

THÈSE

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Discipline : INFORMATIQUE

Spécialité : Informatique

Présentée et soutenue publiquement par

KEVIN THOMAS

Le 1er septembre 2020

Communications Bluetooth pour les Systèmes de Transport Intelligents Coopératifs

Thèse dirigée par **HACENE FOUCHAL** et **FRANCIS ROUSSEAU**

JURY

M. Hacène Fouchal	Professeur	Université de Reims Champagne-Ardenne	Directeur de thèse
M. Francis Rousseaux	Professeur	Université de Reims Champagne-Ardenne	Co-Directeur de thèse
M. Stéphane Cormier	Maître de Conférences	Université de Reims Champagne-Ardenne	Co-Encadrant de thèse
M. Jalel Ben-Othman	Professeur	Université Paris 13	Rapporteur
M. Rabie Ben Atitallah	Maître de Conférences HDR	Université de Galatasaray	Rapporteur
M. Benoît Hilt	Maître de Conférences HDR	Université de Mulhouse	Examineur
Mme Selma Boumerdassi	Maitre de Conférences HDR	CNAM Paris	Examineur
Mme Leila Merghem-Boulahia	Professeur	Université de Technologie de Troyes	Examineur
M. Eric Olliger	Ingénieur	Ministère de la transition écologique et solidaire	Invité



Cette thèse est financée par le projet

C-Roads



C-Roads

Remerciements

Je ne saurais commencer ces lignes, sans auparavant remercier très sincèrement Hacène Fouchal, un professeur qui, après m'avoir vu sur les bancs de l'université, m'a permis de débiter cette thèse. Il m'a intégré à son équipe et a su m'accompagner, tout au long de ces travaux, en tant que directeur de thèse. Sans lui et son implication, je n'en serais probablement pas arrivé là.

Je souhaiterais, également, exprimer toute ma reconnaissance à Francis Rousseau, un professeur qui a aussi été témoin de mon parcours en études supérieures et qui a accepté de me soutenir en tant que co-directeur de thèse. Il m'a toujours poussé vers l'avant et fourni des recommandations pour mes rédactions.

Je voudrais également remercier mon co-encadrant Stéphane Cormier qui a toujours pris le temps de me conseiller et de m'encourager dans la réalisation de mes travaux.

Je souhaiterais également exprimer ma reconnaissance à Benoît Hilt et Rabie Ben Atitallah pour avoir accepté d'être mes rapporteurs de thèse.

Je remercie Marwane Ayaida qui m'a beaucoup aidé dans la compréhension de certaines notions et m'a fait de nombreuses suggestions.

Je souhaiterais également remercier les équipes du CReSTIC. Notamment Bernard Riera, directeur du CReSTIC, avec qui j'ai pu partager ma passion de l'impression 3D. Mais aussi Ida et Marielle, sans qui nous serions administrativement totalement perdus. Je ne saurais omettre mon équipe, Emilien et Geoffrey qui m'ont épaulé et qui ont su rester fidèles à eux-même, l'un pour ses goûts musicaux métalliques et l'autre pour sa passion du chanteur Renaud. Je remercie également Brice qui me suit depuis le BTS, c'est un plaisir de travailler avec lui. Je remercie enfin mon collègue de sandwich du midi, Dimitri, avec qui nous avons participé à l'enrichissement de notre boulangerie favorite.

Je remercie mes amis Charles, Yann, Jessica pour leur implication dans la relecture de ces travaux, sachez que votre soutien a grandement contribué à l'élaboration de ce document.

Enfin, je ne pourrais jamais assez remercier ma famille, Olivier, mon père, Agnès, ma mère et Marine, ma sœur pour m'avoir soutenu dans tous les moments de ma vies. Je ne pourrais jamais leur dire à quel point ils sont chers à mon cœur.

Résumé

Cette thèse propose une solution permettant de transmettre des informations routières par le biais de réseaux personnels sans fil (*Wireless Personal Area Network* - WPAN).

À travers cette thèse, nous proposons une architecture de communication s'inscrivant dans les Systèmes de transports intelligents coopératifs (*Cooperative Intelligent Transport Systems* - C-ITS). Cette architecture utilisera le Bluetooth Low Energy (BLE) pour transmettre des informations routières. De plus, nous avons adapté le message *Decentralized Event Notification Message* (DENM), connu du monde des C-ITS pour être envoyé dans un réseau BLE. Cette adaptation veille à comporter suffisamment d'informations vitales pour assurer la caractérisation d'un évènement. Cette architecture est validée par des tests sur route.

Nous présentons, ensuite, des solutions pour identifier la voie sur laquelle se déroule l'évènement. Elles doivent permettre, aux véhicules, de savoir si l'évènement concerne la voie sur laquelle ils circulent. Ces solutions font l'objet d'expérimentations. Nous avons sélectionné celles qui donnent les meilleures performances.

Ensuite, nous présentons une architecture étendue proposant l'extension de la couverture de notre réseau, en installant des stations intermédiaires. Ces stations permettent de considérer différentes fonctionnalités à ajouter à l'architecture.

Abstract

This thesis proposes a solution to transmit traffic information via Wireless Personal Area Networks (WPAN).

Through this thesis, we propose a communication architecture that fits within the Cooperative Intelligent Transport Systems (C-ITS). This architecture will use Bluetooth Low Energy (BLE) to transmit traffic information. In addition, we have adapted the Decentralized Event Notification Message (DENM) message known from the C-ITS world to be sent in a BLE network. This adaptation provides sufficient vital information to ensure the characterization of an event. This architecture is validated by road tests.

Then, we will present solutions to identify the path on which the event takes place. They must allow vehicles to know if the event is on the lane on which they are travelling. These solutions are the subject of experiments. We have selected those that give the best performance.

We shall present an extended architecture proposing the extension of the coverage of our network by installing intermediate stations. These stations make it possible to consider different functionalities to be added to the architecture.

Table des matières

1	Introduction	1
1.1	Contexte	1
1.2	Problématique	1
1.3	Contributions	2
2	État de l'art	5
2.1	Réseau ad-Hoc de véhicules (VANET)	5
2.1.1	Routage géographique	5
2.2	Vers les C-ITS	6
2.2.1	Les C-ITS	6
2.2.2	Motivations de mise en place des réseaux véhiculaires	7
2.2.3	Faisabilité d'une telle solution	8
2.2.4	Cas d'usages des C-ITS	9
2.3	Pile protocolaire des C-ITS	9
2.3.1	DSRC/Wave - Norme US	9
2.3.2	ETSI ITS-G5 - Norme EU	12
2.3.3	<i>GeoNETworking</i>	13
2.3.4	La couche <i>Security</i>	14
2.3.5	Messages de la couche <i>facilities</i>	14
2.3.6	Autre réseau des C-ITS	15
2.4	Les C-ITS du point de vue des Gestionnaires routiers	18
2.5	Les réseaux personnels sans fil	18
2.5.1	ZigBee	19
2.5.2	Z-Wave	20
2.5.3	Bluetooth	21

2.5.4	Cas d'illustration - DP3T	33
3	Architecture Proposée	35
3.1	Architecture	35
3.1.1	Motivations du choix du BLE	36
3.1.2	Fonctionnement de l'architecture	36
3.2	Scénario d'illustration	37
3.3	Flux de communication	38
3.4	Comportement des appareils de l'architecture	38
3.4.1	Sur le Backoffice ITS	38
3.4.2	Sur un BLETransmitter	40
3.4.3	Sur l'OBU	41
3.5	BLE et <i>Advertisement</i>	42
3.5.1	Diffuser des informations par balises dans les C-ITS	42
3.5.2	Le <i>beaconing</i> en BLE	42
3.6	Messages	43
3.6.1	Contraintes de taille des messages	43
3.6.2	UUID	43
3.6.3	Datagramme	44
3.7	Tests sur route	45
3.7.1	Expérimentations sur la route	45
3.7.2	Analyse des expérimentations	49
3.8	Discussion	50
4	Dissémination dans un cadre BLE	51
4.1	Introduction	51
4.1.1	Situation actuelle	51
4.1.2	Problématique de ciblage et de limite de taille	52

4.1.3	Identification des critères de recherche	53
4.1.4	Liste des solutions étudiées	53
4.2	Mesure du RSSI	54
4.2.1	Le <i>Received Signal Strength Indication</i> - RSSI	54
4.2.2	Présentation	54
4.2.3	Illustration	55
4.2.4	Expérimentations	57
4.2.5	Évaluation des résultats	63
4.2.6	Analyse	63
4.3	Relevé de séquence de messages	63
4.3.1	Présentation	64
4.3.2	Illustration	64
4.3.3	Expérimentations	66
4.3.4	Analyse	66
4.4	Calcul de la direction	67
4.4.1	La direction	67
4.4.2	Présentation	68
4.4.3	Illustration	70
4.4.4	Comparaison des directions	71
4.4.5	Expérimentations	72
4.4.6	Analyse	75
4.5	Discussion	77
5	Architecture Étendue	79
5.1	Introduction	79
5.1.1	Problématique de couverture	80
5.1.2	Hypothèse formulée	80
5.2	Répartition au sein de l'architecture proposée	80

5.3	Déclaration d'évènements	81
5.3.1	Flux de communication	82
5.3.2	Processus de déclaration	82
5.3.3	Comportement des appareils dans l'architecture générale .	83
5.3.4	Messages diffusés	85
5.3.5	Évaluation de la fonctionnalité	86
5.4	Remontée de données véhiculaires	86
5.4.1	La zone de surveillance accrue	86
5.4.2	Comportement de l'OBU	87
5.4.3	Messages utilisés	88
5.5	Distribution d'informations semi-ciblées	91
5.5.1	Fonctionnement	91
5.5.2	Message de position - Le Light CAM	94
5.5.3	Évaluation de la fonctionnalité	95
5.6	Discussion	96
6	Conclusion et Perspectives	97
6.1	Conclusion	97
6.2	Perspectives	99
A	Correspondance CC et SCC en BLE	107

Table des figures

2.1	Architecture du protocole WAVE : IEEE 802.11p et IEEE 1609 [29]	10
2.2	Tableau des messages DSRC du standard SAE J2735 [30]	11
2.3	Architecture de référence pour une station ITS [32]	12
2.4	Architecture de la station centrale [39]	16
2.5	Routine présente sur l'ITSS-C [39]	17
2.6	Les différentes topologies réseaux. [44]	19
2.7	Pile protocolaire Bluetooth Low Energy [51]	22
2.8	Canaux Bluetooth Low Energy[52]	24
2.9	Intervalles de diffusion et de récupération des paquets d' <i>advertising</i> [54]	25
2.10	Temps nécessaire à l'établissement d'une connexion BLE en fonction de l' <i>Advertising interval</i> sans interférences [56]	26
2.11	Temps nécessaire à l'établissement d'une connexion BLE en fonction de l' <i>Advertising interval</i> avec interférences [56]	27
2.12	Structure d'un paquet d' <i>Advertising</i> en BLE [55]	29
2.13	Fonctionnement de l'Eddystone-UID	31
2.14	Fonctionnement de l'Eddystone-URL	31
2.15	Fonctionnement de l'iBeacon	32
2.16	structure d'un AltBeacon [63]	33
3.1	Architecture globale de Communication en BLE	36
3.2	Affichage sur l'équipement C-ITS	39
3.3	Flux de communication	40
3.4	Diagramme d'activité du Gestionnaire	40
3.5	Diagramme d'activité du BLETransmitter	41
3.6	Diagramme d'activité du système embarqué	41

3.7	RSUs autours de Reims.	48
4.1	BLETransmitter sur la route	52
4.2	Cas d'un signal en diminution à l'approche d'un BLETransmitter	56
4.3	Cas d'un signal en augmentation à l'approche d'un BLETransmitter	56
4.4	Architecture de test	57
4.5	Mesures de RSSI au passage de la RSU Saint-Remi	60
4.6	Mesures de RSSI au passage de la RSU Bezannes	61
4.7	Relevé des RSSI du passage du 10/10/19	62
4.8	Réception de deux messages sur la voie A	64
4.9	Réception de deux messages sur la voie B	65
4.10	Définition de la direction d'un véhicule	67
4.11	Illustration des différentes directions.	69
4.12	Correspondance entre la direction du véhicule et la direction de la voie concernée.	70
4.13	Décalage entre la direction du véhicule et la direction de la voie concernée.	71
4.14	Validation de l'évènement par comparaison des directions	72
4.15	Comparaison des directions relevées du véhicule et de la direction paramétrée sur la RSU Saint-Remi	75
4.16	Comparaison des directions relevées du véhicule et de la direction sur la RSU Bezannes	76
5.1	Localisation des Raspberrys	81
5.2	Flux de communication de l'architecture générale	82
5.3	Diagramme d'activité du Gestionnaire dans l'architecture générale	83
5.4	Diagramme d'activité du BLETransmitter dans l'architecture générale	84
5.5	Diagramme d'activité du système embarqué dans l'architecture générale	84
5.6	Zone de surveillance accrue	87

5.7	Diagramme d'activité du système embarqué dans l'architecture générale	87
5.8	Cas de détection de queue de bouchon par observation des vitesses des Véhicules	89
5.9	Cas de détection d'obstacle sur la route par observation des directions des Véhicules	90
5.10	Communication ciblée	92
5.11	Réception d'information sur le Serveur	93
5.12	Routine d'envoi.	94

Liste des tableaux

3.1	Résultats tests Phase 1	46
3.2	Concaténation des résultats de réception sur tablette lors de la Phase 2	47
3.3	Récapitulatif des tests effectués	48
4.1	Relevés des RSSIs en dBm lors des tests avec le Raspberry	58
4.2	Séquences enregistrées durant le test	66
4.3	Résultats de l'évaluation des directions avec le Raspberry	73
4.4	Résultats de l'évaluation des directions avec une RSU	74

Introduction

1.1 Contexte

L'automobile est devenue plus qu'un simple moyen de locomotion dans la vie de tous les jours. C'est un véritable système informatique, bardé de capteurs et connecté avec son utilisateur. Aujourd'hui, ces véhicules deviennent des mines d'informations qui ne demandent qu'à être exploitées.

Les technologies réseaux deviennent de plus en plus performantes et permettent la communication de ces données dans des environnements mobiles. De nombreux projets [1] [2] s'inscrivent dans cette optique en élaborant des moyens de transmissions, et d'échanges de ces données. Ils constituent alors des avancées significatives dans le domaine de l'optimisation de flux routiers, de la sécurité routière et de l'écologie. De plus, la mise en place de ce type de communications alimente et prépare la création et le déploiement du véhicule de demain : le véhicule autonome.

1.2 Problématique

Assurer la transmission de données routières est une tâche nécessitant des couvertures réseaux importantes. En effet, ces technologies ont pour vocation d'intervenir sur l'ensemble des axes routiers, il n'est pas chose aisée de fournir une couverture optimale sur une telle surface. De plus, élaborer un réseau de communication sur la route soulève des problématiques de distribution de l'information dues à la forte mobilité des véhicules.

Afin de satisfaire ces besoins, certains projets [1] [2] mettent en place des solutions de routages et d'hybridations réseaux permettant, ainsi, de fournir une qualité de services élevée qui assure une distribution, en tous lieux, des informations routières nécessaires au bon déroulement du trajet de l'utilisateur.

Mais, aujourd'hui, il reste encore de nombreuses zones non-couvertes par de tels réseaux. Des véritables "Zones Blanches" dans lesquelles le véhicule n'a aucun moyen de recevoir ou transmettre des données et donc de fournir les fonctionnalités pour lesquelles il a été créé. Des solutions simples et efficaces doivent donc être trouvées pour assurer une communication des informations routières et ainsi, combler ces zones.

1.3 Contributions

Dans cette thèse, nous proposons un état de l'art traitant des réseaux véhiculaires. Nous présentons les motivations et cas d'usages relatifs à ce type de réseaux. Nous détaillons, également, les piles protocolaires en usage aux Etats-Unis et en Europe. Une fois ces points clarifiés, nous étudions différents réseaux personnels sans fil disponibles qui peuvent répondre à nos besoins. Ensuite, trois contributions sont présentées dans le but de fournir une solution à la problématique énoncée.

La première contribution consiste à mettre en place une architecture de communication s'inscrivant dans le modèle de mobilité que sont les réseaux véhiculaires. Cette architecture utilise un réseau personnel sans fil afin de pallier les "Zones Blanches" précédemment énoncées et ainsi proposer une alternative viable aux technologies actuellement déployées. Ce réseau va assurer la transmission des informations caractérisant des événements routiers. Pour transmettre ces données, nous positionnons des infrastructures que nous qualifions. Ainsi, nous mettons en place des balises BLE sur la route, et embarquons, dans les véhicules des équipements BLE de réception ou utilisons les smartphones des usagers.

De plus, nous redéfinissons la structure des informations routières transmises pour proposer notre propre format, qui se veut être plus léger, sans pour autant omettre de données vitales pour l'identification de l'évènement.

Cette solution est éprouvée par une série de tests visant à garantir sa faisabilité. Ces tests se composeront de différentes phases. La première constituera un test simple pour illustrer que la technologie employée est viable dans un environnement mobile. La seconde phase portera sur l'élaboration et la mise en place de l'architecture avec des équipements accessibles et simples. Enfin, la troisième phase se rapprochera le plus fidèlement possible d'une situation de déploiement de l'architecture afin d'évaluer les performances de celle-ci.

La seconde contribution assure un moyen de disséminer une information sur une voie précise de la route. De plus, nous présentons un moyen de comparer la voie sur laquelle le véhicule circule avec celle désignée par l'information routière dans le but de filtrer les données échangées sur l'architecture, ceci dans le but de distribuer les données aux usagers concernés. Cette solution doit être établie sur un minimum d'espace sur le paquet, dans le but de rester conforme à la légèreté de la structure définie dans le Chapitre 3.

Nous choisissons d'étudier trois pistes pour établir cette solution. Nous proposons d'utiliser les variations de la puissance du signal d'une balise relevées sur un appareil récepteur pour orienter le message et déduire si le véhicule est sur la voie concernée par cet évènement. Nous proposons également d'utiliser des séquences d'identifiants dans les messages pour associer une voie à un évènement, l'appareil recevant les messages sera alors en mesure de déterminer l'ordre de réception et en conclure s'il est concerné par ledit évènement. Enfin, nous énonçons une troisième solution qui compare l'orientation de la voie où se situe un évènement avec l'orientation du véhicule. Cette comparaison per-

met de déduire si le véhicule circule sur cette voie et donc s'il est concerné par l'évènement en cours.

La troisième contribution présente une extension de la proposition énoncée dans le Chapitre 3. Cette extension consiste à positionner des équipements légers sur la route, afin d'augmenter la couverture du système que nous présentons. Ce gain de couverture sert non seulement à la diffusion d'informations sur de plus grandes surfaces, mais, également, à envisager la remontée d'informations depuis les véhicules.

Nous sommes donc en mesure de proposer différentes fonctionnalités. Tout d'abord, nous proposons la mise en place d'un système déclaratif d'évènements, sur les véhicules, dans le but d'exploiter une nouvelle source d'informations, les usagers de la route. Ensuite, nous proposons de remonter des données de conduite pour identifier, passivement, des anomalies routières. De même, nous élaborons une solution utilisant la remontée d'informations pour limiter les transmissions dans un tel environnement. Nous fixons une structure légère d'un message mettant à jour la position du véhicule et donnant l'état de sa base de données d'évènements. Un serveur, collectant ces messages, pourra identifier les évènements non connus par les véhicules et sera en mesure de stopper toutes diffusions en cours si tous les véhicules connaissent les évènements dans leurs zones, dans le but de ne pas surcharger le réseau.

Nous terminons ce document par une conclusion et des perspectives de recherches qui découlent de ces travaux.

État de l'art

C'est avec les objectifs définis dans la section précédente que nous allons étudier le champs des possibilités actuelles dans le paysage scientifique. Nous déterminerons ce que désigne les réseaux Ad-Hoc de véhiculaires (*Vehicular Adhoc NETWORKs* - VANETs), ainsi que les *Cooperative-Intelligent Transports Systems* (C-ITS), nous détaillerons la pile protocolaire de ces C-ITS. Ceci afin de comprendre dans quel paysage s'inscrivent nos travaux, et quels éléments seraient amenés à interagir avec notre proposition.

Une fois ces points définis, nous étudierons différents réseaux personnels sans fil disponibles.

2.1 Réseau ad-Hoc de véhicules (VANET)

Un VANET est un type de réseau mobile Ad-Hoc (Mobile Ad-Hoc Network - MANET) (un réseau hors infrastructures fixes avec des noeuds mobiles [3]). Les VANETs définissent les véhicules communicants comme les noeuds du réseau.

Cette mobilité au sein de ce réseau engendre beaucoup de problèmes. En effet, l'ajout et la suppression très rapide de noeuds au sein du réseau entraînent non seulement des problèmes de routage lors de la transmission des messages, mais également des problèmes de couverture [4].

2.1.1 Routage géographique

Le routage de paquet, au sein d'un réseau évoluant massivement, est une tâche difficile. En effet, comme l'illustre l'article [5], à cause de la vitesse des véhicules qui composent les noeuds de ce type de réseau, les changements observés entraînent des difficultés de transmission de messages, le temps que le réseau se mette en place.

L'étude [6] présente, également, une étude dans laquelle il énonce les contraintes rencontrées par les MANETs et VANETs pour disséminer leurs informations efficacement et de manière sécurisée.

Comme le montre l'article [7], les délais de transmission des messages observés

dans un réseau véhicule à véhicule sont assez long tant que ce dernier n'est pas mis en place et stable.

De même, l'article [8] explique que les protocoles de routage présents dans les MANETs ne suffisent pas à garantir des performances efficaces lors de leurs applications dans des VANETs. L'auteur illustre son propos en mettant en place différents scénarios qu'il exécute dans un simulateur afin de comparer ces différents protocoles en prenant en compte diverses paramètres (ratio de paquets reçus, nombre de noeuds, etc...). L'article [9] présente aussi des pistes d'améliorations des algorithmes de routage et de diffusion de messages dans les VANETs et étaye ces propos avec des simulations.

De surcroît, des travaux innovent en présentant des solutions utilisant les caractéristiques de la route. C'est le cas de l'article [10] qui présente un nouvel algorithme de routage utilisant la densité de différentes sections de voies de circulation pour établir une route la plus dense possible, que le message empruntera.

2.2 Vers les Systèmes de transports intelligents coopératifs (C-ITS)

2.2.1 Les C-ITS

Les véhicules connectés présents dans les Systèmes de transports intelligents coopératifs (Cooperative Intelligent Transport Systems - C-ITS) sont capables de communiquer avec différents acteurs. Différents types de transmissions peuvent s'effectuer [3] :

- Véhicule à Véhicule (Vehicle to Vehicle - V2V)
- Véhicule à Infrastructure (Vehicle to Infrastructure - V2I)
- Infrastructure à Véhicule (Infrastructure to Vehicle - I2V)

L'ensemble de ces transmissions permettent de considérer que nous avons des Communications *Vehicle to Everything* (V2X).

De plus, au sein de ces C-ITS, nous avons différents acteurs :

- Les stations ITS des véhicules désignées comme des Unités Embarquées Véhicule (ou *On Board Unit* - OBU).
- Les stations ITS fixes déployées sur les routes sont désignées comme des Unités Bords de Route (ou *Road Side Unit* - RSU).

Cet ensemble compose le C-ITS, comme l'illustre l'article [11] qui présente un réseau de ce type en utilisant une approche utilisant le *Software-defined networking* (SDN).

2.2.2 Motivations de mise en place des réseaux véhiculaires

Nous allons présenter pourquoi les C-ITS sont devenus un enjeu majeur de ces dernières années.

En 2018, lors de son bilan annuel, l'Observatoire National Interministériel de la Sécurité Routière (ONISR) recense que, pour 10% des Auteurs Présumés d'Accidents Mortels (APAM), un défaut d'attention était en cause (éléments extérieurs, interactions, radio, téléphone, ...) [12]. Il peut donc paraître risqué d'ajouter des interactions avec l'utilisateur ; celles-ci augmenteraient ces statistiques, et pourtant, l'apport de ces technologies est bien réel.

Il y a plus de 20 ans, des travaux de véhicules communicants étaient déjà en cours dans le but d'améliorer le confort des usagers de la route. [13] présente un système d'automatisation dans le but de décongestionner les routes et d'améliorer la sécurité routière.

C'est en effet les buts principaux des C-ITS, de proposer des solutions pour augmenter la sécurité routière et proposer des moyens de gérer plus facilement les flux de trafic, par la transmission des informations en temps réel en ayant l'exigence de réduire, au maximum, les délais de transmissions. L'étude [3] présente ces aspects, elle illustre les enjeux des VANETs avec les applications possibles de tels réseaux, comme la communication en temps réel d'informations routières pour éviter les collisions ou optimiser les flux de trafic, pour la circulation de véhicules d'urgences par exemple. De même que, l'article [14] qui présente un système permettant la diffusion de messages spécifiques pour signaler la circulation d'un véhicule d'urgence, à laisser passer.

De surcroît, une autre étude, [6], présente, entre autre, les différentes applications des VANETs en matière de sécurité routière.[15] présente également une étude traitant des applications possibles d'un système de communication inter-véhicules qui permet de rendre la route plus sûre en transmettant des informations routières, tout en diminuant la congestion des axes routiers et l'émission des gaz à effet de serre.

De plus, l'article [16] présente une solution de communication inter-vehicules permettant de passer par les réseaux cellulaires et véhiculaires pour transmettre des données.

C'est aussi le sujet de [4] qui fait état des solutions présentes en 2008 pour assurer une communication V2X et ainsi transmettre des données relatives aux événements routiers (freinages d'urgence, obstacles, ...).

L'article [17] présente, quant à lui, une architecture évaluant l'état du trafic sur un axe routier grâce aux différents équipements placés aux bords des routes et dans les véhicules, dans le but d'optimiser le trafic de l'axe routier en question.

Ces sujets sont fédérateurs puisqu'ils regroupent de nombreux acteurs d'un même domaine d'activité autour d'un même sujet que sont les C-ITS. C'est

ce qu'illustre le rapport [18] qui fait état des résultats de l'application VSC-A. Elle regroupe plusieurs constructeurs automobiles comme Toyota ou Ford ou encore Mercedes-Benz et utilise les DSRC pour transmettre des informations de sécurité routière.

Les innovations en matière de systèmes coopératifs alimentent d'ailleurs les avancées vers le véhicule autonome. En effet, l'article [19] nous présente, au travers de plusieurs projets, des systèmes d'aide à la conduite de plus en plus perfectionnés et autonomes.

Nous dégageons donc que les C-ITS, dans leurs généralités, ont pour but :

- d'augmenter la sécurité routière pour les usagers de la route et les agents d'exploitations [6] [15] [18] [4] [16] [3] [13] [19] [1],
- de gérer plus efficacement l'état du trafic via un apport d'informations en temps réel [6] [15] [4] [17] [3] [1],
- d'optimiser les flux de trafic afin de réduire la congestion des axes routiers et les émissions de gaz à effet de serres [15] [3] [14] [13] [20] [1],
- et enfin, de s'inscrire dans le véhicule collaboratif de demain [1] [19].

2.2.3 Faisabilité d'une telle solution

La perspective de mise en place de réseaux véhiculaires n'est pas irréalisable, de nombreux travaux vont dans ce sens. Des études, sur simulateurs, ont été effectuées pour évaluer les performances de tels réseaux avec des conditions de fonctionnement variables.

L'article [7] analyse un scénario mettant en jeu des communications véhiculaires. L'auteur met en place un modèle de mobilité de type *Manhattan Grid*, afin d'étudier le trafic. Il joue, ainsi, différents scénarios dans un simulateur afin de déterminer les performances de son modèle, il en dégage les latences, les paquets perdus, etc. Via ses travaux, l'auteur donne des indications sur la faisabilité de telles technologies dans un environnement spécifique.

De même, l'article [21] illustre cette faisabilité en réalisant une étude de mise en situation de réseaux véhiculaires sur le simulateur NS-2. Les auteurs utilisent un modèle de simulation de Monté-Carlo pour soumettre le réseau à plusieurs scénarios possibles et à différentes conditions (différentes portés et conditions de transmissions, noeuds mobiles changeants). Malgré des performances relevées plus basses de ce qui était prévu par les auteurs, assurer des transmissions dans ce type de réseaux semble réalisable. Il apparaît même que, sur de courtes distances, les délais de transmissions ne sont pas trop affectés par les différentes conditions paramétrées.

Des déploiements en situations réelles sont également testés. Prenons pour exemple [22], il réalise un aperçu des différents projets européens et surtout allemand, expose les motivations de tels projets et donne des clés sur les bénéfices attendus. L'auteur évalue ensuite l'efficacité et le réalisme des réseaux véhiculaires déployés.

2.2.4 Cas d'usages des C-ITS

Nous avons vu précédemment que les VANETs ont pour but d'augmenter la sécurité routière. Avec l'objectif d'explicitier cet aspect de sécurité routière, nous pouvons dégager ces cas d'usages.

L'article [23] présente différents cas d'usages auxquels les VANETs peuvent y répondre, comme l'affichage embarqué de la signalisation routière, l'augmentation de la perception environnementale de l'utilisateur (qui est informé d'évènement qu'il ne peut, potentiellement, pas voir), la transmission des conditions météorologiques et de l'état du trafic, et enfin, de l'assistance à la conduite. L'article [24] fait de même en présentant les points précédents et en y ajoutant la préparation du véhicule à un impact soudain.

De plus, nous avons l'article [20] qui ajoute des notions de gestion de trafic avec l'eco-conduite et de cas d'urgences tels que faciliter la circulation des véhicules d'urgences ou la mise en place d'itinéraires d'urgence.

L'usage de communications C-ITS peut s'effectuer dans le cadre :

- de la transmission de données véhiculaires (position géographique, vitesse, direction, caractéristiques du véhicule),
- d'un évènement provoqué par l'utilisateur du véhicule (freinage d'urgence, impact),
- de conditions nécessitant le signalement du véhicule (avertisseurs lumineux actifs, véhicules à l'arrêt),
- d'évènements signalés sur la route (accident, piétons sur la route, chaussée glissante, chaussée en travaux, conditions météorologiques difficiles ...),
- d'informations sur le trafic (feu de signalisation, panneaux à messages variables, signalisation, ...),
- d'informations sur les infrastructures routières (localisation et disponibilité de places de parking).

2.3 Pile protocolaire des C-ITS

Nous allons étudier deux piles C-ITS actuellement utilisées, la pile américaine, utilisant la norme DSRC et la pile européenne, que nous détaillerons, qui utilise la norme European Telecommunications Standards Institute (ETSI) [25] ITS-G5.

2.3.1 DSRC/Wave - Norme US

Aux Etats-Unis, la mise en place de réseaux C-ITS est assurée par le système *Wireless Access in Vehicular Environments* (WAVE) permettant la communication V2X, un système issu des *Dedicated Short-Range Communications* (DSRC) intervenant sur la bande de fréquence des 5.850-5.925 GHz [26] [27]. L'architec-

ture de ce système et son fonctionnement se basent sur la série de normes IEEE 1609 (elles-même basées sur l'IEEE 802.11p) [28].

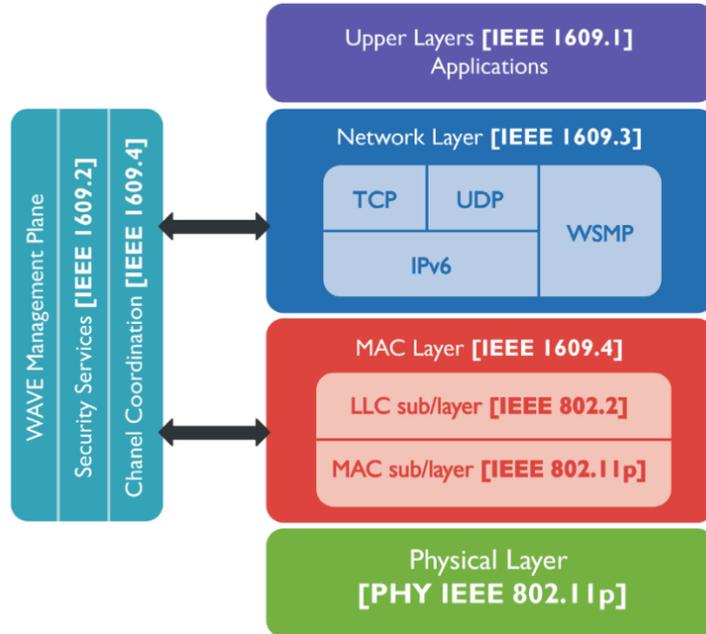


FIGURE 2.1 – Architecture du protocole WAVE : IEEE 802.11p et IEEE 1609 [29]

La pile américaine se répartit comme suivant la figure 2.1 :

- la *Physical layer* et *MAC layer* se basent sur l'IEEE 802.11p. Elles fixent les éléments permettant la communication au sein de VANETs et le canal de communication employé dans la transmission. Deux types de canaux sont définis : le canal de contrôle (CCH) qui transmet les services mis à disposition par les différents équipements C-ITS, canal qui, par ailleurs, est scruté périodiquement par ces équipements, et les canaux de service (SCH) qui permettent la transmission de données,
- la *Network layer* définit différents protocoles permettant l'échange de messages. L'IEEE 1062.3 définit deux structures de messages permettant de véhiculer les informations de la couche application : Le *WAVE Service Advertisement* (WSA) et le *WAVE Short Message* (WSM),
- l'*Upper layer* constituant le message à envoyer.

Détaillons, à présent, les deux types de messages véhiculés par le WSMP :

- WSA - permet de fournir les services DSRC disponibles sur un équipement C-ITS. Ce type d'information est surtout diffusé par les RSUs mais peut être aussi diffusé par les OBUs. Ces messages sont émis sur le CCH,
- WSM - message permettant la transmission de données entre différents équipements C-ITS. Ces messages sont émis sur un SCH fourni dans le message WSA ayant annoncé la disponibilité de l'information.

Différents types de messages peuvent être envoyés notamment ceux du stan-

dard SAE J2735, reconnu par l'*United States Department of Transportation* (USDOT) pour les C-ITS (Présenté dans la figure 2.2.

