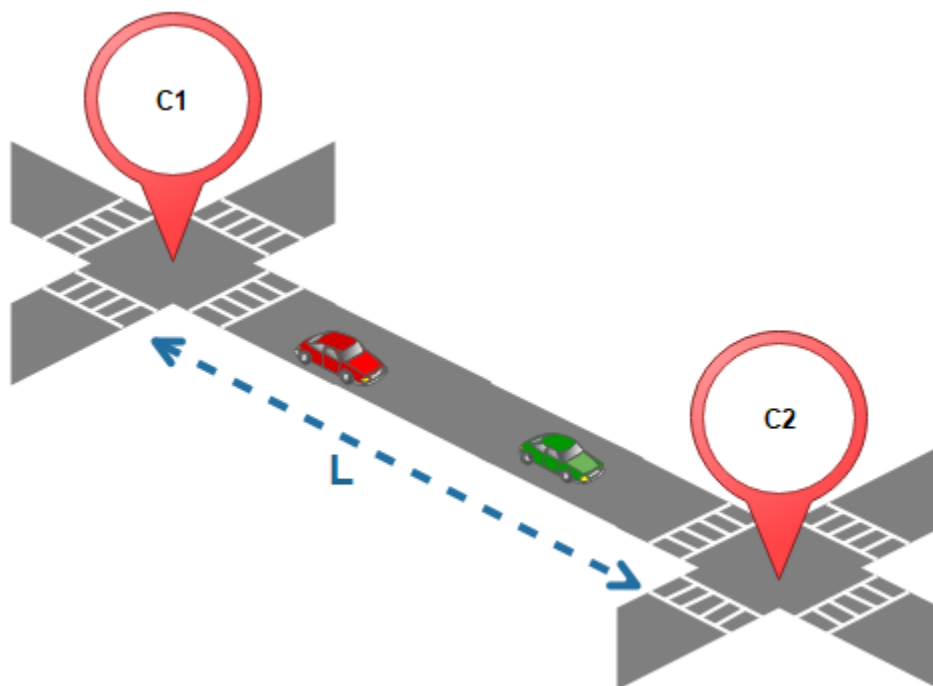


FIGURE 42 - ILLUSTRATION D'UN SEGMENT

La Figure 42 schématise la configuration générique d'un carrefour, avec $L = 2 * R_c$

Il existe, pour un segment, 4 configurations possibles :

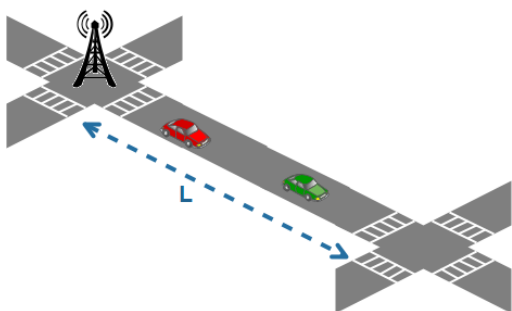


FIGURE 44 - CONFIGURATION 1

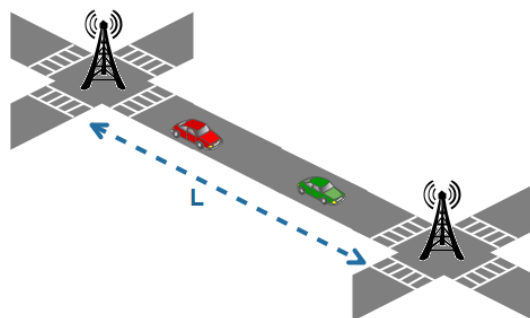


FIGURE 43 - CONFIGURATION 2

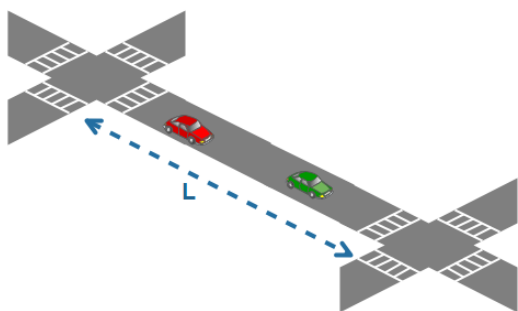


FIGURE 46 - CONFIGURATION 3

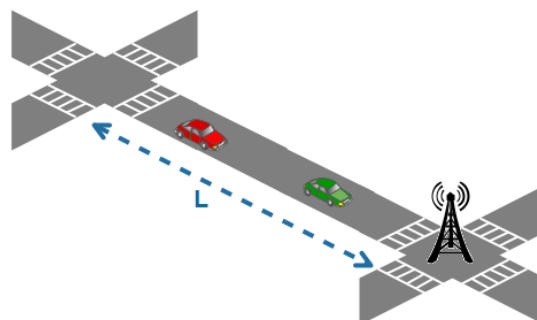


FIGURE 45 - CONFIGURATION 4

La Figure 44 et la Figure 45, représentent des carrefours avec une seule UBR. La Figure 46 représente un segment sans UBR, tandis que la Figure 43, représente un segment avec deux UBRs. Il est à noter ici, que les configurations 1 et 4 sont les mêmes.

5.1.1.1 PROBABILITE DE PRESENCE DES UBRs

On désigne par X_c la variable indiquant la présence ou non d'une UBR à un carrefour : $X_{c1} = 1$ s'il y a une UBR sur le carrefour 1 et $X_{c1} = 0$ sinon.

La probabilité que X_{c1} ait une UBR est la suivante : $P(X_{c1} = 1) = \beta/\varepsilon$, qui désigne le nombre de feux divisé par le nombre de carrefours

Et la probabilité que X_{c2} ait une UBR sachant que X_{c1} en a une, est de $P(X_{c2} = 1) = \beta - 1/\varepsilon - 1$. Nous soustrayons à la fois le feu que nous avons déjà placé et le nombre de carrefours restants.

La répartition des feux sur un segment, respecte la règle définie sur la Figure 47, avec K le nombre de feux et N le nombre de carrefours :

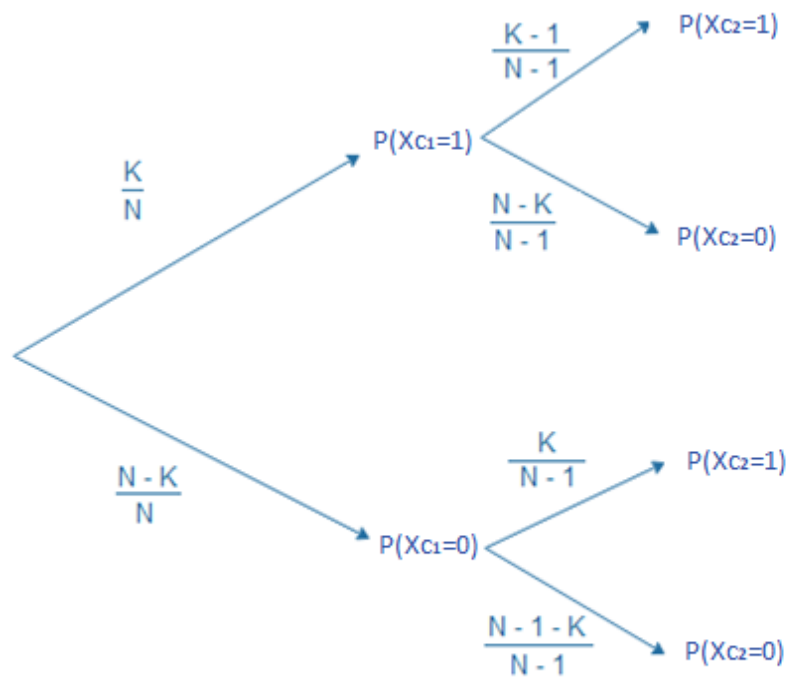


FIGURE 47 - LOI DE PROBABILITE DES FEUX

Ainsi, nous obtenons pour la configuration 2 :

$$P(X_{c1} = 1) = \beta/\varepsilon$$

$$P(X_{c2} = 0) = \beta - 1/\varepsilon - 1$$

Pour les configurations 1 et 4 :

$$P(X_{c1} = 1) = \beta / \varepsilon$$

$$P(X_{c2} = 0) = \varepsilon - \beta / \varepsilon - 1$$

Et pour la configuration 3 :

$$P(X_{c1} = 0) = \varepsilon - \beta / \varepsilon$$

$$P(X_{c2} = 0) = \varepsilon - \beta - 1 / \varepsilon - 1$$

5.1.1.2 POSITION DANS UN SEGMENT

Ensuite, nous désignons par P_r , la probabilité qu’un véhicule soit à la portée d’une UBR. De plus, nous désignons par P_{or} , la probabilité qu’un véhicule puisse être hors de la portée d’une UBR. Nous obtenons donc, pour la configuration 2 (avec 2 UBRs) :

La probabilité pour un véhicule de se trouver à portée d’une UBR est de deux fois P_{UBR} , la portée d’une UBR, divisé par la longueur du segment L, ce qui donne :

$$P_{r1} = \frac{2P_{UBR}}{L}$$

La probabilité de ne pas être à portée d’une UBR dans cette même configuration est donc donnée en soustrayant 2 fois la portée d’une UBR à la longueur du segment que nous divisons ensuite par cette même longueur L, nous obtenons :

$$P_{or1} = \frac{L - 2P_{UBR}}{L}$$

Ainsi, il en va de même pour les autres configurations en adaptant le facteur placé devant P_{UBR} en fonction du nombre d’UBR sur le segment.

Pour les configurations 1 et 4 (1 seule UBR) :

$$P_{r2} = \frac{P_{UBR}}{L}$$

$$P_{or2} = \frac{L - P_{UBR}}{L}$$

Pour la configuration 3 (Pas d’UBR) :

$$P_{r3} = 0$$

$$P_{or3} = 1$$

5.1.1.3 SAUTS POUR UN SEGMENT

Pour chacune des configurations, nous cherchons la valeur H qui est le nombre maximum de sauts nécessaires pour couvrir le segment et M qui est le nombre minimum de sauts nécessaires pour couvrir le segment. HM représente le nombre moyen de sauts nécessaires pour couvrir un segment.

On obtient pour chaque configuration les valeurs H, M et HM comme suit :

Pour la configuration 2 :

$$H_1 = \left\lceil \frac{(L - 2P_{UBR})}{2(P_V - \gamma)} \right\rceil$$

$$M_1 = \left\lceil \frac{(L - 2P_{UBR})}{2(P_V)} \right\rceil$$

$$HM_1 = \left(\frac{H_1 - M_1}{2} \right)$$

Pour les configurations 1 et 4 :

$$H_2 = \left\lceil \frac{(L - P_{UBR})}{(P_V - \gamma)} \right\rceil$$

$$M_2 = \left\lceil \frac{(L - P_{UBR})}{(P_V)} \right\rceil$$

$$HM_2 = \left(\frac{H_2 - M_2}{2} \right)$$

Et pour la configuration 3 :

$$H_3 = \left\lceil \frac{L}{(P_V - \gamma)} \right\rceil$$

$$M_3 = \left\lceil \frac{L}{P_V} \right\rceil$$

$$HM_3 = \left(\frac{H_3 - M_3}{2} \right)$$

5.1.1.4 NOMBRE SEGMENTS

Soit S_0 la probabilité qu'un segment n'ait pas d'UBR, S_1 la probabilité qu'un segment ait une seule UBR et S_2 la probabilité qu'un segment ait deux UBR ; ces probabilités sont calculées comme suit :

$$S_0 = \left(\frac{\varepsilon - \beta}{\varepsilon} \right) \left(\frac{\varepsilon - 1 - \beta}{\varepsilon - 1} \right)$$

$$S_1 = 2 \left(\frac{\varepsilon - \beta}{\varepsilon} \right) \left(\frac{\beta}{\varepsilon - 1} \right)$$

$$S_2 = \left(\frac{\beta}{\varepsilon} \right) \left(\frac{\beta - 1}{\varepsilon - 1} \right)$$

5.1.2 DISSEMINATION

Un segment peut être lié à 3, 4, 5 ou 6 segments comme suit :

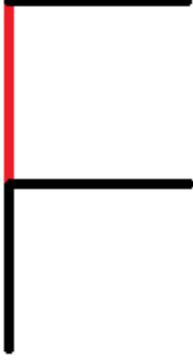


FIGURE 49 - SEGMENT A 3 ROUTES



FIGURE 48 - SEGMENT A 4 ROUTES

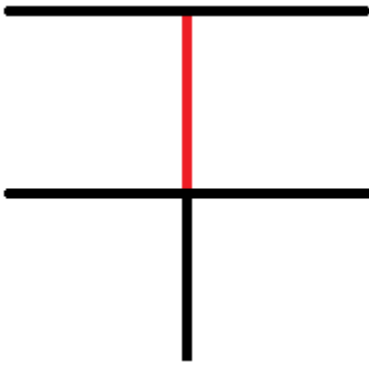


FIGURE 50 - SEGMENT A 5 ROUTES

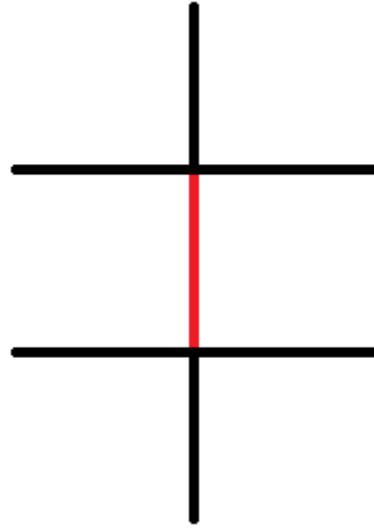


FIGURE 51 - SEGMENT A 6 ROUTES

$O_3 = 8$, est le nombre de segments étant liés à 3 autres segments (Figure 49).

$O_4 = (D - 3) * 4$, est le nombre de segments étant liés à 4 autres segments (Figure 48).

$O_5 = (D - 2) * 4$, est le nombre de segments étant liés à 5 autres segments (Figure 50).

$O_6 = 2 * (D - 3) * (D - 2)$, est le nombre de segments étant liés à 6 autres segments (Figure 51).

Soit N_0 le nombre de segments qui n’ont qu’une seule UBR :

$$N_0 = N_s * S_1$$

N_{UBR} le nombre de segments qui ont deux UBRs :

$$N_{UBR} = N_s * S_2$$

On cherche à déterminer $N_1, N_2, \dots, N_n$ avec N_1 étant le nombre de segments sans UBR qui sont liés à au moins un segment avec une UBR, et N_2 étant le nombre de segments sans UBR qui sont liés à au moins un segment $\in N_1$, sans être liés à un segment N_0 etc.

Pour déterminer le nombre de segments d'un niveau n , nous séparons le calcul en fonction du nombre de routes puis nous sommions l'ensemble. Un segment peut être lié de $x = 3$ à 6 routes et N_{n_x} est la probabilité pour un segment étant lié à x routes, d'être lié à au moins un segment de niveau $n - 1$. Nous en déduisons donc l'expression globale suivante :

$$N_n = \sum_{x=3}^6 O_x N_{n_x}$$

N_{n_x} est exprimé de la façon suivante :

$$N_{n_x} = \sum_{y=1}^x C_x^y \left(\frac{(\prod_{w=0}^{y-1} N_{n-1} - w) * (\prod_{m=0}^{x-y-1} N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k - m)}{\prod_{z=1}^x N_s - z - N_{UBR} - \sum_{k=0}^{n-2} N_k} \right)$$

La variable x désigne le nombre de routes liées aux segments, ceci provient de l'équation globale donnant N_n et prend donc une valeur entre 3 et 6 inclus, et y désigne le nombre de ces routes qui sont de niveau $n - 1$ et prend une valeur entre 1 et x inclus.

Le calcul de N_{n_x} consiste à sommer les probabilités d'avoir un segment lié à une route de niveau $n - 1$ avec celle d'avoir 2 routes $n - 1$ et 3 routes d'un niveau $n - 1$, etc. jusqu'à x routes.

Le résultat du calcul de la probabilité d'avoir y routes de niveau $n - 1$, doit être multiplié par le nombre de combinaisons de y parmi x , puisque la route, ou les routes, de niveau $n - 1$ peuvent se trouver à des endroits différents. Par exemple, si un segment de type O_5 , donc lié à 5 routes, est adjacent à un seul segment de niveau $n - 1$, ce dernier peut se trouver à 5 endroits différents.

La formule $(\prod_{w=0}^{y-1} N_{n-1} - w)$, permet d'exprimer la probabilité d'être lié à w segments de niveau $n - 1$, et nous devons soustraire au fur et à mesure les segments de ce niveau que nous avons déjà considérés, c'est ce que permet d'exprimer w . Nous obtenons le résultat suivant :

$$(N_{n-1} - 0) * (N_{n-1} - 1) * \dots * (N_{n-1} - w)$$

Nous multiplions le résultat de cette expression par la probabilité que les autres segments ne soient pas d'un niveau inférieur à n . Ceci s'exprime de la façon suivante : $(\prod_{m=0}^{x-y-1} N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k - m)$.

$N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k$, permet d'exprimer le nombre total de segments de niveau supérieur à n , c'est-à-dire le nombre total de segments dans la ville auquel nous soustrayons le nombre de segments bénéficiant de deux UBRs et le nombre de segments de niveau inférieur à n . Il faut également soustraire les segments de niveau supérieur à n que nous avons déjà considérés dans le calcul. C'est ce que permet d'exprimer m . La variable peut donc prendre une valeur comprise entre 0 et $x-y-1$ et si

nous développons, nous obtiendrions une expression ayant la forme suivante : $(N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k) * (N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k - m) * \dots * (N_s - N_{UBR} - \sum_{k=0}^{n-1} N_k - m)$

Enfin, nous divisons l’ensemble par le nombre de segments dont le niveau est supérieur à $n-1$. Ce nombre s’obtient en partant du nombre total de segments, exprimé par N_s , auquel nous soustrayons le nombre de segments avec 2 UBRs (N_{UBR}) et le nombre de segments de niveau inférieur ou égal à $n-2$, que nous exprimons par $\sum_{k=0}^{n-2} N_k$. De la même façon que précédemment, il faut au fur et à mesure retirer les segments considérés à chaque étape du produit et ceci est exprimé par z , qui a donc une valeur comprise entre 1 et x . Nous commençons à 1 parce qu’il faut, d’emblée, soustraire le segment que nous sommes en train de traiter (représenté en rouge dans les Figure 48 à Figure 51).

5.2 Charge des intérêts

Chaque véhicule émet son intérêt dans chacun des CAMs, ce qui nous donne pour la charge réseau dédiée aux intérêts, l’expression suivante :

$$CR_i = PO_i * \alpha * F_{CAM}$$

5.3 Charge des données

La charge réseau engendrée par les segments sans UBR est donnée par le nombre de segments sans UBR multiplié par la valeur médiane de H_3 et M_3 .

H_3 étant le nombre de sauts dans le pire des cas et M_3 dans le meilleur et puisque les véhicules sont répartis uniformément, alors le nombre moyen de CAMs nécessaires pour couvrir un segment est égal à l’arrondi supérieur de la valeur médiane des deux. Nous obtenons :

$$HL = \frac{\sigma}{10} \text{ et } n = (HL - HM_2)/HM_3$$

Pour déterminer le nombre de segments sans UBR qui ont été contaminés entre deux générations de données nous utilisons N_n qui donne une borne supérieure du nombre de segments qui peuvent être contaminés à l’étape n . Le résultat de CR sera donc toujours au-dessus de ce que nous aurons en réalité.

$$CR_{d0} = PO_d * \vartheta(HM_3, HL) * \left(N_{[n]} * \{n\} + \sum_{x=1}^{[n]} (N_x) \right) * \vartheta(\sigma, F_{CAM})$$

La charge réseau engendrée par les segments avec 1 UBR est la somme des CAM d’origine envoyés par l’UBR et des CAMs envoyés par les véhicules pour couvrir le reste du segment. Nous divisons le nombre de CAMs d’origine par le nombre moyen d’entrées d’un carrefour pour ne pas les compter plusieurs fois. Nous multiplions cette somme, par le nombre de segments avec 1 UBR, puis par le poids d’une donnée et enfin par la fréquence de génération de la donnée, en veillant à ce que celle-ci ne dépasse par la fréquence des CAMs, ainsi nous obtenons :

$$CR_{d1} = \left(PO_d * \left((N_s * S_1) * \left(\vartheta(HM_2, HL) + \frac{1}{\varphi} \right) \right) \right) * \vartheta(\sigma, F_{CAM})$$

La charge réseau engendrée par les segments avec 2 UBRs est la somme des CAMs d'origines envoyés par les UBR et des CAMs envoyés par les véhicules pour couvrir le reste du segment. Nous divisons le nombre de CAMs d'origine par le nombre moyen d'entrées d'un carrefour pour ne pas les compter plusieurs fois. Nous multiplions cette somme, par le nombre de segments avec 2 UBRs, puis par le poids d'une donnée et enfin par la fréquence de génération de la donnée, en veillant à ce que celle-ci ne dépasse par la fréquence du CAM, cela nous donne :

$$CR_{d2} = \left(PO_d * \left((N_s * S_2) * \left(\vartheta(HM_1, HL) + \frac{2}{\varphi} \right) \right) \right) * \vartheta(\sigma, F_{CAM})$$

5.4 Charge réseau

On obtient donc pour l'expression de la charge réseau liée à notre approche :

$$CR = CR_i + CR_{d0} + CR_{d1} + CR_{d2}$$

Le résultat obtenu est une borne maximum du résultat réel attendu, puisque nous avons maximisé le nombre de segments, sans d'UBR, contaminés à chaque étape.

