

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE AMAR TELIDJI – LAGHOUAT –

Faculté des Sciences



Département de: **Mathématiques et Informatique**

MÉMOIRE

Présenté par :

BRIK Bouziane

Pour l'obtention du diplôme de MAGISTER en Informatique

Option : Informatique **Répartie et Mobile (IRM)**

THÈME

**Etude des protocoles de collecte de données
dans les réseaux véhiculaires**

Soutenu le 25/04/2013, devant le jury formé de :

Mr M. B. YAGOUBI	Professeur	Université de Laghouat	Président
Mr Y. OUINTEN	Maître de conférences	Université de Laghouat	Examineur
Mme F. CHOUIREB	Maître de conférences	Université de Laghouat	Examinatrice
Mr N. LAGRAA	Maître de conférences	Université de Laghouat	Rapporteur
Mme H. BOUZOUAD	Maître de conférences	Université de Laghouat	Invitée

N° d'ordre :/2013-M/INF

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

J'E tiens à remercier en premier lieu mon Dieu qui m'a donné la santé, la force et le courage pour mener à bien la réalisation de ce travail.

Je remercie chaleureusement mes parents qui m'ont beaucoup aidé matériellement et moralement, depuis mon enfance jusqu'à ce que je suis arrivé à ce point là, sans oublier aussi mes sœurs, mes frères et tous les membres de ma famille.

J'exprime tous mes remerciements à mon directeur de thèse, M. Nasreddine LAGRAA, qui m'a donné une grande chance de travailler dans un domaine de recherche passionnant. Son soutien, ses conseils, ses encouragements ainsi que ses remarques très pointues m'ont certainement permis de mener à bien mon travail de recherche.

Je souhaite adresser mes remerciements à tous les membres de jury d'avoir accepté d'évaluer mon travail :

- Mr M .B YAGOUBI, professeur à l'université de Laghouat.,
- Mr Y.OUINTEN, maître de conférences à l'université de Laghouat,
- Mme F.CHOUIREB, maître de conférences à l'université de Laghouat.

Je remercie également tous mes enseignants qui ont contribué à ma formation et un remerciement particulier à ma tante Mme Hadda BOUZOUAD qui était toujours disponible pour m'apporter son aide. Ses conseils m'ont été très utiles scientifiquement et humainement.

Merci à tous ceux qui ont participé de près ou de loin à l'aboutissement de ce travail.

*À ma Mère,
À mon Père,
À mes Sœurs,
À mes Frères,
À toute ma Famille,
À tous ceux qui me sont chers,...*
Je dédie affectueusement ce modeste travail.

Sommaire

Table des figures	8
Liste des tableaux	10
Introduction générale	12
1 Introduction aux réseaux véhiculaires	14
1.1 Les réseaux sans-fil	14
1.1.1 Mode avec infrastructure (cellulaire)	15
1.1.2 Mode sans infrastructure (Ad-hoc)	15
1.2 Les réseaux véhiculaires	18
1.2.1 Architectures de communication	19
Communications Véhicule à Véhicule (V2V)	19
Communications Véhicule à Infrastructure (V2I)	19
Communications hybrides	19
1.2.2 Environnements de déploiement	20
Milieu urbain	20
Milieu autoroutier	20
1.2.3 Caractéristiques des réseaux véhiculaires	20
1.2.4 Applications des réseaux véhiculaires	21
1.2.5 Travaux de standardisation et de normalisation	21
Le modèle IEEE	22
ISO : TC204/WG16 - CALM	23
ETSI : TC ITS	23
1.2.6 Projets et consortiums	24
1.3 Conclusion	26
2 La collecte de données dans les réseaux de capteurs sans-fil	27
2.1 Les réseaux de capteurs sans-fil classiques (WSNs)	27
2.2 La collecte de données dans WSNs	28
2.2.1 Le protocole Directed Diffusion (DD)	29
2.2.2 Le protocole SPIN (Sensor Protocols for Information via Negotiation)	30
2.2.3 Le protocole CBRP (Cluster Based Routing Protocol)	32
2.3 Les réseaux de capteurs sans-fil avec des éléments mobiles (WSNs-ME)	32
2.3.1 Les avantages et les défis des WSNs-ME	32
2.3.2 Les composants des WSNs-ME	33

2.3.3	Les architectures des WSNs-ME	33
2.4	La collecte de données dans WSNs avec des éléments mobiles	35
2.4.1	Taxonomie des protocoles de la collecte de données dans WSN-ME	35
2.4.2	La découverte de l'élément mobile	36
	Les protocoles du rendez-vous programmé (planifié)	36
	Les protocoles sur demande	36
	Les protocoles asynchrone	37
2.4.3	Le transfert de données	37
2.4.4	Le routage de données	37
	Le routage à plat	37
	Le routage basé sur des proxys	38
2.5	Conclusion	40
3	Les protocoles de collecte de données dans VANets	42
3.1	Les critères de classification	42
3.2	Les méthodes d'accès au canal dans les réseaux véhiculaires	43
3.2.1	Les méthodes Contention-based	44
3.2.2	Les méthodes Contention-free	44
3.3	Les protocoles de collecte dans les réseaux véhiculaires	45
3.3.1	Le protocole CGP (Clustered Gathering Protocol)	45
	Phase d'élection d'un ClusterHead	45
	Phase de collecte de données locales	46
	Phase d'agrégation des données collectées	46
	Phase de diffusion de données agrégées vers les RSUs	46
	Bilan et Discussion	47
3.3.2	Le protocole TrafficGather	47
	Phase de configuration du réseau	47
	Phase de collecte de données	48
	Phase d'envoi des données vers l'initiateur de la collecte	49
	Bilan et Discussion	50
3.3.3	Le protocole Road Probing	50
	La sélection des véhicules de la collecte	50
	La communication entre les véhicules sélectionnés	52
	Bilan et Discussion	52
3.3.4	Le protocole COL	53
	La notion de vue locale et r-opérateur	53
	La collecte de données	53
	La gestion du voisinage dynamique	54
	La terminaison de la collecte	54
	Bilan et Discussion	54
3.4	Etude comparative	54
3.5	Conclusion	55

4	Nos protocoles de collecte de données	56
4.1	Le protocole CDGP (Clustered Data Gathering Protocol)	56
4.1.1	Initiation du processus de la collecte	57
	La diffusion du paquet Col par le RSU initiateur	57
	La diffusion du paquet Col par les véhicules	58
4.1.2	Le partitionnement de la route	59
	Le partitionnement de la route en segments	59
	Le partitionnement des segments de collecte en blocs	60
4.1.3	Élection des ClusterHeads	62
	Le principe générale de notre algorithme	62
	Les avantages de notre algorithme d'élection	63
4.1.4	La collecte et l'agrégation des données	64
	Les formats de paquets utilisés	64
	La collecte par le ClusterHead	65
	La collecte par les véhicules membres	67
4.2	Le protocole TCDGP (Token based Clustered Data Gathering Protocol)	68
4.2.1	La collecte et l'agrégation des données dans TCDGP	69
	Le principe générale du TCDGP	69
	La collecte par le ClusterHead	69
	La collecte par les véhicules membres	70
4.3	Comparaison avec les autres protocoles de collecte	71
4.4	Conclusion	72
5	Simulations et résultats	73
5.1	La simulation	73
5.1.1	Environnement de simulation	73
5.1.2	Le modèle de mobilité	74
5.1.3	Les paramètres de simulation	74
5.1.4	Les scénarios de simulation	75
5.2	Résultats de simulation et analyse des performances	76
5.2.1	Analyse des performances du protocole CDGP	76
5.2.2	Analyse des performances du protocole TCDGP	77
5.2.3	Comparaison entre nos deux protocoles CDGP et TCDGP	79
5.2.4	Comparaison avec le protocole CGP	82
5.3	Conclusion	84
	Conclusion générale	85
	Annexe A	87
.1	Le protocole CDGP(Clustered Data Gathering Protocol)	87
.1.1	La phase de propagation de message de collecte(CMP)	87
.1.2	La phase d'agrégation et de dissémination de données (DAD)	89
.2	Le protocole TCDGP(Token-based Clustered Data Gathering Protocol)	92
.2.1	La phase d'agrégation et de dissémination de données (DAD)	92
	Bibliographie	94

Table des figures

1.1	Réseau en mode avec infrastructure.	15
1.2	Un réseau Ad-hoc mobile.	16
1.3	Un réseau de mesh.	17
1.4	Réseau Ad-hoc de véhicules.	18
1.5	Véhicule intelligent.	19
1.6	Le modèle DSRC/WAVE.	23
1.7	ETSI : TC ITS.	24
2.1	Un réseau de capteurs sans-fil.	28
2.2	Phase de propagation d'intérêt.	29
2.3	Phase d'établissement des gradients.	30
2.4	Phase de renforcement des chemins.	30
2.5	Le déroulement du protocole SPIN : Etape 1, Etape 2 et Etape 3.	31
2.6	Architectures des WSN-ME.	34
2.7	Scénario de la collecte de données dans WSN-ME.	35
2.8	Taxonomie des approches de collecte de données dans WSN-ME.	36
2.9	Le Two-Tier Data Dissemination (TTDD).	39
2.10	Le Scalable Energy-efficient Asynchronous Dissemination(SEAD).	40
3.1	Architecture en clusters.	43
3.2	L'élection du cluster head.	46
3.3	Communication V2V puis V2I.	47
3.4	Formation des Clusters Space.	48
3.5	Le partitionnement d'un CS en blocs.	49
3.6	Synchronisation des cluster spaces adjacents.	49
3.7	Les zones de sélection.	51
3.8	Le protocole Road Probing : modèle 1.	51
3.9	Le protocole Road Probing : modèle 2.	52
3.10	Communication multi-sauts vers le RSU.	52
4.1	Le format du paquet Col.	58
4.2	La diffusion du paquet Col par un RSU puis par les véhicules.	59
4.3	Le partitionnement en segments.	60
4.4	Le partitionnement d'un segment de collecte en blocs.	61
4.5	Le tableau Vector.	65
4.6	Un exemple de réservation du tableau Vector, itération 1 et itération 2.	67

5.1	La fenêtre principale de notre simulation.	74
5.2	Le partitionnement de la route.	75
5.3	Le nombre de slots consommés pour une communication V2V.	76
5.4	La retransmission des messages.	77
5.5	Le nombre de slots consommés pour une communication V2V.	78
5.6	La retransmission des messages.	79
5.7	Le nombre de slots consommés pour une communication V2V.	80
5.8	Le nombre des slots perdus.	81
5.9	Le nombre de réélection des ClusterHead.	82
5.10	Le nombre de messages échangés pour une communication V2V.	83

Liste des tableaux

1.1	Propriétés des différents types du réseau Ad-hoc.	18
1.2	Projets et consortiums dans VANet.	25
3.1	Comparaison entre les protocoles de collecte existants.	55
4.1	Comparaison entre nos deux protocoles et les autres protocoles de collecte.	72
5.1	Les paramètres de la simulation.	75
5.2	Comparaison entre nos deux protocoles.	81

Résumé

Le développement rapide des technologies sans-fil au cours de ces dernières années a vu l'apparition de nouveaux systèmes de communication dont les réseaux Ad-hoc véhiculaires (VANet). Ces réseaux permettent des échanges de données entre véhicules ou encore entre véhicules et infrastructure afin d'améliorer la sécurité et le confort sur le réseau routier.

Acquises individuellement, les informations collectées par les différents capteurs d'un même véhicule fournissent le plus souvent une connaissance moins précise et incomplète sur son environnement. Cependant, l'utilisation des informations issues de plusieurs véhicules permet de combler les lacunes de perception d'un seul véhicule sur son environnement et l'information devient plus précise et plus pertinente.

Dans ce travail, nous nous intéressons au problème de la collecte et d'échange de données au sein des réseaux Ad-hoc véhiculaires. À cet effet, nous avons proposé deux protocoles de collecte de données adaptés à l'environnement autoroutier, baptisés CDGP (Clustered Data Gathering Protocol) et TCDGP (Token-based Clustered Data Gathering Protocol). Les deux protocoles sont basés sur une architecture clusterisée et permettent aux points d'accès (RSU, Road Side Unit) de collecter des données depuis le réseau routier. La plateforme développée en Java a montré via des simulations que CDGP et TCDGP ont assuré une grande fiabilité de communication inter-véhicules et une collecte de données efficace en termes de quantité de données collectées et du temps nécessaire à la collecte.

Mots clés : VANet, Collecte de données, CDGP, TCDGP.

Abstract

Over the past few years, the rapid development of wireless technologies has seen the emergence of new communication systems including Vehicular Ad-hoc Network (VANet). These networks allow the exchange of data between vehicles or between vehicles and infrastructure in order to improve the safety and the comfort on the road.

Taken individually, collected information by the various sensors of the same vehicle gives a less precise and incomplete knowledge about its environment. However, the use of information gathered by multiple vehicles makes the knowledge built up more accurate and relevant.

In this work, we focus on the data collection and sharing problem in Vehicular Ad-hoc Networks. To this end, we have proposed two data collection protocols adapted to the freeway environment, CDGP (Clustered Data Gathering Protocol) and TCDGP (Token-based Clustered Data Gathering Protocol). Both protocols are based on a clustered architecture and enable access points (RSU, Road Side Unit) to collect data on the road network. Via a set of simulations, the developed platform in Java showed that CDGP and TCDGP ensured a high reliability for inter-vehicle communications and an efficient data collection in terms of amount of collected data and the time required for that purpose.

Keywords : VANet, Data collection, CDGP, TCDGP.

Introduction générale

Contexte et problématiques

Ces dernières années, le domaine des télécommunications sans-fil a connu un développement rapide et une évolution importante, ce qui l'a rendu un domaine de recherche parmi les plus étudiés. Cette attention émergente est due principalement aux besoins actuels de la disponibilité, de l'accès et de l'utilisation des données, n'importe où et à n'importe quel moment "*Any Where and Any Time*" [1].

Ces technologies de communications sans-fil représentent aujourd'hui une des briques de base sur lesquelles se basent les systèmes intelligents et qui permettent à ces derniers d'offrir à leurs usagers une multitude de services ; les systèmes de transport intelligent, les maisons intelligentes, les voitures intelligentes, aide et surveillance médicale, etc. Parmi ces environnements "*intelligents*", on trouve les systèmes de transport intelligent (ITS, Intelligent Transportation System) dont les objectifs principaux sont (i) l'amélioration de la sécurité des usagers de la route en leur offrant par exemple : des informations sur le trafic, les accidents, les dangers, les déviations possibles, les informations météorologiques, etc. (ii) L'amélioration du confort des usagers : l'accès mobile à l'Internet, le service de chat, jeux en réseaux, etc. (iii) L'amélioration de l'efficacité globale du réseau routier, en réduisant le temps de parcours et les congestions par exemple [2].

Une des principales composantes des systèmes du transport intelligent est la communication inter-véhicules sans-fil, ou ce qu'on appelle les réseaux VANets pour Vehicular Ad-hoc Network [3]. Dans ces réseaux, les véhicules sont capables de communiquer d'une manière autonome, via un support sans-fil, entre eux ou avec les infrastructures existantes pour profiter des applications offertes.

Toutefois, vu les différentes spécificités et la nature dynamique qui caractérisent les réseaux véhiculaires, ils ne sont qu'en phase expérimentale et plusieurs problématiques doivent être traitées avant de pouvoir déployer un tel réseau, telles que l'accès au canal [4], le routage [5], la dissémination [6], la collecte et l'échange de données [7, 8] ou encore la sécurité de données échangées [9].

Dans les systèmes ITS, les véhicules sont supposés doter de plusieurs dispositifs de détection ; capteurs, caméras, radars . . . , qui leur permettent d'avoir différentes informations, en temps réel ou non. L'utilisation des informations fournies par plusieurs véhicules permet d'avoir une perception plus complète et riche sur l'environnement, le trafic routier et les événements temps réels ou non sensibles au délai.

Pour collecter ces données, il s'avère nécessaire de développer des protocoles de collecte et d'échange de données fiables et efficaces, qui prennent en compte les différentes propriétés qui caractérisent les réseaux véhiculaires telles que, la forte mobilité, le changement dynamique de la topologie, la déconnexion fréquente du réseau, la non fiabilité des communications, le délai, etc.

Objectif et contributions

Dans notre travail, on s'intéresse à étudier le problème de la collecte de données dans les réseaux véhiculaires. Dans ce contexte, nous avons pu apporter les contributions suivantes :

- ◇ Recenser et étudier les différents protocoles de collecte de données au sein des réseaux véhiculaires avant d'établir une étude comparative entre eux selon plusieurs critères de classification que l'on a pu les tirer.
- ◇ Proposer deux protocoles baptisés CDGP (Clustered Data Gathering Protocol) [10] et TCDGP (Token-based Clustered Data Gathering Protocol). Les deux protocoles sont basés sur une architecture de clusters et utilisent une version dynamique de Space Division Multiple Access (SDMA) [11] proposée elle aussi au sein de ce travail. En plus, ils assurent aussi une grande fiabilité en implémentant une technique de retransmission des données erronées.
- ◇ Développer une plateforme en Java dédiée aux protocoles de collecte de données dans VANets.

Organisation du mémoire

Le présent mémoire est scindé en cinq chapitres organisés comme suit :

- ◇ Le premier chapitre présentera les réseaux Ad-hoc de véhicules, VANet. Nous décrivons en premier lieu, le réseau sans-fil avec ses deux configurations, sans et avec infrastructures. Par la suite, nous détaillons les principaux concepts liés aux réseaux sans-fil de véhicules.
- ◇ Le deuxième chapitre est consacré à la collecte de données au sein des réseaux de capteurs sans-fil et aux différents travaux proposés dans les architectures avec des nœuds statiques, WSN pour Wireless Sensors Networks, ou des nœuds mobiles, WSN-ME pour Wireless Sensor Networks with Mobile Elements.
- ◇ Le troisième chapitre décrira les protocoles de collecte de données dans les réseaux Ad-hoc de véhicules, VANets, ainsi qu'une étude comparative entre ces protocoles selon des critères de classification.
- ◇ Dans le quatrième chapitre, nous exposerons nos deux protocoles de collecte de données proposés dans VANet ainsi qu'une étude comparative entre eux et les autres protocoles existants.
- ◇ Le cinquième chapitre analysera via un ensemble de simulations, les performances de nos deux protocoles, CDGP et TCDGP, en termes de plusieurs paramètres.

Enfin, nous terminons le mémoire par donner des conclusions sur ce travail et par exposer certaines perspectives de notre recherche.

Chapitre 1

Introduction aux réseaux véhiculaires

Actuellement, les Systèmes de Transport Intelligent (en Anglais ITS, pour Intelligent Transportation System) désignent les applications des Technologies de l'Information et de la Communication (TIC) au domaine des transports, dont les objectifs principaux sont : (i) l'amélioration de la sécurité routière, (ii) l'amélioration de l'efficacité globale du système de transport, en réduisant le temps de parcours et les congestions, et (iii) l'amélioration du confort de l'utilisateur [2].

Parmi ces technologies, on trouve les transmissions sans-fil qui font l'objet, depuis plus d'une dizaine d'années, d'une attention émergente dans le contexte du transport, de la part des entreprises, des particuliers, des industries etc. Ces technologies représentent aujourd'hui une des briques de base sur lesquelles se basent les systèmes intelligents et qui permettent à ces derniers d'offrir une multitude de services (les maisons intelligentes, les voitures intelligentes, aide et surveillance médicale, ...).

L'utilisation de ces technologies de communication sans-fil dans les ITSs et l'intégration des cartes réseaux sans-fil dans les véhicules a permis l'apparition des réseaux de véhicules sans-fil, appelé VANet (pour Vehicular Ad-hoc Network). Dans ce réseau, les véhicules forment un système distribué et sont capables de communiquer d'une manière autonome entre eux, ou avec les infrastructures existantes.

Dans ce chapitre, nous présentons les réseaux de véhicules sans-fil. Nous décrivons en premier lieu, le réseaux sans-fil avec ses deux configurations, sans et avec infrastructures. Puis, nous détaillons tous les concepts liés aux réseaux de véhicules sans-fil.

1.1 Les réseaux sans-fil

"Un réseau sans-fil (en Anglais, *Wireless network*) est un ensemble de périphériques (ordinateur, PDA, imprimante, ...), interconnectés les uns avec les autres via une liaison utilisant des ondes radio et donc sans le besoin d'une liaison filaire" [1]. En d'autres termes, c'est un système de communication qui a été conçu afin d'assurer une liaison indépendante à l'emplacement des nœuds, qui composent le réseau.

Le déploiement de tels réseaux a permis aussi un traitement de l'information grâce à des périphériques portables, créant ainsi un nouvel environnement dit "**Environnement mobile**",

qui est caractérisé par : une fréquente déconnexion, un changement dynamique de la topologie, et des sources d'énergie limitées.

Un réseau mobile sans-fil a deux modes de fonctionnement, (i) le mode avec infrastructure et (ii) le mode sans infrastructure ou mode Ad-hoc.

1.1.1 Mode avec infrastructure (cellulaire)

Le mode avec infrastructure, qui est également appelé mode BSS (Basic Service Set), est composé d'un ensemble de stations fixes interconnectées entre elles via un réseau de communication filaire ou sans-fil, appelées points d'accès (AP, Access Point) ou stations de base (BS, Base Station). Chaque AP couvre une zone géographique dans laquelle les stations mobiles peuvent émettre et recevoir des messages via les APs (voir FIGURE 1.1). On peut prendre comme exemple, les réseaux de la téléphonie mobile (GSM, UMTS, GPRS, ...) [12].

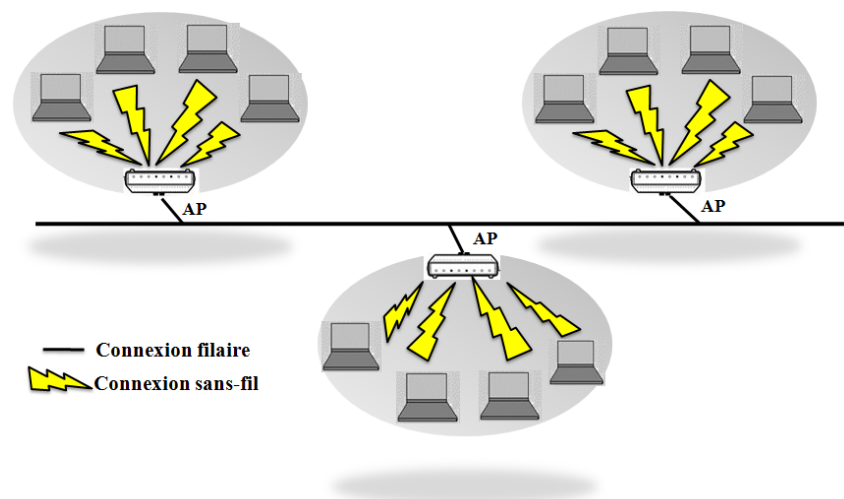


FIGURE 1.1 – Réseau en mode avec infrastructure.

Ce mode de réseau est utilisé généralement en entreprises car toutes les informations passent par le point d'accès, ce qui permet de vérifier qui accède au réseau et l'intégrité des données circulant.

1.1.2 Mode sans infrastructure (Ad-hoc)

Le mode sans infrastructure ou IBSS (Independent Basic Service Set), appelé aussi le mode Ad-hoc ou MANet (Mobile Ad-hoc Network), consiste en un ensemble de stations mobiles équipées des interfaces radio sans-fil qui leur permettent de communiquer entre elles en absence d'une infrastructure fixe préexistante [13, 14, 9](voir FIGURE 1.2).

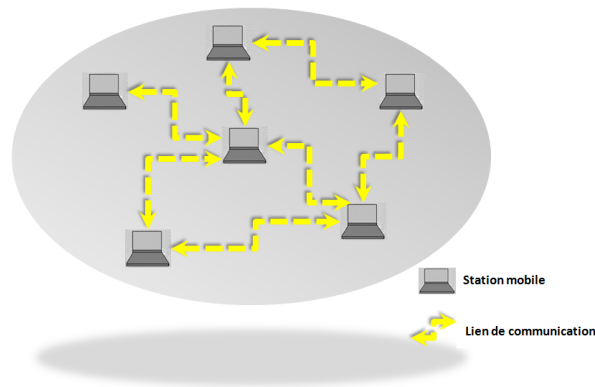


FIGURE 1.2 – Un réseau Ad-hoc mobile.

Une des principales caractéristiques des réseaux Ad-hoc mobiles est le concept de la communication multi-sauts. Contrairement aux réseaux sans-fil avec infrastructure, une station mobile peut envoyer des données vers des stations en dehors de sa portée sans passer par un AP. Ainsi, chaque station intermédiaire joue le rôle d'une station relais.

Comme mentionné précédemment, les réseaux Ad-hoc sont caractérisés par la mobilité de leurs nœuds et par l'absence d'une infrastructure centralisée. Cependant, parfois, certains types de ces réseaux peuvent avoir des configurations centralisées et nécessitent une communication avec l'infrastructure, tels que les réseaux véhiculaires et mesh, ainsi ils peuvent avoir aussi des nœuds qui sont statiques tel que les réseaux de capteurs. Parmi les types de réseaux Ad-Hoc on trouve :

1. **Les réseaux de capteurs (WSN, Wireless Sensor Networks)** : Est un réseau Ad-hoc composé d'un ensemble de dispositifs de détection, capteurs, sans-fil dispersés, d'une manière aléatoire ou déterministe, dans une zone géographique, appelée "**champ de capture**". Généralement, les nœuds capteurs qui composent un WSN sont des petits dispositifs qui ont des ressources de calcul, de stockage, de communication et d'énergie très limitées. Ces capteurs sont capables de collecter et de transmettre les données environnementales vers des nœuds puits¹ via une communication multi-sauts. Ensuite, les données collectées seront envoyées à l'utilisateur via l'Internet ou le satellite [14].

Les applications des réseaux de capteurs couvrent un spectre très large, incluant les applications militaires, médicales et environnementales [15, 16].

2. **Les réseaux véhiculaires (VANet, Vehicular Ad-hoc Network)** : Est un cas particulier des réseaux MANets dont les nœuds mobiles sont des véhicules. Ces véhicules sont équipés par des interfaces radio et peuvent communiquer entre eux ou encore avec l'infrastructure via une communication sans-fil [9]. Toutefois, la mobilité des véhicules est contrainte par la carte routière, les rues, les limites de la vitesse, ...etc.

Les réseaux véhiculaires se caractérisent par une forte densité, puisque les routes peuvent contenir des milliers de véhicules qui ont de grandes capacités de stockage, une puissance

1. Sink en anglais

de calcul importante, ainsi qu'une source d'énergie illimitée. Comme notre travail se porte sur ces réseaux, nous les présentons en détail dans la section suivante.

3. **Les réseaux de capteurs véhiculaires (VSN, Vehicular Sensor Networks)** : Sont une nouvelle architecture de réseaux de véhicules qui peut être considérée comme une fusion entre les réseaux Ad-hoc de véhicules (VANet) et les réseaux de capteurs sans-fil (WSN), donc il est possible d'utiliser plusieurs types de dispositifs de détection, capteurs et caméras, disponibles dans les nouveaux véhicules, afin de collecter des informations depuis le réseau routier (la vitesse, l'accélération, la température, vidéos, localisation ...), et améliorer la sécurité, l'efficacité et le confort des usagers de la route [17].
4. **Les réseaux mesh (WMN, Wireless Mesh Networks)** : Dans ce réseau, on trouve deux types de nœuds, les nœuds mesh qui sont des stations de base, BS, et communiquent entre eux via une communication sans-fil multi-sauts, et les nœuds clients mobiles qui peuvent communiquer entre eux ou avec les nœuds mesh (voir FIGURE 1.3). Dans ce réseau, les ressources de stockage, de calcul et d'énergie ne posent pas des problèmes et les nœuds mesh peuvent être dotés de grandes capacités de ces ressources [13].

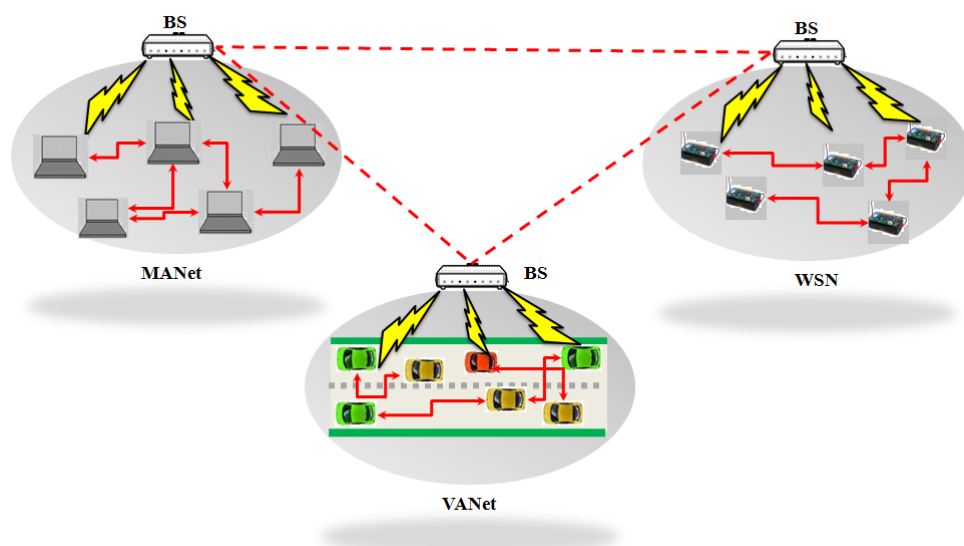


FIGURE 1.3 – Un réseau de mesh.

Ce type de réseau est appelé aussi "*le réseau des réseaux*" car les nœuds mesh sont équipés de plusieurs interfaces radio sans-fil, et peuvent donc interconnecter différents types de réseaux, MANet, VANet, WSN, GSM, ... (voir FIGURE 1.3).

Dans la TABLE 1.1, nous résumons une étude comparative entre ces différents types de réseau Ad-hoc, en se basant sur les critères de la densité du réseau, la mobilité des nœuds, les capacités des ressources des nœuds (de stockage, de calcul, et d'énergie). À noter que dans la colonne des réseaux mesh (WMN), on présente les caractéristiques des nœuds mesh seulement.

TABLE 1.1 – Propriétés des différents types du réseau Ad-hoc.

Propriétés \ Réseaux Ad-hoc	MANet	WMN	WSN	VANet
La taille du réseau	Moyenne	Moyenne	Grande	Très grande
La mobilité des nœuds	Moyenne	Sans mobilité	Faible/Forte	Forte
Limitations d'énergie	Grande	Très faible	Très grande	Très faible
La puissance de calcul des nœuds	Faible	Grande	Très Faible	Grande
La capacité de stockage des nœuds	Faible	Grande	Très Faible	Grande

1.2 Les réseaux véhiculaires

Les réseaux Ad-hoc de véhicules sont en effet une application bien spécifiée et dédiée des réseaux Ad-hoc mobiles, MANet, où les nœuds mobiles sont des véhicules équipés avec des interfaces radio sans-fil (voir FIGURE 1.4). Ces véhicules peuvent communiquer entre eux ou avec les infrastructures installées au bord de la route.

