



UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT D'INFORMATIQUE ET D'INGÉNIERIE

Capacités réactives d'un réseau VANET Cognitif

PAR

MUKTAR BAPPA

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION DU GRADE DE

PHILOSOPHIÆ DOCTOR (Ph. D.)

EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

JUIN 2021

© MUKTAR BAPPA, 2020

CETTE THÈSE INTITULÉE

Capacités réactives d'un réseau VANET cognitif

PRÉSENTÉE PAR

Muktar Bappa

A ÉTÉ ÉVALUÉ(E) PAR UN JURY COMPOSÉ DES PERSONNES SUIVANTES

Prof. Soulimane Berkane, Ph.D.
Université du Québec en Outaouais (UQO), Canada
Président du jury

Prof. Ilham Benyahia, Ph.D.
Université du Québec en Outaouais (UQO), Canada
Directrice de recherche

Prof. Luigi Logrippo, Ph.D.
Université du Québec en Outaouais (UQO), Canada
Membre du jury

Prof. Mohamed Shawky, Ph.D.
Université de Technologie de Compiègne (UTC), France
Examineur externe

ABSTRACT

Emerging technologies in different fields such as communications and sensors facilitate the development and deployment of solutions to tackle critical transport problems. Applications associated with such situations are part of Intelligent Transportation Systems (ITS). They are distributed and require specific Quality of Service (QoS).

Connected vehicles have exciting operational capabilities for critical ITS applications. These vehicles represent nodes of a VANET (Vehicular Ad-hoc Networks) network. On the other hand, these vehicles' environment presents various obstacles that can interrupt the communication signals they exchange. In an emergency, the resulting transmission delays may be catastrophic.

Within the framework of this Ph.D., our objective is to propose solutions that provide VANET networks with new capacities to ensure the efficiency of inter-vehicular communications for critical situations in an urban environment. To achieve this, we take the urban components of a road network infrastructure into account when disseminating information between connected vehicles. It should allow us to prevent or minimize transmission delays. We base our approach on modeling the connected vehicles as agents with cognitive abilities. The perception of obstacles by these vehicles and the initiation of actions will affect the quality of transmission.

In our experiment phase, we use multiple scenarios related to the operations of emergency vehicles. In addition, we estimate transmission errors using a supervised learning algorithm. The test results enabled us to collect information samples according to estimations by considering different situations at the transport network infrastructure level. These research works have helped us to innovate regarding VANET communication protocols, which consider vehicle cognitive behavior.

Keywords: QoS, Reliability, Delay, Routing, VANET, IEEE802.11p / WAVE, CL-AntHocNet protocol, cross-layer architecture, emergency vehicles, cognitive networks, STI.

By Muktar Bappa

RÉSUMÉ

Les technologies émergentes dans divers domaines tels que les communications et les senseurs encouragent le développement et le déploiement de solutions aux problèmes critiques dans le domaine des transports. Les applications associées à de telles situations font partie des systèmes de transports intelligents (STI). Elles sont distribuées et requièrent des qualités de services (QoS) spécifiques.

Les véhicules connectés présentent des capacités prometteuses au niveau des opérations des applications critiques des STI. Ces véhicules représentent des nœuds d'un réseau VANET (Vehicular Adhoc Networks). En revanche, l'environnement de ces véhicules présente divers obstacles qui peuvent perturber les signaux des transmissions qu'ils peuvent échanger. Les délais de transmissions qui en résultent peuvent s'avérer désastreux en cas d'urgence.

Dans le cadre de cette thèse, notre objectif consiste à proposer des solutions en vue de doter les réseaux VANET de nouvelles capacités permettant d'assurer la fiabilité des communications inter-véhiculaires pour les applications critiques dans un environnement urbain. Pour y parvenir, nous prenons en considération les composantes urbaines de l'infrastructure d'un réseau routier lors de la dissémination de l'information entre les véhicules connectés. Ceci doit permettre d'éviter ou de minimiser les retards de transmissions. Notre approche est basée sur la modélisation des véhicules connectés par des agents ayant des capacités cognitives. La perception des obstacles par ces véhicules et le déclenchement d'actions auront un impact sur la qualité des transmissions.

La phase d'expérimentations est basée sur plusieurs scénarios liés aux opérations des véhicules d'urgence. L'estimation des erreurs de transmission est basée sur un algorithme d'apprentissage par supervision. Les résultats issus des tests ont permis de collecter des échantillons d'informations par rapport aux estimations en considérant différentes situations au niveau de l'infrastructure du réseau des transports. Ces travaux de

recherche nous ont permis d'innover relativement aux protocoles de communication des réseaux VANET considérant des comportements cognitifs des véhicules.

Mots-clés : QoS, Fiabilité, Délai, Routage, Protocoles de communication, VANET, IEEE802.11p/WAVE, protocole CL-AntHocNet, architecture inter-couches, véhicules d'urgences, réseaux cognitifs, STI.

Par Muktar Bappa

REMERCIEMENTS

Tout d’abord, je ne pourrais adresser mes sincères remerciements sans référer à *Allah*, le Tout Puissant, le Très Miséricordieux, qui est l’architecte de ma vie ainsi que le pivot qui me donne la force et le courage d’avancer durant les périodes difficiles de mon existence.

Les travaux réalisés dans le cadre de cette thèse ont été effectués à l’Université du Québec en Outaouais (UQO), sous la supervision de Professeure *Ilham Benyahia* du département d’informatique et d’ingénierie. Je tiens à lui témoigner toute ma gratitude pour avoir accepté de m’encadrer. Sa patience, sa disponibilité et surtout ses judicieux conseils ont contribué à l’aboutissement de cette thèse.

J’adresse également mes remerciements au Département d’Informatique et d’Ingénierie ainsi qu’à ses professeurs et personnel d’appui, plus précisément aux professeurs *Souleimane Berkane*, *Marek Zaremba* et *Luigi Logrippo* qui ont accepté d’évaluer ce travail en tant qu’examineurs internes.

Mes remerciements à l’encontre de Professeur *Larbi Talbi* et Professeur *Shamsodin Tahéri* qui ont accepté de superviser respectivement mon atelier de recherche ainsi que le sujet secondaire de mon examen de synthèse.

Au même titre, je remercie également Professeur *Mohamed Shawky* de l’université Technologique de Compiègne (UTC) pour avoir accepté d’évaluer ce travail en tant qu’examineur externe.

Dans la même optique, je tiens à remercier mes amis et en particulier : *Dr. Vincent Fono*, *Channa Town*, *Farhad Khosrojerdi*, *Roberto Maoualé*, *Kosnodji Richard*, *Adama Nouboukpo*, *Sabrina Azzi* et tous les autres dont je n’aurais pas mentionné les noms.

Enfin, pour clore cette rubrique de remerciements, j'adresse ma profonde reconnaissance à tous ceux qui ont contribué de loin ou de près à l'aboutissement de cette thèse.



DÉDICACE

*À ma chère et tendre mère **Fatimé Alladoum Nadingar**, pour son soutien indéfectible sur tous les plans à mon égard,*

*À mon défunt père **Bappa Hamandjoda**, il y aura toujours une place pour toi dans mon cœur,*

*À ma femme **Marwa Bekrar**, pour sa patience et son soutien à mon égard durant cette période de thèse,*

*À mes beaux-parents **Djoudi Houria, Rachid Bekrar**, sans oublier **Amel** pour leurs encouragements,*

*À toute ma famille, et en particulier mon frère **Hissein Bappa**, mes sœurs **Zenabou Bappa, Mariamou Bappa et Zakiatou Bappa** pour leurs supports moral et spirituel,*

À tous mes compatriotes tchadiennes et tchadiens...

Je dédie cette thèse.

Liste des figures

Figure 2.1: Évanouissement par trajets multiples.....	34
Figure 2.2 : Avertissement d'un véhicule d'urgence pour éviter une congestion (Benyahia et al., 2017).....	35
Figure 4.1 : Phases d'apprentissage supervisé.....	73
Figure 4.2 : Environnement pour la collecte des données	74
Figure 4.3 : Processus de spécification du scénario urbain via SUMO.....	76
Figure 4.4 : Fonctions d'apprentissage testées sur MATLAB.....	84
Figure 4.5 : Modèle d'apprentissage pour la prévision du TEB.....	84
Figure 4.6 : Parcelle de régression du réseau pour l'apprentissage avec 10 neurones	86
Figure 4.7 : Prédiction du TEB sur 300 échantillons en utilisant 10 neurones.....	86
Figure 4.8 : Parcelle de régression du réseau pour l'apprentissage avec 15 neurones	88
Figure 4.9 : Prédiction du TEB sur 300 échantillons en utilisant 15 neurones.....	88
Figure 4.10 : Parcelle de régression du réseau pour l'apprentissage avec 20 neurones ..	90
Figure 4.11 : Prédiction du TEB sur 300 échantillons en utilisant 20 neurones.....	90
Figure 4.12 : Organisation architecturale du véhicule autonome.	93
Figure 4.13 : Perception des obstacles statiques dans CRV	93
Figure 4.14 : Structure dynamique de l'interaction des modules.	95
Figure 4.15 : processus d'identification des obstacles statiques.....	96
Figure 4.16 : Identification des obstacles statiques par un véhicule autonome.....	98
Figure 4.17 : Algorithme d'identification d'obstacles.....	99
Figure 4.18 : Algorithme traitement de la perception.....	100
Figure 4.19 : Algorithme traitement de la prédiction.	102
Figure 4.20 : Phase réactive pour l'initialisation de la route du protocole CRV.....	109
Figure 4.21: Réaction lors de la réception d'un événement	110
Figure 4.22: Processus découverte chemins alternatifs dans AntHocNet (Di Caro et al., 2004).	113
Figure 4.23: Processus découverte chemins alternatifs dans CRV.....	115
Figure 4.24: Algorithme phase proactive protocole CRV.....	116

Figure 4.25 : Architecture concurrente du nœud intelligent.....	117
Figure 4.26 : Architecture concurrente des processus de communication.	118
Figure 4.27 : Réseau VANET étudié.....	120
Figure 4.28 : Processus fonctionnement d'un véhicule autonome	128
Figure 5.1 : Fonctionnement protocole GPSR (Hu, Ding & Shi, 2012).....	135
Figure 5.2 : PCP CL-AntHocNet vs. Nœuds intermédiaires	145
Figure 5.3 : PCP GPSR vs. Nœuds intermédiaires	146
Figure 5.4 : PCP CRV vs. Nœuds intermédiaires	147
Figure 5.5 : TLPD vs. Flux du trafic (Scénario 1).....	153
Figure 5.6 : DBB vs. Flux du trafic (Scénario 1).....	154
Figure 5.7 : TLPD vs. Flux du trafic (Scénario 2).....	155
Figure 5.8 : DBB vs. Flux du trafic (Scénario 2).....	155
Figure 5.9 : TLPD vs. Flux du trafic (Scénario 3).....	156
Figure 5.10 : DBB vs. Flux du trafic (Scénario 3).....	157
Figure A.1 : composantes d'un réseau de neurones artificiels (Bre, Gimenez & Fachinotti, 2018).....	164

Liste des tableaux

Tableau 3.1 : Critères d'analyse des travaux de l'état de l'art	62
Tableau 4.1 : Choix de la méthode d'apprentissage (Russell & Norvig, 2010).....	71
Tableau 4.2 : Paramètres d'expérimentation sur NS-3.....	78
Tableau 4.3 : Résultats illustrant la simulation NS-3	79
Tableau 4.4 : Extrait des données d'entrées prétraitées pour l'apprentissage.....	81
Tableau 4.5 : Test des fonctions d'apprentissage	82
Tableau 4.6 : Précision du modèle basé sur 15 itérations d'apprentissage et 10 neurones cachés.....	85
Tableau 4.7 : Précision du modèle basé sur 15 itérations d'apprentissage et 15 neurones cachés.....	87
Tableau 4.8 : Précision du modèle basé sur 15 itérations d'apprentissage et 20 neurones cachés.....	89
Tableau 4.9 : Structure table de routage CRV.....	104
Tableau 4.10 : Illustration du contenu de la table de liens.	111
Tableau 4.11 : Table de liens de V1 à l'initialisation.....	121
Tableau 4.12 : Table de routage de V1 à l'initialisation.....	121
Tableau 4.13 : Table de routage de V1 lors du processus proactif avant interruption ..	122
Tableau 4.14 : Table de routage de V3 lors du processus proactif avant interruption ..	122
Tableau 4.15 : Table de routage de V5 lors du processus proactif avant interruption ..	122
Tableau 4.16 : Table de routage de V1 après exécution des modules de prédiction et localisation	123
Tableau 4.17 : Table de routage de V3 après exécution des modules de prédiction et localisation	124
Tableau 4.18 : Table de routage de V7 après exécution des modules de prédiction et localisation	124
Tableau 4.19 : Table de routage de V8 après exécution des modules de prédiction et localisation	125
Tableau 4.20 : Table de liens V1	125