5.5 Fraîcheur réseau

Pour la fraîcheur, le pire des cas est celui où les segments qui n'ont pas du tout d'UBR ne reçoivent pas l'information. Nous assumons donc l'hypothèse suivante : le rayon d'un croisement est supérieur à la portée d'un véhicule multiplié par HL, divisé par deux (pour aboutir au segment), ce qui se traduit par :

$$(L - Prsu) > Pv * HL$$

La fraîcheur est donnée par le nombre de véhicules se trouvant dans la zone de couverture au moment où la donnée est générée, auxquels nous ajoutons, en fonction de la relation inter-distance/portée des véhicules, soit le nombre de véhicules qui rentrent dans la zone de couverture, soit le nombre de véhicules qui bénéficient de l'information par d'autres véhicules, entre deux générations de données, ainsi nous obtenons :

D'une part, si l'inter-distance est supérieure à la portée de transmission des véhicules, α_D s'exprime de la façon suivante :

$$\alpha_D = \left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\vartheta(\sigma, F_{CAM})}$$

Et donc si on en revient à la définition générale de la fraîcheur réseau, nous avons :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta \right) + \frac{\lambda * \varphi * \beta}{\vartheta(\sigma, F_{CAM})}}{\alpha} * 100$$

D’autre part, si l’inter-distance est inférieure à la portée de transmission des véhicules, α_D est obtenu par l’expression suivante :

$$\alpha_D = \left(\frac{\alpha}{100} * \delta \right) + \frac{\alpha}{N_s} * P_{or1} * N_{UBR} * \vartheta \left(\frac{HM_1}{HL}, 1 \right) + \frac{\alpha}{N_s} * P_{or2} * N_0 * \vartheta \left(\frac{HM_2}{HL}, 1 \right)$$

Ce qui nous donne pour la définition générale de la fraîcheur :

$$FR = \frac{\left(\frac{\alpha}{100} * \delta \right) + \frac{\alpha}{N_s} * P_{or1} * N_{UBR} * \vartheta \left(\frac{HM_1}{HL}, 1 \right) + \frac{\alpha}{N_s} * P_{or2} * N_0 * \vartheta \left(\frac{HM_2}{HL}, 1 \right)}{\alpha} * 100$$

VI – Discussions

L’approche NDN permet de se tenir informé de la mise à jour d’une donnée. Dans ce cas, nous sommes obligés d’envoyer régulièrement des requêtes. Le NDN n’a pas du tout été pensé pour qu’une donnée soit mise à jour. Chaque message est indépendant et doit donc être accompagné de toute les informations nécessaires pour assurer son authentification et son intégrité. Ensuite, aucun mécanisme ne permet de savoir si un client possède déjà la donnée qu’il demande. Enfin, si un véhicule n’est pas à portée d’un feu, il n’a aucun moyen de bénéficier de la donnée.

L’approche RENE est de type rendez-vous, ce qui implique la nécessité d’une connexion cellulaire, afin qu’un client puisse s’abonner à une donnée. Avec cette approche, la donnée n’est envoyée que quand elle est mise à jour. Le client doit se réabonner auprès du serveur chaque fois qu’il change de zone pour que celui-ci puisse calculer une nouvelle route. De plus, cette donnée est indépendante et doit être accompagnée de toutes les informations nécessaires à s’assurer de son authenticité et de son intégrité. Enfin, la donnée est envoyée indépendamment à chaque véhicule se trouvant à portée d’un feu et ne peut parvenir à un véhicule ne se trouvant pas à portée de l’un d’eux.

Quant à notre approche GeoVCDN, elle donne la possibilité de regrouper plusieurs données en une seule. De plus, nous intégrons celle-ci dans les CAMs, ce qui nous permet à la fois de réduire le nombre d’intérêts nécessaires (on a besoin que d’un seul intérêt) et de ne pas ajouter d’en-tête-sécurité (pour l’authenticité et l’intégrité), puisque nous profitons de celle du CAM qui est déjà présente dans tous les cas. Les clients sont aussi fournisseurs de données, ce qui permet aux paquets d’être distribués à plus de véhicules. Enfin, nous avons mis en place un mécanisme permettant de stopper la propagation de la donnée quand nous considérons qu’elle n’est plus forcément valide.

Enfin, dans ce chapitre nous avons proposé de modéliser une ville intelligente sous la forme d’un Cloud de contexte, composé des véhicules, des capteurs et d’un serveur central. Nous avons adopté une approche ICN pour les échanges de données entre les différents acteurs. Les approches ICN existantes dans la littérature présentaient des lacunes pour être utilisées sur le canal G5, à cause, notamment, de la taille des paquets en grande partie engendrée par la sécurité. Nous avons donc proposé une méthode efficace pour obtenir un ensemble de données de petite taille, en intégrant les échanges ICN à l’intérieur du CAM, et en regroupant toutes ces données dans une donnée plus

conséquence avec un numéro de version. L'ajout de la version et le regroupement des données est adapté aux données de faible taille mises à jour régulièrement mais pas trop souvent. L'intégration des échanges ICN dans le CAM est, elle, adaptée à tous les échanges, excepté les échanges où le délai est critique, puisque nous sommes alors dépendant de la fréquence d'envoi des CAMs. Dans le chapitre suivant, nous apportons une évaluation des performances des différentes approches ICN que nous venons de modéliser. Cette évaluation se fait d'abord sur les modèles théoriques, puis par des simulations.

Chapitre 4 : Evaluation des performances ICN

I – Comparaison des modèles

Dans ce chapitre, nous allons comparer les approches NDN, RENE, VCDN, VCDNoCAM et GeoVCDN, d'après les modèles proposés dans le chapitre précédent. Ces comparaisons se font sur deux indicateurs de performance : la charge réseau et la fraîcheur réseau. L'objectif de celles-ci étant de confirmer l'apport bénéfique de notre approche au niveau des performances réseau. De plus, nous avons comparé ces approches à l'aide de simulations. Pour cela nous avons développé un simulateur ad hoc qui prend en compte certains aspects de la communication radio et la mobilité des véhicules. Les résultats des simulations sont ensuite comparés à ceux des modèles théoriques. Enfin, nous appliquons ces approches sur un cas d'usage particulier, à savoir la recherche du meilleur chemin, l'objectif étant de valider, cette fois-ci, l'apport de notre contribution au niveau applicatif.

I – Comparaison des modèles

Nous voulons comparer les différentes approches, cinq au total, puisque notre solution se décline sous 3 versions : la première (VCDN) regroupe plusieurs données en une seule, la deuxième (VCDNoCAM) inclus, en plus, ses échanges dans les CAM, et la troisième (GeoVCDN) est l'extension de la seconde à laquelle s'ajoute les communications entre les véhicules. Pour ce faire, nous faisons apparaître le comportement de chacune d'elles par rapport à chaque variable.

Le Tableau 8 rappelle la description des variables que nous utiliserons dans ce chapitre et leur valeur par défaut, réaliste, basée sur nos observations (projets Scoop, InterCor et C-Roads). Chaque fois qu'une variable diffère de sa valeur par défaut, nous le mentionnons avant de présenter les résultats.

TABLEAU 8 - DESCRIPTION DES VARIABLES ET VALEURS PAR DEFAUT

DESIGNATION	DESCRIPTION	VALEUR
R_C	Rayon d'un carrefour	300
P_V	Portée de transmission d'un véhicule	200
P_{UBR}	Portée de transmission d'une infrastructure	250
D	Dimension de la grille	15
α	Nombre de véhicules	10000
β	Nombre de feux	80
σ	Fréquence de génération de la donnée en Hz	1

1. Charge réseau

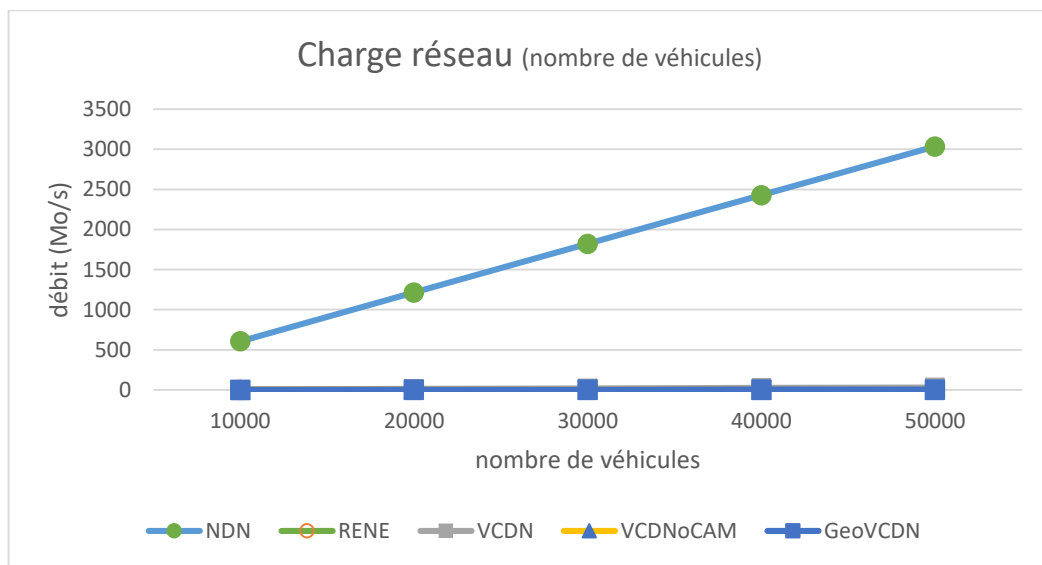
On définit la charge réseau comme étant le débit de données qui transitent entre les différents protagonistes, sur le canal G5, sur la totalité du réseau, et qui est dédié aux mécanismes ICN.

Ces mécanismes se déclinent sous deux formes : l'envoi d'intérêts et la réception de données.

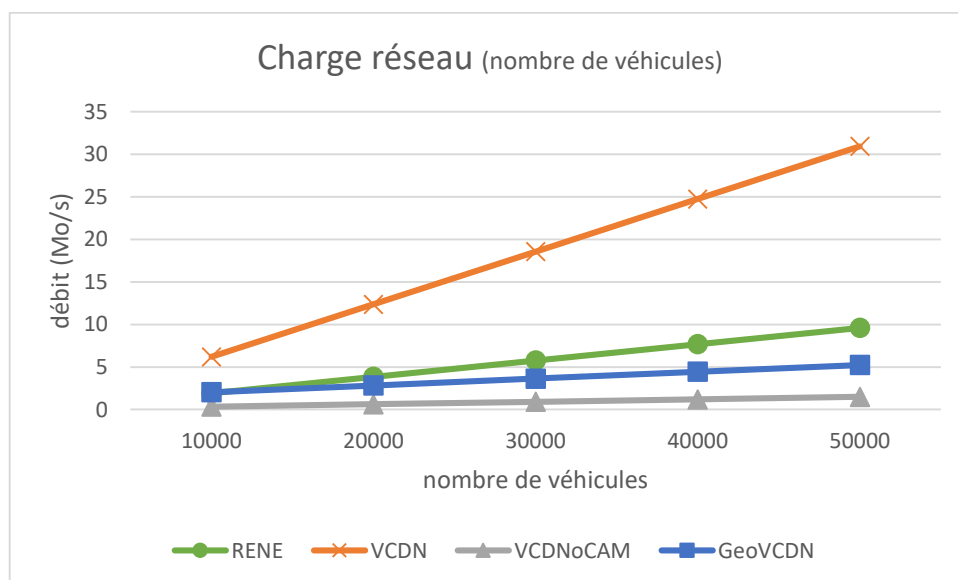
Au niveau des figures qui vont suivre, nous ne ferons pas apparaître l'approche NDN, pour laquelle les résultats sont en moyenne cent fois plus élevés que les autres approches et empêchent d'observer les différences entre ces dernières (comme on pourra le constater dans la Figure 52.a)

Dans un premier temps, nous constatons dans la Figure 52 que pour l'approche VCDN, le nombre de véhicules a un fort impact, puisque, comme pour NDN, tous les véhicules reçoivent la donnée de chaque capteur pour chaque intérêt envoyé. La donnée est envoyée proportionnellement au nombre de véhicules qui se trouvent dans la zone de couverture.

Pour l'approche RENE, le principe est le même mais l'impact est moins fort parce que le poids de la donnée est réduit.



(a)



(b)

FIGURE 52 - CHARGE RESEAU ANALYTIQUE - NOMBRE DE VEHICULES

Le meilleur résultat est obtenu par VCDNoCAM, le nombre de véhicules n'ayant aucune répercussion sur la charge réseau liée à l'envoi de la donnée. Cela est dû au fait que les informations sont envoyées par les UBR en total indépendance du nombre de véhicules à sa portée. Puis, avec GeoVCDN, nous faisons mieux que RENE. Par exemple pour 50 000 véhicules, la consommation de la bande passante avec l'approche RENE est 83% plus élevée que celle avec l'approche GeoVCDN. Les raisons sont les mêmes que pour l'approche VCDNoCAM, auxquelles nous rajoutons l'échange de la donnée entre véhicules, en dehors de la zone de couverture. C'est aussi pour cela que GeoVCDN consomme plus de bande passante que VCDNoCAM.

Ensuite, nous constatons que quand nous augmentons le rayon d'un croisement, nous baissions la zone de couverture et diminuons la densité de la circulation, ce qui explique la baisse de la charge réseau pour les approches qui ne gèrent pas les communications de client à client. Pour le GeoVCDN, la croissance de la courbe est due à l'augmentation du nombre de sauts (et donc d'envois de la donnée) nécessaires pour couvrir un segment. En effet, si le nombre de sauts maximum que peut faire une donnée permet de couvrir la totalité du réseau, alors l'augmentation de la taille d'un segment a pour effet d'augmenter le nombre de sauts nécessaires pour couvrir la totalité du réseau.

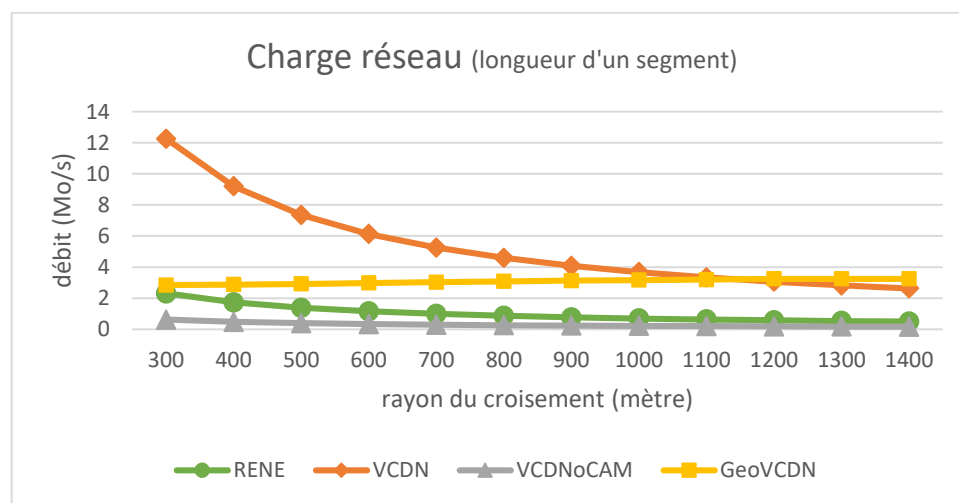


FIGURE 53 - CHARGE RESEAU ANALYTIQUE - LONGUEUR D'UN SEGMENT

Dans la Figure 54, nous voyons qu'à partir d'un moment nous ne pouvons plus couvrir la totalité des segments, à cause de la limite de sauts. A partir de ce seuil, la charge réseau devient constante.

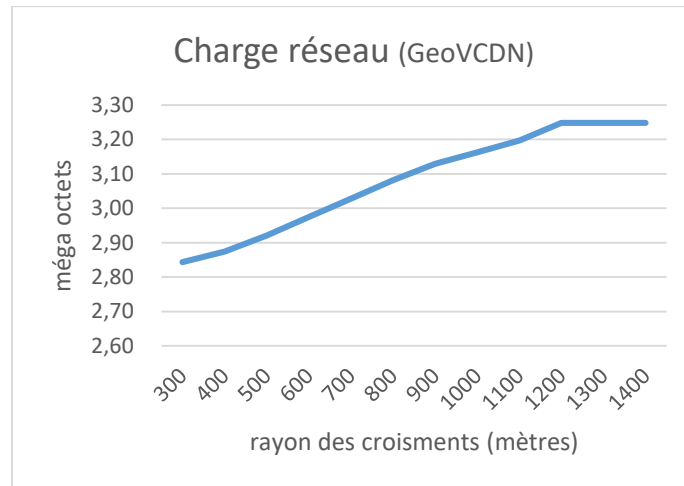


FIGURE 54 - CHARGE RESEAU ANALYTIQUE - LONGUEUR D'UN SEGMENT – GEOVCDN

De plus, la portée des UBRs modifie la zone de couverture et donc le nombre de véhicules qui reçoivent la donnée. Dans le cas du VCDN, la conséquence est la même que si nous augmentions le nombre de véhicules : plus d'intérêts et donc plus de bande passante consommée.

Pour RENE, la couverture n'a aucun impact sur la charge des intérêts, qui est le plus gros consommateur de bande passante dans son fonctionnement. Cependant nous envoyons la donnée à plus de véhicules, ce qui explique cette légère augmentation.

Avec VCDNoCAM, l'impact de la couverture est nul sur la charge de données. Elle influe seulement sur la charge des intérêts.

GeoVCDN est la seule approche qui tend à diminuer, puisque nous réalisons moins d'échanges entre véhicules, alors que pour une portée de 50 mètres, l'approche GeoVCDN consomme 25% plus de bande passante que l'approche RENE. A 250 mètres de portée, elle n'est plus que de 6%.

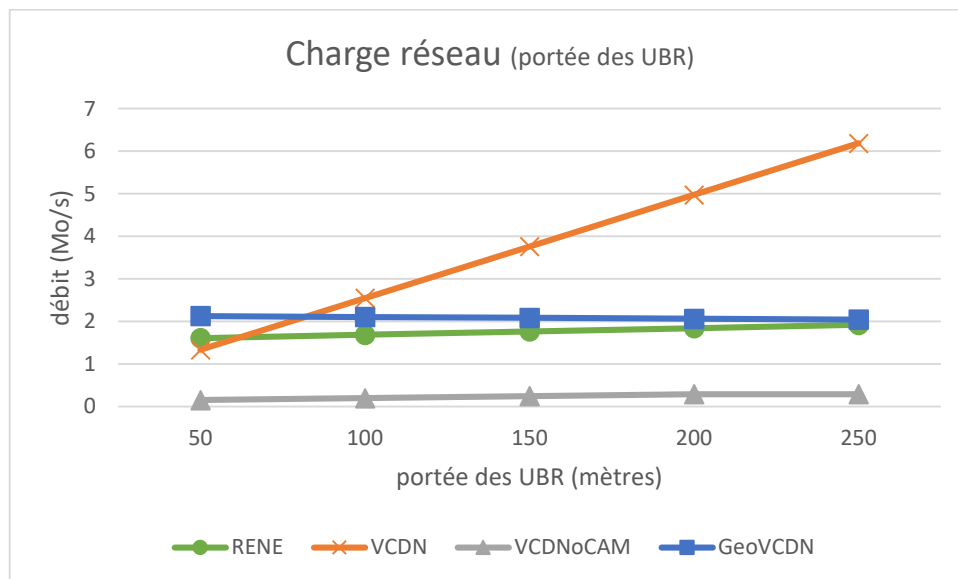


FIGURE 55 - CHARGE RESEAU ANALYTIQUE - PORTEE DES UBR

I – Comparaison des modèles

D'autre part, nous observons qu'à terme, avec une couverture des UBRs atteignant 100%, GeoVCDN et VCDNoCAM se rejoignent, puisqu'il n'y a plus aucun échange de véhicule à véhicule. Dans la configuration proposée, et à 100% de couverture, la consommation de la bande passante pour les approches VCDNoCAM et GeoVCDN est inférieure de 85% par rapport à celle obtenue par une approche de type RENE.