Message Type	Purpose
A La Carte Message	Generic message with flexible content
Basic Safety Message	Conveys vehicle state information necessary to support V2V safety applications
Common Safety Request	A vehicle uses this to request specific state information from another vehicle
Emergency Vehicle Alert Message	Alerts drivers that an emergency vehicle is active in an area
Intersection Collision Avoidance	Provides vehicle location information relative to a specific intersection
Map Data	Sent by RSU to convey the geographic description of an intersection
NMEA Corrections	Encapsulates one style of GPS corrections – NMEA style 183
Probe Data Management	Sent by RSU to manage the collection of probe data from vehicles
Probe Vehicle Data	Vehicles report their status over a given section of road; aggregated to derive road conditions
Roadside Alert	Sent by RSU to alert passing vehicles to hazardous conditions
RTCM Corrections	Encapsulates a second style of GPS corrections – RTCM
Signal Phase and Timing Message	Sent by RSU at a signalized intersection to convey the signal's phase and timing state.
Signal Request Message	A vehicle uses this to request either a priority signal or a signal preemption.
Signal Status Message	Sent by RSU to convey the status of signal requests.
Traveler Information	Sent by RSU to convey advisory and road sign types of information

FIGURE 2.2 – Tableau des messages DSRC du standard SAE J2735 [30]

Maintenant que nous avons une idée de comment fonctionnent les C-ITS outre-atlantique, observons les du côté européen.

2.3.2 ETSI ITS-G5 - Norme EU

Les projets Européens, relatifs aux C-ITS, se basent sur les standards fixés par l'*European Telecommunication Standards Institute* (ETSI).

Ces standards se basent sur l'ETSI ITS-G5 (se basant lui-même sur une sorte de wifi véhiculaire nommé IEEE 802.11p, ce qui rend l'ITS-G5 compatible avec cette norme).

En Europe, l'ITS-G5 utilise la bande des 5.9 GHz [31].

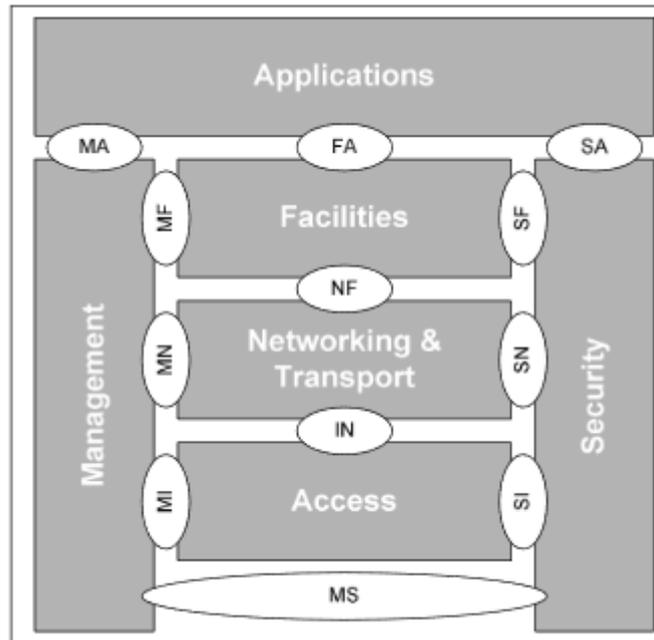


FIGURE 2.3 – Architecture de référence pour une station ITS [32]

La pile protocolaire Européenne se répartie comme suit (Figure 2.3) :

- la couche *Access* correspondant à la couche Physique et Liaison du modèle OSI. Elle caractérise l'ITS-G5,
- la couche *Networking and Transport* comportant le protocole de routage que nous utilisons : GeoNetworking et le *Basic Transport Protocol* (BTP) qui présentent un numéro de port et un numéro de sous-port pour identifier le type de message,
- la couche *facilities* contenant le message qui peut être de type : Cooperative Awareness Messages (CAM) (Section 2.3.5.1), Decentralize Event Notificaiton Messages (DENM) (Section 2.3.5.2), Signal Phase and Time (SPAT) (Section 2.3.5.3), Map Data (MAP) (Section 2.3.5.3).

La couche *Security* assure la sécurité de la Pile et la couche *Management* assure la coordination de l'ensemble.

2.3.3 *GeoNETworking*

La couche GeoNetworking permet le routage du paquet dans cet environnement mobile utilisant l'ITS-G5 [33]. Elle contient trois en-têtes : Le *Basic Header* définissant notamment le temps de vie du paquet et si ce dernier est sécurisé, le *Common Header* définissant notamment le type de *Payload* et le type d'*Extended Header*, et enfin, l'*Extended Header* (optionnel) qui contient les informations de routage.

Il existe différents *Extended Header* qui influent sur le routage du paquet [33] :

- le *Beacon* - ne transmet qu'une position d'une station et sans *payload* à la suite,
- le *GeoUnicast* (GUC) - il permet l'envoi du paquet à une station ITS particulière,
- le *GeoAnycast* (GAC) - il permet l'envoi d'un paquet vers une zone de destination sans se diffuser en *broadcast*, une fois la zone atteinte,
- le *GeoBroadcast* (GBC) - il permet l'envoi d'un paquet vers une zone de destination où il se diffusera en *Broadcast*, une fois la zone atteinte,
- le *Topologically-Scope Broadcast* (TSB) - il permet d'envoyer un paquet multi-saut à ses voisins, permettant ainsi d'atteindre des stations plus distantes. Dans le cas d'un transmission avec un seul saut, il s'agit d'un *Single-Hop Broadcast* (SHB),
- le *Location Service* (LS) - il permet la mise à jour de la table des voisins des stations ITS via les paquets *LS Request* et *LS Reply*.

Pour se faire, le GeonetWorking utilise quatre algorithmes de routage :

- le *Greedy Forwarding* (GF) - Il détermine le voisin à cibler, le plus proche de la zone de destination du message et dans le voisinage du véhicule,
- le *Contention Based Forwarding* (CBF) - Ici, le véhicule temporise en stockant le message dans un *buffer* afin d'observer s'il le reçoit de nouveau, si c'est le cas, le message est supprimé du *buffer*. Cela a pour but d'éviter les collisions de messages. L'algorithme agit de telle sorte que la station se trouvant au plus proche de la zone de destination, diffusera le message,
- le *Simple Forwarding* - Le véhicule, dans le cas où il se trouve dans la zone de destination du message, va rediffuser les messages qu'il reçoit en *broadcast* afin qu'ils soient captés par l'ensemble de ses voisins. Dans le cas où il ne se situe pas dans la zone de destination, il émettra le message à son voisin le plus proche de la zone visée,
- l'*Advanced Forwarding* - Ce routage permet d'affiner le ciblage effectué par le CBF et ainsi de réduire les délais mis en cause dans ce type d'algorithme. Pour se faire, le récepteur d'un message va contrôler s'il se trouve dans une zone de contention. Cette zone est un angle calculé entre lui, l'émetteur et le centre de la zone cible. Ainsi, si le récepteur se considère dans la zone de contention, il réémettra le message suivant les mécanismes du CBF, sinon, il le supprimera. L'émission ciblera donc un nombre réduit de véhicules qui seront potentiellement plus concernés par le dit message.

2.3.4 La couche *Security*

La sécurité encapsule la couche GeoNetworking afin d'assurer l'intégrité du message, tout en garantissant que la donnée en elle-même reste en clair, afin d'être consultable par n'importe quelle station le recevant, permettant ainsi d'être le plus "ouvert" possible à un maximum d'utilisateurs.

La sécurité va ajouter une en-tête, le *Secure Header*, contenant l'ensemble des informations nécessaires aux vérifications du message. Puis, à la suite du paquet, un *Secure Trailer* va être ajouté contenant le certificat et le message signé.

Ce certificat est obtenu par un système de PKI et contient la clé publique de l'émetteur ainsi que le hash du certificat de l'autorité de confiance.

2.3.5 Messages de la couche *facilities*

2.3.5.1 *Cooperative Awareness Messages - CAM*

Le *Cooperative Awareness Message* ou CAM est un message de mise à jour de position, diffusé par les véhicules en *Broadcast*, en permanence, à leurs voisins (à un saut).

Ces messages ont une fréquence d'envoi variable afin de ne pas saturer le réseau des zones de circulation dense [34].

Ils permettent d'avoir une vue de l'état du véhicule à un instant T.

Ainsi, ces messages transmettent :

- la vitesse du véhicule,
- sa position GPS,
- la direction du véhicule par rapport au Nord,
- le statut des éclairages du véhicule,
- éventuellement, le rôle particulier d'un véhicule et les caractéristiques spéciales associées,
- etc ...

Grâce à ces messages, chaque véhicule, dans un cadre C-ITS, est conscient de son environnement.

2.3.5.2 *Decentralized Event Notification Messages - DENM*

Le *Decentralized Event Notification Message* ou DENM est un message ponctuel de caractérisation d'évènement.

Le DENM contient notamment la position GPS de l'évènement, sa durée de validité, l'identification de l'évènement (Paragraphe suivant) ou encore la zone concernée [35].

Les évènements sont désignés par un couple d'identifiants présent au sein du message : les *Cause Codes* et *SubCause Codes*. Voici les principaux cas identifiés dans ces messages (ensemble des couples *Cause Codes* - *SubCause Codes* disponibles en annexe A) :

- *accident - unsecuredAccident* - un accident non sécurisé par les agents de la routes ou les secours,
- *roadworks - majorRoadworks* - des travaux sur la route,
- *vehicleBreakdown - lackOfFuel* - un véhicule en panne de carburant sur la chaussée,
- etc...

Ces messages peuvent être émis par un gestionnaire routier, par un utilisateur de la route ou automatiquement grâce aux capteurs embarqués du véhicule.

C'est au final, avec l'ensemble des couches, un message qui va osciller aux alentours des 100 octets (mesure relevé sur un DENM envoyé par un véhicule stationnaire).

2.3.5.3 Signal Phase and Time - SPAT / Map Data - MAP

Nous détaillons brièvement ces deux types de messages car nous ne les utilisons pas lors de ces travaux.

Le *Signal Phase and Time* ou SPAT renseigne de l'état des feux de signalisation d'un carrefour à un véhicule.

Le *Map Data* ou MAP fournit la topologie d'un carrefour à un véhicule.

2.3.6 Autre réseau des C-ITS

Les C-ITS en Europe ne se limitent pas à la communication en ITS-G5.

En effet, la couverture de ce type de réseau n'étant pas encore développée, des solutions d'hybridation de réseaux sont mises en place afin de conserver une qualité de service la plus optimale possible.

Certains travaux traitent déjà d'hybridation de réseau, comme l'article [36] qui présente une hybridation WIFI Direct - cellulaire, ce qui constituerai une alternative au WAVE/DSRC, ou, l'article [37] qui présente également une alternative du WAVE/DSRC avec une combinaison de WIFI Direct - cellulaire - Zigbee, ou encore, l'article [38] qui présente une hybridation entre l'IEEE 802.11p et la 3G.

Lors de ces travaux, nous avons eu l'occasion de travailler sur une hybridation ITS-G5/Cellulaire [39].

Le but de cette solution est de fournir, via le réseau cellulaire, des DENMs aux véhicules concernés.

Il y a une mise en place d'une duplication de l'envoi des CAMs et DENMs. Ils

sont ainsi émis sur l'ITS-G5 et sur le cellulaire par le véhicule. Une station centrale (ITSS-C), gérant l'ensemble des véhicules, récupère les données provenant du cellulaire.

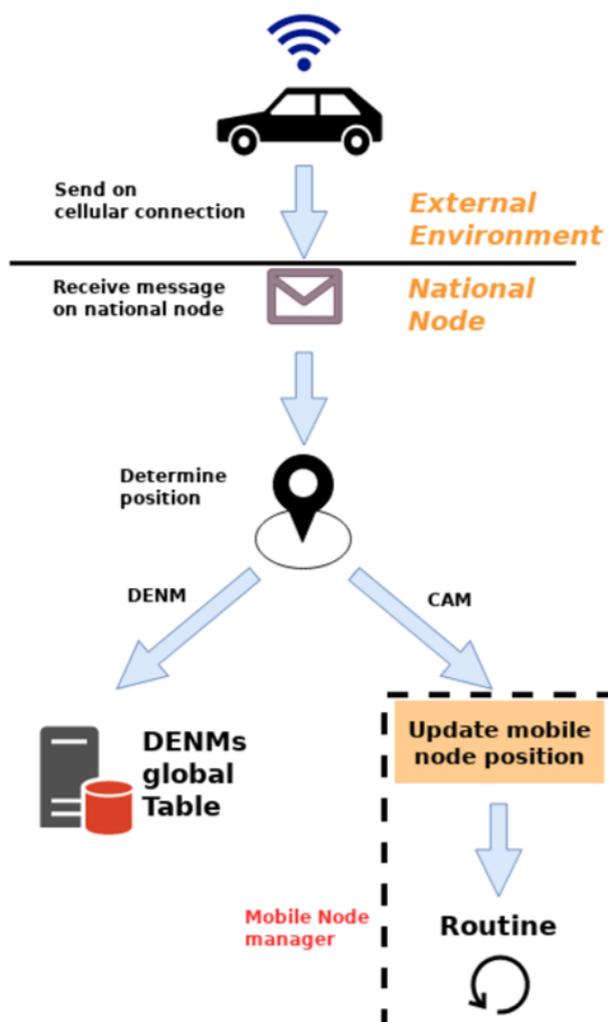


FIGURE 2.4 – Architecture de la station centrale [39]

La figure 2.4, illustre le traitement des messages collectés. Le véhicule transmet les messages sur le cellulaire. Le serveur, les recevant, va identifier la position du message via les données présentes dans le GeoNETworking.

Ensuite, en fonction du type de message reçu :

- DENM - Le message est stocké dans une base de donnée référençant l'ensemble des évènements connus en fonction de leur localisation. Il est donc possible, pour une localisation donnée, de sortir les évènements en cours dans cette zone.
- CAM - Le message caractérisant une mise à jour de la position d'un

véhicule permet plusieurs choses. Soit le véhicule est connu de l'ITSS-C, le manager gérant le véhicule mettra donc sa position à jour en fonction des données du CAM. Soit le véhicule n'est pas connu de l'ITSS-C, un manager sera donc créé afin de gérer le véhicule.

La connexion cellulaire s'effectue dans un canal sécurisé assurant l'intégrité et l'authenticité des données y circulant.

Afin de s'assurer que le manager gère toujours le même véhicule, le canal de communication ne se ferme pas. En l'absence de communication sur le canal pendant un certain temps, le manager considérera la connexion comme perdue et fermera le canal. Si le véhicule rétablit une connexion cellulaire, il sera considéré comme un nouvel équipement à gérer et donc sera géré par un nouveau manager.

Une routine est exécutée périodiquement sur ces managers afin de fournir des données au véhicule.

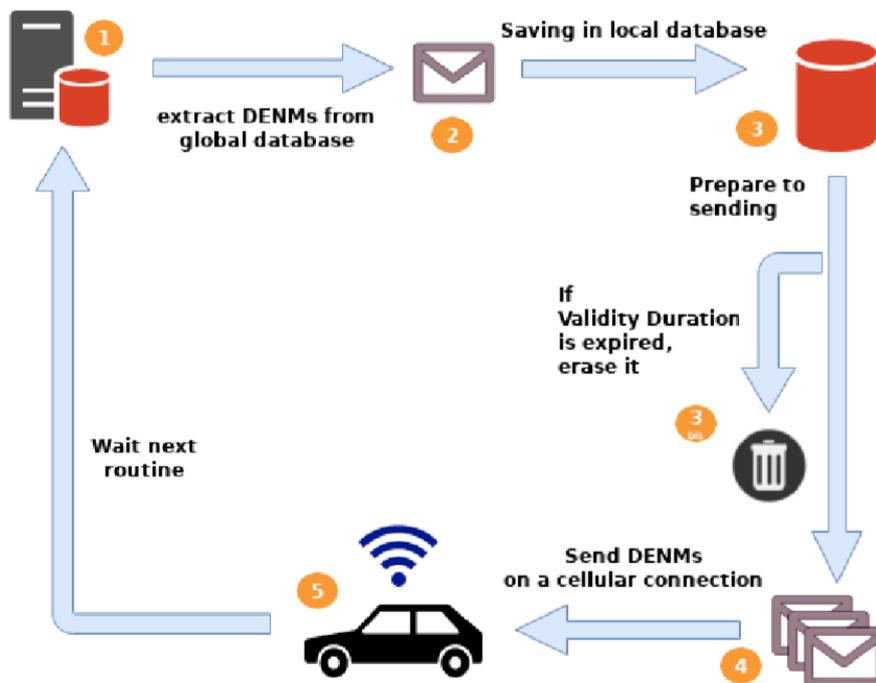


FIGURE 2.5 – Routine présente sur l'ITSS-C [39]

La figure 2.5 présente les différentes étapes de cette routine.

- Étape 1 - le processus accède à la base de données de l'ITSS-C.
- Étape 2 - les DENMs situés dans la zone de la dernière position du véhicule sont collectés.
- Étape 3 - ces DENMs sont stockés dans une base de donnée locale. Elle contiendra ainsi l'ensemble des évènements concernant le véhicule.
- Étape 3bis - Si les DENMs sont expirés, ils sont supprimés et archivés sur la base de données de l'ITSS-C.
- Étape 4 - les DENMs sont envoyés par le cellulaire vers le véhicule cible.

- Étape 5 - l'OBU reçoit les DENMs et est en mesure de les afficher sur l'Interface Homme-Machine (IHM) du véhicule.

Par conséquent, cette solution assure une solution viable d'hybridation de communication pour les C-ITS et a fait l'objet d'une communication avec acte [39].

2.4 Les C-ITS du point de vue des Gestionnaires routiers

Dans l'objectif de s'intégrer dans les C-ITS, les gestionnaires ont normalisés leurs moyens d'échanger des informations. Ils se sont accordés sur une format Européen pour la transmission de leurs données : Le DATEX II.

Les DATEX II sont des formats de messages échangés entre les RSUs et les serveurs centraux. Des serveurs aux RSUs, le DATEX II transmet des données relatives à un DENM et, des RSUs aux serveurs, le DATEX II permet la remonté de données (DENMs déclarés depuis les véhicules ou remonté de CAMs afin qu'ils soient agrégés sur le serveur). Donc, dans le cas où un gestionnaire souhaite déclarer un DENM sur une RSU, il envoie un DATEX II à l'équipement qui va effectuer la traduction en DENM afin de l'envoyer au véhicule. De même, si un véhicule déclare un DENM, il est capté par la RSU, qui va le traduire en DATEX II afin de l'envoyer aux serveurs centraux.

Ce format de message est normalisé au niveau Européen, les Gestionnaires routiers de différents pays peuvent donc s'échanger ce type de message et assurer ainsi une coopération internationale.

2.5 Les réseaux personnels sans fil

Dans cette section, nous allons expliciter différents réseaux courtes portées disponibles afin de déterminer s'ils peuvent répondre à nos attentes et donc être utilisés au sein de nos travaux.

Nous détaillerons les réseaux suivants :

- Zigbee,
- Z-Wave,
- Bluetooth Low Energy (BLE).

Nous ne traiterons pas du WIFI, étant donné que l'ITS-G5 se base déjà sur l'IEEE 802.11p (un dérivé du IEEE 802.11a), nous essayons d'étudier des alternatives à ce type de réseau.

2.5.1 ZigBee

[40] C'est en 1997 que la Zigbee Alliance fut formée dans le but de fournir une solution de communication open-source, fiable, peu coûteuse en énergie et sans fil. Des réseaux s'apparentant à cette technologie voient le jour mais c'est en 2004 que la version 1.0 de la spécification est publiée.

[40] [41] Le protocole de communication radio nommé Zigbee se base sur le protocole IEEE 802.15.4, un protocole sans fil de faible consommation et portée. Ainsi, il intervient sur 3 bandes de fréquences se basant sur celles de ce protocole : La bande des 868 MHz, des 915 MHz et des 2.4 GHz.

[40] [42] Zigbee est un protocole de routage de haut niveau permettant la mise en place des maillages réseaux complexes dont les différents appareils Zigbee seraient les noeuds. Ainsi, cette technologie permet de contourner des obstacles réseaux en déterminant des routes afin d'assurer la transmission de l'information d'un nœud à un autre.

Nous avons 3 types d'équipements possibles dans un réseau Zigbee [43] :

- Le Coordinateur, il organise le réseau et maintient les tables de routage à jour.
- Le Routeur qui peut échanger avec les différents noeuds du maillage.
- Le Dispositif terminal qui ne peut échanger qu'avec soit le coordinateur soit un seul routeur.

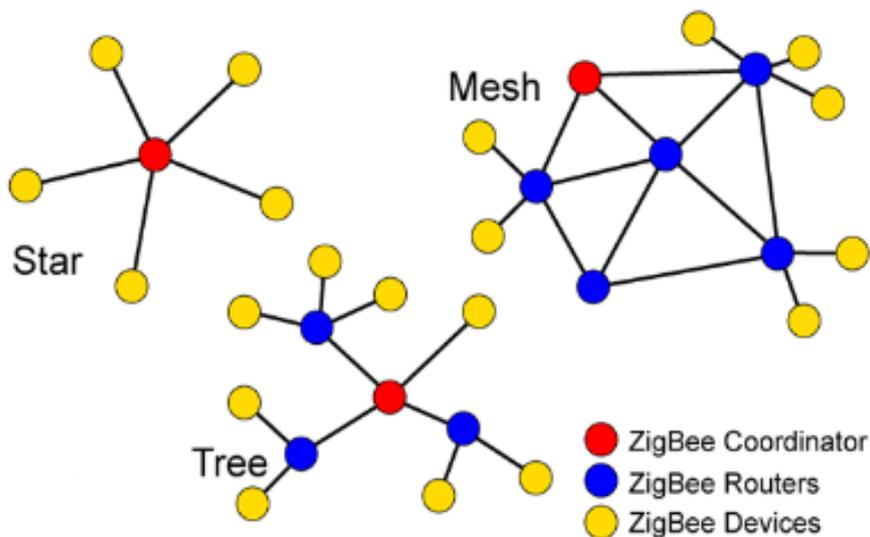


FIGURE 2.6 – Les différentes topologies réseaux. [44]

Ces différents rôles permettent, comme l'illustre la figure 2.6, la mise en place de 3 topologies possibles : les réseaux en étoile, les réseaux arborescents et les réseaux maillés.

Ainsi avec ces topologies il est possible de réaliser des tables de routage et donc

de déterminer plusieurs chemins différents afin de transmettre une information d'un nœud vers un autre.

L'idée de VANETs utilisant du Zigbee n'est pas un non sens. En effet, les travaux [45] présentent un moyen de communication entre les véhicules et les infrastructures routières en utilisant du Zigbee, l'article [46] présente une utilisation de Zigbee, et plus précisément de l'algorithme *Ad-Hoc On-Demand Distance Vector* (AODV), sur le simulateur NS2 et SUMO, et l'article [37] présente une hybridation réseau utilisant du ZigBee pour constituer une alternative au WAVE/DSRC.

Cette technologie n'est pas appropriée car elle se base sur de la communication entre appareils se questionnant et se répondant mutuellement, afin de déterminer la route pour envoyer un message. Or, nos travaux, opérant dans un milieu constamment en mouvement, avec des nœuds entrants et sortants de la zone de couverture, il est difficile d'implanter ce type de technologie.

Nous avons donc choisi de ne pas retenir ce type d'approche.

2.5.2 Z-Wave

Z-Wave est un protocole de communication sans fil apparu en 1999 et développé par l'entreprise Zensys. Ce protocole permet la mise en place de réseaux maillés, il devient ainsi très présent dans le domaine de la domotique [47].

Il intervient sur la bande des 868MHz sur la bande industrielle, scientifique et médicale (ISM) [48] avec une portée théorique annoncée à une centaine de mètres.

Ce protocole offre une forte interopérabilité pour l'ensemble des appareils certifiés "Z-Wave" due à la certification assurée par la Z-Wave Alliance qui valide chaque appareils utilisant cette technologie.

Ainsi, Z-Wave permet la mise en place de réseaux maillés complexes où chaque nœuds peut communiquer avec d'autres tant qu'ils sont certifiés Z-Wave.

Dans un réseau Z-Wave, les périphériques peuvent se comporter principalement de deux manières :

- Les Contrôleurs - Ces périphériques sont capables de visualiser l'ensemble des tables de routage du réseau Z-Wave et de déterminer les routes les plus efficaces pour transmettre une information à un nœud réseau distant. Ces contrôleurs peuvent être primaires ou secondaires, dans le cadre d'un primaire, le périphérique distribuera les valeurs de "Home ID" et "Node ID" à un nouveau périphérique du réseau, un seul primaire peut être défini par réseau, le secondaire quant à lui, ne servira seulement qu'à maintenir les tables de routage [47].
- Les Esclaves - Ces périphériques n'ont pas de possibilités de gérer le réseau, ils ne contiennent que les routes menant aux nœuds qui leurs

sont assignés par le contrôleur. Ils peuvent également servir de relais si leurs configurations le permettent.

Nous ne souhaitons pas utiliser ce type de réseau car la mise en place d'infrastructures Z-Wave, notamment pour les usagers de la route, paraît assez contraignante et risquerait de diminuer notre base d'utilisateurs potentiels. En effet, cela nécessiterait d'intégrer des puces Z-Wave pour l'ensemble des usagers ce qui paraît difficilement réalisable.

2.5.3 Bluetooth

2.5.3.1 Présentation

La norme de communication Bluetooth fait son apparition en Mai 1998 avec la création de la *Bluetooth Special Interest Group* (SIG) et c'est en 1999 que la version 1.0 de la spécification est publiée [49].

Cette technologie permet de faciliter l'échange de données à courte portée entre différents appareils (téléphone, ordinateur, etc...). Sans fil, elle permet la mise en place de liaisons point à point et point à multipoints. Le Bluetooth permet ainsi de créer des réseaux locaux loin de toute infrastructure réseau et avec un simplicité notable.

La version 1.0 et 1.0B définit la fonction d'association de matériels et des spécifications d'interopérabilité entre différentes marques. Ce sont les bases de cette norme de communication.

C'est en 2010 que la version 4.0 est publiée avec l'ajout d'une nouvelle fonctionnalité. La réduction de consommation des périphériques Bluetooth avec le *Bluetooth Low Energy* (BLE) ou *Bluetooth smart*.

2.5.3.2 Bluetooth Low Energy

Le Bluetooth Low Energy (BLE) est une technologie de communication, parue, comme énoncée précédemment, dans la version 4.0 de la *Bluetooth core specification*. Cette fonctionnalité offre une solution économe en consommation énergétique pour assurer une transmission en Bluetooth.

À noter que certains appareils équipant ce nouveau type de Bluetooth (différent du Bluetooth classique - post version 4.0) ne sont pas interopérables avec les anciens types, on les désigne comme "*single-mode*". Contrairement à des appareils Bluetooth implémentant le *Low Energy* et le Bluetooth "classique" que l'on désigne comme "*dual-mode*" ou "*Bluetooth smart Ready*".

Cette technologie offre une manière de communiquer en direct entre les différents appareils, sans intermédiaire, ce qui est très différent avec ce que nous avons observé dans les autres réseaux courte portée [50].

2.5.3.3 Couches protocolaires

Nous allons décrire dans les paragraphes suivant l'architecture mise en place autour de cette technologie.

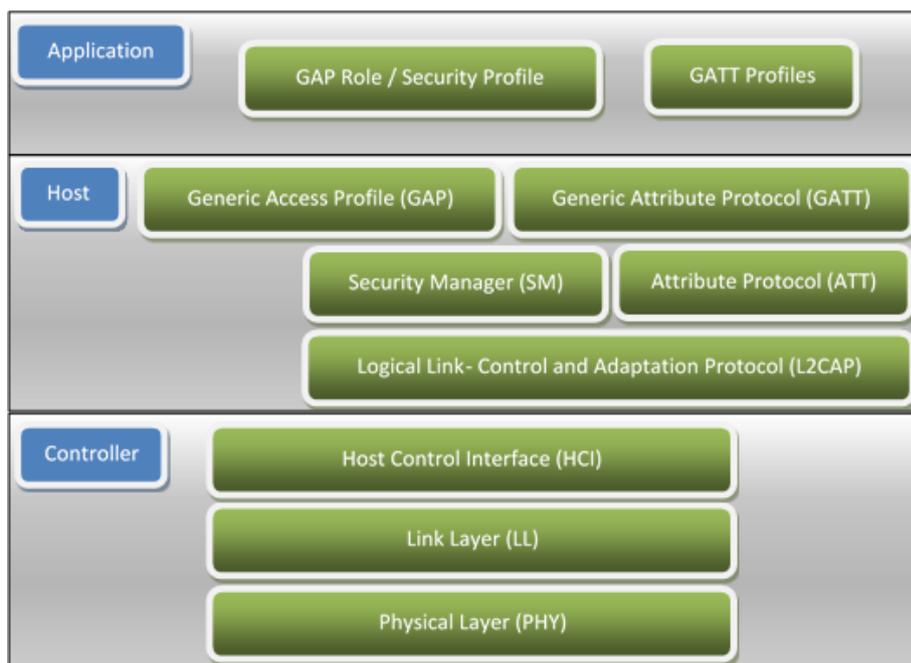


FIGURE 2.7 – Pile protocolaire Bluetooth Low Energy [51]

La figure 2.7 nous présente la pile protocolaire BLE. Elle se compose de deux éléments distincts : le *Controller*, qui se charge de l'accès radio et de tout le paramétrage nécessaire (synchronisation temporelle et des fréquences) et l'*Host* qui se charge de fonction de plus haut niveau.

Le *Host Controller Interface (HCI)* est en charge de faire le lien entre ces deux éléments.

Enfin, la partie application qui n'est pas définie dans la spécification.

La pile protocolaire BLE se compose des différentes couches suivantes :

La couche *Physical Layer*, tout d'abord, où nous avons 40 canaux disponibles. Nous les détaillons en Section 2.5.3.4.

La couche *Link Layer* définit les différents types de communications BLE.

Un appareil BLE peut se comporter de deux manières différentes

- Soit il émet une information sur les canaux d'*advertising* (voir Section

2.5.3.4), diffusant ainsi une information à l'ensemble des appareils BLE à proximité. On le désigne comme un *advertiser*.

- Soit il écoute son environnement afin de capter les messages des *advertiser*, les *advertising packet*. On désigne ces appareils comme des "scanners".

C'est sur cette couche qu'un réseau que l'on désigne sous l'appellation de "Piconet" peut se créer. En effet, un *scanner* peut repérer un *advertiser* et souhaiter s'y connecter, il établit alors une demande de connexion.

Cette connexion bi-directionnelle qui s'établit passe alors sur les "data channels" (voir Section 2.5.3.4). L'appareil dont provient la demande de connexion deviendra un *master* tandis que l'*advertiser* passera en *slave*.

- Le "master" - appareil capable de gérer des connexions multiples et de délivrer de l'information à ses interlocuteurs.
- Le "slave" - appareil qui peut uniquement échanger qu'avec un seul "master" lorsqu'une connexion BLE est établie.

La couche *Logical Link Control and Adaptation Protocol* (L2CAP) permet de multiplexer les données provenant des couches supérieures (GAP / GATT / SM / ATT) de la pile protocolaire.

Le *Security Mode* (SM) permet de fixer le mode de sécurité employé dans l'échange. Le *LE Security Mode 1* concerne la couche liaison tandis que le *LE Security Mode 2* concerne la couche ATT. C'est par les fonctionnalités offertes par cette couche que le couplage entre des appareils s'effectue; cette couche permet de garantir l'intégrité et la véracité des messages transmis. [50]

La couche *Attribute Protocole* (ATT) introduit la notion de "clients" et de "serveurs" dans la communication BLE. Ainsi, un Serveur pourra fournir des structures de données, des *attributes* tandis qu'un client pourra s'y connecter pour récupérer les dites structures de données. Ces notions sont indépendantes de la notion de "master" et "slave" mais en générale, un *master* sera un client et un *slave* sera un serveur. [52]

Ensuite, la couche *Generic Attribute Protocole* (GATT) définit les données contenues dans les *attributes*. Chaque *attribute* contient une ou plusieurs structures caractérisant tous les aspects de la donnée (ex : valeur de sonde, unité de mesure, type de mesure, ...); on appelle ces structures des *characteristics* et, des ensembles de ces structures, des *services*. On les manipule avec des opérations *read*, *write*, *notify*, *indicate* entre le client GATT et le serveur GATT. [53]

Les opérations *Read* et *Write* sont initiées par le client, à ce moment-là, le serveur accusera réception. Tandis que, les opérations *Notify* et *Indicate* sont initiées par le serveur et validées par le client, ainsi le serveur pourra envoyer des données au client.

Enfin, la couche *Generic Access Profile* (GAP) introduit quatre types de profils différents pouvant être appliqués dans une communication BLE. Les profils *Periferal* et *Central* qui font référence, respectivement, aux notions de *slaves* et *master*. Ainsi un *periferal* n'établira qu'une connexion vers un appareil BLE. Et un *central* sera capable de gérer une multitude de connexions BLE. Deux autres

profils sont fixés : le *Broadcaster* et l'*Observer*. Dans cette configuration, le *Broadcaster* diffusera des données en permanence sur les canaux d'*advertising*, et ne laisse pas la possibilité à des appareils BLE de pouvoir s'y connecter. L'*observer*, quant à lui, scrute ses environs et récupère l'ensemble des données fournies par les *broadcaster* à sa portée.

2.5.3.4 Canaux de communication

Le Bluetooth Low Energy possède 40 canaux dans la Bande Industrielle, Scientifique et Médicale (ISM) des 2.4GHz. Les canaux sont séparés de 2MHz.

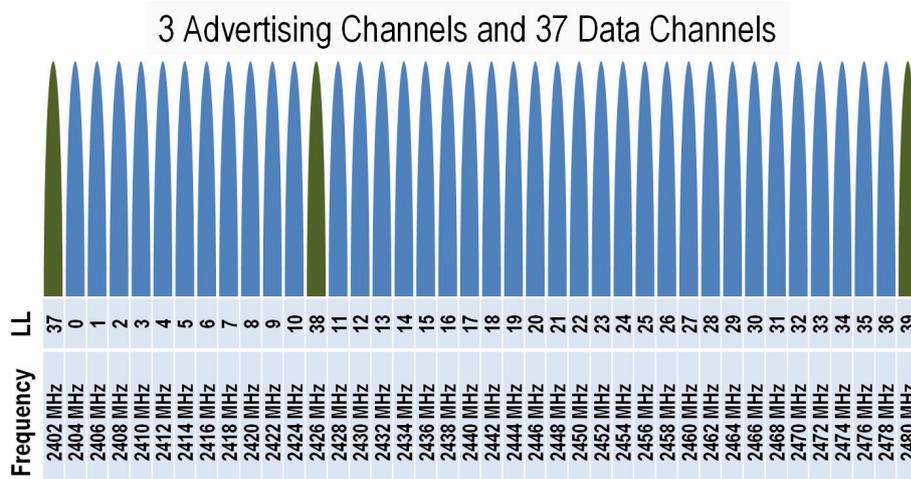


FIGURE 2.8 – Canaux Bluetooth Low Energy[52]

La figure 2.8 nous montre la répartition des différents canaux. On peut y voir 37 canaux qui sont dédiés à la transmission des données et 3 qui sont dédiés à l'*Advertising*. Afin d'éviter les problèmes d'interférences, de *fading*, de *multipath*, le BLE utilise un système de saut de fréquence adaptatif sur les canaux dédiés à la transmission de données.