Tableau 4.21 : Table de liens <i>V3</i>	126
Tableau 4.22 : Table de liens <i>V5</i>	126
Tableau 4.23 : Table de liens <i>V7</i>	126
Tableau 4.24 : Table de liens <i>V8</i>	126
Tableau 5.1 : Table de routage du nœud <i>V1</i> à $\Delta t1$	136
Tableau 5.2 : Table de routage du nœud <i>V7</i> à $\Delta t4$	136
Tableau 5.3 : Table de routage du nœud <i>V1</i> à $\Delta t1$	137
Tableau 5.4 : Table de routage du nœud <i>V7</i> à $\Delta t5$	138
Tableau 5.5 : Table de routage du nœud <i>V1</i> à $\Delta t1$	138
Tableau 5.6 : Table de routage du nœud <i>V7</i> à $\Delta t5$	139
Tableau 5.7 : Configuration des densités pour le scénario 1	142
Tableau 5.8 : Configuration des densités pour le scénario 2	142
Tableau 5.9 : Configuration des densités pour le scénario 3	143
Tableau 5.10 : Paramètres de simulation dans NS-3 pour l'étude empirique	144
Tableau 5.11 : PCP CL-AntHocNet STDMA VS PCP CL-AntHocNet 802.11p	145
Tableau 5.12 : PCP GPSR STDMA VS. PCP GPSR IEEE 802.11p	146
Tableau 5.13 : PCP CRV STDMA VS. PCP CRV 802.11p	146
Tableau 5.14 : Calibrage des paramètres de simulation NS-3.....	148
Tableau 5.15 : Paramètres de simulation dans NS-3.....	149
Tableau 5.16 : TLPD pour le scénario 1 (Grille de Manhattan).....	151
Tableau 5.17 : DBB pour le scénario 1 (Grille de Manhattan).....	151
Tableau 5.18 : TLPD pour le scénario 2 (Pont).....	151
Tableau 5.19 : DBB pour le scénario 2 (Pont)	151
Tableau 5.20 : TLPD pour le scénario 3 (tunnel).....	152
Tableau 5.21 : DDB pour le scénario 3 (tunnel).....	152

Liste des abréviations et sigles

A-STAR	<i>Anchor-based Street and Traffic Aware Routing</i>
AMR	<i>Adaptative Message Routing</i>
AODV	<i>Ad hoc On Demand Distance Vector</i>
AOMDV	<i>Ad-hoc On-demand Multipath Distance Vector</i>
CDMA	<i>Code Division Multiple Access</i>
CDP	<i>Cell Density Packets</i>
CFP	<i>Contention Free Period</i>
CRV	<i>Cognitive Routing Protocol for VANet</i>
CRV	<i>Cognitive Routing Protocol for VANET</i>
CSMA	<i>Carrier Sense Multiple Access</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CTS	<i>Clear To Send</i>
DCF	<i>Distributed Coordination Function</i>
DGR	<i>Directional Greedy Routing</i>
DIFS	<i>DCF Interframe Space</i>
DREAM	<i>Distance Effect Routing Algorithm for Mobility</i>
DRVC	<i>Direct and Relay protocol for Vehicle Communications</i>

DSDV	<i>Destination-Sequenced Distance Vector routing</i>
DSRC	<i>Dedicated Short Range Communications</i>
DTN	<i>Delay Tolerant Network</i>
DYMO	<i>Dynamic Manet on demand routing protocol</i>
E2ED	<i>End-to-End-Delay</i>
EDCA	<i>Enhanced Distributed Channel Access</i>
EDD	<i>Expected Disconnected Degree</i>
FDMA	<i>Frequency Division Multiple Access</i>
FER	<i>Frame-Error-Rate</i>
GPCR	<i>Greedy Perimeter Coordinator Routing</i>
GPS	<i>Global Positioning System</i>
GPSR	<i>Greedy Perimeter Stateless Routing</i>
GpsrJ+	<i>Enhanced Perimeter Routing for Geographic Forwarding Protocols</i>
GSR	<i>Geographic Source Routing</i>
GyTAR	<i>Greedy Traffic-aware Routing</i>
HyBR	<i>A hybrid bio-inspired bee swarm routing protocol</i>
IFS	<i>Inter Frame Spaces</i>
ITS	<i>Intelligent Transport System</i>
IVCs	<i>Inter-Vehicles Communications</i>
LLC	<i>Logical Link Control</i>

MAC	<i>Media Access Control</i>
MANET	<i>Mobile Ad-hoc Networks</i>
MCTTP	<i>Maximum Coverage with Time Threshold Problem</i>
MORA	<i>Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks</i>
MSE	<i>Mean Square Error</i>
MURU	<i>MUlti-hop Routing protocol for Urban vehicular ad hoc networks</i>
NS-3	<i>Network Simulator 3</i>
OLSR	<i>Optimized Link State Routing Protocol</i>
OSM	<i>OpenStreetMap</i>
PCF	<i>Point Coordination Function</i>
PDGR	<i>Predictive Directional Greedy Routing</i>
PDR	<i>Packet Delivery Rate</i>
PDVR	<i>Position-Based Directional Vehicular Routing</i>
PSO	<i>Particle Swarm Optimization</i>
QoS	<i>Qualité de Services</i>
QoS	<i>Quality of Service</i>
RNAs	<i>Réseaux de Neurones Artificiels</i>
RNCan	<i>Ressources Naturelles Canada</i>
RSUs	<i>Road Side Units</i>
RTS	<i>Ready To Send</i>

SIFS	<i>Short Inter Frame Spaces</i>
SiFT	<i>Simple Forwarding over Trajectory</i>
SINR	<i>Signal-to-Interference-plus-Noise Ratio</i>
SNR	<i>Signal-to-Noise Ratio</i>
STI	<i>Système des transports intelligents</i>
SUMO	<i>Simulation of Urban MObility</i>
TDMA	<i>Time Division Multiple Access</i>
TraCi	<i>Traffic Control Interface</i>
TS	<i>Time Slot</i>
V2I	<i>Vehicle-to-Infrastructure</i>
V2V	<i>Vehicle-to-Vehicle</i>
VADD	<i>Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks</i>
VANETS	<i>Vehicular Ad-hoc Networks</i>
VAV	<i>Véhicule-à-Véhicule</i>
WAVE	<i>Wireless Access in Vehicular Environments</i>

Table des matières

ABSTRACT	v
RÉSUMÉ	vii
REMERCIEMENTS	ix
DÉDICACE	xi
Liste des figures	xiii
Liste des tableaux.....	xv
Liste des abréviations et sigles.....	xviii
Table des matières.....	xxii
Chapitre 1 Introduction Générale.....	26
Chapitre 2 Contexte et Problématique	31
Sommaire du chapitre 2.....	31
2.1 Introduction.....	31
2.2 Contexte	32
2.3 Problématique	34
2.4 Conclusion	36
Chapitre 3 Revue de la Littérature sur les Protocoles de Routage pour les Réseaux VANET	37
Sommaire du chapitre 3.....	37
3.1 Le routage au niveau des réseaux VANET.....	37
3.1.1 Protocoles de routage basés sur les métaheuristiques.....	38
3.1.2 Protocoles de routage basés sur la géographie.....	47
3.1.3 Protocoles de routage basés sur le trafic.....	53
3.2 Protocoles de contrôle d'accès dans les réseaux VANET	56

3.3	Synthèse Générale.....	61
Chapitre 4 Vers un nouveau routage basé sur les capacités de véhicules autonomes pour les réseaux VANET		
	Sommaire du chapitre 4.....	66
4.1	Introduction.....	67
4.2	Principe de résolution des objectifs de recherche.....	68
4.2.1	Tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires	68
4.2.2	Optimiser le routage des communications véhiculaires pour les applications critiques	69
4.2.3	Tester notre solution dans un contexte de couche MAC optimale	69
4.3	Étude de cas : tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires	70
4.3.1	Choix des RNA.....	70
4.3.2	Choix de la méthode d'apprentissage	70
4.3.3	Prévision du taux d'erreurs binaires par apprentissage supervisée.....	72
4.3.4	Résultats de l'apprentissage pour la prévision du TEB	84
4.3.5	Analyse des résultats.....	90
4.4	Un Routage Cognitif pour les réseaux VANET	92
4.4.1	Organisation Architecturale du Véhicule Autonome.....	92
4.4.2	Rôle des différents modules du véhicule autonome	93
4.4.3	Structure dynamique de l'interaction des modules.....	94
4.4.4	Structure de la table de routage du protocole CRV	103
4.4.5	Principe de fonctionnement généralisé du protocole CRV.....	105
4.5	Conclusion	129

Chapitre 5 Routage Cognitif pour les réseaux VANET : Analyse de la performance et apports.....	131
Sommaire du chapitre 5.....	131
5.1 Introduction.....	132
5.2 Environnement de simulation	133
5.2.1 Choix de NS-3 comme simulateur réseaux.....	133
5.2.2 Choix de SUMO comme simulateur de mobilité urbaine.....	133
5.3 Étude du mécanisme de mise à jour de la table de routage de CRV	134
5.3.1 Scénario de simulation	134
5.3.2 Séquences de transmission.....	135
5.3.3 Discussion.....	140
5.4 Comparaison de CRV par rapport à GPSR et CL-AntHocNet.....	140
5.4.1 Scénarios de comparaison.....	141
5.4.2 Optimisation de la couche 2.....	143
5.4.3 Calibrage des paramètres de simulation NS-3	148
5.4.4 Paramètres de simulation NS-3.....	149
5.4.5 Métriques d'évaluation de performance	149
5.4.6 Résultats de la simulation	150
5.4.7 Analyse des résultats.....	152
5.5 Conclusion	157
Chapitre 6 Conclusions et perspectives de recherche.....	159
Sommaire du chapitre 6.....	159
6.1 Conclusions.....	159
6.2 Perspectives de recherche	161
6.2.1 Publications arbitrées dans des actes de colloque.....	162

6.2.2 Communication arbitrée.....	163
Annexe A Réseaux de Neurones Artificiels (RNA)	164
Annexe B Fonction apprentissage pour la prédiction du TEB	166
Annexe C Classe perception des obstacles en 2D	168
Annexe D Modélisation des obstacles dans NS-3	170
Bibliographie.....	172

Chapitre 1

Introduction Générale

Notre étude se situe dans le contexte des systèmes de transport intelligents (STI) ([Dimitrakopoulos & Demestichas, 2010](#); [Weiland & Purser, 2000](#)) qui se caractérisent par l'intégration de technologies de l'information (senseurs, communications, etc.) au niveau des réseaux des transports. Le principal objectif est d'optimiser la mobilité et la sécurité de tous les utilisateurs des réseaux des transports.

Certaines applications critiques des STI telles que la priorité de voie ou l'évitement de la congestion pour les ambulances s'appuient sur les réseaux VANET (Vehicular Ad hoc networks) ([Al-Sultan, Al-Doori, Al-Bayatti & Zedan, 2014](#); [Hartenstein & Laberteaux, 2008](#)). Ces applications nécessitent une qualité de service (QoS) pour le délai de transmission. En effet, les informations disséminées entre les véhicules doivent parvenir en temps opportun aux véhicules concernés. Des retards au niveau des transmissions peuvent avoir des conséquences désastreuses, surtout s'il s'agit d'une transmission impliquant une ambulance. En effet, plusieurs facteurs (interférences dues aux obstacles, forte mobilité des nœuds, etc.) peuvent affecter la qualité des transmissions, ce qui rend complexe la problématique des délais des transmissions, présentant ainsi un défi de recherche.

Nous considérons spécifiquement les réseaux VANET en intégrant les technologies émergentes CAV (Connected and Autonomous Vehicles) des STI. Ainsi, les nœuds des réseaux VANET que nous étudions sont représentés par des véhicules connectés et intelligents.

Les principaux services fournis par les réseaux VANET incluent la réduction du nombre d'accidents de la route tout en améliorant le trafic routier par le biais des

applications de contrôle du trafic et de gestion d'incidents. Les applications associées aux réseaux VANET nécessitent la dissémination d'informations entre les véhicules ou entre les véhicules et l'infrastructure.

Les cas que nous étudions dans le contexte des applications critiques des STI considèrent les véhicules d'urgence pour lesquels les délais de déplacement sont critiques. Tout retard au niveau des communications entre les véhicules relativement aux informations qui concernent le trafic, peuvent être inutiles ou même pénalisantes si elles manquent de précision. De telles communications peuvent cependant être altérées par des objets qui font partie de l'infrastructure du réseau routier comme les ponts, les bâtiments, etc.

Les protocoles de routage dans un réseau VANET jouent alors un rôle important pour trouver le bon chemin des messages entre les véhicules. Néanmoins, le risque des retards des transmissions augmente avec la présence d'obstacles qui demeurent partie intégrante des infrastructures. Ceci pourra certainement invalider le rôle des réseaux VANET pour les applications critiques malgré les évolutions au niveau des technologies de communications.

L'objectif principal de cette étude consiste à définir une nouvelle solution qui concerne directement le protocole de routage en vue de minimiser voire éviter tout retard lié à la présence d'obstacles dans un réseau routier.

Cette thèse se compose de six chapitres définis comme suit.