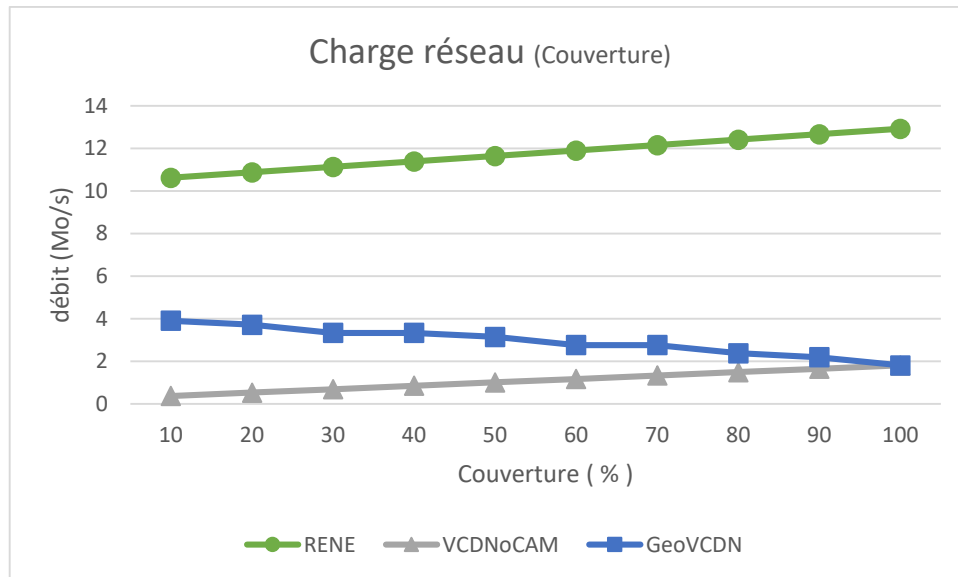


FIGURE 56 - CHARGE RESEAU ANALYTIQUE - COUVERTURE

De plus, en partant du principe que la portée des véhicules est inférieure à celle des UBR (ce qui est le cas dans la réalité), elle n'a d'importance que pour l'approche GeoVCDN, puisque c'est la seule qui gère les échanges de client à client, comme nous pouvons le voir dans la Figure 57.

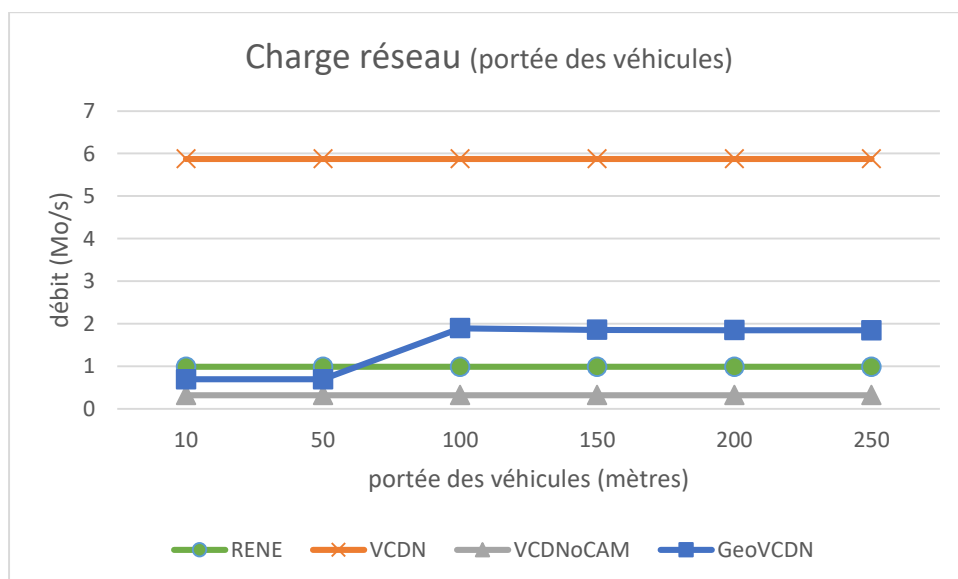


FIGURE 57 - CHARGE RESEAU ANALYTIQUE - PORTEE DES VEHICULES

Entre 0 et 20 mètres, la portée de communication ne suffit pas à assurer la connectivité du réseau, car l'inter-distance des véhicules est supérieure à la portée de transmission. Quand la portée est suffisante, nous observons un pic qui correspond au coût de partage des données entre les véhicules (Figure 58). Après ce pic, nous observons une baisse légère et régulière des échanges inter-véhicules, puisqu'un véhicule en couvre de plus en plus, diminuant ainsi le nombre de sauts nécessaires pour couvrir un segment.

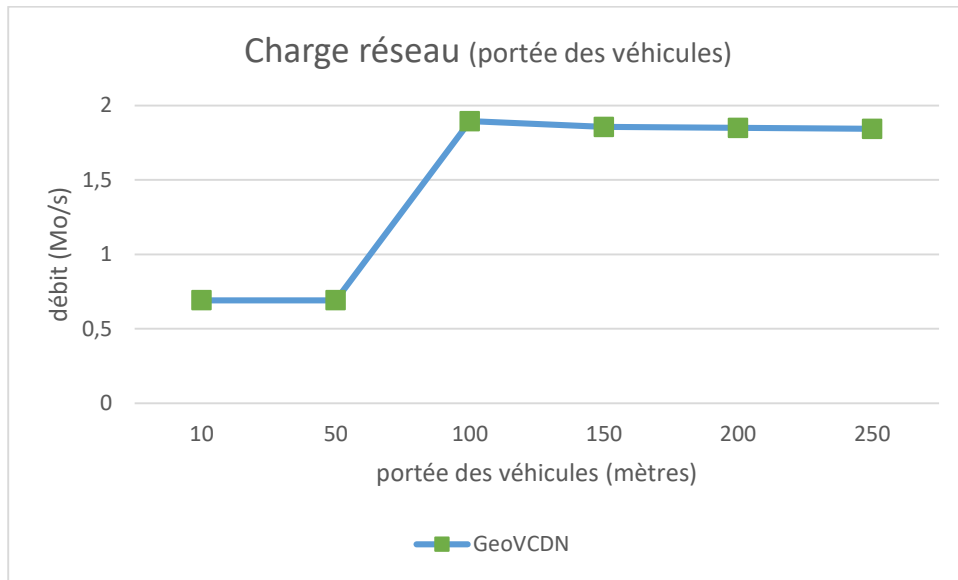


FIGURE 58 - CHARGE RESEAU ANALYTIQUE - PORTEE DES VEHICULES - GEOVCDN

D'autre part, en augmentant le nombre d'UBRs, nous augmentons, aussi, le nombre de points d'entrée dans la zone de couverture. Dans la Figure 59, nous voyons que cette augmentation se traduit, pour l'approche RENE, par un plus grand nombre d'intérêts envoyés.

De plus, la croissance du nombre d'UBRs entraîne celle de la zone de couverture, qui elle-même implique un plus grand nombre de véhicules concernés par l'envoi de la donnée et/ou d'intérêts. Pour RENE et VCDN, l'impact est fort, puisque la charge des données est liée au nombre de véhicules dans la zone de couverture. Pour VCDNoCAM, la charge des données est proportionnelle au nombre d'UBRs, d'où cette légère augmentation.

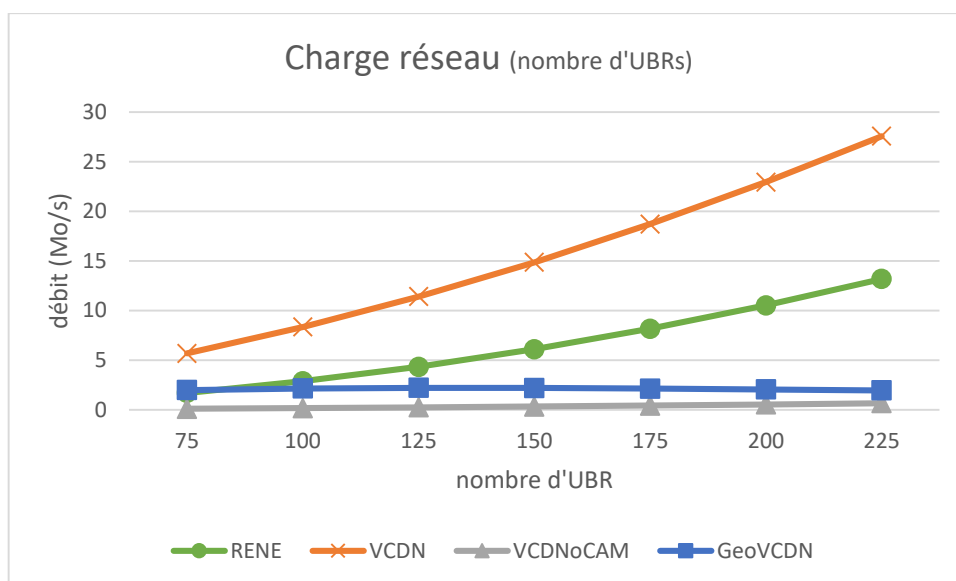


FIGURE 59 - CHARGE RESEAU ANALYTIQUE - NOMBRE D'UBRS

Dans la Figure 60, nous isolons une partie des courbes et nous constatons que pour GeoVCDN, l'augmentation du nombre d'UBRs a des conséquences positives. Cela est dû au fait que cette augmentation n'a aucun impact sur la charge des intérêts envoyés mais a une incidence bénéfique sur la charge des données. En augmentant la densité des UBRs, nous réduisons le nombre de messages échangés entre véhicules.

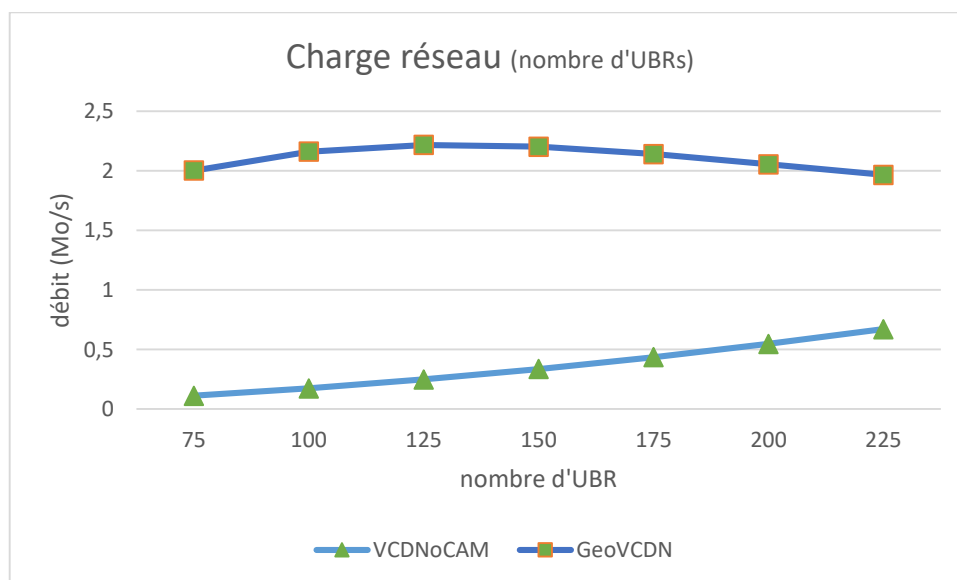


FIGURE 60 - CHARGE RESEAU ANALYTIQUE - NOMBRE D'UBRS

Ensuite, pour VCDN, RENE et VCDNoCAM, l'augmentation de la fréquence de génération de la donnée induit un comportement similaire de la charge liée à celle-ci, comme le montre la Figure 61.

Plus le poids de la donnée est élevé, plus la pente est forte. En VCDN, le poids de la donnée est autant de fois supérieur à celui de RENE qu'il y a de capteurs.

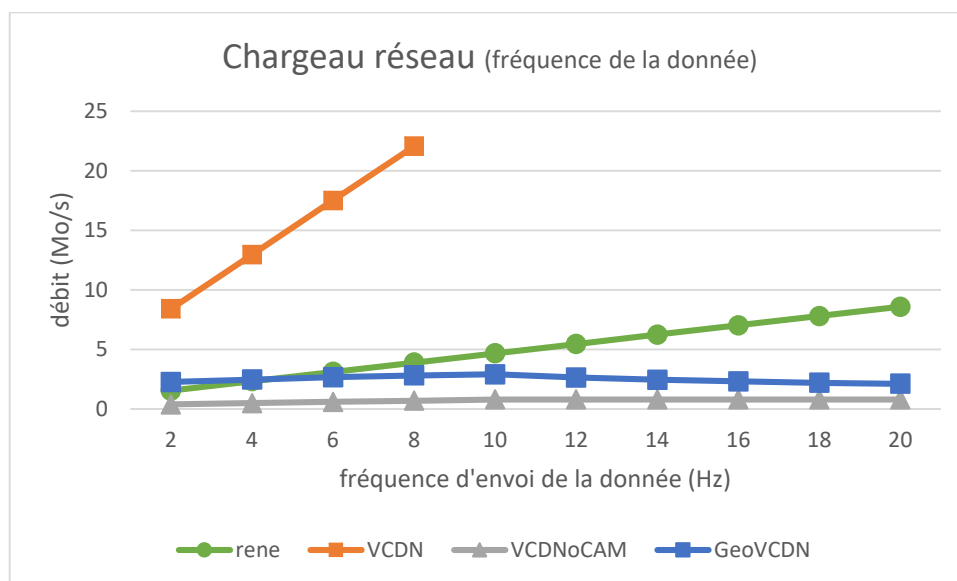


FIGURE 61 - CHARGE RESEAU ANALYTIQUE - FREQUENCE DE LA DONNEE

En VCDNoCAM, nous sommes limités par la fréquence d'envoi des CAMs (10 Hz) d'où cette constance, passé cette limite. Dans la Figure 62, nous pouvons voir qu'en GeoVCDN, plus la fréquence augmente, moins les échanges entre véhicules sont nombreux. Passé 10 Hz, la charge au niveau des UBRs devient constante et les échanges entre véhicules diminuent, d'où la décroissance à partir de ce seuil. C'est pour ces mêmes raisons que, plus la fréquence est haute, plus la consommation de la bande passante avec l'approche GeoVCDN est basse par rapport à celle d'une approche de type RENE. Par exemple à 10Hz, l'approche RENE consomme 60% de bande passante de plus que les véhicules qui échangent en GeoVCDN.

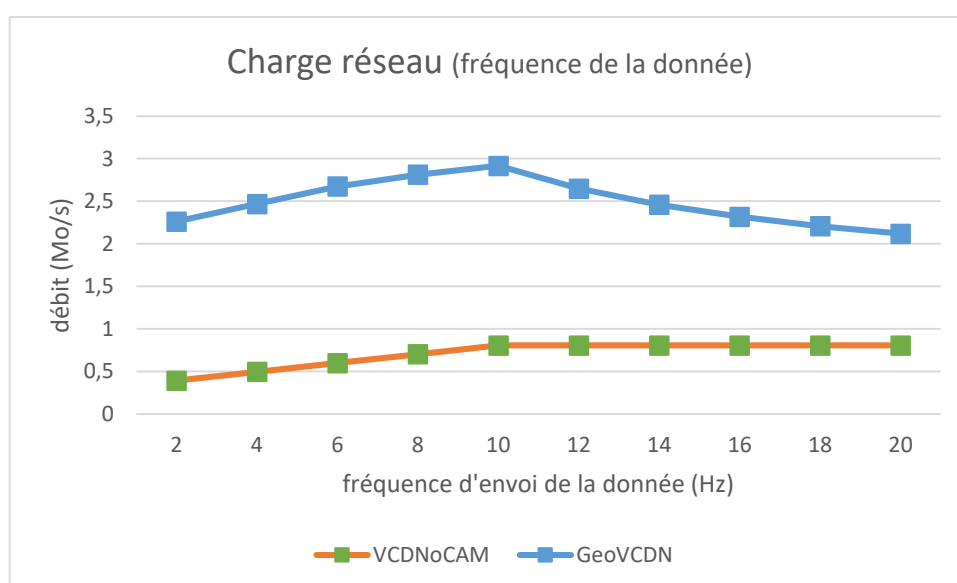


FIGURE 62 - CHARGE RESEAU ANALYTIQUE - FREQUENCE DE LA DONNEE - GEOVCDN & VCDNoCAM

Ensuite, alors que le nombre de capteurs a un impact sur le poids des intérêts envoyés par l'approche RENE, il n'en a aucun sur les approches où la donnée est centralisée (VCDN, VCDNoCAM et GeoVCDN), comme nous pouvons l'observer sur la Figure 63.

Et au contraire, alors qu'avec l'approche RENE le poids de la donnée n'est pas affecté (on n'envoie que la donnée du capteur qui a évolué), là où la donnée est centralisée, son poids est proportionnel au nombre de capteurs. Avec VCDN, la charge explose, car ce poids est également multiplié par le nombre de véhicules, tandis qu'avec les autres approches (VCDNoCAM et GeoVCDN), il n'est multiplié que par le nombre d'UBRs.

Il apparaît que la charge des intérêts envoyés avec l'approche RENE évolue plus vite que la charge des données envoyées. Cela est dû à la taille de la couche sécurité.

Par exemple, avec 50 000 capteurs, l'approche RENE consomme 31% de bande passante de plus que l'approche GeoVCDN et 403% de plus que l'approche VCDNoCAM

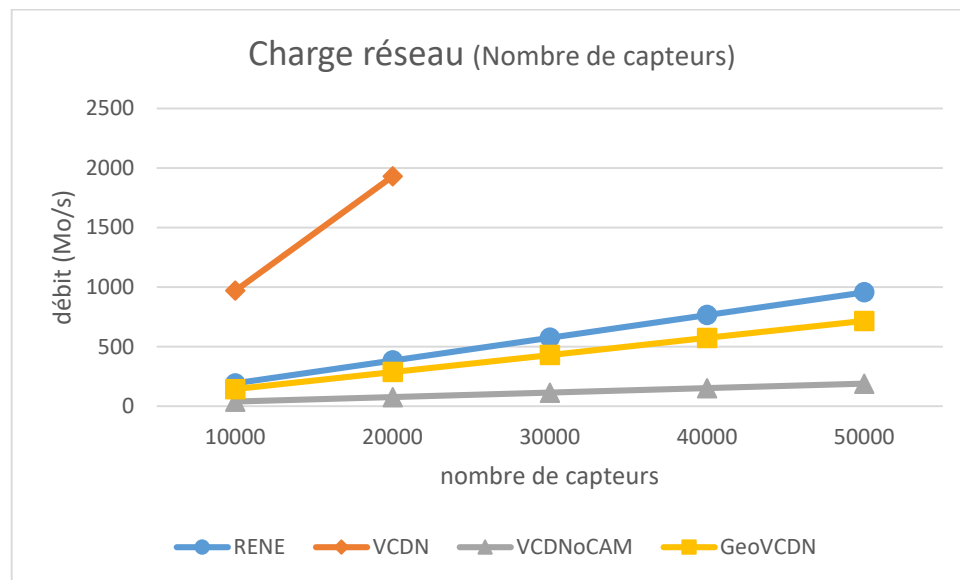


FIGURE 63 - CHARGE RESEAU ANALYTIQUE - NOMBRE DE CAPTEURS

Enfin, la densité évolue selon deux facteurs : le nombre de véhicules et la longueur des segments. Si nous augmentons les deux au même rythme, la densité n'évolue pas et nous perdons en performance pour l'approche GeoVCDN. Ceci s'explique par le fait que le nombre de véhicules à l'intérieur de la zone de couverture est stable aussi, d'où la constance des courbes que nous observons sur la Figure 64 pour les approches sans communication de client à client. Pour l'approche GeoVCDN, plus nous augmentons le nombre de véhicules, plus nous envoyons d'intérêts. De la même façon, plus nous augmentons la longueur d'un segment, plus il faut de sauts pour couvrir un segment. Ainsi, nous perdons sur les deux tableaux, d'où cette nette augmentation.

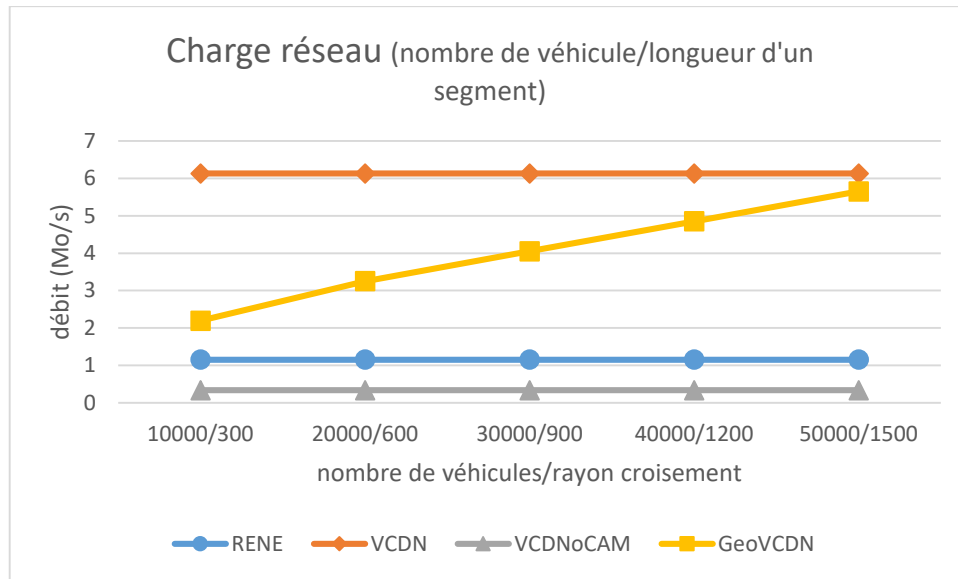


FIGURE 64 - CHARGE RESEAU ANALYTIQUE - NOMBRE DE VEHICULES/LONGUEUR D'UN SEGMENT

2. Fraîcheur

La fraîcheur du réseau est mesurée comme étant le pourcentage de véhicules bénéficiant de l'information avant que celle-ci ne soit périmée. La donnée est périmée quand une nouvelle donnée a été produite. Dans notre environnement, c'est la fréquence de la donnée qui détermine ce délai de péremption.