A noter qu'en Bluetooth 5.0 (version récente du Bluetooth mais encore peu répandue), il est possible de diffuser des paquets d'*advertising* sur les *data channels*, ainsi l'ensemble des canaux peut être utilisés.

2.5.3.5 L'évènement de diffusion

Nous allons détailler le comportement d'un *advertiser* et d'un *scanner* lors d'un évènement de diffusion d'informations en BLE.

Lorsqu'un *advertiser* souhaite s'annoncer, nous avons vu précédemment que celui-ci envoie un message sur les canaux d'*advertising*.

La figure 2.9 nous illustre ce scénario.

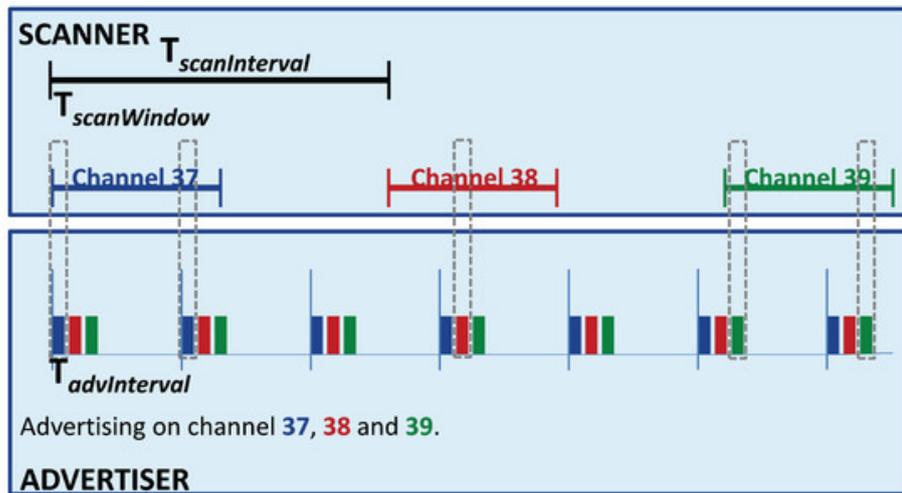


FIGURE 2.9 – Intervalles de diffusion et de récupération des paquets d'*advertising* [54]

L'*advertiser* va alors diffuser son message sur les canaux 37, 38, 39 durant un envoi. La durée de cet envoi est caractérisée par le **TadvInterval**, désignant l'intervalle de temps de diffusion entre deux envois (minimum : 20ms / maximum : 10.24s [55]). Durant cette période, l'émetteur va parcourir les 3 canaux et pour chacun diffuser l'information dessus. Une fois les canaux parcourus, l'émetteur se met en pause et attend la fin du **TadvInterval**. Puis une fois l'intervalle **TadvInterval** terminé, le processus recommence après un délais aléatoire (entre 0 et 10 ms [55]) et ce, jusqu'à l'arrêt de la diffusion.

Ainsi, le **TadvInterval** est la valeur fixant la fréquence d'envoi des messages. Plus un **TadvInterval** est bas, plus la fréquence d'envoi sera élevée.

Face à cette diffusion, un récepteur BLE, un *scanner* va scruter l'ensemble des canaux d'*advertising*, l'un après l'autre afin de capter d'éventuels *advertising packets*. Il sera actif durant une période désignée comme le *Scan Interval*, notée **TscanInterval**. Durant celle-ci, le récepteur scrute l'un des 3 canaux, et ce pendant un intervalle de temps noté **TscanWindow** désigné comme le *Scan Window*. Si un *advertiser* émet sur un canal scruté par un *scanner* durant son *Scan Window*, le récepteur recevra l'information.

Une fois un premier canal scruté durant le *Scan Window*, le récepteur va observer un temps de pause jusqu'à la fin de son *Scan Interval*. Et une fois ce dernier arrivé à expiration, cette routine recommencera mais en scrutant le canal suivant et ce, jusqu'à l'arrêt du *scanner*.

Par ailleurs, nous observons qu'un problème de synchronisation peut apparaître puisqu'un *advertiser* peut émettre un message durant le temps de pause d'un *scanner* ou inversement un *scanner* peut scruter un canal durant le temps de pause d'un *advertiser*.

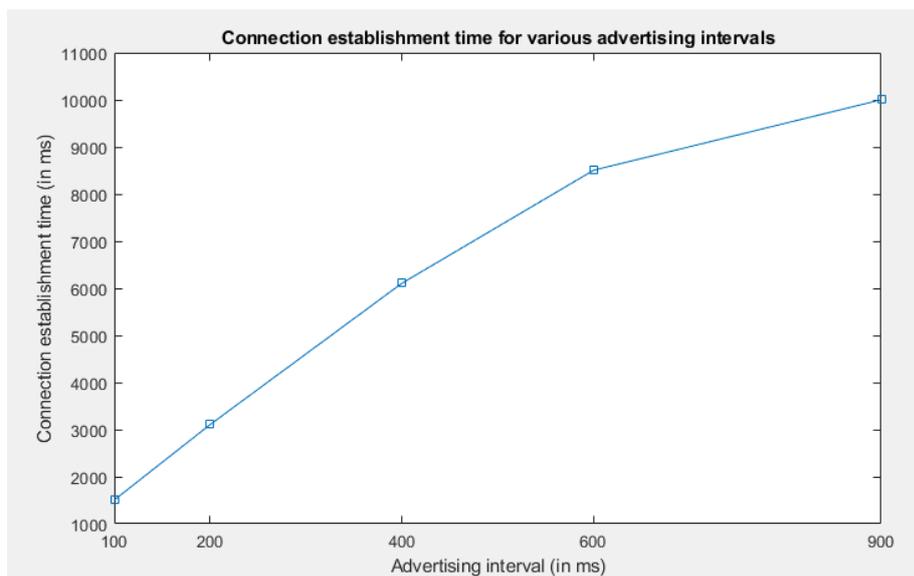


FIGURE 2.10 – Temps nécessaire à l'établissement d'une connexion BLE en fonction de l'*Advertising interval* sans interférences [56]

2.5.3.6 Évaluation d'une transmission BLE

Dans cette section, nous étudions le temps nécessaire entre un *advertiser* et un *scanner* pour initialiser une transmission de données. Nous utilisons ce résultat pour déterminer si nous pouvons utiliser les canaux de données pour nos transmissions en BLE ou si nous devons nous limiter aux canaux d'*advertising*.

Il faut, d'abord, évaluer le temps nécessaire à l'initiation d'une procédure de transmission de données, c'est-à-dire qu'un *advertiser* transmet un *Advertising Packet*, un *scanner* captant ce message y répond par un *Scan Request* lui-même transmis au dit *advertiser* qui y répondra à son tour par un *Scan Reply*.

Des simulations ont déjà été menées par [56] pour déterminer ce temps nécessaire à l'établissement d'une connexion en fonction de l'*Advertising interval*.

Les figures 2.10 et 2.11 représentent les résultats de ces simulations. Elles ont été effectuées avec les paramètres suivants :

Paramétrages pour la simulation illustrée sur la figure 2.10 :

- SCAN WINDOW : 10 millisecc.
- SCAN INTERVAL : 100 millisecc.

Paramétrages pour la simulation illustrée sur la figure 2.11 :

- SCAN WINDOW : 100 millisecc.
- SCAN INTERVAL : 100 millisecc.

Il apparaît que dans tous les cas, la procédure prend au moins une seconde à s'effectuer.

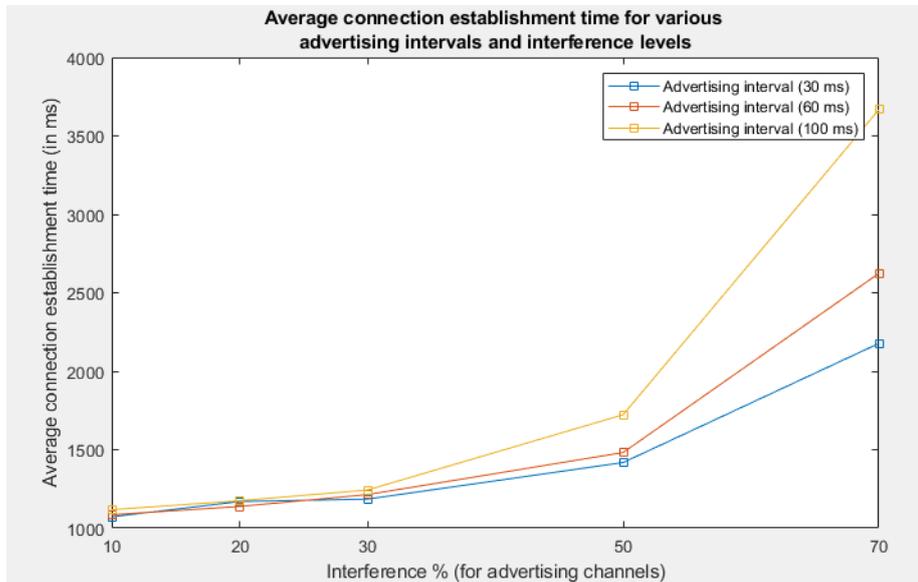


FIGURE 2.11 – Temps nécessaire à l'établissement d'une connexion BLE en fonction de l'*Advertising interval* avec interférences [56]

Maintenant essayons de transposer ces résultats dans l'environnement mobile que l'on vise.

Les dispositifs d'envoi que nous souhaitons utiliser sont munis de puces BLE de classe 1. Ils ont donc une portée de l'ordre de la centaine de mètres. Les *smartphones* et les Raspberrys Pi modèle 3B+ (équipés des puces de classe 2) n'ont qu'une portée d'une dizaine de mètres. Ces chiffres fluctuent selon la **topographie** du terrain et les **conditions météorologiques**.

Il est possible de calculer le temps théorique T lors duquel nos *scanners* vont être dans la portée des dispositifs d'envoi avec la formule 2.1.

$$T = \frac{P}{V} \quad (2.1)$$

Où P est la portée théorique du dispositif d'envoi (en m) et V la vitesse du *scanner* (en m/s).

On peut également en déduire le nombre de messages théoriquement reçus par le *scanner* via la formule 2.2.

$$N = \frac{T}{ADV_INT} \quad (2.2)$$

Où T est le temps théorique lors duquel le *scanners* va être à portée du dispositif d'envoi (en s) et ADV_INT l'intervalle de temps de diffusion de l'*Advertising Packet* (en s).

Nous considérons que dans le meilleur des cas, notre environnement se constitue de dispositifs d'envoi équipés de puces BLE de classe 1 (avec une portée théorique de 100 m) et les *scanners* embarqués, de puces de classe 2 (avec une portée théorique de 10 m).

Nous considérons également que le *scanner* est embarqué dans un véhicule circulant sur autoroute à 130 km/h, soit 36.111... m/s.

En appliquant les équations 2.1 et 2.2, nous déduisons que le *scanner*, en passage à proximité d'un dispositif d'envoi, va recevoir des messages provenant de ce dernier pendant environ 2.7 sec avec 27 messages réceptionnés théoriquement.

C'est suffisant pour établir une transmission comme celle décrite dans [56].

Mais l'appareil embarqué dans le véhicule doit lui aussi transmettre au dispositif d'envoi. Si nous appliquons ce calcul à une puce de classe 2 donc 10 m de portée, le *scanner* sera à portée théoriquement pendant 277 ms avec 2 messages diffusés théoriquement (arrondis par défaut).

Donc il est certes possible que le dispositif d'envoi ait la portée et le nombre de messages suffisants pour réaliser ce type de transmission. Mais le *scanner* risque de manquer de portée et donc de messages correctement transmis au dispositif d'envoi, pour finaliser la procédure et échanger des données.

Par ailleurs, nous avons observé dans la Section 2.5.3.5, que certains messages peuvent être perdus par les deux dispositifs BLE, ce qui ralentira d'autant plus la finalisation de la procédure.

Pour ces raisons, nous pensons que la mise en place d'une transmission, utilisant les canaux de données, est à exclure, et nous devrions, comme pour l'ITS-G5, faire du *beaconing*. Nous choisissons donc de transmettre nos données sur les canaux d'*advertising*.

2.5.3.7 Les paquets d'*advertising*

Dans ces travaux, nous avons choisi d'utiliser des appareils BLE diffusant de l'information. Similairement aux C-ITS, nous choisissons que ces appareils se comportent comme des balises, elles doivent diffuser de l'information, dans leur voisinage, de manière continue et ce, tant que cela leur est demandé.

Ainsi, nos appareils BLE *periferal* diffuseront de l'information. L'action de diffusion des paquets, par ces *advertisers*, est désignée comme l'*advertising*. Ces paquets seront ainsi captés par les *scanners* ou *centrals* à proximité.

Voici la structure d'un *advertising packet* [57] :

La figure 2.12 nous montre la structure d'un paquet transmis par un *advertiser*. On peut tout d'abord détailler les éléments du *Link Layer Packet*. Le *Preamble* sert à la synchronisation entre les différents appareils (dans le cadre de la diffusion d'informations sur les canaux d'*advertising*, ces valeurs sont fixes). L'*Access*

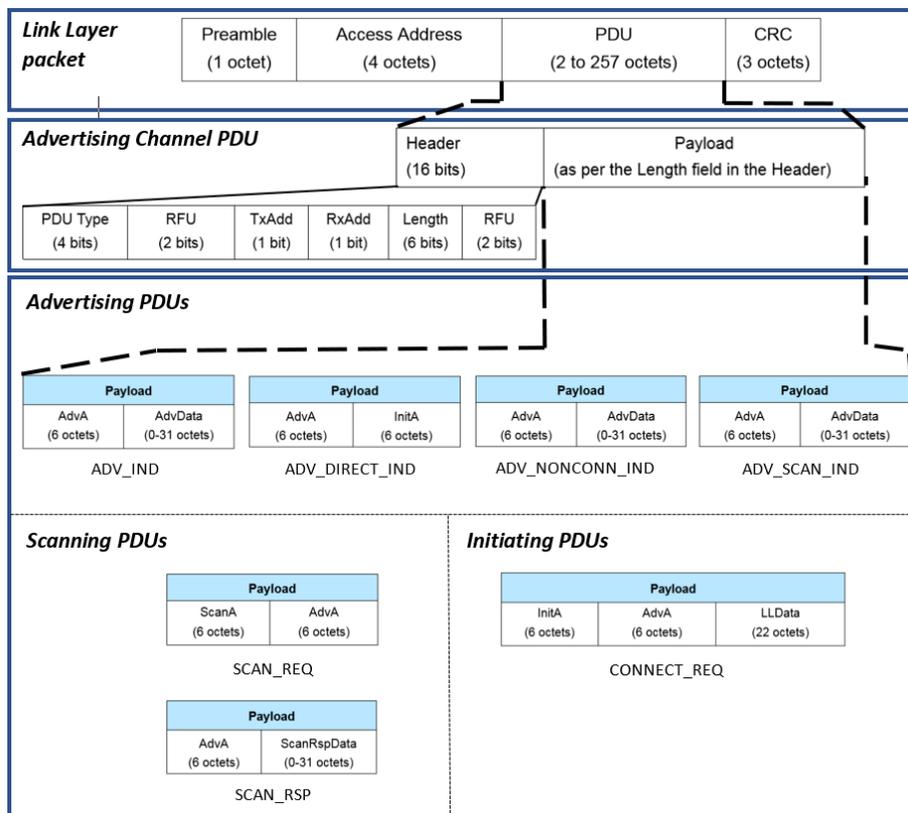


FIGURE 2.12 – Structure d'un paquet d'Advertising en BLE [55]

Address (AA) est une valeur fixe dans le cadre de la diffusion d'informations sur les canaux d'advertising. Dans le cadre d'une transmission entre deux appareils BLE, c'est une valeur aléatoire fixée par deux appareils au moment de l'initiation de la transmission sur les canaux de données afin d'éviter les collisions de paquets. La *Cyclic Redundancy Check* (CRC) est l'élément permettant de s'assurer de l'intégrité de la donnée transmise dans le paquet.

Entre l'AA et la CRC, nous avons l'Advertising Channel PDU, qui contient deux informations : le Header et le Payload.

Un Data Channel PDU peut également être présent à cet endroit mais ce type de structure ne sera pas étudiée dans ces travaux car il ne concerne pas l'advertising en Bluetooth 4.2.

A noter que le Payload est ici à un maximum de 37 octets, mais qu'avec le Bluetooth Version 5.0 (version récente du Bluetooth mais encore peu répandue), le Payload peut monter jusqu'à 255 octets.

Le Header permet de paramétrer :

- le Payload Data Unit (PDU) Type qui indique le type de Payload suivant

- dans le paquet (4 types de *Advertising PDUs*, 2 types de *Scanning PDUs*, 1 type de *Initiating PDU*),
- les deux emplacements réservés pour de futures utilisations (Reserved For Future Use - RFU) qui sont à des valeurs fixes,
 - le TxAdd qui renseigne une valeur utilisée dans les PDUs, cette valeur déterminera si l'adresse associée est publique ou aléatoire,
 - le RxAdd qui renseigne une valeur utilisée dans les PDUs (différente de celle associée au TxAdd), cette valeur déterminera si l'adresse associée est publique ou aléatoire,
 - la *Lengh* qui donne la taille du PDU.

En fonction des types de paquets diffusés, un *scanner* va avoir différents comportements en réponse. La Bluetooth SIG définit 4 types d'*Advertising PDUs* [55] [58] [59] :

- *Le ADV_IND* - diffuse un paquet avec une donnée et laisse la possibilité à un *scanner* d'initier une demande de transmission,
- *Le ADV_DIRECT_IND* - donne également la possibilité d'initier une demande de transmission mais la requête est transmise à un appareil spécifique,
- *Le ADV_NONCONN_IND* - diffuse un paquet d'*advertising* contenant une donnée, et ne permet pas à des appareils BLE d'initier une demande de transmission (c'est ce type de paquet que nous utiliserons dans ces travaux),
- *Le ADV_SCAN_IND* - Similaire à l'*ADV_NONCONN_IND*, mais le récepteur, récupérant ce paquet, peut retransmettre des informations en réponse.

Des *Scannings PDUs* et un *Initiating PDU* existent aussi, ils permettent l'initialisation de la transmission, entre deux appareils BLE, d'informations sur les canaux de données, mais ne concernent pas ces travaux et donc ne seront pas détaillés.

Structure du *ADV_NONCONN_IND* :

- l'*Advertise Address* (AdvA) qui représente l'adresse BLE (*BLE_ADDR* dans le standard) de l'appareil diffusant le paquet (adresse publique ou aléatoire fixée par le TxAdd),
- l'*Advertise Data* (AdvData) qui contient la structure de données définissant l'UUID et sa donnée correspondante.

2.5.3.8 Déclinaison existante des paquets d'*advertising*

Certains organismes proposent des versions "pseudos-standardisées" de découpage du *Payload* (voir Section 2.5.3.7). Nous devons étudier, dans un premier temps, les options existantes et ayant fait leurs preuves afin de faciliter le déploiement et le maintien de nos solutions. Nous allons donc étudier des types de *Beacon* massivement déployés et utilisés. Dans l'éventualité où ces solutions ne répondent pas à nos exigences, nous nous orienterons sur la solution des *Beacons* standard vue précédemment (voir Section 2.5.3.7).

Liste des *Beacons* étudiés et disponibles :

- Eddystone [60],
- Ibeacon [61],
- AltBeacon [62].

2.5.3.9 Eddystone

Le Eddystone est une structure de balise BLE *open source* développée en 2015 par Google [60].

Elle met à disposition plusieurs types de paquets permettant de récupérer une information. Voici les deux types que nous allons étudier :

- Eddystone-UID,
- Eddystone-URL.

Eddystone-UID

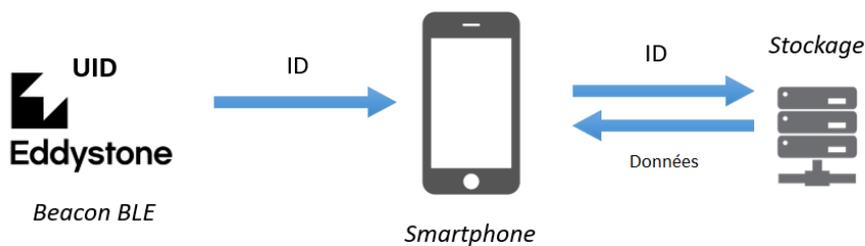


FIGURE 2.13 – Fonctionnement de l'Eddystone-UID

L'Eddystone-UID permet aux balises de diffuser, comme l'illustre la figure 2.13, un ID en BLE. Cet id permettra à l'Interface de Programmation d'Application (API) disponible sur le récepteur (ici le *smartphone*) de récupérer, sur un stockage externe, une URL afin d'aller chercher l'information sur internet.

Eddystone-URL

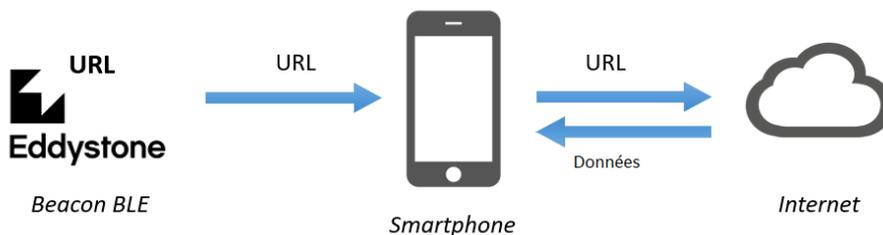


FIGURE 2.14 – Fonctionnement de l'Eddystone-URL

L'Eddystone-URL quant à lui, et comme l'illustre la figure 2.14, permet de

diffuser une URL en BLE via une balise BLE. Cette URL, une fois récupérée par un récepteur sera traitée par l'API embarquée ce qui lui permettra d'aller chercher l'information sur internet.

Deux autres types de paquets sont utilisables, à savoir l'Eddystone-EID et -TLM. La version -EID permet la diffusion d'identifiants éphémères afin d'augmenter la sécurisation des balises. Et, la version -TLM, quand à elle, fournit des informations télémétriques de la balise afin d'effectuer des actions de maintenance.

Nous en déduisons que ces différents types de balises ne correspondent pas à nos exigences. En effet, pour la plupart, il s'agit de récupérer des informations sur Internet ce qui est impossible puisque nos travaux interviennent dans le cadre de zones géographiques non couvertes par le réseau cellulaire. De plus, pour la version -UID, il est nécessaire d'effectuer une connexion vers un stockage externe afin de récupérer l'URL, or dans un environnement mobile, établir des communications entre deux appareils nous paraît difficile et sort de la philosophie de balise que l'on souhaite adopter.

2.5.3.10 iBeacon

Le Ibeacon, développé par Apple et déployé en 2013 [61], possède un fonctionnement similaire au Eddystone-UID.

Une balise diffuse un couple d'identifiants, *major* et *minor*, qui permettent d'identifier un groupe et sous groupe pour une balise sur un UUID spécifique.

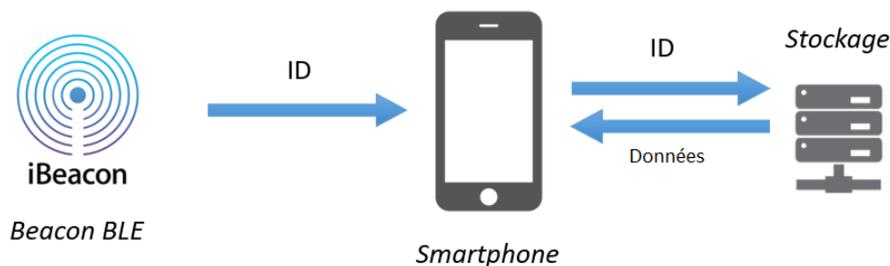


FIGURE 2.15 – Fonctionnement de l'iBeacon

Ainsi, comme l'illustre la figure 2.15, une balise va être en mesure de diffuser sa présence avec son identification (par exemple dans une ville, une marque de soda possède un ensemble de balises réparties dans la zone. Le *major* désigne la valeur correspondant à la rue et le *minor*, si la balise est sur la devanture du magasin ou à l'intérieur). L'API, qui traitera le message sur le récepteur, sera en mesure d'aller récupérer des informations sur Internet en fonction des données du message BLE et donc, de la localisation de la balise captée.

2.5.3.11 AltBeacon

Le AltBeacon est un type de balise BLE open-source [62]. Développée par la Radius Networks, cette balise permet de fournir une solution pour identifier si un *scanner* est à proximité d'un *advertiser*. Pour se faire, le AltBeacon propose une solution de balisage BLE.

Un paquet de type "ADV_NONCONN_IND" est utilisé.

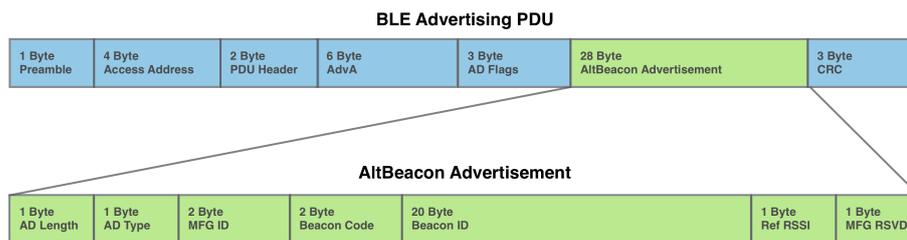


FIGURE 2.16 – structure d'un AltBeacon [63]

La figure 2.16 nous présente la structure d'un paquet d'*advertising* en AltBeacon. Nous allons détailler les 5 premiers champs. L'*AD length* désigne la taille du *Payload*. L'*AD type* désigne une structure spécifique relative au champ suivant. Le *MFG field* désigne le code de la société ayant fabriqué la balise, fixé par la Bluetooth SIG. Le *Beacon Code* désigne le code caractérisant l'AltBeacon. Le *Beacon ID* caractérise l'identification de la balise (celui-ci est séparé en deux : 16 octets pour identifier l'organisation propriétaire de la balise et 4 octets pour identifier la balise en elle-même).

Ce type de balise ne sera pas utilisé dans nos travaux. En effet, le découpage du *payload* ne correspond pas, selon nous, à nos besoins.

L'ensemble de ces déclinaisons ne nous satisfaisant pas, nous choisissons d'utiliser le paquet standard (Voir Section 2.5.3.7) dans ces travaux pour, non seulement ne pas dépendre d'API non-standardisées mais aussi, parce que cette solution propose une structuration de données correspondant à nos besoins.

2.5.4 Cas d'illustration - DP3T

Utiliser les paquets d'*advertising* pour transmettre une information importante n'est pas un non sens. Des travaux de recherche vont en ce sens. Afin d'illustrer notre propos, nous allons présenter un protocole créé dans le but de lutter contre la propagation du COVID-19, le protocole *Decentralized Privacy-Preserving Proximity Tracing* (DP3T) [64].

C'est un protocole *open source* porté par le Professeur Carmela Troncoso, de l'Ecole Polytechnique Fédérale de Lausanne (EPFL) en Suisse, et al.

Ce *White paper* consiste à définir un moyen d'établir des solutions de traçage protectrices des données personnelles. Elle est appliquée dans le cadre de la lutte contre le COVID-19 en fournissant une solution algorithmique identifiant les contacts établis par un individu via l'*advertising* BLE de son *smartphone*, tout en garantissant une anonymisation des messages.

Dans le cadre de l'utilisation de DP3T, les *smartphones* embarquent une application suivant ce protocole. Ces *smartphones* vont émettre en permanence un pseudo, via les paquets d'*advertising*, et également scruter le BLE afin de capter les pseudonymes des autres *smartphones* à portée. Ainsi, les téléphones se constituent une liste de pseudonymes qui ont été physiquement à leurs portées à un instant T lors d'une durée d et à une distance $dist$.

Ces pseudos sont constitués de la manière suivante :

- Pour la première utilisation : une clé SK_t est générée, elle constitue la première clé secrète du téléphone.
- Pour chaque jour suivant la première utilisation, cette clé change. Cette nouvelle clé SK_{t+1} est issue d'un hachage cryptographique de la clé du jours précédent. Ce hachage est effectué par une fonction H , Donc : $SK_{t+1} = H(SK_t)$
- Chaque jour, pour chaque tranche de 5 min de la journée notée n , le téléphone va établir un identifiant éphémère noté *ephID* qui constituera le pseudo **à envoyer dans le message BLE**. Cet identifiant est calculé en utilisant une fonction de hachage H' prenant en comptes : la clé du jour SK_t et le créneau n . Donc : $ephID = H'(SK_t, n)$

De plus, si un individu est testé positif au COVID-19 depuis le jour j , la clé SK_j qu'il possède est remontée à un serveur central regroupant une liste de clés d'utilisateurs testés positifs.

Le serveur envoie cette liste aux différents *smartphones*. Ils seront ainsi en mesure de calculer les *ephIDs* de ces utilisateurs positifs au COVID-19 et donc constituer une liste d'*ephIDs* atteints par le virus. L'intersection de cette liste avec la liste des *ephIDs* captés par le BLE (traduisant les contacts physiques du téléphone) correspondra à l'exposition de l'utilisateur à des personnes atteintes. Si cette exposition est jugée "à risque" par l'algorithme, cette information sera notifiée à l'utilisateur qui pourra agir en conséquence.

Voici donc un exemple d'une application de la transmission d'informations par le biais de paquets d'*advertising*, certes dans un environnement mobile plus stable que les VANETs, mais avec des changements assez fréquents et des distances variables.

Architecture Proposée

Le champ d'étude des réseaux personnels sans fil appliqués aux *Vehicular Adhocs NETWORKs* (VANETs) a peu d'occurrences. Dans la plupart du temps, l'application de ces technologies intervient dans des environnements statiques. Les appliquer dans des environnements mobiles tels que les VANETs demande une adaptation des infrastructures.

Dans ce chapitre, nous présentons une architecture de communication fiable, se basant sur un réseau personnel sans fil, et permettant de transmettre des informations routières importantes, depuis des infrastructures, aux véhicules circulant sur la route.

L'objectif de cette contribution va être de proposer une architecture de communication utilisant le réseau choisi. Elle devra être compatible avec un maximum d'appareils afin d'être ouverte à un maximum d'utilisateurs.

De même, il sera nécessaire d'explicitier clairement la philosophie que nous appliquons sur le protocole de communication et de fixer les rôles et processus de chacun des acteurs de la solution.

Par ailleurs, notre hypothèse se doit de s'intégrer dans le paysage des *Cooperative - Intelligent Transport Systems* (C-ITS) existants. La solution doit donc être en adéquation avec les technologies employées et maîtrisées par les partenaires du projet C-ITS Européen C-Roads [2].

Notre solution devra fournir un format de message léger permettant d'assurer la transmission d'une information routière la plus complète possible.

Enfin, cette architecture devra être éprouvée par des résultats expérimentaux afin de la valider.

3.1 Architecture

Tout d'abord, nous proposons notre architecture de communication utilisant le réseau personnel sans fil choisi, à savoir le Bluetooth Low Energy (BLE).

Commençons, dans un premier temps, par justifier le choix du BLE et, dans un second temps, nous fixons l'environnement ainsi que les différents éléments qui

composent notre architecture. Nous détaillons non seulement les acteurs mais aussi les aspects technologiques et les flux mis en cause.

3.1.1 Motivations du choix du BLE

Cette technologie, forte de 20 ans d'expérience, est, de ce fait, massivement déployée sur de nombreux supports sur le marché et pour de nombreuses applications, allant des smartphones au réseau *Internet Of Things* (IOT).

De cette présence massive découle deux conséquences. Non seulement, de nombreuses documentations et APIs sont disponibles sur différents supports accélérant la compréhension de ce type de communication et facilitant fortement le développement et le maintien des applications.

Mais aussi, ce déploiement massif du Bluetooth nous procure un accès à de nombreux supports déjà équipés et à des puces Bluetooth peu coûteuses. Cela nous ajoute de la liberté d'action sur nos choix d'équipements et donc une modularité accrue sur nos travaux. De plus, par le biais de cette disponibilité de la technologie, nous avons l'opportunité d'augmenter notre base d'utilisateurs sur nos systèmes C-ITS.

Nous nous sommes concentrés sur l'utilisation de la partie *advertising* disponible dans le BLE, c'est à dire la diffusion de données, par un émetteur BLE, à tout récepteur à proximité.

3.1.2 Fonctionnement de l'architecture

Définissons, avant toutes choses, l'aspect global de l'architecture.

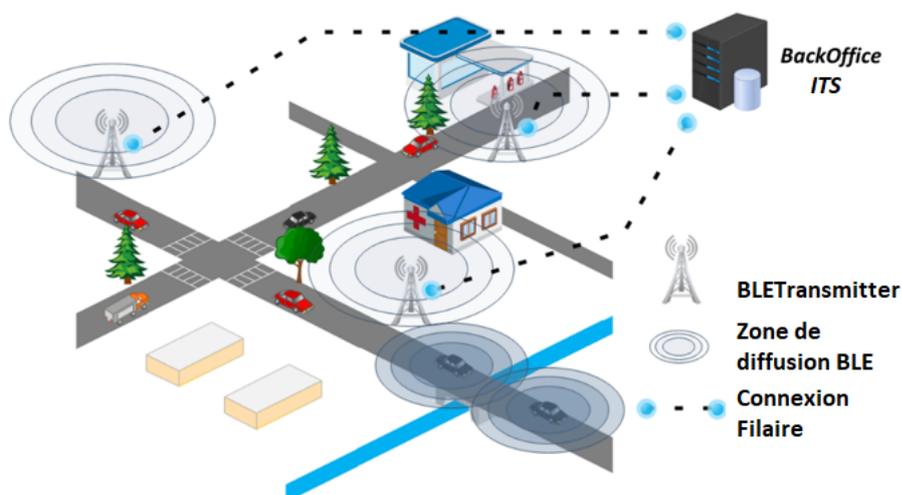


FIGURE 3.1 – Architecture globale de Communication en BLE

La figure 3.1 nous présente un environnement routier. Il est couvert par des antennes Bluetooth en *standalone* ou des *Road Side Unit* (RSU) équipées d'antennes BLE, nous les désignons, toutes deux, sous l'appellation de **BLETransmitters**. Les véhicules sont également équipés d'émetteurs-récepteurs Bluetooth, il peut s'agir d'équipements internes aux véhicules, à savoir les *On Board Unit* (OBU), ou des appareils des usagers, à savoir leurs *smartphones* (que nous intégrerons dans la désignation "OBU" dans ces travaux).