- Le chapitre 1 : Introduction

Ce chapitre présente un aperçu général sur les STI et souligne les enjeux de recherche relatifs à la dissémination efficace de l'information pour les applications critiques basées sur les réseaux VANET.

- Chapitre 2 : Contexte et problématique

Ce chapitre présente les réseaux VANET en soulignant leur intérêt pour la recherche liée aux communications sans fil. En particulier, pour les applications critiques, garantir la qualité de service pour le délai de transmission dans les réseaux VANET est complexe. Cette complexité est due à un certain nombre de facteurs, dont la forte mobilité des nœuds, l'affaiblissement du signal par trajets multiples, les interférences, etc. C'est dans ce contexte précis que se situent nos recherches. Pour ce faire, dans ce chapitre, nous illustrons également le problème de l'affaiblissement par trajets multiples afin de démontrer sa complexité pour les transmissions d'applications critiques des STI basées sur les réseaux VANET. Nous considérons dans ce chapitre des scénarios qui permettent de poser le problème que nous étudions relativement aux véhicules d'urgence.

- Chapitre 3 : Revue de la Littérature sur les Protocoles de Routage pour les réseaux VANET

Le chapitre 3 fait office d'état de l'art et situe notre problématique par rapport aux travaux existants. De ce fait, il examine certaines techniques de routage prometteuses pour la dissémination d'informations et qui sont basées sur les métaheuristiques, la géographie et le trafic. En outre, le chapitre 3 traite également des protocoles d'accès au support de communication dans les réseaux VANET. Ce chapitre se termine par une synthèse générale qui met en relief les forces et faiblesses de certains protocoles de routage résultant de l'analyse de l'état de l'art et définit les objectifs de recherche.

- Chapitre 4 : Vers un nouveau routage basé sur les capacités de véhicules autonomes pour les réseaux VANET

Ce chapitre constitue le socle de notre approche méthodologique. Nous présentons donc une brève description du principe de résolution des objectifs de recherche, puis nous procédons à une étude de cas détaillée pour démontrer comment prendre en compte l'impact des composantes urbaines au niveau des communications véhiculaires. L'étude de cas est basée sur un apprentissage supervisé utilisant des réseaux de neurones artificiels.

L'idée consiste à utiliser l'apprentissage pour prédire spécifiquement le taux d'erreurs binaires (TEB) encouru par rapport aux composantes urbaines (bâtiments) perçues par le nœud mobile (véhicule intelligent). Pour ce faire, le véhicule intelligent utilise une carte numérique embarquée et un algorithme pour déterminer la position d'obstacles dans son environnement. Les données de simulation pour l'apprentissage ont été collectées par simulation en utilisant SUMO et NS-3. Les informations de la carte routière et la structure géométrique des bâtiments sont obtenues à partir de la base de données géographique OpenStreetMap. En outre, ce chapitre présente le protocole CRV (Routage Cognitif pour les réseaux VANET ou Cognitive Routing for VANET), la solution développée pour la dissémination de l'information dans un contexte d'applications critiques basées sur les réseaux VANET. Le chapitre 4 aborde également l'aspect architectural d'un véhicule autonome en précisant le rôle ainsi que la structure dynamique des interactions entre les différents modules de cette architecture.

- Chapitre 5 : Routage Cognitif pour les réseaux VANET : Analyse de la performance et apports

Ce chapitre est consacré à l'analyse des performances du protocole CRV dans le contexte des applications critiques des STI basées sur les réseaux VANET. Trois scénarios impliquant des structures géométriques ont été développés pour examiner leurs effets sur les transmissions impliquant des applications critiques. Pour chaque scénario, différentes densités de véhicules ont été prises en compte et les performances de CRV ont été évaluées et comparées à celles de deux protocoles identifiés dans le chapitre 3 soit CL-AntHocNet et GPSR. Les résultats de nos simulations montrent une bonne performance du protocole CRV par rapport aux deux autres protocoles, considérant comme critère le PDR (Taux de livraison des paquets ou Packet Delivery Rate) pour la plupart des cas.

- Chapitre 6 : Conclusion et Perspectives de recherche

Ce chapitre présente une synthèse générale en mettant l'accent sur les contributions importantes de ce travail de recherche. Les principales contributions de cette thèse se résument comme suit :

- a) Contribution par rapport au protocole de routage pour la dissémination de l'information pour les applications critiques dans les réseaux VANET.
- b) Contribution par rapport aux véhicules intelligents, notamment avec la mise en place d'une nouvelle architecture de communications pour les réseaux ad-hoc.

Parmi les perspectives de travaux futurs, nous considérons :

- a) Un enrichissement des traitements de perception et de prédictions pour considérer des objets complexes des réseaux routiers présentant des obstacles pour les communications véhiculaires.
- b) Une API permettant de traiter les problèmes de communications entre un véhicule intelligent (embarqué dans les véhicules) et l'architecture de communications véhiculaires (des normes telles que le DSRC). Des problèmes d'interopérabilité doivent être étudiés pour assurer l'efficacité de telles interactions.

Le chapitre se termine par une liste de publications soulignant les contributions issues de ce travail.

Chapitre 2

Contexte et Problématique

Sommaire du chapitre 2

2.1	Introduction.....	31
2.2	Contexte	32
2.3	Problématique	34
2.4	Conclusion	36

2.1 Introduction

Au cours de ces dernières années, les communications inter-véhicules (CIVs) et les réseaux ad-hoc de véhicules (VANET) ont suscité un intérêt considérable dans le domaine de la recherche liée à la communication sans fil. Cette motivation est issue entre autres des progrès réalisés dans le domaine des STI, notamment, avec l'avènement du DSRC (Dedicated Short Range Communication) ([Kenney, 2011](#)), des normes IEEE 802.11p et WAVE (Wireless Access In Vehicular Environments) ([Parker & Valaee, 2006](#)). En effet, les STI tirent profit des nouvelles technologies de l'information et de la communication en les appliquant au domaine des transports et de sa logistique. Leur développement repose sur des capacités telles que : la communication, le traitement de l'information, le comportement adaptatif, etc.

Les réseaux VANET constituent un cas particulier des réseaux MANET (Mobile Ad-hoc Networks) et partagent certaines caractéristiques communes telles que : la mobilité et l'auto-organisation des nœuds. Cependant, compte tenu du comportement du

conducteur, les réseaux VANET se diffèrent des réseaux MANET génériques par un changement de topologies rapides([Blum, Eskandarian & Hoffman, 2004](#)). En outre, les réseaux VANET sont également caractérisés par la forte mobilité des nœuds, la scalabilité et une densité de réseau variable. Ils sont considérés comme l'une des technologies prometteuses permettant d'améliorer l'efficacité et la sécurité des systèmes de transports intelligents. Par exemple, les véhicules peuvent communiquer des informations de détournement, de congestion, et d'accident de la route aux autres véhicules plus proches afin de réduire les embouteillages à proximité des zones touchées. Plus particulièrement, pour une application critique des STI, qui consiste à accorder la priorité de route à un véhicule d'urgence, les informations communiquées par les véhicules peuvent être utilisées pour la préemption des feux de la circulation afin de faire passer l'état de ces derniers au vert en direction du véhicule d'urgence tout en bloquant les autres directions de passage. Le VANET constitue donc un pilier important pour la transmission des données des applications critiques dans les STI. L'absence du VANET dans les STI impliquerait l'inexistence d'une communication inter-véhicules, ce qui pourrait rendre le problème de transmission complexe pour les applications critiques des STI. En effet, lorsqu'on place les VANET dans les STI, ils prennent de l'importance, car ils permettent la dissémination de l'information en temps réel si les ressources et traitements nécessaires y sont intégrés.

Dans la suite de ce chapitre, nous présentons le contexte de notre recherche, puis nous abordons quelques situations problématiques dans les STI, qui impactent sur la transmission des données, et enfin nous clôturons ce chapitre par une conclusion.

2.2 Contexte

Lors de nos travaux de recherche publiés dans([Benyahia, Bappa & Perrot, 2017](#)), nous illustrons par des exemples, certaines applications critiques des STI qui requièrent une qualité de service (QoS) pour la transmission des données. Dans l'un des exemples, nous démontrons l'importance du délai de transmission pour un véhicule d'urgence. Dans cet exemple, il est question d'avertir un véhicule d'urgence par rapport à une congestion afin qu'il puisse l'éviter en empruntant une intersection de détournement. Il est donc crucial que l'information parvienne au véhicule d'urgence avant qu'il ne dépasse l'intersection de

détour. Si la communication est suffisamment rapide, le véhicule d'urgence recevra l'information en temps opportun et il pourra quitter la voie congestionnée. Dans le cas contraire, il sera bloqué par la congestion, ce qui entraînera un retard sur son trajet. Dans un tel scénario, des multiples problématiques peuvent être à l'origine de la perturbation de la transmission. Certaines problématiques (par exemple les interférences) sont issues des réseaux sans fils, d'autres sont liées aux propriétés intrinsèques des réseaux VANET (vitesse élevée des nœuds, fréquents changements de la topologie, forte mobilité, etc.). Par exemple, l'une des propriétés du canal sans fil est son comportement aléatoire qui varie dans le temps à cause de la mobilité des véhicules et des obstacles présents dans l'environnement. Plus précisément, le canal destiné à un véhicule donné peut être sujet à des conditions médiocres à certains moments et à des conditions favorables à d'autres moments. Cette variation des conditions du canal est appelée "*comportement d'évanouissement du canal* ([Rappaport, 1996](#))". Dans d'autres situations, les copies du signal transmis par les véhicules peuvent être reçues avec des délais et des puissances différentes. Ce phénomène est appelé "*évanouissement par trajets multiples*" et peut induire à une dégradation des performances lorsque les signaux transmis ont une courte durée (cas d'une transmission large bande)([Patzold, 2001](#)). En outre, comme mentionnées ci-haut, les interférences peuvent également dégrader le signal de transmission dans un réseau VANET. La principale raison de cette dégradation est due aux interférences causées par les transmissions d'autres véhicules dans une zone congestionnée([Schmidt, Köllmer, Leinmüller, Bøddeker & Schäfer, 2009](#)). Outre cela, les réseaux VANET opèrent dans différents environnements (urbains, autoroutes, etc.), ce qui rend les communications difficiles. Notamment, lorsque les réseaux VANET opèrent dans un environnement urbain, ceux-ci sont confrontés à divers obstacles (immeubles, arbres, etc.) qui affectent le signal de transmission et le dégrade. Dans un contexte d'applications critiques, l'utilisation des réseaux VANET rend les communications difficiles et problématiques, car les exigences de QoS sont élevées au niveau du temps ainsi qu'au niveau de la précision des informations. En dépit des difficultés rencontrées par les applications critiques des STI dans un réseau VANET, ce dernier leur confère l'avantage d'exploiter les informations du trafic en temps réel pour identifier des chemins de navigation efficace.

2.3 Problématique

Tel que souligné précédemment l'utilisation des réseaux VANET pour les applications critiques des STI est problématique à cause de différents facteurs. Dans cette section, nous illustrons les problématiques en lien avec notre étude par deux cas d'applications critiques des STI. Nous considérons, 1) l'impact du délai et 2) l'évanouissement par trajets multiples sur la transmission. Ces problématiques sont illustrées par les figures 2.1 et 2.2 ci-dessous. À l'heure actuelle, il est important de souligner qu'il n'existe pas assez d'études permettant de démontrer l'impact de la dégradation du signal par des métriques mesurable dans les réseaux VANET.

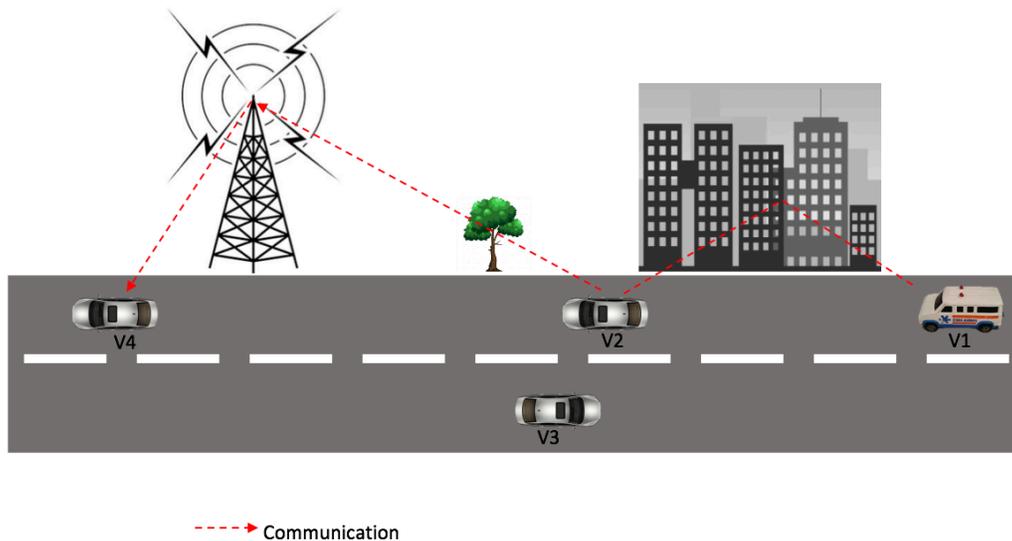


Figure 2.1: Évanouissement par trajets multiples

La figure 2.1 illustre la dégradation du signal causée par plusieurs atténuations. On constate que le signal de transmission subit plusieurs atténuations avant d'atteindre sa destination. Ces atténuations dégradent le signal, ce qui implique que l'information ne sera pas reçue avec la même puissance émettrice. Par conséquent, le risque d'échec de transmission est présent.