Dans un premier temps, pour la comparaison sur la portée des véhicules, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 9.

TABLEAU 9 - FRAICHEUR RESEAU ANALYTIQUE - PORTEE DES VEHICULES

β	R_c
225	700

La portée de transmission des véhicules, n'a, évidemment, aucun impact sur les approches qui ne font pas de communication de client à client, comme le montre la Figure 65. Pour l'approche GeoVCDN, en revanche, à partir du moment où la portée est assez grande pour communiquer avec les véhicules environnants, la fraîcheur grimpe en flèche jusqu'à rapidement atteindre 100%, dès lors que la limite de sauts le permet. Par exemple, avec une portée de 150 mètres, avec GeoVCDN, nous atteignons 100% de fraîcheur contre 36% pour les autres approches, soit une différence de 64 points.

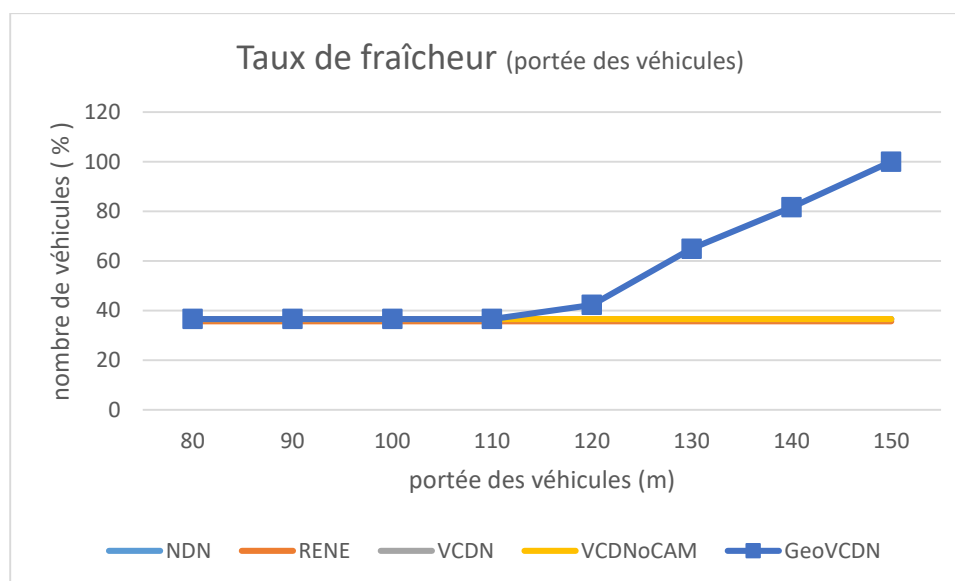


FIGURE 65 - FRAICHEUR RESEAU ANALYTIQUE - PORTEE DES VEHICULES

Ensuite, pour la comparaison par rapport au nombre de véhicules, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 10.

TABLEAU 10 - FRAICHEUR RESEAU ANALYTIQUE - NOMBRE DE VEHICULES

β	R_c
150	700

Le nombre de véhicules ne fait pas évoluer la zone de couverture, ni le nombre de points d'entrée dans celle-ci. Puisque le taux de fraîcheur s'exprime en pourcentage, la quantité absolue de véhicules n'a d'importance que pour l'approche GeoVCDN, comme nous pouvons le constater sur la Figure 66. Avec celle-ci, plus il y a de véhicules, plus ils sont rapprochés et plus nous avons de chance d'utiliser la portée maximale de véhicules pour une couverture optimale. A noter, que le taux de fraîcheur est borné par d'autres paramètres, tel que le « hop limit ».

Avec 10 000 véhicules, l'approche GeoVCDN voit son taux de fraîcheur plus élevé que les autres, de 25 points. Avec 30 000 véhicules, il est plus élevé de 45 points avec 75% des véhicules qui obtiennent l'information avant son expiration. L'augmentation qu'on constate est de type logarithmique, comme l'illustre la figure.

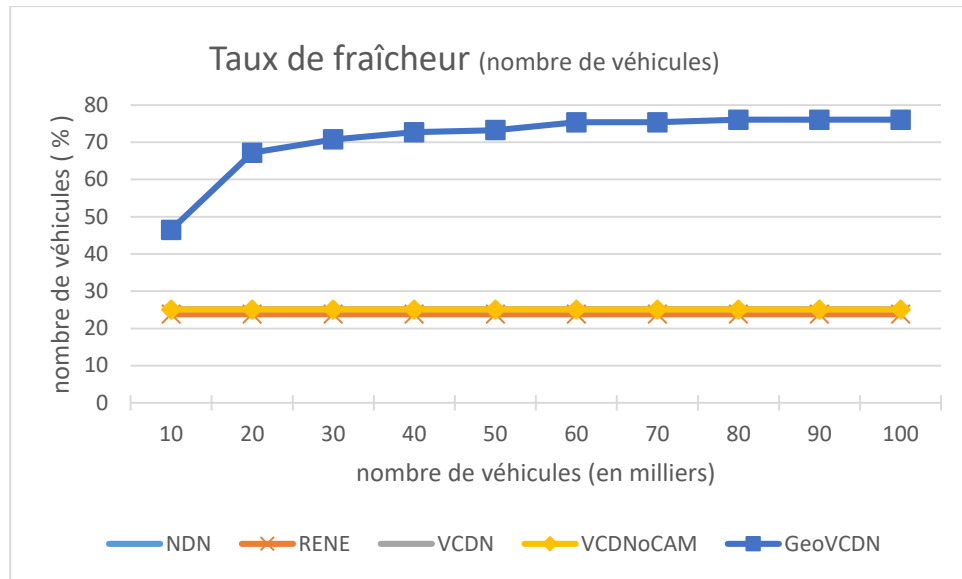


FIGURE 66 - FRAICHEUR RESEAU ANALYTIQUE - NOMBRE DE VEHICULES

D'autre part, pour la comparaison sur le nombre d'UBRs, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 11.

TABLEAU 11 - FRAICHEUR RESEAU ANALYTIQUE - NOMBRE D'UBRS

P_V	R_c
75	700

Dans la Figure 67 nous voyons que le nombre d'UBRs a un impact positif sur le taux de fraîcheur de chacune des approches. Nous augmentons le nombre de points d'entrée dans la zone de couverture, ce qui a pour effet d'augmenter le nombre de véhicules s'y trouvant, au moment de la génération de la donnée. Pour GeoVCDN, augmenter le nombre d'UBRs c'est aussi augmenter le nombre de points de départs pour propager la donnée entre clients et donc affecter plus de véhicules, d'où cette pente plus forte sur la figure. Par exemple pour 25 UBRs, le taux de fraîcheur est pratiquement égal entre GeoVCDN et les autres (8% pour GeoVCDN et 4% pour les autres), tandis qu'avec 200 UBRs, il atteint 60% pour le GeoVCDN et 33% pour les autres.

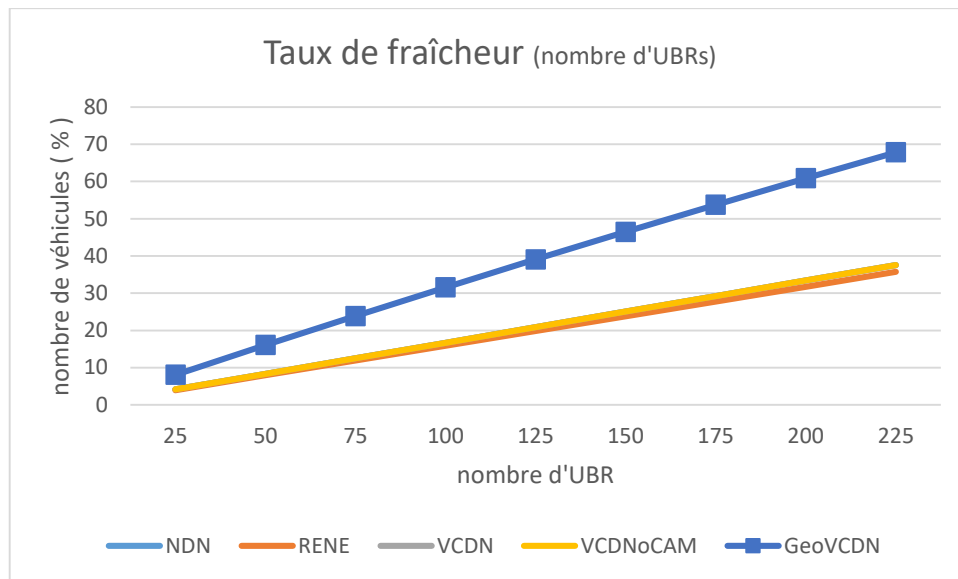


FIGURE 67 - CHARGE RESEAU ANALYTIQUE - NOMBRE D'UBRS

De même, pour la comparaison par rapport à la portée des UBRs, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 12.

TABLEAU 12 - FRAICHEUR RESEAU ANALYTIQUE - PORTEE DES UBRs

β	R_c	P_v
225	700	75

Augmenter la portée des UBRs, c'est augmenter le nombre de véhicules qui sont à portée de celles-ci. Dans la Figure 68, nous pouvons observer que la conséquence est la même pour toutes les approches. En GeoVCDN, si nous enlevions l'axiome de départ concernant le rapport entre la taille d'un croisement et la portée des véhicules, la courbe aurait une autre forme et grimperait plus vite.

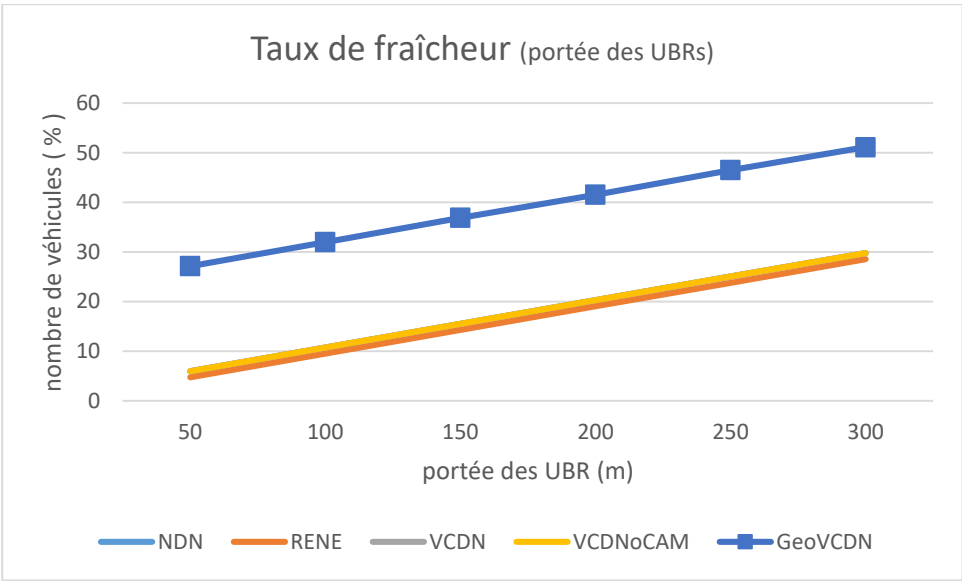


FIGURE 68 - FRAICHEUR RESEAU ANALYTIQUE - PORTEE DES UBRs

Enfin, pour la comparaison par rapport à la fréquence de la donnée, certains paramètres ne sont de nouveau pas dans leur état par défaut. Les variations sont précisées dans le Tableau 13.

TABLEAU 13 - FRAICHEUR RESEAU ANALYTIQUE - FREQUENCE DE LA DONNEE

β	α	R_c	P_v
150	20000	700	40

En augmentant la fréquence de génération d’une donnée, nous diminuons dans le même temps, la limite des sauts et donc le nombre de véhicules qui peuvent être affectés par les communications de client à client (GeoVCDN). Pour VCDN, NDN et VCDNoCAM, nous observons, dans la Figure 69,

également une légère perte, dûe à la diminution du nombre de véhicules qui ont le temps de rentrer dans la zone de couverture entre deux générations.

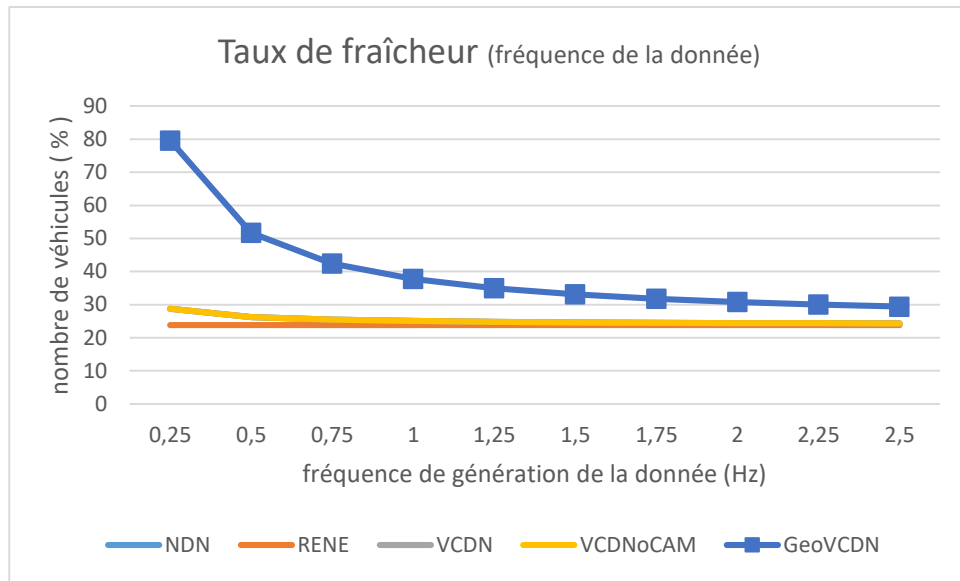


FIGURE 69 - FRAICHEUR RESEAU ANALYTIQUE - FREQUENCE DE LA DONNEE

3. Résumé

Ces courbes mettent en évidence l'efficacité de nos approches. VCDNoCAM assure le même service que les approches qui existent déjà (aussi bien que NDN et mieux que RENE), en termes de livraison de l'information, et à un moindre coût.

La méthode NDN donne des résultats catastrophiques, dans un environnement où les données sont nombreuses et évoluent souvent ; impliquant la nécessité d'envoyer très (trop) souvent des intérêts pour chacune de ces données, pour espérer les garder à jour. NDN a été pensé pour alléger le trafic dans le réseau sans tenir compte du lien vers le client. Or, en véhiculaire, le canal utilisé, l'ITS-G5, a une bande passante réduite, que nous voudrions économiser au maximum. Alors, si nous voulons garantir l'authenticité et l'intégrité de chacun des messages qui transitent sur ce canal, cela devient vite coûteux.

On a vu que nous arrivions à économiser sur cette charge, liée à la couche sécurité, en regroupant un ensemble de données dans une métadonnée, plus lourde, mais qui permet de réduire la couche sécurité à une seule occurrence. Nous avons également rajouté un système de versions, qui évite d'envoyer une donnée à un protagoniste qui la possède déjà. Malgré ces deux mécanismes, les résultats ne sont toujours pas satisfaisants (VCDN).

La méthode RENE est bien plus adaptée à l'environnement, puisque la donnée n'est envoyée que quand elle change. Néanmoins, comme pour NDN, le lien final vers le client n'est pas pris en compte et chaque véhicule d'une même zone va recevoir individuellement l'information, alors qu'un ensemble de véhicules aurait pu la recevoir.

En exploitant les technologies existantes, nous arrivons à réduire drastiquement le coût des données dédiées aux échanges ICN (VCDNoCAM), en intégrant ces derniers dans les CAMs qui sont nativement émis par les ITS. Nous profitons ainsi de la couche sécurité des CAMs et la donnée est envoyée à toutes les ITS à portée de transmission en une seule fois. Les courbes mettent en évidence le gain sur la charge réseau qu’offre cette méthode.

Enfin, dans l’idée d’améliorer la distribution de l’information dans le réseau G5, nous exploitons, toujours grâce aux CAMs, les communications de client à client, pour permettre à la donnée de voyager plus loin que la simple zone de couverture des UBRs. Cet ajout de fonctionnalité a un coût, qui reste inférieur aux méthodes ICN déjà existantes, et qui améliore de façon considérable le service, tout en étant mieux adapté, comme le montrent les courbes, à un passage à grande échelle.

II - Simulation

Afin de valider les modèles proposés, nous avons développé un simulateur à l’image de la Figure 39. Nous avons des véhicules qui évoluent et communiquent dans une ville dont la topologie correspond au type Manhattan. Comme nous l’avons déjà précisé, ce modèle a été retenu pour faciliter l’implémentation.

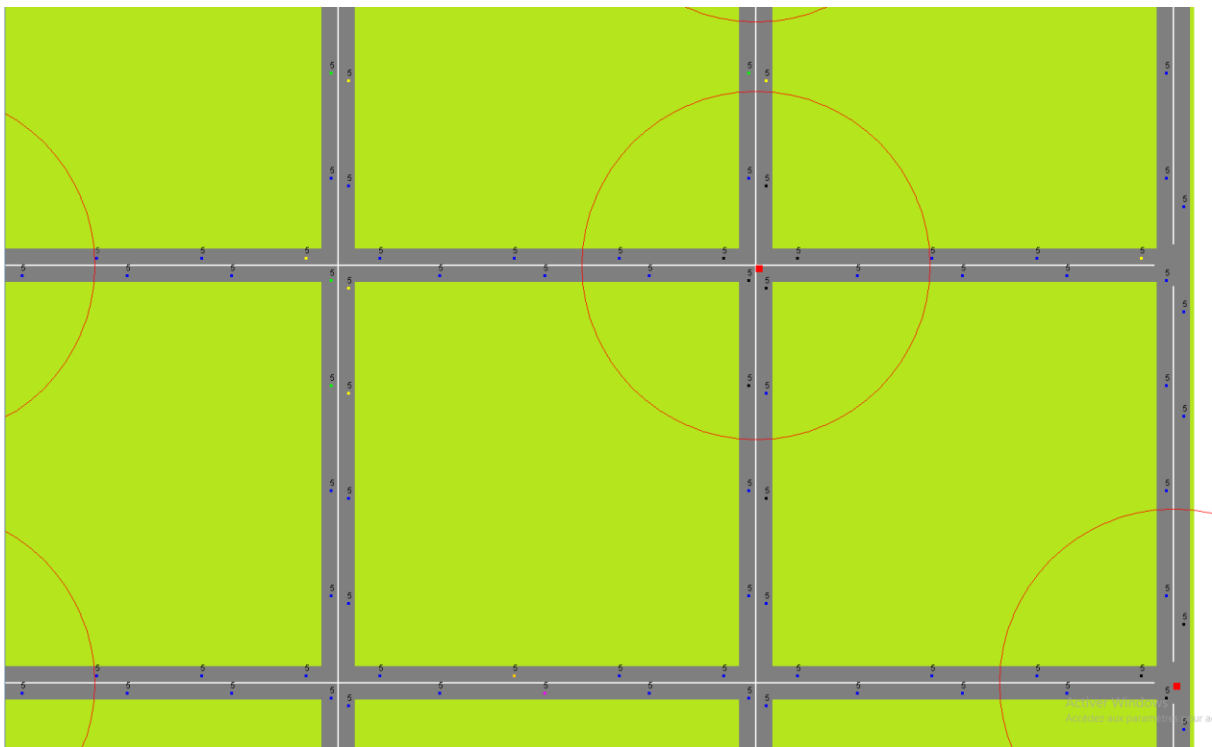


FIGURE 70 - EXTRAIT D'UNE SIMULATION

II - Simulation

La Figure 70 est une capture d'écran du simulateur en question. Les véhicules sont introduits et circulent de façon à respecter l'inter-distance nécessaire à une répartition homogène dans la topologie.

Le simulateur implémente les portées de communication et la mobilité des véhicules est prise en charge. Il est à temps discret et l'unité de temps est de 100ms. A l'échéance, le noyau du simulateur se réveille, déplace les véhicules en fonction du modèle de mobilité retenu et procède aux échanges des messages ICN entre véhicules et entre UBRs et véhicules, en prenant en compte la portée radio.

Au cours d'un scénario, les UBRs sont placées de façon aléatoire sur les carrefours. A l'initialisation, nous laissons s'écouler un certain temps, pour permettre aux véhicules de se mettre en place. Une fois ce temps écoulé, les UBRs commencent à émettre des données à intervalle régulier, pendant 10 secondes. A la fin des 10 secondes, nous relevons les résultats concernant la quantité d'informations échangées et la réception des données. Puis l'opération est répétée 20 fois, à l'issue de quoi nous faisons une moyenne sur les 20 exécutions

Les simulations sont exécutées sur un ordinateur dont les caractéristiques sont décrites dans le Tableau 14.