Les BLETransmitters sont reliés par liaison physique aux infrastructures des Gestionnaires routiers désignés ici par le serveur *BackOffice*. Ces acteurs seront, par ailleurs, désignés sous le nom de "**Gestionnaire**".

Grâce aux liaisons entre les antennes et les serveurs, les Gestionnaires peuvent piloter l'ensemble de leurs BLETransmitters en temps réel. Ils peuvent donc à tout moment démarrer une diffusion sur des BLETransmitters bien précis.

Des informations peuvent donc être diffusées instantanément aux véhicules à proximité des antennes, à la décision du Gestionnaire.

Les données, que le Gestionnaire transmet, sont des signalements d'évènements sur la route, or, dans les projets ITS, des messages de déclaration d'évènements, désignés comme les *Decentralized Event Notification Messages* (DENMs), sont implémentés (Voir Section 2.3.5.2). Dans ces DENMs, le type d'évènement en lui-même est encodé selon une table fixée par les standards ETSI [65] [35]. De plus, ces messages sont connus des acteurs du secteur (Gestionnaires et constructeurs automobiles) ce qui en fait une technologie maîtrisée. Cette architecture va donc s'aligner sur ces standards afin d'utiliser ces acquis (Voir Section 3.6.3 et Annexe A).

3.2 Scénario d'illustration

Mettons l'architecture en situation, afin d'avoir une idée précise de la solution.

Un évènement apparaît sur la route, par exemple, un accident. Or, la portion de route concernée se situe dans une zone non couverte par l'ITS G5 ou par le cellulaire. Les véhicules, dans la zone en question, sont donc dans l'incapacité de communiquer entre eux et avec les infrastructures. Malgré leurs outils ITS, l'information ne peut pas être diffusée par les voies actuelles de communication.

C'est à ce moment-là que nous décidons d'utiliser le Bluetooth Low Energy et plus précisément la fonctionnalité d'*advertisement*, elle permet la diffusion de l'information de manière simplifiée avec une technologie massivement déployée. Nous détaillerons le fonctionnement de cette méthode de communication dans la Section 3.5.

Le Gestionnaire va donc dans un premier temps localiser l'évènement sur la route. La géolocalisation (par Global Positioning System - GPS) précise de l'évènement permet de déterminer les balises qui se situent dans la zone concernée

et qui permettront de diffuser l'information à l'ensemble des véhicules à proximité. Le choix des balises est à la discrétion du Gestionnaire : ces derniers se réfèrent à la position GPS de l'évènement, leurs cartographies ainsi que leurs parcs de BLETransmitters pour choisir les balises les plus pertinentes à activer.

L'OBU embarquée dans le véhicule, à l'approche de la balise, capte le message, le traduit et notifie l'utilisateur de la route (Figures 3.2 a et b).

3.3 Flux de communication

Dans cette architecture, figure 3.3, nous faisons une distinction précise entre les infrastructures sur la route et les utilisateurs (véhicules et *smartphones*).

Dans cet environnement, on observe les infrastructures du Gestionnaire (*BackOffice* ITS et BLETransmitter), elles constituent le domaine de confiance. En effet, l'autorité centrale, qu'est le *BackOffice* ITS, garantit la conformité des messages. Il sera donc chargé de les valider, de les désigner et de les diffuser.

Par conséquent, la communication véhicule à véhicule (V2V) n'est pas prise en compte dans cette architecture car les informations potentiellement déclarées sur ces véhicules ne sont pas validées par une autorité de confiance (un gestionnaire, gouvernement, etc...). Malgré le fait que ce type d'étude a été initié par des articles comme [66] qui étudie la communication en V2V et qui met en place des mécanismes de transition rapides entre les différents mode de BLE afin qu'un véhicule, détectant une anomalie, puisse se comporter comme une balise pour prévenir son voisinage.

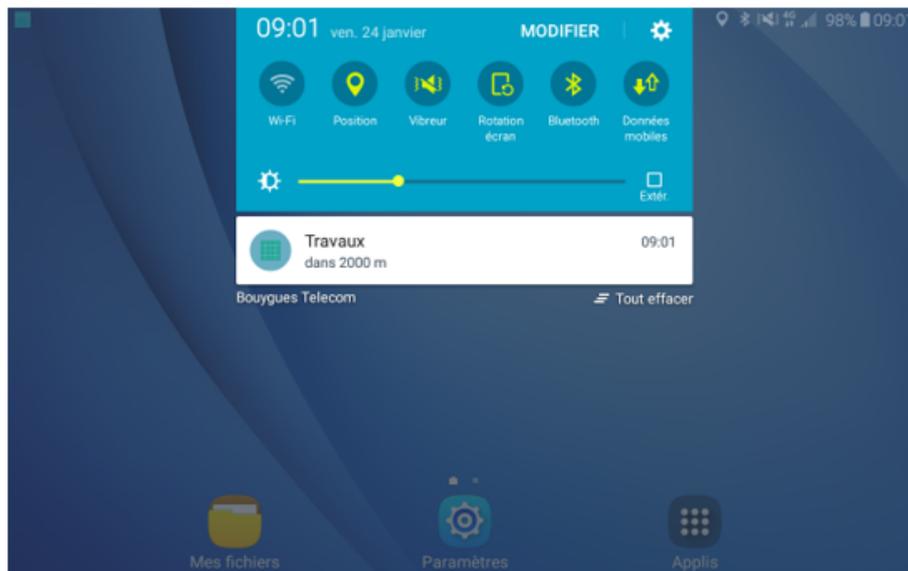
Ici, les BLETransmitters (RSUs ou Antennes BLE autonomes) étant directement reliés et pilotés par le Gestionnaire, font aussi bel et bien partie du domaine de confiance. L'ensemble des informations diffusées sur ces éléments seront donc considérées comme fiables. Ainsi, le *BackOffice* ITS émet des ordres aux BLETransmitters afin de commander la diffusion ou l'arrêt d'un message BLE.

3.4 Comportement des appareils de l'architecture

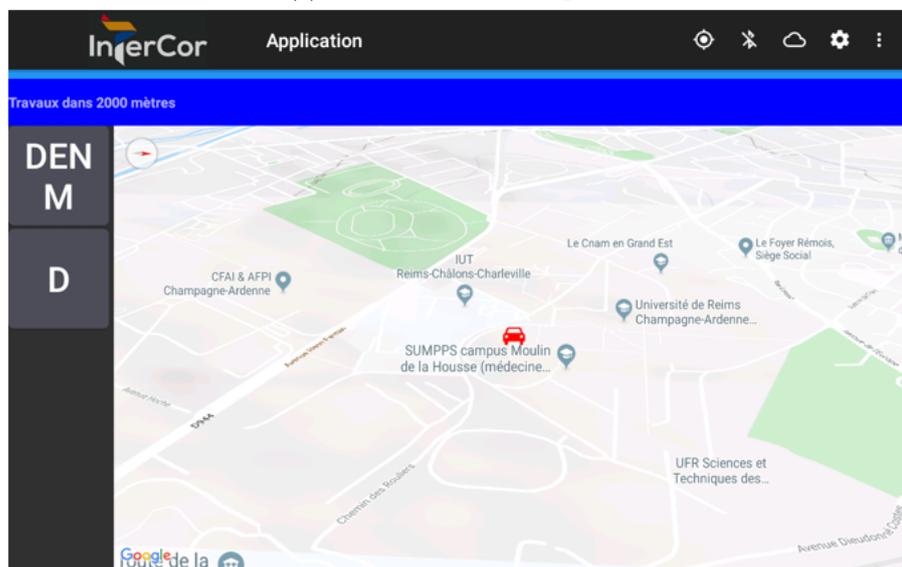
3.4.1 Sur le Backoffice ITS

Le *BackOffice* se doit de **piloter** l'ensemble de ses balises, il va donc émettre des ordres et **concentrer** des informations reçues depuis ses balises.

Sur la figure 3.4 nous pouvons voir la fonctionnalité principale du *BackOffice*. Il y a lors de la sélection d'un évènement à transmettre, une détermination des BLETransmitters concernés, c'est à ce moment-là que le *BackOffice* va envoyer l'ordre d'émission avec les paramètres de diffusion.



(a) Notification du message BLE



(b) Affichage du message BLE sur l'Interface Homme-Machine (IHM)

FIGURE 3.2 – Affichage sur l'équipement C-ITS

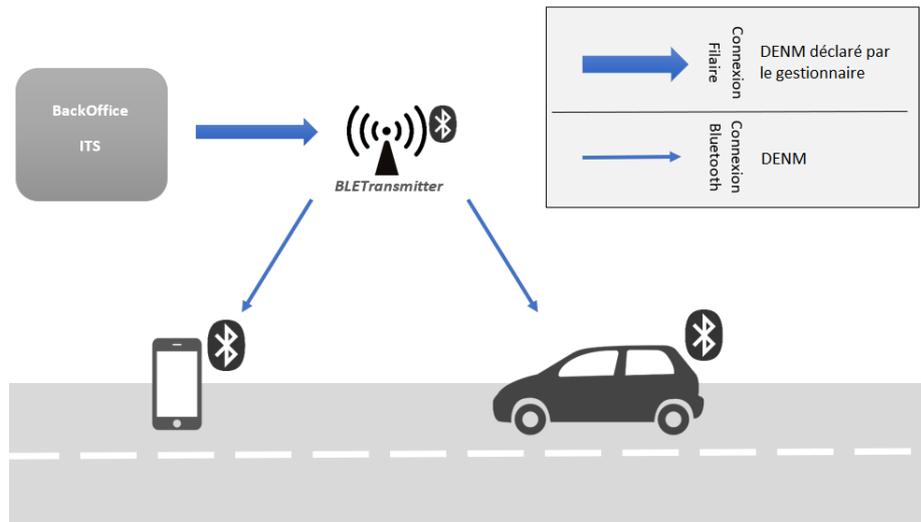


FIGURE 3.3 – Flux de communication

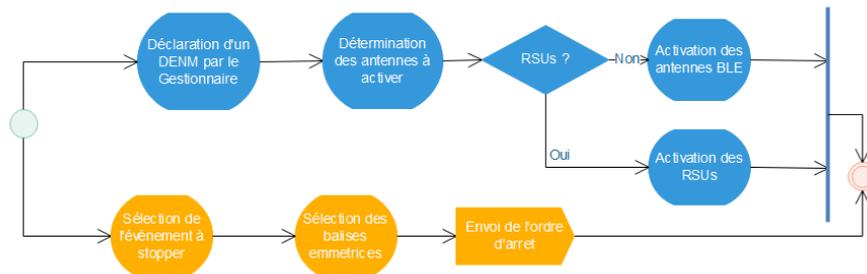


FIGURE 3.4 – Diagramme d'activité du Gestionnaire

Aussi, le *BackOffice* peut émettre un ordre d'arrêt pour différentes balises à tout moment.

3.4.2 Sur un BLETransmitter

Un BLETransmitter possède assez peu d'intelligence, cette intelligence est plutôt située du côté des serveurs *BackOffice* ITS.

Ainsi, la figure 3.5 nous montre les actions/réactions de nos BLETransmitters. L'équipement **réceptionne des ordres** provenant du Gestionnaire et les applique. On a donc la possibilité de réceptionner un ordre de démarrage de diffusion avec l'ensemble des paramètres passés dans le message, à savoir l'information et l'intervalle de diffusion (Voir Section 2.5.3.5).

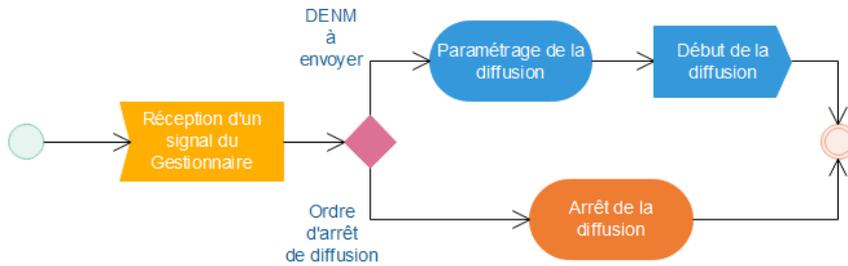


FIGURE 3.5 – Diagramme d'activité du BLETransmitter

Le BLETransmitter peut également recevoir plusieurs DENMs à envoyer, la Balise va donc s'accorder sur la fréquence d'envoi la plus élevée et alterner successivement entre les différents messages, l'un après l'autre, et ce jusqu'à réception de l'ordre d'arrêt de la diffusion.

Cet ordre peut arrêter l'un des DENMs ou l'ensemble.

A noter que certains événements du Gestionnaire peuvent être prioritaires par rapport à d'autres traitements sur le BLETransmitter (par exemple, le Gestionnaire juge qu'un accident est prioritaire par rapport à un service de la route). Pour que soit transmis ces événements prioritaires, Le Gestionnaire peut arrêter tout scénario / diffusion / réception pour lancer la diffusion dudit événement grâce à son pilotage à distance. Tout ceci, dans le but de réduire l'occupation des différents canaux de diffusion, en cas de danger jugé prioritaire et imminent par le Gestionnaire, avec des informations peu importantes.

3.4.3 Sur l'OBU



FIGURE 3.6 – Diagramme d'activité du système embarqué

La figure 3.6 nous montre que sur l'application embarquée ou sur l'application *smartphone* nous ne pouvons que recevoir des informations depuis les BLE-Transmitters.

Le véhicule peut **recevoir un message** depuis le *backoffice* du Gestionnaire via les BLETransmitters ("Travaux dans 2000 m" par exemple). Dans le cas où un tel message est reçu, il est décodé et affiché sur l'interface afin d'être pris en compte par l'utilisateur. Cette vérification s'effectue pour tous les messages provenant des infrastructures.

3.5 BLE et *Advertisement*

Dans cette section, nous définissons notre utilisation de la diffusion d'informations en BLE.

3.5.1 Diffuser des informations par balises dans les C-ITS

Les projets Scoop@f [1] et C-Road [2], visant à déployer des technologies de C-ITS en France, se heurtent aux mêmes conditions de fonctionnement, à savoir travailler dans des VANETs. Ces projets se basent sur les standards ETSI des C-ITS, or la philosophie de l'architecture de communication est de **diffuser**, en clair, l'ensemble des informations afin de les communiquer à un maximum d'appareils, pour que, même des appareils hors projet puissent réceptionner des informations routières. Tout ceci dans le but de garantir et d'améliorer la sécurité routière. [67]

Il serait intéressant de suivre cette philosophie dans nos choix technologiques.

3.5.2 Le *beaconing* en BLE

Nous proposons une solution qui se rapproche de la philosophie choisie pour les technologies C-ITS déjà fixées.

En accord avec des recommandations du Bluetooth SIG [58] énoncées et avec les informations vues précédemment Section 2.5.3.6, il paraît difficile de faire des récupérations de données via une connexion.

Nous avons vu comment fonctionne le BLE et plus particulièrement l'*Advertising* dans la Section 2.5.3.7.

Afin de diffuser les informations que l'on souhaite, nous utilisons des BLETransmitters, qui s'apparentent à des balises. L'idée est qu'ils diffusent une information de manière continue tant qu'on leur demande. Par conséquent ces appareils vont simplement véhiculer la donnée à transmettre directement dans les *Advertising Packet* et plus précisément utiliser le *Protocole Data Unit* (PDU) de type ADV_NONCONN_IND [58] un format spécifique de PDU désignant le fait que *Advertising Packet* n'a pas pour but d'initier une procédure de connexion (Voir Section 2.5.3.7).

De plus, les calculs théoriques de la Section 2.5.3.6 (et expérimentaux voir section 3.7.1.3) ont montrés qu'il est possible d'émettre plusieurs messages si on se limite à 1 message = 1 DENM, on peut donc envisager de faire de la **répétition d'information**, chose extrêmement intéressante lorsqu'il s'agit de travailler avec des VANETs.

Nous adoptons donc la même philosophie que les standards C-ITS de l'ETSI et nous choisissons de transmettre l'ensemble des données via du *beaconing*.

3.6 Messages

Dans cette section, nous détaillons l'ensemble des informations relatives aux messages. Nous commençons par énoncer les contraintes s'appliquant aux technologies que nous visons, puis, nous expliquons comment les différents appareils peuvent s'identifier entre eux, et enfin, nous définissons la structure du message créé.

3.6.1 Contraintes de taille des messages

Tout d'abord, nous souhaitons que les messages soient accessibles à la plupart des puces Bluetooth et donc à la plupart des appareils embarquant du Bluetooth et compatible BLE. Nous nous focalisons sur une compatibilité avec le Bluetooth 4.x et Bluetooth 5.0.

Les messages se doivent d'être légers car même si le Bluetooth 5.0 permet une limite de 255 octets [68] pour le *payload* des paquets d'*advertisement*, le Bluetooth 4.x est limité à 32 octets [69].

Or, les DENMs actuels sont assez volumineux, comme énoncés en Section 2.3.5.2. Un des enjeux, induit par le Bluetooth 4.x, est de réduire le nombre d'octets constituant un DENM tout en garantissant de fournir une information la plus complète possible.

De plus, nous proposons que le *payload* soit également le plus léger possible afin de garantir de bonnes performances. Pour cela, nous choisissons de reprendre l'expérience acquise au travers des travaux initiaux réalisés dans le cadre du projet C-Roads [2], notamment concernant la limite du Payload.

Lors de ce dernier, 15 octets suffisaient pour effectuer l'envoi des données brutes. Nous conserverons cette taille minimale de *payload*. Dans le but de maximiser les chances de transmission de chaque paquet.

3.6.2 UUID

L'*Universally Unique Identifier* (UUID) est un identifiant unique sur 128 bits permettant de caractériser un appareil, une information ou une catégorie spécifique.

En Bluetooth cet UUID permet d'identifier un service (contenant des informations) mis à disposition par un appareil BLE. Il est créé à partir d'une base que l'on désigne sous BASE_UUID selon le standard du Bluetooth 4.0 [55].

BASE_UUID : 00000000-0000-1000-8000-00805F9B34FB

Ensuite, on peut fixer des UUID à 16 bits (*uuid16*) et 32 bits (*uuid32*) qui

peuvent être étendus à 128 bits (`uuid128`) selon le schéma suivant [70] :

$$uuid128 = uuid16 \times 2^{96} + BASE_UUID \quad (3.1)$$

$$uuid128 = uuid16 \times 2^{96} + BASE_UUID \quad (3.2)$$

il est important de noter que, dans équation 3.1, l'`uuid16` est au préalable ramené à un `uuid32` bits en ajoutant des bits à 0.

Ainsi, prenons comme exemple le `uuid16 = 0x1257` :

`uuid16 = 0x1257` devient `uuid32 = 0x00001257`

$$\begin{aligned} &0x00001257 \times 2^{96} + 0x00000000 - 0000 - 1000 - 8000 - 00805F9B34FB \\ &= 0x00001257 - 0000 - 1000 - 8000 - 00805F9B34FB \end{aligned}$$

Une fois l'UUID d'un service Bluetooth fixé et diffusé, il pourra être découvert par un appareil en utilisant le protocole *Service Discovery Protocol* (SDP).

Pour choisir quel UUID appliquer, il existe deux possibilités :

1. *Demande d'assignation* - Il est possible de demander d'**assigner un UUID** par le Bluetooth SIG afin de fixer un identifiant sur lequel l'ensemble des appareils pourront s'accorder, y compris des appareils extérieurs souhaitant s'aligner sur nos communications. C'est une solution à privilégier pour des déploiements à grande échelle. [71]
2. *L'unicité probable* - Il s'agit de **générer un UUID aléatoirement**. Il est très peu probable d'obtenir deux fois, ou plus, le même UUID lors de l'assignation aléatoire. D'autant qu'il faudrait que les deux UUIDs redondants soient à portée l'un de l'autre pour que cela soit réhibitoire.

Pour nos travaux, nous utilisons l'option de *l'unicité probable*, ne pouvant nous permettre de déclarer un UUID auprès de la Bluetooth SIG. Cependant, c'est une option qui ne doit pas être choisie en cas de déploiement, au risque d'être en conflit avec d'autres appareils. Par ailleurs, elle n'offre pas non plus un aspect collaboratif pour la solution, puisqu'elle ne met pas à disposition l'identifiant au public. Et donc, elle ne permet pas à des applications extérieures de s'intégrer à la solution.

Ainsi, nous choisissons de réaliser nos tests avec l'UUID suivant :

`UUID_DENM : 0x00009999-0000-1000-8000-00805F9B34FB`

3.6.3 Datagramme

La structure du *Decentralized Environmental Notification Message* (DENM) est décrite par l'European Telecommunication Standard Institute (ETSI) et plus précisément dans le standard ETSI [72]. L'objectif de ce type de message est,

comme dit précédemment, d'annoncer et de définir les caractéristiques d'un évènement sur la route.

Nous proposons ici un autre apport de notre contribution qui fixe une version légère de ce message afin de garder une taille de message réduite conformément aux contraintes de taille en BLE (voir 3.6.1).

Rappelons-nous que ces messages sont transmis depuis le BackOffice ITS aux usagers par le biais des BLETransmitters.

0	1	2	3	4	5	6	7	8	9	10
BC	DC	D	Dist	T						

Dans ce datagramme, nous avons :

- *Beacon Code (BC)* (2 octets) - L'identifiant de la balise émettrice.
- *DENM Code (DC)* (1 octet) - Code de l'évènement sur la route, obéissant à un mapping normé, CF. annexe A.
- *Direction (D)* (1 octet) - Angle formé entre le Nord et la direction de l'évènement. Cette donnée permet de déterminer si un véhicule est concerné par un message ou non (voir section 4). 1 unité = 2 degrés.
- *Distance (Dist)* (1 octet) - Distance entre la balise émettrice du message et l'évènement. 1 unité = 100 m.
- *Timestamp (T)* (6 octets) - Horodatage du message.

Ce message regroupe l'ensemble des informations permettant d'identifier un DENM sur la voie tout en étant le plus léger possible.

3.7 Tests sur route

Afin de tester la pertinence et la fiabilité de cette solution, nous menons plusieurs études sur les routes. Cette phase de tests est composée de 3 jalons. Tout d'abord, nous testons l'utilisation du BLE sur la route par un scénario simple. Ensuite, faisons de même pour la réception de messages dans un scénario correspondant à la solution proposée avec des appareils classiques et des équipements liés aux projets ITS. Il est à noter que nous menons ces tests avec différents équipements ; ils visent à prouver la faisabilité de la solution. Dans un premier temps, des puces Bluetooth de classe 2, peu coûteuses et très répandues, sont utilisées. Dans un second temps, nous effectuons une phase de test avec des équipements ITS, ces derniers intègrent directement des puces de classe 1, plus performantes mais moins accessibles. C'est la raison pour laquelle nous intégrons différentes classes de puces au cours de ces travaux.

3.7.1 Expérimentations sur la route

L'ensemble des tests a été mené avec les paramètres suivants :

- Taille du *payload* : 15 octets (relatif aux contraintes présentées en Section 3.6.1).

- ADV_INT : 100ms.
- SCAN_INT : 100ms.

3.7.1.1 Phase 1 - mise à l'épreuve de la technologie

Dans un premier temps, il est nécessaire de savoir s'il est possible de véhiculer des informations en BLE sur autoroute.

Nous mettons donc en place le scénario de test suivant :

- *Émetteur BLE* : Smartphone Samsung Galaxy S7 en Android 8.0.0 embarquant du Bluetooth 4.2 (Puce BLE de classe 2).
- *Récepteurs BLE* : 2 x Tablettes Samsung Galaxy Tab A en Android 5.1.1 embarquant du Bluetooth 4.2 (Puce BLE de classe 2) - Une en charge et une autre autonome.
- *Localisation* : Autoroute A4.
- *Vitesse* : 130km/h.

Nous mettons en place une application ad-hoc sur l'émetteur BLE que l'on designera comme BLETransmitter et, en réception, une application également ad-hoc qui reçoit ces messages et les affiche.

	Tablette en charge	Tablette autonome
Taissy vers Champfleury	Messages reçus	Messages perdus
Champfleury vers Taissy	Messages reçus	Messages perdus

TABLE 3.1 – Résultats tests Phase 1

Nous obtenons le Tableau 3.1, il illustre les passages réalisés lors des premiers tests pour étudier le réalisme de l'utilisation du protocole BLE. Nous en déduisons que cette technologie est utilisable dans un milieu mobile.

3.7.1.2 Phase 2 - Scénario de production simple

Afin d'évaluer ce que nous pouvons attendre comme performances, nous mettons en situation l'architecture avec des équipements simples.

Nous avons positionné un Raspberry sur un pont (faisant office de BLETransmitter) et plus précisément sur un de ses côtés puis, nous circulons dessous afin de récupérer les messages. À noter que le pont agit comme un écran puisque le Raspberry est positionné sur un de ses côtés, nous avons donc une communication unidirectionnelle.

Les caractéristiques du test sont les suivantes :

- *Émetteur BLE* : Raspberry sous Raspbian embarquant du Bluetooth 4.2 (Puce BLE de classe 2). Diffusion fixée à 5 Messages/sec.
- *Récepteurs BLE* : Tablette Samsung Galaxy Tab A en Android 5.1.1 embarquant du Bluetooth 4.2 (Puce BLE de classe 2).

- *Localisation* : Autoroute A4.
- *Vitesse* : 130km/h.

Le Raspberry utilise une bibliothèque nommée "Bluez" pour piloter la puce BLE. Nous envoyons 15 octets de données. Le tout est logué à la réception.

Passage	NB messages enregistrés
1	2 messages
2	3 messages
3	3 messages
4	2 messages
5	2 messages
6	3 messages
7	2 messages
8	3 messages

TABLE 3.2 – Concaténation des résultats de réception sur tablette lors de la Phase 2

Le Tableau 3.2 montre la concaténation du nombre de messages reçus lors de cette phase ; ce sont des résultats prévisibles. En effet, des puces de classe 2 ont une dizaine de mètres de portée (variable en fonction de la puissance précise délivrée lors de l'envoi, le paramétrage et la topographie du lieu).

Malgré le fait que nous puissions réceptionner des messages à 130km/h, ces tests ne sont pas tout à fait satisfaisants car certes, la transmission de DENM est possible, mais le nombre de messages reçus ne permet pas de satisfaire une redondance de données.

Les résultats seraient différents avec des équipements plus adaptés notamment avec des puces BLE de classe 1. Nous avons donc testé la solution avec des RSUs déjà équipées de ce type de puces et positionnés sur la route.

3.7.1.3 Phase 3 - Tests avec des équipements ITS

Les RSUs disposent de puces BLE de classe 1, beaucoup plus puissantes que celles des Raspberrys. Leur portée, en théorie est de l'ordre de la centaine de mètres ce qui semble bien plus adaptée pour de la communication véhiculaire.

Nous utilisons toujours ici des messages non signés de 15 octets dans le but de récupérer le plus grand nombre de messages recevables dans cette situation.

La figure 3.7 montre la position des RSUs placées sur la route et équipés d'une option de diffusion de BLE. La RSU rouge désigne la RSU de Saint-Remi et la bleue, la RSU de Bezannes.

À noter que nous déplaçons l'environnement de test étant donné que les RSUs sont installées sur les routes physiquement, et qu'il n'est donc pas possible de les déplacer.



FIGURE 3.7 – RSUs autours de Reims.

Le logiciel interne de la RSU est la propriété du fabricant, nous avons dialogué avec ce dernier afin de s'accorder sur la structure du message à envoyer. Elles ont donc été mises à jour pour supporter ces messages.

Les caractéristiques du test sont les suivantes :

- *Émetteur BLE* : RSU embarquant du Bluetooth (Puce BLE de classe 1). Diffusion fixée à 5 Messages/sec.
- *Récepteurs BLE* : Tablette Samsung Galaxy Tab A en Android 5.1.1 embarquant du Bluetooth 4.2 (Puce BLE de classe 2).
- *Localisation RSU A* : Autoroute A4 - PMV Bezanne - Long : 49.216585 / Lat : 3.984819
- *Vitesse au niveau de A* : 130km/h.
- *Localisation RSU B* : Autoroute A344 - PMV Saint-Remi - Long : 49.2411604 / Lat : 4.032578
- *Vitesse au niveau de B* : 90km/h.
- *Advertising Interval* : 100 ms.

Les RSUs diffusent en permanence les messages, il nous est donc possible d'effectuer plusieurs passages en voiture afin de tenter des récupérations d'évènements toujours via notre application ad-hoc embarquant une cartographie et un encart affichant le message BLE traduit à sa réception.

	Vitesse	Intervalle de réception	Nb Messages Relevés	Portée estimée
Moyenne des passages	90	12,75	39,25	292
Moyenne des passages	130	10,5	28,5	328,9

TABLE 3.3 – Récapitulatif des tests effectués

Le Tableau 3.3 montre les résultats obtenus lors des tests sur route avec les RSUs et les tablettes. Ces tests montrent que nous sommes dans la capacité d'effectuer une forte redondance de messages et donc maximiser les chances d'effectuer une transmission avec un véhicule circulant sur la route.

En ce qui concerne la portée, il faut être critique sur ces valeurs car le GPS des tablettes se met à jour de manière périodique mais avec des intervalles assez longs.

Malgré cela, ces résultats sont intéressants car ils démontrent la viabilité de la solution avec de tels équipements.

3.7.2 Analyse des expérimentations

Ces expérimentations ont prouvé que la diffusion d'évènements dans un environnement autoroutier est possible malgré les contraintes de mobilité qui lui sont associées.

L'ensemble des tests menés en Phase 3 (voir section 3.7.1.3) sont des tests correspondant le plus aux situations réelles, c'est-à-dire, transmettre des informations, depuis les infrastructures, aux différents véhicules.

Nous observons que les résultats de portées obtenus sont revus à la hausse par rapport aux résultats théoriques présentés en Section 2.5.3.6 concernant le calcul du nombre de messages, aux passages des dispositifs d'envoi.

En effet, en moyenne, les RSUs offrent une meilleure portée que les 100m théoriques d'une puce de classe 1. Le temps mesuré à portée de la RSU est donc plus important. Cela s'explique par deux éléments. Tout d'abord, la topographie du lieu facilite la propagation du signal. En effet, pour les deux RSUs, elles sont dans un environnement autoroutier donc une route bien rectiligne et avec assez peu d'obstacles. De plus, les RSUs étant placées en hauteur, sur des Panneaux à Message Variable (PMV), cela facilite d'autant plus la propagation de leurs signaux.

En revanche, en moyenne, nous captions autant de messages malgré les performances accrues, cela s'explique par le paramétrage de la puce réceptrice et plus précisément du *Scan Windows* et *Scan Interval* (voir section 2.5.3.5). À cause de ces deux éléments, certains messages sont ratés par les appareils récepteurs.

Il est possible d'approximer les messages perdus à cause du paramétrage et des éléments extérieurs, type interférences, météo ou trafic routier dense. En théorie, nous ne recevons qu'entre 10% et 50% des messages. Cela peut être également affecté par notre matériel de réception (les tablettes) qui ont un *Hardware* un peu vieillissant.

Enfin, les différentes phases de tests, de l'architecture que nous avons proposée, montrent qu'envisager l'utilisation du BLE pour la transmission d'informations

C-ITS n'est pas un non sens et permet de remplir les conditions nécessaires fixées par les contraintes énoncées en Section 3.6.1.

3.8 Discussion

Dans ce chapitre nous avons vu les aspects suivants :

1. L'architecture BLE globale. Cette partie nous a permis de proposer une solution de transmission d'informations ITS en BLE et d'explicitier les différents acteurs de la solution.
2. Un scénario d'illustration permettant d'exposer le cas d'usage générique de la solution et comment elle peut y répondre.
3. Les flux d'informations mis en place qui caractérisent la transmission en I2V de la solution.
4. Les comportements des différents acteurs composant l'architecture. Leurs actions et réactions possibles.
5. Le fonctionnement du BLE au sein de notre architecture et les raisons qui nous ont poussé à adapter cette technologie à nos besoins.
6. Une structure de message réduite permettant de fournir une information, concernant un évènement sur la route, la plus complète possible.
7. Les expérimentations que nous avons pu mener sur la route afin de mettre à l'épreuve la solution et confirmer sa faisabilité et son réalisme.

Cette solution permet de diffuser l'information en clair et à l'ensemble des véhicules en écoute sur le BLE, et est conforme à la philosophie des C-ITS. Cependant, la solution proposée a pour but d'améliorer la sécurité routière. Dans un tel environnement il est alors nécessaire de capter l'attention de l'utilisateur, uniquement si la situation l'exige. Il faut alors trouver un moyen d'affiner les cibles de nos messages. Nous devons mettre en place une solution permettant d'identifier la voie sur laquelle l'évènement se déroule. Nous traitons cet aspect dans le chapitre suivant.

Dissémination dans un cadre BLE

Dans ce chapitre, nous mettons en place une solution permettant aux messages de cibler précisément les voies de circulation concernées et donc les usagers concernés. Le but est d'attirer l'attention de l'utilisateur, si les informations transmises justifient d'être notifiées, une fois ces dernières réceptionnées.

En plus de l'architecture, présentée dans la Section 3, nous proposons une fonctionnalité permettant à un message de cibler une partie précise de la route, et ce, tout en gardant cet aspect de *beaconing* qui a été présenté dans la Section précédente (Voir Section 3).

Avec les contraintes de messages énoncées antérieurement, cette fonctionnalité doit définir une manière de cibler une voie avec le moins d'informations à transmettre.

De surcroît, nous souhaitons que les OBUs aient le moins de charge de travail possible, c'est pourquoi, notre solution doit demander le moins de ressources possibles à l'appareil.

Dans cette section, nous présentons l'étude du ciblage de la voie qui nous permet de résoudre ce problème.

4.1 Introduction

4.1.1 Situation actuelle

En utilisant l'architecture présentée dans la Section 3, nous sommes en mesure de définir l'environnement suivant.