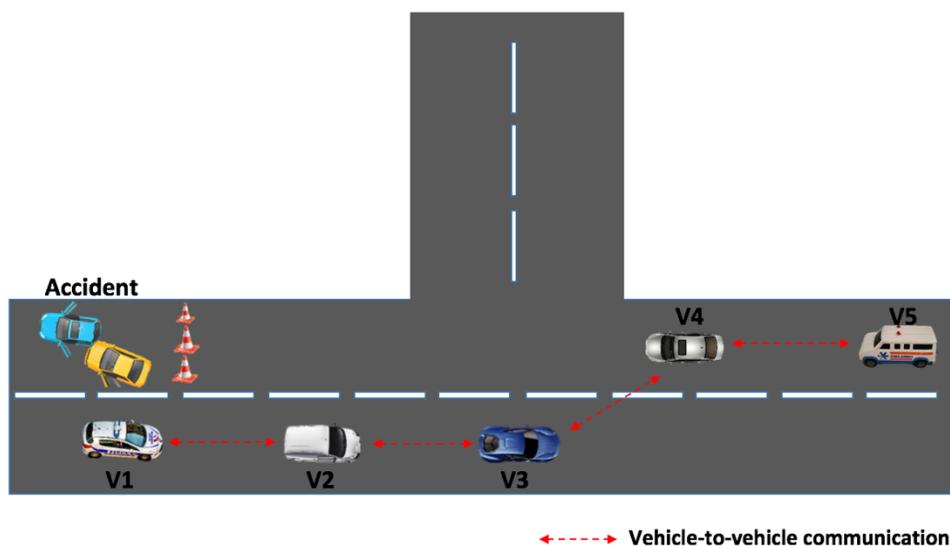


Figure 2.2 : Avertissement d'un véhicule d'urgence pour éviter une congestion
([Benyahia et al., 2017](#))

La figure 2.2 illustre l'importance du délai de transmission pour une application critique. En effet, un message d'avertissement est transmis vers le véhicule pour l'indiquer de quitter une voie congestionnée à la suite d'un accident. Dans ce scénario, si la transmission est suffisamment rapide, le message va atteindre le véhicule d'urgence avant qu'il n'atteigne l'intersection de détour. Dans le cas contraire, le véhicule va dépasser l'intersection de détour. Par conséquent, il va se retrouver bloqué dans la congestion, ce qui risque d'entraîner des retards sur son trajet. Ceci implique d'assurer une QdS élevée en termes de délai de transmission de bout-en-bout.

Sur la base de notre analyse précédente, nous constatons que l'utilisation des réseaux VANET pour les applications critiques des STI est problématique, à cause des contraintes exigeantes de la QdS. De ce fait, pour la suite de notre étude, nous répartissons les problématiques inhérentes aux applications critiques des STI suivant deux catégories, à savoir : 1) la problématique d'applications critiques temps réel et 2) celle d'applications critiques sujettes à la dégradation du signal. En outre, tout au long de ce document, l'utilisation du terme obstacle fait référence à un « *obstacle statique* » (par exemple un immeuble).

2.4 Conclusion

Dans ce chapitre, nous avons présenté quelques scénarios illustrant des applications critiques dans les STI pour lesquelles l'utilisation du réseau VANET est problématique à cause des contraintes exigeantes de QoS. À la suite de notre analyse, nous avons classé ces problèmes en fonction de deux familles comme suit : la problématique d'applications critiques temps réel et celle d'applications critiques sensibles à la dégradation du signal. Nous montrons également que l'importance d'utiliser les réseaux VANET pour ce type d'applications est liée à l'avantage de pouvoir disséminer l'information en temps réel.

La dissémination de l'information dans un réseau VANET se fait par le biais des protocoles de routage qui jouent un rôle important au niveau de la qualité des transmissions. Par conséquent, il est essentiel de considérer dans la littérature les approches de routage existantes et en faire des analyses relativement aux problèmes posés. L'étude des protocoles de routage fait l'objet du prochain chapitre qui fait office d'état de l'art.

Chapitre 3

Revue de la Littérature sur les Protocoles de Routage pour les Réseaux VANET

Sommaire du chapitre 3

3.1	Le routage au niveau des réseaux VANET.....	37
3.1.1	Protocoles de routage basés sur les métaheuristiques.....	38
3.1.1.1	Algorithmes évolutionnaires ou génétiques.....	38
3.1.1.2	Protocoles de routage basés sur l'intelligence en essaims (Swarms Intelligence)	41
A.	Routage basé sur l'optimisation par colonies des fourmis.....	41
B.	Routage basé sur l'optimisation par essaims particuliers	43
C.	Routage basé sur l'optimisation par colonies d'abeilles.....	46
3.1.2	Protocoles de routage basés sur la géographie.....	47
3.1.3	Protocoles de routage basés sur le trafic.....	53
3.2	Protocoles de contrôle d'accès dans les réseaux VANET	56
3.3	Synthèse Générale.....	61

3.1 Le routage au niveau des réseaux VANET

Dans ce chapitre, nous passons en revue les protocoles de routage les plus adaptés à la transmission des données dans les réseaux VANET. Ces protocoles sont basés sur :

- des métaheuristiques pour VANET ;
- des informations géographiques ;
- des informations liées au trafic du réseau routier.

3.1.1 Protocoles de routage basés sur les métaheuristiques

Selon l'analyse faite dans le chapitre 2, le routage des réseaux VANET présente de grands défis à cause de l'environnement de ces réseaux et des besoins des applications émergentes des STI nécessitant des communications véhiculaires.

Sachant que le problème de routage se formalise comme problème d'optimisation, nous considérons dans cette section les travaux de recherche existants relativement aux métaheuristiques.

Les métaheuristiques constituent un ensemble de méthodes de résolution de problèmes d'optimisation complexe ([Boussaïd, Lepagnot & Siarry, 2013](#)). Les algorithmes génétiques, colonies de fourmis, colonies d'abeilles, essais de particules figurent parmi les techniques de résolution existantes qui sont utilisées dans les réseaux VANET pour le routage.

3.1.1.1 Algorithmes évolutionnaires ou génétiques

Les algorithmes génétiques ([Holland, 1973](#)) sont inspirés de la théorie de l'évolution selon Darwin, ils sont basés sur une population qui évolue à travers plusieurs générations jusqu'à atteindre un état stable et de convergence.

([Gupta & Kumar, 2014](#)) proposent un protocole de routage basé sur les algorithmes génétiques pour minimiser le retard de transmission entre la source et la destination en utilisant l'arbre couvrant (spanning tree) et l'arbre de routage (routing tree). L'approche proposée comprend trois phases :

- algorithme génétique : permet de déterminer le chemin optimal (plus court) en associant à chaque nœud une valeur d'aptitude.
- arbre couvrant (spanning tree) : permet d'éliminer les liens qui provoquent les boucles lors de la transmission de données.
- arbre de routage (routing tree) : permet de faciliter l'insertion et la suppression des nœuds.

Toutefois, l'opération consistant à éliminer les boucles de routage peut introduire une latence supplémentaire au niveau de la transmission.

Dans ([Saleet, Langar, Basir & Boutaba, 2009](#)), les auteurs proposent le protocole de routage AMR (Adaptative Message Routing) pour minimiser le retard de transmission lors de la mise à jour des informations de localisation des destinations au niveau de la station de base appelée RSU (Road Side Unit). Ce protocole détermine les routes qui minimisent le délai de bout-en-bout en satisfaisant les conditions inhérentes à la probabilité de connectivité ainsi qu'au nombre de sauts du chemin sélectionné. La problématique consistant à minimiser le délai a été formulée sous forme de problème d'optimisation sous contraintes, puis résolue par un algorithme génétique. Celui-ci est caractérisé par une fonction qui minimise le délai de retard sur le segment de route sélectionnée entre la source et la destination en tenant compte du nombre de sauts ainsi que de la probabilité de connectivité.

([Cavalcante, Aquino, Pappa & Loureiro, 2012](#)), ont mené une étude sur la problématique du déploiement des RSUs pour une dissémination efficace de l'information dans les réseaux VANET. L'approche adoptée dans le cadre de cette étude est basée sur deux phases :

- modéliser le problème du déploiement des RSUs en utilisant un formalisme basé sur les mathématiques.
- résoudre le modèle mathématique grâce à l'algorithme génétique.

Pour modéliser la problématique mathématiquement, celle-ci a été formalisée sous la forme d'un problème de couverture maximale avec seuil de temps (Maximum Coveragewith Time Threshold Problem, MCTTP).

L'originalité de l'approche proposée par ([Cavalcante et al., 2012](#)), réside dans le fait d'utiliser un algorithme génétique au détriment d'un algorithme glouton pour résoudre le problème de couverture maximale avec seuil de temps. En effet, d'après les résultats issus de leur analyse des performances, utiliser une approche génétique pour résoudre le

problème de couverture maximale avec seuil de temps permet d'obtenir des meilleurs résultats par rapport à l'approche gloutonne.

Identifier les meilleurs emplacements pour un point d'accès (PA) pour la diffusion d'informations dans les réseaux ad-hoc de véhicules représente un défi majeur. Afin de résoudre ce problème et minimiser la bande passante globale utilisée, ([Lochert, Scheuermann, Wewetzer, Luebke & Mauve, 2008](#)) ont développé une approche optimisée basée sur l'algorithme génétique. Cette approche est composée de deux phases : (1) développer un schéma d'agrégation basé sur l'agrégation hiérarchique et le routage Landmark, (2) utiliser un algorithme génétique pour évaluer le nombre et l'emplacement des unités de soutien (supporting units, par exemple les RSUs) en vue de faire face à la nature clairsemée d'un réseau VANET.

Lorsque le trafic routier est moins dense (moins de véhicules) dans les réseaux VANET, il est complexe d'identifier et sélectionner efficacement le véhicule capable de transporter les paquets de données vers la destination. C'est dans cette optique que ([Bitaghsir & Hendessi, 2011](#)) proposent un protocole de routage basé sur l'algorithme génétique en vue de faire face à ce problème.

Dans ce protocole, un mécanisme de routage géographique est utilisé pour les réseaux VANET denses ou connectés. Cependant, pour un transfert tolérant au délai, le protocole de routage prend en considération d'autres paramètres (direction du nœud, distance, etc.) utiles afin de choisir le meilleur nœud DTN (Delay Tolerant Network) capable de stocker et d'acheminer les paquets de données vers la destination. L'algorithme génétique est utilisé pour quantifier l'impact de chaque paramètre au niveau du choix du nœud DTN. Les résultats issus de l'évaluation des performances montrent que les paramètres générés par la fonction objective de l'algorithme génétique réduisent les délais moyens de livraison et augmentent le taux de transmission moyen.

La particularité du protocole de routage proposé réside dans le fait d'utiliser un algorithme génétique lorsque le réseau est moins dense pour sélectionner le meilleur nœud

capable de transporter les données vers la destination. Toutefois, l'opération qui consiste à générer des nouveaux paramètres par le biais de la fonction objective pour le choix du meilleur nœud DTN peut introduire une latence au niveau du délai de transmission. Par conséquent, ce protocole ne peut garantir une QoS en termes de délai de bout-en-bout.

3.1.1.2 Protocoles de routage basés sur l'intelligence en essaims (Swarms Intelligence)

Cette catégorie d'algorithmes de routage s'inspire de l'intelligence des essaims tels que : les fourmis, les abeilles, les oiseaux, etc. Elle englobe également une variante d'algorithmes de routage qui combinent l'intelligence d'essaims et optimisation inter-couches (cross-layer) ([Srivastava & Motani, 2005](#)).

A. Routage basé sur l'optimisation par colonies des fourmis

Les algorithmes de colonies de fourmis s'inspirent du comportement collectif des fourmis pour résoudre des problèmes d'optimisations NP-difficiles ([Majumdar, Prasad, Kumar & Kumar, 2016](#)).

([Benyahia et al., 2017](#)) proposent le protocole de routage CL-AntHocNet (Cross-Layer AnthocNet) basé sur l'optimisation par colonies de fourmis ainsi que l'architecture inter-couches en vue de réduire le retard de transmissions et améliorer la fiabilité du réseau VANet. L'approche consiste à considérer les changements qui ont lieu au niveau du canal de transmission et qui ont un impact sur le routage. Ainsi, lors du fonctionnement du protocole de routage (couche 3), lorsque les liens entre certains nœuds du réseau deviennent non-fonctionnels à la suite des changements de la couche physique (couche 1), le protocole de routage est déclenché pour trouver des nouvelles routes. Pour ce faire, le protocole de routage CL-AntHocNet exploite le concept d'architecture inter-couches pour réagir immédiatement aux changements de la couche physique, ce qui permet d'éviter les délais d'attente tout en réduisant la perte des paquets de données.