TABLEAU 14 – CONFIGURATION DE L'ORDINATEUR POUR LA SIMULATION

Processeur	i7 @3.6GHZ
Mémoire vive	32Go
Système d'exploitation	Windows 10 x64
Carte graphique	Intel HD Graphics 630
IDE	Eclipse neon 3
Langage de programmation	Java 8

Pour des raisons de performance, nous sommes obligés de rester à une petite échelle, or nous avons constaté que l'approche que nous proposons était plus efficace quand nous augmentons les valeurs des variables. Cependant, nous arrivons à distinguer les tendances des courbes.

1. Charge

Dans les graphiques suivants, nous omettons certaines données de l'approche VCDN et l'ensemble des données liées au NDN afin de laisser apparaître les tendances des autres approches.

D'abord, dans cette comparaison, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 15.

TABLEAU 15 - CHARGE RESEAU SIMULATION - NOMBRE DE VEHICULES

β	D
8	4

Sur la Figure 71, on constate, comme pour le modèle analytique, qu’augmenter le nombre de véhicules débouche sur un augmentation de la consommation de la bande passante pour chacune des approches, mais à des degrés différents.

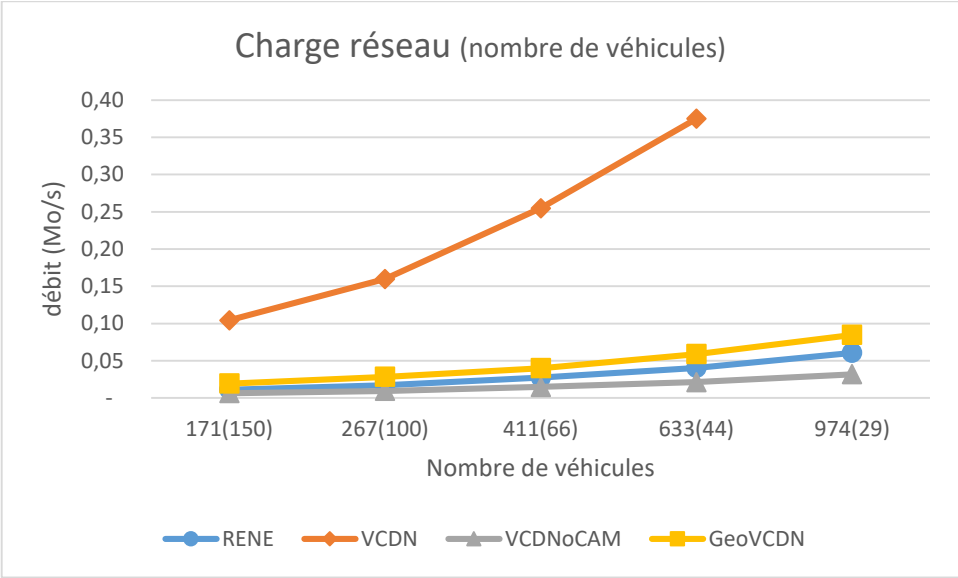


FIGURE 71 - CHARGE RESEAU SIMULATION - NOMBRE DE VEHICULES

Dans cette figure, nous ne distinguons pas la tendance entre GeoVCDN et RENE et nous proposons donc de comparer, sur la même échelle, la représentation de RENE en pourcentage de GeoVCDN :

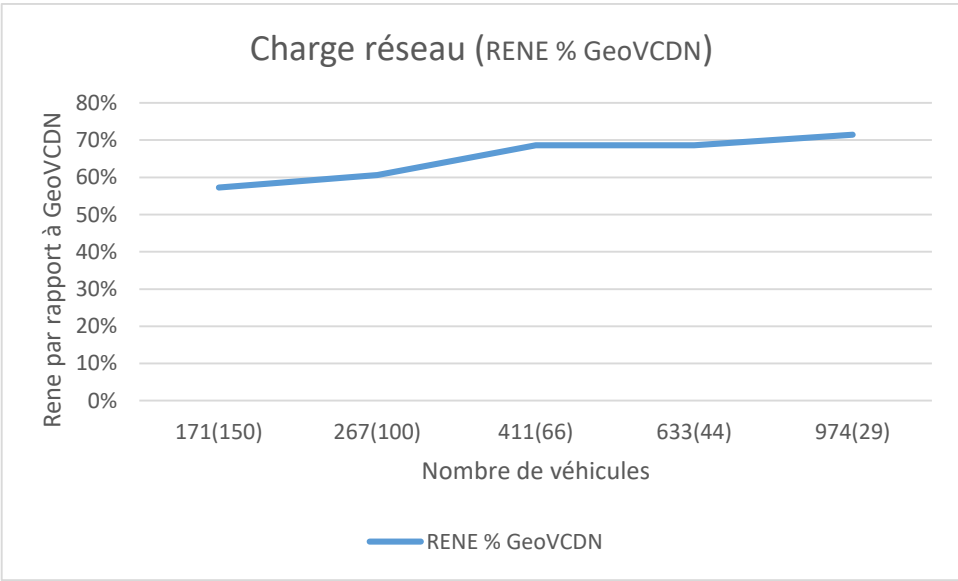


FIGURE 72 - CHARGE RESEAU SIMULATION - RENE % GEOVCDN

II - Simulation

Sur la Figure 72, comme nous l'avons déjà constaté sur les modèles analytiques, nous voyons que GeoVCDN consomme plus de bande passante que RENE, mais que l'écart tend à se réduire. Cela est dû au fait qu'avec l'approche RENE, plus il y a de véhicules, plus il y a d'informations échangées. Avec l'approche GeoVCDN, en revanche, le nombre de véhicules à l'intérieur de la zone n'a aucun impact sur la quantité d'informations échangées, puisque nous envoyons un paquet pour l'ensemble des véhicules. Par conséquent, avec l'augmentation du nombre de véhicules, GeoVCDN croît moins vite que l'approche RENE.

Afin de valider les modèles proposés, on présente dans le Tableau 16 une comparaison entre les valeurs attendues et les valeurs observées pour 411 véhicules :

TABLEAU 16 - COMPARAISON CHARGE RESEAU (NOMBRE DE VEHICULES)

Approche	Valeur attendue (modèle analytique) en octets par seconde	Valeur observée (simulation) en octets par seconde	Différence
NDN	3 507 200,00	3 712 913,93	+5,86%
RENE	27 021,88	27 453,20	+1,5%
VCDN	252 934,88	254 684,26	+0,69%
VCDNoCAM	14 596,58	14 678,86	+0,56%
GeoVCDN	40 220,45	39 822,73	-0,99%

Nous constatons des écarts de l'ordre de 1%, ce qui confirme le comportement de nos modèles en ce qui concerne le nombre de véhicules.

De même, pour la comparaison par rapport au nombre d'UBRs, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 17.

TABLEAU 17 - CHARGE RESEAU SIMULATION - NOMBRE D'UBRS

α	R_c	P_{UBR}
267	300	250

L'augmentation du nombre d'UBRs est bénéfique pour l'approche GeoVCDN, comme nous pouvons le voir sur la Figure 73. Dans un premier temps, plus il y a d'UBRs, plus il y a de foyers « d'infection » pour les échanges de véhicules à véhicules. Dans un deuxième temps, quand il y a assez d'UBRs, les échanges entre véhicules se réduisent, car les UBRs couvrent déjà une partie du réseau.

Pour les autres approches, augmenter le nombre d'UBRs, c'est augmenter le nombre de véhicules qui envoient des intérêts et le nombre de données envoyés.

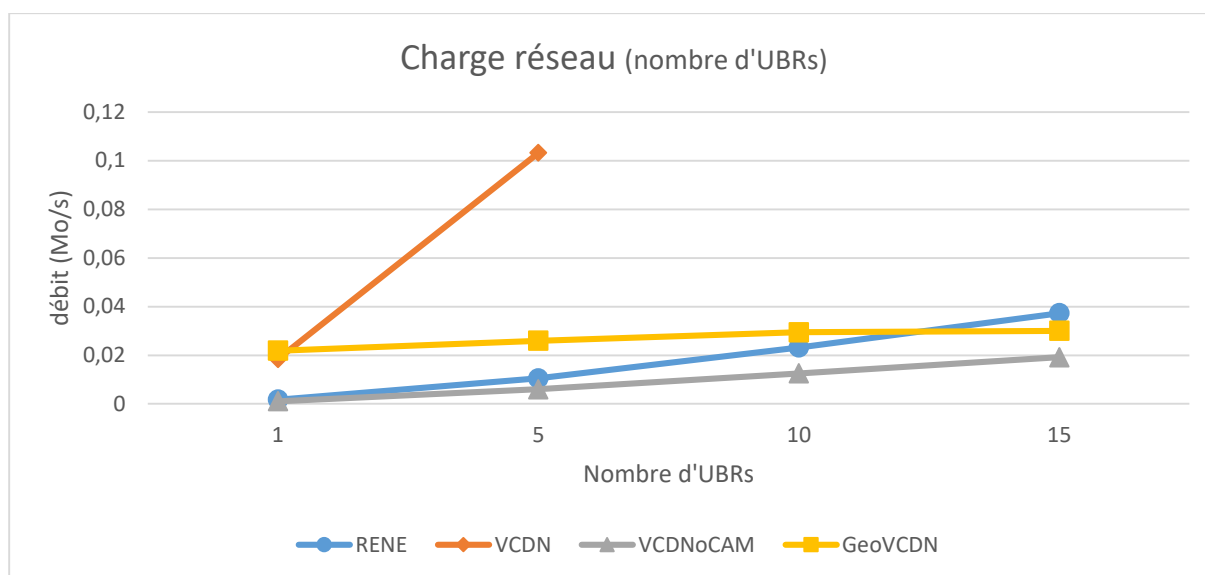


FIGURE 73 - CHARGE RESEAU SIMULATION - NOMBRE D'UBRS

De même, le Tableau 18 présente la comparaison entre les valeurs attendues et les valeurs observées pour 5 UBR :

TABLEAU 18 - COMPARAISON CHARGE RESEAU (NOMBRE D'UBRS)

Approche	Valeur attendue (modèle analytique) en octets par seconde	Valeur observée (simulation) en octets par seconde	Différence
NDN	890 000,00	956 702,13	+7,49%
RENE	10 298,97	10 469,80	+1,69%
VCDN	100 984,96	103 212,86	+2,2%
VCDNoCAM	5 889,28	6 001,46	+1,9%
GeoVCDN	27 812,43	25 939,06	-6,73%

Dans le Tableau 18, pour les approches RENE, VCDN et VCDNoCAM, l'écart entre les résultats de la simulation et ceux des modèles analytiques est d'environ 2%. Pour l'approche NDN, la simulation donne un résultat plus élevé de 7%, tandis que pour GeoVCDN la simulation donne un résultat en-dessous de celui attendu par le modèle analytique, à hauteur de 7%, ce qui conforte le modèle analytique concernant l'évolution du nombre d'UBRs.

Ensuite, pour la comparaison par rapport à la portée des UBRs, certains paramètres ne sont, de nouveau, pas dans leur état par défaut. Les variations sont précisées dans le Tableau 19.

TABLEAU 19 - CHARGE RESEAU SIMULATION - PORTEE DES UBRs

β	α	D
12	267	4

II - Simulation

L'augmentation de la portée des UBRs a eu l'effet attendu. Comme le montre la Figure 74, pour l'approche GeoVCDN, en augmentant la portée des UBRs, nous réduisons les échanges de véhicules à véhicules, d'où cet effet positif sur la charge réseau. Pour les autres approches, nous augmentons le nombre de véhicules présents dans la zone de couverture et donc concernés par les échanges ICN.

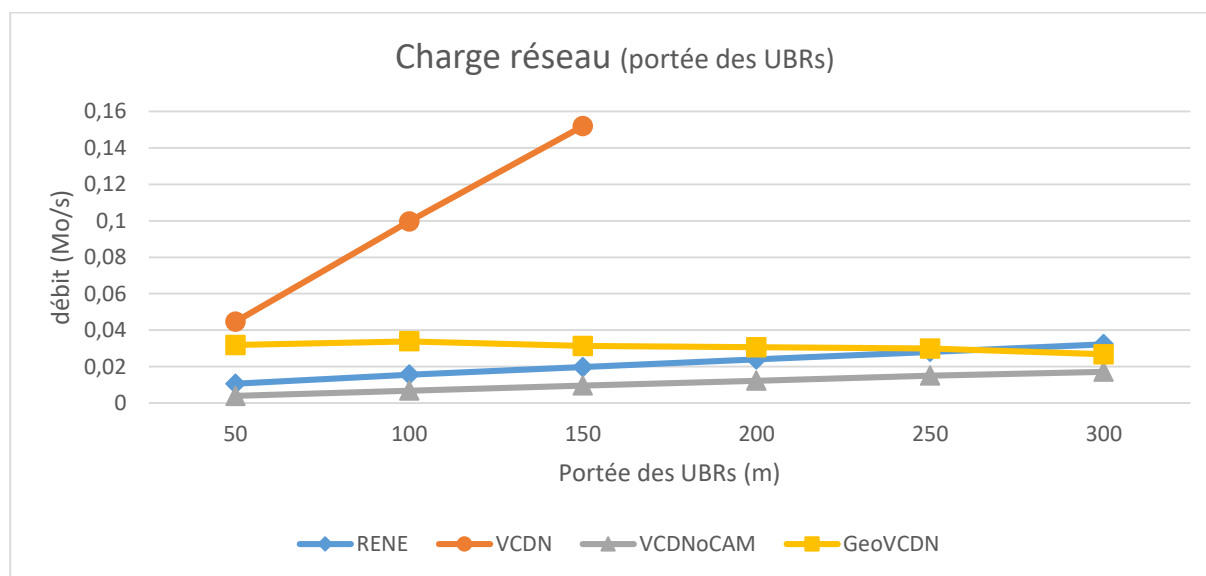


FIGURE 74 - CHARGE RESEAU SIMULATION - PORTEE DES UBRs

Le Tableau 20 présente une comparaison des valeurs attendues et des valeurs observées pour une portée de 150 mètres :

TABLEAU 20 - COMPARAISON CHARGE RESEAU (PORTEE DES UBRs)

Approche	Valeur attendue (modèle analytique) en octets par seconde	Valeur observée (simulation) en octets par seconde	Différence
NDN	3 075 840,00	3 240 097,20	+5,34%
RENE	19 672,56	19 734,53	+0,32%
VCDN	151 559,88	151 847,26	+0,19%
VCDNoCAM	9 709,23	9 680,13	-0,30%
GeoVCDN	30 725,85	31 360,93	+2,07%

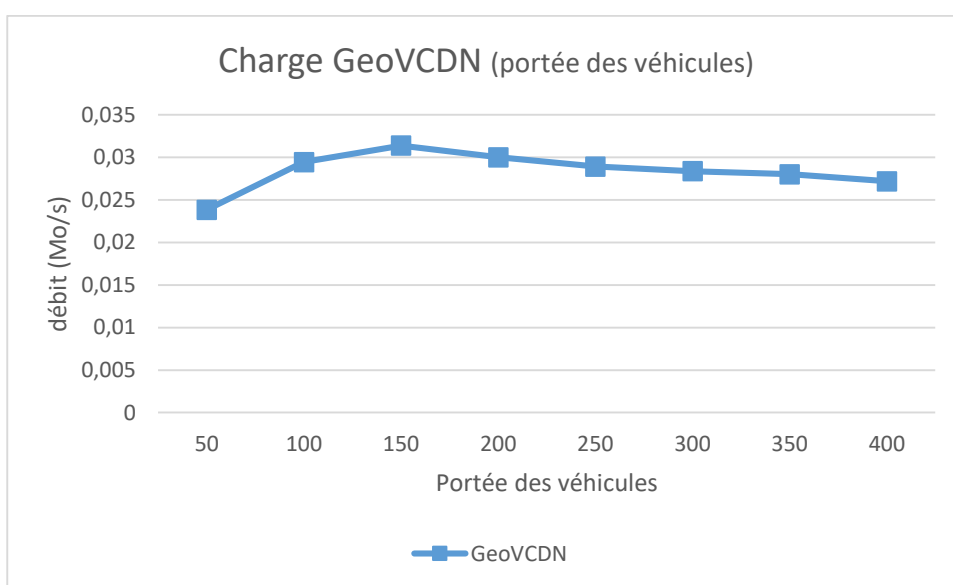
Encore une fois, le tableau confirme le modèle analytique pour la portée des UBR. En effet, on peut constater que la différence entre le modèle analytique et la simulation est faible, avec des écarts entre 0 et 2%. Cependant, l'approche NDN présente un plus grand écart (5%), mais celui n'a pas d'incidence sur les comparaisons avec les autres approches.

D'autre part, pour la comparaison concernant la portée des véhicules, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 21.

TABEAU 21 - CHARGE RESEAU SIMULATION - PORTEE DES VEHICULES

β	D	α
12	4	267

On a déjà vu, que la portée de transmission des véhicules n'avait d'impact que sur GeoVCDN. La forme de la courbe, observée avec le modèle analytique, est similaire à celle obtenue en simulation (Figure 75). Nous avons une augmentation plus souple, entre 50 et 100, qui correspond à la phase où l'inter-distance entre les véhicules est presque suffisante, pour que tous les véhicules soient connectés. Dans cette simulation, l'inter-distance était de 107 mètres. Après le passage de ce seuil, nous observons la faible descente que nous observions déjà avec le modèle analytique (Figure 58).

**FIGURE 75 - CHARGE RESEAU SIMULATION - PORTEE DES VEHICULES - GEOVCDN**

Voici un extrait de comparaison entre les modèles analytiques et la simulation pour une portée de communication des véhicules de 150 mètres :

TABEAU 22 - COMPARAISON CHARGE RESEAU (PORTEE DES VEHICULES)

Approche	Valeur attendue (modèle analytique) en octets par seconde	Valeur observée (simulation) en octets par seconde	Différence
NDN	5 126 400,00	5 422 778,06	+6%
RENE	28 483,56	28 486,66	+0,01%
VCDN	251 951,88	253 434,53	+0,59%
VCDNoCAM	15 049,23	15 079,13	+0,20%
GeoVCDN	31 614,90	31 383	-0,73%

II - Simulation

Dans le Tableau 22, l'écart entre le modèle analytique et la simulation est très faible (de l'ordre de 0.5%), excepté, encore une fois, pour l'approche NDN qui présente un écart plus grand. Ceci confirme les modèles analytiques en ce qui concerne l'impact de la portée des véhicules sur la bande passante.

Enfin, avec l'augmentation de la fréquence de génération de la donnée, nous augmentons la quantité des données qui circulent sur le réseau. Dans la Figure 76, nous constatons qu'avec une approche de type RENE, l'augmentation est linéaire, proportionnelle à l'augmentation de la fréquence. Avec l'approche GeoVCDN, la croissance existe mais la pente est moins forte, puisqu'entre deux intervalles de génération, nous répétons parfois l'envoi de celle-ci, contrairement à l'approche RENE, où l'envoi est ponctuel au moment de la génération.

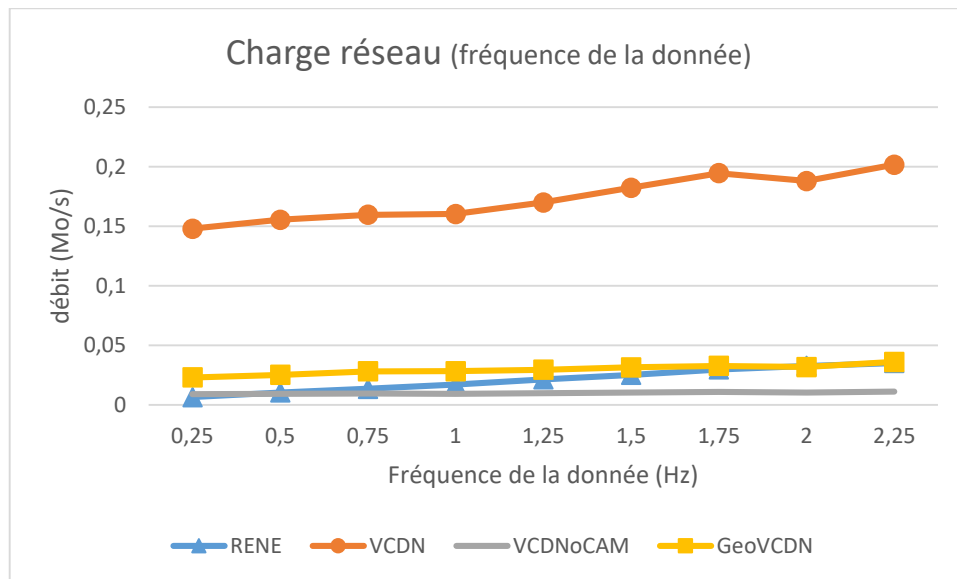


FIGURE 76 - CHARGE RESEAU SIMULATION - FREQUENCE DE LA DONNEE

Dans cette section, nous avons présenté les résultats obtenus avec notre simulation en termes d'occupation de la bande passante du canal G5 pour chacune des cinq approches (NDN, RENE, VCDN, VCDNoCAM, GEOVCDN). Afin de les comparer, nous présentons leurs évaluations en fonction des différentes variables du modèle, sous forme de graphes. Pour l'approche NDN, le résultat de la simulation est supérieur d'environ 5% à celui du modèle. Cette différence n'a aucun impact sur la comparaison des modèles puisque l'approche NDN était déjà, de très loin, supérieure aux autres. Pour les autres approches, la différence entre les valeurs attendues (modèle analytique) et les valeurs obtenues (simulation) est de l'ordre de 1%, ce qui valide le modèle analytique concernant la charge du réseau. De plus, même à petite échelle, nous voyons déjà apparaître les écarts et les tendances des courbes entre les différentes approches. Ainsi, nous notons que nos contributions, GeoVCDN et VCDNoCAM, font mieux que les autres approches en termes d'occupation de la bande passante.