La figure 4.1 montre une route sur laquelle nos équipements sont déployés. Supposons qu'un évènement, tel un accident, vienne d'avoir lieu et que le Gestionnaire ait validé l'évènement en question. Le BLETransmitter reçoit un ordre d'émission du *Road Operator* (Gestionnaire) afin de prévenir les usagers.

Puisque le BLETransmitter diffuse son information telle une balise, les deux

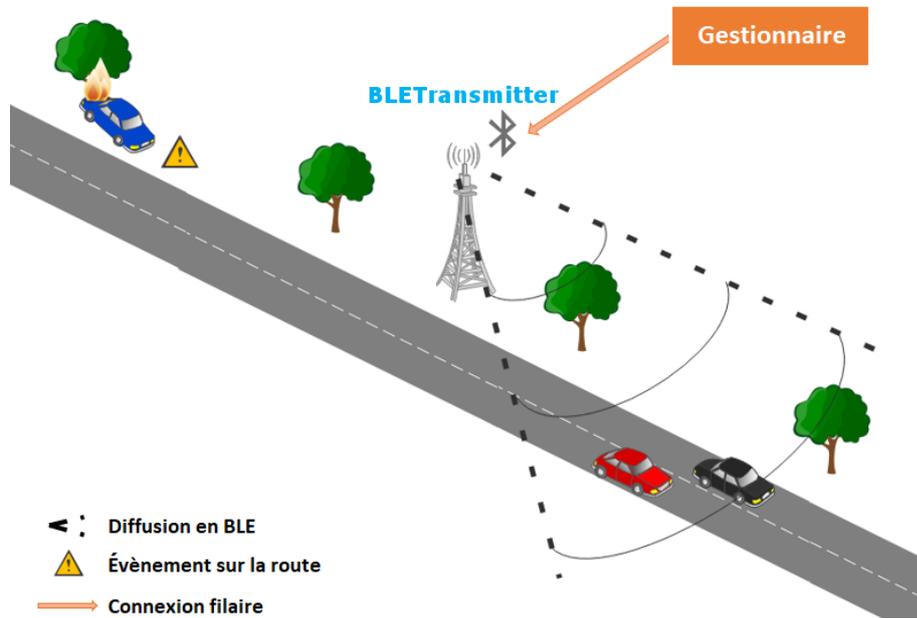


FIGURE 4.1 – BLETransmitter sur la route

voies de circulation vont être couvertes par la diffusion de l'information. Ainsi les usagers circulant sur les deux voies la recevront et seront notifiés.

4.1.2 Problématique de ciblage et de limite de taille

Comme le présentait la figure 4.1, un évènement comme un accident situé en amont du véhicule noir a eu lieu et a été déclaré et validé. Cet accident se positionne à un endroit précis et ne concerne qu'une partie de la route et donc une voie particulière. Le véhicule rouge est passé à côté précédemment mais n'a pas été concerné car le véhicule accidenté n'était pas sur sa voie.

La situation est donc la suivante :

- l'OBU du véhicule noir doit recevoir l'information concernant l'accident et notifier l'utilisateur car il circule sur la voie concernée par l'évènement.
- l'OBU du véhicule rouge n'a pas besoin de notifier l'information étant donné que l'utilisateur circule sur la voie opposée à l'évènement.

Nous déterminons donc qu'il est nécessaire de trouver une solution pour cibler une voie précise avec un évènement transmis en BLE.

De plus, conformément aux contraintes énoncées dans la Section 3.6.1, par lesquelles les messages BLE doivent être les plus légers possibles, nous optons pour une solution qui est également la plus économique en terme de taille de paquet transmis.

4.1.3 Identification des critères de recherche

Afin de garder la philosophie du *beaconing*, nous excluons l'idée de faire du service à la personne, pour identifier un véhicule précisément. Nous souhaitons que la fonctionnalité permette de diffuser l'information à tout les usagers de la route présents dans la zone.

Une fois un évènement repéré et validé, le Gestionnaire va être en mesure de déterminer quelle voie est concernée. Cela fait partie du processus d'identification des BLETransmitters vu en Section 3.4.1. Le Gestionnaire est donc capable d'établir si un évènement concerne une voie de circulation précise ou les deux et active les balises en conséquence.

Nous devons solutionner le problème en trouvant un moyen de cibler une voie dans nos messages BLE. Cela nécessite d'imaginer une désignation qui sera économe en terme de taille sur les messages et également, récupérable par les véhicules. En effet, l'OBU devra déterminer sur quelle voie elle circule et donc s'il y a corrélation entre sa voie et celle du message.

4.1.4 Liste des solutions étudiées

Nous dégageons et expérimentons différentes solutions qui répondent à ce problème : désigner une voie précise dans un DENM et déterminer la voie de circulation du véhicule afin de rapprocher ces deux informations.

Cette question a été étudiée lors de la Phase de test numéro 2 (voir Section 3.7.1.2).

Nous envisageons les solutions suivantes :

- *Mesure du Received Signal Strength Indication (RSSI)* - Le relevé des puissances du signal des différents messages en BLE, ou RSSI, peuvent remplir ce rôle. En effet, une fois les différentes valeurs enregistrées au passage d'un BLETransmitter, une tendance peut être déterminée afin de fixer le sens du déplacement du véhicule. En théorie, le véhicule est donc en capacité de déterminer la voie sur laquelle il circule. Le Gestionnaire peut également déterminer les relevés caractéristiques de la voie afin de désigner celle qui est concernée par l'évènement. C'est une solution qui demande des infrastructures particulières vues en Section 4.2.
- *Relevé de séquences de messages* - L'identification de chaque message BLE par un numéro de séquence peut permettre au Gestionnaire de déterminer des séquences d'identifiants caractérisant les voies. Deux éléments sont nécessaires : des BLETransmitters avec un signal unidirectionnel et un Gestionnaire qui définit des séquences de messages pour chaque voie. Une valeur au sein du message permettra au receveur de déterminer si le message concerne une séquence précise ou une autre. C'est donc, également, une solution qui demande des infrastructures particulières vues en Section 4.3.
- *Calcul de la direction* - C'est une solution utilisant l'orientation de l'évè-

nement. Le Gestionnaire ayant les informations de l'accident, ainsi que la cartographie, peut déterminer les voies concernées par l'évènement au niveau de ses balises en orientant ces différents éléments. Il peut donc transmettre, dans le message, l'orientation concernée par le DENM. Le véhicule est également en capacité de déterminer sa direction et donc déterminer s'il circule sur la voie en question ou non. Cette solution est détaillée en Section 4.4.

Nous allons donc dans les sections suivantes expliciter les solutions proposées, leurs faisabilités afin de les valider.

4.2 Mesure du *Received Signal Strength Indication* (RSSI)

Nous utilisons, avec cette solution, la mesure de puissance de signal du message afin de désigner une voie cible. Cette dernière est identifiée par une variation de cette puissance. Nous envoyons cette information au receveur qui sera, ainsi, en mesure de déterminer la désignation de sa voie et en déduire s'il est ciblé par le message.

4.2.1 Le *Received Signal Strength Indication* - RSSI

Il s'agit d'un indicateur définissant la puissance d'un signal reçu. Elle indique la qualité de la réception d'une information émise depuis un émetteur.

Mesurée en dBm (ou dBmW) - décibel-milliwatt, elle caractérise une puissance de signal de 0 dBm à -120 dBm.

A chaque paquet reçu, le capteur de l'appareil récepteur va évaluer la valeur du RSSI et la fournir.

Le RSSI est en corrélation avec la distance entre l'émetteur et le récepteur d'un signal. Certes, une marge d'erreur est à prévoir due à l'évaluation peu précise que procure cette méthode sur la distance, il n'en est pas moins possible de déterminer une tendance en lissant les valeurs relevées du RSSI [73].

Par conséquent, nous pensons qu'il est possible de rapprocher les relevés des différents RSSIs présents dans les messages BLE afin de déterminer si un véhicule se déplace vers une balise ou s'il s'en éloigne.

4.2.2 Présentation

Ce type de solution nous demande de déployer une installation particulière sur les routes. En effet, les antennes Bluetooth qui vont être utilisées doivent être unidirectionnelles.

Le Gestionnaire, ayant installé ses balises, connaît leurs orientations par rapport à la route. Et donc, à la déclaration d'un évènement, il peut définir au sein du message une valeur annonçant que le message concerne un signal en diminution ou un signal en augmentation.

C'est ainsi que nous proposons de distinguer les voies, via la variation du signal de la balise unidirectionnelle. Ainsi, un évènement peut cibler une voie caractérisée par un signal en diminution ou par un signal en augmentation.

Un véhicule, au passage du BLETransmitter unidirectionnel, captera un ensemble de messages avec cette caractérisation. L'OBU sera donc en mesure de consulter le relevé des différents messages reçus et surtout leurs RSSIs. Elle pourra ainsi déterminer la tendance qui se dégage de ces relevés de puissance de signal.

C'est pour cette raison que nous avons besoin d'une antenne unidirectionnelle, pour que l'OBU puisse relever une tendance à la hausse ou à la baisse à l'approche du BLETransmitter.

Cette tendance exposera une diminution ou une augmentation du signal, qui devra être rapprochée de la donnée disponible dans le message de l'évènement définissant quelle voie est ciblée.

Ainsi, l'OBU sera capable de déterminer si elle est ciblée par ce message et donc si elle doit le consumer ou non.

C'est une solution qui répond bien à nos contraintes de taille de paquet puisqu'elle ne demande, en soit, qu'un bit pour déterminer s'il s'agit d'un signal en augmentation ou en diminution qui doit être observé. Le reste de l'information est fourni par l'appareil récepteur qui va évaluer la puissance du signal.

Valeurs du bit de désignation :

- *Signal en diminution* : 0
- *Signal en augmentation* : 1

4.2.3 Illustration

Nous présentons le cas illustré en figure 4.2. **Le Gestionnaire souhaite annoncer un évènement sur la voie B**, avec ce BLETransmitter, il devra annoncer la valeur équivalente à B, c'est-à-dire un signal en **diminution**. Donc le bit de désignation sera à 0.

Le véhicule rouge, en passant aux abords de la balise, capte un ensemble de messages.

L'OBU relève une **diminution** de la force du signal au passage de la balise. Dans ce cas, le véhicule circule manifestement sur la voie B et donc sera concerné par ce message.

Nous présentons le cas illustré en figure 4.3. **Le Gestionnaire souhaite an-**

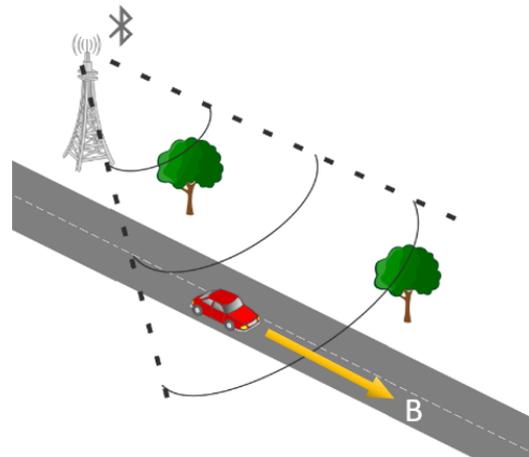


FIGURE 4.2 – Cas d'un signal en diminution à l'approche d'un BLETransmitter

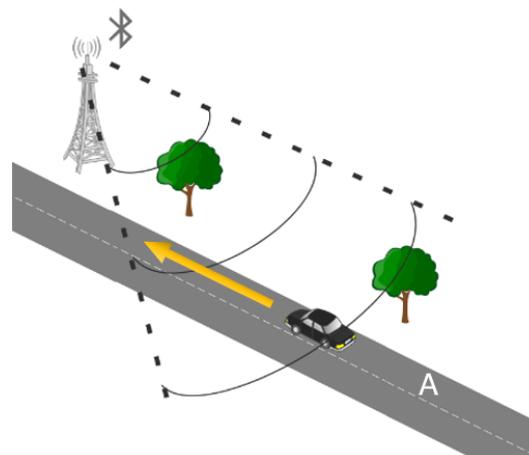


FIGURE 4.3 – Cas d'un signal en augmentation à l'approche d'un BLETransmitter

noncer un évènement sur la voie A, avec ce BLETransmitter, il devra annoncer la valeur équivalente à A, c'est-à-dire un signal en **augmentation**. Donc le bit de désignation sera à 1.

Le véhicule noir, en passant aux abords de la balise, capte un ensemble de messages.

L'OBU relève une **augmentation** de la force du signal au passage de la balise. Dans ce cas, le véhicule roule manifestement sur la voie A et donc sera concerné par ce message.

Dans le cas illustré, **si un message déclaré cible la voie A** (Avec un bit de désignation à 1) :

- Le véhicule noir, étant sur la **voie A**, constatera une **augmentation** du signal à l'approche du BLETransmitter. Il constatera également que le message contient la valeur équivalente à un signal en **augmentation**. Il déterminera qu'il **est concerné** par le message et donc le traitera.
- Le véhicule rouge, étant sur la **voie B**, constatera une **diminution** du signal à l'approche du BLETransmitter. Il constatera également que le message contient la valeur équivalente à un signal en **augmentation**. Il déterminera qu'il **n'est pas concerné** par le message et donc l'**ignorer**a.

4.2.4 Expérimentations

Pour évaluer la faisabilité de cette solution, lors de nos tests sur route, nous avons relevé les puissances de signal des différents messages. On se base donc sur les données relevées dans la Section 3.7.1.

Les tests ont été effectués avec deux types d'équipements en émission : des Raspberrys et des RSUs, respectivement équipés de puces de classe 2 et 1. Il est donc être possible d'observer si une tendance peut se dégager des puissances relevées avec ces deux types d'appareils.

Nous rappelons les conditions de tests :

- *Taille de message* : 15 octets
- *Fréquence d'envoi* : 10 msg/sec

4.2.4.1 Tests avec Raspberry

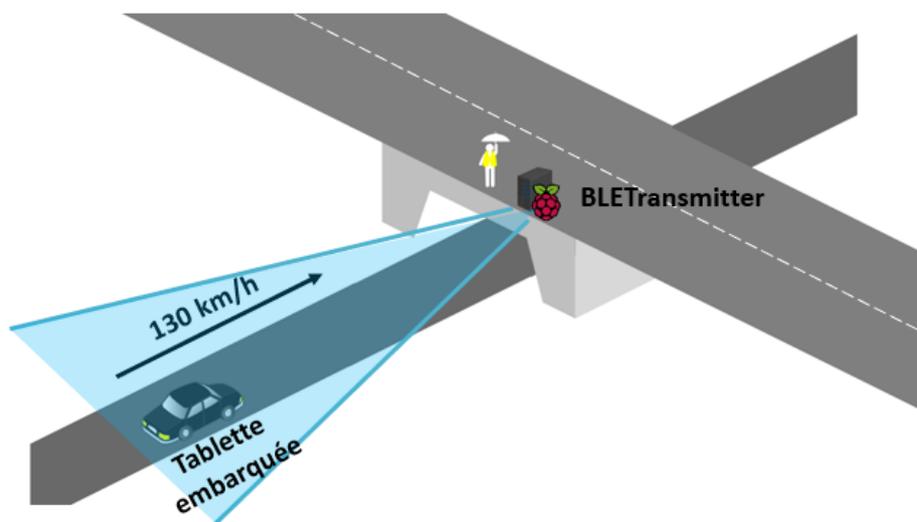


FIGURE 4.4 – Architecture de test

Les tests sont menés sur autoroute à 130km/h. Comme le présente la figure

4.4, l'émetteur est un Raspberry avec une antenne omnidirectionnelle. Afin, de réaliser des tests avec un signal unidirectionnelle, le Raspberry est donc posé sur l'un des côtés d'un pont. Naturellement, la structure va agir comme un écran et le signal va uniquement se propager dans un seul sens. Le rôle de l'OBU sera assuré par des Samsung Galaxy Tab A avec une application adhoc, 4 récepteurs sont embarqués dans le véhicule de test.

Par conséquent, lors de cette phase de tests, nous recherchons la formation d'une tendance croissante ou décroissante des relevés de RSSI caractérisant l'approche ou l'éloignement du véhicule par rapport à une balise.

	Tablette 1	Tablette 2	Tablette 3	Tablette 4
1er Passage	-43 -38	-38 -38	-45 -46 -38	-38 -37
2nd Passage	-45 -45	-38 -38	-42 -43	-38 -32 -33

TABLE 4.1 – Relevés des RSSIs en dBm lors des tests avec le Raspberry

Le tableau 4.1 montre les résultats obtenus lors de cette phase. Selon le positionnement du Raspberry lors des tests, les résultats attendus sont :

- *1er Passage* : Tendance croissante.
- *2nd Passage* : Tendance décroissante.

Il apparaît que 50% de nos relevés sont en corrélation avec les résultats attendus (en vert sur le tableau) mais que, dans certains cas, les relevés ne permettent pas de déterminer une tendance précise - ils sont donc traités comme des résultats non exploitables (en Orange). Enfin, sur un appareil, il a été relevé un faux positif puisque la tablette a détecté une croissance du signal là où une tendance décroissante aurait du être observée.

Nous expliquons ces résultats par une réception de messages trop peu nombreux, il est impossible de distinguer une tendance significative d'une simple fluctuation de signal due aux éléments externes (interférences, topographie, météo, condition de trafic, ...).

L'utilisation d'antenne plus puissante pourrait résoudre ce problème.

4.2.4.2 Tests avec RSU

Ici, l'émetteur BLE est omnidirectionnel puisque des RSUs sont utilisées comme BLETransmitters. De ce fait, la tendance devrait être sous forme de Gaussienne représentant dans un premier temps l'approche de la balise avec une augmentation du signal entrant puis à l'éloignement de celle-ci, une diminution de ce même signal. En guise de récepteur, des tablettes Samsung Galaxy Tab A sont utilisées avec une application ad-hoc.

L'utilisation de RSU permet de supposer que les résultats attendus peuvent être exploités du fait de la portée accrue de l'antenne et donc de la réception de plus de messages sur l'OBU.

De même, les tests ne peuvent pas être menés avec des antennes unidirectionnelles sur les RSUs, ces équipements appartenant au gestionnaire. Il n'est donc pas envisageable, lors de ces travaux, de changer le matériel des équipements déployés sur les routes.

Nos tests sont menés à 90 km/h et 130 km/h avec les RSUs définies en Section 3.7.1.

Pour identifier la tendance sur les relevés, nous utilisons la méthode des moyennes mobiles :

$$\bar{x}_n = \frac{1}{N} \sum_{N-1}^{k=0} x_{n-k} \quad (4.1)$$

L'équation 4.1 présente le calcul de la moyenne mobile \bar{x}_n sur un ensemble de données n et une fenêtre de données N .

Nous allons présenter différents relevés obtenus en pointant le passage de la RSU, ainsi, nous pourrions constater si nous obtenons bien l'augmentation puis la diminution du signal attendue.

Les Figures 4.5 a et b montrent deux passages **à proximités de la RSU Saint-Remi, à 90 km/h**, avec respectivement 51 et 29 messages reçus. Elles illustrent les variations de puissances de signal relevés à proximité de la RSU.

Sur ces figures, nous établissons les tendances de ces différents passages en utilisant l'équation 4.1. Nous traçons une première tendance avec des fenêtres de 4 données ; celle-ci traduit une tendance très localisée dans le temps. Nous traçons également une seconde tendance avec des fenêtres de 14 données ; nous obtenons ainsi une tendance plus globale.

Nous avons donc deux tendances obtenues avec l'équation 4.1 avec des valeurs de N fixés à 4 et 14.

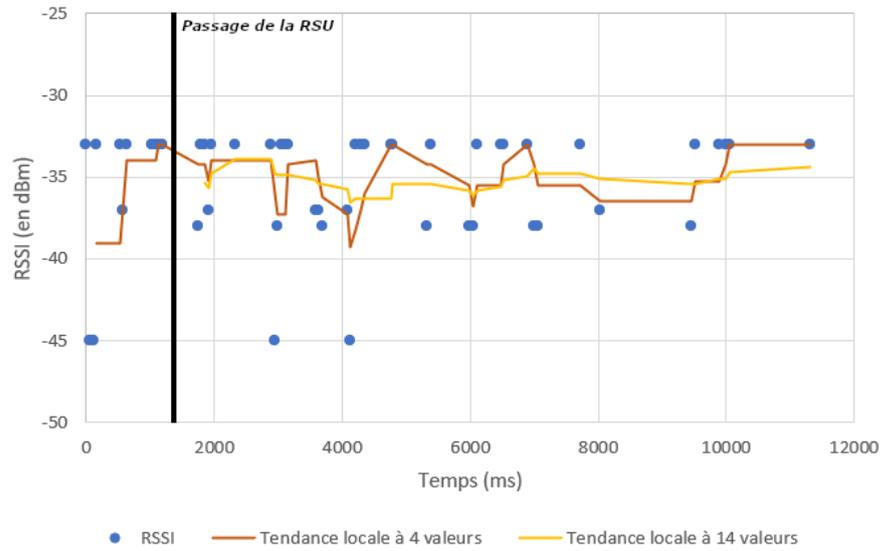
Nous sommes dans l'impossibilité d'en déduire une tendance exploitable, à savoir une Gaussienne définissant le rapprochement et l'éloignement du véhicule par rapport à la RSU.

D'autres tests ont été menés **sur la RSU Bezannes**. Sur cette portion, les tests peuvent être menés à 130km/h.

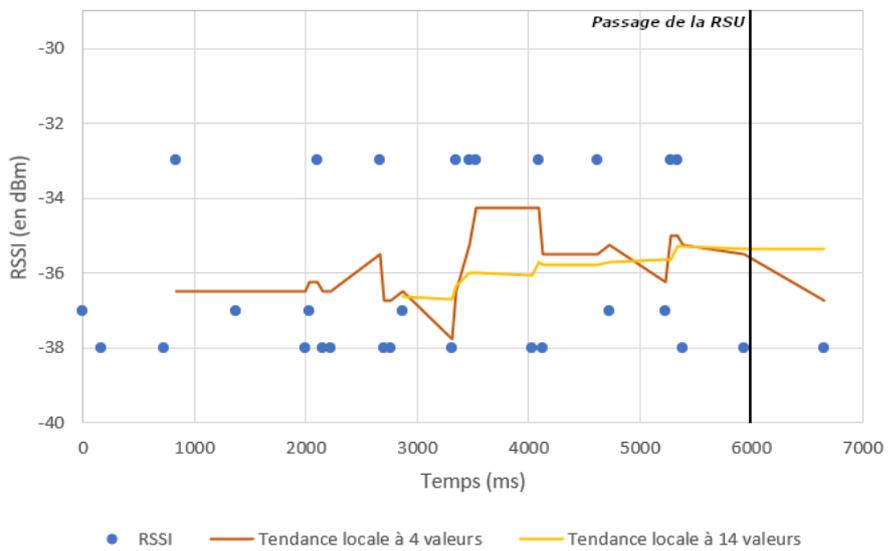
Les figures 4.6 a et b présentent les variations de puissances de signal relevés à proximité de la RSU.

Comme dans les Figures 4.5 a et b, nous traçons les deux tendances avec une fenêtre de données N de 4 et 14 afin de déterminer nos tendances locales et globales.

Les relevés montrent le même problème : les valeurs de RSSI ne permettent pas

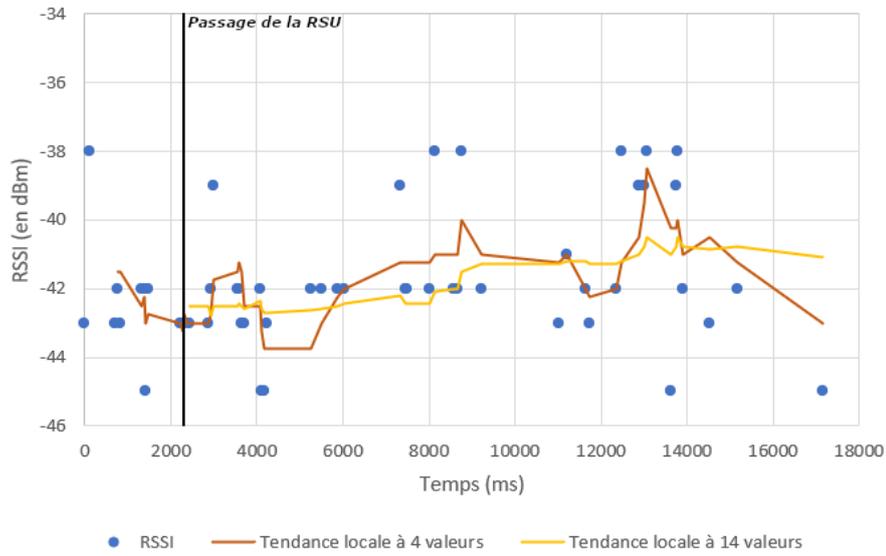


(a) 11/10/19

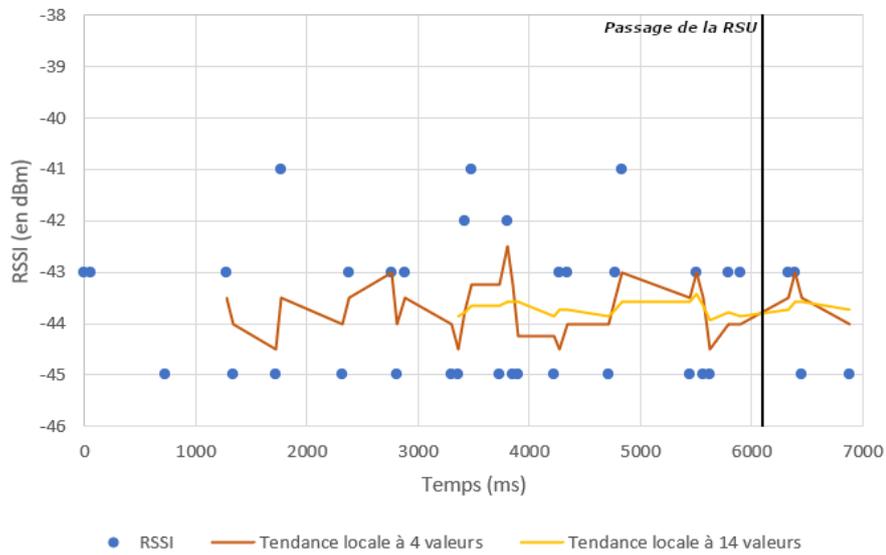


(b) 15/10/19

FIGURE 4.5 – Mesures de RSSI au passage de la RSU Saint-Remi



(a) 1er passage



(b) 2nd passage

FIGURE 4.6 – Mesures de RSSI au passage de la RSU Bezannes

de dégager de tendances exploitables afin de déterminer la phase d'approche et d'éloignement du véhicule par rapport à la balise.

De manière très **exceptionnelle**, cette analyse a pu être concluante lors de nos tests, mais, encore une fois, il s'agit d'un cas **rarissime** et donc non généralisable.

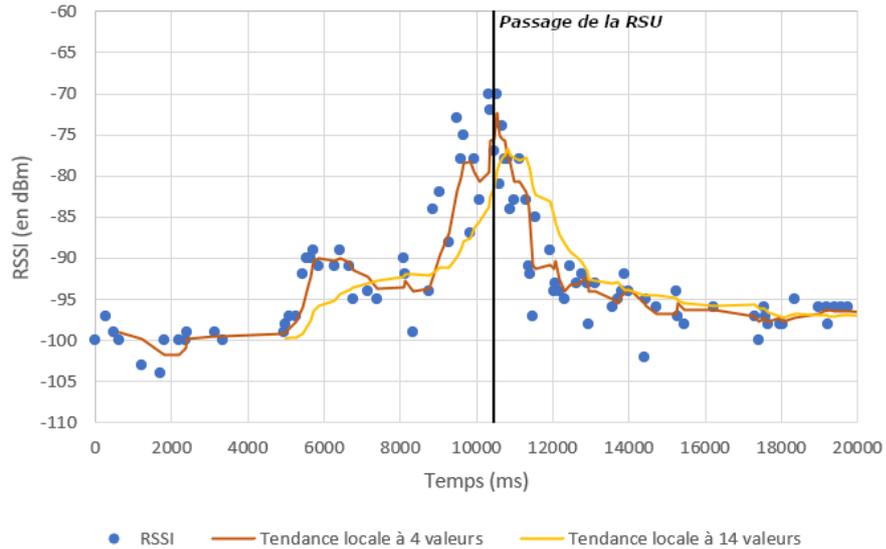


FIGURE 4.7 – Relevé des RSSI du passage du 10/10/19

La figure 4.7 présente les relevés obtenus à proximité de la RSU et les tendances. Ces dernières sont toujours déterminées via l'équation 4.1 et avec des fenêtres de données N à 4 et 14 valeurs afin d'avoir une tendance locale et une plus globale.

Nous constatons que la tendance locale traduit des changements plutôt erratiques localement, mais une tendance globale se dessine, une gaussienne en l'occurrence.

Dans le cadre d'une fenêtre de données petite, avec $N = 4$, des fluctuations peuvent fausser l'analyse car elles montrent une diminution de la force du signal là où l'OBUS se rapproche de la RSU.

Mais avec une fenêtre de données à $N = 14$, nous pouvons déterminer à quel moment le véhicule s'est approché de la RSU et à quel moment il s'en est éloigné (avec un léger décalage dû à la méthode de calcul).

L'ensemble de ces résultats montrent donc que, malgré des cas, très exceptionnels, exploitables, d'une manière générale, même avec un jeu de données conséquent, il semblerait que nous ne soyons pas en mesure de déterminer les tendances nécessaires pour cibler les véhicules correctement.

À noter que cette analyse étant assez erratique localement, elle demande une collecte d'informations conséquentes pour en déterminer un résultat exploitable.

4.2.5 Évaluation des résultats

Les résultats que nous avons présentés précédemment, Section 4.2.4, montrent que :

- Dans le cas d'un Raspberry, le nombre de messages n'est pas suffisant pour déterminer si une tendance peut se dégager
- Dans le cas d'une RSU, le nombre de messages est plus important mais on observe qu'une tendance ne se dégage pas assez pour déterminer si le véhicule se déplace dans un sens ou un autre et donc si un évènement le cible ou non.

Nous constatons donc que cette solution ne donne visiblement pas de résultats exploitables car la tendance ne se dessine pas correctement dans tous les cas. De plus, elle demande un nombre suffisant de messages afin de dégager une tendance ; cela nécessite, non seulement, de réaliser des études sur le nombre approprié de messages pour considérer le message comme valide, mais aussi du temps de traitement supplémentaire.

4.2.6 Analyse

Au vu des résultats des expérimentations, cette solution n'apparaît pas viable pour son utilisation au sein des C-ITS. La solution demandant un grand nombre de messages et des valeurs moins erratiques.

Si cette solution semble assez simple à déployer au niveau des messages. En effet, une information passive est désignée pour déterminer le sens de déplacement d'un véhicule, et que la seule information ajoutée au sein du message est la tendance concernée ; en revanche, il est nécessaire de déployer des balises directionnelles qui sont difficiles à mettre en place pour le gestionnaire, du fait d'un coût d'installation trop élevé.

Au niveau de l'autonomie, le Gestionnaire doit juste identifier les BLETransmitters concernés et quelles sont leurs orientations, il n'aura ensuite qu'à envoyer les tendances requises dans le message.

Cette solution est certes autonome et simple au niveau des messages mais les résultats expérimentaux ne sont pas satisfaisants et elle est difficile à déployer physiquement. **Nous avons donc décidé de ne pas poursuivre vers cette option.**

4.3 Relevé de séquence de messages

Nous essayons d'utiliser des séquences de messages dans le but de définir la voie ciblée par un message émis depuis les BLETransmitters.

4.3.1 Présentation

Cette solution demande également de déployer des balises unidirectionnelles. De plus, une RSU devra se composer de deux BLETransmitters au lieu d'un seul. Grâce à ces deux éléments, l'ensemble constituera un BLETransmitter qui possédera deux antennes émettant unidirectionnellement de part et d'autre de son emplacement. Ces deux antennes pourront ainsi émettre deux messages différents de part et d'autre du BLETransmitter.

Étant donné que le Gestionnaire connaît la cartographie de la route et l'emplacement de ses équipements, il pourra, lors de la procédure d'envoi d'un message, ajouter un identifiant de séquence au sein du *payload* du message (0 ou 1), ainsi que la séquence identifiant la voie concernée par ce message.

Cette solution demande donc au Gestionnaire de :

- Définir l'identifiant de séquence pour chaque message de chaque sous balise.
- Définir la séquence de validation désignant la voie concernée par un évènement.

Elle possède comme avantage de ne nécessiter au minimum qu'un message par ID de séquence. En revanche, l'information transmise est plus volumineuse.

4.3.2 Illustration

Nous illustrons les différentes situations possibles dans le cadre de cette solution. Le BLETransmitter comporte deux antennes unidirectionnelles.

Définition des Figures 4.8 et 4.9 :

- *Msg 1* - identifiant de séquence : "0".
- *Msg 2* - identifiant de séquence : "1".

4.3.2.1 Cas 1 : Le véhicule circule sur la voie A

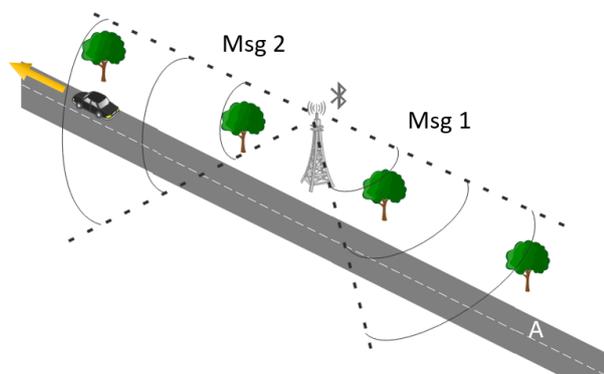


FIGURE 4.8 – Réception de deux messages sur la voie A

La figure 4.8 présente le cas où **le gestionnaire déclare un DENM sur la voie A**. Il identifie donc sa voie par la séquence de validation suivante "0-1" et la transmet dans ses messages.

Le véhicule noir circule sur la voie A, il va alors recevoir deux ensembles de messages, les messages avec l'identifiant 0 en premier puis ceux avec l'identifiant 1.

Ainsi par regroupement de l'ensemble des messages reçus, le véhicule déterminera la séquence reçue.

Ici, le véhicule devrait détecter la séquence "0-1". Or le gestionnaire ayant détecté un DENM sur la voie A, le message donnera la séquence "0-1" comme séquence de validation. Le véhicule est donc sur la voie ciblée par le message, va le prendre en compte et prévenir l'utilisateur.