Les capacités réactives du protocole de routage CL-AntHocNet sont caractérisées par une nouvelle structure définie par des composantes autonomes et distribuées ayant des capacités de réactions.

Afin de tester l'efficacité de la nouvelle structure du protocole de routage CL-AntHocNet, les auteurs ont simulé des changements au niveau de la couche physique. Un événement est alors déclenché au niveau de la couche 3 dont les conséquences consistent à lancer une recherche dans le but de trouver les nouveaux meilleurs chemins entre les nœuds concernés par les changements de la couche physique. Les métriques utilisées pour représenter l'efficacité du fonctionnement de CL-AntHocNet dans le contexte d'une architecture cross layer sont présentées par le taux d'erreur des paquets et les délais de transmission.

Le protocole CL-AntHocNet est un candidat potentiel pour la poursuite de nos travaux de recherche. En effet, ce dernier démontre déjà de bonnes performances en termes de délai de transmission pour les véhicules d'urgences selon les tests présentés dans ([Benyahia et al., 2017](#)).

Dans ([Rana, Thulasiraman & Thulasiram, 2013](#)), les auteurs proposent un protocole de routage hybride pour exploiter efficacement la bande passante dans un réseau VANET. La nature hybride du protocole s'explique par le fait qu'il utilise deux approches de routage (proactive et réactive) différentes selon les zones. En conséquence, le protocole se base sur l'optimisation par colonies de fourmis pour la découverte des routes en divisant le réseau de véhicules en plusieurs zones. Ainsi, une stratégie de routage proactive est adoptée au sein d'une zone et une stratégie de routage réactive entre les zones pour déterminer les routes. Il exploite également les informations locales stockées au niveau de chaque zone, ce qui permet de réduire les problèmes de diffusion des messages et de congestion. En outre, lors de son processus de routage, ce protocole prend en considération les informations telles que : le modèle de mobilité des véhicules, la densité, la vitesse et les conditions d'évanouissements du canal.

Les auteurs de l'article ont évalué les performances du protocole proposé en termes de délai de bout-en-bout et taux de livraison de paquets par rapport aux protocoles AODV (Ad hoc On Demand Distance Vector), AOMDV (Ad-hoc On-demand Multipath Distance

Vector) et GPSR (Greedy Perimeter Stateless Routing). Les résultats issus de leurs expérimentations montrent une meilleure performance de ce protocole en comparaison aux protocoles AODV, AOMDV et GPSR. En dépit des bonnes performances du protocole, ce dernier met à jour la table de routage d'une zone de manière proactive toutes les vingt secondes. Même si les auteurs n'abordent pas la question dans leurs travaux, le mécanisme de maintien de table de routage basé sur une approche proactive peut induire à un nombre élevé de message de contrôle (overhead) et par la même occasion détérioré la performance du réseau.

Dans ([Balaji, Sureshkumar & Saravanan, 2013](#)), un protocole de routage basé sur l'optimisation de colonies de fourmis (ACO, Ant Colony Optimization) a été proposé pour améliorer les performances du protocole réactif DYMO (Dynamic Manet on demand routing protocol) ([Sommer & Dressler, 2007](#)) en termes de scalabilité et d'organisation du réseau. L'approche proposée utilise un graphique cinétique pour fournir des informations de base (vitesse et position) qui permettent aux véhicules de prendre des décisions de routage appropriées ([Härri, Bonnet & Filali, 2008](#)). Toutefois, compte tenu de son approche de routage réactive, ce protocole peut être confronté aux problèmes de latence ainsi qu'à l'inondation par paquets causés par le processus de découverte de route.

B. Routage basé sur l'optimisation par essaims particuliers

Cette catégorie d'algorithmes de routage VANET s'inspire du comportement social des animaux évoluant en essaim, tels que les oiseaux migrateurs ou les poissons([Del Valle, Venayagamoorthy, Mohagheghi, Hernandez & Harley, 2008](#); [Wang, Huang, Zhou & Pang, 2003](#)). Ces algorithmes présentent de nombreux avantages par rapport aux autres catégories d'algorithmes de routage. En particulier, ils permettent de réduire le nombre de messages de contrôle en partageant des informations (position, vitesse, direction du véhicule, etc.) locales lors des décisions de routage([Kalambe, Deshmukh & Dorle, 2015](#)).

Le protocole OLSR (Optimized Link State Routing Protocol) ([Jacquet et al., 2001](#)) a été initialement conçu pour les réseaux ad-hoc mobiles. Par conséquent, ce protocole

n'est pas directement applicable aux réseaux VANET. C'est dans cette optique que, ([Bandi, 2015](#)) propose une version modifiée d'OLSR pour VANET, basée sur les métaheuristiques.

L'approche proposée comprend deux phases : optimisation et simulation. Dans la première phase, quatre techniques (l'algorithme génétique, l'optimisation par essais particuliers, le recuit simulé, l'évolution différentielle) ont été conçues pour trouver la solution optimale.

À l'issue du processus d'optimisation, la nouvelle solution générée par le protocole OLSR modifié sera évaluée par le biais de la fonction d'aptitude (fitness function) caractérisée par trois paramètres : taux de livraison de paquets, charge de routage normalisée et délai moyen de bout-en-bout.

Dépendamment, du résultat issu de la fonction d'aptitude, un traitement itératif sera exécuté jusqu'à ce que la condition d'arrêt soit satisfaite.

Les résultats issus de la simulation montrent une bonne performance du protocole de routage OLSR modifié en termes de taux de livraison de paquets par rapport au protocole d'origine. Néanmoins, l'opération effectuée par la fonction d'aptitude peut introduire une latence (lié au calcul effectué) au niveau de la transmission.

D'après([Musliu, 2010](#)), le protocole de routage DSDV (Destination-Sequenced Distance Vector routing) comprend plus de 229 combinaisons possibles pour le paramétrage. Par conséquent, déterminer le paramétrage adéquat pour ce dernier en vue d'initialiser le routage s'avère complexe. C'est dans cette optique que ([Sharma, Giri & Singhal, 2014](#)) ont proposé une approche de paramétrage basée sur l'algorithme PSO (Particle Swarm Optimization) pour déterminer la configuration optimale des paramètres du protocole DSDV en fonction de trois métriques (taux de livraison de paquets, délai moyen de bout-en-bout, charge de routage normalisée).

L'approche proposée se compose de deux phases : (1) utiliser l'algorithme PSO pour déterminer la meilleure configuration des paramètres en s'appuyant sur la fonction

d'aptitude (fitness function), (2) injecter chaque solution générée à la première phase dans DSDV.

Dans ([Lobiyal, Katti & Giri, 2015](#)), les auteurs présentent un algorithme basé sur PSO en vue de déterminer une combinaison optimale des paramètres utilisée par AOMDV ([Yuan, Chen & Jia, 2005](#)) pour le cas de scénario réel dans VANET. Compte tenu du nombre élevé de combinaisons possibles pour les neuf paramètres (Type, Reversed, Last hop, Hop count, RREQ ID, Destination IP Address, Destination Sequence Number, Originator IP Address, Originator Sequence Number) ([Yuan et al., 2005](#)) du protocole AOMDV, les auteurs transforment la problématique consistant à déterminer une combinaison optimale des paramètres en un problème d'optimisation combinatoire, résolu en utilisant PSO.

L'algorithme proposé utilise trois critères QoS (taux de perte de paquets, délai moyen de bout-en-bout, charge de routage normalisée) et comprend deux phases : (1) appliquer PSO pour sélectionner la meilleure valeur des paramètres, (2) calculer la fonction d'aptitude (fitness function) définie par rapport aux trois critères QoS sélectionnés précédemment.

Pour développer un protocole de routage performant, ([Kalambe et al., 2015](#)) ont proposé de combiner le protocole réactif SIFT (Simple Forwarding over Trajectory) au protocole DREAM (Distance Effect Routing Algorithm for Mobility) basé sur la localisation. Les auteurs utilisent également l'algorithme PSO pour résoudre le problème du contrôle centralisé et améliorer l'efficacité du protocole hybride dans un réseau à grande échelle tel que le VANET.

Afin d'améliorer la qualité du prochain saut véhiculaire sélectionné, ([Kaiwartya & Kumar, 2014](#)) ont utilisé l'algorithme PSO pour optimiser les critères liés à la sélection du prochain saut véhiculaire. Pour réaliser cet objectif, les auteurs élaborent l'équation qui calcule la fonction d'aptitude de chaque solution candidate en se basant sur six métriques (probabilité de déconnexion de lien, livraison de paquets, charge réseau, délai de bout-en-

bout, débit, nombre de sauts). Toutefois, cette équation peut introduire un coup en termes de délai de calcul, lorsque plusieurs nœuds candidats sont identifiés comme prochain saut véhiculaire.

C. Routage basé sur l'optimisation par colonies d'abeilles

Cette catégorie d'algorithmes de routage VANET s'inspire du comportement des abeilles mellifères observé lors de la recherche d'une source de nourriture ([Karaboga & Basturk, 2007, 2008](#)).

Dans ([Bitam, Mellouk & Zeadally, 2013](#)), les auteurs ont proposé le protocole de routage HyBR pour VANET en vue d'améliorer l'efficacité de la transmission des paquets. C'est un protocole de routage unicast et multi-trajets qui garantit que les exigences strictes (délai de bout-en-bout, taux de livraison de paquets, ...) des applications de sécurité VANET soient satisfaites. HyBR adopte un comportement de routage hybride. La nature hybride du protocole est liée à l'application d'une stratégie de routage basée sur la topologie en cas de faible densité dans le réseau ou à l'application d'une stratégie de routage géographique dans le cas contraire.

La densité du réseau est calculée à l'aide du coefficient de densité α (rapport entre le périmètre de la portée de transmission TR et le paramètre observateur β spécifiant la densité de la portée de transmission). Ensuite, le réseau est subdivisé en sous-réseaux. Si le nombre de nœuds dans un sous-réseau est supérieur au coefficient de densité α , le routage basé sur la topologie est appliqué. Sinon, le routage basé sur la géographie est appliqué.

Toutefois, compte tenu de son approche de routage basée sur la topologie, ce protocole est sujet à des risques de congestion provoqués par les changements fréquents de la configuration du réseau.

Synthèse routage basé sur les métaheuristiques

Dans cette section, nous avons passé en revue les protocoles de routage VANET qui tirent profit des approches métaheuristiques. Nous regroupons ces protocoles suivants deux familles, à savoir : les protocoles de routages qui utilisent les algorithmes génétiques

et ceux basés sur l'intelligence en essaims (colonies de fourmis, essaims particuliers et colonies d'abeilles). Ainsi, nous avons pu constater que certains travaux ([Gupta & Kumar, 2014](#); [Saleet et al., 2009](#)) basés sur les algorithmes génétiques cherchent à minimiser le retard de transmissions. Cependant, ces travaux introduisent une latence en termes de délai de calcul (utilisation de la fonction d'aptitude), ce qui peut avoir un impact sur la transmission. Par contre, les travaux ([Cavalcante et al., 2012](#); [Lochert et al., 2008](#)) basés sur les algorithmes génétiques s'intéressent respectivement à la dissémination efficace de l'information ainsi qu'à la minimisation de la bande passante globale utilisée en s'appuyant sur les RSUs. Néanmoins, ces travaux montrent leurs limites lorsqu'on est dans un environnement sans RSUs.

Les travaux ([Boussaid et al., 2013](#); [Schmidt et al., 2009](#)) basés sur l'intelligence en essaims cherchent à améliorer la performance du réseau VANET par le biais des paramètres tels que : la scalabilité, l'exploitation efficace de la bande passante, l'organisation du réseau, etc. Toutefois, ces travaux présentent un risque de congestion, car ils génèrent des messages de contrôle élevés. D'autres travaux présentés dans ([Patzold, 2001](#)) combinent optimisation par colonies de fourmis et architecture Cross-Layer pour juguler le retard de transmissions et améliorer la fiabilité du réseau VANET. Ces travaux montrent de bonnes performances en termes de délai de transmission et du taux d'erreurs de paquets pour des situations critiques impliquant des véhicules d'urgences. Par conséquent, ils peuvent constituer une bonne piste pour la suite de nos travaux de recherche.

De façon générale, les approches métaheuristiques sont efficaces et permettent d'améliorer la performance des protocoles de routage dans les réseaux VANET.

3.1.2 Protocoles de routage basés sur la géographie

Dans cette catégorie d'algorithmes de routage, la décision de transmettre est principalement basée sur la position de la destination d'un paquet et la position à un saut du nœud ([Kumar & Verma, 2015](#); [Lee, Lee & Gerla, 2010](#); [Li & Wang, 2007](#)).