2. Fraîcheur

Nous proposons maintenant, une comparaison par rapport à la fraîcheur réseau.

D'abord, pour la comparaison par rapport à l'évolution de la portée des UBRs, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 23.

TABLEAU 23 - FRAICHEUR RESEAU SIMULATION – PORTEE DES UBRs

β	α	D
12	267	4

Plus nous augmentons la portée des UBRs, plus nous réduisons la distance que doivent parcourir les messages à travers les échanges de client à client ; ce qui est bénéfique à l'approche GeoVCDN, jusqu'à atteindre 100% de fraîcheur, comme l'illustre la Figure 77.

Pour les autres approches, l'augmentation de la portée des UBRs est bénéfique également, car nous augmentons la zone de couverture et par conséquent le pourcentage de véhicules qui s'y trouvent.

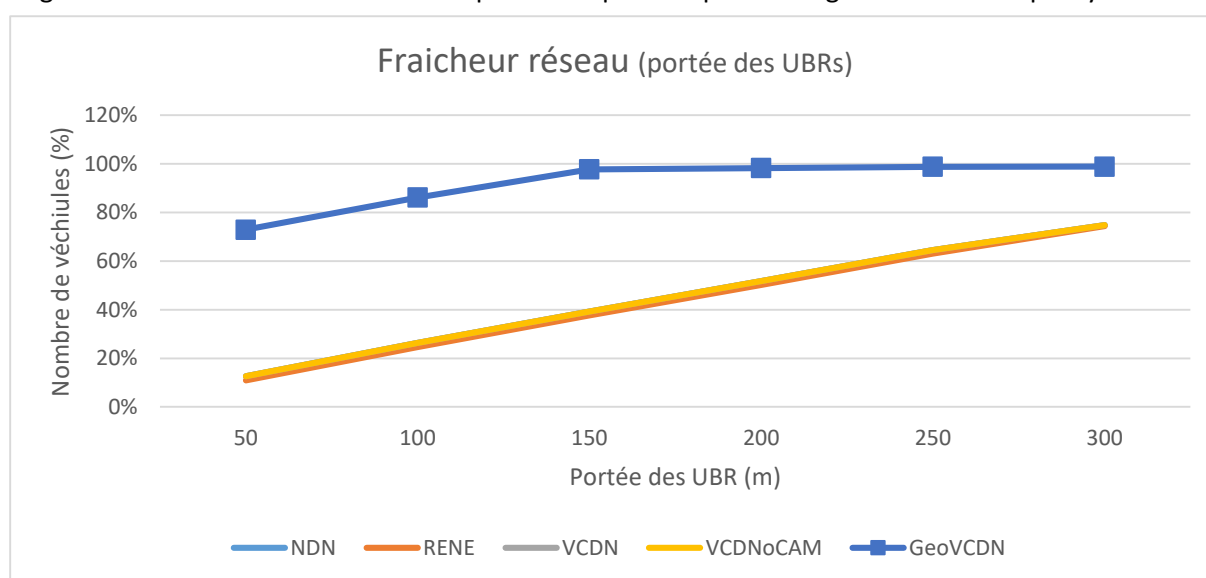


FIGURE 77 - FRAICHEUR RESEAU SIMULATION - PORTEE DES UBRs

Une comparaison entre les valeurs attendues et les valeurs observées pour une portée des UBR de 100 mètres est présentée dans le Tableau 24.

TABEAU 24 - COMPARAISON DE LA FRAICHEUR RESEAU (PORTEE DES UBR)

Approche	Valeur attendue (modèle analytique) en octets	Valeur observée (simulation) en octets	Différence
NDN	26,63%	26%	-0,63 pt
RENE	25%	25%	0 pt
VCDN	26,63%	26%	-0,63 pt
VCDNoCAM	26,63%	26%	-0,63 pt
GeoVCDN	95%	86%	-9 pt

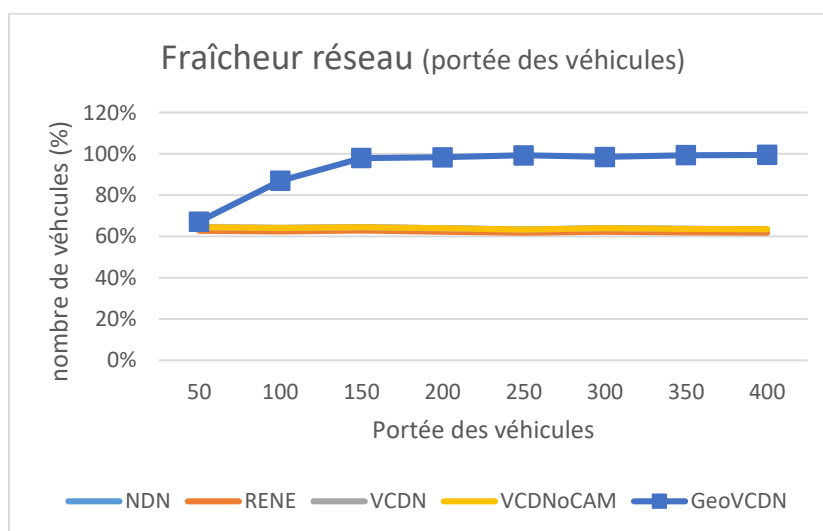
Dans le Tableau 24, pour les approches NDN, RENE, VCDN et VCDNoCAM, l'écart de résultat est quasiment nul. Pour GeoVCDN, la simulation donne un résultat de 9 points inférieur au modèle analytique, mais reste tout de même bien au-dessus des autres approches. Ces résultats confirment les modèles analytiques concernant la « fraîcheur » du réseau pour la portée des UBRs.

Ensuite, pour la comparaison sur la portée des véhicules, certains paramètres ne sont de nouveau pas dans leur état par défaut. Les variations sont précisées dans le Tableau 25.

TABEAU 25 - FRAICHEUR RESEAU SIMULATION - PORTEE DES VEHICULES

β	D	α
12	4	267

La portée de communication des véhicules n'a d'impact que sur l'approche GeoVCDN, comme nous pouvons le constater sur la Figure 78. Avec une portée inférieure à l'inter-distance des véhicules, nous faisons tout de même jusqu'à 35% mieux que les autres approches, grâce au double sens de circulation. Plus nous augmentons la portée, plus nous augmentons, également, la vitesse de diffusion de l'information de client à client ; jusqu'à atteindre la fraîcheur optimale du réseau, si les autres paramètres le permettent.

**FIGURE 78 - FRAICHEUR RESEAU EN SIMULATION - PORTEE DES VEHICULES**

Le Tableau 26 présente une comparaison entre les valeurs attendues et les valeurs observées pour une portée des véhicules de 150 mètres :

TABLEAU 26 - COMPARAISON FRAICHEUR RESEAU (PORTEE DES VEHICULES)

Approche	Valeur attendue (modèle analytique) en octets	Valeur observée (simulation) en octets	Différence
NDN	64,13%	64%	-0,13 pt
RENE	62,50%	63%	+0,5 pt
VCDN	64,13%	64%	-0,13 pt
VCDNoCAM	64,13%	64%	-0,13 pt
GeoVCDN	95%	98%	+3 pt

Les différences entre les valeurs obtenues grâce aux modèles et celles relevées en simulation sont très proches (au maximum 3 points de différence). Ce qui valide les modèles analytiques concernant l'impact de la portée des véhicules sur la « fraîcheur » du réseau.

De même, pour la comparaison sur le nombre d'UBRs, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 27.

TABLEAU 27 - FRAICHEUR RESEAU SIMULATION - NOMBRE D'UBRS

α	D
267	4

Etant donné la répartition de la circulation, l'augmentation du nombre d'UBRs fait croître de façon linéaire la zone de couverture et le nombre de véhicules couverts. Pour notre approche GeoVCDN, c'est différent, l'augmentation du nombre d'UBRs augmente le nombre de véhicules capables de transmettre la donnée. C'est pourquoi la courbe sur la Figure 79 a une allure logarithmique.

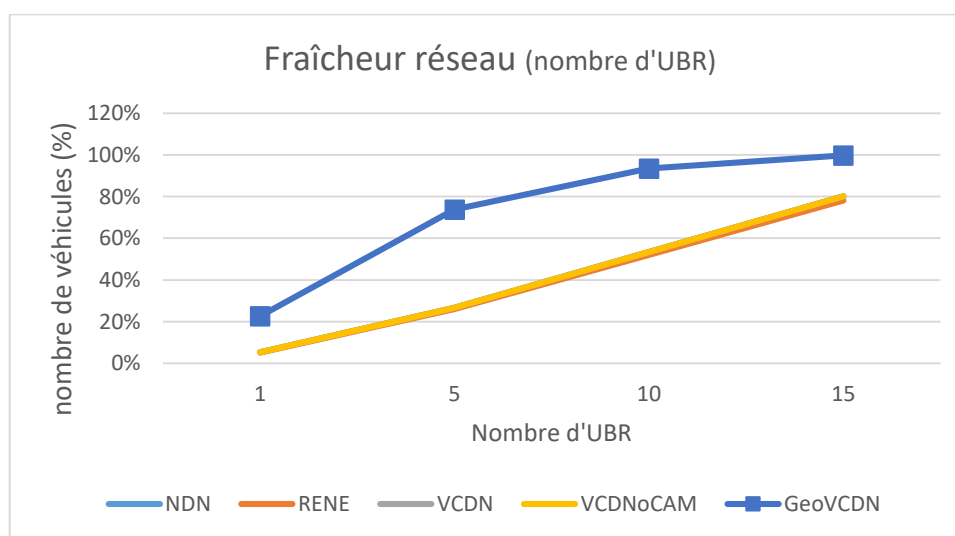


FIGURE 79 - FRAICHEUR RESEAU EN SIMULATION - NOMBRE D'UBRS

Une comparaison entre les valeurs attendues et les valeurs observées pour 5 UBRs est disponible dans le Tableau 28.

TABLEAU 28 – COMPARAISON DE LA FRAICHEUR RESEAU (NOMBRE D'UBR)

Approche	Valeur attendue (modèle analytique) en octets	Valeur observée (simulation) en octets	Différence
NDN	26,72%	27%	+0,28 pt
RENE	26,04%	26%	-0,4 pt
VCDN	26,72%	27%	+0,28 pt
VCDNoCAM	26,72%	27%	+0,28 pt
GeoVCDN	75,1%	74%	-1,1 pt

Dans le Tableau 28, on valide les modèles analytiques pour le nombre d'UBRs concernant la fraîcheur réseau. En effet, l'écart entre modèle et simulation est très faible (entre 0 et 1,1 point). Les différences entre chaque approche sont confirmées, ce qui atteste une fois de plus la validité des modèles analytiques.

De la même façon, pour la comparaison par rapport au nombre de véhicules, certains paramètres ne sont pas dans leur état par défaut. Les variations sont précisées dans le Tableau 29.

TABLEAU 29 - FRAICHEUR RESEAU SIMULATION - NOMBRE DE VEHICULES

β	D
8	4

Comme le montre la Figure 80, augmenter le nombre de véhicules n'a aucun impact sur les approches qui ne gèrent par les communications entre clients. Le pourcentage de véhicules dans la zone de couverture et qui pénètrent la zone de couverture entre deux générations d'une donnée, reste le même et par conséquent le taux de fraîcheur aussi. Pour l'approche GeoVCDN, l'augmentation du nombre de véhicules a un impact positif sur la fraîcheur, cela s'explique par l'optimisation de l'envoi d'un message. Plus il y a de véhicules autour de soi, plus notre voisin le plus éloigné se trouve proche du bord de la portée de transmission, la donnée parcourt donc une plus grande distance.

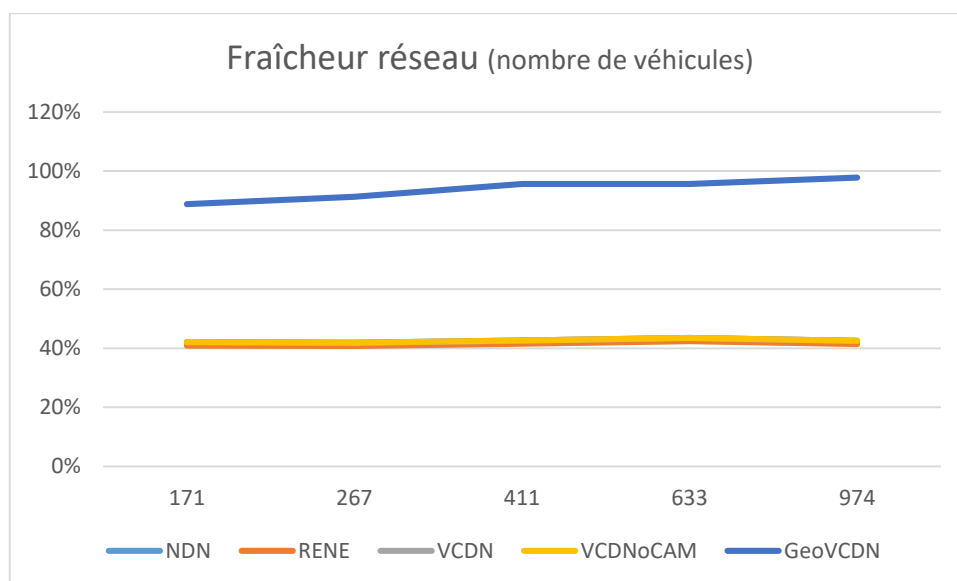


FIGURE 80 - FRAICHEUR RESEAU EN SIMULATION - NOMBRE DE VEHICULES

La comparaison entre les valeurs attendues et les valeurs observées pour 267 véhicules est résumée dans le Tableau 30.

TABLEAU 30 – COMPARAISON DE LA FRAICHEUR RESEAU (NOMBRE DE VEHICULES)

Approche	Valeur attendue (modèle analytique) en octets	Valeur observée (simulation) en octets	Différence
NDN	42,75%	42%	-0,75 pt
RENE	41,67%	41%	-0,67 pt
VCDN	42,75%	42%	-0,75 pt
VCDNoCAM	42,75%	42%	-0,75 pt
GeoVCDN	93,45%	91%	-2,45 pt

Le Tableau 30 montre, une fois de plus, que l'écart entre les simulations et le modèle analytique est très faible (entre 0 et 2,45 point). Ces résultats confirment la validité du modèle analytique quand on fait évoluer le nombre de véhicules.

Enfin, plus une donnée est envoyée fréquemment, moins nous laissons le temps aux véhicules de transmettre celle-ci à leur tour. C'est pourquoi, comme nous pouvons le voir sur la Figure 81, la fraîcheur diminue pour l'approche GeoVCDN. Pour l'approche RENE, la fréquence n'a aucun impact ; nous devrions observer une courbe constante. Pour les autres approches, l'impact est négatif mais se fait faiblement ressentir : si la donnée est envoyée toutes les 500ms au lieu de 1000ms, moins de véhicules auront eu le temps de pénétrer la zone de couverture. Cependant, le nombre de véhicules qui rentrent dans cette zone, entre deux générations de données, ne représentent qu'une partie mineure de l'ensemble des véhicules qui reçoivent l'information.

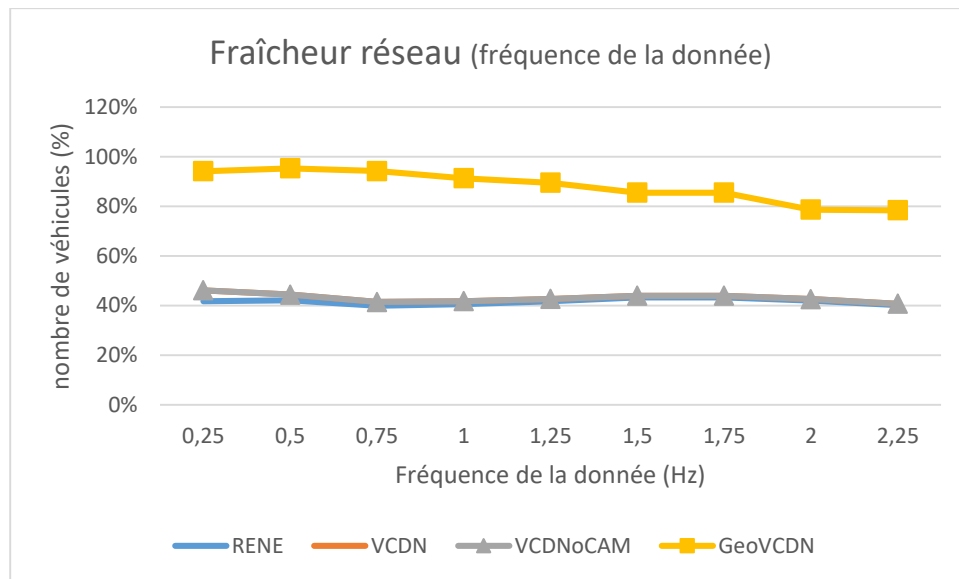


FIGURE 81 - FRAICHEUR RESEAU EN SIMULATION - FREQUENCE DE LA DONNEE

3. Contribution à la recherche du meilleur chemin

La recherche du meilleur chemin d'un véhicule (aussi connu sous le terme de Pathfinding [66]) décrit sa capacité à trouver le chemin correspondant au mieux aux exigences du conducteur. La plupart du temps, cette exigence est d'arriver le plus rapidement possible. C'est d'ailleurs ce critère qui dirigera notre contribution. Même si celui-ci est omniprésent, il ne faut pas oublier qu'il en existe une multitude d'autres : coût, paysage, temps à l'arrêt, luminosité, etc. Parmi ces informations, certaines peuvent être obtenues par des capteurs et ainsi rentrer dans le cadre de notre étude.

Dans cette partie, nous proposons d'appliquer notre méthode d'échanges de données, dans un Cloud de contexte, au problème de recherche du meilleur chemin. Pour cela, nous proposons une simulation qui consiste à faire évoluer l'état des feux d'une ville de sorte à modifier le plus court chemin pour la traverser.

Le simulateur utilisé est le même que celui présenté dans la Figure 70 : Il est à l'image d'une ville de type Manhattan et des véhicules sont répartis de façon homogène.

Dans un premier temps, nous plaçons des feux tricolores et des UBRs sur un nombre fixe de 5 carrefours. Un feu peut être dans deux états : rapide ou lent. S'il est dans l'état rapide, il alterne souvent entre rouge et vert (toutes les 5 secondes) et s'il est dans l'état lent, l'alternance se fait toutes les 15 secondes.

Puis nous sélectionnons aléatoirement un ou plusieurs carrefours que nous équipons également de feux et d'UBRs.

Au début de la simulation, nous insérons 3 véhicules supplémentaires. L'un utilise la méthode ICN de type NDN, un autre de type RENE et le dernier utilise l'approche que nous propose : le GeoVCDN. Dans leur état initial, chaque véhicule a connaissance de la configuration et de l'état des feux de la ville et

exécute le même algorithme de PathFinding obtenant ainsi le même résultat.

Après 2 secondes, chaque feu inverse son état, passant de rapide à lent et de lent à rapide, et ayant pour conséquence de faire changer le chemin le plus court. Un message ICN est alors émis, pour informer de cette modification. Au moment du changement, si les véhicules se trouvent à portée d'une UBR, alors chacun reçoit l'information et ré-exécute son algorithme de recherche du plus court chemin. S'ils ne sont pas à la portée d'une UBR, le véhicule qui communique avec l'approche RENE ne recevra jamais l'information, le véhicule en NDN la recevra s'il passe à portée d'un feu et celui en GeoVCDN recevra l'information grâce aux autres véhicules.