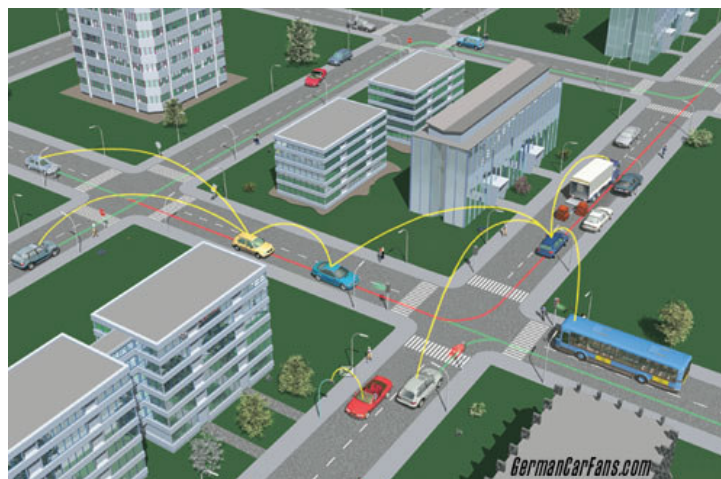


FIGURE 1.4 – Réseau Ad-hoc de véhicules.

Un véhicule dans un tel réseau, dit "*Intelligent*" (voir FIGURE 1.5), est doté de plusieurs dispositifs de détection (caméras, capteurs, radars, etc.), d'un système de localisation (GPS, Global Positioning System) ainsi que de ressources de stockage et de traitement [18].

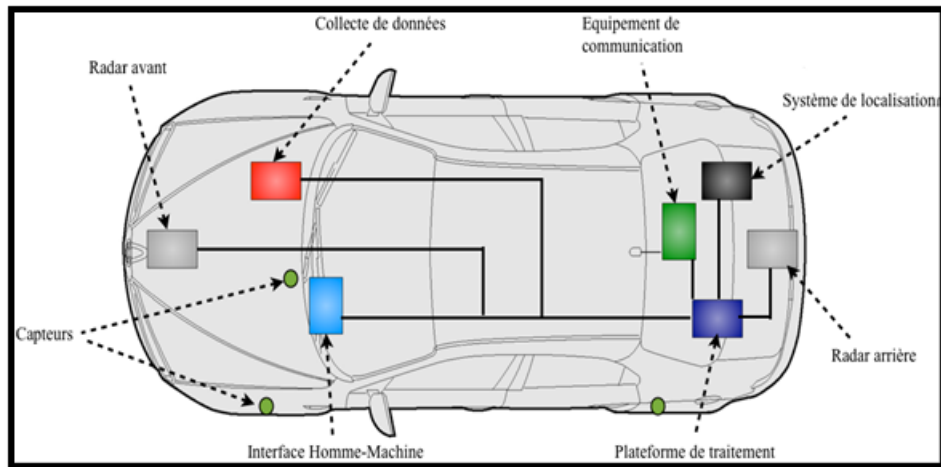


FIGURE 1.5 – Véhicule intelligent.

1.2.1 Architectures de communication

Dans les réseaux véhiculaires, les nœuds peuvent communiquer entre eux ou bien avec l'infrastructure. De plus, ces deux modes de communication peuvent être combinés afin d'obtenir une communication hybride [19].

Communications Véhicule à Véhicule (V2V)

Dans ce mode, les véhicules ne peuvent communiquer qu'entre eux. Un tel type de communication peut être utilisé pour les applications de diffusion des messages d'alerte, lorsque des événements inattendus se produisent (freinage d'urgence, accidents, ralentissement,...), ainsi que pour la conduite coopérative (la priorité au carrefour, l'assistance de changement de voie,...).

Donc, une communication V2V multi-sauts est plus appropriée qu'une communication passant par l'infrastructure car le passage par l'infrastructure augmente le délai. Par conséquent, les conducteurs ne seront pas informés en temps réel par ces événements inattendus.

Communications Véhicule à Infrastructure (V2I)

Certaines applications nécessitent une communication avec l'infrastructure et sont à base d'infrastructure telles que l'accès à Internet, l'échange de données voiture-domestique, communications de voiture-à-garage pour le diagnostic distant,..etc.

Pour cela, des points d'accès (RSU, Road Side Units) sont déployés aux bords des routes, afin de permettre aux véhicules de réaliser ces applications.

Communications hybrides

La combinaison des deux modes communications V2V et V2I engendre un mode de communication hybride, qui permet d'étendre la zone de couverture des RSUs, où l'installation des

points d'accès est difficile ou coûteuse. L'utilisation des véhicules comme relais permet de réaliser cela, ce qui rend ce mode de communication très intéressant.

1.2.2 Environnements de déploiement

En fonction de leurs spécificités (vitesse, densité du trafic routier), on distingue deux environnements de déploiement des réseaux véhiculaires : le milieu urbain et le milieu autoroutier [9].

Milieu urbain

Formé d'intersections et de points d'arrêt (feux tricolores, plaques de signalisation, . . .). Le milieu urbain se caractérise par une densité de véhicules importante, ainsi qu'une vitesse des véhicules réduite (par exemple inférieure à 60 km/h). Dans ce milieu, il y a déjà une infrastructure importante (panneaux), ce qui facilite l'ajout des équipements (RSUs) sur cette infrastructure afin de déployer les réseaux sans-fil véhiculaires (V2I).

Cependant, en raison de la forte présence des obstacles (immeubles, bâtiments, . . .), il s'agit d'un environnement qui est très perturbant pour les ondes radio, lors d'une communication V2V.

Milieu autoroutier

Le milieu autoroutier est caractérisé par une forte densité de véhicules et une impossibilité, en terme de coût, de couvrir toutes les autoroutes avec des RSUs. La vitesse élevée des véhicules dans ce milieu peut engendrer des accidents graves, donc la nécessité d'une grande fiabilité de la part des conducteurs. Cependant, en raison de l'absence d'obstacles tels que les immeubles, cet environnement est moins perturbant pour les ondes radio par rapport au milieu urbain.

1.2.3 Caractéristiques des réseaux véhiculaires

Comme mentionné précédemment, un réseau VANet n'est autre qu'une classe particulière des réseaux MANets. Cependant, les protocoles étudiés et réalisés dans MANet ne peuvent pas être directement appliqués dans VANet à cause des propriétés qui caractérisent les réseaux véhiculaires et qui doivent être pris en considération lors de la conception des protocoles dans les réseaux VANets. Parmi ces caractéristiques on peut citer [20] :

- **Grandes capacités de ressources des véhicules** : À la différence des réseaux Ad-hoc où les ressources de stockage, de calcul et d'énergie représentent des problématiques pour les nœuds mobiles, les véhicules du réseau VANet sont dotés de grandes capacités de stockage, de puissance de traitement et de sources d'énergie illimitées. De plus, les véhicules peuvent être équipés de plusieurs interfaces de communications (Wi-Fi, Bluetooth et autres).
- **Le modèle de mobilité** : Les nœuds dans un réseau Ad-hoc se caractérisent généralement par une mobilité aléatoire. Par contre, la mobilité des véhicules dans VANet est limitée par l'infrastructure routière (Par exemple, route, autoroute, les panneaux de signalisations, . . .). De plus, la mobilité des véhicules est liée directement au comportement des conducteurs et leurs réactions face à des situations différentes, telles que les accidents, l'embouteillage, . . .
- **Topologie du réseau et connectivité** : Contrairement aux réseaux Ad-hoc mobiles, les réseaux VANet sont caractérisés par une forte mobilité, puisque les nœuds sont des véhicules, par conséquent un changement très fréquent et dynamique de la topologie des réseaux véhiculaires. De plus, cette forte mobilité affecte directement la connectivité du réseau, car un véhicule peut rejoindre et quitter le réseau en un temps très court.

- **La sécurité de l'information échangée** : Une des principales applications des VANets est la prévention et la sécurité routière. En outre, la communication inter-véhicules est basée généralement sur la diffusion de données d'une source vers plusieurs destinations, ce qui rend l'aspect sécuritaire l'un des paramètres clés dans la conception de ces réseaux [21].

1.2.4 Applications des réseaux véhiculaires

Grâce à la capacité de communication de véhicules et les capteurs embarqués, plusieurs applications des réseaux véhiculaires sont envisageables. Dans [22], les auteurs ont classé ces applications en trois grandes catégories :

1. **Les applications de la sécurité routière** : Le principal objectif des systèmes de transport intelligent, ITS, est d'améliorer la sécurité routière, en rendant le trafic plus sûr et plus efficace. Cet objectif est motivé par le nombre croissant d'accidents sur les routes. Ces applications comprennent la diffusion des messages d'alertes sur l'état du trafic, des routes, de la météo, les accidents et les travaux sur la route, les limites de vitesse, la distance de sécurité. . .etc.
2. **Les applications d'aide à la conduite** : Dans le but de faciliter la conduite, les applications des ITS comprennent aussi les systèmes d'aide à la conduite telle que, aide aux dépassements, prévention de sortie des voies en virage, . . .etc.
3. **Les applications de confort** : Les réseaux véhiculaires peuvent aussi améliorer le confort des passagers, en offrant des services de communications et d'informations aux utilisateurs tels que, l'accès mobile à Internet, la messagerie, le chat inter-véhicules, les jeux en réseaux etc.

En fonction de l'application, on distingue trois types de message [9], qui peuvent être échangés dans un réseau véhiculaire :

1. **Message de contrôle (beacon)** : diffusé à intervalle régulier et contient généralement la position, la vitesse et la direction du véhicule émetteur. Ce message permet à chaque véhicule de créer une vue locale de son voisinage, de prédire et d'anticiper des situations de congestions.
2. **Message d'alerte** : généré et diffusé à intervalle régulier lors de la détection d'un événement (accident, obstacle). Ce message est de taille très réduite afin d'être diffusé le plus rapidement possible et comprend essentiellement les coordonnées du lieu de l'accident (obstacle).
3. **Autres messages** : comprend tous les messages qui ne sont pas des messages de contrôle ou d'alerte. Ces messages ne sont généralement pas diffusés à intervalle régulier, tel que l'envoi d'un courrier électronique.

1.2.5 Travaux de standardisation et de normalisation

Beaucoup d'activités de recherche et de standardisation portant sur la communication inter-véhicules ont été lancées, en Europe et dans le monde entier, par des organismes internationaux tels que IEEE (Institute of Electrical and Electronics Engineers), ISO (International Organization for Standardization), ETSI (European Telecommunications Standards Institute).

Le modèle IEEE (WAVE, Wireless Ability in Vehicular Environnements)

La norme IEEE 802.11 [23] est une norme internationale, lancée en 1997, décrivant les caractéristiques d'un réseau locale sans-fil (WLAN, Wireless Local Area Network). Cette norme offre un débit de 1 à 2 Mbps. Ensuite, des modifications ont été apportées à cette norme initiale, afin d'améliorer le débit de communication et d'assurer une meilleure sécurité, c'est le cas des normes 802.11a, 802.11b, 802.11g, 802.11n etc.

En 2002, American Society for Testing and Materials (ASTM) a dédié une bande de fréquences spécifique aux communications inter-véhicules, c'est le DSRC (Dedicated Short Range Communication) [24] qui œuvre dans la bande de fréquences 5.9GHz (Europe et Etats-Unis) ou 5.8GHz (Japon). Cette bande de fréquences a été reprise ensuite, en 2003, par IEEE afin de définir un standard dédié aux communications inter-véhicules, en ajoutant la norme 802.11p [25]. En effet, La norme IEEE 802.11p n'est chargé de définir que les protocoles de la sous couche MAC et la couche physique. Pour cela, elle est basée sur la couche physique de la norme IEEE 802.11a, ainsi que sur la couche MAC de la norme IEEE 802.11e afin de répondre aux caractéristiques des VANets.

En complément, pour définir les protocoles des couches supérieures se situant entre la couche liaison et la couche application, IEEE a défini la famille des protocoles 1609 [26], appelée aussi WAVE (Wireless Ability in Vehicular Environnements)(voir FIGURE 1.6). Ce standard est structuré en quatre composantes :

1. IEEE 1609.1 - WAVE Resource Manager : pour la gestion des ressources,
2. IEEE 1609.2 - WAVE Security Services for Applications and Management Messages : pour la sécurisation des messages,
3. IEEE 1609.3 - WAVE Networking Services : pour les services de niveau réseau et transport incluant l'adressage, le routage,
4. IEEE 1609.4 - WAVE Multi-Channel Operation : pour la coordination et la gestion des canaux de DSRC.

Ce modèle 802.11p/WAVE spécifient deux piles protocolaires. Une pile pour les applications de sécurité routière et une autre pour les autres catégories d'applications.

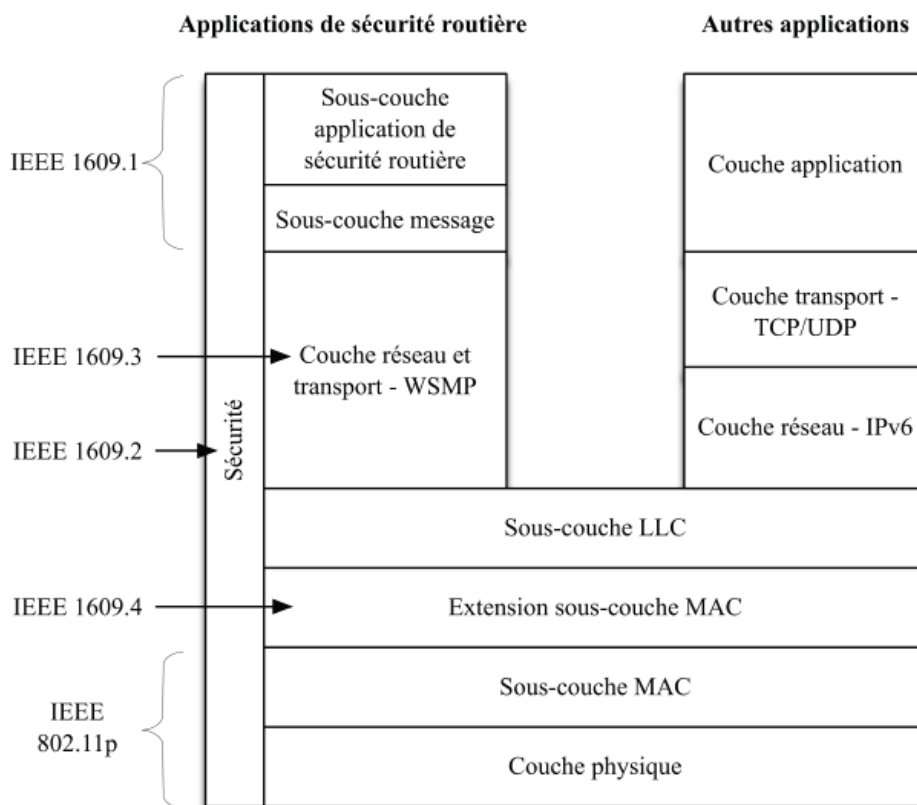


FIGURE 1.6 – Le modèle DSRC/WAVE.

ISO : TC204/WG16 - CALM

Au niveau mondial, le groupe de travail WG16 dans l'organisme ISO a développé une plateforme logicielle embarquée dans les véhicules dite CALM (Continuous Air-interface, Long and Medium range) [27], dont l'objectif est d'assurer une interface entre plusieurs technologies de communication 2G, 3G, DSRC, ainsi que le handover entre ces technologies. Par exemple, CALM choisira automatiquement de passer du wifi, du GSM ou du DSRC en fonction de la disponibilité des réseaux et du message à transmettre.

ETSI : TC ITS

Au niveau Européen, ETSI [28] a créé un comité technique TC ITS, afin de développer des standards et spécifications pour les ITS. Le comité a mis en place un plan de route pour les prochaines années, afin de produire un ensemble de standards allant de l'architecture de communication à la spécification de protocoles, et il est organisé en 5 groupes de travail : WG1 - User Application requirements, WG2 -Architecture and cross layer issues, WG3 - Transport and Network, WG4 - Media and related issues et le WG5 - Security. Dans le WG3 par exemple, ils s'intéressent à la spécification des protocoles d'adressage et de routage géographiques. La figure 1.7 schématise ce modèle.

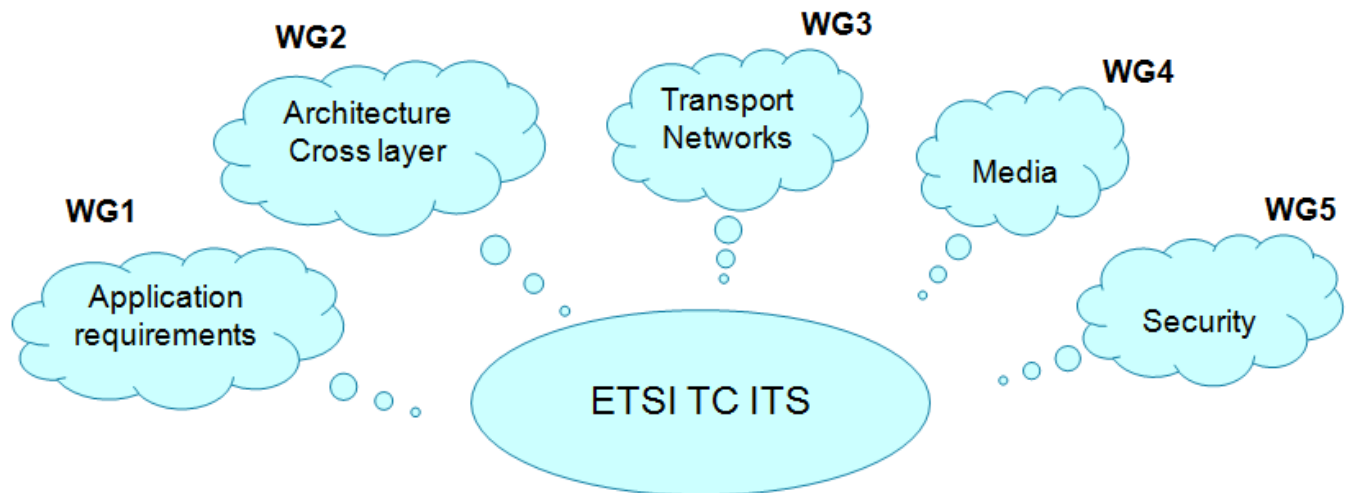


FIGURE 1.7 – ETSI : TC ITS.

1.2.6 Projets et consortiums

Plusieurs travaux de recherche, de développement et de standardisation des communications inter-véhicules ont été lancés dans le contexte des réseaux véhiculaires. Ces travaux visent principalement à améliorer la sécurité routière et le confort des usagers de la route. Nous illustrons dans la TABLE 1.2 les principaux projets qui ont été lancés dans ce contexte.

TABLE 1.2 – Projets et consortiums dans VANet.

Projets et consortiums	Pays/ continent	Initiateur/ contributeur	Objectifs
FleetNet (Internet on the road) [29]	Allemagne/ Europe	Un consortium de six industriels et trois universités	Développer une plate-forme de communication et standardiser toutes les solutions proposées
Car2Car [30]	Europe	Six constructeurs automobiles	Améliorer la sécurité routière.
CarTalk2000 [31]	Europe	Des constructeurs automobiles et des universités.	Développer un système coopératif d'aide à la conduite.
NOW (Network-on-Wheels) [32]	Allemagne /Europe	Des constructeurs automobiles, des acteurs de télécommunication et des universitaires.	Développer des protocoles de communication et d'algorithmes de sécurité de données.
AHS (Automated Highway System)	Japon	Le gouvernement Japonais.	Concevoir un système d'autoroutes automatisées permettant une conduite autonome.
NAHSC (National Automated Highway System Consortium)	USA	Le gouvernement Américain.	Evaluer tous les concepts proposés dans le contexte des systèmes d'autoroutes automatisés.
ITS America (Intelligent Transportation Society of America) [2]	USA	Un groupe d'industriels, d'organismes publics, et des universités	Le développement et le déploiement des applications ITS . partout aux États-Unis
Safety Pilot [33] Transportation Society	USA	Le ministère des Transports américain et l'université du Michigan	Accroître la sécurité des conducteurs

1.3 Conclusion

Depuis quelques années, le développement rapide des technologies telles que la communication sans-fil a favorisé une grande évolution des systèmes de transport intelligent dans le but (i) d'améliorer la sécurité routière en rendant le déplacement des usagers plus sécurisé, (ii) améliorer l'efficacité du système de transport en diminuant le temps de parcours et les congestions par exemple, et (iii) améliorer le confort des usagers.

Nous avons essayé dans ce chapitre de mieux comprendre le concept des réseaux véhiculaires ou ce qu'on appelle les réseaux Ad-hoc mobiles de véhicules (VANets), qui sont une classe dédiée des MANets, et de présenter les caractéristiques, les modes de communications, les applications des réseaux de véhicules et toutes les notions de base liées à ce type de réseaux.

Par ces connaissances de base sur les réseaux Ad-Hoc, nous traiterons plus facilement les protocoles de collecte au sein des réseaux de capteurs sans-fil dans le chapitre suivant.

Chapitre 2

La collecte de données dans les réseaux de capteurs sans-fil

La problématique de la collecte et l'échange de données, depuis l'environnement physique, a été pratiquement traitée dans tous les types des réseaux Ad-hoc, tels que dans les réseaux mesh [34, 35] et dans les réseaux véhiculaires [7, 8]. Cependant, cette problématique a été plus visée et étudiée dans les réseaux de capteurs sans-fil (WSNs, Wireless Sensors Networks), par rapport aux autres types de réseaux Ad-hoc [36, 37, 38, 39]. Cette attention est justifiée par le fait que l'objectif principal du déploiement d'un WSN est de collecter des données depuis l'environnement physique, sur lequel les capteurs sont déployés, pour faciliter le suivi et le contrôle à distance de ce dernier avec précision.

En effet, un WSN est composé d'une collection d'unités appelées nœuds capteurs. Ces capteurs sont capables de communiquer entre eux, via une communication sans-fil, selon une topologie fixe ou mobile du réseau, afin de collecter et d'acheminer les données vers l'utilisateur final (l'observateur). Pour collecter les données, on trouve dans la littérature deux architectures différentes. La première est l'architecture classique des WSNs, où les nœuds capteurs et collecteurs, sinks, sont statiques. La deuxième consiste à utiliser des nœuds collecteurs mobiles, WSN-ME (Wireless Sensor Networks with Mobile Elements).

L'objectif de ce chapitre est de présenter le processus de la collecte de données dans WSN et dans WSN-ME, ainsi que les différents travaux qui ont été proposés dans ce contexte.

Le reste de ce chapitre est organisé comme suit : dans la section suivante, nous introduisons les réseaux de capteurs avec des nœuds statiques, ainsi que leurs caractéristiques. La section 2 est dédiée aux protocoles de routage et d'échange de données dans les réseaux de capteurs avec des nœuds statiques. Ensuite, nous présentons les réseaux de capteurs avec des éléments mobiles (WSN-ME) dans la section 3. Dans la section 4, nous abordons la collecte de données au sein des WSN-MEs.

2.1 Les réseaux de capteurs sans-fil classiques (WSNs)

Un réseau de capteurs sans-fil est constitué de milliers de nœuds appelés nœuds capteurs, qui sont déployés dans une zone géographique, appelée "**Zone de captage**" pour la surveiller. Ces

nœuds captent et collectent des données depuis la zone de captage, avant de les traiter et de les transmettre vers une station de base, nœud puits, via une communication multi-saut. Ensuite, les données collectées sont envoyées vers l'utilisateur final par le nœud puits via Internet ou le satellite (voir FIGURE 2.1) [40].

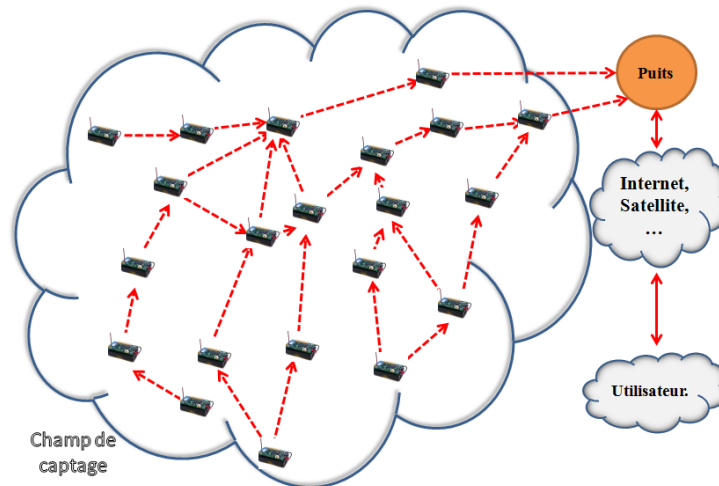


FIGURE 2.1 – Un réseau de capteurs sans-fil.

Dans un WSN, les capteurs sont des petits dispositifs électroniques à faible coût. Cela implique forcément qu'ils ont des capacités réduites (capacité de calcul, de stockage, d'énergie, portée de communication). De plus, grâce au principe d'auto-organisation des réseaux Ad-hoc, ces capteurs peuvent être déployés sans avoir besoin d'une infrastructure lourde [40].

Un WSN n'est rien d'autre qu'une application des réseaux Ad-hoc. Ces deux types de réseaux ont de nombreux points en commun :

- Architecture décentralisée,
- réseaux sans infrastructure,
- l'utilisation des ondes radios pour communiquer,
- le concept d'auto-organisation,
- ressources limitées,
- ...

Cependant, les principaux points de différences entre ces deux types de réseaux sont [41] :

- La forte densité des réseaux de capteurs sans-fil,
- les nœuds capteurs ont une durée de vie très limitée,
- la communication entre les nœuds se fait par diffusion et non pas point à point,
- le déploiement des capteurs peut être aléatoire ou déterministe,
- ...

2.2 La collecte de données dans WSNs

Une des principales fonctionnalités d'un réseau de capteurs est la collecte de données. Cette fonctionnalité doit prendre en considération toutes les spécificités et les caractéristiques de ce

type de réseau, afin d'assurer une meilleure performance : la durée de vie, la fiabilité, le temps de réponse, . . . etc. Pour cela, plusieurs recherches sont orientées vers les protocoles d'échange et de routage de données entre les nœuds capteurs [42, 43, 44], dont certaines entre elles sont présentées par la suite.

2.2.1 Le protocole Directed Diffusion (DD)

L'un des protocoles les plus célèbres pour la dissémination de données dans les réseaux de capteurs statiques. Le protocole DD [42] est un protocole de propagation de données, qui comprend trois grandes phases :

1. **Phase de propagation d'intérêt** : Durant laquelle, le nœud puits initié la propagation d'un intérêt, qui est un message se référant à un événement bien spécifié, ainsi que le débit d'information désiré, qui sera diffusé et inondé dans tout le réseau (voir FIGURE 2.2).

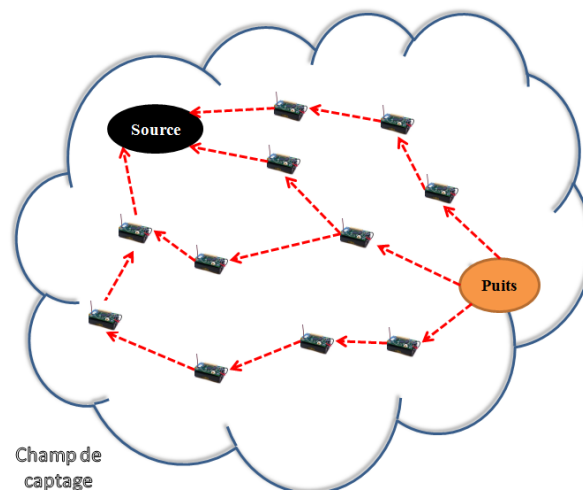


FIGURE 2.2 – Phase de propagation d'intérêt.

2. **Phase d'établissement des gradients** : Dès que l'intérêt est inondé dans tout le réseau et atteint les sources ciblées, les nœuds capteurs commencent la collecte de données depuis les nœuds sources. En conséquence, plusieurs chemins de nœuds sources vers le nœud puits sont mis en place (voir FIGURE 2.3).

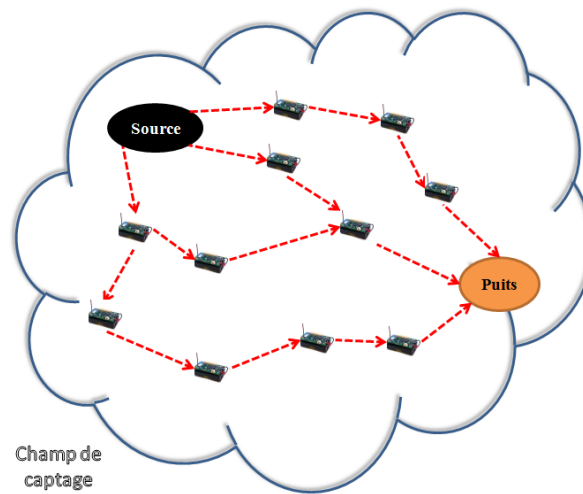


FIGURE 2.3 – Phase d'établissement des gradients.

- 3. Phase de renforcement des chemins :** Lorsque le nœud puits reçoit les premiers paquets de données, il renforce le chemin vers le nœud voisin émetteur, en augmentant le débit. À noter que, le renforcement ne doit pas s'arrêter au niveau des nœuds voisins du puits, mais doit se propager jusqu'aux sources. Pour cela, lorsqu'un nœud reçoit un message de renforcement, il doit aussi, à son tour, renforcer un de ses voisins (voir FIGURE 2.4).

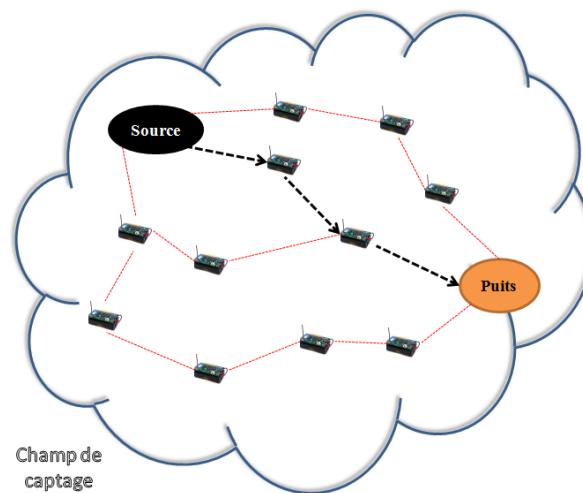


FIGURE 2.4 – Phase de renforcement des chemins.