4.3.2.2 Cas 2 : Le véhicule circule sur la voie B

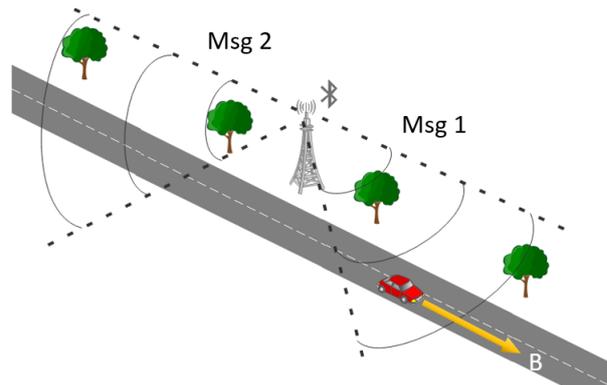


FIGURE 4.9 – Réception de deux messages sur la voie B

La figure 4.9 présente le cas où **le gestionnaire déclare un DENM sur la voie A**. Il identifie donc sa voie par la séquence de validation suivante "0-1" et la transmet dans ses messages.

Le véhicule rouge circule sur la voie B, il va alors recevoir deux ensembles de messages, les messages avec l'identifiant 1 en premier puis ceux avec l'identifiant 0.

Ainsi par regroupement de l'ensemble des messages reçus, le véhicule déterminera la séquence reçue.

Ici, le véhicule devrait détecter la séquence "1-0". Or le gestionnaire ayant détecté un DENM sur la voie A, le message donnera la séquence "0-1" comme séquence de validation. Le véhicule n'est donc pas sur la voie ciblée par le message, ne va pas le prendre en compte et le supprimer.

4.3.3 Expérimentations

Nous avons mené des tests sur l'autoroute sur des portions à 130km/h. Des Raspberrys font office de BLETransmitters, ils sont positionnés de part et d'autre d'un pont afin qu'ils se comportent comme des antennes unidirectionnelles, le pont faisant office d'écran. Les Raspberrys sont toujours équipés de puces de classe 2.

Rappel des conditions de tests :

- *Taille de message* : 15 octets
- *Fréquence d'envoi* : 10 msg/sec

	Sequence enregistrée
1er Passage	0 0
2nd Passage	1 1 1

TABLE 4.2 – Séquences enregistrées durant le test

Le tableau 4.2 montre que les résultats observés lors des différents passages ne permettent pas de déduire une séquence valide. Nous constatons donc que les résultats ne sont pas exploitables pour pouvoir déterminer si un véhicule circule sur une voie ciblée par un évènement.

Dans ce cas de figure, il nous faut des équipements plus performants. Nous disposons bien des RSUs déployées sur les routes, mais le problème réside dans le fait qu'elles ne sont pas équipées de deux antennes unidirectionnelles sur lesquelles nous pourrions paramétrer deux messages avec des identifiants de séquences différents.

4.3.4 Analyse

Nous ne retiendrons pas cette solution au vu des résultats rencontrés avec les Raspberrys. Les résultats seraient potentiellement différents avec des antennes de classe 1 unidirectionnelles, mais des contraintes logistiques et financières interviennent.

En effet, d'une part, doubler les antennes entraînent inévitablement un surcoût sur chacun des équipements, et, d'autre part, les Gestionnaires partenaires du projet annoncent également que positionner et orienter ces antennes, lors d'installation de ce genre d'équipement, est un tache assez ardue entraînant également des surcoûts de main d'oeuvre. **Nous passons à l'étude de la solution suivante.**

4.4 Calcul de la direction

Nous utilisons la comparaison entre l'orientation d'une voie concernée par un évènement au niveau d'une balise, déterminée à partir de l'orientation d'un évènement en lui-même, avec la direction empruntée par un véhicule.

4.4.1 La direction

La direction désigne l'orientation H du véhicule par rapport au Nord. Pour se faire, nous déterminons l'angle formé entre le nez du véhicule et le Nord comme l'illustre la figure 4.10.

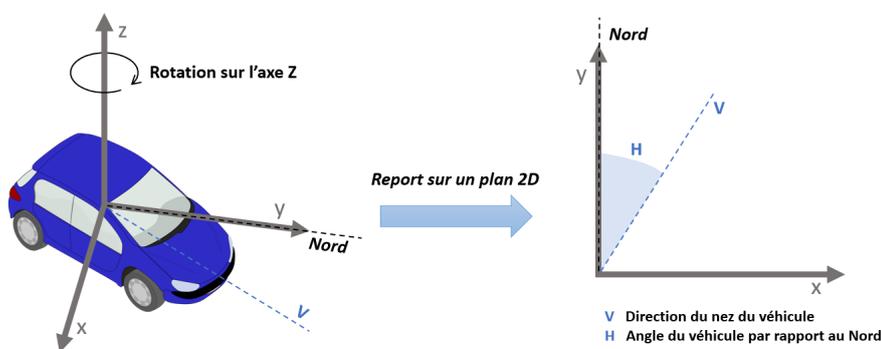


FIGURE 4.10 – Définition de la direction d'un véhicule

Les OBUs peuvent être des appareils de nature différentes (*smartphones* ou systèmes embarqués). En fonction de ces derniers, et de leurs capteurs, il est possible que la direction, ne se détermine pas de la même manière. En général, elle est calculée en interne mais nous réservons, tout de même, deux méthodes de calculs afin d'anticiper sur les différents appareils pouvant intervenir dans l'architecture.

4.4.1.1 Accéléromètre et magnétomètre

Si un accéléromètre et un magnétomètre sont disponibles sur l'OBU, une matrice de rotation est récupérée via ces deux capteurs.

L'appareil détermine sa matrice de rotation en fonction de ses capteurs. Soit la matrice de rotation R [74] [75] :

$$R \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad (4.2)$$

Suivant les [76] [75] [74], nous pouvons établir la matrice de rotation R suivante :

$$R \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (4.3)$$

Où ϕ défini l'angle de roulis, θ l'angle de tangage, ψ l'angle de lacet selon les angles d'Euler et de Tait-Bryan.

Nous cherchons à calculer ψ (que nous désignons comme l'orientation du véhicule H). Pour se faire, une fois R récupérée, nous extrayons les valeurs de R_{21} et R_{11} et effectuons l'opération suivante pour déterminer la direction [76] [77].

$$H = \arctan(R_{21}/R_{11}) \times 180/\pi \quad (4.4)$$

Cette opération détermine l'orientation H en degrés du nez du véhicule par rapport au Nord.

4.4.1.2 Tracé GPS

En cas d'absence d'accéléromètre et/ou de magnétomètre sur l'OBUE, le tracé GPS peut être utilisé pour récupérer l'orientation.

Il faut récupérer deux positions GPS successives afin de déterminer le déplacement relevé et donc l'orientation.

$$\begin{cases} \Delta l = X_{long} - Y_{long} \\ B = \arctan2(\sin(\Delta l) \cos Y_{lat}, \cos X_{lat} \sin Y_{lat} - \sin X_{lat} \cos Y_{lat} \cos(\Delta l)) \end{cases} \quad (4.5)$$

Avec B désignant le sens de déplacement, X et Y deux points GPS.

L'équation 4.5 permet de déterminer la direction du véhicule avec deux points GPS successifs. C'est la méthode que nous utilisons pour nos expérimentations.

4.4.2 Présentation

Dans cette solution que nous proposons, le gestionnaire peut utiliser des balises omnidirectionnelles.

Nous savons que la procédure de diffusion d'un évènement requiert de déterminer quelle voie est concernée par celui-ci. Le Gestionnaire localise donc l'évènement via sa position GPS, et, grâce à sa cartographie, cible les voies concernées par un évènement.

Étant donné que le Gestionnaire connaît ses BLETransmitters, il est en mesure de construire des messages pour chacune de ses balises ; ces messages créés contiennent la valeur de l'angle formé par l'orientation de la voie ciblée par

rapport au Nord, au niveau de la balise, que l'on nomme la "Direction". Ces messages peuvent alors être diffusés par les différentes balises.

Ainsi la voie en question est désignée par une valeur d'angle codée sur un octet. 1 unité = 2 degrés.

C'est une solution assez légère en terme réseaux puisque non seulement la donnée ajoutée n'est pas très volumineuse, mais en plus, il ne suffit que d'un seul message pour l'analyse.

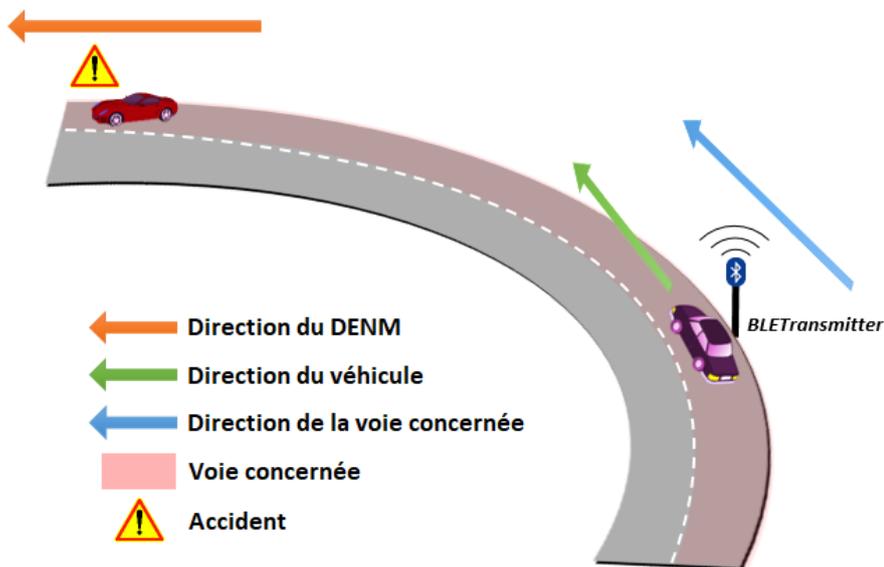


FIGURE 4.11 – Illustration des différentes directions.

L'angle de la voie concernée est utilisé afin d'anticiper le tracé de la route. En effet, comme l'illustre la figure 4.11, un événement peut être orienté de manière différente selon le tracé de la route. De ce fait, le Gestionnaire va identifier la voie concernée en combinant l'orientation de l'évènement ou sa position géographique avec sa cartographie, et donc, lors de l'identification des balises, va calculer, pour chacune d'entre elles, la direction de la voie concernée, à leurs niveaux.

Une fois le message reçu sur une OBU, la direction de la voie concernée (contenue dans le message) est comparée avec l'orientation du véhicule par rapport au Nord, une direction pour le véhicule. Une marge d'erreur de + ou - 45 degrés est appliquée afin d'anticiper sur un éventuel virage entre la réception de l'évènement et son traitement par l'OBU (Voir Section 4.4.4).

Ainsi, si la direction du message et la direction du véhicule correspondent à + ou - 45 degrés, nous en déduisons que le véhicule circule sur la voie ciblée par le message et donc, par extension, l'évènement.

4.4.3 Illustration

Nous illustrons les différentes situations. Un BLETransmitter est placé en amont d'un accident. Des véhicules passent à proximité de la balise.

Dans ce cas, nous supposons que le DENM est déclaré sur la voie A.

4.4.3.1 Cas 1 : Le véhicule circule sur la voie A

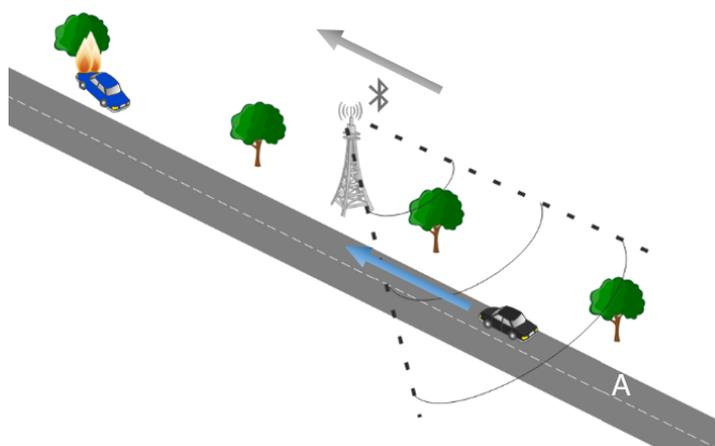


FIGURE 4.12 – Correspondance entre la direction du véhicule et la direction de la voie concernée.

La figure 4.12 illustre le cas dans lequel le message cible la voie A. Le BLE-Transmitter émet ce message.

Dans ce cas, le véhicule noir va être concerné par l'évènement se situant en amont de la route. Le véhicule capte le message, il va alors récupérer sa direction et la comparer à la direction contenue dans le message (Voir Section 3.6.3).

La comparaison s'effectue à ± 45 degrés et, ici, les directions correspondent. Le véhicule noir est donc ciblé par le message puisqu'il circule sur la voie A, l'évènement est alors notifié à l'utilisateur.

4.4.3.2 Cas 2 : Le véhicule circule sur la voie B

La figure 4.13 illustre le cas dans lequel le message cible la voie A. Le BLE-Transmitter émet ce message.

Dans ce cas, le véhicule rouge ne va pas être concerné par l'évènement car il l'a déjà dépassé. Le véhicule capte le message, il va alors récupérer son orientation et la comparer à la direction contenue dans le message (Voir Section 3.6.3).

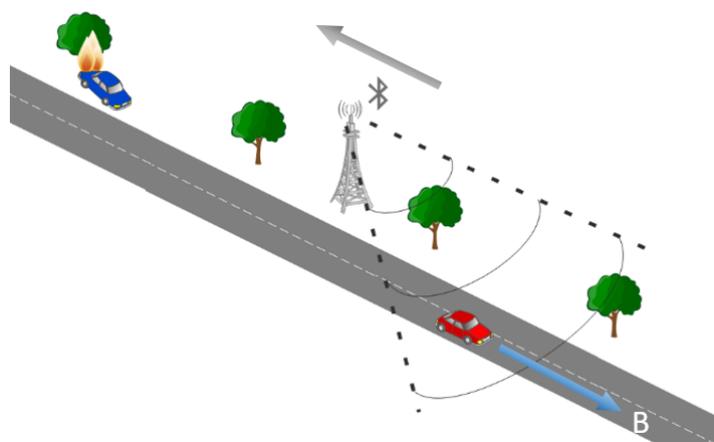


FIGURE 4.13 – Décalage entre la direction du véhicule et la direction de la voie concernée.

La comparaison s'effectue à ± 45 degrés et, ici, les directions diffèrent. Le véhicule rouge n'est donc pas ciblé par le message puisqu'il circule sur la voie B, l'évènement est alors supprimé.

4.4.4 Comparaison des directions

Lors de la récupération d'un message, l'OBU extrait la direction contenue dans le message. Il s'agit de la direction fixée par le Gestionnaire déterminant quelle voie est concernée par le message.

L'OBU récupère ensuite son orientation selon différentes méthodes dont celles utilisées lors de ces travaux (voir Section 4.4.1).

Une fois ces deux éléments récupérés, toutes les données nécessaires sont établies pour lancer la comparaison.

Tout d'abord, la marge d'erreur choisie est de ± 45 degrés. C'est une marge définie dans le Projet C-Roads, on se base sur cette valeur.

Les deux directions sont projetées sur un cercle, comme l'illustre la figure 4.14. Les vecteurs, en bleu, représentent le système d'origine établi via les données récupérées. Ce système se constitue de la direction de l'OBU (VH), de la direction de la voie concernée (EH) et enfin la marge d'erreur définit l'angle $\alpha\beta$ qu'on notera ERR .

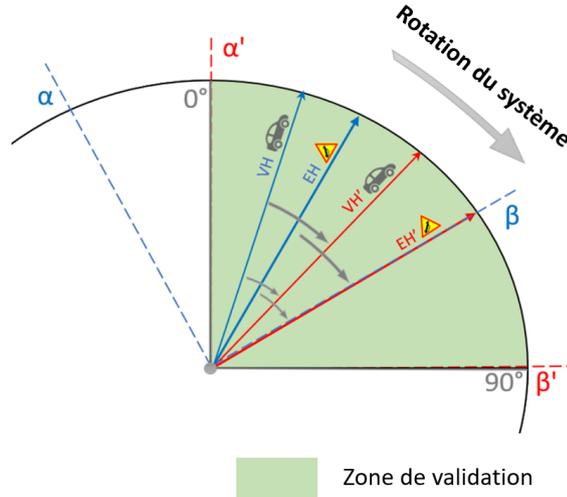


FIGURE 4.14 – Validation de l'évènement par comparaison des directions

Nous appliquons ensuite une rotation afin de comparer les différentes directions. Pour se faire la rotation s'effectue comme suit (voir Système 4.6) :

$$\begin{cases} \alpha' = 0 \\ \beta' = ERR \\ VH' = (VH + ERR - \beta) \pmod{360} \\ EH' = (EH + ERR - \beta) \pmod{360} \end{cases} \quad (4.6)$$

Une fois la rotation du système effectuée, deux cas sont possibles :

- $EH' \in \alpha'\beta'$: l'évènement EH' se situe dans la zone de validation $\alpha'\beta'$ (en vert). Dans ce cas, l'évènement en question concerne le véhicule, il sera donc **traité**,
- $EH' \notin \alpha'\beta'$: l'évènement EH' se situe hors de la zone de validation $\alpha'\beta'$ (en vert). Dans ce cas, l'évènement en question ne concerne pas le véhicule, il sera donc **supprimé**.

4.4.5 Expérimentations

Pour tester la viabilité de cette solution, différents équipements ont été utilisés. Nous nous basons donc sur les données relevées dans la Section 3.7.1.

Dans un premier temps, un Raspberry est déployé afin de vérifier si nous recevons au moins un message.

Et, dans un second temps, des RSUs seront utilisées pour valider les résultats

observés et réaliser des tests avec un scénario correspondant un maximum aux situations de production potentielles.

Nous utilisons, comme appareils récepteurs, des Tablettes Samsung Galaxy Tab A.

Rappel des conditions de tests :

- *Taille de message* : 15 octets
- *Fréquence d'envoi* : 10 msg/sec

4.4.5.1 Tests sur Raspberry

Le Raspberry est placé sur un pont et le véhicule test a réalisé plusieurs passages sous le pont afin de capter les messages du BLETransmitter ; véhicule circulant sur l'autoroute et à 130km/h.

	Voie du DENM ?	NB messages reçus	DENM affiché ?
1er Passage	Oui	2 messages	Oui
2nd Passage	Non	3 messages	Non
3e Passage	Oui	3 messages	Oui
4e Passage	Non	2 messages	Non

TABLE 4.3 – Résultats de l'évaluation des directions avec le Raspberry

La réception des messages permet, comme le présente le tableau 4.3, de recevoir et d'afficher les messages paramétrés au préalable et désignant un évènement sur la route. Les messages sont affichés sur une application ad-hoc des tablettes. Aucun faux positif n'a été relevé lors de ces tests.

L'analyse des résultats nous a permis de poursuivre les tests de cette solution avec des RSUs.

4.4.5.2 Tests sur RSU

Les RSUs sont déployées avec des antennes omnidirectionnelles. Elles se situent selon les localisations définies en Section 3.7.1. Les tests sont menés à 90 km/h et 130 km/h avec les RSUs.

Nous utilisons toujours des tablettes Samsung en réception avec l'application adhoc.

Le tableau 4.4 illustre les résultats des différents passages effectués lors des tests sur autoroute avec RSUs. Sur l'ensemble des passages, 100% des résultats sont cohérents avec les diffusions paramétrées.

En moyenne, à 90 km/h nous récupérons environ 39 messages, et, à 130 km/h, nous recevons environ 28 messages lors de notre passage. Il est d'ailleurs impor-

	Vitesse	Voie du DENM?	Nb Messages Relevés	DENM affiché?
Passage 1	90	Oui	6	Oui
Passage 2	90	Oui	94	Oui
Passage 3	90	Oui	6	Oui
Passage 4	90	Non	51	Non
Passage 1	130	Oui	20	Oui
Passage 2	130	Oui	4	Oui
Passage 3	130	Oui	29	Oui
Passage 4	130	Oui	28	Oui
Passage 5	130	Oui	54	Oui
Passage 6	130	Non	36	Non

TABLE 4.4 – Résultats de l'évaluation des directions avec une RSU

tant de noter qu'ici la réception d'un seul message nous suffit à déterminer si la voie sur laquelle circule le véhicule est concernée par l'évènement.

Nous constatons donc une forte redondance ce qui assure, au véhicule, un maximum de chance de récupérer l'information diffusée.

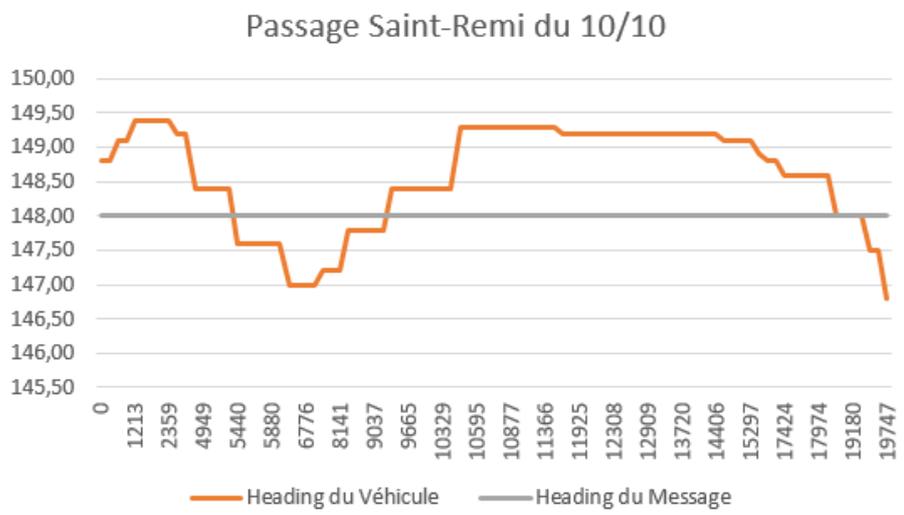
4.4.5.3 Marge d'erreur relevée

Nous définissons une marge d'erreur. Elle intervient lors de la comparaison de la direction contenue dans le message avec celle du véhicule (voir Section 4.4.4). Nous relevons ces deux directions afin de valider l'utilisation de la marge d'erreur.

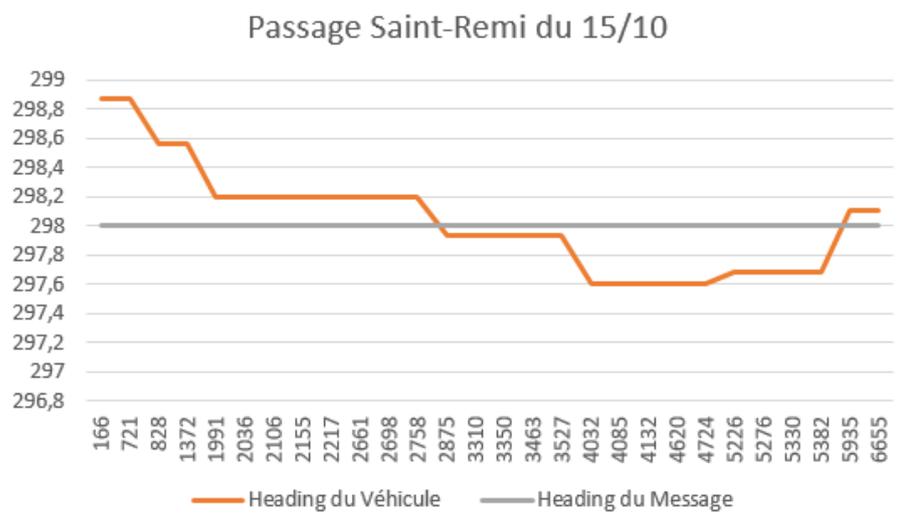
En effet, avec notre marge d'erreur, nous anticipons des changements dans la topographie du terrain et donc sur le tracé de la route.

Les figures 4.15 (a et b) et 4.16 (a et b) illustrent cette fluctuation lors de la récupération des directions du véhicule à la réception d'un message. Cette fluctuation est faible car le véhicule circule sur une portion d'autoroute rectiligne.

En fonction des méthodes utilisées pour établir la direction du véhicule, les traitements peuvent être coûteux en terme de temps. Ainsi lors de ce laps de temps supplémentaire, le tracé de la route peut varier, cette marge d'erreur permet donc de valider le message malgré un virage éventuel. C'est un cas plutôt exceptionnel à l'heure où ces travaux sont réalisés puisque les tests sont menés sur autoroute. Les échangeurs étant réellement les seuls endroits où des changements aussi forts d'orientation peuvent être observés. Cependant, des changements de tracé de route plus fréquents pourraient survenir si l'architecture était appliquée, dans le futur, à des environnements urbains.



(a) 1er passage



(b) 2nd passage

FIGURE 4.15 – Comparaison des directions relevées du véhicule et de la direction paramétrée sur la RSU Saint-Remi

4.4.6 Analyse

Avec cette solution, les messages reçus sont suffisants pour identifier le sens de circulation du véhicule, donc, identifier si un événement doit être traité ou non.

De plus, le Gestionnaire routier met simplement une information dans le mes-

Nous estimons que cette solution permet le ciblage des voies par un message diffusé et permet à l'OBU de mettre en corrélation le ciblage de voie d'un message avec l'orientation du véhicule. Orientation qui peut être récupérée au niveau du véhicule, sans aide de la part de l'infrastructure. **Cette proposition constitue donc une solution fiable pour traiter les messages de notre architecture et est donc validée dans notre cas d'étude.**

4.5 Discussion

Dans ce chapitre nous avons développé les aspects suivants :

1. La nécessité de déterminer l'orientation du véhicule afin de savoir si l'utilisateur circule sur une voie concernée par un événement et donc s'il doit être dérangé en le notifiant pour capter son attention.
2. Les différentes solutions ont été proposées.
3. Les différentes solutions ont été explicitées et éprouvées par des tests.
4. La solution de la direction a été validée pour être utilisée dans l'architecture. Ceci grâce aux résultats obtenus qui sont validés et à sa simplicité de déploiement.

Il est dorénavant possible de déterminer si un usager circule sur la voie concernée par un événement et donc s'il doit être notifié. Ainsi, l'attention du conducteur est sollicitée uniquement pour un message le concernant. Ce chapitre a fait l'objet d'une communication avec acte [78]. Mais le Gestionnaire est-il capable de détecter l'ensemble des événements sur la route et, par conséquent, est-il capable de garantir une sécurité routière complète ? Les usagers ne sont-ils pas une autre source d'information que l'on pourrait prendre en compte ? Nous en discutons dans le chapitre suivant.

Architecture Étendue

Les usagers constituent une source d'informations non négligeable. En effet, ils sont témoins de tout ce qui se passe sur leurs trajets, ils peuvent donc repérer de potentiels événements qui auraient une incidence sur la circulation. De plus, les véhicules sont capables de générer un nombre significatif de données [79] [80]. Dans les C-ITS, des travaux comme [81] utilisent déjà la remontée d'informations, depuis les véhicules, comme source de données. Ici l'auteur propose une nouvelle méthode pour identifier des profils d'utilisateurs de la route en se basant sur la remontée de CAMs.

Dans les projets Scoop@f [1] et C-Roads [2], les OBUs offrent la possibilité de déclarer des DENMs afin de les transmettre aux autres utilisateurs, directement, ou par le biais des infrastructures. C'est ainsi que l'on parle, respectivement, de transmission en V2V ou en V2I.

Si l'on souhaite se conformer à la philosophie de ces technologies il est nécessaire d'envisager des solutions pour assurer cette remontée d'informations et ce, avec un minimum de temps de récupération entre sa constatation et sa réception sur les infrastructures.

De même, dans les zones non couvertes par le BLE, aucun dispositif sera en mesure de fournir des informations aux véhicules, il serait donc intéressant d'envisager une solution pour augmenter la couverture des messages déjà fixés dans le Chapitre 3.

5.1 Introduction

Aujourd'hui, les RSUs sont réparties sur les autoroutes, elles ont des portées de 1 à 2 km en ITS-G5 (très variable en fonction de la topographie, de la hauteur d'installation de l'antenne, de la météo, ...). En BLE leur portée en émission est de l'ordre des 250 m. Cependant, la portée de réception dépend de l'appareil embarqué.

L'OBU possède une portée d'environ une dizaine de mètres dans le cadre des tests effectués en Section 3.7.1 du fait de l'utilisation de puces de classe 2.

5.1.1 Problématique de couverture

Lorsqu'un évènement a lieu sur la route, il est possible qu'il ne se situe pas aux abords d'une RSU. Un utilisateur, le constatant, déclarera et activera la diffusion de l'évènement.

Étant donné que les véhicules, pour des raisons de sécurité, ne font pas partie du domaine de confiance, il incombe au Gestionnaire de valider l'information déclarée. Pour se faire, elle doit être remontée. Or, l'éloignement des RSUs peut augmenter le temps de validation de l'évènement.

En effet, l'information diffusée par le véhicule sera remontée au Gestionnaire via les RSUs, et uniquement aux abords de celles-ci du fait de la faible portée d'émission des OBUs.

Plutôt qu'envisager la multiplication des RSUs, est-il possible d'augmenter la couverture de la zone de réception des RSUs et ce avec des solutions simples et à bas coûts ?

De plus, est-ce possible que cette solution à bas coûts puissent se comporter comme une RSU afin d'augmenter la couvertures des messages I2V ?

5.1.2 Hypothèse formulée

En Section 3.7.1, nous avons utilisé des Raspberrys. Ces équipements étaient capables d'assurer une certaine communication entre le *BackOffice* et les véhicules. Certes, les performances relevées étaient basses dues aux puces Bluetooth utilisées, mais il était possible d'envoyer des informations et les recevoir sur une OBU embarquée dans un véhicule.

Ces équipements sont donc en mesure d'assurer une extension de couverture afin de diffuser des DENMs provenant du Gestionnaire.

De même, nous avons aussi choisi de tester l'inverse, c'est-à-dire utiliser ces Raspberrys pour augmenter les points de récupération de données qui capteront les messages déclarés par les utilisateurs.

5.2 Répartition au sein de l'architecture proposée

Nous localisons, tout d'abord, précisément ces nouveaux équipements dans l'architecture présentée dans la Section 3. Ceci permet de clarifier leur utilité afin de pallier les limites de la technologie employée.

La figure 5.1 illustre l'extension de couverture proposée. Les RSUs sont toujours présentes aux abords des routes. En plus, des Raspberrys, qui seront dorénavant

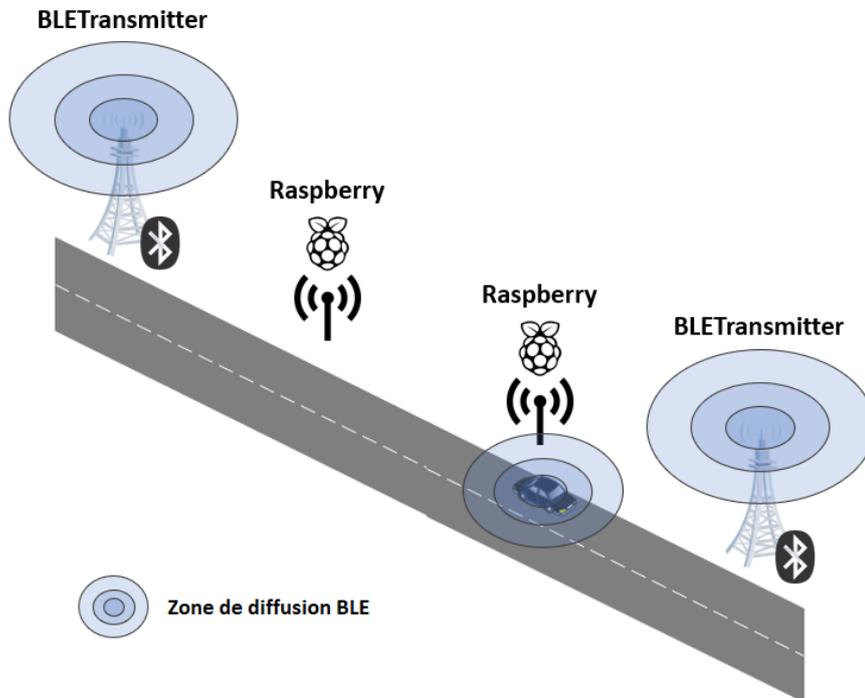


FIGURE 5.1 – Localisation des Raspberrys

désignés comme des "station intermédiaires", sont répartis entre ces différentes RSUs. Ces stations intermédiaires sont en écoute afin de remonter les informations au *BackOffice* ITS et en diffusion afin de transmettre des DENMs. Les stations intermédiaires sont reliées au *BackOffice* ITS via les connexions filaires parcourant les différents axes autoroutiers.

Le véhicule, diffusant une information déclarée par son utilisateur, passe à proximité d'une RSU ou d'une balise intermédiaire. Cette information est captée en BLE et remontée au *BackOffice* afin d'être traitée.

De même, au passage de cette balise, le véhicule, en écoute, peut capter un DENM émis par la station intermédiaire.

5.3 Déclaration d'évènements

En cas d'évènement constaté sur la route, nous mettons à disposition un système déclaratif aux usagers de la route afin d'alimenter le *BackOffice* en données.

5.3.1 Flux de communication

Ici, nous apportons de nouvelles données à remonter, les flux de communication, fixés en Section 3.3 figure 3.3, vont donc s'en retrouver changés.

C'est pourquoi, à partir de maintenant, nous dissocions les DENMs :

- Les DENMs transmis par le Gestionnaire, aux véhicules, via les BLE-Transmitters et les stations intermédiaires comme des "Light DENM I2V".
- Les DENMs déclarés par les véhicules et retransmis au Gestionnaire, via les BLETransmitters et les stations intermédiaires comme des "Light DENM V2I".