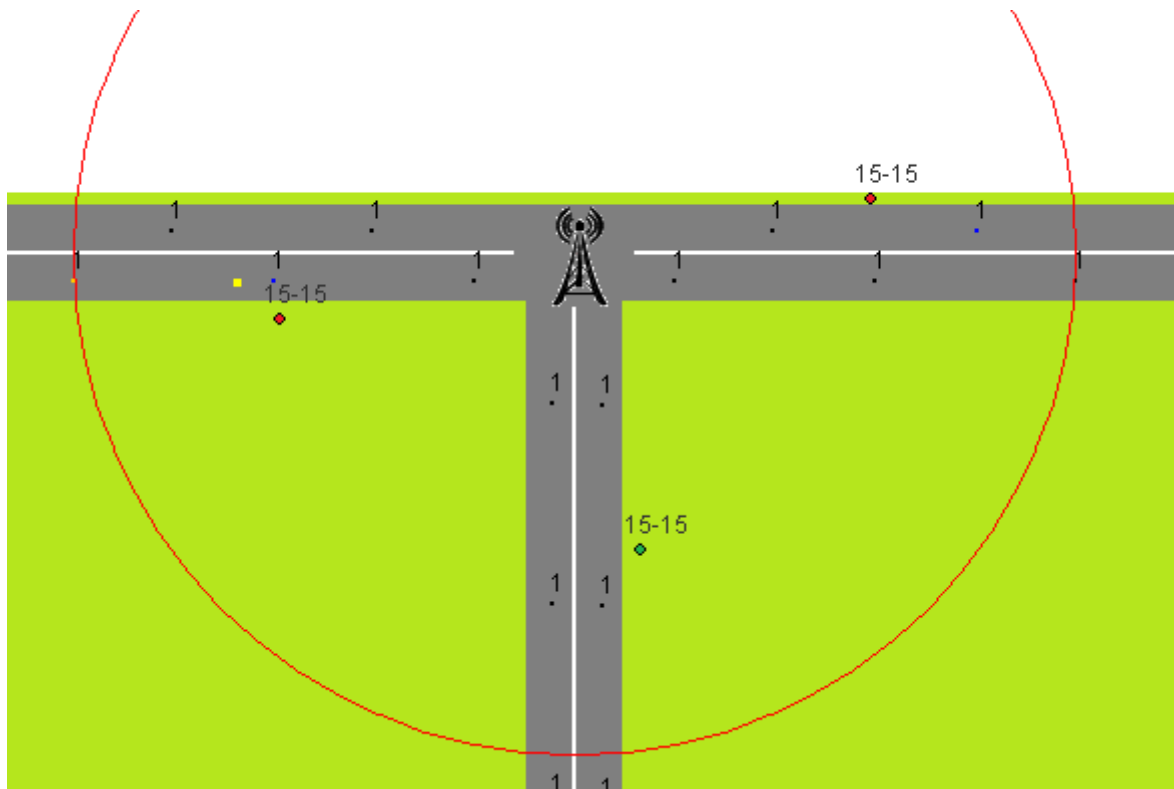


FIGURE 82 - CONFIGURATION D'UN CARREFOUR

La Figure 82 est extraite du simulateur et représente la configuration d'un carrefour. Une UBR est placée en son centre, des feux tricolores sont placés à chaque entrée du carrefour et les feux sur une même ligne droite sont synchronisés. Le cercle rouge représente la portée de communication de l'UBR. Le point jaune représente un véhicule qui implémente notre approche GeoVCDN.

Au début de chaque simulation, nous plaçons aléatoirement des UBRs en plus des 5 fixes. Dans la première configuration, une seule UBR est placée aléatoirement. Dans la seconde, ce sont deux UBRs qui sont placées aléatoirement, puis nous montons à 8 UBRs pour la troisième et pour la dernière configuration nous plaçons des UBRs à tous les carrefours.

Chaque configuration est exécutée 100 fois (et le placement change à chacune des exécutions). A la fin, nous mesurons le temps qu'un véhicule a pris pour traverser la ville.

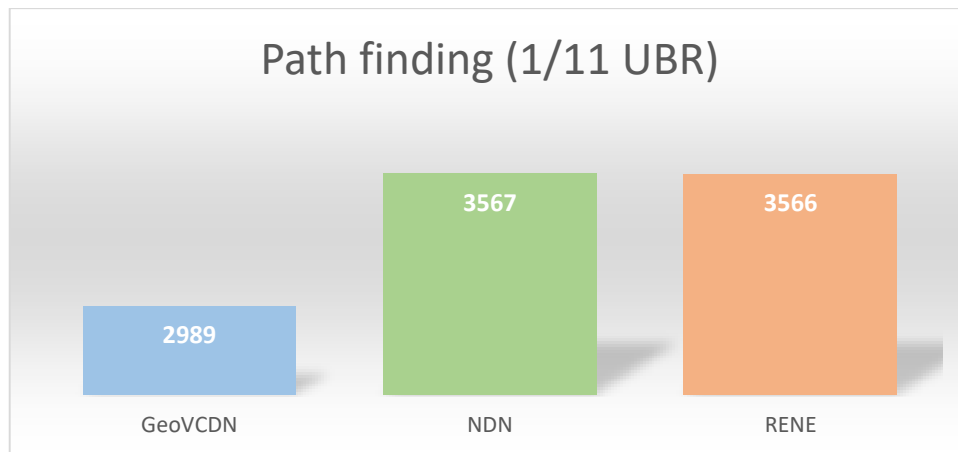


FIGURE 83 - PATH FINDING AVEC 1 UBR ALEATOIRE

La Figure 83 présente les résultats de la première configuration. Le véhicule communiquant avec l'approche GeoVCDN a passé 2989 secondes sur la route tandis que les véhicules avec NDN ou RENE y ont passé 3566 secondes, ce qui représente une amélioration de 17%.

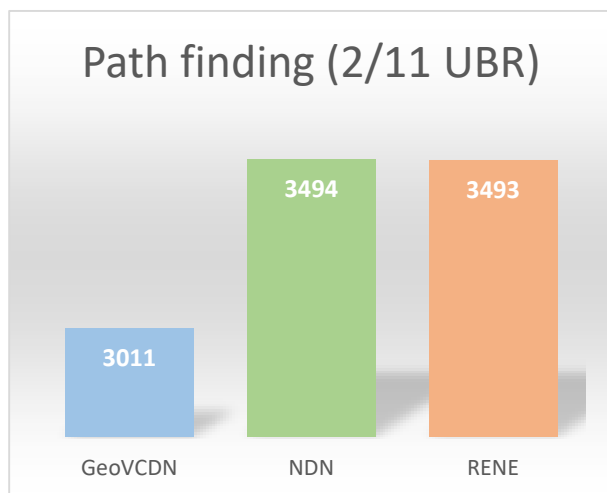


FIGURE 85 - PATH FINDING AVEC 2 UBRs

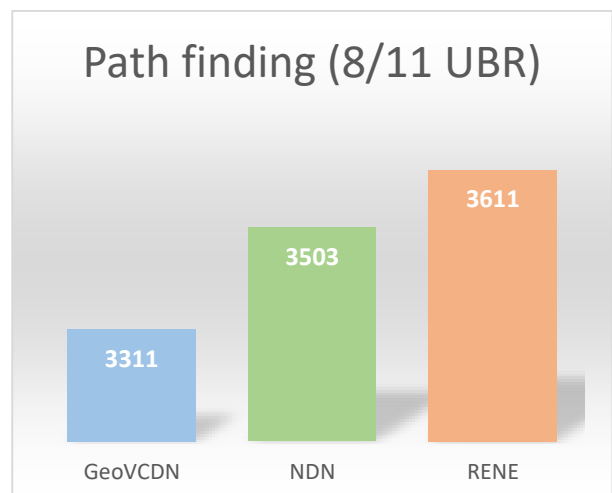


FIGURE 84 - PATH FINDING AVEC 8 UBRs

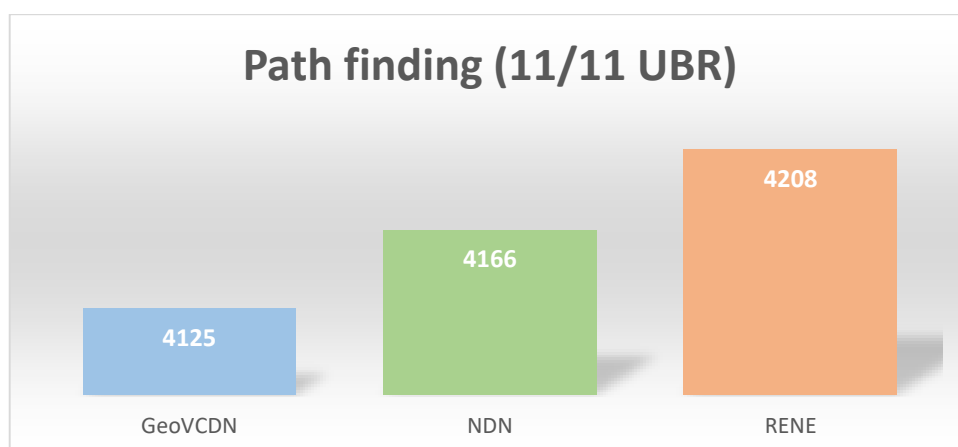


FIGURE 86 - PATH FINDING AVEC 11 UBRs

La Figure 84, la Figure 85 et la Figure 86 sont les résultats des autres configurations. Nous constatons que plus nous mettons d'UBRs, plus les chiffres tendent à se rapprocher. Avec des UBRs partout, nous ne sommes plus qu'à 1 ou 2 % de différence. Avec peu d'UBRs, le pathfinding initial avait pour conséquence d'éviter les feux pour gagner du temps. L'ironie de la situation étant qu'en évitant les feux, ils se privent de la possibilité d'obtenir les mises à jour de la ville et par conséquent, ils perdent du temps. En augmentant le nombre d'UBRs, nous augmentons par la même occasion les chances que les véhicules avec NDN ou RENE obtiennent l'information, puisqu'il devient plus compliqué de trouver un chemin sans feux tricolores.

III – Discussions

La Tableau 31 présente une évaluation des résultats obtenus pour chaque approche, sur les deux indicateurs de performance qu'on a utilisé, à savoir la charge réseau et la fraîcheur réseau.

TABLEAU 31 - RESULTATS DES APPROCHES

	Paramètres	NDN	RENE	VCDN	VCDNoCAM	GeoVCDN
Charge réseau	Nombre de véhicules	Très élevée	Moyenne	Elevée	Très basse	Basse
	Longueur d'un segment	Très élevée	Basse	Elevée	Très basse	Moyenne
	Portée des UBRs	Très élevée	Moyenne	Elevée	Très basse	Moyenne
	Couverture	Très élevée	Moyenne	Elevée	Très basse	Basse
	Portée des véhicules	Très élevée	Basse	Elevée	Très basse	Basse
	Nombre d'UBRs	Très élevée	Moyenne	Elevée	Très basse	Très basse
	Fréquence de la donnée	Très élevée	Basse	Elevée	Très basse	Très basse
Fraîcheur réseau	Nombre de véhicules	Mauvaise	Mauvaise	Mauvaise	Mauvaise	Très bonne
	Portée des UBRs	Moyenne	Moyenne	Moyenne	Moyenne	Très bonne
	Portée des véhicules	Mauvaise	Mauvaise	Mauvaise	Mauvaise	Très bonne
	Nombre d'UBRs	Moyenne	Moyenne	Moyenne	Moyenne	Bonne
	Fréquence de la donnée	Mauvaise	Mauvaise	Mauvaise	Mauvaise	Moyenne

Que ce soit sur les modèles théoriques, ou sur les simulations, les résultats s'améliorent au fur et à mesure que nous augmentons la densité ou le nombre de capteurs. L'approche NDN est complètement hors course pour les réseaux véhiculaires, à l'image du VCDN qui fait beaucoup mieux et qui, pourtant, pèse déjà très lourd sur la charge réseau en comparaison avec les trois autres approches. Pour une meilleure fraîcheur du réseau et un coût beaucoup moins élevé, l'approche VCDNoCAM surpasse l'approche RENE, comme nous pouvons l'observer dans le tableau ci-dessus. D'autre part, l'approche GeoVCDN présente une charge réseau un peu plus élevée que pour l'approche VCDNoCAM mais reste tout de même bien meilleure que l'approche RENE. Ce léger surcoût par rapport à VCDNoCAM est largement compensé par le gain en performance sur la fraîcheur du réseau.

En comparant VCDN à NDN, nous constatons qu'une donnée plus grosse mais versionnée a un impact positif sur la charge réseau. En comparant VCDNoCAM et RENE, nous constatons que l'utilisation des CAMs pour assurer les échanges ICN est aussi positive. Avec VCDNoCAM, en termes de fraîcheur de réseau, nous faisons aussi bien qu'une approche NDN et mieux qu'une approche de type RENE, tout cela pour un coût bien inférieur. Notre approche coûte donc plus cher quand il y a peu de véhicules ou peu de capteurs. C'est aussi à ce moment-là que la charge réseau est la plus faible et donc qu'il est moins important de l'optimiser.

Enfin, les valeurs obtenues en simulation sont très proches de celles attendues avec les modèles ; permettant ainsi de les valider.

Pour conclure ce chapitre, nous avons proposé d'illustrer l'efficacité de notre méthode en améliorant la recherche du meilleur chemin d'un véhicule pour traverser une ville. En étant informé de l'évolution de l'état des feux, un véhicule est capable de s'adapter pour éviter ceux qui sont trop longs.

Chapitre 5 : Conclusion et travaux futurs

Conclusion

Dans la première partie de cette thèse, nous avons présenté un état de l'art thématique et chronologique sur les quatre sujets traités dans celle-ci : l'ICN (Information Centric Networking), le VCC (Vehicular Cloud Computing), les villes intelligentes et l'ITS-G5. Ainsi, nous avons pu faire apparaître ce qui définit chacun de ces domaines et les fonctionnalités qui les caractérisent. Pour l'ICN il y a de nombreuses propositions [67] sur les techniques de routage [68], de mise en cache [69], de nommage [70] ou d'accès à la donnée [71]. Ainsi, au fil des ans, des projets ayant vocation à fournir une architecture ICN complète, apparaissent, chacun avec leur spécificité (PURSUIT [72], SAIL [73], COMET [74], CONVERGENCE [75], ANR [76], GreenICN [77], ICE-AR [78], ICN-2020 [79], NDN [80], Mobility First [81], TRIAD [82] [83]). C'est ainsi que nous avons identifié l'absence d'attention sur les technologies de communication des véhicules intelligents et que nous avons proposé une solution adaptée. Pour le VCC, nous avons constaté que les discussions sont toutes orientées sur la théorie et que la mise en pratique est très rare. De plus, le terme est très vaste [84] [85] [86] [87] [88], il peut désigner l'utilisation d'un cloud d'infrastructure par des véhicules aussi bien que la formation d'un cloud dynamique par les véhicules. Ensuite, nous avons présenté les villes intelligentes, les nouveaux services Clouds qui en découlent, tel que Sensing as a Service, et les défis de recherche, tel que de proposer un « framework » permettant de rendre transparente l'hétérogénéité des équipements. Ce qu'il ressort de l'étude de la littérature de ces 3 domaines, c'est qu'ils semblent pratiquement indissociables les uns des autres [89] [19] [90]. Et le VCN [91] (Vehicular Cloud Networking) en est une illustration. La recherche sur ces sujets ne prend pas assez en compte la technologie ITS-G5, que nous présentons dans ce même Chapitre, et qui pose certaines contraintes mais ouvrent également des perspectives.

Dans le Chapitre 2, nous avons proposé une solution Cloud permettant d'améliorer la connectivité des véhicules grâce à une communication hybride sur le réseau G5 et le réseau cellulaire. Dans cette approche, le Cloud, représenté par un serveur distant, collecte les données des véhicules sous la forme de CAM. Nous proposons deux déclinaisons. La première consiste à voir le Cloud comme un SaaS, c'est-à-dire comme une API pour accéder à des événements routiers. En effet, le serveur est capable de détecter des événements grâce aux données qu'il collecte et un véhicule, par l'intermédiaire de son CAM, effectue une requête pour être notifié des événements qui l'intéressent. La deuxième consiste à router les événements détectés par les véhicules et envoyés au Cloud, vers des véhicules plus lointains.

Nous avons proposé cette solution après avoir constaté les problèmes de connectivité qui peuvent exister dans les réseaux véhiculaires, si nous nous contentons de communiquer en G5. Ces problèmes ont été déduits en croisant les standards sur lesquels sont basés les ITS actuellement déployées et les observations qui ont été faites durant les tests dans le cadre des différents projets européens de déploiement des véhicules connectés (SCOOP@F, InterCor, C-Roads).

Enfin, nous avons expérimenté nos deux solutions sur route afin de relever des délais de communication et ainsi déterminer si oui ou non la solution était viable. Pour pouvoir comparer les délais relevés sur route avec les délais associés à l'ITS-G5, nous avons développé un simulateur qui reproduit le comportement des véhicules sur autoroute. Dans ce simulateur, nous avons implémenté

la pile protocolaire de l'ETSI pour les véhicules connectés, sauf pour la couche physique qui est simulée d'après nos relevés. Les simulations ont permis de mettre en évidence des délais de plusieurs secondes, même avec une circulation dense, causés par des absences de connectivité dans le flux de véhicules. On a ainsi obtenu, avec l'ITS-G5, un délai moyen d'environ 1 seconde pour faire parcourir 5km à un message, dans des conditions favorables à la dissémination sur le canal G5. Les solutions Cloud proposées procurent de meilleures performances pour le même service, à savoir 600ms de moyenne pour la solution de type SaaS, avec 91% des messages qui sont délivrés en moins de 250ms, et 169ms pour la solution de type NaaS.

Dans le chapitre 3, nous avons introduit un modèle de ville intelligente sous la forme d'un Cloud de contexte, pour ensuite proposer une solution d'échange de données basée sur le paradigme ICN, intitulé « GeoVehicular Central Data Networking (GeoVCDN) », pour que les véhicules puissent collecter des données de petite taille, générées par un grand nombre de capteurs et dont l'état ne change pas trop souvent. L'approche est basée sur l'utilisation d'un serveur central qui collecte les données des capteurs pour produire une seule donnée, plus conséquente, mais permettant de réduire le nombre d'intérêts qu'envoie un véhicule pour se tenir informé de l'état de tous les capteurs. Par ailleurs, nous avons défini sous quelles conditions il était préférable de regrouper les données en une seule. Nous avons également proposé des modèles théoriques de notre approche et de deux autres approches ICN (NDN et RENE), issues de la littérature, afin de les comparer selon deux critères : la bande passante consommée pour les échanges ICN, pour rendre le même service, et la qualité du service rendu.

Dans le chapitre 4, nous avons mis en évidence que notre approche permettait d'améliorer sans équivoque le premier critère par rapport à l'approche NDN. L'approche RENE se rapproche plus de nos résultats, mais elle est moins adaptée au passage à grande échelle que la nôtre. Quant à la qualité du service, que nous mesurons par le nombre de véhicules ayant reçu la donnée avant son expiration, elle est, en comparaison avec les approches NDN et RENE, bien meilleure avec le GeoVCDN. On a pu constater qu'avec l'approche GeoVCDN on avait, parfois, une charge 3 fois inférieure à celle engendrée par l'approche RENE et une fraîcheur parfois 4 fois supérieure à celle obtenue par les autres approches. Avec VCDNoCAM on fait encore mieux que GeoVCDN en terme de charge réseau mais le gain sur la fraîcheur, par rapport à NDN et RENE est nul ou faible.

D'autre part, nous avons proposé un autre simulateur, que nous avons développé dans le but de vérifier la validité de nos modèles théoriques. Dans la majorité des cas, l'écart de résultat entre le modèle et la simulation est d'environ 1%. De plus, pour GeoVCDN les résultats obtenus par simulation sont meilleurs que ceux obtenus par les modèles, tandis que pour les autres approches c'est l'inverse.

Enfin, pour illustrer notre solution, nous avons proposé une autre simulation orientée, cette fois, sur la recherche du meilleur chemin pour les véhicules. Notre approche, comparée aux approches NDN et RENE, permet d'améliorer de 15 à 20 pourcent le temps passé sur la route, grâce à une réception plus précoce des informations concernant la mise à jour des feux, leur permettant ainsi de modifier leur route si nécessaire.

Travaux futures

Dans le Chapitre 2, les deux solutions proposées collectent une grande quantité de données qui ne sont, pour l'instant, pas exploitées dans leur intégralité. Il serait intéressant de faire appel à des technologies de type Big Data et d'Intelligence Artificielle pour dessiner et faire sortir des modèles de circulation, et ainsi anticiper des événements routiers afin d'apporter des solutions préventives plutôt que réactives.

Ensuite, nous avons distingué l'approche NaaS de l'approche SaaS. Aussi, il semblerait judicieux de fusionner ces deux services et de faire appel à l'un ou à l'autre en incluant un peu plus d'informations dans nos requêtes. Dans la solution de type SaaS que nous avons proposée, il n'existe qu'un service : donner une liste de DENMs en échange d'un CAM. En se basant sur nos travaux du Chapitre 3, nous pourrions étendre le CAM pour faire des requêtes plus précises et proposer ainsi d'autres applications.

Selon nous, les données de petite taille générées en grand nombre et par de nombreux acteurs représentent le vecteur principal de surcharge d'un réseau, c'est pourquoi nous nous sommes concentrés dessus dans le Chapitre 3. Cependant, notre approche peut se décliner, avec quelques changements, en une version pour des données plus complexes, et donc plus conséquentes. De plus, la vision que nous avons introduite pour la ville intelligente sous forme d'un Cloud de contexte, mérite d'être développée et nous espérons transposer cette vision à n'importe quelle échelle : le pays est un contexte, la centre commerciale est un contexte et la voiture est un contexte. L'idée serait de fournir un framework permettant de faciliter la mise en place d'échanges dans un contexte grâce aux CAMs, pour n'importe quelles applications.

D'autre part, nous avons pu étudier, dans cette thèse, l'interaction entre les réseaux véhiculaires et le Cloud distant et entre les réseaux véhiculaires et Cloud de contexte. Nous allons nous atteler dans les travaux futurs à proposer une solution de type Cloud dynamique, formé entre plusieurs véhicules voisins, qui s'appuierait sur les échanges de type ICN que nous avons proposés dans le Chapitre 3.