2.2.2 Le protocole SPIN (Sensor Protocols for Information via Negotiation)

Le but du protocole SPIN [43] est de réduire la surcharge du réseau et d'éviter les problèmes d'inondation. Pour cela, il est basé sur un modèle de négociation avant de diffuser une information dans le réseau de capteurs. Ce protocole comprend essentiellement trois étapes et utilise trois types de messages, ADV/REQ/DATA. Durant la première étape, si un nœud veut émettre une

donnée, il envoie un message ADV à tous ces voisins, qui comprend une brève description sur la donnée à émettre. Puis, dans la deuxième étape, seuls les voisins qui sont intéressés par cette donnée vont répondre, en diffusant un message REQ. Ensuite, le nœud qui a débuté la communication envoie le message DATA vers tous les nœuds qui ont envoyé un message REQ (voir FIGURE 2.5).

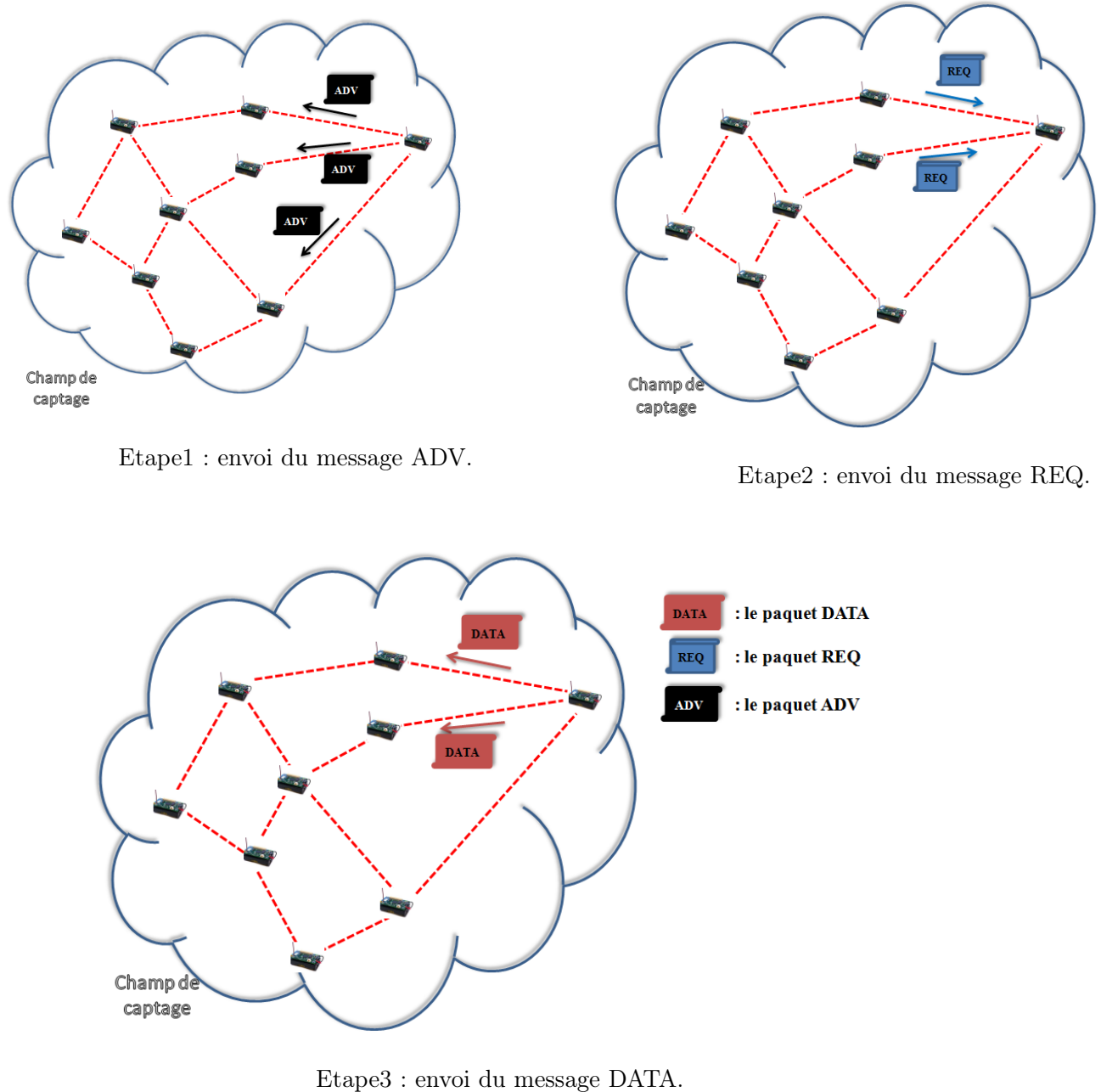


FIGURE 2.5 – Le déroulement du protocole SPIN : Etape 1, Etape 2 et Etape 3.

De plus, ce protocole permet aussi d'économiser la consommation d'énergie des nœuds cap-

teurs. Par exemple, un nœud qui trouve que le niveau de sa batterie est faible, il peut ne pas répondre à un message ADV.

2.2.3 Le protocole CBRP (Cluster Based Routing Protocol)

Basé sur une architecture en clusters, le protocole CBRP [44] décompose l'ensemble des nœuds du réseau en plusieurs groupes, clusters, et sélectionne dans chaque groupe un nœud comme ClusterHead (CH). Dans ce protocole, une table de voisins est utilisée au niveau de chaque nœud et indique pour chaque voisin l'état de lien (uni ou bidirectionnelle) et le statut du voisin (membre ou CH de cluster). De plus, chaque CH maintient des informations sur ses nœuds membres et des informations sur les clusters adjacents.

Si un nœud source veut envoyer des données à un nœud destination, il diffuse une requête demandant un chemin vers les CHs des clusters voisins. Puis, chaque CH vérifie l'existence du nœud destination dans son cluster. Ensuite, le CH du cluster qui contient le nœud destination informe ce dernier. Dans le cas contraire, la requête est rediffusée vers les CHs des clusters voisins. Quand la requête est reçue par sa destination, elle répond par l'envoi du chemin qui a été sauvegardé dans le paquet de la requête. Dans le cas où le nœud source ne reçoit pas de réponse après une certaine période, il envoie une nouvelle requête de demande du chemin.

2.3 Les réseaux de capteurs sans-fil avec des éléments mobiles (WSNs-ME)

Dans WSN, les nœuds sont capables de communiquer entre eux à travers une communication multi-saut. De plus, ces nœuds sont supposés statiques et aucune mobilité n'est considérée. Cependant, des nouvelles architectures des WSNs basées sur des éléments mobiles ont été proposées (WSN-ME, Wireless Sensor Networks with Mobile Elements), où la mobilité de certains éléments est exploitée afin de traiter le problème de la collecte de données au sein des WSNs [45, 46].

2.3.1 Les avantages et les défis des WSNs-ME

Similaire aux travaux qui ont été proposés dans Mobile Ad-hoc Networks (MANets) [5] et Delay Tolerant Networks (DTNs) [47], la mobilité a été aussi introduite dans WSNs [48] pour plusieurs raisons [37] :

- **Connectivité** : Comme certains nœuds sont mobiles, la forte densité du WSN peut ne pas être une exigence et les éléments mobiles peuvent visiter les régions isolées, et faire face aux contraintes de la connectivité du réseau.
- **Coût** : Le coût du réseau de capteurs est réduit puisque un petit nombre de nœuds capteurs peuvent être déployés. De plus, bien que l'ajout de la mobilité aux nœuds peut être coûteux, dans plusieurs cas, il est possible d'exploiter les éléments mobiles déjà présents dans la zone de captage, comme les trains, les bus et les voitures pour servir comme des collecteurs mobiles.
- **Fiabilité** : Dans WSN traditionnel, la communication entre les nœuds capteurs est souvent multi-saut, ce qui peut engendrer des problèmes de collisions, d'interférences et de perte de messages. Cependant, dans un WSN-ME, les éléments mobiles peuvent visiter les nœuds

capteurs et collecter les données via une transmission d'un seul saut, ce qui augmente la fiabilité de communication.

- **L'efficacité énergétique** : Dans WSN, les messages sont générés par les nœuds capteurs et collectés par le nœud puits. Par conséquent, les nœuds les plus proches du nœud puits souffrent souvent de grande consommation d'énergie, puisque tous les messages passent par eux. Dans WSN-ME, les nœuds mobiles peuvent réduire ce problème d'énergie car ils peuvent visiter les différentes régions du réseau et donc réduire la consommation d'énergie même sous les conditions d'une forte densité [49].

Cependant, la mobilité dans WSN introduit aussi plusieurs défis qui ne sont pas présents dans le traditionnel WSN, tels que :

- **La détection du contact** : Vu que la communication est possible seulement quand les nœuds sont à la portée les uns des autres, il est nécessaire donc de pouvoir détecter correctement et efficacement la présence du nœud mobile.
- **La gestion de l'énergie** : Dans certain cas, il est possible d'exploiter la connaissance sur le modèle de mobilité, afin d'économiser la consommation d'énergie et d'optimiser la détection du nœud mobile. Donc, si l'instant de visite est connu ou bien peut être prédit avec une certaine précision, les nœuds capteurs peuvent se réveiller seulement quand le nœud mobile sera à leurs portées.
- **La fiabilité du transfert de données** : Comme les nœuds mobiles sont disponibles pour une courte durée, il est nécessaire de maximiser le nombre de messages correctement transmis vers le nœud puits.
- **Le contrôle de la mobilité** : Le chemin et la vitesse des nœuds mobiles doivent être définis afin de maximiser les performances du réseau et de définir l'ensemble des nœuds capteurs qui peuvent être visités par les nœuds mobiles.

2.3.2 Les composants des WSNs-ME

Un WSN-ME peut être constitué de trois types de nœuds :

1. **Nœuds capteurs réguliers** : Sont les sources d'informations dans le réseau. Ils peuvent aussi relayer les messages entre les nœuds voisins.
2. **Puits (Station de base)** : Sont les nœuds destinataires d'informations. Ils collectent les données depuis les nœuds capteurs soit directement (ils visitent les nœuds capteurs) soit indirectement (via des nœuds intermédiaires).
3. **Nœuds spécifiques** : Exécutent des tâches bien spécifiées, ils peuvent jouer le rôle des collecteurs intermédiaires (entre les nœuds capteurs et les nœuds puits) ou des passerelles mobiles. Ces nœuds ne sont ni les sources ni les destinataires des messages, mais exploitent leur mobilité afin de réaliser des tâches bien spécifiées.

A noter que, la mobilité peut toucher les trois composants du réseau. Par exemple, les nœuds capteurs peuvent être mobiles et les nœuds puits statiques et vice versa. Par conséquent, un WSN-ME peut être défini comme étant un réseau où au moins un des trois types de nœuds est mobile.

2.3.3 Les architectures des WSNs-ME

Selon les différents types des éléments mobiles, on distingue trois architectures des WSNs-ME :

1. **Architecture avec des nœuds collecteurs mobiles** : Sont des éléments mobiles visitant le réseau pour collecter les données générées par les nœuds capteurs. Ces collecteurs mobiles peuvent être des nœuds puits mobiles (voir FIGURE 2.6 (a)) ou bien des nœuds relais mobiles (voir FIGURE 2.6 (b)).

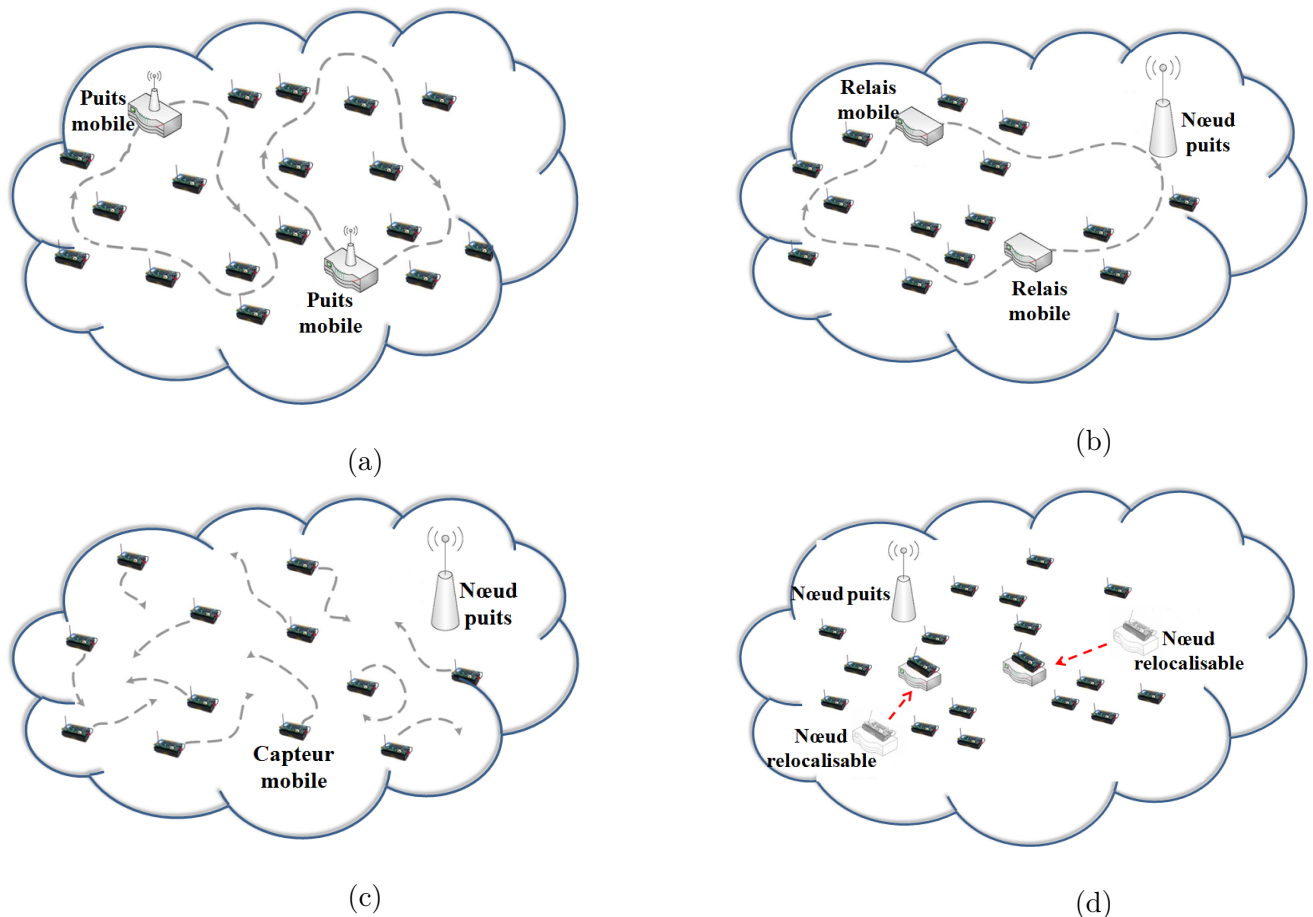


FIGURE 2.6 – Architectures des WSN-ME.

2. **Architecture avec des nœuds capteurs mobiles** : À la différence de l'architecture avec des nœuds collecteurs mobiles où seulement les nœuds puits ou les nœuds relais sont mobiles, dans cette architecture les nœuds capteurs sont mobiles et peuvent être les sources et les relais de données dans le réseau. Quand un nœud mobile est à la portée du nœud puits, il lui transmet ses données agrégées avec celles collectées par les autres nœuds capteurs dans le réseau (voir FIGURE 2.6 (c)).
3. **Architecture avec des nœuds relocalisables** : Sont des nœuds mobiles qui changent leurs localisations dans le réseau afin de transmettre les données des nœuds capteurs vers les nœuds puits. À la différence des nœuds collecteurs mobiles, ces nœuds jouent le rôle des nœuds relais statique, c.à.d. ils ne portent pas des données lorsqu'ils se déplacent (voir FIGURE 2.6 (d)).

Des algorithmes de placement des nœuds relocalisables ont été proposés dans [50, 51], dans

le contexte d'améliorer la connectivité et l'efficacité énergétique.

2.4 La collecte de données dans WSNs avec des éléments mobiles

Dans cette section, nous présentons le processus de la collecte de données dans les réseaux de capteurs avec des éléments mobiles, WSN-ME, ainsi que les différentes phases qui le composent. Comme les nœuds capteurs, statiques ou mobiles, ne peuvent communiquer avec les nœuds collecteurs mobiles sauf s'ils sont en contact, c.à.d. dans leurs portées de communication, la première phase du processus de la collecte est la phase de découverte, qui permet aux nœuds capteurs de détecter la présence des nœuds mobiles dans leurs portées de communication. Après la découverte des éléments mobiles, et dans la phase de transfert de données les nœuds capteurs échangent leurs données avec les éléments mobiles découverts, et localisé d'un seul saut. La dernière phase de routage de données consiste à acheminer les données via une communication multi-sauts, entre les nœuds capteurs et les éléments mobiles si la distance entre eux, dépasse un seul saut (voir FIGURE 2.7).

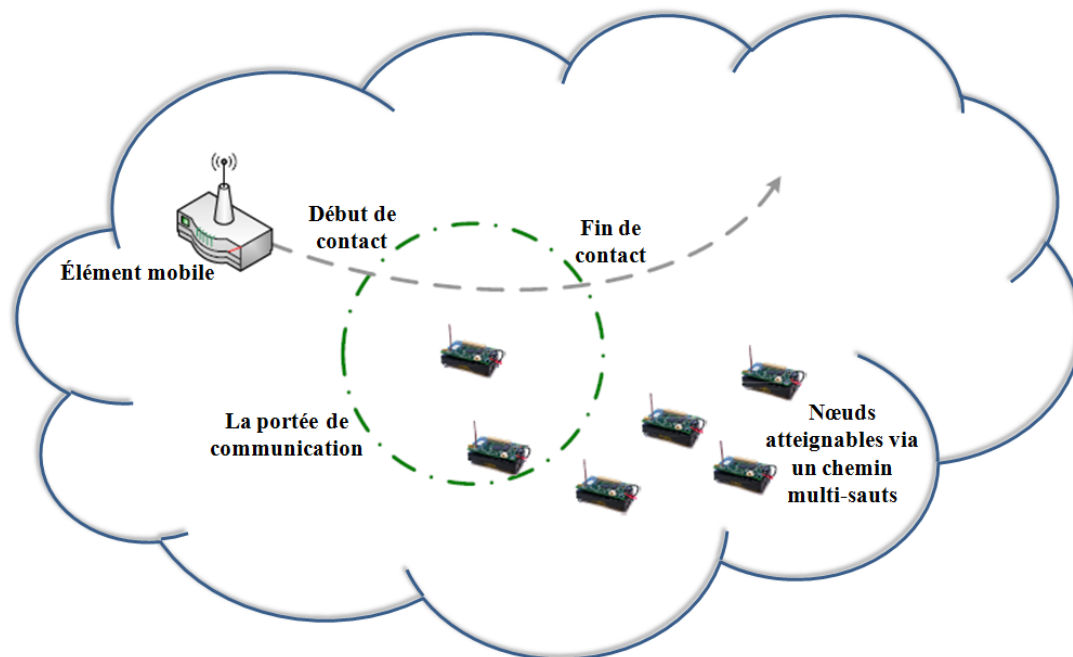


FIGURE 2.7 – Scénario de la collecte de données dans WSN-ME.

2.4.1 Taxonomie des protocoles de la collecte de données dans WSN-ME

Sur la base de la discussion ci-dessus, le processus de la collecte est composé de trois phases principales : la découverte des éléments mobiles, le transfert de données et le routage de données vers les MEs. La figure 2.8 illustre une taxonomie des techniques utilisées dans chaque phase du processus de la collecte dans WSN-ME. Dans les sections suivantes, nous allons détailler quelques travaux proposés dans la littérature pour chaque phase du processus.

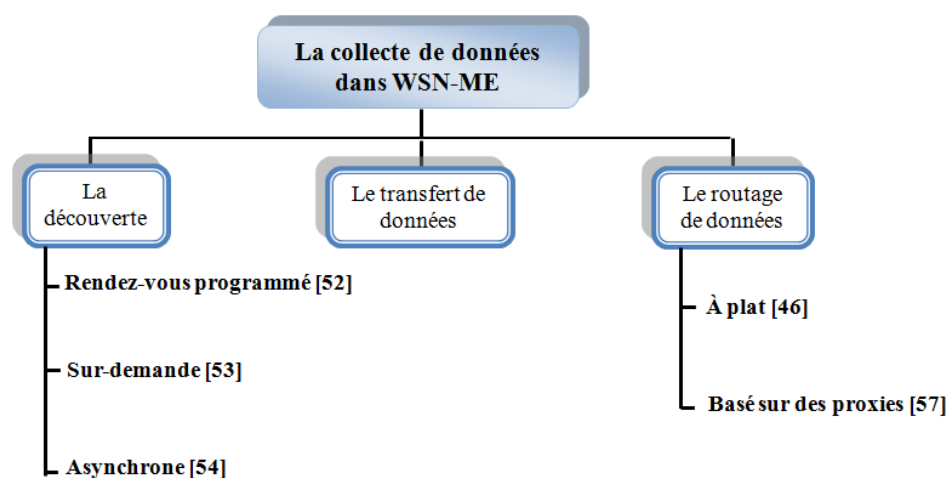


FIGURE 2.8 – Taxonomie des approches de collecte de données dans WSN-ME.

2.4.2 La découverte de l'élément mobile

C'est la première phase du processus de la collecte, qui permet aux nœuds capteurs de détecter la présence des éléments mobiles dans leurs portées de communication. On distingue trois types de protocoles de découverte : rendez-vous programmé, sur-demande, et asynchrone.

Les protocoles du rendez-vous programmé (planifié)

Ces protocoles supposent que les nœuds capteurs et les éléments mobiles se mettent d'accord sur l'instant auquel ils seront en contact. Cela peut se produire lorsque les nœuds capteurs savent exactement quand l'élément mobile entrera dans la zone du contact, et peuvent donc se réveiller à des moments prédéfinis. Dans [52], les éléments mobiles sont supposés être à bord des bus de transport public qui visitent les nœuds capteurs selon un horaire bien défini. Ensuite, les nœuds capteurs définissent seulement les moments où ils doivent se réveiller.

Les protocoles du rendez-vous programmé sont simples à mettre en œuvre car ils sont basés seulement sur l'échange des instants de visites. Cependant, ces protocoles nécessitent une synchronisation stricte entre les nœuds capteurs et mobiles, ce qui limite l'application de ces protocoles en pratique.

Les protocoles sur demande

Sont fondés sur l'idée que les nœuds capteurs peuvent se réveiller à la suite d'un processus initié par l'élément mobile. Dans [53], les nœuds utilisent plusieurs interfaces radios : une interface à longue portée et haute puissance, utilisée pour la communication et l'échange de données tandis qu'une interface de courte portée est utilisée pour réveiller les nœuds capteurs. Ces protocoles minimisent significativement la consommation d'énergie. De plus, ils offrent une détection très efficace des éléments mobiles. Cependant, à cause de la courte portée de l'interface radio, dans certains cas seulement un petit nombre des nœuds capteurs sont informés, puisque la distance entre les éléments mobiles et les nœuds capteurs n'est pas toujours courte.

Les protocoles asynchrone

Définissent un modèle Veille / Réveil, de telle sorte que les nœuds peuvent communiquer sans être obligés à se mettre d'accord sur leurs instants d'activation. Ces protocoles sont basés généralement sur l'écoute périodique [54, 37], où les éléments mobiles diffusent périodiquement des messages de découverte des nœuds capteurs et les nœuds capteurs se réveillent et écoutent périodiquement le canal, s'ils ne détectent pas de message de découverte ils peuvent dormir, sinon ils commencent le transfert de données vers l'élément mobile. L'inconvénient de ces protocoles c'est que les nœuds capteurs peuvent recevoir des messages de découverte lorsqu'ils sont en état Veille.

2.4.3 Le transfert de données

Après la détection de la présence de l'élément mobile, la phase de transfert de données entre l'élément mobile et ses voisins situés à un seul saut commence. Lors du transfert, le processus de communication est affecté non seulement par les conditions du canal (interférence, bruit, . . .), mais aussi par la distance entre la source et la destination, qui change avec le temps selon la vitesse de l'élément mobile.

Des expériences menées dans [55] ont montré que la quantité de données collectées dépend fortement de la vitesse de déplacement de l'élément mobile. C'est-à-dire, une grande quantité de données peut être collectée si l'élément mobile se déplace lentement. Par contre, une petite quantité de données sera collectée si la vitesse est élevée.

Dans [56], les auteurs ont étudié la perte de messages, en fonction de la distance qui sépare l'élément mobile du nœud capteur pendant le transfert de données. Pour cela, ils ont considéré deux vitesses différentes : 1 m/s si les éléments mobiles sont des piétons et de 20 à 40 Km/h si les éléments mobiles sont des véhicules. Ils ont constaté que la perte des messages est proportionnelle à la distance qui sépare l'élément mobile du nœud capteur.

2.4.4 Le routage de données

Il existe deux grandes classes de techniques de routage nommées routage à plat et routage basé sur des proxys. Dans les deux cas, des chemins de communication entre les nœuds capteurs et l'élément mobile sont sélectionnés et modifiés en fonction de la mobilité de l'élément mobile.

Le routage à plat

Le routage à plat est caractérisé par le fait que tous les nœuds du réseau jouent le même rôle et aucun nœud ne joue un rôle spécial.

- **Le protocole OLSR (Optimized Link State Routing)** : Une modification du protocole de routage proactif proposé pour les réseaux Ad-hoc OLSR est proposée dans [45] pour un réseau de robots, où tous les nœuds sont mobiles. L'idée générale de cette solution est d'exploiter deux repères liés à la mobilité des nœuds afin de choisir les routes les plus stables. Le premier repère est lié à la direction du mouvement des nœuds alors que le deuxième est lié à la position des nœuds. Ensuite, les routes sont sélectionnées à l'aide d'une métrique de stabilité, qui estime combien de temps un lien va durer. La stabilité

est obtenue en fonction de la direction, ou bien en fonction de la direction et la position quand les deux sont disponibles.

- **Directed Diffusion (DD)** : La version initiale du DD a été conçue pour les réseaux de capteurs statiques, par conséquent, il n'est pas approprié pour WSN-ME. Une modification de ce protocole est proposée dans [55], où deux mécanismes sont introduits pour faire face à la mobilité des nœuds. Le premier donne une priorité plus élevée à la propagation d'intérêt provenant de l'élément mobile, par rapport aux nœuds statiques. Cela permet de réduire la propagation inutile des données et donc de ne pas surcharger le réseau. Le deuxième mécanisme exploite l'acquiescement des messages envoyés par l'élément mobile, afin de détecter le départ de l'élément mobile quand il quitte la portée de communication des nœuds capteurs.
- **Le protocole Weighted Entropy Data diSsemination (WEDAS)** : Introduit dans [46], s'adresse aussi au routage de données dans WSN-ME. Des nœuds intermédiaires formant le chemin de routage de données vers l'élément mobile sont choisis selon deux paramètres : l'énergie restante des nœuds capteurs et la position de l'élément mobile. Pour cela, un problème d'optimisation est formulé afin de sélectionner des nœuds intermédiaires possédant plus d'énergie.

Le routage basé sur des proxys

Les protocoles de routage basés sur des proxys, comme dans [57, 58], sélectionnent certains nœuds (statiques), nommés proxys, Gateways, passerelles ou points de rendez-vous, qui sont en charge de collecter les données depuis des régions spécifiques dans le réseau. Ensuite, ils transmettent ces données vers l'élément mobile quand il sera dans leurs portées de communication.

- **Le protocole Two-Tier Data Dissemination (TTDD)** : Proposé dans [57] se base sur une structure en grille pour envoyer les données vers l'élément mobile. Les intersections de lignes de la grille sont appelés points de passage, tandis que les carrés sont considérés comme des cellules (voir FIGURE 2.9). Lors de l'envoi d'un message, le nœud source devient tout d'abord un point de passage, puis il commence à envoyer des messages d'annonces vers ces quatre intersections adjacentes. La localisation des points de passage est exploitée afin de réduire la propagation des messages dans le réseau, puisque seulement les nœuds proches de l'intersection destinée traitent le message et répètent la même procédure. Cette propagation récursive élit des nœuds de dissémination, qui agissent comme proxys et établissent des chemins de routage vers l'élément mobile. L'élément mobile initie le processus de la collecte en diffusant un message de requête, les nœuds de dissémination propagent ensuite la requête le long de la grille et collectent les données depuis les nœuds capteurs avant de les envoyer vers l'élément mobile.

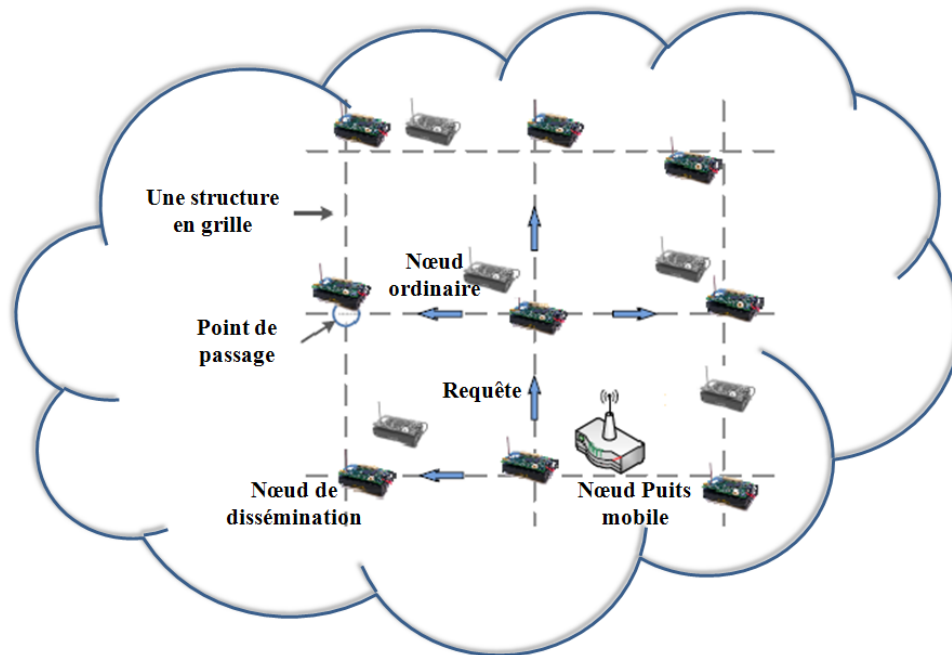


FIGURE 2.9 – Le Two-Tier Data Dissemination (TTDD).