GPSR (Greedy Perimeter Stateless Routing) ([Karp & Kung, 2000](#)) est le protocole de routage géographique le plus populaire destiné aux réseaux VANET. Ce protocole sélectionne le nœud le plus proche de la destination en utilisant une balise (beacon). L'approche consiste à utiliser l'algorithme glouton (greedy forwarding) pour la sélection du prochain nœud vers lequel les paquets de données seront transmis. En cas d'échec de l'algorithme glouton, le mode de transfert périmétrique est utilisé pour acheminer les paquets de données vers la destination. Plus spécifiquement, GPSR utilise la règle de la main droite en acheminant le paquet de données dans le sens inverse des aiguilles d'une montre en partant du nœud courant par rapport au segment séparant ce dernier de la destination tout en évitant les routes déjà parcourues (crossing links).

Le protocole de routage GPSR démontre une meilleure performance dans un espace ouvert ayant des nœuds distribués uniformément. Toutefois, ce protocole éprouve de nombreuses difficultés. En effet, dans le cas des scénarios urbain, la transmission gloutonne se montre inefficace. Cette inefficacité est due aux obstacles (arbres, bâtiments, véhicules, etc.) qui impactent sur les communications directes entre les nœuds. Outre cela, le problème de boucles de routage subsiste en raison de la forte mobilité des nœuds.

GPCR (Greedy Perimeter Coordinator Routing) ([Lochert, Mauve, Füßler & Hartenstein, 2005](#)) est un protocole de routage unicast destiné aux réseaux VANET. Il exploite la disposition des routes et des intersections pour une prise de décision de routage. Lors du processus de routage, GPCR utilise deux mécanismes : le greedy forwarding restreint et une politique de réparation qui prend en considération l'architecture réelle du réseau routier. Toutefois, ce protocole comporte trois défauts :

- L'algorithme qui détermine si un nœud se trouve dans une intersection ou non augmente le coût de surcharge du réseau ;
- Le protocole est trop dépendant du coordinateur (nœud situé dans l'intersection) ;
- Plusieurs paquets sont transmis par le coordinateur, ce qui peut résulter en une collision et une perte de paquets de données.

Afin de réduire la perte de paquets de données, on peut mettre en place un protocole d'accès au canal de transmission adapté à GPCR. Ainsi, lorsque le coordinateur transmet, les autres nœuds se mettent en mode écoute du support.

Une autre solution consisterait à mettre en place un mécanisme de transmission sélectif au niveau du coordinateur, c'est-à-dire que le coordinateur transmet uniquement les paquets de données prioritaires (par exemple les messages d'urgences).

GpsrJ+ ([Lee, Härrri, Lee & Gerla, 2007](#)) est un protocole qui a été développé pour combler les lacunes du protocole GPCR. Ce protocole utilise une carte électronique pour déterminer si un nœud se trouve dans une intersection ou non, ce qui permet de réduire les coûts de calculs introduits par GPCR lors du même procédé. En outre, afin de réduire le problème de dépendance au coordinateur ainsi que celui des collisions et perte de paquets dans GPCR, le protocole de routage GpsrJ+ prédit la direction de transmission en acquérant des données sur le prochain saut à effectuer après le coordinateur. Si la direction de propagation est inchangée, le paquet de données traverse directement l'intersection sans être retransmis par le coordinateur.

GpsrJ+ améliore la performance de GPCR en termes de taux de livraison de paquets. Toutefois, le protocole de routage GpsrJ+ considère la route comme une ligne rectiligne, ce qui implique un maximum de deux nœuds voisins dans la même section de route et au plus seulement quatre nœuds voisins à l'intersection. Ce dernier ne reflète pas un environnement routier réaliste.

PDVR (Position-Based Directional Vehicular Routing) ([Tian, Shafiee & Leung, 2009](#)) est un protocole de routage qui permet de déterminer une route stable et efficace pour la transmission des paquets de données vers la destination. Étant donné que la forte mobilité des véhicules affecte la communication dans un réseau VANET, la stratégie de transfert dans PDVR utilise deux approches pour le choix du prochain saut à effectuer. Tout d'abord, le véhicule désigné comme prochain saut doit se déplacer dans la même

direction que le véhicule source. Ensuite, le prochain saut devrait également se trouver dans la direction générale de la destination.

PDVR essaie de maintenir un chemin stable et efficace en se basant sur les informations relatives à la position du véhicule voisin, à la direction du déplacement et à la position du nœud de destination pour sélectionner le véhicule qui servira de prochain saut.

DGR (Directional Greedy Routing) ([Bilal, Bernardos & Guerrero, 2013](#)) est un protocole de routage basé sur la position destiné à la communication véhicule-à-véhicule et fonctionnant dans un environnement ouvert. Ce protocole exploite le GPS et des cartes numériques statique (i.e. que les informations ne sont pas connues en temps réel) pour recueillir des informations sur la position des véhicules. Il suppose l'existence d'un service de localisation capable de déterminer la position du véhicule destinataire. DGR suppose également que chaque véhicule est en mesure de déterminer sa vitesse et sa direction par le biais des capteurs embarqués. L'approche directionnelle de transmission gloutonne (directional greedy forwarding) exploite la direction des véhicules comme une stratégie d'expédition. Ce qui permet, de transmettre le paquet de données vers la direction générale de la destination.

La limite de ce protocole est liée au fait que chaque nœud diffuse sa position dans le réseau. Ceci peut mener à une inondation par paquet et conduire à une congestion du réseau.

PDGR (Predictive Directional Greedy Routing) ([Gong, Xu & Holle, 2007](#)) est une version améliorée du protocole de routage DGR. De ce fait, il repose sur le même principe que DGR. Dans ce protocole, le véhicule ne se contente pas seulement de diffuser sa position, mais il diffuse également celle de ses voisins à un saut. La stratégie prédictive gloutonne est utilisée pour transmettre les paquets vers leur destination. Cette stratégie détermine les futurs voisins possibles du prochain saut, ce qui permet d'améliorer le routage.

L'inconvénient majeur de PDGR est le coût induit par le calcul et la diffusion des voisins à deux sauts. En effet à chaque fois qu'il faut transmettre, le protocole calcul le prochain saut à effectuer en fonction d'informations telles que : la plus petite distance séparant le nœud courant de la destination, la vitesse, la position. Ce qui peut introduire une latence au niveau de la transmission.

GyTAR (Greedy Traffic-Aware Routing) ([Jerbi, Senouci, Rasheed & Ghamri-Doudane, 2009](#)) est un protocole de routage géographique qui exploite les intersections en vue de minimiser : les messages de contrôle, le délai de bout-en-bout et le taux de perte de paquets. Dans GyTAR, on suppose qu'un système de navigation (GPS et carte numérique globale) est disponible. Les cartes et les informations sur le trafic routier sont exploitées pour la transmission des données. On suppose également que la position courante du nœud de destination est connue à l'avance et que le véhicule a la faculté de maintenir une table voisine par l'intermédiaire des paquets Hello. La stratégie de routage dans GyTAR se fait en trois étapes :

- Première étape : estimation de la densité du trafic basé sur la grille ;
- Deuxième étape : mécanisme de sélection d'intersections dynamique ;
- Troisième étape : amélioration de la transmission gloutonne (greedy forwarding) entre deux intersections.

Grâce aux trois étapes précédentes, le protocole de routage est robuste et peut s'adapter à un environnement urbain compliqué. La robustesse de ce protocole vient du fait qu'en cas de changement rapide de la topologie du réseau, ce dernier s'adapte rapidement en estimant la densité du trafic et en déclenchant le processus de routage.

Le protocole de routage GyTAR divise chaque route en petites cellules de zone fixe en définissant un groupe et sélectionne le véhicule central de chaque groupe en tant que leader du groupe. Ensuite, l'information sur la densité du trafic est calculée par chaque chef de groupe et transmise entre les groupes par le biais du CDP (Cell Density Packets). Le CDP recueille la densité d'une route donnée permettant aux véhicules leaders du groupe de collecter la densité de trafic d'une zone adjacente. Le prochain mécanisme de sélection

d'intersections est basé sur la densité du trafic et la distance. À la différence des protocoles de routage GSR (Geographic Source Routing) et A-STAR (Anchor-based Street and Traffic Aware Routing), où le nœud émetteur calcule de façon statique une séquence d'intersections que le paquet doit traverser pour atteindre sa destination, les intersections intermédiaires dans GyTAR sont choisies dynamiquement et en séquence tout en considérant la variation dans la densité du trafic véhiculaire ainsi que la distance vers la destination. Ainsi, si une intersection se trouve aux abords d'une faible densité, le protocole choisira une autre intersection présentant une forte densité véhiculaire. Ceci permet d'améliorer le processus de routage en évitant le bris de lien.

Toutefois, compte tenu du fait que ce protocole ne considère pas les informations concernant la direction des véhicules lors du mécanisme de sélection d'intersections, ce dernier risque d'être confronté au problème du maximum local. C'est-à-dire que l'intersection sélectionnée présente une forte densité, mais le déplacement des véhicules s'oppose à celui du nœud destinataire.

GSR (Geographic Source Routing) ([Lochert et al., 2003](#)) est un protocole de routage basé sur la localisation et destiné aux réseaux VANET. Il se caractérise par l'utilisation de l'algorithme Dijkstra ainsi que le mode de transmission gloutonne le long du chemin présélectionné.

GSR suppose que les véhicules sont dotés d'un GPS (Global Positioning System) et d'une carte numérique. La position précise d'un véhicule ainsi que la structure de la topologie routière sont déterminées par le biais des capteurs embarqués. Afin de surmonter les lacunes du routage basé sur la position telles que : la déconnexion du réseau, les erreurs dues à trop de sauts, les boucles de routage et le déplacement des véhicules dans la mauvaise direction ([Karp, 2001](#)), GSR utilise une carte numérique de la rue, la position du nœud source et la position du nœud de destination pour déterminer le nœud qui recevra le paquet de données dans l'intersection de la route. Ceci permet d'améliorer la précision de routage, c'est-à-dire la prochaine direction vers laquelle le paquet est transmis.

Toutefois, l'analyse des performances de GSR ne prend pas en compte le cas où la densité du trafic pour la transmission de paquets de données est faible ([Gong et al., 2007](#)). En effet, une faible densité du trafic peut affecter la recherche de chemins et donc les performances du protocole.

Synthèse routage basé sur la géographie

Dans la section précédente, nous passons en revue quelques protocoles de routage qui adoptent une approche basée sur la géographie. Certains travaux ([Jerbi et al., 2009](#); [Lochert et al., 2003](#)) exploitent les intersections ainsi que les informations sur la position des véhicules pour améliorer le processus de routage dans les réseaux VANET. Ces travaux démontrent des limites lorsqu'ils sont confrontés au problème du local maximum.

En dépit du problème de local maximum auquel sont confrontés les protocoles de routage géographique, l'exploitation des informations de positionnement lors du processus de routage dans les réseaux VANET permet d'améliorer la performance de ces derniers. Par conséquent, pour la suite de nos travaux, nous considérerons les informations de positionnement.

3.1.3 Protocoles de routage basés sur le trafic

Cette famille de protocole utilise les informations du trafic pour une prise de décision de routage ([Darwish & Bakar, 2016](#)).

A-STAR (Anchor-based Street and Traffic Aware Routing) ([Seet et al., 2004](#)) est un protocole de routage basé sur la position pour un environnement métropolitain caractérisé par des communications IVCs (inter-vehicle communications). Il exploite les informations inhérentes aux itinéraires d'autobus en vue d'identifier une route d'ancre (anchor route) ayant une forte connectivité pour l'acheminement des paquets. A-STAR adopte une approche de routage basée sur l'ancrage (anchor based) en considérant les caractéristiques des rues. Un poids est attribué à chaque rue en fonction de sa capacité (petite, grande, moyenne, etc). Les informations de routes fournies par les bus permettent

de déterminer la densité du trafic au niveau de chaque rue. Ceci permet d'avoir un aperçu de la ville à différents instants.

La particularité de ce protocole est liée au fait qu'il détermine les chemins ancrés (anchor paths) en fonction de la densité du trafic (itinéraires des bus). Toutefois, A-STAR détermine un chemin de la source à la destination en se basant sur les informations statiques des itinéraires de bus de la ville, ce qui provoque un problème de connectivité sur certains segments de rues.

MORA (Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks) ([Granelli, Boato & Kliazovich, 2006](#)) est un algorithme de routage proposé pour faire face à la forte mobilité des nœuds et aux changements fréquents de topologie dans les réseaux VANET. Son approche consiste à exploiter la position des nœuds (source, intermédiaires, destinations) et la direction de mobilité des véhicules pour une prise de décision de routage. Plus spécifiquement, les auteurs intègrent dans l'algorithme une fonction distribuée basée sur la distance (séparant le nœud source et destination) et la direction du déplacement. De cette façon, chaque véhicule est en mesure de faire appel à la fonction lors d'une prise de décision de routage.

L'atout majeur de MORA est l'utilisation d'une métrique qui permet d'exploiter les informations de positionnement et la direction du déplacement des véhicules lors d'une prise de décision de routage. Toutefois, son processus de découverte de route par inondation de paquets conduit à des messages de routages élevés qui peuvent causer la congestion du réseau.