Enfin, afin de faciliter la modélisation théorique, nous avons introduit certaines hypothèses qui nous ont éloignés un peu de la réalité. Il serait intéressant de les supprimer pour fournir un modèle le plus proche possible de la réalité.

Bibliographie

- [1] «ETSI,» [En ligne]. Available: <https://www.etsi.org/>.
- [2] R. Stanica, E. Chaput et A.-L. Beylot, «Enhancements of IEEE 802.11 p protocol for access control on a VANET control channel,» chez *Communications (ICC), 2011 IEEE International Conference on*, 2011.
- [3] P. Bedi et V. Jindal, «Use of Big Data technology in Vehicular Ad-hoc Networks.,» chez *ICACCI*, 2014.
- [4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher et B. Ohlman, «A survey of information-centric networking,» *IEEE Communications Magazine*, vol. 50, 2012.
- [5] A. Zahemszky, B. Gajic, C. E. Rothenberg, C. Reason, D. Trossen, D. Lagutin, J. Tuononen et K. Katsaros, «Experimentally-driven research in publish/subscribe information-centric inter-networking,» chez *International Conference on Testbeds and Research Infrastructures*, 2010.
- [6] C. Dannewitz, «Netinf: An information-centric design for the future internet,» chez *Proc. 3rd GI/ITG KuVS Workshop on The Future Internet*, 2009.
- [7] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker et I. Stoica, «A data-oriented (and beyond) network architecture,» chez *ACM SIGCOMM Computer Communication Review*, 2007.
- [8] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang et others, «Named data networking,» *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 66-73, 2014.
- [9] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa et L. Zhang, «VANET via named data networking,» chez *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, 2014.
- [10] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos et others, «A survey of information-centric networking research.,» *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 1024-1049, 2014.
- [11] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar et A. V. Vasilakos, «Information-centric networking for the internet of things: challenges and opportunities,» *IEEE Network*, vol. 30, pp. 92-100, 2016.

- [12] M. Amadeo, C. Campolo et A. Molinaro, «Information-centric networking for connected vehicles: a survey and future perspectives,» *IEEE Communications Magazine*, vol. 54, pp. 98-104, 2016.
- [13] X. Liu, Z. Li, P. Yang et Y. Dong, «Information-centric mobile ad hoc networks and content routing: a survey,» *Ad Hoc Networks*, vol. 58, pp. 255-268, 2017.
- [14] S. H. Bouk, S. H. Ahmed, D. Kim et H. Song, «Named-data-networking-based ITS for smart cities,» *IEEE Communications Magazine*, vol. 55, pp. 105-111, 2017.
- [15] F. M. Modesto et A. Boukerche, «An analysis of caching in information-centric vehicular networks,» chez *Communications (ICC), 2017 IEEE International Conference on*, 2017.
- [16] A. Boukerche, R. W. L. Coutinho et X. Yu, «LISIC: A Link Stability-Based Protocol for Vehicular Information-Centric Networks,» chez *Mobile Ad Hoc and Sensor Systems (MASS), 2017 IEEE 14th International Conference on*, 2017.
- [17] M. Tarroumi et I. Jabri, «EVNDN: Enhanced vehicular named data networking,» chez *Networks, Computers and Communications (ISNCC), 2017 International Symposium on*, 2017.
- [18] D. Grewe, M. Wagner et H. Frey, «A Domain-Specific Comparison of Information-Centric Networking Architectures for Connected Vehicles,» *IEEE Communications Surveys & Tutorials*, 2018.
- [19] B. B. P. Rao, P. Saluia, N. Sharma, A. Mittal et S. V. Sharma, «Cloud computing for Internet of Things & sensing based applications,» chez *Sensing Technology (ICST), 2012 Sixth International Conference on*, 2012.
- [20] R. Morello, S. C. Mukhopadhyay, Z. Liu, D. Slomovitz et S. R. Samantaray, «Advances on sensing technologies for smart cities and power grids: A Review,» *IEEE Sensors Journal*, vol. 17, pp. 7596-7610, 2017.
- [21] G. Suci, A. Vulpe, S. Halunga, O. Fratu, G. Todoran et V. Suci, «Smart cities built on resilient cloud computing and secure internet of things,» chez *Control Systems and Computer Science (CSCS), 2013 19th International Conference on*, 2013.
- [22] A. Zaslavsky, C. Perera et D. Georgakopoulos, «Sensing as a service and big data,» *arXiv preprint arXiv:1301.0159*, 2013.
- [23] R. Petrolo, V. Loscri et N. Mitton, «Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms,» *Transactions on Emerging Telecommunications Technologies*, vol. 28, p. e2931, 2017.
- [24] Z. Xiao, H. B. Lim et L. Ponnambalam, «Participatory sensing for smart cities: A case study on transport trip quality measurement,» *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 759-770, 2017.

- [25] T. Qiu, X. Liu, K. Li, Q. Hu, A. K. Sangaiah et N. Chen, «Community-Aware Data Propagation with Small World Feature for Internet of Vehicles,» *IEEE Communications Magazine*, vol. 56, pp. 86-91, 2018.
- [26] M. Gerla, «Vehicular cloud computing,» chez *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, 2012.
- [27] S. Bitam et A. Mellouk, «Its-cloud: Cloud computing for intelligent transportation system,» chez *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012.
- [28] M. Whaiduzzaman, M. Sookhak, A. Gani et R. Buyya, «A survey on vehicular cloud computing,» *Journal of Network and Computer Applications*, vol. 40, pp. 325-344, 2014.
- [29] R. Housley, W. Ford, W. Polk et D. Solo, «Internet X. 509 public key infrastructure certificate and CRL profile,» 1998.
- [30] K. Zheng, H. Meng, P. Chatzimisios, L. Lei et X. Shen, «An SMDP-based resource allocation in vehicular cloud computing systems,» *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 7920-7928, 2015.
- [31] M. U. Farooq, M. Pasha et K. U. R. Khan, «Contextual communication patterns for vehicular cloud computing: Towards realizing internet of things,» chez *Intelligent Systems and Control (ISCO), 2016 10th International Conference on*, 2016.
- [32] A. Festag, «Cooperative intelligent transport systems standards in Europe,» *IEEE communications magazine*, vol. 52, pp. 166-172, 2014.
- [33] E. T. S. I. Draft, «EN 302 636-3 v1. 1.2 (2014-03),» *Intelligent Transport Systems (ITS)*, 2014.
- [34] E. T. S. I. TS, «102 724 (V1. 1.1)(10-2012):" Intelligent Transport Systems (ITS),» *Harmonized Channel Specifications for Intelligent Transport Systems operating in the*, vol. 5, 2012.
- [35] E. T. S. I. TS, «102 687: Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range,» *Access layer part*, 2011.
- [36] E. T. S. I. TS, «103 175 (2015) Intelligent Transport Systems (ITS),» *Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium, v1*, vol. 1, 2015.
- [37] E. T. S. I. EN, «302 663 (V1. 2.1)(11-2012):" Intelligent Transport Systems (ITS),» *Access layer specification for Intelligent Transport Systems operating in the*, vol. 5, 2012.
- [38] E. T. S. I. EN, «302 636-4-1: Intelligent Transport Systems (ITS),» *Vehicular communications*, 2013.
- [39] E. T. S. I. EN, «302 637-2 V1. 3.1-Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,» *ETSI*, Sept, 2014.

- [40] E. T. S. I. EN, «302 637-3 V1. 2.2 (2014-11) Intelligent Transport Systems (ITS),» *Vehicular Communications*, 2014.
- [41] E. T. S. I. TS, «103 097 V1.3.1-Intelligent Transport Systems (ITS); Security; Security Header and Certificate Formats,» 2017.
- [42] E. T. S. I. TS, «101 539-1 V1. 1.1 (2013-08) Intelligent Transport Systems (ITS),» *V2X Applications*, 2013.
- [43] M. Jutila, J. Scholliers, M. Valta et K. Kujanpää, «ITS-G5 performance improvement and evaluation for vulnerable road user safety services,» *IET Intelligent Transport Systems*, vol. 11, pp. 126-133, 2017.
- [44] «Projet Scoop,» [En ligne]. Available: <http://www.scoop.developpement-durable.gouv.fr>.
- [45] «Projet InterCor,» [En ligne]. Available: <http://intercor-project.eu/>.
- [46] «Projet C-Roads,» [En ligne]. Available: <https://www.c-roads.eu/platform.html>.
- [47] «Documentation_Prototype,» [En ligne]. Available: http://scoop.univ-reims.fr/ITS_documentation/.
- [48] K. Mori, O. Shagdar, S. Matsuura, M. Tsukada, T. Ernst et K. Fujikawa, «Experimental study on channel congestion using ieee 802.11 p communication system,» chez *IPSJ Technical Workshop on Mobile Computing and Ubiquitous Communications*, 2013.
- [49] D. Eckhoff, N. Sofra et R. German, «A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE,» chez *Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on*, 2013.
- [50] U. Lamping, R. Sharpe et E. Warnicke, *Wireshark User's Guide for Wireshark 2.1*, 2014.
- [51] E. T. S. I. TS, «ETSI TS 102 940 V1.1.1-Intelligent Transport Systems (ITS);Security;ITS communications security architecture and security management,» 2012.
- [52] M. Wolf et T. Gendrullis, «Design, implementation, and evaluation of a vehicular hardware security module,» chez *International Conference on Information Security and Cryptology*, 2011.
- [53] «Eclipse,» [En ligne]. Available: <https://www.eclipse.org/org/>.
- [54] L. Gu, D. Zeng et S. Guo, «Vehicular cloud computing: A survey,» chez *Globecom Workshops (GC Wkshps), 2013 IEEE*, 2013.
- [55] D. K. Yadav, P. Kumar et A. Kaur, «Implementation of Data Center in Vehicular Cloud Computing,» *International Journal of Innovations & Advancement in Computer Science*, 2017.

- [56] A. Botta, W. De Donato, V. Persico et A. Pescapé, «On the integration of cloud computing and internet of things,» chez *Future internet of things and cloud (FiCloud), 2014 international conference on*, 2014.
- [57] W. He, G. Yan et L. Da Xu, «Developing vehicular data cloud services in the IoT environment,» *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1587-1595, 2014.
- [58] P. Ballon, J. Glidden, P. Kranas, A. Menychtas, S. Ruston et S. Van Der Graaf, «Is there a need for a cloud platform for european smart cities?,» chez *eChallenges e-2011 Conference Proceedings, IIMC International Information Management Corporation*, 2011.
- [59] M. A. Salahuddin, A. Al-Fuqaha et M. Guizani, «Software-defined networking for rsu clouds in support of the internet of vehicles,» *IEEE Internet of Things journal*, vol. 2, pp. 133-144, 2015.
- [60] R. Yu, Y. Zhang, S. Gjessing, W. Xia et K. Yang, «Toward cloud-based vehicular networks with efficient resource management,» *IEEE Network*, vol. 27, pp. 48-55, 2013.
- [61] E. T. S. I. TS, «101 539-2:" Intelligent Transport Systems (ITS),» *V2V application*, 2018.
- [62] M. N. O. Sadiku, A. E. Shadare et S. M. Musa, «Named data networking,» *International Journal of Engineering Research*, vol. 6, pp. 371-372, 2017.
- [63] V. Dimitrov et V. Koptchev, «PSIRP project--publish-subscribe internet routing paradigm: new ideas for future internet,» chez *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, 2010.
- [64] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos et others, «Named data networking (ndn) project,» *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, vol. 157, p. 158, 2010.
- [65] P. Crowley, C. P. L. W. L. Zhang, A. Afanasyev et B. Zhang, «Named data networking,» chez *China-America frontiers of engineering symposium, frontiers of engineering*, 2013.
- [66] H. Reddy, «PATH FINDING-Dijkstra's and A* Algorithm's,» *International Journal in IT and Engineering*, pp. 1-15, 2013.
- [67] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin et F. R. Yu, «A survey of mobile information-centric networking: Research issues and challenges,» *IEEE Communications Surveys & Tutorials*, 2018.
- [68] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba et B. Mathieu, «A survey of naming and routing in information-centric networks,» *IEEE Communications Magazine*, vol. 50, 2012.
- [69] G. Zhang, Y. Li et T. Lin, «Caching in information centric networking: A survey,» *Computer Networks*, vol. 57, pp. 3128-3141, 2013.

- [70] X. Zhang, G.-Q. Wang et R. Ravindran, *Principal-identity-domain based naming scheme for information centric networks*, Google Patents, 2016.
- [71] J. Xiaoke, B. Jun, N. Guoshun et L. Zhaogeng, «A survey on information-centric networking: rationales, designs and debates,» *China Communications*, vol. 12, pp. 1-12, 2015.
- [72] «FP7 PURSUIT Project,» 2010. [En ligne]. Available: <http://www.fp7->.
- [73] «FP7 SAIL Project,» 2010. [En ligne]. Available: <http://www.sailproject.eu/>.
- [74] «FP7 COMET Project,» 2010. [En ligne]. Available: <http://www.cometproject.org/>.
- [75] «FP7 Convergence Project,» 2010. [En ligne]. Available: <http://www.ictconvergence.eu/>.
- [76] «ANR Connect Project,» 2011. [En ligne]. Available: <http://anrconnect.org/>.
- [77] «GreenICN Project,» 2013. [En ligne]. Available: <http://www.greenicn.org/>.
- [78] «ICE-AR Project,» 2017. [En ligne]. Available: <http://ice-ar.nameddata.net>.
- [79] «ICN-2020 Project,» 2016. [En ligne]. Available: <http://www.icn2020.org>.
- [80] «NSF Named Data Networking Project,» 2010. [En ligne]. Available: <http://www.named-data.net/>.
- [81] «NSF Mobility First Project,» 2010. [En ligne]. Available: <http://mobilityfirst.winlab.rutgers.edu/>.
- [82] D. R. Cheriton et M. Gritter, *TRIAD: A scalable deployable NAT-based internet architecture*, June 09 2000. www.wdsd.stanford.edu/triad/triad.ps.gz.
- [83] D. R. Cheriton et M. Gritter, «TRIAD: A new next-generation Internet architecture,» 2000.
- [84] T. Ye, Z. Su, J. Wu, L. Guo, J. Li et G. Li, «A Safety Resource Allocation Mechanism against Connection Fault for Vehicular Cloud Computing,» *Mobile Information Systems*, vol. 2016, 2016.
- [85] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou et H. Cai, «VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing,» *Mobile Networks and Applications*, vol. 19, pp. 153-160, 2014.
- [86] A. Botta, W. De Donato, V. Persico et A. Pescapé, «Integration of cloud computing and internet of things: a survey,» *Future Generation Computer Systems*, vol. 56, pp. 684-700, 2016.
- [87] S. Bitam, A. Mellouk et S. Zeadally, «VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks,» *IEEE Wireless Communications*, vol. 22, pp. 96-102, 2015.
- [88] M. Aloqaily, B. Kantarci et H. T. Mouftah, «Vehicular clouds: State of the art, challenges and future directions,» chez *Applied Electrical Engineering and Computing Technologies (AEECT)*, 2015 IEEE Jordan Conference on, 2015.

- [89] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis et A. Vakali, «Cloud computing: Distributed internet computing for IT and scientific research,» *IEEE Internet computing*, vol. 13, 2009.
- [90] S. K. Dash, S. Mohapatra et P. K. Pattnaik, «A survey on applications of wireless sensor network using cloud computing,» *International Journal of Computer Science & Emerging Technologies*, vol. 1, pp. 50-55, 2010.
- [91] E. Lee, E.-K. Lee, M. Gerla et S. Y. Oh, «Vehicular cloud networking: architecture and design principles,» *IEEE Communications Magazine*, vol. 52, pp. 148-155, 2014.
- [92] M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas et R. Rivest, «Proxy-based security protocols in networked mobile devices,» chez *Proceedings of the 2002 ACM symposium on Applied computing*, 2002.

Etude de l'interaction entre réseaux véhiculaires et Clouds

Cette thèse fait un état des lieux des solutions Cloud qui visent à améliorer certains aspects des réseaux véhiculaires comme la latence et l'utilisation de la bande passante. L'état de l'art présente les principaux sujets qui nous concernent, à savoir, le Vehicular Cloud Computing (VCC), l'Information Centric Networking (ICN), les villes intelligentes ainsi que les standards ETSI (European Telecommunications Standards Institute) pour les ITS (Intelligent Transport System).

Les véhicules intelligents ouvrent de nouveaux horizons pour les applications de sécurité routière ou l'optimisation de trafic. Le canal de communication G5 (Wifi 802.11p), utilisé par les ITS qui sont actuellement déployées en Europe et aux États-Unis notamment, ne permet pas toujours d'assurer la connectivité continue d'un véhicule au réseau. A ceci s'ajoute une bande passante limitée. Ces deux phénomènes ont pour conséquence d'allonger les délais de communication d'une station ITS à une autre.

A travers cette thèse, nous apportons des solutions qui bénéficient du large déploiement du réseau cellulaire. Ces solutions, de type SaaS ou NaaS, collectent tout ou une partie des messages générés par les véhicules, pour les redistribuer, ou en déduire des événements routiers. Elles ont été expérimentées sur route à l'aide d'applications embarquées. Ceci a permis de mesurer les délais moyens de communication sur le réseau cellulaire, et le temps nécessaire à la distribution d'un événement. Afin de comparer ces résultats avec les VANET, nous présentons un simulateur qui reproduit le comportement des véhicules sur route et les communications G5 en implémentant la pile ETSI des ITS.

Nous avons abordé la ville intelligente que nous modélisons sous la forme d'un Cloud de contexte, composé d'un serveur, de multiples capteurs et des véhicules. Et nous présentons notre architecture ICN qui consiste, entre autres, à intégrer les échanges dans la couche « facilities » de l'ETSI après avoir condensé un ensemble de données en une seule au niveau d'un serveur central, et a pour but de limiter l'impact des échanges, des données de capteurs, sur la bande passante du G5. De plus grâce aux échanges de véhicule à véhicule, nous cherchons à étendre la zone du réseau dans laquelle l'information est distribuée. La présentation de notre approche passe par une modélisation analytique prenant en compte la mobilité et le placement aléatoire des UBR dans une ville de type Manhattan. Cette modélisation permet de mesurer la charge réseau engendrée par les échanges et du pourcentage de véhicules qui reçoit l'information. Nous comparons notre approche avec deux autres : *Named Data Network (NDN)* et *Rendez-Vous Network (RENE)*, après avoir donné les modèles analytiques de celles-ci. Pour valider ces modèles, nous présentons un simulateur qui reproduit la ville, la mobilité des véhicules et les communications liées à chaque approche. Enfin, nous comparons les performances de ces approches dans la recherche du plus court chemin dans une ville intelligente.

Mots-clés : VANETs ; systèmes de transport intelligents ; réseau orienté donnée ; Cloud ; simulation ; modèle analytique ; ITS-G5 ; SaaS ; NaaS ; simulateur ITS ; performances ITS

Study about interactions between Clouds and Vehicular Networks

This thesis deals with Cloud solutions which aim to enhance some aspects of the vehicular networks such as the latency and the use of the (bandwidth). The state of the art presents the main subjects that concern us, namely the vehicular Cloud Computing (VCC), the Information Centric Networking (ICN), the smart cities and at last, the ETSI (European Telecommunications Standards Institute) standards for the ITS (Intelligent Transport System).

The intelligent vehicles increase the scope of the road safety apps and the traffic optimisation. The communication channel G5 (Wifi 802.11p), used by the ITS currently deployed in Europe and in the USA, doesn't always enable to secure the non-stop connectedness of a vehicle to the network. Moreover, a limited bandwidth creates a growth of crashes. The consequence of these 2 phenomena is to extend the communication period from a ITS station to another.

This thesis offers solutions which work thanks to the wide deployment of the cellular network. These solutions have been tested on the road, with the help of the Android app. Through this procedure, we can now demonstrate all the benefits offered by these solutions to improve connectedness as well as the time of the receipt of what can happen on the road. In addition, we support the problems of connectedness of the G5 using a simulator that we have designed; this device implements, in a very precise way, the ETSI stack of the ITS. Thus, we can now analyse the strengths and the weaknesses of the different approaches as well as those from the scientific publications. For instance, within the framework of a smart city that we have modelled into a context Cloud, composed of a server, many sensors and vehicles, we partly solved the problem of the bandwidth linked to the sending of the data collected by the sensors towards the vehicles.

One of our contributions is an ICN architecture which uses the ETSI facilities among other things. It both enhances the quality of the service (which is providing the data to the largest number of vehicles) and the network overhead applied on the G5 channel, compared with the existing approaches. We have created an analytic model for each of these ones, taking into account the mobility of the vehicles into a city and the random positions of the UBR. In order to verify and approve the results, we have developed the simulator to take into account each aspects of the model and the communication of the different approaches.

Mots-clés : VANETs ; Intelligent Transport Systems ; Information Centric Networking ; Cloud ; Simulation ; Analytic model ; ITS-G5 ; SaaS ; NaaS ; ITS simulator ; ITS performances.

Discipline : INFORMATIQUE

Spécialité : Informatique

Université de Reims Champagne-Ardenne
Crestic
UFR Sciences Exactes et Naturelles,
Moulin de la Housse, 51867 Reims