- **Scalable Energy-efficient Asynchronous Dissemination protocol (SEAD)** : Une approche similaire à TTDD est introduite dans [58], qui construit et maintient un arbre de dissémination (d-tree). L'élément mobile peut rejoindre l'arbre en diffusant une requête vers le plus proche voisin dans le réseau. Ce dernier deviendra le nœud d'accès pour l'élément mobile c.à.d. il agit comme un nœud proxy entre l'élément mobile et le reste du réseau (voir FIGURE 2.10). Après la réception d'une requête, le nœud d'accès commence à construire le d-tree récursivement, puis le d-tree est recalculé dynamiquement quand l'élément mobile se déplace dans le réseau. Lorsque la distance entre l'élément mobile et son nœud d'accès dépasse un seuil donné, un autre nœud d'accès est sélectionné ainsi que son d-tree est construit. Ce seuil est un compromis entre la latence et la consommation énergétique.

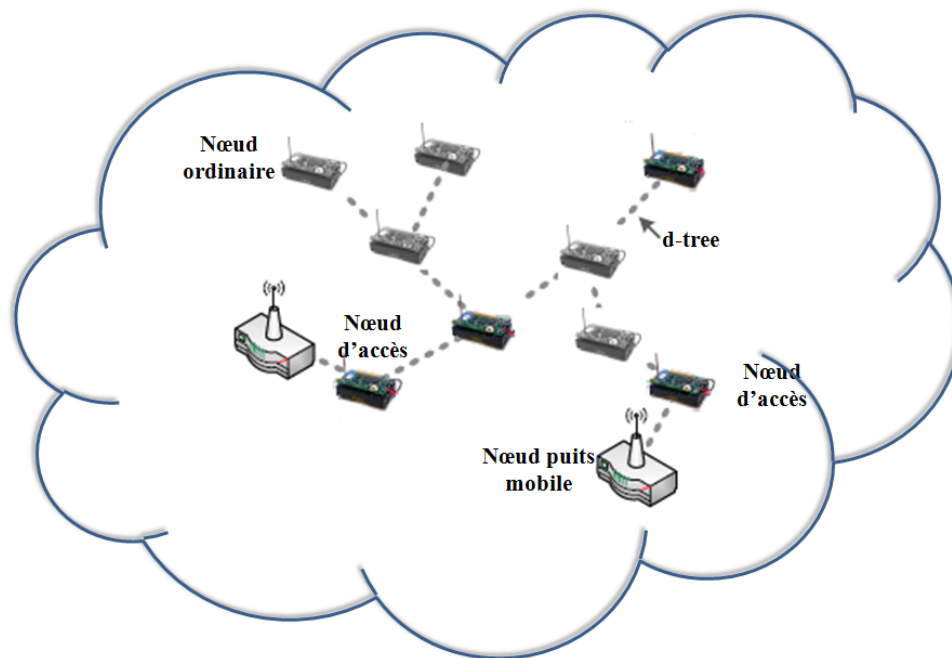


FIGURE 2.10 – Le Scalable Energy-efficient Asynchronous Dissemination(SEAD).

- **Cluster-based protocol** :Un autre protocole de routage basé sur une structure en clusters est proposé dans [59], où les nœuds sont divisés en plusieurs groupes appelés clusters, qui sont visités par l'élément mobile. Dans chaque groupe un nœud est élu comme ClusterHead (CH) afin de gérer la collecte dans son groupe et de transmettre les données collectées vers l'élément mobile. L'organisation en clusters est initiée par l'élément mobile en diffusant un Hello message. Puis, les nœuds qui sont à une distance d'un seul saut de l'élément mobile sont élus comme CHs et propagent à leurs tours le message dans leurs groupes. Chaque CH collecte les données depuis son groupe à l'aide du protocole de dissémination Directed Diffusion [42]. Ensuite, les données collectées sont envoyées par les CHs vers l'élément mobile, qui diffuse périodiquement un message Poll afin qu'il soit détecté par les CHs, selon un modèle de communication basé sur des acquittements.

2.5 Conclusion

Les réseaux de capteurs sont une technologie récente. L'objectif d'un réseau de capteurs est de collecter les données depuis l'environnement physique sur lequel il est déployé. Pour cela, le réseau de capteurs est constitué de deux éléments principaux, les capteurs qui captent et envoient l'information et les puits qui récoltent les informations collectées.

L'une des principales problématiques de recherche dans les réseaux de capteurs est l'échange de données entre les nœuds capteurs tout en respectant les contraintes qui caractérisent ces réseaux, comme la consommation énergétique. À cet effet, plusieurs protocoles de collecte de données ont été proposés dans la littérature, permettant de définir l'échange de données et la communication entre les nœuds capteurs, et de collecter efficacement des données depuis le monde physique.

Dans ce chapitre, nous avons donné une vue générale sur le processus de la collecte de données au sein des réseaux de capteurs avec des nœuds statiques, ainsi qu'avec des éléments mobiles, avant de présenter un état de l'art sur les protocoles de la collecte de données dans les réseaux véhiculaires dans le chapitre suivant.

Chapitre 3

Les protocoles de collecte de données dans VANets

Aujourd'hui, les nouveaux véhicules sont dotés de plusieurs équipements de détection (capteurs, caméra, radar, . . .), qui leur permettent de collecter différentes informations depuis le trafic routier telles que, la vitesse, l'accélération, la température, la densité, etc [18]. Ensuite, ces informations sont envoyées aux RSUs et aux centres de gestion du trafic, dans le but de fournir un trafic routier plus sûr, plus efficace et plus confortable.

Dans ce contexte, différents protocoles de collecte dans les réseaux véhiculaires ont été proposés dans la littérature dont l'objectif est de collecter des informations, en temps réel ou non, sur le trafic routier pour un traitement ultérieure.

Dans ce chapitre, nous présentons les protocoles de collecte dans les réseaux véhiculaires. Nous commençons par définir les critères de classification de ces protocoles. Puis, nous présentons les techniques d'accès au canal utilisées dans les réseaux véhiculaires. Par la suite, nous exposons les protocoles de collecte de données dans les réseaux véhiculaires. Enfin, nous élaborons une étude comparative entre ces protocoles.

3.1 Les critères de classification

Les protocoles de collecte de données dans les réseaux véhiculaires peuvent être classés selon plusieurs critères, on peut citer :

- ◇ **Architecture en clusters** : Le principe de cette architecture est de diviser la route en plusieurs segments et d'élire sur chaque segment un ou plusieurs, CH (Cluster Head), qui s'occupe d'organiser l'échange et la collecte des données depuis les véhicules se trouvant sur son segment (voir FIGURE 3.1).

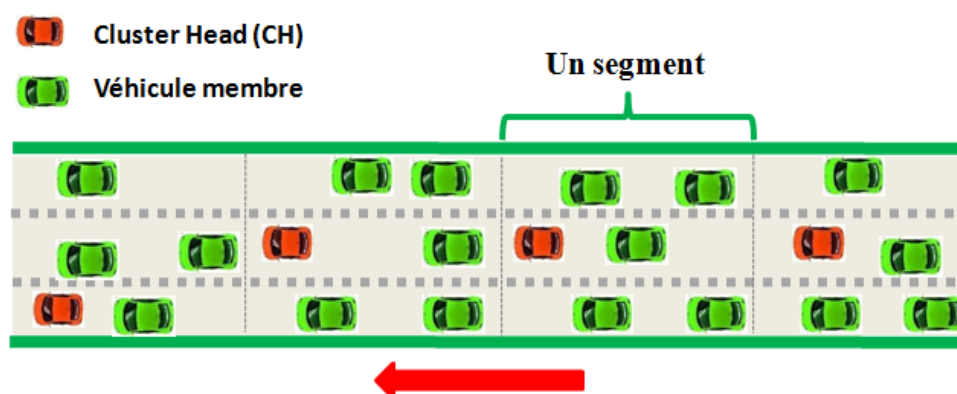


FIGURE 3.1 – Architecture en clusters.

La segmentation de la route peut être statique où la route est divisée préalablement, avant la collecte, en segments de taille fixe, ou dynamique où chaque CH lors de la collecte forme un segment en fonction de sa portée [60].

- ◇ **Mode de communication** : On trouve des protocoles qui sont basés seulement sur une communication V2V, véhicule à véhicule, c'est-à-dire sans passer par l'intermédiaire d'une infrastructure fixe, alors que d'autres nécessitent la présence des infrastructures, RSUs (Road Side Units), afin d'effectuer la collecte des données via des communications V2V et I2V [19].
- ◇ **Méthode d'accès au canal** : Pour faire face aux problèmes de collisions et d'interférences, les protocoles de collecte de données doivent implémenter une méthode d'accès, qui permet de gérer le tour de la parole et d'ordonner l'accès au canal entre les véhicules [4].
- ◇ **Détection et correction des erreurs** : Ce dernier critère permet d'augmenter la fiabilité de communication inter-véhicules et de fournir un mécanisme de retransmission des données erronées. Pour cela, les protocoles doivent utiliser, en premier lieu, une méthode de détection des erreurs avant de mettre en œuvre un mécanisme de retransmission des données.

3.2 Les méthodes d'accès au canal dans les réseaux véhiculaires

Une des problématiques traitées dans les réseaux véhiculaires est la conception des protocoles de contrôle d'accès au canal (MAC) robustes et efficaces, pour une communication V2V. Cette attention émergente est justifiée par le caractère dynamique et de la déconnexion fréquente dans ce type de réseaux. Par exemple, il a été démontré que la durée pendant laquelle deux véhicules, roulant avec une vitesse de 120 Km/h et dans deux directions différentes, sont dans la portée de communication l'un de l'autre est moins de 30 secondes, avec une portée de communication égale à 1000 m [3][4].

En effet, les protocoles de contrôle d'accès au canal dans les réseaux véhiculaires peuvent être partitionnés en deux grandes catégories, les méthodes contention-based et contention-free.

3.2.1 Les méthodes Contention-based

Les méthodes contention-based sont basées sur le principe de l'accès aléatoire et asynchrone au canal. Parmi ces méthodes, on trouve la méthode DCF (Distributed Coordination Function) [61] présentée dans le standard 802.11 [23], cette méthode est basée sur plusieurs techniques d'accès au canal, parmi lesquelles on trouve la technique CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) avec RTS et CTS dont le principe générale est le suivant :

1. Chaque nœud veut envoyer des données déclenche un back-off pendant un temps aléatoire,
2. à la fin du temps du back-off, le nœud envoie un message RTS (Request To Send) vers le destinataire,
3. le destinataire accuse le message RTS en envoyant un message CTS (Clear To Send),
4. les autres nœuds qui ont reçu les deux messages RTS et CTS doivent s'abstenir pendant la durée de communication,
5. le nœud émetteur envoie ses données vers le destinataire,
6. à la fin de la communication, le nœud destinataire acquitte la bonne réception des données en envoyant un paquet Ack (Acknowledge).

Les méthodes de cette catégorie présentent l'avantage, contrairement aux méthodes contention-free, qu'elles ne sont pas sensible à la forte mobilité des nœuds et aux changements fréquent de la topologie. Cependant, le taux important de collisions et le délai illimité de communication présentent les majeurs points négatifs de ces méthodes, ceci est dû principalement au caractère d'accès aléatoire au canal.

3.2.2 Les méthodes Contention-free

Les méthodes contention-free, appelées aussi les méthodes structurées, sont basées sur le principe de l'accès ordonné et synchrone au canal. Le principe général de ces méthodes est que des intervalles de temps, time-slots, sur des bandes de fréquences sont attribués à certains nœuds. Parmi ces méthodes on trouve :

- **La technique TDMA (Time Division Multiple Access)** : La technique TDMA [62] consiste à partager le temps d'utilisation du support en intervalles, qui seront alloués aux différents véhicules. Ainsi, la totalité de la Bande Passante est allouée à un véhicule à un instant donné, ce qui permet d'éviter les collisions de paquets. Autrement dit, les différents véhicules accèdent successivement et périodiquement au canal unique, pendant des tranches de temps égales (timeslots).
- **La technique SDMA (Space Division Multiple Access)** : Dans laquelle l'accès au canal est contrôlé en fonction de la localisation géographique des véhicules. Dans SDMA [11], un intervalle de temps est alloué à chaque véhicule en fonction de sa position géographique. Ainsi, un seul véhicule accède au canal à un instant donné.

À noter que dans cette technique, les véhicules nécessitent un système de localisation tel que GPS (Global Positioning System) afin de mesurer leurs positions géographiques.

- **CDMA (Code Division Multiple Access)** : Le principe de cette technique [63] repose sur le fait que tous les véhicules peuvent accéder au canal en même temps, mais chaque véhicule envoie leurs données grâce à un code, qui le partage seulement avec le destinataire. Ensuite, le destinataire réalise l'opération inverse et décode les données reçues. Avec cette

technique, il est possible de coder plusieurs signaux et de les envoyer simultanément sur un même canal de communication.

L'avantage de ces méthodes c'est qu'elles minimisent les problèmes de collisions lors de la communication. Cependant, elles nécessitent une grande coordination entre les nœuds du réseau afin de communiquer.

3.3 Les protocoles de collecte dans les réseaux véhiculaires

Nous présentons dans cette section les protocoles de collecte des données au sein des réseaux véhiculaires, en présentant à chaque fois les points forts et les faiblesses de chaque protocole.

3.3.1 Le protocole CGP (Clustered Gathering Protocol)

Dans [8], les auteurs ont proposé CGP (Clustered Gathering Protocol), qui est basé sur des mécanismes de dissémination et d'agrégation de données. CGP utilise une architecture en clusters et les véhicules envoient leurs données seulement au CH présent dans leurs segments. Pour délivrer les données au RSU, une technique basée sur le principe du routage géographique est utilisée, où les données sont envoyées au véhicule le plus proche du prochain point d'accès. Le but de ce protocole est de collecter des données depuis le réseau véhiculaire afin d'offrir différents types de services tels que :

- Un service de collecte, en temps réel, des positions et des vitesses de tous les véhicules,
- un service de localisation pour les clients qui veulent suivre la mobilité de leurs véhicules,
- un service de détection d'espace vide de stationnement dans un parking,
- un service de diffusion des messages d'avertissements, quand un événement inhabituel se produit (une diminution soudaine de la vitesse de plusieurs véhicules, par exemple).
- un service pour les applications de surveillance tel que proposé dans [64], où les véhicules détectent et sauvegardent les plaques d'immatriculation de tous les autres véhicules rencontrés sur le trajet.

Basé sur une architecture en clusters où la segmentation est statique, CGP divise la route préalablement en plusieurs segments de même longueur et comprend quatre grandes phases, chacune d'elle commence périodiquement et a une durée prédéfinie :

Phase d'élection d'un ClusterHead

Avant la collecte des données, une première phase d'élection d'un ClusterHead est nécessaire. Pour cela, les auteurs ont implémenté un algorithme d'élection exécuté par chaque véhicule basé sur Back-off, qui est calculé en fonction de la distance entre chaque véhicule et l'extrémité du segment dans le sens du trafic, de telle sorte que le véhicule le plus proche de l'extrémité de fin sera choisi comme CH.

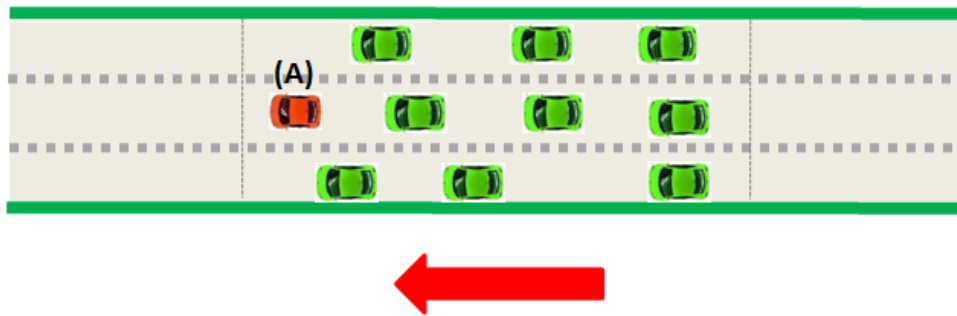


FIGURE 3.2 – L'élection du cluster head.

Un exemple d'élection d'un ClusterHead est illustré dans la figure 3.2. Le véhicule (A) sera choisi comme ClusterHead car il est le plus proche véhicule de l'extrémité de fin du segment.

Phase de collecte de données locales

Durant cette phase, tous les nœuds envoient leurs données au ClusterHead. Pour l'accès au canal, les auteurs utilisent un mécanisme similaire à CSMA/CA utilisée par DCF [61], où :

1. Chaque nœud veut envoyer des données déclenche un back-off pendant un temps aléatoire,
2. à la fin du temps du back-off, le nœud envoie un message RTS au ClusterHead,
3. le ClusterHead accuse le message RTS en envoyant un message CTS,
4. les autres nœuds qui ont reçu les deux paquets RTS et CTS doivent s'abstenir pendant la durée de communication,
5. le nœud envoie ses données au ClusterHead,
6. et à la fin, le ClusterHead accuse la réception des données, en envoyant un paquet ACK.

Phase d'agrégation des données collectées

Au lieu que tous les véhicules envoient leurs données vers le point d'accès, chaque ClusterHead agrège les données collectées dans son segment. Par exemple, chaque nœud envoie son identifiant, sa position et sa vitesse, ce qui permet au ClusterHead de calculer le nombre des nœuds dans son segment et la vitesse moyenne.

Phase de diffusion de données agrégées vers les RSUs

C'est la dernière phase qui consiste à envoyer les données collectées au point d'accès, pour ce faire, si le ClusterHead est dans le segment le plus proche du point d'accès, il diffuse ses données sinon, il les envoie au ClusterHead sur le prochain segment (voir FIGURE 3.3).

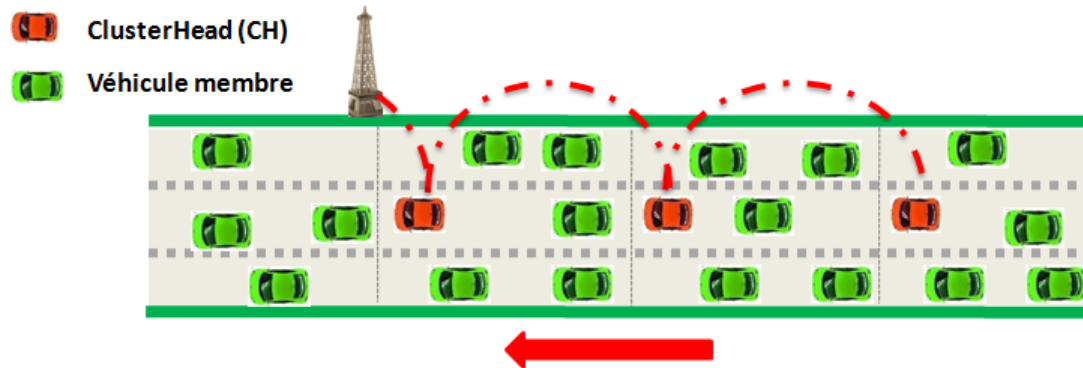


FIGURE 3.3 – Communication V2V puis V2I.

Bilan et Discussion

Le protocole CGP est un protocole de collecte basé sur une architecture en clusters, il permet de fournir des informations, en temps réel, sur le trafic routier. Par la suite, ces informations seront envoyées aux points d'accès pour un traitement ultérieur. Cependant, CGP souffre de certaines limites : Lors de l'élection d'un ClusterHead sur chaque segment, un nombre important de messages doivent être échangés. Par exemple, si un véhicule (i) reçoit un message CH-ANNOUNCE d'un véhicule (j) et si la position du (j) est plus proche de l'extrémité par rapport à (i), le véhicule (i) annule son back-off, sinon, le véhicule (i) attend l'expiration de son back-off et diffuse un CH-ANNOUNCE. Ainsi, pour les applications qui ne sont pas en temps réel, le choix du véhicule le plus proche de l'extrémité du segment comme ClusterHead implique que le ClusterHead sera le premier véhicule qui quitte le segment ce qui diminue la quantité des données collectées ainsi que la durée de vie du ClusterHead. Un autre inconvénient, la portée des véhicules est nettement supérieure à la longueur des segments, ce qui oblige d'autres véhicules se trouvant sur d'autres segments de ne pas accéder au canal, même si la communication est effectuée sur d'autres segments puisque ils reçoivent le paquet CTS, envoyé par le CH. De plus, pour la détection et la correction des erreurs lors de la transmission des données, CGP utilise une technique d'accès au canal qui existe déjà (CSMA/CA).

3.3.2 Le protocole TrafficGather

Dans [7], les auteurs ont proposé un protocole de collecte basé sur une architecture en cluster appelé TrafficGather, dont le processus de la collecte est déclenché par un véhicule et sans l'intervention des points d'accès (RSUs). En adoptant le principe du Space Division Multiple Acces (SDMA) [11], où chaque cluster est divisée en plusieurs blocs de la même longueur, 5 mètres par exemple, et chaque véhicule n'accède au canal qu'au time-slot correspond au bloc courant du véhicule. Ce protocole se déroulera en trois grandes phases :

Phase de configuration du réseau

Durant cette phase, l'ensemble des véhicules sont répartis en plusieurs groupes (Clusters) et pour chaque groupe un véhicule appelé "Cluster Vehicle" (CV) s'occupe de la collecte des données, depuis tous les véhicules se trouvant sur son cluster.

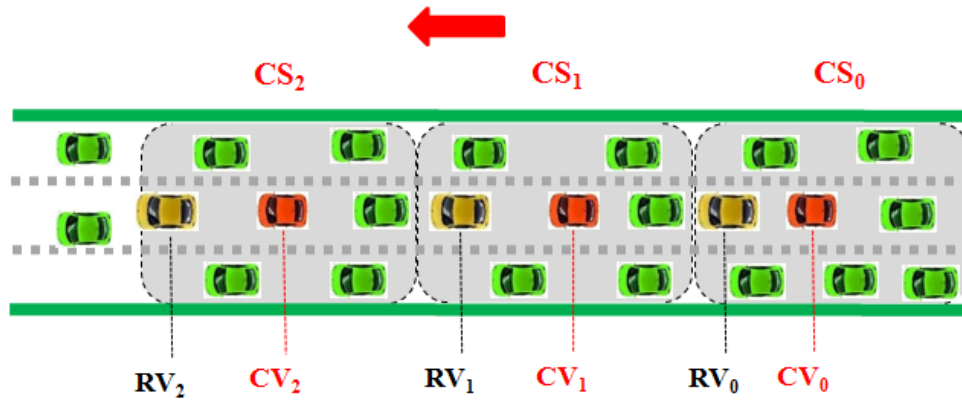


FIGURE 3.4 – Formation des Clusters Space.

Le CV qui déclenche l'opération de la collecte est appelé CV_0 , forme un premier Cluster Space (CS_0) dont la longueur est équivalente à la portée du CV_0 . Puis, le CV_0 diffuse un paquet RM (Request Message) indiquant son identifiant, sa position, un champ indiquant que le paquet a été envoyé par un CV, et la direction de la collecte. Ensuite, le véhicule voisin le plus éloigné du CV_0 sera choisi comme un véhicule relai (RV_0). Le véhicule RV_0 diffuse à son tour un paquet RM afin de choisir un autre CV_1 et forme un autre CS_1 ainsi de suite (voir FIGURE 3.4). Cette phase se terminera après une durée T initialisée par le CV_0 . À noter que le même processus sera appliqué pour les véhicules se trouvant derrière le CV_0 .

Phase de collecte de données

Afin de garantir un accès au canal efficace et sans collisions, les auteurs ont mis en œuvre un mécanisme d'accès au canal fortement inspiré du SDMA. Pour cela, chaque cluster space (CS) est divisé en plusieurs blocs d'une longueur $L_s = 5$ mètres (voir FIGURE 3.5). Donc, le nombre total des blocs par voie sera $N = 2R/L_s$ où R est la portée du CV, et chaque bloc est référencé par (i, j) où "i" est le numéro du bloc sur la voie "j". Par conséquent, le nombre total des blocs par cluster space sera :

$$B = N * M, \text{ où } M \text{ est le nombre des voies.}$$

Pour entamer la collecte, chaque CV diffuse un HELLO message indiquant son identifiant et sa position, ce qui permet à chaque véhicule d'un CS de calculer son numéro de segment, i , courant en fonction de la distance entre lui et le CV, qui a diffusé le message. Par la suite, chaque véhicule doit attendre $i + (j * N) + 1$ slots avant d'accéder au canal.

Par exemple, dans la figure 3.5 suivante le cluster véhicule 0 (CV_0) est dans le bloc $(4, 1)$ et doit attendre une durée égale à $T_{HELLO} + 15t_{time-slot}$, où T_{HELLO} est l'instant de réception du message HELLO.

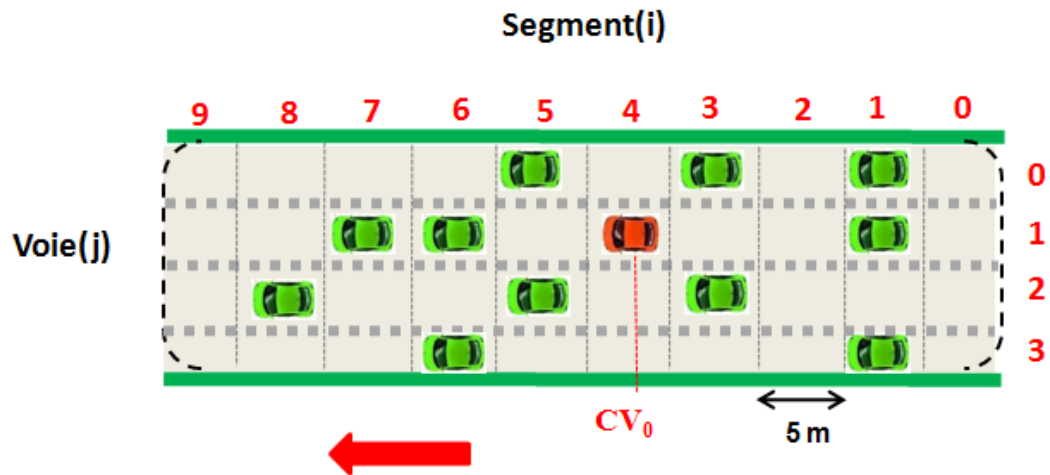


FIGURE 3.5 – Le partitionnement d'un CS en blocs.

Enfin, puisque il y aura toujours une zone en commun entre deux cluster spaces adjacents et afin d'éviter le problème de collision, les auteurs ont différé l'instant d'accès des clusters adjacents de tel sorte que les cluster spaces pair accèdent à l'instant T et après une période égale à Δ , où $\Delta = B * t_{time-slot}$ où $t_{time-slot}$ est la durée d'un time-slot, les cluster spaces impair accèdent au canal (voir FIGURE 3.6).

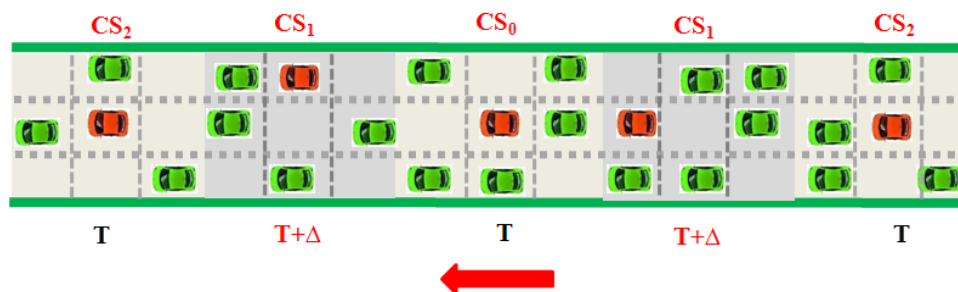


FIGURE 3.6 – Synchronisation des cluster spaces adjacents.

Phase d'envoi des données vers l'initiateur de la collecte

Après une durée égale à $T + 2\Delta$, pendant laquelle tous les véhicules sur tous les clusters space accèdent au canal, chaque CV envoie ses données collectées, depuis son cluster, vers le véhicule qui a initié le processus de la collecte CV_0 . Ainsi, ils se basent sur la technique de dissémination de données simple Flooding [6], c'est-à-dire chaque CV diffuse ses données et chaque véhicule recevant ces données les rediffuse à nouveau jusqu'à ce que ces données atteignent leur destination CV_0 .

Bilan et Discussion

TrafficGather est un protocole de collecte de données, qui est basé sur une communication véhicule à véhicule, V2V, seulement. L'inconvénient majeur de ce protocole est l'affectation d'un time-slot à chaque bloc même si le bloc est vide, c'est-à-dire que la distribution des slots est statique, ainsi que lors de la dernière phase d'envoi de données vers l'initiateur, les auteurs se sont basés sur le simple Flooding ce qui peut produire le problème du broadcast storm [65]. De même, ce protocole n'implémente aucun mécanisme de retransmission des données erronées.

3.3.3 Le protocole Road Probing

Dans [66], les auteurs ont proposé un nouveau protocole de collecte de données dans les réseaux véhiculaires, nommé Road Probing. Dans ce protocole, le processus de la collecte de données est déclenché par un RSU qui sélectionne certains véhicules, Probes, pour effectuer la collecte en se basant sur leur mobilité.

Chaque véhicule choisi par le RSU a trois responsabilités :

1. Collecter les données depuis le réseau routier,
2. collecter des données depuis les véhicules sélectionnés par le RSU devant lui,
3. et envoyer les données agrégées aux véhicules sélectionnés par le RSU derrière lui.

La sélection des véhicules de la collecte

Le protocole Road Probing est basé sur l'hypothèse que chaque véhicule peut diffuser un message de contrôle tous les 100-300 ms, contenant la vitesse, la localisation, etc., vers le RSU ou les autres véhicules.

Pour sélectionner les véhicules, les voies de la route sont divisées en trois types de voies, la voie la plus à gauche, la voie la plus à droite et les voies intermédiaires. De même, la zone couverte par le RSU est divisée en deux zones de sélection, SS_1 et SS_2 , et une dernière zone dite Exit section (voir FIGURE 3.7). Chaque fois qu'un véhicule entre dans la zone du RSU, il envoie son profil de mobilité c'est-à-dire son temps moyen du trajet sur chaque type de voie. Pour choisir un probe, le RSU favorise les véhicules qui ont passé le temps moyen le plus long sur les voies intermédiaires, et qui seront logiquement les plus permanents et les plus stables dans le réseau.