MURU (A Multi-hop Routing protocol for Urban vehicular ad hoc networks) ([Mo, Zhu, Makki & Pissinou, 2006](#)) est un protocole conçu pour les réseaux VANET urbains. En effet, l'approche de ce protocole consiste à déterminer des chemins stables qui maximisent le taux de livraison de paquets tout en minimisant les messages de routages dans les réseaux VANET urbains. Ainsi, pour évaluer le niveau de stabilité d'un chemin de routage, MURU exploite une nouvelle métrique appelée EDD (Expected Disconnected

Degree). Cette métrique permet de calculer la période de rupture d'un chemin pendant un laps de temps prédéfini. La métrique est calculée en fonction de la vitesse prédite, la trajectoire du déplacement de chaque véhicule, puis le chemin ayant la plus petite valeur EDD est sélectionné. De plus, en vue de réduire les messages de routage, ce protocole intègre un mécanisme backoff qui exploite la géométrie de la carte routière pour supprimer les messages de contrôles inutiles. Cependant, ce protocole montre ces limites en cas d'optimum local (c'est-à-dire que le nœud le plus proche de la destination est le nœud courant).

VADD (Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks) ([Zhao & Cao, 2008](#)) est un protocole de routage sensible au retard, conçu pour les réseaux VANET. Ce protocole utilise un mécanisme de transfert de paquets basé sur la position des nœuds expéditeurs. Plus spécifiquement, lors d'une prise de décision de routage, le nœud expéditeur transmet le paquet de données via un chemin qui induit un retard minimum pour la livraison de ce dernier. L'estimation du retard est déterminée en fonction de certains paramètres du trafic (longueur de la route, vitesse moyenne du véhicule et densité du trafic routier), puis exprimée sous forme d'ensembles d'équations linéaires ($n \times n$), n représente le nombre d'intersections (jonctions). Après avoir sélectionné l'intersection, le nœud expéditeur détermine le prochain nœud relais. Sur un segment de route rectiligne, la priorité est accordée au nœud qui est plus proche de l'intersection suivante. Lorsqu'il n'y a pas de nœuds voisins direct se trouvant à portée de transmission du nœud expéditeur, le paquet est transporté jusqu'au prochain voisin approprié. Toutefois, VADD a un inconvénient majeur dû à la scalabilité des réseaux.

Synthèse routage basé sur le trafic

Les travaux de routage relativement au paramètre du trafic mettent en œuvre différentes approches en vue d'améliorer la communication dans les réseaux VANET. En occurrence, certains travaux ([Granelli et al., 2006](#); [Seet et al., 2004](#); [Zhao & Cao, 2008](#)) cherchent à améliorer le routage en se basant sur des critères tels que : la densité du trafic, la position des nœuds (source, intermédiaires et destinations), la direction du déplacement des nœuds, la vitesse moyenne du nœud, la longueur de la route. D'autres travaux ([Mo et](#)

[al., 2006](#)) utilisent une approche basée sur une nouvelle métrique de routage appelée EDD (Expected Disconnection Degree). Cette métrique permet de déterminer le temps de rupture d'une route pendant un laps de temps en vue de sélectionner le chemin le plus robuste (en termes de connectivité) entre la source et la destination. L'on constate également que ces travaux présentent des lacunes telles que :

- Les problèmes de connectivité sur les segments de rues dus à l'utilisation d'une densité statique du trafic lors d'une prise de décision de routage ;
- La congestion du réseau due au processus de découverte de route par inondations de paquets ;
- Le problème d'optimum local ;
- La scalabilité du réseau ;

En dépit des lacunes présentées précédemment, l'exploitation de l'information relative au trafic lors du processus de routage permet d'améliorer la performance des réseaux VANET. En effet, cette information permet de quantifier la densité du trafic véhiculaire et ainsi déterminer la zone du réseau fortement connectée vers où envoyer le paquet de données dans la direction du nœud destinataire. Néanmoins, cette approche nécessite des précisions particulières relativement au trafic à considérer.

Il est important aussi de considérer la qualité d'un protocole de routage relativement au protocole d'accès avec lequel il interagit compte tenu des structures des architectures de communications déployées actuellement. Nous considérons dans la section suivante des travaux récents sur la couche 2 relativement aux avancées des protocoles d'accès. Notre objectif est de tester notre solution relativement à un fonctionnement global optimal.

3.2 Protocoles de contrôle d'accès dans les réseaux VANET

La couche MAC (Media Access Control) présente un élément clé au niveau de la pile protocolaires de communication dans tout environnement réseau. Cette couche est chargée de maintenir et gérer l'accès au canal partagé. Elle détermine le nœud qui peut

accéder au support de communication physique à un instant donné. Les applications temps réel (par exemple : alerter un véhicule d'urgence pour qu'il évite un embouteillage) requièrent des contraintes strictes en termes de délai de transmission pour permettre au conducteur de réagir efficacement face à un incident survenu sur la route. Donc, la conception d'un protocole MAC, doit prendre en considération ces contraintes et fournir un accès efficace au réseau tout en tenant compte des exigences telles que : la qualité de services, l'équité ainsi que l'évolutivité. Les protocoles MAC sont classés suivant les stratégies d'acquisition du canal avec contention et sans contention ([Booyesen, Zeadally & Van Rooyen, 2011](#)). Les approches basées sur la contention s'appuient sur l'écoute du support (carrier sensing), la retransmission et le temps d'attente calculé aléatoirement avant une retransmission (Back-offs), tandis que les approches sans contention reposent sur la synchronisation et l'accès multiple à répartition dans le temps ([Englund, Chen, Vinel & Lin, 2015](#)). Pour la catégorie des protocoles MAC sans contention, on peut citer comme exemples les techniques d'accès TDMA (Time Division Multiple Access), CDMA (Code Division Multiple Access) et FDMA (Frequency Division Multiple Access). Ces techniques pré-allouent l'accès au canal de communication et elles nécessitent une entité centrale afin de coordonner une distribution équitable des ressources. Un exemple pour la catégorie des protocoles MAC avec contention est la technique d'accès CSMA (Carrier Sense Multiple Access). Dans CSMA, un seul canal de communication est partagé pour la diffusion et le délai de transmission de paquets ne peut être garanti à cause des collisions qui peuvent survenir. En outre, le problème de nœud exposé/caché complexifie la garantie d'une transmission fiable.

Une variante du protocole TDMA intitulée Self-organized TDMA ([Bilstrup, K., Uhlemann, Ström & Bilstrup, 2009](#); [Lans, 1996](#)) a été développée et utilisée pour diverses applications industrielles (automatisation d'usine, domaine naval et aéronautique) ([Series, 2010](#)). Contrairement à TDMA qui nécessite une station centrale pour l'attribution des créneaux (slot assignment) et la synchronisation temporelle, STDMA adopte une approche décentralisée pour l'attribution des créneaux. La synchronisation temporelle est prise en compte par le biais du temps universel coordonné (UTC). En outre, STDMA utilise un schéma de réservation afin de permettre aux nœuds de communiquer leurs prochaines

transmissions, ce qui entraîne moins d'éléments aléatoires par rapport à CSMA ([Gaugel, Mittag, Hartenstein, Papanastasiou & Ström, 2013](#)).

De plus, plusieurs travaux ([Bilstrup, K. S., Uhlemann & Strom, 2010](#); [TR, 2012](#)) ont permis de souligner la performance de STDMA pour les applications temps réel impliquant les communications inter-véhiculaires. Compte tenu de sa capacité d'auto-organisation et de ses performances pour les applications temps réels dans les VANET, STDMA constitue un candidat potentiel à utiliser au niveau de la couche MAC en vue d'optimiser les performances de la solution que nous avons développée dans le cadre de cette thèse.

Les problématiques inhérentes aux réseaux VANET tels que : la difficulté de détection de collision, le problème de nœud caché, la fiabilité incertaine, le temps de réponse ainsi que le support de la priorité ont conduit à l'émergence de nouveaux protocoles d'accès MAC. C'est dans cette optique que la technique d'accès MAC RR-Aloha ([Hartenstein et al., 2001](#)) a été proposée. RR-Aloha s'appuie sur le protocole Slotted-aloha et exploite le Time Slot (TS) de façon distribuée. Similaire à RR-Aloha, le protocole d'accès ADHOC MAC ([Borgonovo, Capone, Cesana & Fratta, 2002](#)) est basé sur une structure temporelle. Ce protocole fournit un mécanisme de réservation distribué en vue d'assurer la fiabilité d'une transmission sur un canal de diffusion (broadcast) à un saut. L'inconvénient majeur de cette technique d'accès MAC est lié au fait que le nombre de nœuds qui s'échangent des trames dans une même portée ne doit pas excéder le nombre de TS de la trame temporelle.

Le protocole DRVC (Direct and Relay protocol for Vehicle Communications) ([Daizo, Iwahashi, Bandai & Watanabe, 2004](#)) fut proposé en vue de fournir une accessibilité au-delà des limites du DSRC. Ce protocole aborde le problème du nœud caché en utilisant le mécanisme RTS (Ready To Send) / CTS (Clear To Send). Toutefois, l'utilisation du mécanisme RTS/CTS entraîne un délai supplémentaire, ce qui peut être critique pour les applications temps réel.

D'autres approches d'accès au support de communication telles que : PCF (Point Coordination Function) et DCF (Distributed Coordination Function) ont été introduites par IEEE 802.11. La technique d'accès DCF est utilisée pour les infrastructures ne nécessitant pas de point d'accès, tandis que celle du PCF est appliquée dans une infrastructure dotée d'un point d'accès. DCF s'appuie sur le principe de fonctionnement du CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), ce qui implique d'écouter le réseau avant toute émission. Le IFS (Inter Frame Spaces) représente un paramètre important. Il permet d'indiquer le temps d'observation du support inoccupé avant la transmission d'une station.

La technique d'accès MAC EDCA (Enhanced Distributed Channel Access) a été conçue pour présenter une distinction par rapport à la qualité de service.

Le protocole MAC proposé dans ([Mak, Laberteaux & Sengupta, 2005](#)) est dédié à l'opération multicanaux et exploite les RSUs. Cette étude est axée autour de deux objectifs : s'assurer que les messages de sécurité puissent être transmis toutes les cent millisecondes à proximité des RSUs, et maximiser la bande passante pour les connexions non sécurisées. Pour pouvoir y parvenir, les auteurs utilisent un RSU pour assurer la synchronisation et la coordination des canaux aux véhicules voisins. Le canal de contrôle divise le temps sous forme d'intervalles périodiques régulés, appelés intervalle de répétition (repetition interval). La période sans contention est désignée par CFP (Contention Free Period). Durant cette période, tous les véhicules dans une certaine portée sont interrogés (consultés) et autorisés à transmettre des messages de sécurité. Lorsque le véhicule consulté n'a aucun message de sécurité à envoyer, ce dernier transmet une courte trame nulle. Lors de la période de contention (Contention Period), les actions suivantes peuvent être entreprises :

- Les véhicules situés dans la région du service peuvent passer aux canaux de service, après avoir écouté les annonces de service effectuées par le RSU ;
- Les véhicules peuvent envoyer des messages de sécurité à l'aide du protocole DCF distribué ;

- Les RSUs effectuent une gestion de groupe pour mettre à jour la liste de consultation ou interrogation (c'est-à-dire inclure les nouveaux nœuds et retirer les véhicules qui se déplacent hors portée de la liste de consultation) ;
- Le RSU transmet des balises (beacons) pour indiquer la portée de la prochaine période sans contention (CFP). Les véhicules recevant cette balise restent silencieux jusqu'à ce qu'ils soient interrogés au prochain CFP.
- La prévision de la prochaine période sans contention (CFP) est indiquée par la trame service `release`.

Les performances (en termes de probabilité d'échec de réception de la balise) du protocole ont été analysées dans NS-2 par les auteurs en utilisant le modèle de propagation d'espace libre de Friis (Friis Free-space model) pour les distances courtes et le modèle réfléchi au sol à deux rayons (Two-ray ground-reflection model) pour les longues distances en vue de déterminer la puissance de réception. Les résultats de l'analyse des performances montrent que si les véhicules sont proches du point d'accès (PA), la probabilité d'échec de réception de balises est faible, et donc les véhicules peuvent échanger des messages de sécurité avec une forte probabilité de succès. Ce protocole d'accès pourrait être un candidat potentiel pour améliorer les performances de notre solution, en particulier pour maximiser le délai des canaux de service pour les messages de sécurité.