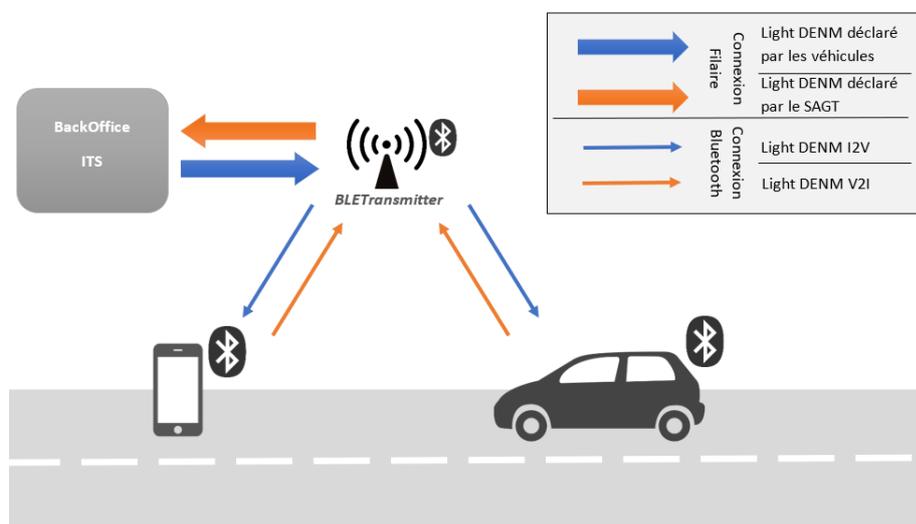


FIGURE 5.2 – Flux de communication de l'architecture générale

Comme le montre la figure 5.2, nous ajoutons, aux flux déjà en place, une possibilité de remonter de l'information.

Un utilisateur peut choisir de déclarer un évènement, il lance donc une diffusion du message déclaré, un **Light DENM V2I**, cela pendant un certain temps, afin que celui-ci soit collecté par une balise. Cette balise remontera alors l'information au *BackOffice* ITS qui, avec plusieurs déclarations successives du même message dans la même zone par plusieurs utilisateurs, va déterminer si un évènement est bel et bien présent. Si le message initial reste isolé, le Gestionnaire peut l'ignorer.

5.3.2 Processus de déclaration

Nous définissons le processus permettant la déclaration d'un évènement.

Un utilisateur constate un évènement sur la route. Il dispose, alors, sur l'interface

de l'OBU, d'un moyen de diffuser cette information. Pour se faire, il caractérise l'évènement via une interface comportant les principaux pictogrammes des anomalies les plus courantes (accident, piéton sur la route, queue de bouchon, etc.).

Le message, au passage d'un BLETransmettreur ou d'une station intermédiaire, est capté. Il peut alors être renvoyé directement au *BackOffice* afin d'être agrégé avec les messages provenant d'autres véhicules.

Un message provenant des véhicules n'est pas issu du domaine de confiance. Ainsi, nous pensons qu'un processus de validation par le nombre permet de trier certains messages frauduleux pouvant être captés. En effet, des personnes mal intentionnées peuvent vouloir déclarer de faux messages. Ce processus d'agrégation et de validation est à la discrétion du gestionnaire, par conséquent, une émission, s'il le décide peut être vecteur déclencheur d'une émission automatique de l'évènement constaté en fonction du nombre de messages reçus, ou alors, la déclaration de l'utilisateur peut servir à faciliter la vision instantanée de l'état du trafic pour le Gestionnaire.

5.3.3 Comportement des appareils dans l'architecture générale

En se basant sur le processus de déclaration défini en Section 5.3.2 et les comportements définis dans la Section 3.4, nous pouvons étendre ces derniers afin de prendre en compte ce processus.

5.3.3.1 Sur le Backoffice ITS

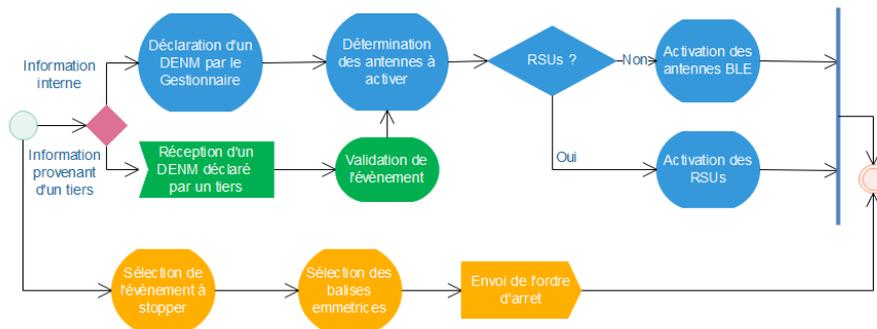


FIGURE 5.3 – Diagramme d'activité du Gestionnaire dans l'architecture générale

Sur la figure 5.3, nous ajoutons le parcours vert, qui montre le cas dans lequel un message est reçu d'un tiers, donc d'une autre autorité ou des véhicules. Dans ce cas, une vérification va s'opérer avant la diffusion, afin de valider le DENM reçu, mais, étant donné que ces messages ne font pas partie du domaine de confiance, il n'est pas conseillé de les prendre en compte.

5.3.3.2 Sur un BLETransmitter

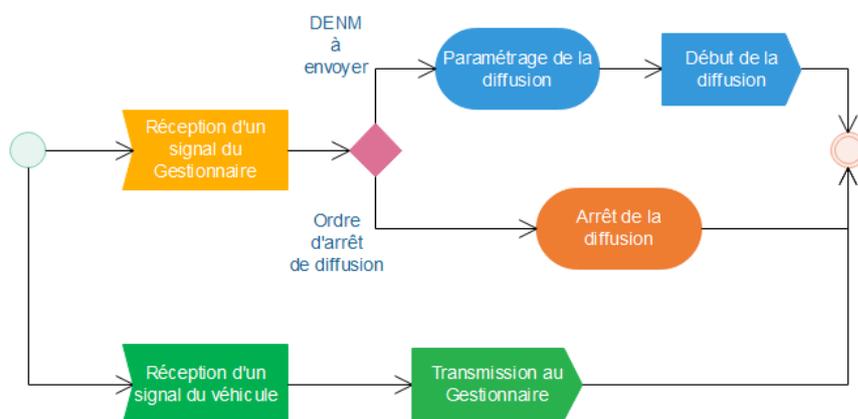


FIGURE 5.4 – Diagramme d'activité du BLETransmitter dans l'architecture générale

Ainsi, la figure 5.4 nous montre qu'en plus des actions/réactions de nos BLE-Transmitters, un BLETransmitter peut **réceptionner et retransmettre**, au Gestionnaire, des **messages provenant des véhicules**. Ces messages seront ainsi traités sur les serveurs du Gestionnaire.

5.3.3.3 Sur l'application *smartphone* ou sur le véhicule

La figure 5.5 nous montre que, maintenant, sur l'application embarquée ou sur l'application *smartphone* nous pouvons recevoir des informations depuis le Gestionnaire mais, également, déclarer un évènement sur la route, ce qui déclenche une diffusion de celui-ci afin qu'il soit capté par un BLETransmitter et remonté au *BackOffice* ITS.

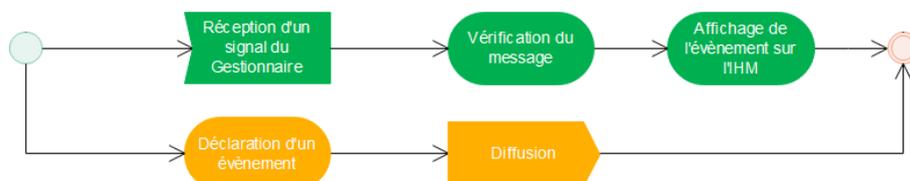


FIGURE 5.5 – Diagramme d'activité du système embarqué dans l'architecture générale

5.3.4 Messages diffusés

Dans le but de déclarer un évènement, nous présentons un format de message permettant d'assurer sa caractérisation. Nous le désignons comme un **Light DENM V2I**.

Ce message représente l'information déclarée depuis un véhicule et retransmise au *BackOffice* ITS, toujours par le biais des BLETransmitters.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
BC	DC	Lat				Long				H	T						

Dans ce datagramme, nous avons :

- *Beacon Code (BC)* (2 octets) - L'identifiant de la balise émettrice.
- *DENM Code (DC)* (1 octet) - Code de l'évènement sur la route, obéissant à un mapping normé, CF. annexe A.
- *Latitude (Lat)* (4 octets) / *Longitude (Long)* (4 octets) - La localisation GPS.
- *Direction (D)* (1 octet) - Angle formé entre le Nord et la direction de l'évènement. Cette donnée permet de déterminer si un véhicule est concerné par un message ou non (voir section 4). 1 unité = 2 degrés.
- *Timestamp (T)* (6 octets) - Horodatage du message.

Le *Beacon Code* (BC) permet de donner l'identifiant de la balise émettrice, donc l'identifiant de l'OBU. Cet identifiant est anonymisé sur le *BackOffice*, il sert simplement à agréger les messages d'une même station.

Le *DENM Code* (DC) définit le type d'évènement relevé sur la route, selon l'annexe A. C'est une approximation du type d'évènement évalué par l'utilisateur lors de sa constatation et de sa déclaration sur son Interface Homme-Machine (IHM) embarquée. Cette information guide le Gestionnaire dans ses recherches lors de la validation de l'évènement.

La localisation GPS, latitude (Lat) et la longitude (Long). À l'instant où l'utilisateur déclare un évènement, l'OBU récupère la position GPS du véhicule afin de localiser approximativement l'anomalie en question. Le Gestionnaire est en mesure de définir une zone de recherche plus réduite pour valider l'évènement.

La *Direction* (D), conformément à la Section 4, définit le sens de déplacement du véhicule. Ainsi, avec la position GPS et sa cartographie, le Gestionnaire peut déterminer sur quelle voie la station, déclarant l'évènement, circule. Par conséquent, le Gestionnaire peut anticiper la voie concernée par un potentiel DENM.

Enfin, le *Timestamp* (T) permet de situer dans le temps le potentiel évènement et ainsi d'anticiper une éventuelle évolution du DENM au cours du temps.

L'ensemble de ces évènements constitue les informations véhiculées par les Light DENM V2I.

5.3.5 Évaluation de la fonctionnalité

Les résultats obtenus en laboratoire sont encourageants.

En effet, en test sur table, il est possible de récupérer des messages, tout en diffusant en BLE.

Caractéristique des tests :

- *Émetteur* - Smartphone Galaxy S7 (puce BLE 4.2 de classe 2).
- *Récepteur* - Raspberry Pi 3 Model B+ (puce BLE 4.2 de classe 2).

Le Light DENM V2I est correctement réceptionné par le Récepteur qui, après traitement, peut sauvegarder le potentiel évènement dans sa base de données.

5.4 Remontée de données véhiculaires

Dans le cadre de zone très accidentogène, nous proposons qu'une zone de surveillance accrue puisse être décrétée. Cette zone aura pour but de détecter plus facilement et plus rapidement d'éventuelles anomalies sur la route par le biais de remontées d'informations de conduite provenant des véhicules.

5.4.1 La zone de surveillance accrue

Cette zone se caractérise par un fort taux d'accidents ou un fort trafic. Dans ce cas, le Gestionnaire peut décider de la déclarer comme zone à surveiller. Il peut donc déployer cette fonctionnalité.

Comme le présente la figure 5.6, des stations intermédiaires sont disposées sur le bord de la route entre deux ou plusieurs RSUs. Toute cette installation caractérise la zone de surveillance.

Lors de l'entrée d'un véhicule dans cette zone, celui-ci va capter un message provenant du BLETransmitter en *Etape 1*. Ce message ordonne aux véhicules de diffuser des informations telles que la vitesse et leurs directions. En *Etape 2 et 3*, les véhicules vont passer à proximité des stations intermédiaires ou des RSUs intermédiaires (entre les RSUs extremum délimitant la zone). Ces stations vont pouvoir capter les messages des véhicules et donc collecter les informations pour les retransmettre au serveur. Enfin, en *Etape 4*, le véhicule passe à proximité de la balise de fin de la zone qui émet un message d'arrêt de surveillance. L'OBV va donc stopper sa diffusion d'informations et revenir dans un état nominal.

A noter que la réception d'un light DENM I2V est encore possible, ce qui ne supprime en rien les fonctionnalités définies en Section 3.

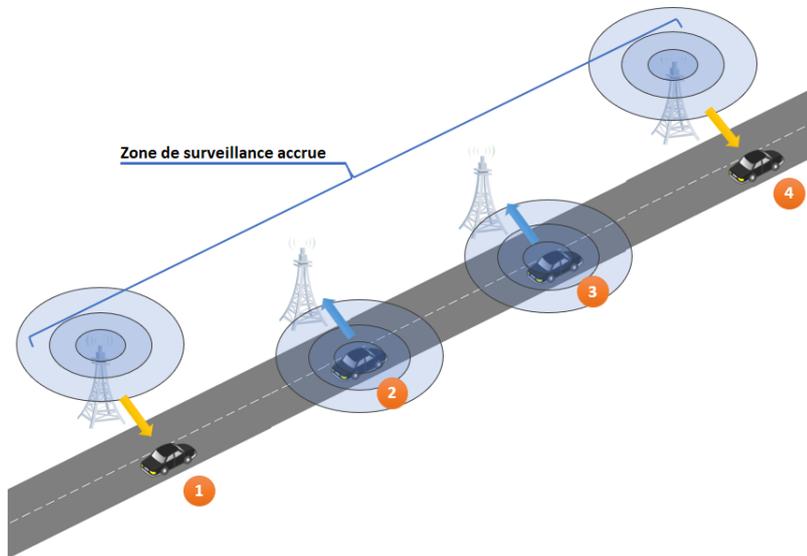


FIGURE 5.6 – Zone de surveillance accrue

5.4.2 Comportement de l’OBU

Cette fonctionnalité modifie le comportement de l’appareil BLE embarqué, nous devons donc mettre à jour notre diagramme disponible en Section 5.3.3 figure 5.5.

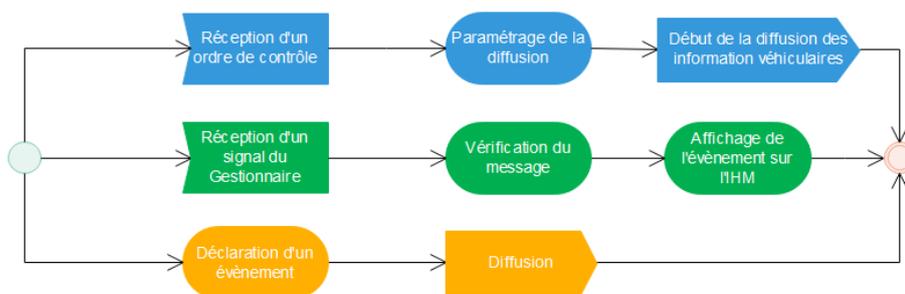


FIGURE 5.7 – Diagramme d’activité du système embarqué dans l’architecture générale

Nous ajoutons, dans la figure 5.7, la possibilité de réceptionner un ordre de contrôle. Cet ordre lance la diffusion BLE de l’OBU avec les paramètres fournis afin de transmettre les informations du véhicule qui nous intéresse.

Ces données transitent ainsi par les BLETransmitters et sont réceptionnées par le *BackOffice* ITS.

5.4.3 Messages utilisés

5.4.3.1 Message d'initialisation de surveillance

0	1	2	3	4	5
BC	DC	CM	nE	T	

Nous positionnons, dans cette structure, les informations suivantes :

- *Beacon Code (BC)* (2 octets) - L'identifiant du BLETransmitter émetteur de l'ordre.
- *DENM Code (DC)* (1 octet) - Valeur définissant le scénario, pouvant être ajoutée dans la table de correspondance disponible en Annexe A.
- *Control Mode (CM)* (1 octet) - 4 Modes possibles : **Mode Début - Fin** : une balise se situe au début, une balise à la fin pour mettre fin à la zone de contrôle (ce message constitue le début de la zone de ce mode). **STOP** : Message de fin de zone (ce message constitue la fin de la zone du "Mode Début - Fin"). **Mode kilomètre** collecte pour x kilomètres. **Mode temps** collecte pour x secondes.
- *Nombre d'enregistrements (nE)* (1 octet) - Définit le nombre d'enregistrements présents dans la table de données servant à déterminer les différentes informations.
- *Timeout (T)* (1 octet) - Définit les conditions d'arrêt des différents modes.
 - **Si Mode Début - Fin** : nombre de secondes (sécurité en cas de non réception du message de fin)
 - **Si Mode kilomètre** : nombre de kilomètres
 - **Si Mode temps** : nombre de secondes.

De plus, en cas de Collecte en "Mode Kilomètre", il est sage d'assurer un *Timeout* applicatif afin d'anticiper une impossibilité de récupérer la distance parcourue.

5.4.3.2 Message d'information véhiculaire

Le véhicule récupère périodiquement des données des capteurs et calcule les informations à transmettre à partir des données récupérées à l'instant T avec celles déjà récupérées précédemment (historique limité par "Nombre d'enregistrements (nE)" dans le Message d'initialisation disponible en Section 5.4.3.1).

L'utilisation d'une file FIFO nous permet de réaliser ce stockage des valeurs qui servent à calculer les informations à émettre (couple de variance des directions et de vitesses moyennes).

0	1	2	3	4	5	6	7
VD				MV			

- *Variance des Directions (VD)* (4 octets) - Variance calculée à partir des dernières directions enregistrées.
- *Moyenne des Vitesses (MV)* (4 octets) - Le véhicule établit une liste des dernières vitesses relevées et la moyenne de cette liste.

Ce message permet de faire remonter ces informations sur le serveur afin de lancer des traitements de mise en évidence d'anomalies sur la route.

Ce message doit être émis sur un UUID spécifique différent des UUIDs présentés en Section 3.6.2.

5.4.3.3 Cas d'usages envisagés

Nous effectuons la détection sur une collecte de plusieurs messages provenant de plusieurs véhicules via plusieurs balises. Les véhicules sont anonymes.

Deux cas de détections envisagés pour l'instant : Obstacle sur la route ou queue de bouchon.

- *Queue de bouchon* - Détection d'une diminution de la vitesse moyenne des véhicules dans une zone et d'une conservation de leur orientation par rapport à l'axe routier.
- *Obstacle sur la route* - Détection d'une diminution potentielle de la vitesse moyenne des véhicules dans une zone et d'une modification de leur orientation par rapport à l'axe routier.

Exemple d'illustration théorique : Route avec une direction de 135 et une vitesse limitée à 130 km/h.

En entrant dans la zone de surveillance le véhicule est considéré comme ayant reçu un Message d'initialisation de type : "Mode Début - Fin" avec comme nombre de données maximum, des files de relevés de 8.

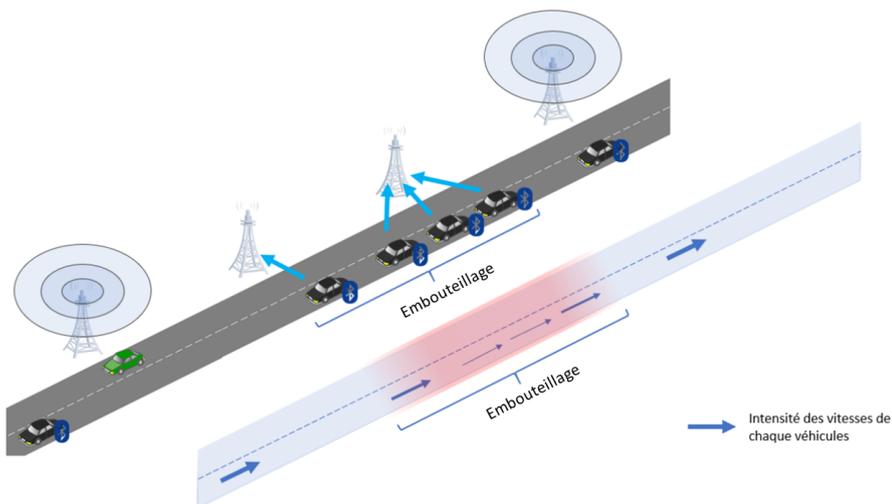


FIGURE 5.8 – Cas de détection de queue de bouchon par observation des vitesses des Véhicules

La figure 5.8 illustre le cas d'un bouchon sur la route. L'ensemble des véhicules

circulant sur la voie sont équipés d'OBU BLE. Lors de leur entrée dans la zone de surveillance, les véhicules diffusent les informations annoncées en Section 5.4.3.2.

Les relevés observés nominalement devraient être ceux-ci :

File de Relevé des vitesses - état nominal							
130	130	130	130	130	130	130	130

Or les relevés, que l'on pourrait attendre, seraient semblables à ceux-ci :

File de Relevé des vitesses - queue de bouchon							
130	130	130	130	100	70	110	130

La moyenne va être calculée à partir de la fenêtre de données énoncée. La divergence des différentes valeurs est ainsi mise en évidence par une moyenne plus basse. Cette donnée est diffusée par le véhicule et captée par une station BLE au bord de la route. Cette dernière retransmet le message au *BackOffice* qui peut rapprocher les données du véhicule avec celles des autres véhicules.

Si cette baisse de vitesse est également constatée, nous suspectons fortement qu'un ralentissement est bel et bien présent.

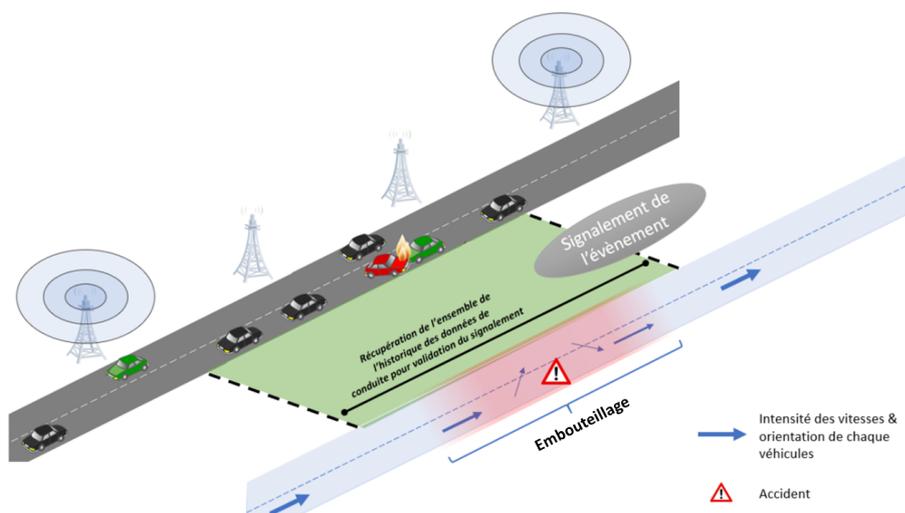


FIGURE 5.9 – Cas de détection d'obstacle sur la route par observation des directions des Véhicules

La figure 5.9 illustre le cas d'un accident sur la route. L'ensemble des véhicules circulant sur la voie sont équipés d'OBU BLE. Lors de leur entrée dans la zone de surveillance, les véhicules diffusent les informations annoncées en Section 5.4.3.2.

Les relevés observés nominalement devraient être ceux-ci :

File de Relevé des directions - état nominal							
135	135	135	135	135	135	135	135

Or les relevés, que l'on pourrait attendre, seraient semblables à ceux-ci :

File de Relevé des directions - obstacle sur la route							
135	135	135	135	135	150	110	135

La variance de ces données va être calculée afin de mettre en évidence cette fluctuation. Cette donnée est diffusée par le véhicule et captée par une station BLE au bord de la route. Cette dernière retransmet le message au *BackOffice* qui peut rapprocher les données du véhicule avec celles des autres véhicules.

Si cette variance est constatée sur l'ensemble des véhicules, nous suspectons qu'un obstacle sur la route est potentiellement présent, ce qui force les véhicules à le contourner.

À noter que dans le cas d'un obstacle sur la route, un abaissement de la vitesse peut également être constaté.

Nous avons donc illustré une solution pour remonter des informations véhiculaires afin qu'elles soient traitées et interprétées sur un serveur central. Mais, ici, il s'agit de surveiller une zone bien précise.

Nous avons vu dans la Section 2.3.5.1 qu'un autre type de message était envoyé dans les C-ITS, les CAMs. Pourrions-nous constituer une fonctionnalité exploitant ces messages ?

5.5 Distribution d'informations semi-ciblées

Dans le cadre de C-ROADS, une solution de communication hybride est mise en place entre le ITS-G5 et le cellulaire (Voir Section 2.3.6). Il est possible d'appliquer le processus de fonctionnement de la communication cellulaire du projet mais en transmettant les informations en BLE par le biais de l'architecture proposée.

Cette solution est étudiée dans le cadre de l'utilisation du Bluetooth 5.0, encore peu rependu, mais qui est en mesure d'assurer les transmissions présentés comme suit.

5.5.1 Fonctionnement

Nous pensons que cette solution est capable de contourner le problème d'identification de voie présenté en Section 4. Elle a pour but de délivrer des messages

ciblés aux véhicules en fonction de ceux qu'ils connaissent déjà et de leur position.

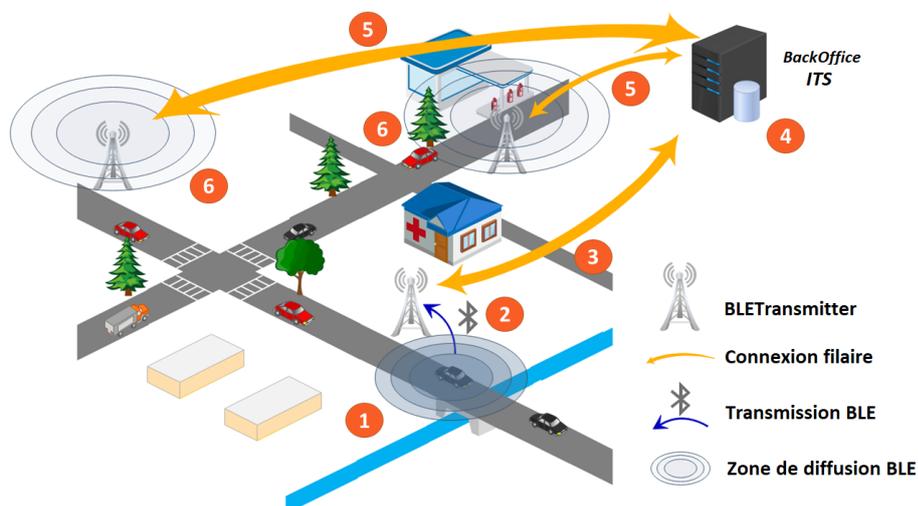


FIGURE 5.10 – Communication ciblée

La figure 5.10 illustre le processus de communication que nous proposons. *Étape 1*, le véhicule équipé d'une OBU BLE diffuse un message de mise à jour de position, un Light CAM, défini en Section 5.5.2. Cette diffusion s'effectue en BLE. *Étape 2*, le Message BLE émis par le véhicule est capté par un BLETransmitter (Une Station Standard de l'architecture ou une Station intermédiaire). Ce message est alors retransmis au BackOffice, en *Étape 3*, via une liaison physique. L'*Étape 4* consiste à traiter le message du véhicule sur le BackOffice.

Le traitement est semblable à celui de la communication en cellulaire.

Sur la figure 5.11, le triage à la réception d'un message sur le BackOffice est illustré. En fonction du type de message reçu, le traitement sera différent.

Dans le cas d'un **light DENM V2I**, sa position est identifiée. Une fois sa localisation établie, le message est archivé dans la base de données. Dans l'éventualité où le message est déjà connu, reconnu par le DENM Code et le *TimeStamp*, le message est ignoré.

Dans le cas d'un **light CAM**, si le *Station ID* du véhicule (défini dans la Section 5.5.2) est déjà connue du *BackOffice*, sa position est mise à jour dans le manager qui gère cet utilisateur, si le véhicule est inconnu, un nouveau manager est créé afin de le gérer.

Une routine est exécutée périodiquement sur le *BackOffice* afin de constituer la liste des DENMs à envoyer pour chaque véhicules.

La figure 5.12 présente le fonctionnement de cette routine. Périodiquement, pour chaque véhicule connu, un processus récupère le dernier couple "Position GPS - direction" renseigné et, grâce à ces deux informations, localise le véhicule.

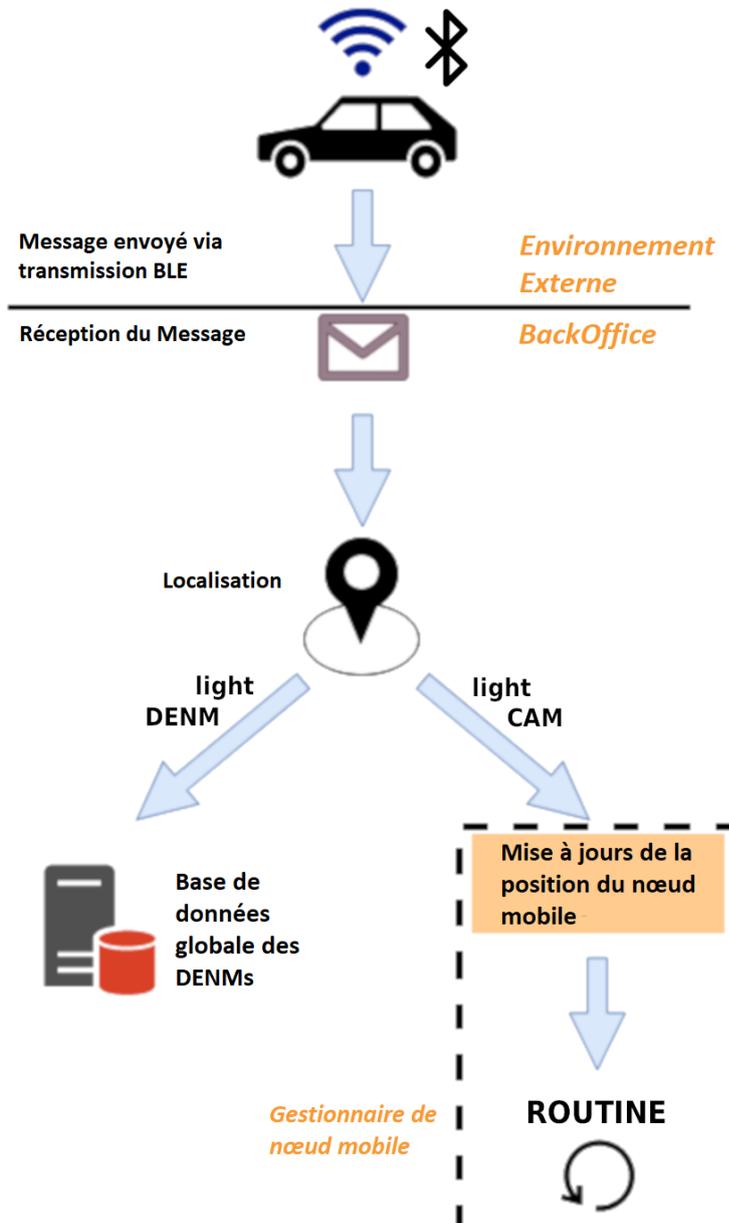


FIGURE 5.11 – Réception d'information sur le Serveur

L'ensemble des DENMs correspondant à cette localisation est alors sauvegardé dans une liste à envoyer.

A noter que cette routine nettoie également périodiquement la base de données en effaçant les DENMs périmés (sur lesquels la durée de validité est arrivée à expiration).

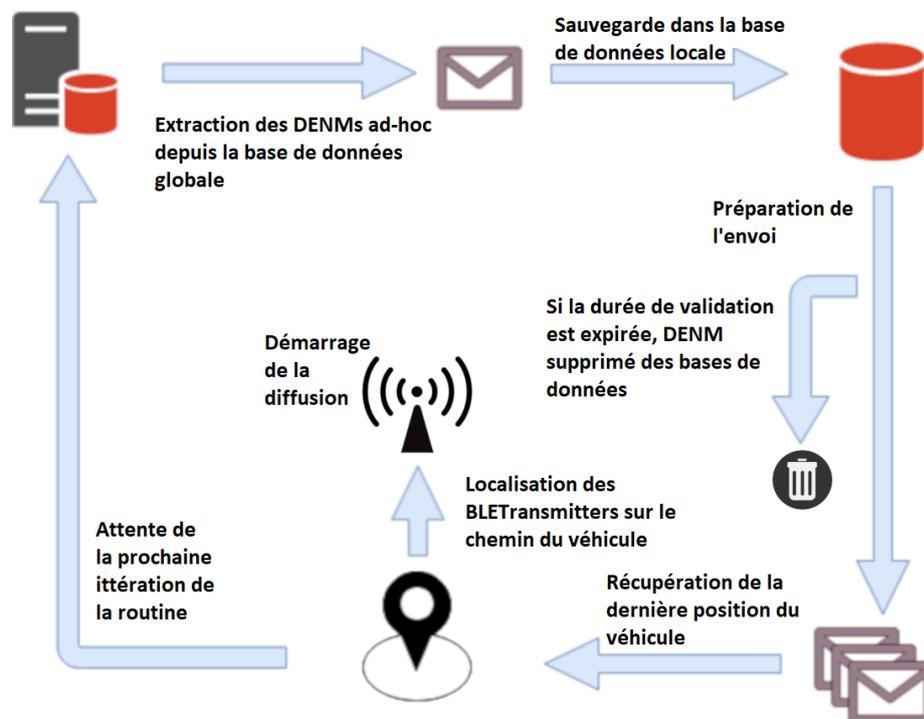


FIGURE 5.12 – Routine d'envoi.

Une fois la liste des DENMs à envoyer complétée, la position et la direction du véhicule sont une nouvelle fois sollicitées avec la cartographie afin de sélectionner les BLETransmitters se trouvant sur la route du véhicule. Enfin, les différents ordres de diffusion sont envoyés et les DENMs sont convertis en Light DENM I2V.

Maintenant que le processus sur le *BackOffice* est clarifié, nous pouvons présenter les étapes suivantes illustrées sur la figure 5.10.

Les ordres d'envoi sont transmis aux différents BLETransmitters en *Étape 5*. Ainsi, en *Étape 6*, les BLETransmitters activés diffusent les informations identifiées par le serveur et à destination du véhicule. Cette diffusion est ordonnée pendant un temps défini par le *BackOffice* selon différents critères (cartographie, condition de circulation, durée de validité de l'évènement, etc...).

5.5.2 Message de position - Le Light CAM

Au sein du projet C-ROADS, pour la communication, des messages de mise à jour de position, appelés, *Cooperative Awareness Message* (CAM) sont utilisés (Voir Section 2.3.5.1). Pour correspondre au processus de communication utilisé dans la communication cellulaire, une version "light" de ce type de message doit être définie.

Nous positionnons, dans cette structure, les données fournies par un light CAM :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Station ID				Lat				Long				D	T			DL ...		