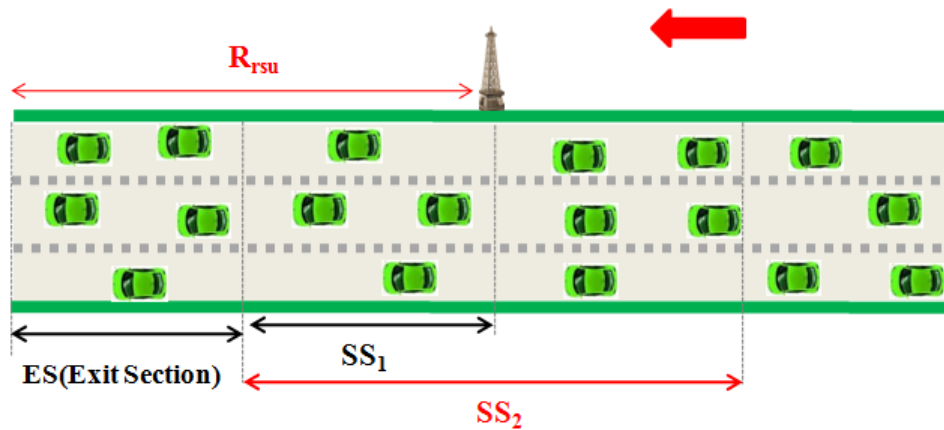


FIGURE 3.7 – Les zones de sélection.

Pour la zone de la sélection, une fois le véhicule sélectionné entre dans la dernière zone, Exit Section, les auteurs ont implémenté deux modèles de sélection :

1. **Cas 1** : Le prochain véhicule sera sélectionné dans la zone SS_1 . Donc, la distance entre deux véhicules consécutifs sélectionnés par le RSU ne dépasse pas $R/2$ (voir FIGURE 3.8).

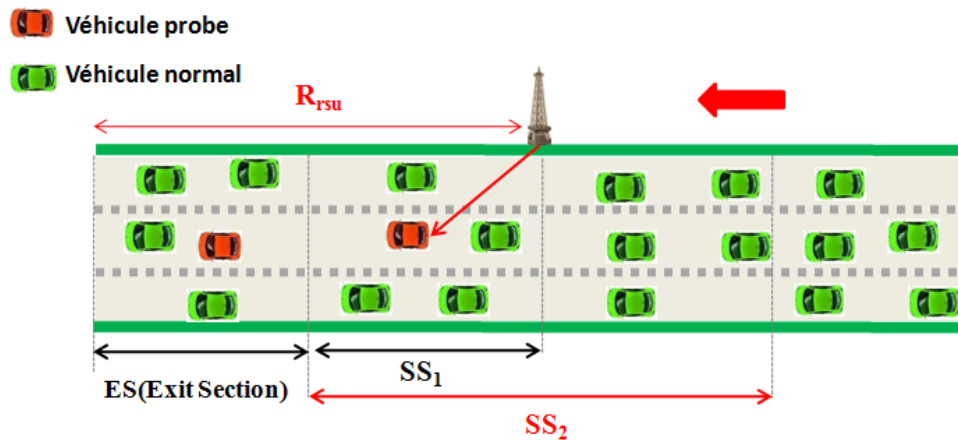


FIGURE 3.8 – Le protocole Road Probing : modèle 1.

2. **Cas 2** : Le prochain véhicule sera sélectionné dans la zone SS_2 . Donc, la distance entre deux véhicules consécutifs sélectionnés par le RSU ne dépasse pas R (voir FIGURE 3.9).

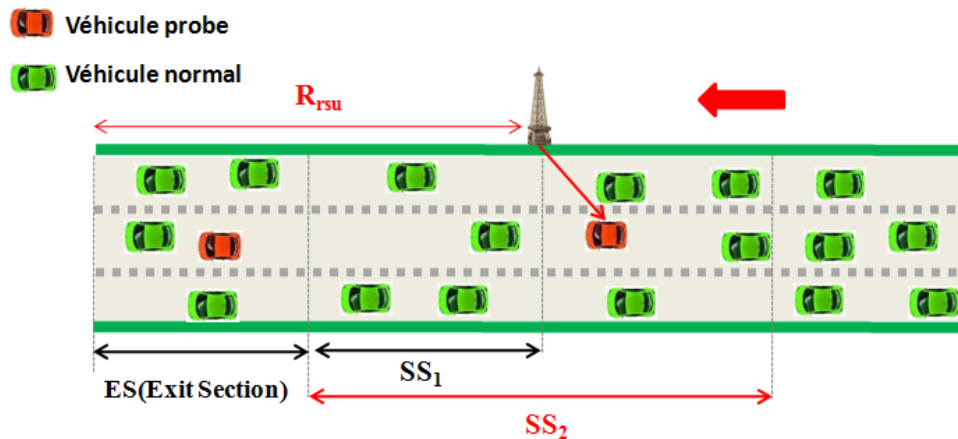


FIGURE 3.9 – Le protocole Road Probing : modèle 2.

La communication entre les véhicules sélectionnés

Les données collectées depuis le réseau routier sont envoyées au RSU par les véhicules sélectionnés, Probes, via une communication multi-saut. Pour ce faire, chaque véhicule sélectionné par le RSU collecte les données désirées et les agrège avec celles provenant des véhicules devant lui (voir FIGURE 3.10).

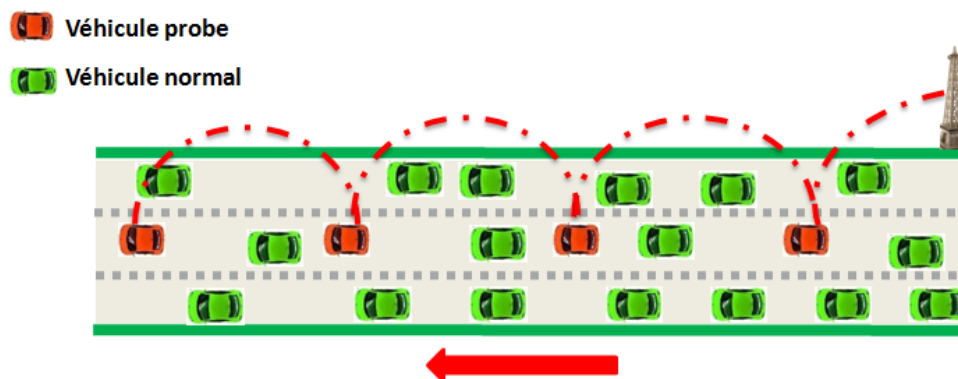


FIGURE 3.10 – Communication multi-sauts vers le RSU.

Bilan et Discussion

Road Probing est un protocole de collecte dont le processus de la collecte est initié par le RSU. Le RSU sélectionne certains véhicules, Probes, en se basant sur leurs temps moyen du trajet sur les trois types de voies. Ensuite, les données collectées sont envoyées vers le RSU via une communication multi-saut. Cependant, ce protocole présente l'inconvénient que la collecte des données n'est effectuée que par certains véhicules alors que pour avoir une vue globale sur le trafic routier, on doit avoir le plus grand nombre possible de véhicules qui participent à la collecte. Un autre point négatif de ce protocole, pour le premier modèle de sélection, SS_1 , on

aura un nombre important de sauts, qui peut poser des problèmes pour les applications non tolérantes au délai, ainsi que pour le deuxième modèle de sélection, SS_2 , la disparition d'un seul saut, véhicule probe, annule tous le processus de la collecte puisque la distance entre deux véhicules probes est égale à R . De plus, l'inconvénient majeur de ce protocole concerne l'accès au canal, les auteurs se sont basés sur l'hypothèse que chaque véhicule peut envoyer un message de contrôle tous les 100-300 ms, et ils n'ont pas défini un mécanisme d'accès au canal qui évite les problèmes de collisions. Ainsi, la retransmission de données en cas d'erreurs n'est pas prise en compte par ce protocole.

3.3.4 Le protocole COL

Un protocole de collecte auto-stabilisant appelé COL, basé seulement sur une communication véhicule à véhicule (V2V) a été proposé dans [67]. Ce protocole garantit trois propriétés pouvant s'avérer avantageuses dans un réseau véhiculaire : (1) n'importe quel véhicule a la possibilité de collecter les données depuis son voisinage direct, (2) sa tolérance vis à vis des partitionnements du réseau, (3) la collecte de données s'effectue à la demande et se termine une fois que cette dernière est achevée.

Dans ce protocole, chaque véhicule confronte sa vue locale avec celles provenant des autres véhicules afin d'obtenir une information plus précise et plus pertinente sur le réseau routier. Pour cela, un opérateur algébrique, r-opérateur, est utilisé [68][69]. Cet opérateur permet à ce protocole de supporter le changement fréquent de la topologie du réseau véhiculaire.

La notion de vue locale et r-opérateur

Après l'initiation d'un processus de collecte, chaque véhicule diffuse périodiquement sa vue locale à son voisinage. La vue locale, de profondeur j , est simplement une liste $(N_0, N_1, N_2, \dots, N_p)$ tel que pour tout j , N_j est l'ensemble des couples $(v; x_v)$ où v est un véhicule se situant à j sauts de l'initiateur de la collecte et x_v correspond à la donnée locale de v visée par la collecte.

Parallèlement à cela, chaque véhicule remet périodiquement sa vue locale à jour en fonction des autres vues, qu'il a reçu et ce, en appliquant le r-opérateur *ant* qui consiste à fusionner les vues reçues avec la vue locale, terme à terme, tout en supprimant l'information redondante. Par exemple, si la vue du véhicule 1 est $V_1 = ((1, a), (2, b), (3, c), (4, d))$ et si ce même véhicule reçoit la vue d'un autre véhicule $V_2 = ((3, c), (1, a), (4, d), (5, e), (2, b), (6, f), (8, h), (7, g))$, alors il calculera la vue :

$$\begin{aligned} ant(V1; V2) &= ((1, a), (2, b), (3, c), (4, d), \cancel{(3, c)}, \cancel{(1, a)}, \cancel{(4, d)}, (5, e), \cancel{(2, b)}, (6, f), \\ &(8, h), (7, g)) = ((1, a), (2, b), (3, c), (4, d), (5, e), (6, f), (8, h), (7, g)). \end{aligned}$$

La collecte de données

Dans ce protocole, le processus de la collecte est initié par un véhicule appelé l'initiateur, ce dernier diffuse un message composé de quatre champs à destination de son voisinage. Ces champs correspondent respectivement au numéro de la collecte (*col_id*), l'identifiant de l'initiateur (*col_initiator*), les paramètres de la collecte (*col_param*), et sa vue locale (*col_view*). Le champs (*col_param*) comprend le type de données à collecter, la profondeur de la collecte, la

durée maximale de la collecte.

Toutefois, au départ de chaque collecte, seul l'initiateur est impliqué dans le processus de collecte. Mais le nombre de véhicules concernés augmente au fur et à mesure de la propagation des messages issus de l'initiateur dans le réseau, qui s'arrêtera une fois atteint $maxdst$ (maximal distance) définie par l'initiateur. Ensuite, après l'initiation de la collecte, chaque véhicule informe et remet périodiquement sa vue locale à jour, en fonction des autres vues qu'il a reçu et ce, en appliquant le r-opérateur ant expliqué dans la section précédente.

La gestion du voisinage dynamique

Pour faire face à la forte mobilité des véhicules et l'instabilité de voisinage, chaque fois qu'un véhicule v reçoit un message d'un véhicule u , il lui alloue localement une durée de vie choisie aléatoirement. De cette manière, si v ne reçoit aucun autre message de la part de u pendant cette durée de vie, v considèrera que u ne fait plus partie de son voisinage et toutes les données relatives à u sont effacées.

La terminaison de la collecte

Le processus de la collecte a une durée déterminée initialement par l'initiateur. Après cette durée, chaque véhicule quitte le processus de la collecte, cessant alors de mettre à jour et d'émettre sa vue locale.

Lorsque la collecte est terminée, c'est la vue locale de l'initiateur qui est considérée comme étant le résultat, qui contient les différentes données classées selon la distance à l'initiateur. Ce résultat est ensuite exploité localement par l'initiateur, qui peut aussi l'envoyer à d'autres véhicules ou vers un serveur de l'infrastructure via une passerelle [70].

Bilan et Discussion

COL est un protocole de collecte de données basé seulement sur une communication véhicule à véhicule, V2V. Dans ce protocole, le processus de la collecte est initié par un véhicule en précisant la profondeur de la collecte et le type de données à collecter. Ensuite, chaque véhicule diffuse périodiquement sa vue locale et l'agrège avec les autres vues reçues depuis les autres véhicules et ceci grâce à l'outil r-opérateur (ant). Les auteurs de ce protocole ont proposé une technique d'agrégation de données au niveau de chaque véhicule. Cependant, ils n'ont pas proposé une méthode permettant de définir l'échange de données pour une communication inter-véhicule, ils se sont basés seulement sur l'hypothèse que chaque véhicule diffuse périodiquement sa vue locale vers son voisinage. Par conséquent, ils n'ont pas proposé un mécanisme d'accès au canal qui évite les problèmes de collisions et qui permet la retransmission de données en cas d'erreurs.

3.4 Etude comparative

Les protocoles de collecte peuvent être classés selon plusieurs critères. On trouve des protocoles qui sont basés sur une architecture en clusters dont la segmentation de la route est statique ou bien dynamique, des protocoles qui sont basés seulement sur une communication véhicule à véhicule c.à.d. sans passer par l'intermédiaire d'une infrastructure, alors que d'autres nécessitent

une infrastructure. De plus, un autre critère qui a une grande influence sur les performances des protocoles, c'est la méthode d'accès au canal. Ce critère permet de réduire les problèmes de collisions et d'interférences. On remarque aussi que tous les protocoles n'ont pas implémenté un mécanisme de détection et de retransmission de données en cas d'erreurs.

Dans la TABLE 3.1, nous présentons une étude comparative entre les protocoles de collecte dans les réseaux véhiculaires selon les critères de classification cités au dessus :

TABLE 3.1 – Comparaison entre les protocoles de collecte existants.

Les protocoles Les critères	CGP	Road Probing	Traffic Gather	COL
Architecture en clusters	Oui	Non	Oui	Non
La segmentation de la route	Statique	—	Dynamique	—
Mode de communication	V2V/V2I	V2V/V2I /I2V	V2V	V2V
Accès au canal	CSMA/CA	—	SDMA statique	N'est pas spécifiée
Le critère de sélection du CH	Proche de l'extrémité de fin de segment	Temps passé sur les voies intermédiaires	Le plus éloigné du véhicule relai	—
Détection et correction des erreurs	Non	Non	Non	Non

3.5 Conclusion

L'utilisation des informations issues de plusieurs véhicules permet de combler les lacunes de perception d'un seul véhicule sur son environnement. Dans ce contexte, quelques protocoles de collecte de données et d'échange d'informations ont été développés dont l'objectif principal est de définir comment les données collectées seront agrégées et échangées entre les véhicules ou bien avec l'infrastructure, afin de fournir des informations, en temps réel ou non, sur le trafic routier, et de le rendre plus sûr, plus efficace et plus confortable.

Dans ce chapitre, nous avons présenté un état de l'art sur les protocoles de collecte de données au sein des réseaux véhiculaires. À cet effet, nous avons présenté en premier lieu des critères de classification de ces protocoles. Puis, nous avons exposé les protocoles de collecte de données, en donnant nos remarques sur chaque protocole. Enfin, nous avons élaboré une étude comparative entre ces protocoles selon les critères de classification.

En profitant des inconvénients et des avantages de ces protocoles, dans le chapitre suivant, nous décrivons nos deux protocoles de collecte des données au sein des réseaux véhiculaires.

Chapitre 4

Nos protocoles de collecte de données

Les protocoles de collecte de données étudiés dans le chapitre précédent présentent l'avantage que les véhicules peuvent collecter des données depuis le réseau routier, en temps réel ou non, et par conséquent permet d'augmenter le degré de perception de chaque véhicule sur son environnement. Cependant, le majeur inconvénient de ces protocoles est la non implémentation d'un mécanisme fiable de transfert de données lors d'une communication, en cas de réception des données erronées, des collisions ou d'interférences par exemple.

À cet effet, les objectifs de notre recherche consistent à résoudre certaines limites dont souffrent ces protocoles et de bénéficier de leurs avantages, tout en assurant une collecte de données fiable et efficace. Ainsi, nous avons conçu deux protocoles de collecte de données depuis le réseau routier, au niveau de l'autoroute.

Dans ces deux protocoles, les véhicules collectent différents types de données depuis le trafic routier, puis les diffusent vers l'infrastructure pour un traitement ultérieur. Ces données collectées peuvent être utilisées ultérieurement pour, le calcul de la densité de véhicules ou bien de la vitesse moyenne des véhicules sur un segment de route, la détection des espaces vides de stationnement dans un parking, la détection et le contrôle de la congestion, la collecte depuis les magasins voisins des produits disponibles ainsi que des promotions, etc.

Dans ce chapitre, nous présentons nos deux protocoles CDGP (Clustered Data Gathering Protocol) [10] et TCDGP (Token-based Clustered Data Gathering Protocol). Nous commençons par une brève description de chaque protocole. Puis, nous exposons les objectifs fixés et les hypothèses prises en compte afin de les développer et nous détaillons par la suite les grandes phases d'exécution. Enfin, nous élaborons une étude comparative entre nos deux protocoles et les autres protocoles existants dans la littérature.

4.1 Le protocole CDGP (Clustered Data Gathering Protocol)

Il est bien clair qu'une technique de collecte de données efficace doit assurer un échange de données fiable et sans erreurs. Pour concevoir notre protocole appelé CDGP [10], nous nous sommes appuyés sur trois principaux objectifs :

1. **Équité** : en se basant sur un mécanisme d'accès au canal qui donne à tous les véhicules une chance égale pour accéder au canal,

2. **Efficacité** : en collectant le maximum de données en minimum de temps,
3. **Fiabilité** : en implémentant un mécanisme de retransmission de données en cas d'erreurs.

Pour cela, nous avons pris en considération les hypothèses suivantes :

- Chaque véhicule est équipé d'un GPS afin de connaître sa position sur la route,
- la longueur des segments est égale à la portée d'un véhicule,
- chaque véhicule possède la carte routière, qui lui permet de connaître les positions des RSUs sur la route et donc le numéro de segment sur lequel il se trouve,
- le processus de la collecte est initié par un RSU, et
- l'existence d'un réseau d'interconnexion entre les RSUs, qui permet d'envoyer les données collectées au RSU initiateur du processus de la collecte.

Le protocole CDGP s'exécute en deux grandes phases (i) Phase de propagation du message de collecte (CMP, Collection Message Propagation) et (ii) Phase d'agrégation et de dissémination de données (DAD, Data Aggregation and Dissemination). Dans ce qui suit, nous allons détailler ces deux phases.

4.1.1 Initiation du processus de la collecte

La première phase du protocole CDGP, phase de propagation du message de collecte (CMP), est la phase d'initialisation du processus de la collecte. Cette phase s'exécute une seule fois au début du processus dont le but est d'informer tous les véhicules, qui sont dans la zone de la collecte du déclenchement de la procédure de collecte de données.

La diffusion du paquet Col par le RSU initiateur

Initialement le RSU, qui initié l'opération de la collecte, diffuse un paquet Col afin d'informer tous les véhicules qui sont dans sa portée de communication. Ce paquet Col comprend essentiellement six champs, l'identifiant du RSU (RSU-ID), la position du RSU (RSU-POS), la profondeur de la collecte exprimée en nombre de segments (A_length), la direction de la collecte (DC), le type de données visé (DT) et le temps de validation de ce message (VT) (voir FIGURE 4.1).

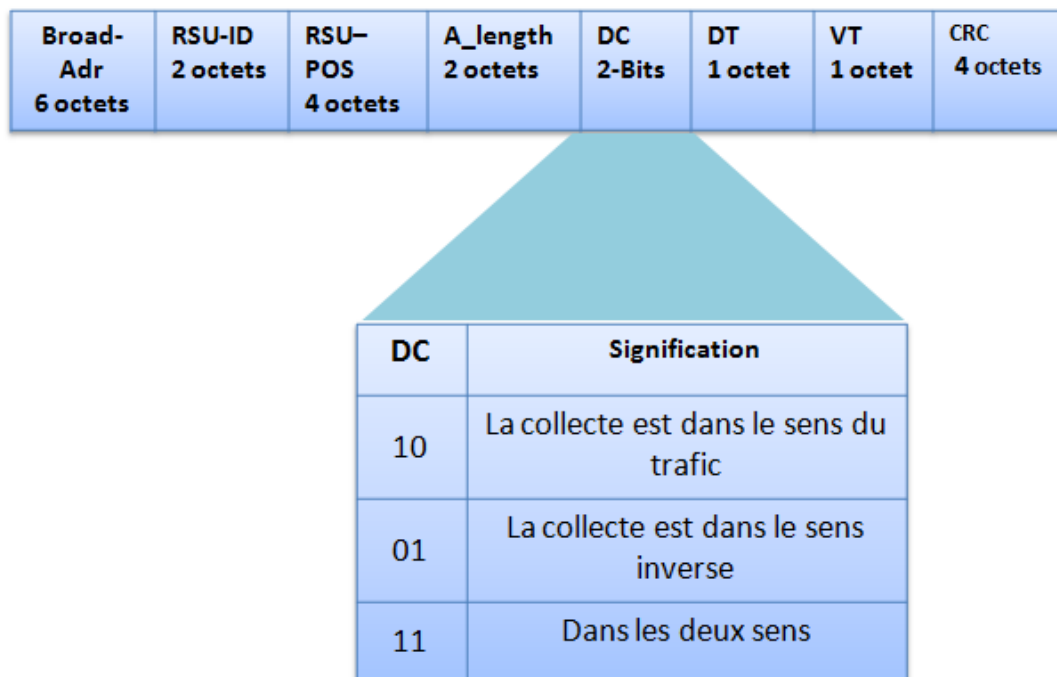


FIGURE 4.1 – Le format du paquet Col.

La diffusion du paquet Col par les véhicules

Ensuite et dans le but d'informer tous les véhicules dans la zone de la collecte, qui sont hors portée du RSU (voir FIGURE 4.2), chaque véhicule recevant le message Col exécute une des deux procédures suivantes :

- **Pour un véhicule qui est avant le RSU** : Après la réception d'un paquet Col depuis le RSU ou bien depuis un véhicule qui est derrière lui, il déclenche un timer dont la durée est proportionnelle à la distance, qui le sépare de l'extrémité de fin de la zone de collecte désignée par le RSU. À l'expiration de ce timer, s'il est encore dans la zone de la collecte et s'il n'a pas reçu un nouveau Col d'un véhicule qui est devant lui, il diffuse à nouveau un paquet Col afin d'informer les véhicules qui sont devant lui.
- **Pour un véhicule qui est derrière le RSU** : Après la réception d'un paquet Col depuis le RSU ou bien depuis un véhicule qui est devant lui, il déclenche un timer dont la durée est proportionnelle à la distance, qui le sépare de l'extrémité de début de la zone de collecte. À l'expiration de ce timer, s'il n'a pas reçu un nouveau Col d'un véhicule qui est derrière lui, il diffuse à nouveau un paquet Col afin d'informer les véhicules qui sont derrière lui.

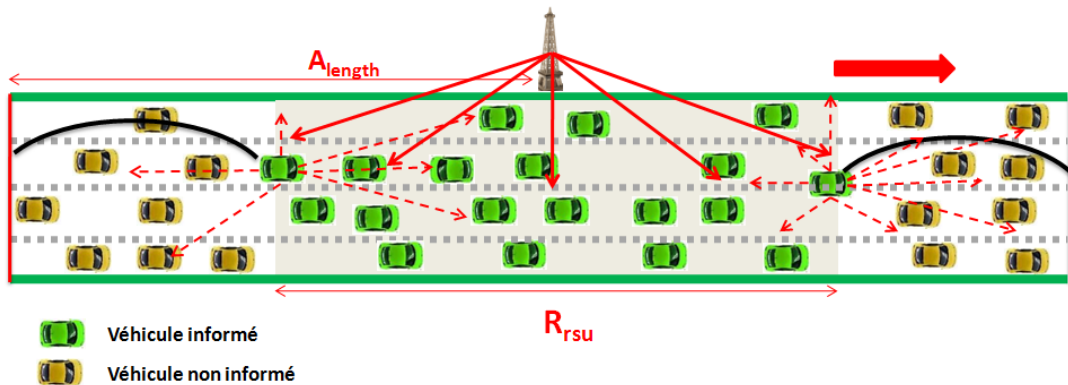


FIGURE 4.2 – La diffusion du paquet Col par un RSU puis par les véhicules.

4.1.2 Le partitionnement de la route

Après avoir informé tous les véhicules du début du processus de la collecte, la phase de collecte de données, la phase d'agrégation et de dissémination de données (DAD), depuis le réseau routier commence. Dans cette phase, nous nous sommes basés sur une architecture en clusters dont la segmentation de la route est statique, où tous les véhicules envoient leurs données collectées vers les ClusterHeads, qui se chargent de l'agrégation et de l'envoi de données vers les RSUs. Cette deuxième phase se terminera lorsque tous les véhicules qui ont été informés quittent la zone de la collecte, ou bien le temps de validation exprimé dans le paquet Col est expiré.

Cependant, avant de détailler la deuxième phase, la question qui se pose maintenant est : quelle technique d'accès au canal faut-il choisir afin d'offrir aux véhicules un accès au canal **équitable et sans collisions** ?

Le partitionnement de la route en segments

À la fin de la première phase, la zone de la collecte est divisée virtuellement en n segments, définis par l'initiateur de la collecte, d'une même longueur L_s . Pour cela, on prend comme axe de début de segmentation le RSU qui a initié le processus de la collecte. De plus, afin d'éviter les problèmes de collisions et d'interférences entre les segments adjacents lors de la collecte, nous avons défini deux types de segments (voir FIGURE 4.3) :

- **Segment de collecte (CS, Collection Segment)**, où les véhicules peuvent échanger leurs données collectées. On peut prendre les segments d'indice pair par exemple comme segments de collecte,
- **Segment de silence (SS, Silence Segment)**, c'est le segment où tous les véhicules n'envoient aucune donnée.

Chaque véhicule dans la zone de collecte doit être capable de déterminer le type de segment sur lequel il se trouve et donc s'il peut envoyer des données ou non. Pour cela, le numéro du segment est déterminé par chaque véhicule, en fonction de la distance qui le sépare du prochain RSU et de la longueur des segments, en appliquant la formule suivante :

$$Num_seg(i) = \lfloor D_i \div L_s \rfloor \quad (4.1)$$

Où D_i est la distance qui sépare le véhicule (i) du prochain RSU dans le sens du trafic et L_s est la longueur du segment (voir FIGURE 4.3).

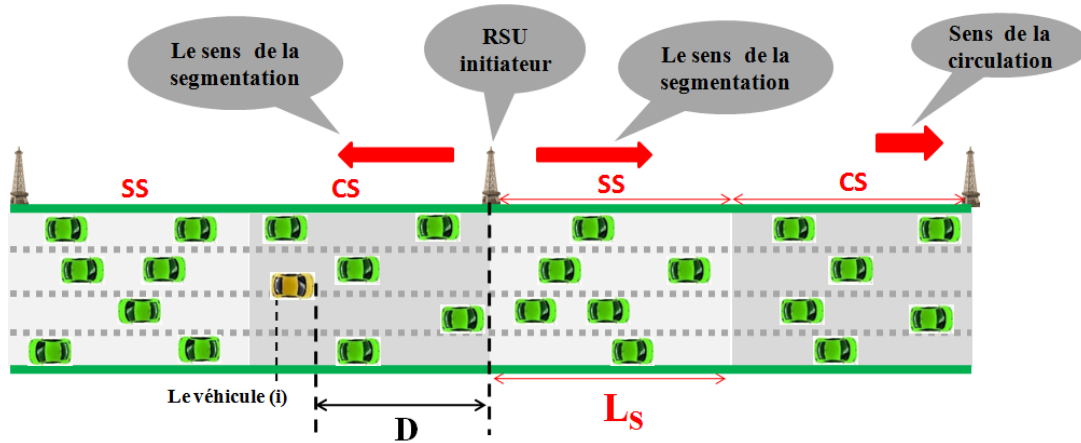


FIGURE 4.3 – Le partitionnement en segments.

Le partitionnement des segments de collecte en blocs

Pour donner à tous les véhicules une chance égale pour accéder au canal et atteindre l'objectif d'équité entre eux, nous nous sommes basés sur la technique d'accès au canal SDMA décrite dans [11] qui consiste à diviser les voies d'un segment de la route en plusieurs blocs et d'affecter à chaque bloc un time-slot. Pour cela, nous avons divisé chaque voie d'un segment de collecte (CS) en plusieurs blocs, L_b , de 5 mètres afin de garantir qu'au plus on aura un seul véhicule par bloc (voir FIGURE 4.4).

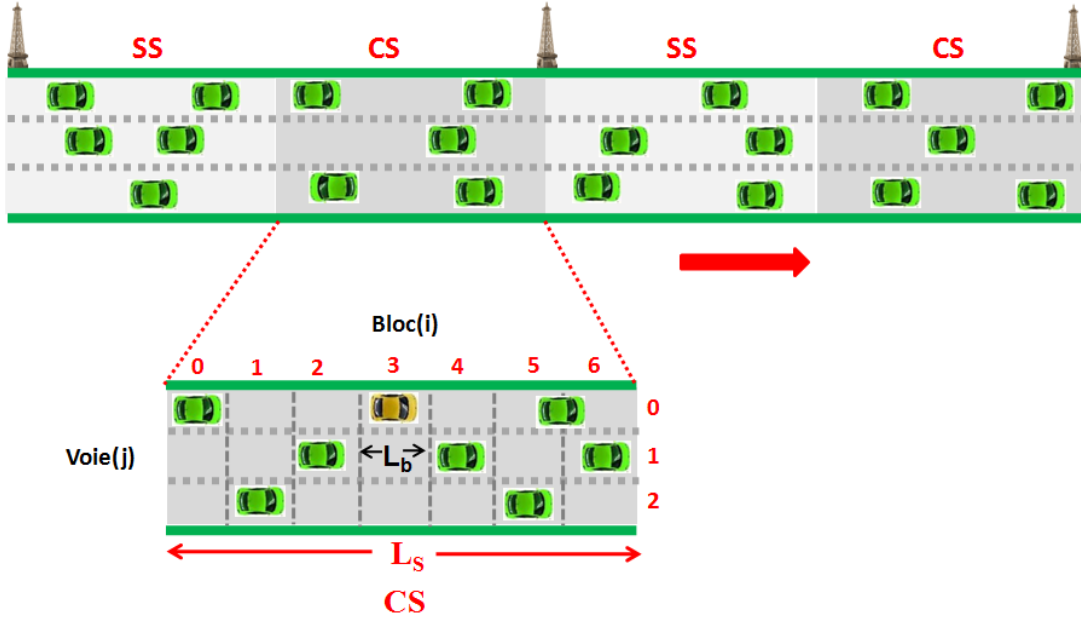


FIGURE 4.4 – Le partitionnement d'un segment de collecte en blocs.

Ainsi, chaque voie d'un segment de collecte peut être partitionnée en $N = L_s \div L_b$ blocs par voie. Ainsi, le nombre total des blocs par segment de collecte, pour un nombre de voies M , sera $B = N * M$ blocs. Dans l'exemple de la figure 4.4, on a $M = 3$ voies, $N = 7$ blocs par voie et le nombre totale des blocs $B = M * N = 21$ blocs.

Chaque segment de collecte maintenant est considéré comme une matrice ($M * N$). Par conséquent, chaque bloc du segment est identifié par (i, j) où $(0 \leq i \leq N-1)$ et $(0 \leq j \leq M-1)$. Donc chaque véhicule peut déterminer à quel bloc il appartient en calculant son (i, j) , à l'aide des deux formules suivantes :

$$\lfloor i \rfloor = (Pos_{xv} - deb_seg) \div L_b \quad (4.2)$$

$$\lfloor j \rfloor = (R_{width} - Pos_{yv}) \div voie_{width} \quad (4.3)$$

Où Pos_{xv} et Pos_{yv} sont respectivement les positions x et y du véhicule v , L_b la longueur d'un bloc, R_{width} et $voie_{width}$ sont respectivement la largeur de la route et d'une voie de la route, et deb_seg est la position x du début de segment courant du véhicule v , qui sera calculé par chaque véhicule selon la formule suivante :

$$Deb_seg(v) = (Num_seg(v) + 1) * L_s \quad (4.4)$$

Où $Num_seg(v)$ est calculé à l'aide de la formule (4.1).

Enfin, le numéro du bloc courant du véhicule v sera déduit à l'aide de la formule suivante :

$$Num_bloc(v) = j * N + i \quad (4.5)$$

À noter que les blocs dans un segment de collecte sont numérotés du 0, 1, 2, ... jusqu'à $B-1$. Par exemple, le véhicule jaune dans la figure 4.4, son bloc courant est identifié par $(i, j) = (3, 0)$ et par conséquent le numéro du son bloc courant sera :

$$Num_bloc(v_jaune) = 0 * 7 + 3 = 3;$$

4.1.3 Élection des ClusterHeads

Pour l'élection des ClusterHeads dans les segments de collecte, nous avons proposé un algorithme basé sur Back-off. Notre algorithme d'élection est implémenté en se basant sur deux critères de sélection :

1. **Le plus proche véhicule du centre du segment** : Afin d'augmenter la durée de vie du ClusterHead et d'éviter le changement fréquent des ClusterHeads, on choisit toujours le véhicule le plus proche du centre de segment.
2. **Le véhicule le plus stable** : Dans le but de choisir le véhicule le plus stable et le plus permanent, nous avons défini trois types de voies de la route, la voie la plus à gauche, la voie la plus à droite et les voies intermédiaires. Dans notre protocole, chaque véhicule sauvegarde son profil de mobilité $(T_{left}, T_{right}, T_{inter})$ c'est-à-dire son temps du trajet sur les trois types de voies. Par la suite, on favorise toujours le véhicule qui a circulé plus sur les voies intermédiaires.

Le principe générale de notre algorithme

Le principe générale de notre algorithme consiste à considérer chaque véhicule informé par la collecte, initialement, à l'état nouveau, dès qu'il arrive à un segment de collecte, il déclenche ensuite un Back-off pendant une durée égale à $B = N * M$ slots, B c'est le nombre total des blocs sur un segment de collecte, afin de tester s'il existe un ClusterHead sur son segment ou non. S'il reçoit un paquet depuis le ClusterHead, il annule son Back-off et il passe à l'état membre, sinon à l'expiration de son Back-off, il déclenche un autre Back-off dont la durée est calculée à l'aide de la formule suivante :

$$T_{backoff}(v) = Dist_to_cen * (T_{left} + T_{right} / (T_{left} + T_{right} + T_{inter})) \quad (4.6)$$

Où $Dist_to_cen$ est la distance qui sépare le véhicule v du centre de segment.

Ensuite, s'il reçoit un paquet, $Dist_slot$ (présenté dans la section suivante), implique qu'un autre véhicule plus proche du centre de segment et plus stable est élu comme ClusterHead, il annule son Back-off et il passe à l'état membre, sinon et à l'expiration de son Back-off il diffuse un paquet $Dist_slot$, et il passe à l'état ClusterHead.

En outre, la formule du $T_{backoff}(v)$ nous permet toujours de choisir le véhicule qui est le plus proche du centre et qui a circulé plus sur les voies intermédiaires. Le pseudo code 1 illustre notre algorithme d'élection des ClusterHeads :

Algorithm 1 ClusterHead Election(Pseudo code)

```

1: if receive(Col)==true then
2:   while is_in_collection_area(v) do
3:     if is_in_collection_segment(v) then
4:       Déclenche un timer_1 pendant  $B$  slots //Afin de tester s'il existe un CH ou non,
5:       À l'expiration du timer_1
6:       if receive(Dist_slot)==true then
7:         Etat  $\leftarrow$  Member Vehicle // Il passe à l'état membre
8:         Vehicle_member_collection() // Il exécute l'algorithme de collecte par les véhicules
          membres
9:       else
10:        Read ( $T_l, T_r, T_i$ )
11:         $center_x \leftarrow Cal\_center(Num\_seg)$ 
12:         $center_y \leftarrow R_{width} \div 2$ 
13:         $Distance \leftarrow Dist\_to\_center(center_x, center_y, X_v, Y_v)$ 
14:         $Rapp \leftarrow Distance * (T_l + T_r / T_l + T_r + T_i)$ 
15:        Déclenche un timer_2 pendant Rapp
16:        À l'expiration du timer_2
17:        if receive(Dist_slot)==false then
18:          Etat  $\leftarrow$  ClusterHead
19:          Send_paquet_Dist_slot(id_ch, Vector, Ack)
20:        else
21:          Etat  $\leftarrow$  Member vehicle
22:        end if
23:      end if
24:    end if
25:  end while
26: end if

```

Où timer_1 c'est le premier Back-off déclenché pour tester s'il existe un ClusterHead ou non, et timer_2 c'est le deuxième Back-off déclenché afin de choisir le véhicule le plus proche du centre et le plus stable.

Les avantages de notre algorithme d'élection

Notre algorithme d'élection présente plusieurs avantages, on peut citer :

- Il permet de sélectionner l'un des véhicules comme ClusterHead, sans échange préalable de messages entre les véhicules.
- La sélection du ClusterHead est dynamique et dès que le ClusterHead courant quitte le segment, détecté par la non réception d'un Dist_slot, un autre véhicule parmi les véhicules qui sont sur le même segment sera sélectionné comme ClusterHead.
- Il garantit toujours, à un moment donné, qu'au plus il y a un et un seul ClusterHead par segment de collecte, ce qui est garanti par le premier Back-off, puisque tous les nouveaux venus sur un segment de collecte doivent déclencher un premier Back-off, afin de tester s'il existe un ClusterHead ou non, s'il existe ils annulent leurs Back-offs et ils passent à l'état membre.

4.1.4 La collecte et l'agrégation des données

Après l'élection des ClusterHeads sur les segments de collecte, les véhicules membres commencent à envoyer leurs données collectées à tour de rôle, pendant un time-slot, vers les ClusterHeads. Cependant, les questions qui doivent se poser maintenant sont : Dans un réseau fortement dynamique, comment rendre la collecte de données **efficace**, en collectant le maximum de données en minimum de temps ?, quel mécanisme faut-il utiliser pour garantir un échange de données **fiable** et avec peu d'erreurs ?

La réponse à ces deux questions nous a permis de proposer deux mécanismes centralisés à l'aide des ClusterHeads,

- **Une gestion de slots dynamique**, pour cela, nous avons proposé une technique d'accès au canal basée sur SDMA que l'on a nommée **D-SDMA (Dynamic-SDMA)**, qui consiste à affecter d'une manière dynamique, grâce aux ClusterHeads, un nombre de slots qui correspond exactement au nombre de véhicules actifs. Ainsi, grâce au champ MD (More Data) intégré dans le paquet Data, détaillé dans la section suivante, permet aux ClusterHeads d'affecter des time-slots seulement aux véhicules membres possédant des données à envoyer, ce qui permet de collecter le maximum de données en le minimum du temps.
- **La retransmission de données en cas d'erreurs**, ce deuxième mécanisme permet la retransmission de données en cas d'erreurs et ceci grâce au tableau Ack intégré dans le paquet Dist_slot (détaillé dans la section suivante).

Avant de détailler le processus de collecte des données par le ClusterHead et par les véhicules membres, nous présentons les formats de paquets utilisés pendant cette phase de collecte.

Les formats de paquets utilisés

Durant cette phase de collecte, nous avons défini deux types de paquets : Le paquet Dist_slot envoyé seulement par le ClusterHead et utilisé pour l'affectation des times-slots de façon dynamique aux différents véhicules membres. ce paquet comprend trois champs :

- **Id_v** : L'identifiant du véhicule ClusterHead,
- **Vector (B+1)** : C'est un tableau dont la taille est $B+1$, qui est égale au nombre total des blocs par segment plus un bit, qui correspond toujours au time-slot réservé au ClusterHead. Chaque bit de ce tableau correspond à un numéro de bloc dans le segment, et chaque véhicule dans un bloc se voit attribuer un time-slot si et seulement si son bit correspond dans le tableau Vector est égale à 1 (Voir FIGURE 4.5).

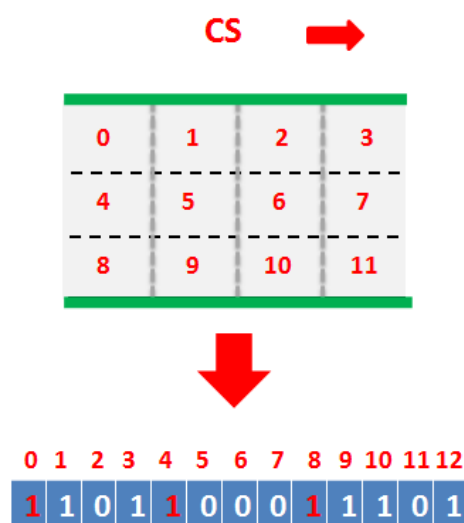


FIGURE 4.5 – Le tableau Vector.

De plus, afin de prendre en considération les nouveaux venus et de leur affecter des time-slots, nous avons fixé un nombre M , M c'est le nombre des voies qui est égale à 3 dans cet exemple, de bits à des uns, les bits en rouge du tableau dans la figure 4.5.

- **Ack (B-bits)** : C'est un autre tableau dont la taille est égale à B , où B est le nombre des blocs sur un segment. Ce tableau est utilisé afin d'accuser la réception des paquets Data depuis les véhicules membres et donc la retransmission de ces paquets en cas d'erreur.

Le paquet Data envoyé seulement par les véhicules membres vers leurs ClusterHeads et contient leurs données collectées. Ce paquet est composé de quatre champs :

- **Id_v (Plusieurs bits)** : L'identifiant du véhicule membre,
- **Num_bloc (2 octets)** : Le numéro du bloc sur lequel il se trouve le véhicule membre,
- **Data (1024-bits)** : Les données collectées par le véhicule membre,
- **MD (1-bit)** : Ce dernier champ est utilisé par le véhicule membre pour indiquer au ClusterHead s'il a d'autres données à envoyer, pour que le ClusterHead lui affecte un time-slots lors de la prochaine itération¹ ou non, et donc de ne pas perdre un time-slot si le véhicule membre ne possède pas des données à envoyer.

La collecte par le ClusterHead

Dès qu'un véhicule est élu comme ClusterHead, il diffuse un paquet `Dist_slot` qui comprend son identifiant et les deux tableaux Vector et Ack.

Le premier message `Dist_slot` envoyé par le ClusterHead

Pour la première itération, la réservation des slots est statique, c'est-à-dire que le ClusterHead avant qu'il diffuse le paquet `Dist_slot`, il met tous les bits du tableau Vector à 1, sauf le bit qui correspond à son numéro du bloc courant. Par conséquent, il affecte à chaque bloc un time-slot même si le bloc est vide.

Réception d'un paquet de données par le ClusterHead

À chaque réception d'un paquet Data par le ClusterHead, il exécute une des :

1. Une itération est le temps nécessaire pour envoyer les données de tous les véhicules une et une seule fois.

- **Si le paquet reçu est erroné** : Le ClusterHead affecte aux deux cases, qui correspondent au numéro du bloc intégré dans le paquet Data reçu, de deux tableaux Vector et Ack 1 et 0 respectivement.
- **Si le paquet reçu n'est pas erroné** : Le ClusterHead teste le champ *MD* (More Data) du paquet Data, si $MD = 1$ implique que le véhicule membre a d'autres données à envoyer, donc le ClusterHead lui réserve un time-slot lors de la prochaine itération, et met $Vector[Num_bloc] = 1$, sinon il met $Vector[Num_bloc] = 0$ puisque le véhicule membre n'a pas des données à envoyer.
Ainsi que, pour les deux cas, $MD = 1$ ou $MD = 0$, le ClusterHead affecte 1 à $Ack[Num_bloc]$ puisque les données reçues ne sont pas erronées.

Enfin, à l'instant d'accès au canal du ClusterHead et qui correspond toujours à la fin d'une itération, il met le bit qui correspond à son numéro du bloc courant à zéro. Par la suite, il diffuse un paquet Dist_slot.

L'algorithme suivant récapitule la collecte de données par le ClusterHead :

Algorithm 2 ClusterHead Collection(Pseudo code)

```

1: Attendre  $\leftarrow$  slot_attendre (B, Vector) // la durée à attendre
2: Déclenche un timer pendant (Attendre)
3: while (le timer n'a pas expiré) do
4:   if (receive(Data)==true) then
5:     Erreur  $\leftarrow$  calcul_erreur (paquet Data)
6:     if (Erreur) then
7:       Vector [num_bloc]  $\leftarrow$  1;
8:       Ack [num_bloc]  $\leftarrow$  0;
9:     else
10:      if ( $MD == 1$ ) then
11:        Vector [num_bloc]  $\leftarrow$  1;
12:        Ack [num_bloc]  $\leftarrow$  1;
13:      else
14:        Vector [num_bloc]  $\leftarrow$  0;
15:        Ack [num_bloc]  $\leftarrow$  1;
16:      end if
17:    end if
18:  end if
19: end while

```

Un exemple illustratif

La figure 4.6 illustre un exemple de réservation du tableau Vector lors des deux premières itérations, tout en supposant que les paquets Data ont été bien reçus par le ClusterHead ainsi que leur *MD* (More Data) égale à 1 :

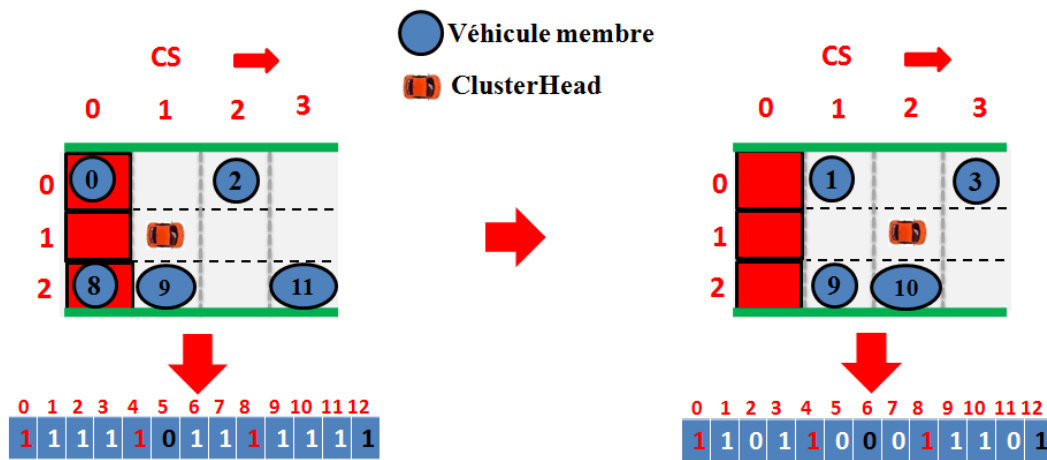


FIGURE 4.6 – Un exemple de réservation du tableau Vector, itération 1 et itération 2.

- **Itération 1** : Après l'élection, le ClusterHead diffuse un paquet `Dist_slot` comprend son identifiant et les deux tableaux Vector et Ack. En effet, tous les bits du tableau Vector sont des uns sauf celui qui correspond au numéro du bloc courant du ClusterHead, puisque l'instant d'accès du ClusterHead est toujours à la fin de l'itération et qui correspond au dernier bit, celui qui est en noir, du tableau Vector. De plus, les trois uns en rouge correspondent aux numéros de blocs des nouveaux venus.
- **Itération 2** : Après la diffusion du premier paquet `Dist_slot`, le ClusterHead écoute le canal et à chaque fois qu'il reçoit un paquet `Data`, il réserve le bit correspond du tableau Vector. Ensuite, à la fin de l'itération on aura 8 uns, 3 bits (qui sont en rouge) correspondent aux numéros de blocs des nouveaux venus, 4 bits (qui sont en blanc) pour les véhicules membres, 1 bit (qui est en noir) correspond au time-slot du ClusterHead. Ainsi, le bit du tableau Vector qui correspond au numéro du bloc courant du ClusterHead est mis toujours à zéro (celui qui est en noir).

La collecte par les véhicules membres

Pour faire face à la forte mobilité des véhicules, chaque véhicule membre lors de l'envoi d'un paquet `Data` sauvegarde son numéro de bloc courant, pour le comparer avec le prochain paquet `Dist_slot` envoyé par le ClusterHead. De plus, il sauvegarde aussi les données envoyées dans le but de réaliser la retransmission en cas d'erreurs.

La réception d'un paquet `Dist_slot` depuis le ClusterHead

À la réception d'un paquet `Dist_slot`, chaque véhicule membre qui trouve le bit du tableau Vector qui correspond à son ancien numéro de bloc égale à un, calcule son instant d'accès au medium en fonction du tableau Vector et son ancien numéro de bloc, et il teste le bit du tableau Ack qui correspond toujours à son ancien numéro de bloc et exécute selon le cas :

- S'il trouve `Ack [anc_num_bloc]` égale à zéro : Implique que le ClusterHead a détecté une erreur lors de l'itération précédente, et par conséquent le véhicule membre doit retransmettre le paquet `Data`.
- S'il trouve `Ack [anc_num_bloc]` égale à un : Implique que le ClusterHead n'a pas détecté une erreur lors de l'itération précédente, et par conséquent le véhicule membre doit

transmettre un nouveau paquet Data.

Par exemple, dans la figure 4.6, si on suppose que tous les véhicules n'ont pas changé de bloc pendant deux itérations consécutives, d'après le tableau Vector, le véhicule 3 doit attendre deux slots avant qu'il accède au canal, celui réservé en cas de nouveau venu et celui du véhicule 1 ($Vector[0] = 1$ et $Vector[1] = 1$). De plus, on remarque bien que le ClusterHead, le véhicule rouge, doit attendre 7 slots, quatre slots pour les véhicules membres 1, 3, 9 et 10 et trois slots réservés en cas de nouveaux venus, puisque on lui réserve toujours le dernier slot, le bit du tableau Vector qui est en noir, $Vector[12]$. Puis, chaque véhicule déclenche un Back-off pendant la durée calculée précédemment avant qu'il puisse accéder au canal à l'expiration de ce Back-off. le pseudo code 3 illustre la procédure à exécuter par chaque véhicule membre lors de la collecte :

Algorithm 3 vehicle member Collection(Pseudo code)

```

1: if ( $Vector[Ancien\_bloc] == 1$ ) then
2:   Attendre  $\leftarrow$  slot_attendre (Ancien_bloc, vector) // la durée à attendre
3:   if ( $Ack[Ancien\_bloc] == 0$ ) then
4:     La détection d'une erreur
5:   else
6:     la non détection d'une erreur
7:     if (has_other_data==true) then
8:       MD  $\leftarrow$  1
9:     else
10:      MD  $\leftarrow$  0
11:    end if
12:  end if
13:  Déclenche un timer pendant (Attendre)
14:  À l'expiration du timer
15:  Accéder au canal
16: end if

```

4.2 Le protocole TCDGP (Token based Clustered Data Gathering Protocol)

Nous avons proposé un autre protocole de collecte de données, TCDGP, Token based Clustered Data Gathering Protocol, autoroutier. Dans le protocole TCDGP, nous avons fixé les mêmes objectifs à savoir : **l'équité, l'efficacité et la fiabilité** de communication et les mêmes hypothèses que l'on a pris en considération dans le protocole CDGP.

Ainsi, le protocole TCDGP se déroulera aussi en deux grandes phases, (i) phase de propagation du message de collection (CMP) et (ii) phase d'agrégation et de dissémination de données (DAD), et partage avec le protocole CDGP la même phase CMP ainsi que le même algorithme d'élection des ClusterHeads.

Cependant, la différence entre les deux protocoles réside dans l'étape de collecte et d'agrégation de données que nous allons la détailler dans la section suivante.

4.2.1 La collecte et l'agrégation des données dans TCDGP

Comme dans le protocole CDGP, pour garantir l'**équité** entre les véhicules et offrir une collecte de données **efficace**, nous nous sommes basés sur notre méthode d'accès au canal **D-SDMA** grâce aux ClusterHeads. Ainsi, pour rendre la communication inter-véhicule **fiable**, nous avons mis en œuvre aussi les deux mécanismes de gestion des slots et de retransmission de données en cas d'erreurs.

De plus, lors de la collecte, le protocole TCDGP est basé sur le principe du réseau en anneau, Token ring. Pour cela, nous avons défini un nouveau paquet Token manipulé et envoyé seulement par les ClusterHeads et comprend essentiellement trois champs, l'identifiant du véhicule ClusterHead (Id_CH), le numéro du bloc destiné par ce paquet (Num_bloc), et le champ Ack d'un seul bit utilisé afin d'accuser la réception des données.

Le principe générale du TCDGP

Le principe générale de ce protocole consiste à générer et envoyer un paquet Token par le ClusterHead vers le premier bloc, et attendre la réception d'un paquet Data au maximum pendant le sixième d'un time-slot. Si le ClusterHead ne reçoit pas des données pendant cette durée, il régénère un nouveau Token, il l'envoie vers le prochain bloc jusqu'au dernier bloc du son segment de collecte. Parallèlement à cela, il réserve les deux tableaux Vector et Ack en fonction des paquets Data reçus. À noter que, dans ce protocole, les deux tableaux Vector et Ack, ceux qui implémentent les deux mécanismes de la gestion des slots et de la retransmission de données, sont manipulés au niveau du véhicule ClusterHead.

Dans ce qui suit, nous allons présenter la collecte des données par le ClusterHead puis par les autres véhicules membres.

La collecte par le ClusterHead

Dans la première itération et comme dans le premier protocole, la réservation des slots est statique, c'est-à-dire que le ClusterHead affecte à tous les blocs de son segment de collecte un time-slot. En commençant par envoyer un paquet Token vers le premier bloc et il attend le sixième d'un time-slot, s'il commence à recevoir des données il attend la fin du slot et il vérifie si les données sont érronées ou non afin de positionner les bits des tableaux Vector et Ack. Par la suite, il régénère un nouveau paquet Token et il l'envoie vers le prochain bloc. Sinon, s'il ne reçoit pas des données pendant un sixième du time-slot, il réserve les deux tableaux, il régénère un nouveau paquet Token et il l'envoie vers le prochain bloc, et ainsi de suite jusqu'au dernier bloc dans le segment.

Le pseudo code 4 illustre les tâches effectuées par le ClusterHead lors de la collecte :

Algorithm 4 ClusterHead Collection(Pseudo code)

```

1: Send (Token)
2: Déclenche un timer pendant  $(1/6 * t_{slot})$ 
3: while (expired_timer==false) do
4:   if (receive (Data)==true ) then
5:     Erreur  $\leftarrow$  calcul_erreur (paquet Data)
6:     if (Erreur) then
7:       Vector [num_bloc]  $\leftarrow$  1
8:       Ack [num_bloc]  $\leftarrow$  0
9:     else
10:      if ( $MD == 1$ ) then
11:        Vector [num_bloc]  $\leftarrow$  1
12:        Ack [num_bloc]  $\leftarrow$  1
13:      else
14:        Vector [num_bloc]  $\leftarrow$  0
15:        Ack [num_bloc]  $\leftarrow$  1
16:      end if
17:    end if
18:    Send (Token)
19:  end if
20: end while
21: if (receive(Data)==false) then
22:   Vector [num_bloc]  $\leftarrow$  0
23:   Ack [num_bloc]  $\leftarrow$  1
24:   Send (Token)
25: end if

```

De plus, à partir de la deuxième itération et comme dans le premier protocole, le ClusterHead envoie le paquet Token seulement vers les blocs dont les bits correspondent dans le tableau Vector égale à un.

La collecte par les véhicules membres

Comme dans le premier protocole, chaque véhicule membre lors de l'envoi d'un paquet Data sauvegarde son numéro de bloc courant, pour le comparer avec le champ Num_bloc de chaque paquet Token reçu. De plus, il sauvegarde aussi les données envoyées dans le but de réaliser la retransmission en cas d'erreur.

À la réception d'un paquet Token depuis le ClusterHead, chaque véhicule membre qui trouve le champ Num_bloc, intégré dans le paquet Token, égale à son ancien numéro de bloc, il teste le champ Ack, intégré aussi dans le paquet Token, s'il le trouve à zéro, il retransmet les données, sinon il transmet un nouveau paquet Data.

Le pseudo code 5 illustre les tâches à exécuter par les véhicules membres, lors de la collecte de données :

Algorithm 5 Vehicle member Collection(Pseudo code)

```
1: if (receive(Token)==true) then
2:   if (Ancien_num_bloc == Num_bloc) then
3:     if (Ack == 0) then
4:       Send (Ancien_Data_paquet) // la retransmission des données
5:     else
6:       Send (New_Data_paquet)
7:     end if
8:   end if
9: end if
```

4.3 Comparaison avec les autres protocoles de collecte

Par rapport aux autres protocoles de collecte de données, dans nos deux protocoles, nous nous sommes basés sur une architecture en clusters dont la segmentation de la route est statique. Lors de l'élection des ClusterHeads, on choisit toujours le véhicule le plus proche du centre de segment et le plus stable comme ClusterHead, dans le but d'augmenter la durée de vie des ClusterHeads et d'éviter l'élection fréquente.

De plus, pour garantir une certaine équité entre les véhicules et offrir une collecte efficace, nous nous sommes basés sur la méthode d'accès au canal D-SDMA que l'on a proposée, avec un mécanisme de retransmission de données en cas d'erreur, afin de rendre la communication véhicule à véhicule plus fiable.

Dans la TABLE 4.1, nous présentons une étude comparative entre les protocoles de collecte vus dans le chapitre précédent et nos deux protocoles selon les critères de classification cités dans le chapitre précédent.

TABLE 4.1 – Comparaison entre nos deux protocoles et les autres protocoles de collecte.

Les critères	CGP	Road Probing	Traffic Gather	COL	CDGP et TCDGP
Architecture en clusters	Oui	Non	Oui	Non	Oui
La segmentation de la route	Statique	—	Dynamique	—	Statique
Mode de communication	V2V/V2I	V2V/V2I /I2V	V2V	V2V	V2V/V2I I2V
Accès au canal	CSMA/CA	—	SDMA statique	N'est pas spécifiée	D-SDMA
Le critère de sélection du CH	Proche de l'extrémité de fin de segment	Temps passé sur les voies intermédiaires	Le plus éloigné du véhicule relai	—	Proche du centre
Détection et correction des erreurs	Non	Non	Non	Non	Oui

4.4 Conclusion

Les deux protocoles CDGP et TCDGP garantissent trois propriétés pouvant s'avérer avantageuses dans un réseau, qui se caractérise par une forte mobilité des nœuds. La première de ces propriétés réside dans le fait de permettre aux véhicules un accès équitable et sans collisions au canal grâce à la technique SDMA. La deuxième mise en évidence est l'efficacité de la collecte, en utilisant la technique D-SDMA proposée dans ce travail. Enfin, la troisième repose sur le fait de mettre en œuvre un mécanisme de retransmission de données en cas d'erreur et donc d'améliorer la fiabilité de communication inter-véhicules.

Dans ce chapitre, nous avons présenté nos deux protocoles de collecte, CDGP et TCDGP, au sein des réseaux véhiculaires. Nous avons commencé, pour chaque protocole, par donner une brève description ainsi que les hypothèses prises en considération et les objectifs fixés. Puis, nous avons détaillé les deux grandes phases dont se composent nos deux protocoles. Enfin, on a terminé par comparer nos deux protocoles et les autres protocoles existants vus dans le chapitre précédent.

Nous décrivons dans le prochain chapitre les scénarios de simulation ainsi que les résultats obtenus.

Chapitre 5

Simulations et résultats

Dans ce chapitre, nous allons réaliser un ensemble de simulations dans le but d'étudier les performances des nos deux protocoles, CDGP (Clustered Data Gathering Protocol) et TCDGP (Token-based Clustered Data Gathering Protocol), en termes de fiabilité de communication inter-véhicules et de retransmission de données erronées. Pour cela, nous avons simulé, pour chaque protocole, trois scénarios différents, afin d'effectuer une étude comparative entre les résultats obtenus dans chaque scénario. Puis, nous avons analysé les performances des nos deux protocoles CDGP et TCDGP avec d'autres protocoles de collecte, où on a choisi le protocole CGP (Clustered Gathering Protocol) [8].

Dans ce qui suit, nous allons tout d'abord décrire l'environnement de simulation utilisé, à savoir le langage Java et le modèle de mobilité simulé. Puis, nous présentons les scénarios et les paramètres de simulation, ainsi que les résultats obtenus.

5.1 La simulation

Dans cette section, nous présentons l'environnement de simulation choisi et le modèle de mobilité simulé. Ensuite, nous abordons les scénarios et les paramètres de simulation des nos deux protocoles.

5.1.1 Environnement de simulation

Pour simuler CDGP et TCDGP nous avons utilisé le langage Java [71]. Le langage Java est devenu un moyen de programmation le plus simple et le plus souple par rapport à n'importe quelle alternative, même si les problèmes augmentent en taille et en complexité. Java permet d'écrire de façon simple et claire des programmes portables sur la majorité des plates-formes. Lié à l'essor du World Wide Web, il a été conçu par l'équipe de James Gosling¹ en fonction des multiples exigences des développements informatiques actuels.