Dans cette structure sont présents :

- *Station ID* (4 octets) - Le pseudonyme du véhicule permettant de l'identifier de manière anonyme et permettant d'agrèger les messages sur le serveur.
- *Latitude* (4 octets) / *Longitude* (4 octets) - La position GPS du véhicule (mise à jour en permanence tant que le GPS est actif).
- *Direction (D)* (1 octet) - L'orientation du véhicule par rapport au Nord.
- *Timestamp (T)* (6 octets) - L'horodatage du message.

A la suite de ces éléments, une liste de *DENM List* (DL) est ajoutée afin de diffuser les messages connus par le véhicule.

Cette *DENM list* se constitue d'ensembles de 3 valeurs (Beacon Code (2 octets) - DENM Code (1 octet) - TimeStamp (6 octets)). Le **Beacon Code** permet d'identifier le BLETransmitter qui a émis ce DENM, le **DENM Code** caractérise l'évènement annoncé et, enfin, le **TimeStamp** donne l'état temporel de l'information. Ainsi, avec ces 3 informations, le BackOffice est dans la capacité de déterminer de quel évènement il s'agit et, s'il est à jour, en rapprochant ses données avec l'historique d'envoi des balises et donc l'historique des différents évènements déclarés par le Gestionnaire.

Ce message doit être émis sur un UUID spécifique, différent des UUIDs présentés en Section 3.6.2.

Enfin, les DENMs, retransmis aux véhicules, suivent l'architecture des Light DENMs I2V.

5.5.3 Évaluation de la fonctionnalité

Théoriquement, nous pensons que cette fonctionnalité présente l'avantage de diminuer les diffusions des RSUs en cas d'environnement ralenti (bouchons, environnement urbain, etc...) et ainsi de diminuer la charge réseau ; ceci est rendu possible par l'arrêt de diffusion, en cas d'absence de nouveaux véhicules dans une zone concernée par un évènement, et par les retours qu'effectuent les véhicules déjà présents dans la zone.

Cela se traduit donc par un service semi-ciblé. C'est-à-dire que les BLETransmitters diffusent toujours une information en *Broadcast* et en clair, mais en prenant aussi en compte les connaissances des différents véhicules sur l'état actuel du trafic (à condition qu'ils embarquent un équipement BLE de l'architecture). Cette fonctionnalité a fait l'objet d'une communication avec acte [82].

Un problème se pose sur l'état actuel des technologies. En effet, celui-ci ne nous permet pas de réaliser une application pratique de cette fonctionnalité de l'architecture. Le BLE 5.0 est trop peu répandu et n'est potentiellement pas

suffisant ; or, pour que la solution fonctionne, il est capital de posséder une taille de *payload* conséquente permettant la transmission de la liste des DENMs au sein du light CAM.

5.6 Discussion

Dans ce chapitre, nous envisageons différentes fonctionnalités afin d'exploiter la remontée d'informations depuis les véhicules.

1. La dispersion géographique des RSUs étant assez éparse, il est possible de mettre en place des stations intermédiaires *low cost* en utilisant des Raspberrys, ce qui étend la couverture BLE.
2. La capacité de pouvoir déclarer des événements en tant qu'utilisateur, de les transmettre au *BackOffice* afin qu'ils puissent être traités et, potentiellement, validés par le Gestionnaire.
3. En exploitant certaines données du véhicule il est possible de détecter passivement, par regroupement d'informations, des accidents ou des queues de bouchon. L'extension de couverture permet d'assurer la communication et la remontée de ces données.
4. En se rapprochant de la philosophie de la communication cellulaire présente dans le projet C-ROADS, il est théoriquement faisable de mettre en place un service de communication semi-ciblé pour certains véhicules. Ceci afin de leur délivrer des informations bien précises et en fonction de leurs besoins. Mais cette fonctionnalité nécessite une évolution des technologies actuellement déployées.

Ces aspects nous montrent la flexibilité d'une telle architecture mais ils soulèvent également un bon nombre de questions définissant des perspectives à suivre pour les futurs travaux.

Conclusion et Perspectives

6.1 Conclusion

Dans cette thèse, nous avons présentés trois contributions s'inscrivant dans les C-ITS.

Dans le Chapitre 3, nous avons prouvé qu'il était possible de mettre en place une architecture de transmission d'informations utilisant le BLE dans un environnement mobile tel que les VANETs. Dans cette solution, nous avons défini deux types d'équipements possibles, les BLETransmitters et les OBU (systèmes embarqués des véhicules et smartphones). Les BLETransmitters sont des balises BLE pilotées par le Gestionnaire, qui ont pour rôle de diffuser des données routières. Ces données sont alors captées par les OBU circulant à portée du BLETransmitter.

Afin de transmettre ces données, nous avons proposé une nouvelle structure de message, plus légère, pour les DENMs afin qu'elle soit transmissible sur des réseaux avec de fortes contraintes de tailles de messages. Ces nouveaux DENMs assurent également de posséder suffisamment d'informations vitales pour assurer la caractérisation d'un évènement.

Cette architecture a fait l'objet de plusieurs phases de tests pour garantir sa faisabilité. Dans un premier temps, par des tests simples afin de déterminer si le BLE est utilisable dans un environnement aussi mobile que les C-ITS. Dans un second temps, par des tests utilisant des Raspberrys Pi afin de mettre en place des scénarios plus complexes et commencer à dégager des performances. Et enfin, dans un troisième temps, avec l'utilisation d'équipements C-ITS déjà déployés (dans le cadre des projets Scoop@f [1] et C-Roads [2]) et plus performants afin de se rapprocher le plus possible d'une situation de production et de ressortir des relevés de performances sur le nombre de messages transmissibles, les portées mis en jeu, les durées pendant lesquelles un véhicule est à portée.

L'utilisation de ce réseau personnel sans fil massivement déployé nous a permis de présenter une réelle alternative aux protocoles de communications déjà mis en place en Europe et qui est en mesure de combler les zones non-couvertes par ces protocoles.

Dans le Chapitre 4, nous avons étudiés différentes solutions pour identifier la voie sur laquelle se déroule un évènement. Nous avons démontré qu'il était

possible de cibler cette voie avec succès, avec un moyen léger. Cette solution garantit également qu'un véhicule, recevant l'information, puisse déterminer si l'évènement est bien sur la voie sur laquelle il circule.

Pour se faire, nous avons énoncés plusieurs propositions et les avons testées afin de choisir celle qui présente les meilleurs performances.

Nous avons étudié une solution prenant en compte la puissance d'un signal (RSSI) en BLE. Celle-ci utilise des équipements unidirectionnels afin que leurs signaux se propagent dans une seule et même direction. Ainsi, il devient possible de renseigner, dans le DENM, si l'évènement concerne un signal en augmentation ou en diminution, suivant l'orientation de la balise activée. Le véhicule est ainsi dans la capacité de repérer une augmentation ou une diminution du signal au passage du BLETransmitter et de repérer si l'évènement est bien sur la voie sur laquelle il circule. En revanche, cette solution ne donnait pas de performances satisfaisantes. Les relevés étaient trop erratiques pour en déduire de quelconques résultats, à l'exception, très rare, de certains.

Ensuite, nous avons énoncé une solution utilisant des identifiants de séquences de messages. Nous avons utilisé un duo d'antennes unidirectionnels (dirigées de part et d'autre du BLETransmitter) qui émettent deux messages avec des identifiants différents. Le Gestionnaire cible la voie concernée par le message en renseignant la séquence à obtenir dans le message et ajoute à chaque message un identifiant. Une fois ceci fait, le gestionnaire ordonne la diffusion de l'évènement, le BLETransmitter émet les deux messages avec les deux identifiants afin qu'ils soient captés par une OBU. Au passage d'un véhicule, celui-ci reçoit les messages, il est en mesure de déterminer l'ordre dans lequel il a reçu les identifiants et donc peut établir s'il circule sur la voie concernée par l'évènement. Les performances relevées, lors des tests de cette solution, n'ont pas permis de l'exploiter.

Par conséquent, nous avons introduit une troisième solution qui utilise la direction. Tout d'abord, il s'agit de présenter ce qu'est la direction : un angle entre un vecteur de direction et le Nord. Cette solution permet donc au gestionnaire de déclarer chacun de ses messages avec une direction donnée. Sachant qu'il connaît la position d'un évènement, il est capable de déterminer l'orientation des voies concernées par celui-ci et ce en tout point. Nous avons proposé qu'il récupère les orientations des voies concernées au niveau du BLETransmitter qu'il souhaite activer. Ainsi, le véhicule, au passage de la balise, capte le message et est également en mesure de récupérer sa propre direction. Il peut, de ce fait, comparer les deux angles obtenus pour établir s'il se trouve sur la voie concernée par l'évènement. Les tests sur route de cette solution ont montré des performances satisfaisantes.

Ainsi, nous avons choisi d'utiliser la solution des directions pour identifier un évènement sur une voie.

Dans le Chapitre 5, nous avons accru les fonctionnalités de l'architecture du chapitre 3, en ajoutant une extension de couverture matérialisée par des stations intermédiaires supplémentaires. Cette extension permet de disséminer des informations routières sur de plus larges zones, avec des équipements simples.

De plus, cette architecture met en lumière un moyen de remonter les informations provenant des véhicules pour exploiter une nouvelle source de données. Nous avons fixé une fonctionnalité pour que les utilisateurs puissent déclarer un évènement. Une structure de message est établie pour remonter ces DENMs déclarés depuis les véhicules, fournissant ainsi des informations essentielles à la caractérisation et au positionnement de l'évènement.

De surcroît, nous avons présenté un moyen de remonter des informations relatives au comportement du véhicule (orientation et vitesse). Et grâce à cette fonctionnalité, nous avons été en mesure de délimiter une zone dans laquelle cette remontée va être surveillée. Le traitement de ces informations permet de mettre en évidence des anomalies sur la route (obstacle sur la voie, queue de bouchon).

De plus, nous avons proposé une procédure supplémentaire, utilisant la remontée d'informations pour limiter les transmissions dans un tel environnement. Pour se faire, nous avons fixé une version légère du CAM. Ce type de message est émis par l'ensemble des véhicules afin qu'un serveur puisse identifier les DENMs qui les concernent. Ainsi, dans une zone donnée, si les véhicules connaissent les évènements en cours, le gestionnaire peut choisir de stopper leurs diffusions dans le but de ne pas surcharger le réseau.

6.2 Perspectives

Dans le Chapitre 3 nous avons présenté l'architecture de transmissions d'informations en BLE, nous avons réalisé de nombreux tests dans le but de valider la solution, mais avec un nombre de véhicules assez réduit. Nous souhaiterions mener une étude volumétrique de la solution afin de déterminer comment se comporte un environnement BLE dense. Nous pourrions ainsi mener une étude plus approfondie, se rapprochant d'autant plus d'une situation de déploiement réelle.

De même, dans l'architecture du chapitre 5 nous avons présenté plusieurs études théoriques des fonctionnalités permises par une extension de la couverture. Nous souhaiterions mener des tests sur route sur la remontée d'informations pour déterminer les performances de telles solutions. Avec ces expérimentations, nous serons en mesure d'élaborer un véritable outil d'aide à la décision pour les Gestionnaires. Ceci dans le but qu'ils puissent cibler, plus facilement, des anomalies sur les axes routiers qu'ils gèrent.

Des travaux devront également être menés sur la détection d'évènements sur la route via la remontée des données provenant des véhicules. Nous souhaitons déployer des équipements physiques afin de réaliser une campagne de tests visant à étudier les performances de cette solution, et les différents cas d'usages supplémentaires pouvant être ajoutés.

Enfin, nous devons chercher un moyen de sécuriser l'ensemble de nos transmis-

sions BLE dans le but d'assurer l'intégrité de nos messages et la robustesse de notre architecture.

Bibliographie

- [1] URL : <http://www.scoop.developpement-durable.gouv.fr/>.
- [2] URL : <https://www.c-roads.eu/>.
- [3] Saif AL-SULTAN et al. « A comprehensive survey on vehicular ad hoc network ». In : *Journal of network and computer applications* 37 (2014), p. 380-392.
- [4] Jakub JAKUBIAK et Yevgeni KOUCHERYAVY. « State of the art and research challenges for VANETs ». In : *2008 5th IEEE Consumer Communications and Networking Conference*. IEEE. 2008, p. 912-916.
- [5] Abduladhim ASHTAIWI, Abdusadik SAOUD et Ibrahim ALMERHAG. « Performance Evaluation of VANETS Routing Protocols ». In : t. 4. Nov. 2014, p. 305-315. DOI : 10.5121/csit.2014.41128.
- [6] Yasser TOOR et al. « Vehicle ad hoc networks : Applications and related technical issues ». In : *IEEE communications surveys & tutorials* 10.3 (2008), p. 74-88.
- [7] Wei LU, Lee D HAN et Christopher R CHERRY. « Evaluation of vehicular communication networks in a car sharing system ». In : *International Journal of Intelligent Transportation Systems Research* 11.3 (2013), p. 113-119.
- [8] Wei XIONG et Qing-Quan LI. « Performance evaluation of data disseminations for vehicular ad hoc networks in highway scenarios ». In : *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37.PART B1 (2008).
- [9] Razvan STANICA, Emmanuel CHAPUT et André-Luc BEYLOT. « Enhancements of IEEE 802.11 p protocol for access control on a VANET control channel ». In : *2011 IEEE International Conference on Communications (ICC)*. IEEE. 2011, p. 1-5.
- [10] T. NEBBOU, H. FOUCAL et M. LEHSAINI. « A Distributed Measurement of Road Density ». In : *2018 IEEE International Conference on Communications (ICC)*. 2018, p. 1-5.
- [11] Ian KU et al. « Towards software-defined VANET : Architecture and services ». In : *2014 13th annual Mediterranean ad hoc networking workshop (MED-HOC-NET)*. IEEE. 2014, p. 103-110.
- [12] URL : <https://www.onisr.securite-routiere.interieur.gouv.fr/>.
- [13] J Karl HEDRICK, Masayoshi TOMIZUKA et Pravin VARAIYA. « Control issues in automated highway systems ». In : *IEEE Control Systems Magazine* 14.6 (1994), p. 21-32.
- [14] Andreas BUCHENSCHWEIT et al. « A VANET-based emergency vehicle warning system ». In : *2009 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2009, p. 1-8.

- [15] Mihail L SICHITIU et Maria KIHIL. « Inter-vehicle communication systems : a survey ». In : *IEEE Communications Surveys & Tutorials* 10.2 (2008), p. 88-105.
- [16] Lars WISCHHOF, André EBNER et Hermann ROHLING. « Information dissemination in self-organizing intervehicle networks ». In : *IEEE Transactions on intelligent transportation systems* 6.1 (2005), p. 90-101.
- [17] Joey ANDA et al. « VGrid : vehicular adhoc networking and computing grid for intelligent traffic control ». In : *2005 IEEE 61st Vehicular Technology Conference*. T. 5. IEEE. 2005, p. 2905-2909.
- [18] F AHMED-ZAID et al. *Vehicle safety communications–applications (vsc-a) final report*. Rapp. tech. 2011.
- [19] Marie-Ange LÈBRE et al. « VANET applications : Hot use cases ». In : *arXiv preprint arXiv :1407.4088* (2014).
- [20] Cristofer ENGLUND et al. « Future applications of VANETs ». In : *Vehicular ad hoc Networks*. Springer, 2015, p. 525-544.
- [21] Wei LU et al. « Performance evaluation of inter-vehicle communication in a unidirectional dynamic traffic flow with shockwave ». In : *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE. 2009, p. 1-6.
- [22] Christian WEISS. « V2X communication in Europe–From research projects towards standardization and field testing of vehicle communication technology ». In : *Computer Networks* 55.14 (2011), p. 3103-3119.
- [23] Kamini KAMINI et Rakesh KUMAR. « Vanet parameters and applications : A review ». In : *Global Journal of Computer Science and Technology* (2010).
- [24] Elias C EZE, Sijing ZHANG et Enjie LIU. « Vehicular ad hoc networks (VANETs) : Current state, challenges, potentials and way forward ». In : *2014 20th International Conference on Automation and Computing*. IEEE. 2014, p. 176-181.
- [25] URL : <https://www.etsi.org/>.
- [26] Ruobing JIANG et Yanmin ZHU. « Wireless Access in Vehicular Environment ». In : *Encyclopedia of Wireless Networks*. Sous la dir. de Xuemin (Sherman) SHEN, Xiaodong LIN et Kuan ZHANG. Cham : Springer International Publishing, 2019, p. 1-5. ISBN : 978-3-319-32903-1. DOI : 10.1007/978-3-319-32903-1_309-1. URL : https://doi.org/10.1007/978-3-319-32903-1_309-1.
- [27] Weidong XIANG et al. *Wireless access in vehicular environments*. 2009.
- [28] Marica AMADEO, Claudia CAMPOLO et Antonella MOLINARO. « Enhancing IEEE 802.11 p/WAVE to provide infotainment applications in VANETs ». In : *Ad Hoc Networks* 10.2 (2012), p. 253-269.
- [29] Ana Maria OROZCO, Roger MICHOD et Gonzalo LLANO. « Routing protocols simulation for efficiency applications in vehicular environments ». In : *Sistemas & Telemática* 11.27 (2013), p. 27-42.
- [30] John KENNEY. « Dedicated Short-Range Communications (DSRC) Standards in the United States ». In : *Proceedings of the IEEE* 99 (août 2011), p. 1162-1182. DOI : 10.1109/JPROC.2011.2132790.

- [31] ETSI ES 202 663; *Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*. 202 663 (ETSI Standard). Nov. 2009.
- [32] ETSI EN 302 663; *Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band*. 302 663 (ETSI Standard). Mai 2013.
- [33] ETSI EN 302 636-4-1; *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4 : Geographical Addressing and forwarding for point-to-point and point-to-multipoint communications; Subpart 1 : Media-Independent Functionality*. 302 636-4-1 (European Standard). Juil. 2014.
- [34] ETSI EN 302 637-3; *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Application; Part 2 : Specification of Cooperative Awareness Basic Service*. 302 637-2 (European Standard). Sept. 2014.
- [35] ETSI EN 302 637-3; *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Application; Part 3 : Specifications of Decentralized Environmental Notification Basic Service*. 302 697-3 (European Standard). Nov. 2014.
- [36] Sangsoo JEONG, Youngmi BAEK et Sang H SON. « A hybrid V2X system for safety-critical applications in VANET ». In : *2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*. IEEE. 2016, p. 13-18.
- [37] Sushma U BHOVER, Anusha TUGASHETTI et Pratiksha RASHINKAR. « V2X communication protocol in VANET for co-operative intelligent transportation system ». In : *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE. 2017, p. 602-607.
- [38] José SANTA, Pedro J FERNÁNDEZ et Fernando PEREÑÍGUEZ. « Deployment of vehicular networks in highways using 802.11 p and IPv6 technologies ». In : *International Journal of Ad Hoc and Ubiquitous Computing* 24.1-2 (2017), p. 33-48.
- [39] Geoffrey WILHELM et al. « A C-ITS central station as a communication manager ». In : *International Conference on Innovations for Community Services*. Springer. 2018, p. 33-43.
- [40] Drew GISLASON. *Zigbee wireless networking*. Newnes, 2008.
- [41] Fabio LECCESE, Marco CAGNETTI et Daniele TRINCA. « A Smart City Application : A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX ». In : *Sensors* 14.12 (2014), p. 24408-24424. ISSN : 1424-8220. DOI : 10 . 3390 / s141224408. URL : <https://www.mdpi.com/1424-8220/14/12/24408>.
- [42] Sinem Coleri ERGEN. « ZigBee/IEEE 802.15. 4 Summary ». In : *UC Berkeley, September 10* (2004), p. 17.
- [43] Michal VARCHOLA et Miloš DRUTAROVSKY. « Zigbee based home automation wireless sensor network ». In : *Acta Electrotechnica et Informatica No 7.1* (2007), p. 4.

- [44] Boris MIHAJLOV et Mitko BOGDANOSKI. « Overview and Analysis of the Performances of ZigBee Based Wireless Sensor Networks ». In : *International Journal of Computer Applications* 29 (sept. 2011), p. 28-35. DOI : 10.5120/3704-5138.
- [45] Amit Singh GAUR et al. « Eco friendly Energy Efficient Smart Wireless Automatic Traffic Light Controller based on Hybrid Usage of Solar Panel, Embedded Systems, Vehicle Sensor, VANET, ZigBee and GSM ». In : ().
- [46] Mamta A RAMTEKE et RK KRISHNA. « Realistic simulation for vehicular ad-hoc network using ZigBee technology ». In : *International Journal of Engineering* 1 (2012).
- [47] O. HERSENT, David BOSWARTHICK et Omar ELLOUMI. *The Internet of Things : Key Applications and Protocols*. Déc. 2011. DOI : 10.1002/9781119958352.
- [48] Behrang FOULADI et Sahand GHANOUN. « Security evaluation of the Z-Wave wireless protocol ». In : *Black hat USA* 24 (2013), p. 1-2.
- [49] C. BISDIKIAN. « An overview of the Bluetooth wireless technology ». In : *IEEE Communications Magazine* 39.12 (2001), p. 86-94.
- [50] Carles GOMEZ, Joaquim OLLER et Josep PARADELLS. « Overview and Evaluation of Bluetooth Low Energy : An Emerging Low-Power Wireless Technology ». In : *Sensors* 12.9 (2012), p. 11734-11753. ISSN : 1424-8220. DOI : 10.3390/s120911734. URL : <https://www.mdpi.com/1424-8220/12/9/11734>.
- [51] Robin HEYDON et Nick HUNN. « Bluetooth low energy ». In : *CSR Presentation, Bluetooth SIG* <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx> (2012).
- [52] URL : <https://blog.groupe-sii.com/le-ble-bluetooth-low-energy/>.
- [53] URL : <https://www.silabs.com/>.
- [54] Jacopo TOSI et al. « Performance Evaluation of Bluetooth Low Energy : A Systematic Review ». In : *Sensors* 17.12 (2017). ISSN : 1424-8220. DOI : 10.3390/s17122898. URL : <https://www.mdpi.com/1424-8220/17/12/2898>.
- [55] SIG BLUETOOTH. *Bluetooth core specification version 4.2*. 2014.
- [56] URL : <https://fr.mathworks.com/help/comm/examples/link-layer-state-machine-for-ble-devices-using-stateflow.html>.
- [57] Joakim LINDH. « Bluetooth® low energy beacons ». In : *Texas Instruments* (2015), p. 2.
- [58] URL : <https://www.bluetooth.com/>.
- [59] Anas BASALAMAH. « BLESS : Opportunistic Crowdsensing Framework for Internet of Things using Bluetooth Low Energy ». In : *The Journal of Engineering* 1 (avr. 2016). DOI : 10.1049/joe.2016.0062.
- [60] URL : <https://developers.google.com/beacons/eddystone>.
- [61] URL : <https://developer.apple.com/ibeacon/>.
- [62] URL : <https://altbeacon.org/>.

- [63] URL : <https://github.com/AltBeacon/spec>.
- [64] URL : <https://github.com/DP-3T/documents>.
- [65] ETSI TS 102 894-2; *Intelligent Transport Systems (ITS); Users and applications requirements; Part 2 : Application and facilities layer common data dictionary*. ETSI TS 102 894-2 (Technical Specification). Sept. 2014.
- [66] Raphael FRANK et al. « Bluetooth Low Energy : An alternative technology for VANET applications ». In : *2014 11th annual conference on wireless on-demand network systems and services (WONS)*. IEEE. 2014, p. 104-107.
- [67] ETSI EN 302 665; *Intelligent Transport Systems (ITS); Communications Architecture*. 302 665 (European Standard). Sept. 2010.
- [68] SIG BLUETOOTH. *Bluetooth core specification version 5.0*. 2016.
- [69] SIG BLUETOOTH. *Bluetooth core specification version 4.0*. 2010.
- [70] Jennifer BRAY et Charles F STURMAN. *Bluetooth 1.1 : connect without cables*. pearson Education, 2001.
- [71] URL : <https://www.bluetooth.com/specifications/assigned-numbers/>.
- [72] EN ETSI. *302 637-3 V1. 2.2 : Intelligent Transport Systems (ITS)*.
- [73] Karel HEURTEFEUX et Fabrice VALOIS. « Is RSSI a Good Choice for Localization in Wireless Sensor Network? » In : *Proceedings - International Conference on Advanced Information Networking and Applications, AINA* (mar. 2012), p. 732-739. DOI : 10.1109/AINA.2012.19.
- [74] James DIEBEL. « Representing attitude : Euler angles, unit quaternions, and rotation vectors ». In : *Matrix* 58.15-16 (2006), p. 1-35.
- [75] Paul BERNER et al. « Technical Concepts : Orientation, Rotation, Velocity and Acceleration, and the SRM ». In : *TENA (Test & Training Enabling Architecture) project by SEDRIS* 21 (2008).
- [76] RD KRIZ. « Extract Euler Angles from the General Rotation Transformation Matrix ». In : (2006).
- [77] URL : <http://planning.cs.uiuc.edu/node103.html>.
- [78] Kevin THOMAS et al. « Efficient Event Dissemination using Bluetooth Protocol ». In : *International Conference on Innovations for Community Services*. Springer. 2019, p. 25-36.
- [79] Yunshu LIU et al. « Traffic big data analysis supporting vehicular network access recommendation ». In : *2016 IEEE International Conference on Communications (ICC)*. IEEE. 2016, p. 1-6.
- [80] N. CHENG et al. « Big Data Driven Vehicular Networks ». In : *IEEE Network* 32.6 (2018), p. 160-167.
- [81] E. BOURDY et al. « New Method for Exemplar Selection and Application to VANET Experimentation ». In : *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, p. 1-6.
- [82] Kévin THOMAS et al. « Intelligent Transport System Based on Bluetooth ». In : *International Workshop on Communication Technologies for Vehicles*. Springer. 2019, p. 50-59.

ANNEXE A

Table de correspondance des CauseCodes et SubCauseCodes en BLE

Ce Tableau détaille les correspondances entre les couples CauseCodes - SubCauseCodes caractérisants chaque DENM dans le DataDictionnary [65] et les Codes Transmis par le BLE dans C-Roads

CauseCode	Value	SubCauseCode	Value	Code BLE
reserved	0	unavailable	0	0
traffic Condition	1	xxx	0	0
		unavailable	0	1
		increasedVolumeOfTraffic	1	2
		trafficJamSlowlyIncreasing	2	3
		trafficJamIncreasing	3	4
accident	2	trafficJamStronglyIncreasing	4	5
		multiVehicleAccident	0	6
		heavyAccident	1	7
		accidentInvolvingLorry	2	8
		accidentInvolvingBus	3	9
roadworks	3	accidentInvolvingHazardousMaterials	4	10
		accidentOnOppositeLane	5	11
		unsecuredAccident	6	12
		assistanceRequested	7	13
		unavailable	8	14
		majorRoadworks	0	15
		roadMarkingWork	1	16
		slowMovingRoadMaintenance	2	17
adverseWeather Condition-Adhesion	6	shortTermStationaryRoadworks	3	18
		streetCleaning	4	19
		winterService	5	20
		unavailable	6	21
		heavyFrostOnRoad	0	22
		fuelOnRoad	1	23
		mudOnRoad	2	24
		snowOnRoad	3	25
		iceOnRoad	4	26
		blackIceOnRoad	5	27
		oilOnRoad	6	28
hazardousLocation-SurfaceCondition	9	looseChippings	7	29
		instantBlackIce	8	30
		roadsSalted	9	31
		unavailable	10	32
		rockfalls	0	33
		earthquakeDamage	1	34
		sewerCollapse	2	35
		subsidence	3	36
		snowDrifts	4	37
		stormDamage	5	38
hazardousLocation-ObstacleOnTheRoad	10	burstPipe	6	39
		volcanoEruption	7	40
		fallingIce	8	41
		unavailable	9	42
		shedLoad	0	43
		partsOfVehicles	1	44
		partsOfTyres	2	45
		bigObjects	3	46
hazardousLocation-AnimalOnTheRoad	11	fallenTrees	4	47
		hubCaps	5	48
		waitingVehicles	6	49
		unavailable	7	50
		wildAnimals	0	51
hazardousLocation-AnimalOnTheRoad	11	herdOfAnimals	1	52
		smallAnimals	2	53
		largeAnimals	3	54
		largeAnimals	4	55

CauseCode	Value	SubCauseCode	Value	Code BLE
humanPresenceOnTheRoad	12	unavailable	0	56
		childrenOnRoadway	1	57
		cyclistOnRoadway	2	58
		motorcyclistOnRoadway	3	59
wrongWayDriving	14	unavailable	0	60
		wrongLane	1	61
		wrongDirection	2	62
rescueAndRecoveryWorkInProgress	15	unavailable	0	63
		emergencyVehicles	1	64
		rescueHelicopterLanding	2	65
		policeActivityOngoing	3	66
		medicalEmergencyOngoing	4	67
childAbductionInProgress	5	68		
adverseWeatherCondition-ExtremeWeatherCondition	17	unavailable	0	69
		strongWinds	1	70
		damagingHail	2	71
		hurricane	3	72
		thunderstorm	4	73
		tornado	5	74
blizzard	6	75		
adverseWeatherCondition-Visibility	18	unavailable	0	76
		fog	1	77
		smoke	2	78
		heavySnowfall	3	79
		heavyRain	4	80
		heavyHail	5	81
		lowSunGlare	6	82
		sandstorms	7	83
swarmsOfInsects	8	84		
adverseWeatherCondition-Precipitation	19	unavailable	0	85
		heavyRain	1	86
		heavySnowfall	2	87
		softHail	3	88
slowVehicle	26	unavailable	0	89
		maintenanceVehicle	1	90
		vehiclesSlowingToLookAtAccident	2	91
		abnormalLoad	3	92
		abnormalWideLoad	4	93
		convoy	5	94
		snowplough	6	95
		deicing	7	96
saltingVehicles	8	97		
dangerousEndOfQueue	27	unavailable	0	98
		suddenEndOfQueue	1	99
		queueOverHill	2	100
		queueAroundBend	3	101
queueInTunnel	4	102		
vehicleBreakdown	91	unavailable	0	103
		lackOfFuel	1	104
		lackOfBatteryPower	2	105
		engineProblem	3	106
		transmissionProblem	4	107
		engineCoolingProblem	5	108
		brakingSystemProblem	6	109
		steeringProblem	7	110
tyrePuncture	8	111		
postCrash	92	unavailable	0	112
		accidentWithoutECallTriggered	1	113
		accidentWithECallManuallyTriggered	2	114
		accidentWithECallAutomaticallyTriggered	3	115
accidentWithECallTriggeredWithoutAccess-ToCellularNetwork	4	116		
humanProblem	93	unavailable	0	117
		glycemiaProblem	1	118
		heartProblem	2	119
stationaryVehicle	94	unavailable	0	120
		humanProblem	1	121
		vehicleBreakdown	2	122
		postCrash	3	123
		publicTransportStop	4	124
carryingDangerousGoods	5	125		
emergencyVehicleApproaching	95	unavailable	0	126
		emergencyVehicleApproaching	1	127
		prioritizedVehicleApproaching	2	128
hazardousLocation-DangerousCurve	96	unavailable	0	129
		dangerousLeftTurnCurve	1	130
		dangerousRightTurnCurve	2	131
		multipleCurvesStartingWithUnknown-TurningDirection	3	132
		multipleCurvesStartingWithLeftTurn	4	133
multipleCurvesStartingWithRightTurn	5	134		
collisionRisk	97	unavailable	0	135
		longitudinalCollisionRisk	1	136
		crossingCollisionRisk	2	137
		lateralCollisionRisk	3	138
vulnerableRoadUser	4	139		
signalViolation	98	unavailable	0	140
		stopSignViolation	1	141
		trafficLightViolation	2	142
		turningRegulationViolation	3	143
dangerousSituation	99	unavailable	0	144
		emergencyElectronicBrakeEngaged	1	145
		preCrashSystemEngaged	2	146
		espEngaged	3	147

CauseCode	Value	SubCauseCode	Value	Code BLE
		absEngaged	4	148
		aebEngaged	5	149
		brakeWarningEngaged	6	150
		collisionRiskWarningEngaged	7	151

Communications Bluetooth pour les Systèmes de Transport Intelligents Coopératifs.

Cette thèse propose une solution permettant de transmettre des informations routières par le biais de réseaux personnels sans fil (*Wireless Personal Area Network - WPAN*).

À travers cette thèse, nous proposons une architecture de communication s'inscrivant dans les Systèmes de transports intelligents coopératifs (*Cooperative Intelligent Transport Systems - C-ITS*). Cette architecture utilisera le Bluetooth Low Energy (BLE) pour transmettre des informations routières. De plus, nous avons adapté le message *Decentralized Event Notification Message (DENM)*, connu du monde des C-ITS pour être envoyé dans un réseau BLE. Cette adaptation veille à comporter suffisamment d'informations vitales pour assurer la caractérisation d'un évènement. Cette architecture est validée par des tests sur route.

Nous présentons, ensuite, des solutions pour identifier la voie sur laquelle se déroule l'évènement. Elles doivent permettre, aux véhicules, de savoir si l'évènement concerne la voie sur laquelle ils circulent. Ces solutions font l'objet d'expérimentations. Nous avons sélectionné celles qui donnent les meilleures performances.

Ensuite, nous présentons une architecture étendue proposant l'extension de la couverture de notre réseau, en installant des stations intermédiaires. Ces stations permettent de considérer différentes fonctionnalités à ajouter à l'architecture.

STI, Réseaux véhiculaires, Méthodologie, STIC, Bluetooth, BLE

Bluetooth Communications for Cooperative Intelligent Transport Systems.

This thesis proposes a solution to transmit traffic information via Wireless Personal Area Networks (WPAN).

Through this thesis, we propose a communication architecture that fits within the Cooperative Intelligent Transport Systems (C-ITS). This architecture will use Bluetooth Low Energy (BLE) to transmit traffic information. In addition, we have adapted the Decentralized Event Notification Message (DENM) message known from the C-ITS world to be sent in a BLE network. This adaptation provides sufficient vital information to ensure the characterization of an event. This architecture is validated by road tests.

Then, we will present solutions to identify the path on which the event takes place. They must allow vehicles to know if the event is on the lane on which they are travelling. These solutions are the subject of experiments. We have selected those that give the best performance.

We shall present an extended architecture proposing the extension of the coverage of our network by installing intermediate stations. These stations make it possible to consider different functionalities to be added to the architecture.

ITS, VANET, Méthodologie, Communication, Bluetooth, BLE

Discipline : INFORMATIQUE

Spécialité : Informatique

Université de Reims Champagne-Ardenne

CRéSTIC – EA 3804

UFR Sciences Exactes et Naturelles,
Moulin de la House, 51687 Reims