1. Un informaticien canadien qui est considéré comme le concepteur du langage Java.

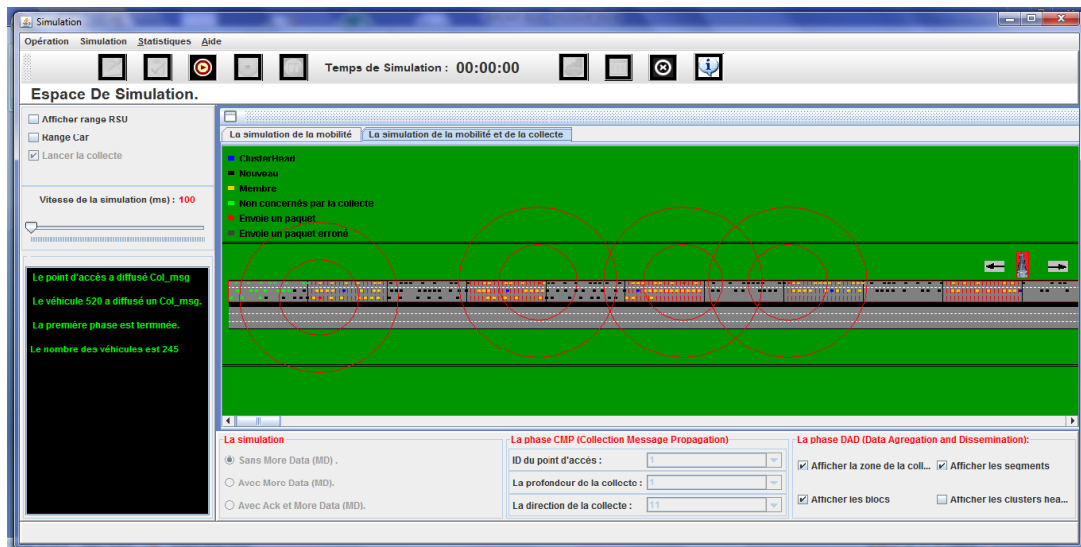


FIGURE 5.1 – La fenêtre principale de notre simulation.

Pour développer nos deux protocoles, nous avons créé une application en Java à l'aide de l'environnement de programmation Eclipse [72]. Afin de paramétrer notre simulation, l'interface graphique de cette application comprend plusieurs éléments tels que : une barre de menu englobant toutes les fonctionnalités de simulation, une barre d'outils qui regroupe des boutons d'accès rapides aux différentes fonctionnalités de l'application, un espace de simulation contenant trois zones, deux réservées aux paramétrages de simulation, à gauche et en bas et la troisième destinée à l'affichage graphique de la carte au centre de l'espace (voir FIGURE 5.1).

5.1.2 Le modèle de mobilité

Nous avons simulé nos deux protocoles sur une autoroute. Pour cela, nous avons implémenté le modèle de mobilité Freeway qui comprend trois voies, dans la même direction. Lors de la simulation, les véhicules sont distribués aléatoirement le long de la route et la vitesse de chaque véhicule est générée aléatoirement entre V_{min} et V_{max} .

5.1.3 Les paramètres de simulation

Les scénarios de simulation sont exécutés sur une route droite de 4 Km partitionnée en 40 segments, 20 segments de collecte et 20 segments de silence, d'une même longueur, 100 mètres (voir FIGURE 5.2). Chaque segment de collecte est divisé en B blocs de 5 mètres.

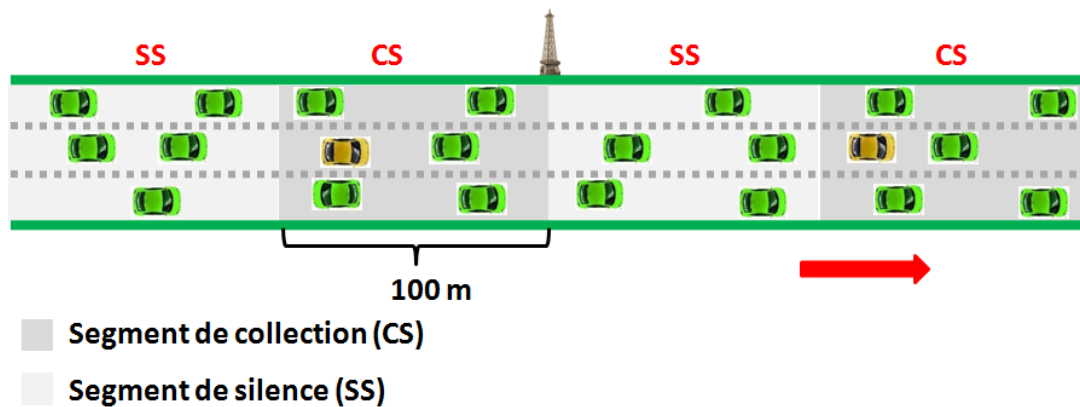


FIGURE 5.2 – Le partitionnement de la route.

La TABLE 5.1 résume les différents paramètres pris en considération lors de la simulation :

TABLE 5.1 – Les paramètres de la simulation.

Paramètres	Valeurs
Modèle de mobilité	Freeway
Longueur de la zone de collecte	4 Km
Nombre des voies	3
Longueur des segments (L_s)	100 m
Longueur des blocs (L_b)	5 m
Portée de communication des RSUs	1 Km
Vitesse des véhicules	[50-170] Km/h
Portée de communication des véhicules	100 m
Nombre des véhicules	[50-300]
Durée d'un time-slot	1 ms
Taille de paquets de données	1500 Octets

5.1.4 Les scénarios de simulation

Afin d'évaluer nos deux protocoles et d'analyser leurs résultats en terme de fiabilité de communication véhicule à véhicule, nous avons pris en compte trois scénarios différents :

1. Scénario 1 : sans MD (More Data) et sans Ack

Dans ce scénario, on n'a pas intégré le champ MD dans le paquet Data, c.à.d. que chaque ClusterHead sur un segment de collecte affecte à chaque véhicule membre un time-slot, même si ce dernier ne possède pas des donnée à envoyer. De plus, on suppose aussi qu'aucun mécanisme de retransmission des données erronées n'est utilisé dans ce scénario (Sans Ack).

2. Scénario 2 : avec MD et sans Ack

Dans ce scénario, on a intégré le champ MD dans le paquet Data, c.à.d. qu'on a utilisé notre technique proposée D-SDMA, qui permet à chaque ClusterHead d'affecter un time-slot à

un véhicule membre, possédant des données à envoyer. Cependant, aucun mécanisme de retransmission n'est utilisé lors de la réception de données erronées (Sans Ack).

3. Scénario 3 : avec MD et Ack

Dans ce dernier scénario, on a simulé nos deux protocoles, comme ils sont présentés dans le chapitre précédent.

Les métriques choisies pour comparer les performances des nos deux protocoles dans chaque scénario sont : le nombre de messages échangés entre les véhicules, qui représente aussi le nombre de slots consommés, et le nombre de paquets retransmis lors de la réception des paquets erronés.

5.2 Résultats de simulation et analyse des performances

Nous présentons dans ce qui suit les résultats de simulations, ainsi que leurs interprétations. On notera que les résultats de simulations correspondent à quatre cas de densité de véhicules, 50, 100, 150 et 300 véhicules.

5.2.1 Analyse des performances du protocole CDGP

- Le nombre de messages échangés

La figure 5.3 représente le nombre de slots attribués par le CH aux véhicules membres, dans les trois scénarios, en fonction de la densité de véhicules, qui représente aussi le temps nécessaire pour la transmission des données.

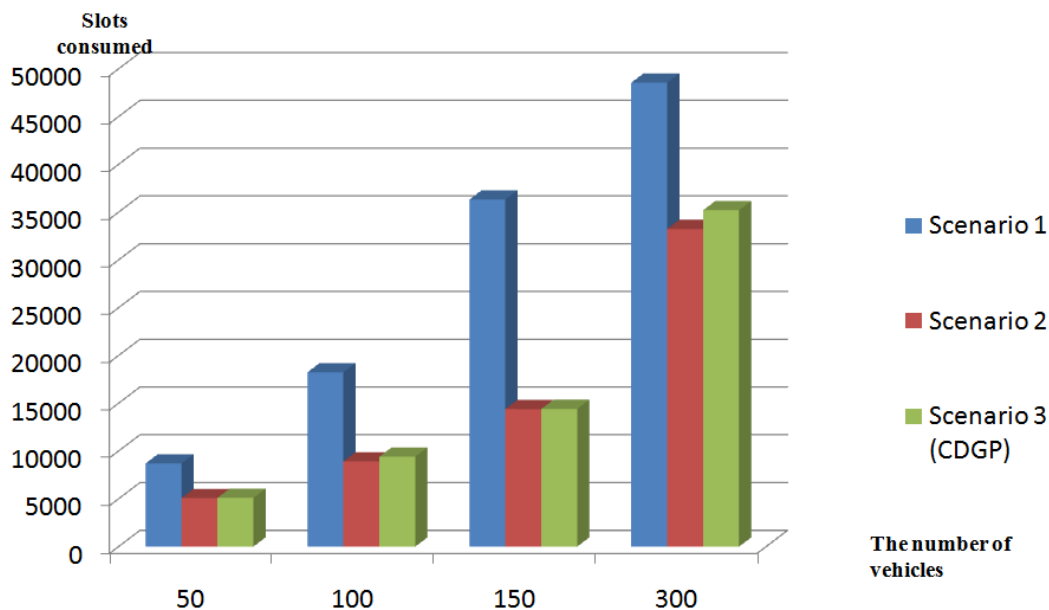


FIGURE 5.3 – Le nombre de slots consommés pour une communication V2V.

On peut bien remarquer depuis cette figure que le nombre de slots attribués dans les deux scénarios 2 et 3 est inférieur à celui du premier scénario, ceci est dû à l'intégration du champ MD

dans le paquet Data, qui nous permet d'appliquer notre technique d'accès au canal D-SDMA. Par conséquent, il nous permet aussi de diminuer le temps nécessaire pour la transmission des données. On remarque aussi que généralement le nombre de slots attribués dans le scénario 3 est supérieur à celui dans le scénario 2, ce qui est justifié par l'utilisation du mécanisme de retransmission de données dans le scénario 3, qui permet aux CHs d'affecter des nouveaux time-slots aux véhicules membres, lors de la détection des paquets Data erronés.

- Le nombre de messages erronés/retransmis

La figure 5.4 représente le nombre de messages erronés et le nombre de message retransmis en fonction de la densité de véhicules. Il est bien évident qu'on parle dans ce cas des résultats obtenus dans le scénario 3.

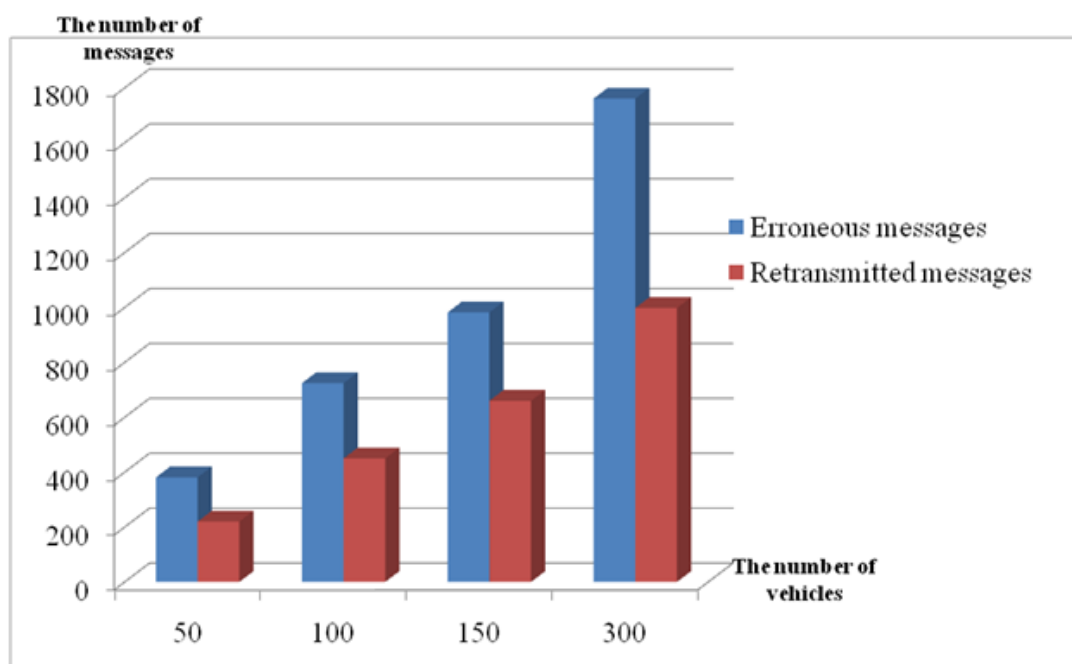


FIGURE 5.4 – La retransmission des messages.

Comme on peut remarquer, le pourcentage de retransmission des paquets Data erronés est proportionnel à la densité de véhicules (58 % pour une la densité de 50 véhicules, 62 % et 67 % pour une densité de véhicule de 100 et 150 véhicules respectivement). Cependant, le nombre de retransmission diminue (57 %) pour le cas d'une densité de 300 véhicules, ce qui peut être expliqué que parfois les véhicules membres, source d'erreurs, quittent le segment de collecte avant qu'ils réalisent la retransmission de données.

5.2.2 Analyse des performances du protocole TCDGP

- Le nombre de messages échangés

La figure 5.5 représente le nombre de slots attribués par le CH aux véhicules membres, dans les trois scénarios, en fonction de la densité de véhicules.

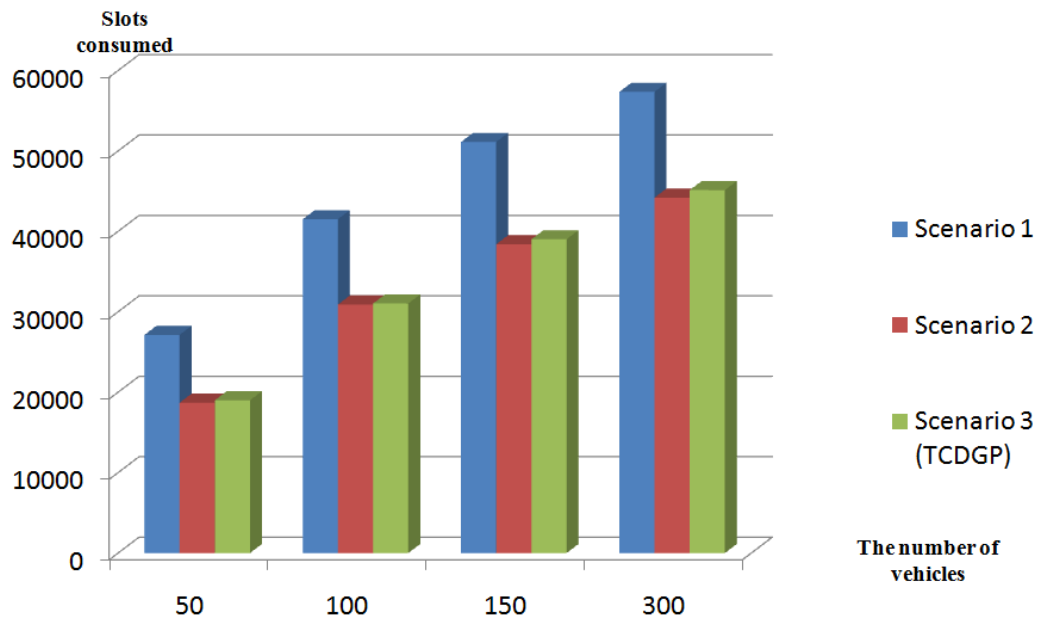


FIGURE 5.5 – Le nombre de slots consommés pour une communication V2V.

L'interprétation de cette figure est pratiquement la même que celle de la figure 5.3 du protocole CDGP, où on a obtenu un nombre de slots consommés dans les deux scénarios 2 et 3 inférieur à celui dans le scénario 1, grâce au champ MD. De plus, le nombre de slots consommés dans le scénario 3 est supérieur à celui dans le scénario 2, ceci est dû à l'utilisation de notre mécanisme de retransmission de données dans scénario 3.

- Le nombre de messages erronés/retransmis

La figure 5.6 représente le nombre de messages erronés et le nombre de messages retransmis en fonction de la densité de véhicules.

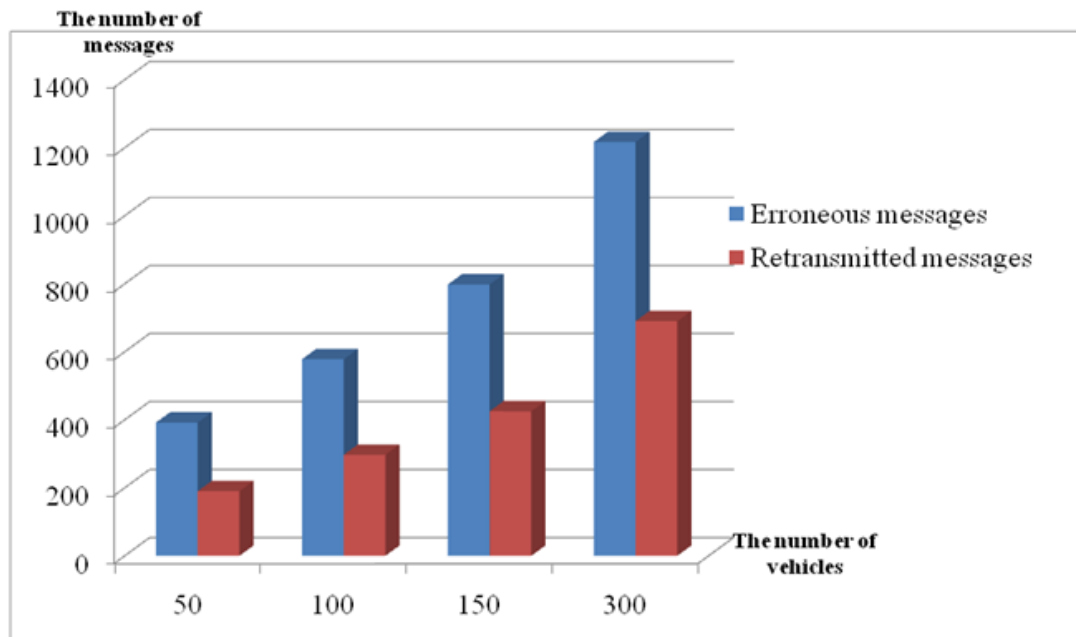


FIGURE 5.6 – La retransmission des messages.

On voit bien que le pourcentage de retransmission des paquets erronés est proportionnel à la densité de véhicules, 48 % pour 50 véhicules, 51 % et 53 % pour une densité de véhicule de 100 et 150 véhicules respectivement, et 56 % pour le cas d'une densité de 300 véhicules. Cependant, le pourcentage de retransmission des paquets n'a pas dépassé 56 %, ce résultat peut être expliqué par l'apparition de deux cas possibles, soit les véhicules membres quittent le segment de collecte avant qu'ils retransmettent leurs paquets de données, soit que le CH quitte le segment de collecte, avant qu'il informe les véhicules membres qu'il a détecté des erreurs et que ces derniers doivent réaliser la retransmission des paquets erronés.

5.2.3 Comparaison entre nos deux protocoles CDGP et TCDGP

Pour réaliser une étude comparative entre nos deux protocoles, nous avons choisi comme métrique, le nombre de slots consommés pour une communication V2V, lors de l'échange des paquets Data et Dist_slot dans le premier protocole, Data et Token dans le deuxième protocole, et le nombre de slots perdus dans chaque protocole.

- Le nombre de slots consommés

La figure 5.7 représente le nombre de slots consommés pour une communication V2V, en fonction de la densité des véhicules.

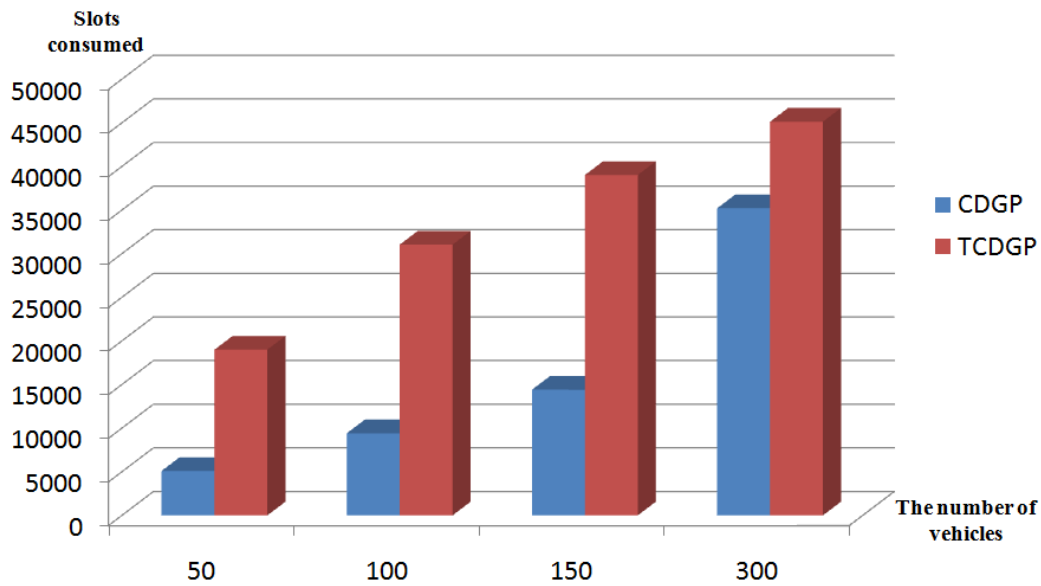


FIGURE 5.7 – Le nombre de slots consommés pour une communication V2V.

De cette figure, nous constatons que le nombre de slots consommés dans le protocole CDGP est inférieur à celui dans le protocole TCDGP, quelque soit la densité de véhicules (le pourcentage de réduction, 74% pour 50 véhicules, 70% et 64% pour une densité de véhicule de 100 et 150 véhicules respectivement, et 40% pour 300 véhicules), ceci est expliqué par le fait que dans le protocole TCDGP, le ClusterHead envoie un paquet Token à chaque bloc indépendamment lors d'une itération tandis que dans le protocole CDGP, le ClusterHead envoie seulement un seul paquet, Dist_slot, à la fin de chaque itération pour affecter les time-slots aux différents blocs, ce qui diminue le nombre total des paquets échangés et par conséquent les slots consommés.

- Le nombre de slots perdus

La figure 5.8 représente le nombre de slots perdus dans chaque protocole, en fonction toujours de la densité de véhicules.

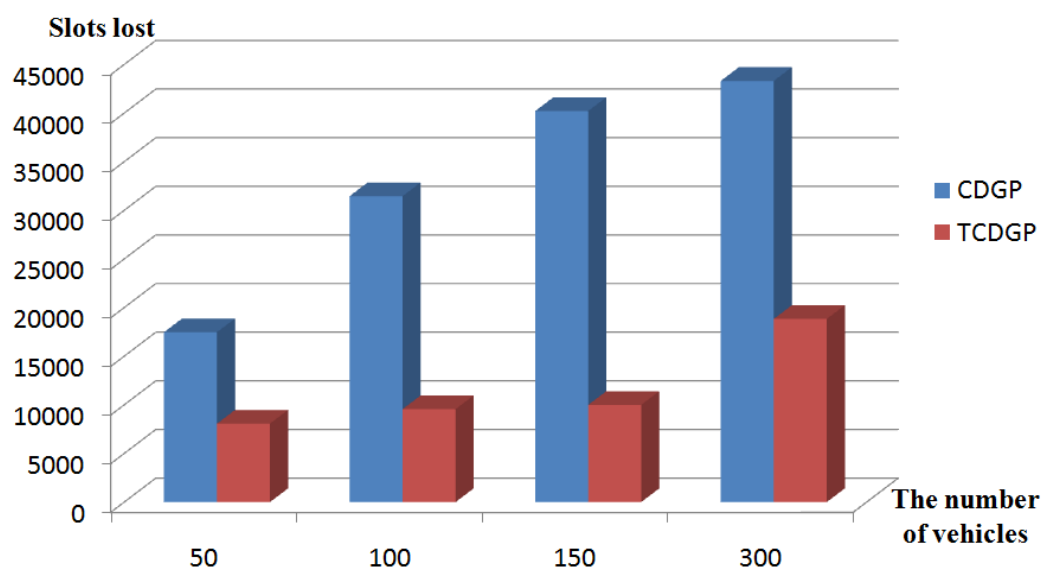


FIGURE 5.8 – Le nombre des slots perdus.

À la différence de la figure 5.7, les résultats obtenus dans la figure 5.8 montre que le nombre des slots perdus dans le protocole CDGP est supérieur à celui dans le protocole TCDGP, ce qui est justifié par le fait que dans le protocole CDGP lorsque le CH affecte un time-slot à un bloc vide, le slot est considéré comme perdu, par contre dans le protocole TCDGP, lorsque le CH envoie un paquet Token vers un bloc vide, il attend seulement le sixième du time-slot, s'il ne reçoit pas des données pendant cette durée, il régénère un autre Token et il l'envoie vers le prochain bloc, c.à.d. que seulement un sixième du time-slot sera perdu et non pas toute la durée d'un slot comme dans CDGP.

D'après ces résultats, on constate que le protocole CDGP par rapport au protocole TCDGP, présente l'avantage que moins de messages seront échangés entre les ClusterHeads et les véhicules membres, mais avec un nombre plus grand de slots perdus. Par contre, dans le protocole TCDGP, plus de messages seront échangés mais avec peu de slots perdus.

La TABLE 5.2 illustre une étude comparative entre nos deux protocoles CDGP et TCDGP.

TABLE 5.2 – Comparaison entre nos deux protocoles.

Protocoles	Les unités de données échangées	Le nombre de messages échangés(les slots consommés)	Les slots perdus
Le protocole CDGP	Data, Dist_slot	Petit	Grand
Le protocole TCDGP	Data, Token	Grand	Petit

5.2.4 Comparaison avec le protocole CGP

Pour évaluer les performances de nos deux protocoles avec d'autres protocoles de collecte, nous avons choisi le protocole CGP (Clustered Gathering Protocol). Ce choix est justifié par plusieurs raisons, parmi elles : les auteurs dans le protocole CGP se sont basés aussi sur une architecture en clusters et dont la segmentation est aussi statique, ils ont proposé aussi un algorithme d'élection des CHs afin de réaliser la collecte de données dans les segments.

Dans un premier temps et comme nous avons appliqué le même algorithme d'élection dans CDGP et TCDGP, nous avons choisi comme critère, le nombre de réélections des ClusterHeads dans le protocole CGP et dans notre protocole CDGP. La figure 5.9 représente le nombre de réélection des ClusterHeads en fonction de la densité de véhicules.

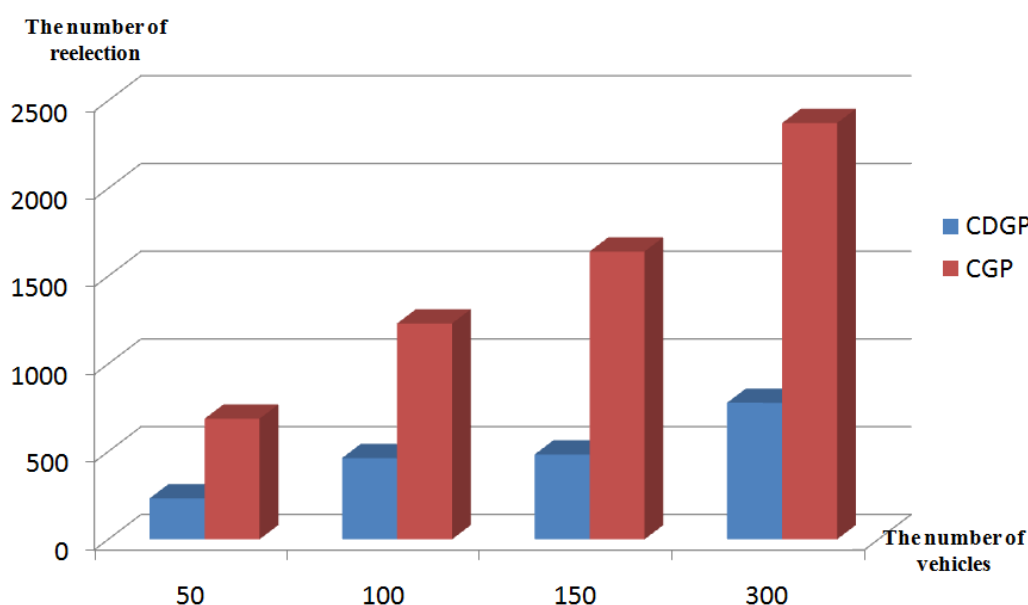


FIGURE 5.9 – Le nombre de réélection des ClusterHead.

Les résultats obtenus dans la figure 5.9 montrent que dans tous les cas de densités, le nombre de réélection des CHs dans notre protocole CDGP est inférieur à celui dans le protocole CGP, 66% pour 50 véhicules, 62% et 70% pour une densité de véhicule de 100 et 150 véhicules respectivement, et 67% pour le cas d'une densité de 300 véhicules, ce résultat est justifié par le fait que dans le protocole CGP et lors de l'élection, ils choisissent toujours le véhicule le plus proche de l'extrémité de fin de segment, alors que dans notre protocole, on choisit le véhicule le plus proche du centre de segment et le plus stable ce qui augmente la durée de vie du ClusterHead et diminue le nombre de réélection.

La figure 5.10 représente le nombre de messages échangés pour une communication véhicule à véhicule dans nos deux protocoles CDGP et TCDGP et dans le protocole CGP, en fonction de la densité de véhicules, 50 véhicules pour une faible densité, 300 véhicules pour une densité moyenne et 1000 véhicules pour une forte densité.

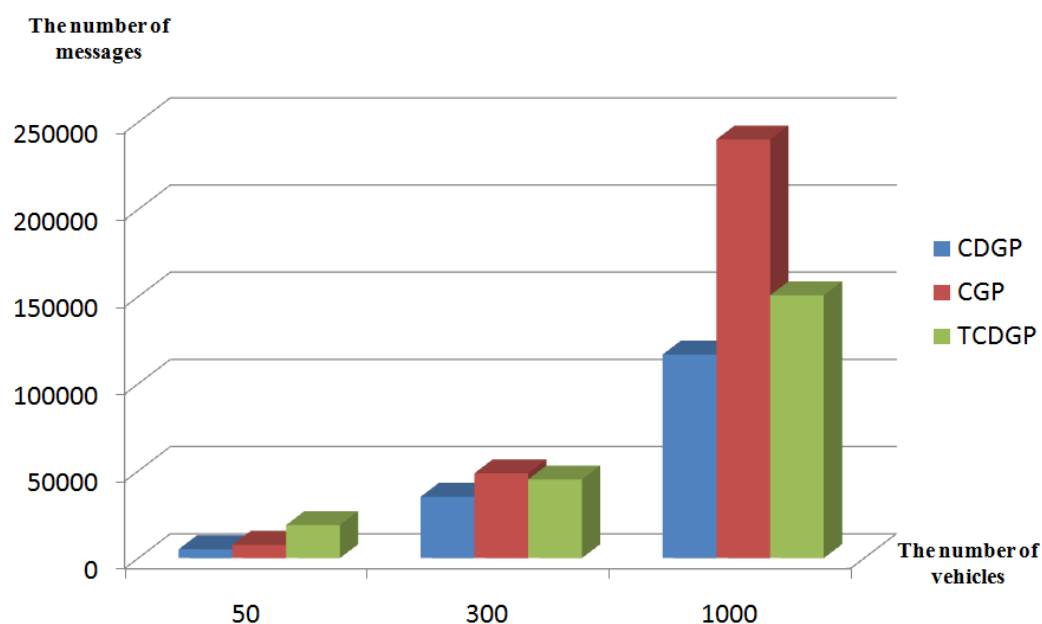


FIGURE 5.10 – Le nombre de messages échangés pour une communication V2V.

Les résultats obtenus dans la figure 5.10 montrent que le nombre de messages échangés dans notre premier protocole CDGP est inférieur à celui du protocole CGP pour les trois types de densité, ceci est justifié principalement par deux raisons :

- Lors de l'élection des ClusterHeads, l'algorithme utilisé dans CGP nécessite un échange de messages entre les véhicules afin de choisir un ClusterHead et avec le nombre élevé de réélection des ClusterHeads (voir FIGURE 5.9), beaucoup de messages doivent être échangés. Par contre, notre algorithme d'élection ne nécessite aucun échange de messages.
- Dans CGP, ils se sont basés sur la technique d'accès au canal CSMA/CA avec RTS et CTS, qui nécessite l'échange des deux paquets RTS (Request To Send) et CTS (Clear To Send) avant de commencer l'envoi des données collectées. Autrement dit, pour chaque communication, elle nécessite au moins quatre paquets, RTS, CTS, Data et Ack, ce qui cause l'augmentation du nombre de messages échangés.

On remarque aussi depuis la figure 5.10 que pour une faible densité, le nombre de messages obtenus dans CGP est inférieur à celui dans notre deuxième protocole TCDGP, car dans notre protocole TCDGP nous nous sommes basés sur le principe du réseau en anneau, où les ClusterHeads génèrent et envoient vers chaque bloc un Token. Une faible densité implique que trop de blocs sont vides et par conséquent un nombre important de messages Token seront envoyés par les ClusterHeads, ce qui permet d'augmenter le nombre total des messages. Pour les autres types de densité (moyenne et forte), le nombre de messages obtenus dans CGP est supérieur à celui du protocole TCDGP, ceci est dû principalement aux deux raisons citées précédemment.

5.3 Conclusion

Dans ce chapitre, nous avons évalué les performances de nos deux protocoles CDGP et TCDGP à travers un ensemble de simulations. Les résultats de simulation ont montré que le protocole CDGP présente l'avantage de minimiser le nombre de messages échangés mais avec un grand nombre de slots perdus. Par contre, le protocole TCDGP nécessite plus de messages échangés mais avec peu de slots perdus.

Nous avons aussi évalué les performances de nos deux protocoles, en les comparant avec le protocole CGP (Clustered Gathering Protocol). Nos deux protocoles ont donné des résultats plus performants surtout en terme d'efficacité de la collecte, de fiabilité de communication et en terme d'élection des CHs.

Conclusion générale

Les réseaux ad hoc véhiculaires (VANet) constituent une nouvelle application des réseaux Ad-hoc mobiles (MANet). Les particularités de ces réseaux véhiculaires proviennent des communications qui peuvent s'établir entre véhicules ou avec les stations de base (BSs).

L'utilisation des informations issues de plusieurs véhicules permet de couvrir l'imperfection et l'insuffisance de perception d'un seul véhicule sur son environnement et rend ces informations plus précises et plus pertinentes. Pour cela, l'objectif principale des protocoles de collecte de données, dans le contexte des réseaux véhiculaires, est de définir comment les données collectées seront échangées entre les véhicules, ou bien avec l'infrastructure, afin de fournir des informations, en temps réel ou non, plus fiables et plus pertinentes.

Dans ce mémoire, en profitant des inconvénients des protocoles de collecte existants, tels que la non implémentation d'un mécanisme fiable de transfert de données inter-véhicules, nous avons proposé deux protocoles de collecte de données dans les réseaux véhiculaires adaptés à l'environnement autoroutier, appelés CDGP (Clustered Data Gathering Protocol) et TCDGP (Token-based Clustered Data Gathering Protocol), en se basant sur des techniques de clustering. Ces deux protocoles offrent la possibilité aux différents RSUs déployés aux bords de l'autoroute de collecter des données depuis le réseau routier, en spécifiant la zone de la collecte et le type de données visées. Ensuite, ces données collectées peuvent être utilisées pour plusieurs applications telles que, le calcul de la densité de véhicules ou bien de la vitesse moyenne des véhicules sur un segment de route, la détection des espaces vides de stationnement dans un parking, la détection et le contrôle de la congestion, etc.

CDGP et TCDGP garantissent trois propriétés pouvant s'avérer avantageuses dans un réseau qui se caractérise par une forte mobilité des nœuds. La première de ces propriétés réside dans le fait qu'ils permettent aux véhicules un accès équitable et sans collisions au canal grâce à la technique SDMA. La deuxième mise en évidence est l'efficacité de la collecte, en proposant la technique D-SDMA. Enfin, la troisième repose sur le fait de mettre en œuvre un mécanisme de retransmission de données en cas d'erreurs et donc d'améliorer la fiabilité de communication inter-véhicule.

Les résultats de simulation montrent que CDGP et TCDGP sont plus performants en terme de fiabilité de communication inter-véhicules et d'efficacité de collecte de données, par rapport aux autres protocoles de collectes existants.

De nombreuses perspectives de recherche peuvent être envisagées. Dans ce qui suit, nous

présentons celles qui nous paraissent être les plus intéressantes :

- ◇ La modification du CDGP et TCDGP de telle sorte que la gestion des slots soit distribuée entre les véhicules sur les CSs, et non pas gérée seulement par les ClusterHeads.
- ◇ Le développement d'un protocole de collecte de données au niveau des zones urbaines, puisque tous les protocoles étudiés y compris nos deux protocoles sont développés à base d'un modèle Freeway.
- ◇ La proposition de techniques d'agrégation de données au niveau des véhicules lors de la collecte, afin de limiter le nombre de messages envoyés et d'économiser la consommation de la bande passante.

Annexe A

.1 Le protocole CDGP(Clustered Data Gathering Protocol)

Dans cet annexe, nous allons présenter les différents pseudo codes mis en œuvre, afin de développer notre protocole CDGP [10].

Les variables globaux :

- $L_s = R = 300$ mètres ;
- $T_{slot} = 1$ ms ;
- T_{left} : Le temps du trajet sur la voie la plus à gauche ;
- T_{right} : Le temps du trajet sur la voie la plus à droite ;
- T_{inter} : Le temps du trajet sur les voies intermédiaires ;
- DT : Le type de données visé ;
- DC : La direction de la collecte ;
- VT : La durée de validation du paquet de la collecte Col ;
- A_{length} : La profondeur de la collecte ;
- R_{width} : La largeur de la route ;
- $Voie_{width}$: La largeur de chaque voies ;
- MD : More Data initialisé à 1 ;
- $L_{bloc} = 5$ mètres ;
- M : Nombre des voies ;
- N : Nombre de blocs par voie ;
- B : Nombre total des blocs ($N * M$) sur un segment ;

.1.1 La phase de propagation de message de collecte(CMP)

Le pseudo code 6 est exécuté par le RSU, qui initié la collecte de données en diffusant un paquet Col.

Algorithm 6 Start collection

- 1: $Id_{rsu} = ID$;
 - 2: $POS_{rsu} = POS_GPS_{rsu}$;
 - 3: $A_{length} = 4Km$;
 - 4: $DC = 11$;
 - 5: $DT =$ la vitesse ;
 - 6: $VT = 5$ minutes ;
 - 7: Send (Col paquet) ;
-

Le pseudo code 7 est exécuté par les véhicules, afin de faire propager le paquet Col dans toute la zone de la collecte, et informé les autres véhicules du déclenchement du processus de la collecte.

Algorithm 7 Col packet propagation

```

1: if ( $Active\_col == faux$ ) then
2:   if ( $X_v > POS_{rsu}$ ) then
3:     if ( $X_v < POS_{rsu} + A_{length}$ ) then
4:        $Active\_col \leftarrow vrai$ ;
5:       if ( $DC == 11$ ) then
6:          $DC \leftarrow 01$ ;
7:       end if
8:        $POS\_fin\_zone \leftarrow POS_{rsu} + A_{length}$ ;
9:        $Flood\_col\_msg(Pos_v, POS_{rsu}, POS\_fin\_zone, DC)$ ;
10:    end if
11:   else
12:     if ( $X_v > POS_{rsu} - A_{length}$ ) then
13:        $Active\_col \leftarrow vrai$ ;
14:       if ( $DC == 11$ ) then
15:          $DC \leftarrow 10$ ;
16:       end if
17:        $POS\_fin\_zone \leftarrow POS_{rsu} - A_{length}$ ;
18:        $Flood\_col\_msg(Pos_v, POS_{rsu}, POS\_fin\_zone, DC)$ ;
19:     end if
20:   end if
21: end if

```

Le pseudo code 8 illustre la procedure de diffusion du paquet Col par les véhicules, Flood_col_msg, appelée dans le pseudo 7.

Algorithm 8 Flooding Col packet

Require: $Pos_v, POS_{rsu}, POS_{fin_zone}, DC$

```

1: if ( $DC == 01$ ) then
2:    $Dist\_to\_fin \leftarrow POS_{fin\_zone} - Posx_v$ ;
3:   Déclenche un timer pendant ( $Dist\_to\_fin$ );
4:   if (receive (Col_msg)==true) then
5:     Stop timer;
6:   else
7:     À l'expiration du timer,
8:     if ( $Pos_{xv} < POS_{fin\_zone}$ ) then
9:       Send (Col packet);
10:    end if
11:  end if
12: else
13:    $Dist\_to\_fin \leftarrow Posx_v - POS_{fin\_zone}$ ;
14:   Déclenche un timer pendant ( $Dist\_to\_fin$ );
15:   if (receive (Col_msg)==true) then
16:     Stop timer;
17:   else
18:     À l'expiration du timer,
19:     Send (Col packet);
20:   end if
21: end if

```

.1.2 La phase d'agrégation et de dissémination de données (DAD)

L'algorithme d'élection des ClusterHeads est illustré dans le pseudo code 9.

Algorithm 9 ClusterHead Election(Pseudo code)

```

1: if receive(Col)==true then
2:   while is_in_collection_area(v) do
3:     if is_in_collection_segment(v) then
4:       Déclenche un timer_1 pendant B slots //Afin de tester s'il existe un CH ou non,
5:       À l'expiration du timer_1
6:       if receive(Dist_slot)==true then
7:         Etat ← Member Vehicle // Il passe à l'état membre
8:         Vehicle_member_collection() // Il exécute l'algorithme de la collecte par les véhi-
           cules membres
9:       else
10:        Read ( $T_l, T_r, T_i$ )
11:         $center_x \leftarrow Cal\_center(Num\_seg)$ 
12:         $center_y \leftarrow R_{width} \div 2$ 
13:         $Distance \leftarrow Dist\_to\_center(center_x, center_y, X_v, Y_v)$ 
14:         $Rapp \leftarrow Distance * (T_l + T_r / T_l + T_r + T_i)$ 
15:        Déclenche un timer_2 pendant Rapp
16:        À l'expiration du timer_2
17:        if receive(Dist_slot)==false then
18:          Etat ← ClusterHead
19:          Send_paquet_Dist_slot(id_ch, Vector, Ack)
20:        else
21:          Etat ← Member vehicle
22:        end if
23:      end if
24:    end if
25:  end while
26: end if

```

Le pseudo code 10 est exécuté par les ClusterHeads lors de la collecte de données.

Algorithm 10 ClusterHead Collection(Pseudo code)

```

1: Attendre  $\leftarrow$  slot_attendre (B, Vector) // la durée à attendre
2: Déclenche un timer pendant (Attendre)
3: while (le timer n'a pas expiré) do
4:   if (receive(Data)==true) then
5:     Erreur  $\leftarrow$  calcul_erreur (paquet Data)
6:     if (Erreur) then
7:       Vector [num_bloc]  $\leftarrow$  1;
8:       Ack [num_bloc]  $\leftarrow$  0;
9:     else
10:      if (MD == 1) then
11:        Vector [num_bloc]  $\leftarrow$  1;
12:        Ack [num_bloc]  $\leftarrow$  1;
13:      else
14:        Vector [num_bloc]  $\leftarrow$  0;
15:        Ack [num_bloc]  $\leftarrow$  1;
16:      end if
17:    end if
18:  end if
19: end while

```

La collecte de données par les véhicules membres est illustrée dans le pseudo code 11.

Algorithm 11 vehicle member Collection(Pseudo code)

```

1: if (Vector[Ancien_bloc] == 1) then
2:   Attendre  $\leftarrow$  slot_attendre (Ancien_bloc, vector) // la durée à attendre
3:   if (Ack[Ancien_bloc] == 0) then
4:     La détection d'une erreur
5:   else
6:     la non détection d'une erreur
7:     if (has_other_data==true) then
8:       MD  $\leftarrow$  1
9:     else
10:      MD  $\leftarrow$  0
11:    end if
12:   end if
13:   Déclenche un timer pendant (Attendre)
14:   À l'expiration du timer
15:   Accéder au canal
16: end if

```

La procédure qui permet aux véhicules de calculer la durée à attendre avant qu'ils accèdent au canal est illustrée dans le pseudo code 12.

Algorithm 12 vehicle member Collection(Pseudo code)

Require: $Pos_v, POS_{rsu}, POS_{fin_zone}, DC$

```
1: Nombre_slot  $\leftarrow$  :=0;  
2: I  $\leftarrow$  :=0;  
3: while ( $I \neq bloc$ ) do  
4:   if ( $vector[I] == 1$ ) then  
       Nombre_slot ++;  
5:   end if I ++;  
6: end while  
7: return Nombre_slot;
```

.2 Le protocole TCDGP(Token-based Clustered Data Gathering Protocol)

Dans cette section, nous allons présenter les différents pseudo codes mis en œuvre, afin de développer notre deuxième protocole TCDGP. TCDGP partage avec le protocole CDGP les mêmes pseudo codes utilisés durant la première phase de propagation de message de la collecte, ainsi que le même pseudo d'élection des ClusterHeads. Dans ce qui suit, nous présentons seulement les pseudo utilisés durant la phase d'agrégation et de dissémination de données.

.2.1 La phase d'agrégation et de dissémination de données (DAD)

Le pseudo code 13 illustre les tâches effectuées par le ClusterHead lors de la collecte :

Algorithm 13 ClusterHead Collection(Pseudo code)

```

1: Send (Token)
2: Déclenche un timer pendant  $(1/6 * t_{slot})$ 
3: while (expired_timer==false) do
4:   if (receive (Data)==true) then
5:     Erreur  $\leftarrow$  calcul_erreur (paquet Data)
6:     if (Erreur) then
7:       Vector [num_bloc]  $\leftarrow$  1
8:       Ack [num_bloc]  $\leftarrow$  0
9:     else
10:      if ( $MD == 1$ ) then
11:        Vector [num_bloc]  $\leftarrow$  1
12:        Ack [num_bloc]  $\leftarrow$  1
13:      else
14:        Vector [num_bloc]  $\leftarrow$  0
15:        Ack [num_bloc]  $\leftarrow$  1
16:      end if
17:    end if
18:    Send (Token)
19:  end if
20: end while
21: if (receive(Data)==false) then
22:   Vector [num_bloc]  $\leftarrow$  0
23:   Ack [num_bloc]  $\leftarrow$  1
24:   Send (Token)
25: end if

```

Le pseudo code 14 illustre les tâches à exécuter par les véhicules membres, lors de la collecte de données :

Algorithm 14 Vehicle member Collection(Pseudo code)

```

1: if (receive(Token)==true) then
2:   if ( $Ancien\_num\_bloc == Num\_bloc$ ) then
3:     if ( $Ack == 0$ ) then
4:       Send (Ancien_Data_paquet) // la retransmission des données
5:     else
6:       Send (New_Data_paquet)
7:     end if
8:   end if
9: end if

```

Bibliographie

- [1] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 46(2) :388–404, 2000.
- [2] M. McDonald. *Intelligent Transport Systems in Europe : Opportunities for Future Research*. World Scientific, 2006.
- [3] Panos Papadimitratos, Arnaud De La Fortelle, Knut Evensen, Roberto Brignolo, and Stefano Cosenza. Vehicular communication systems : enabling technologies, applications, and future outlook on intelligent transportation. *Comm. Mag.*, 47(11) :84–95, November 2009.
- [4] Y. Qian and N. Moayeri. Survey of media access control protocols for vehicular ad hoc networks. *IET Communications*, 5 :1619–1631, 2011.
- [5] W. Zhao and M. Ammar. Message ferrying : Proactive routing in highly-partitioned wireless ad hoc networks. In *Proceedings of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003)*. 308 - 314, March 2003.
- [6] Dharmendra Sutariya and Dr. S. N. Pradhan. Data dissemination techniques in vehicular ad hoc network. *International Journal of Computer Applications*, 8(10) :35–39, October 2010. Published By Foundation of Computer Science.
- [7] Wang-Rong Chang, Hui-Tang Lin, and Bo-Xuan Chen. Trafficgather : An efficient and scalable data collection protocol for vehicular ad hoc networks. In *Proceedings of the 5th IEEE international Conference on Consumer Communications and Networking*, pages 365–369, 2008.
- [8] Ismail Salhi, Mohamed Oussama Cherif, and Sidi Mohammed Senouci. A new architecture for data collection in vehicular networks. In *Proceedings of the IEEE international conference on Communications*, ICC'09, pages 2705–2710, 2009.
- [9] Jonathan Petit. *Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans-fil véhiculaires*. PhD thesis, Université de Toulouse, France, Juillet 2011.
- [10] Brik B. Lagraa N. Yagoubi MB., Lakas A. An efficient and robust clustered data gathering protocol (CDGP) for vehicular networks. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. Pages 69-74. Paphos, Cyprus, October 2012.
- [11] Soheila V. Bana and Pravin Varaiya. Space division multiple access (SDMA) for robust ad hoc vehicle communication networks. In *Proc. IEEE Intelligent Transportation Systems Conference*, pages 962–967, 2001.
- [12] Juha Korhonen. *Introduction to 3G Mobile Communications*. Artech House, Inc., 2003.
- [13] E. Hossain and K.K. Leung. *Wireless Mesh Networks : Architectures and Protocols*. Springer, 2008.

- [14] Abdallah Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données*. PhD thesis, l'Université de Franche-Comté, Besançon, France, Avril 2010.
- [15] Tia Gao, Dan Greenspan, Matt Welsh, Radford R. Juang, and Alex Alm. Vital signs monitoring and patient tracking over a wireless network. In *Proceedings of the 3rd International Conference on Information Communication Technologies in Health (ICICTH'05)*, 2005.
- [16] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2) :18–25, March 2006.
- [17] Paolo Bellavista, U. Lee, E. Magistretti, Mario Gerla, and A. Corradi. Dissemination and harvesting of urban data using vehicular sensing platforms. *IEEE Transactions on Vehicular Technology*, 2008.
- [18] J-P Hubaux. Vehicular networks : How to secure them. *MiNeMa Summer School, Klagenfurt*, Juillet 2005.
- [19] S. Jolapara K.A. Sanketh, K.P. Subbarao. I2V and V2V communication based vanet to optimize fuel consumption at traffic signals. In *Proceedings of 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Sept. 2010.
- [20] Hannes Hartenstein, K. Laberteaux, and Inc Ebrary. *VANET : vehicular applications and inter-networking technologies*. Wiley Online Library, 2010.
- [21] Smita Narendra Pathak and Urmila Shrawankar. Secured communication in real time vanet. In *Proceedings of the 2009 Second International Conference on Emerging Trends in Engineering & Technology, ICETET '09*, pages 1151–1155. IEEE Computer Society, 2009.
- [22] J. Rybicki W. Kiess and M. Mauve. On the nature of inter-vehicle communication. In *Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks*, pp. 493-502, 2007.
- [23] 802.11 draft standard for wireless lan medium access control (mac) and physical layer (phy) specification. 1997.
- [24] White Paper. DSRC technology and the DSRC Industry consortium (DIC) prototype team. January 2005.
- [25] IEEE standard 802.11p. IEEE standard for information technology — telecommunications and information exchange between systems — local and metropolitan area networks — specific requirements — part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6 : Wireless access in vehicular environments, July 2010.
- [26] IEEE 1609 family of standards for wireless access in vehicular environments. *available from IEEE Standards*, 2009.
- [27] Continuous air-interface, long and medium range - CALM, <http://www.calm.hu/>. Page consultée le 09/11/2012.
- [28] ETSI. <http://www.etsi.org/intelligenttransportsystems.aspx>. Page consultée le 11/11/2012.
- [29] Wagner C. Maihofer C. Hartenstein H. Franz, W. Fleetnet : Platform for inter-vehicle communications. *The 1st Intl. Workshop on Intelligent Transportation*, 2004.
- [30] C2C-CC : CAR 2 CAR Communication Consortium. <http://www.car-to-car.org/>. Page consultée le 26/06/2012.

- [31] Safe and comfortable driving based upon inter-vehicle communication, <http://www.cartalk2000.net>. Page consultée le 26/06/2012.
- [32] NoW : Network on wheels. <http://www.network-on-wheels.de>. Page consultée le 26/06/2012.
- [33] Voiture connectée : les états-unis lancent leur expérimentation. <http://pro.01net.com/editorial/571447/voiture-connectee-les-etats-unis-lancent-leur-experimentation/>. Page consultée le 11/11/2012.
- [34] J.Zhu and R. Pecun. A novel automatic utility data collection system using IEEE 802.15.4-compliant wireless mesh networks. In *Proc. of the IAJC-IJME International Conference, Paper 086*, 2008.
- [35] S. L. Shrestha et. al. An open wireless mesh testbed architecture with data collection and software distribution platform. In *Proceeding of 3rd international conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom)*, United States, 2007.
- [36] Conti M. Anastasi, G. and Di Francesco. Data collection in sensor networks with data mules : an integrated simulation analysis. In *Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC 2008)*.1096-1102, 2008.
- [37] Giuseppe Anastasi, Marco Conti, and Mario Di Francesco. Reliable and energy-efficient data collection in sparse sensor networks with mobile elements. *Perform. Eval.*, 66(12) :791–810, 2009.
- [38] Abdelzaher T. F. Kim, H. S. and W. H. 2003. Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*.193-204, 2003.
- [39] Shuai Gao, Hongke Zhang, and Sajal K. Das. Efficient data collection in wireless sensor networks with path-constrained mobile sinks. *IEEE Transactions on Mobile Computing*, 10(4) :592–608, 2011.
- [40] DAVID REY. *Collecte des données d'un réseau de capteurs sans fils en utilisant une source-couche réseau pair à pair*. PhD thesis, UNIVERSITÉ DE MONTRÉAL, Canada, Avril 2010.
- [41] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38 :393–422, 2002.
- [42] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1) :2–16, February 2003.
- [43] J. Kulik W. Heinzelman and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceeding of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, August 1999.
- [44] M. Jiang, J. Li, and Y. C Tay. Cluster based routing protocol (CBRP) functional specification. Technical report, 1998.
- [45] K. Dantu and Sukhatme. Connectivity vs. control : Using directional and positional cues to stabilize routing in robot networks. In *Proceedings of the 2nd International Conference on Robot Communication and Coordination (RoboComm 2009)*.1-6, 2009.
- [46] H. Ammari and Das. Data dissemination to mobile sinks in wireless sensor networks : an information theoretic approach. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2005)*. 86314., 2005.

- [47] Fall. A delay tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003)*. 27 - 34, 2003.
- [48] E. Ekici, Yaoyao Gu, and D. Bozdag. Mobility-based communication in wireless sensor networks. *Communications Magazine, IEEE*, 44(7) :56-62, 2006.
- [49] Basagni S. Melachrinoudis E. Wang, Z. M. and Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS 2005)*, 2005.
- [50] Pelagatti M. Dini, G. and Savino. An algorithm for reconnecting wireless sensor network partitions. In *Proceedings of the 5th European conference on Wireless Sensor Networks (EWSN 2008)*.253-267, 2008.
- [51] Torng E. Xing G. El-Moukaddem, F. and Kulkarni. Mobile relay configuration in data-intensive wireless sensor networks. In *Proceedings of the 6th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*.80-89, 2009.
- [52] Sabharwal A. Chakrabarti, A. and Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN 2003)*.129-145, 2003.
- [53] X. Yang and Vaidya. A wakeup scheme for sensor networks : Achieving balance between energy saving and end-to-end delay. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004)*.19-26, 2004.
- [54] Conti M. Anastasi, G. and Di Francesco. Data collection in sensor networks with data mules : an integrated simulation analysis. In *Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC 2008)*.1096-1102, 2008.
- [55] Somasundara A. Jea D. Srivastava M. Kansal, A. and Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2004)*.111-124, 2004.
- [56] Conti M. Gregori E. Spagoni C. Anastasi, G. and Valente. Motes sensor networks in dynamic scenarios : an experimental study for pervasive applications in urban environments. *International Journal of Ubiquitous Computing and Intelligence*, 1(1), April 2007.
- [57] Ye F. Cheng J. Lu S. Luo, H. and Zhang. TTDD : two-tier data dissemination in large-scale wireless sensor networks. *Wireless Networks*, 11(1-2) :161-175, 2005.
- [58] Abdelzaher T. F. Kim, H. S. and Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*.193-204, 2003.
- [59] Kansal A. Jea D. Estrin D. Somasundara, A. and Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8) :1536-1233, 2006.
- [60] Zaydoun Yahya Rawashdeh and Syed Masud Mahmud. Toward strongly connected clustering structure in vehicular ad hoc networks. In *Proceedings of the 70th IEEE Vehicular Technology Conference : VTC2009-Fall, Anchorage, Alaska, USA*, September, 2009.
- [61] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11 : Wireless lan medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, 2007.

- [62] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Ström, and Urban Bilstrup. On the ability of the 802.11p MAC method and STDMA to support real-time vehicle-to-vehicle communication. *EURASIP J. Wirel. Commun. Netw.*, 9 :5 :1–5 :13, January 2009.
- [63] Nagaosa T and Hasegawa. Code assignment and the multicode sense scheme in an inter-vehicle CDMA communication network (special section on spread spectrum techniques and applications). *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 81(11), 2327-2333, 1998.
- [64] Uichin Lee, Eugenio Magistretti, Biao Zhou, Mario Gerla, Paolo Bellavista, and Antonio Corradi. Efficient data harvesting in mobile sensor platforms. In *In PERCOMW 06*, page 352. IEEE Computer Society, 2006.
- [65] Y. Chen S. Ni, Y. Tseng and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. ACM Intern. Conf. on Mobile Computing and Networking (MOBICOM), Seattle, USA, pp. 151-162*, 1999.
- [66] Lin Yang, Jingdong Xu, Gongyi Wu, and Jinhua Guo. Road probing : RSU assisted data collection in vehicular networks. In *Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing, WiCOM'09*, pages 2717–2720, 2009.
- [67] Y. Dieudonne, B. Ducourthial, and S.M. Senouci. COL : A data collection protocol for VANET. In *Intelligent Vehicles Symposium (IV), IEEE*, pages 711–716, June 2012.
- [68] Olga Brukman, Shlomi Dolev, Yinnon Haviv, and Reuven Yagel. Self-stabilization as a foundation for autonomic computing. In *Proceedings of the The Second International Conference on Availability, Reliability and Security*, pages 991–998. IEEE Computer Society, 2007.
- [69] Bertrand Ducourthial and Sébastien Tixeuil. Self-stabilizing global computations with r-operators. *Distributed computing*, 14(3) :147–162, 2001.
- [70] Farah El Ali and Bertrand Ducourthial. A light architecture for opportunistic vehicle-to-infrastructure communications. In *Proceedings of the 8th ACM international workshop on Mobility management and wireless access, MobiWac '10*, pages 60–67, 2010.
- [71] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *Java(TM) Language Specification, The (3rd Edition) (Java (Addison-Wesley))*. Addison-Wesley Professional, 2005.
- [72] Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, and Pat McCarthy. *Java(TM) Developer's Guide to Eclipse, The (2nd Edition)*. Addison-Wesley Professional, 2004.

Glossaire

Ack Acknowledge.

AHS Automated Highway System.

AP Access Point.

ASTM American Society for Testing and Materials.

BS Base Station.

BSS Basic Service Set.

CALM Continuous Air-interface, Long and Medium range.

CBRP Cluster Based Routing Protocol.

CDMA Code Division Multiple Access.

CDGP Clustered Data Gathering Protocol.

CGP Clustered Gathering Protocol.

CH ClusterHead.

CMP Collection Message Propagation.

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance.

CS Collection Segment.

CTS Clear To Send.

CV Cluster Vehicle.

DAD Data Aggregation and Dissemination.

DC Direction of the Collection.

DCF Distributed Coordination Function.

DD Directed Diffusion.

DSRC Dedicated Short Range Communication.

DT Data Type.

DTNs Delay Tolerant Networks.

D-SDMA Dynamic Space Division Multiple Access.

ES Exit Section.

ETSI European Telecommunications Standards Institute.

GPRS General Packet Radio Service.

GPS Global Positioning System.

GSM Global System for Mobile communications.

IBSS Independent Basic Service Set.

IEEE Institute of Electrical and Electronics Engineers.

ISO International Organization for Standardization.

ITS Intelligent Transportation System.

I2V Infrastructure to Vehicle.

MAC Medium Access Control.

MANet Mobile Ad-hoc Network.

MD More Data.

MV Member Vehicle.

NAHSC National Automated Highway System Consortium.

NOW Network-On-Wheels.

NTIC Nouvelles Technologies de l'Information et de la Communication.

OLSR Optimized Link State Routing.

PDA Personal Digital Assistant.

RM Request Message.

RSU Road Side Units.

RTS Request To Send.

RV Relay Vehicle.

SDMA Space Division Multiple Access.

SEAD Scalable Energy-efficient Asynchronous Dissemination.

SPIN Sensor Protocols for Information via Negotiation.

SS Silence Segment.

TCDGP Token-based Clustered Data Gathering Protocol.

TDMA Time Division Multiple Access.

TTDD Two-Tier Data Dissemination.

UMTS Universal Mobile Telecommunications System.

VANet Vehicular Ad-hoc Network.

VSN Vehicular Sensor Networks.

VT Validating Time.

V2I Vehicle to Infrastructure.

V2V Vehicle to Vehicle.

WAVE Wireless Ability in Vehicular Environments.

WEDAS Weighted Entropy DAta diSsemination.

Wi-Fi Wireless Fidelity.

WLAN Wireless Local Area Network.

WMN Wireless Mesh Networks.

WSN Wireless Sensor Networks.

WSN-ME Wireless Sensor Networks with Mobile Elements.

WWW World Wide Web.