Dans ([Xu, Mak, Ko & Sengupta, 2004](#)), les auteurs proposent un protocole MAC basé sur un saut pour la diffusion des messages d'alerte de sécurité. Ce protocole permet de transmettre les messages de sécurité avec une forte probabilité en fonction de la durée de vie du paquet. L'étude avait pour but de concevoir un protocole permettant de répéter les paquets pendant la durée de vie de ce dernier. Le principe de base de ce protocole consiste à ajouter une couche d'extension MAC entre les couches MAC et LLC (Logical Link Control). Lorsque le message est transmis par LLC, l'extension MAC passe de l'état repos (IDLE) à l'état génération de la répétition (REPETITION GENERATION). Dans cet état, le système planifie plusieurs répétitions des messages en fonction des intervalles de temps sélectionnés (Time Slots) dépendamment de la durée de vie. Chaque répétition est représentée par un événement ayant un numéro de slot. Les événements sont ordonnés par

des numéros de slot dans une file d'attente de paquets (Packet Event Queue). Lorsque la file d'attente est formée, le système se met en mode veille ou repos. Si un événement paquet survient, l'extension MAC passe de l'état repos à l'état expédition (DISPATCH) en envoyant le paquet vers le MAC. En cas de transmission du paquet par MAC, la couche vérifie si le paquet est nouveau ou non. S'il est nouveau, le paquet est transmis à LLC. Sinon, le paquet est supprimé. L'évaluation de la technique d'accès a été effectuée en utilisant la répétition synchronisée par rapport à une répétition synchronisée globalement à des intervalles de temps fixes (fixed time slots). Puis, un autre test a été réalisé par les auteurs de l'article via une répétition à l'aide de k slots par rapport à chaque slot pour la probabilité k/n en ajoutant le MAC modifié qui exploite la détection de la porteuse (carrier sensing), c'est-à-dire que les paquets sans back-off sont supprimés.

Les résultats issus des expérimentations menées par les auteurs de l'article montrent que les meilleurs protocoles MAC sont ceux basés sur la répétition fixe synchrone et asynchrone avec détection de la porteuse. Sachant que la répétition fixe synchrone nécessite une synchronisation avec l'horloge globale, la répétition fixe asynchrone avec détection de la porteuse est le meilleur choix. Ce résultat s'explique par le fait que la répétition asynchrone donne plus de possibilités d'émission à un nœud tout en garantissant l'équité, tandis que la détection de porteuse provoque l'interférence du nœud pouvant stopper la transmission.

3.3 Synthèse Générale

Sur la base de notre revue de la littérature, nous présentons une synthèse qui permet de souligner les principales contributions et aussi les limitations des travaux existants relativement à la QoS des réseaux VANET.

Le tableau 3.1 récapitule les principaux travaux étudiés en considérant des critères importants pour le bon fonctionnement des applications critiques des STI. Nous analysons également les travaux qui se situent au niveau des principales catégories du routage basé sur des informations géographiques ou ceux liés aux informations en lien avec le trafic du réseau routier. Les critères que nous considérons se définissent comme suit :

- Perception des composantes urbaines : ce critère spécifie la capacité d'un véhicule à prendre en considération les différentes composantes urbaines (immeubles, ponts, arbres, tunnels, etc.) lors des décisions de routage.
- QoS : ce critère permet de mesurer la qualité du réseau en termes de délais de transmission, fiabilité (résistance à l'échec de lien) et connectivité. Cependant le critère de QoS renvoi à la mesure qualitative du réseau VANET pour les applications critiques des STI.
- Routage basé sur la géographie : ce critère caractérise l'impact de la localisation sur la qualité de la transmission. En effet, son importance réside dans le fait d'exploiter les informations inhérentes à la localisation des véhicules pour disséminer les paquets de données en temps réels.
- Routage basé sur le trafic : ce critère caractérise l'impact du trafic sur la qualité de la transmission. En effet, son importance réside dans le fait d'utiliser les informations liées à la densité du trafic routier en temps réels pour déterminer les routes efficaces (par exemple le plus court chemin) permettant de transmettre les données.

Tableau 3.1 : Critères d'analyse des travaux de l'état de l'art

Critères Travaux	Perception Composantes urbaines	Qualité de Services (QoS)	Utilisation paramètre géographique	Utilisation Paramètre du trafic
Gaputa et al (Gupta & Kumar, 2014) .	Non	Oui	Non	Non
H. Saleet et al.(Saleet et al., 2009).	Non	Oui	Non	Non

I. Benyahia et al (Benyahia et al., 2017).	Non	Oui	Non	Non
H. Rana et al(Rana et al., 2013).	Non	Non	Non	Non
D. K. Lobiyal et al (Lobiyal et al., 2015).	Non	Oui	Non	Non
Bitam et al (Bitam et al., 2013).	Non	Non	Non	Non
Karp et al (Karp & Kung, 2000).	Non	Non	Oui	Non
DGR (Bilal et al., 2013).	Non	Non	Oui	Non
A-STAR (Seet et al., 2004).	Non	Non	Non	Oui
VADD (Zhao & Cao, 2008).	Non	Oui	Non	Oui

Légende du tableau :

Oui : critère présent ; Non : critère absent.

En se basant sur le tableau précédent, nous constatons que les travaux ([Gupta & Kumar, 2014](#); [Saleet et al., 2009](#)) présentent des avantages en lien avec les algorithmes génétiques. Ces travaux parviennent à minimiser le retard de transmission dans les réseaux VANET étudiés. Toutefois, ils n'utilisent pas les informations liées au trafic routier ainsi que celles inhérentes à la position des véhicules. Ces deux aspects sont cruciaux au niveau de la prise de décision de routage. Par exemple, la prise en considération des changements

inhérents à la densité du trafic routier en temps réel permet de renforcer ou modifier l'itinéraire actuel. En outre, l'utilisation d'un paramètre lié à localisation des véhicules permet de disséminer les informations en temps réels lors du processus de routage.

Les travaux ([Bitam et al., 2013](#); [Lobiya et al., 2015](#); [Rana et al., 2013](#)) exploitent une approche de routage basée sur les métaheuristiques tels que : l'optimisation par colonies de fourmis ou l'optimisation par essaims particulaires. Ces travaux ont la particularité de considérer plusieurs paramètres (mobilité, densité, vitesse, condition d'évanouissement du canal) lors du processus de routage. Ils visent à garantir une QoS en termes de taux de pertes de paquets, délai moyen de bout-en-bout, charge de routage normalisée tout en assurant une transmission évolutive et robuste à l'échec de liaison. Toutefois, ils sont confrontés aux problèmes liés au nombre élevé de paquets de contrôle et à des risques de congestion induite par des changements rapides de la configuration du réseau.

Étant donné que nous cherchons à développer des solutions pour les situations critiques dans les réseaux VANET, la condition d'évanouissement du canal est l'un des aspects à considérer dans nos travaux de recherche.

L'étude présentée dans ([Benyahia et al., 2017](#)) combine une approche métaheuristiques et architecture cross-layer pour réduire le retard de transmissions et améliorer l'efficacité des réseaux VANET. En dépit des bonnes performances démontrées par l'analyse des résultats de ces travaux, ils nécessitent des améliorations qui pourraient davantage optimiser les délais de transmission pour les applications critiques des STI.

Les travaux ([Bilal et al., 2013](#); [Karp & Kung, 2000](#); [Seet et al., 2004](#); [Zhao & Cao, 2008](#)) utilisent une approche de routage basée sur la position des nœuds et le trafic routier pour une prise de décision de routage efficace dans les réseaux VANET. La particularité de ces travaux réside dans leurs approches de routage qui consiste à exploiter une fonction qui estime le retard de transmissions en tenant compte de plusieurs informations (la longueur de la route, la vitesse moyenne du véhicule et la densité du trafic routier, etc.), ou

à utiliser une approche de routage gloutonne. Toutefois, ces travaux sont confrontés aux problèmes tels que : le local maximum, la congestion due à l'inondation par paquet lors du processus de découverte de route, la scalabilité, la perte de lien lié aux informations imprécises (routes déterminées sur la base d'informations statiques).

L'on constate également que l'ensemble des travaux présentés dans le tableau précédent ne prennent pas en considération le critère lié à la perception des composantes urbaines lors de la prise décision de routage. Ce critère est crucial dans la mesure où elle permet de quantifier l'impact d'un obstacle par rapport à la transmission. Par exemple le scénario présenté par la figure 2.1 dans le chapitre 2 illustre ce problème à travers l'évanouissement par trajets multiples. En occurrence, dans cet exemple, le signal de transmission se dégrade à cause des atténuations qu'il subit par rapport aux obstacles rencontrés sur son trajet. Ce qui peut conduire à un échec de transmission au niveau du récepteur. Le fait de corriger les liens concernés par la dégradation des signaux dans un délai dicté par les échanges entre les couches de l'architecture de communication de VANET va certainement induire un délai additionnel au niveau des transmissions de paquets entre les véhicules.

Afin de définir un réseau VANET ayant un comportement qui lui permet d'optimiser les communications dans les contextes critiques (cas urgents) et compte tenu des analyses présentées dans ce chapitre ainsi que des objectifs du projet présentés dans le chapitre 2, nous définissons pour notre étude les objectifs de recherche suivants :

Objectifs de recherche :

- Tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires ;
- Optimiser le routage des communications véhiculaires pour les applications critiques ;
- Tester notre solution dans un contexte de couche MAC optimale.

Chapitre 4

Vers un nouveau routage basé sur les capacités de véhicules autonomes pour les réseaux VANET

Sommaire du chapitre 4

4.1	Introduction.....	67
4.2	Principe de résolution des objectifs de recherche.....	68
4.2.1	Tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires	68
4.2.2	Optimiser le routage des communications véhiculaires pour les applications critiques	69
4.2.3	Tester notre solution dans un contexte de couche MAC optimale	69
4.3	Étude de cas : tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires	70
4.3.1	Choix des RNA.....	70
4.3.2	Choix de la méthode d'apprentissage	70
4.3.3	Prévision du taux d'erreurs binaires par apprentissage supervisée.....	72
4.3.4	Résultats de l'apprentissage pour la prévision du TEB	84
4.3.5	Analyse des résultats.....	90
4.4	Un Routage Cognitif pour les réseaux VANET	92
4.4.1	Organisation Architecturale du Véhicule Autonome.....	92
4.4.2	Rôle des différents modules du véhicule autonome	93
4.4.3	Structure dynamique de l'interaction des modules.....	94
4.4.4	Structure de la table de routage du protocole CRV	103

4.4.5	Principe de fonctionnement généralisé du protocole CRV.....	105
4.5	Conclusion	129

4.1 Introduction

Dans le chapitre 3, nous avons passé en revue les différentes solutions existantes pour le routage dans les réseaux VANET. Nous avons constaté que toute solution valide pour un environnement particulier ne fonctionne pas dans d'autres types d'environnements. En outre, les solutions de routage actuelles ne permettent pas de répondre aux contraintes de délai de transmission pour les applications VANET critiques des STI supportées par des réseaux VANET tel que présenté dans le chapitre 2. En effet, ce délai est la conséquence directe des retransmissions de paquets perdus dus aux interférences et/ou à l'affaiblissement (évanouissement par trajets multiples, chapitre 2) que subit le signal par rapport aux composantes urbaines de son environnement. Les composantes urbaines de l'environnement sont représentées par des bâtiments tels que présenté dans le chapitre 2.

Compte tenu de la nature complexe du problème lié au délai de transmission pour les applications critiques et celle de l'évanouissement par trajets multiples dans les réseaux VANET abordés au chapitre 2 de cette thèse, notre activité de recherche consistera à proposer de nouvelles solutions pour répondre aux objectifs identifiés dans le chapitre 3. Ces objectifs de recherche sont définis comme suit :

- 1) **Tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires** : élaborer une solution qui permet de doter les nœuds (véhicules autonomes) du réseau de nouvelles capacités (perception, localisation des obstacles statiques et estimation du taux d'erreurs binaires) en vue d'associer le niveau de dégradation du signal aux composantes urbaines perçues.
- 2) **Optimiser le routage des communications véhiculaires pour les applications critiques** : proposer une nouvelle technique de dissémination de l'information pour

les applications critiques des STI en s'appuyant sur les nouvelles capacités des véhicules autonomes ;

- 3) **Tester notre solution dans un contexte de couche MAC optimale** : intégrer un protocole MAC optimisé au niveau de la couche 2 afin de soutenir le processus de routage pour les applications critiques supportées par les réseaux VANET.

Dans la suite du chapitre, nous présenterons à la section 4.2 un bref descriptif du principe de résolution des différents objectifs de recherche ; la section 4.3 sera axée sur une étude de cas permettant de démontrer comment traiter l'impact des composantes urbaines sur les communications véhiculaires ; par la suite, nous introduirons le routage cognitif dans la section 4.4 et nous clôturerons ce chapitre par la conclusion.

4.2 Principe de résolution des objectifs de recherche

4.2.1 Tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires

Afin d'atteindre le premier objectif de recherche, nous représentons le nœud d'un réseau VANET selon la modélisation d'un véhicule autonome qui représente un agent intelligent dans le contexte de notre étude. Celui-ci se caractérise par de nouvelles capacités telles que : la perception, la localisation des obstacles statiques et essentiellement la faculté d'estimation du taux d'erreurs binaires. À cet effet, chaque véhicule autonome sera en mesure de localiser les composantes urbaines (immeubles, ponts, etc.) statiques de son environnement par l'entremise d'une *carte numérique embarquée* et d'en estimer le taux d'erreurs binaires encouru en fonction de la composante urbaine perçue. Pour estimer le niveau de dégradation du signal associé à une composante urbaine de l'environnement, le véhicule autonome utilise ses capacités d'apprentissage relativement aux impacts des composantes urbaines sur les niveaux des signaux de ses transmissions.

4.2.2 Optimiser le routage des communications véhiculaires pour les applications critiques

Pour atteindre le second objectif de cette étude, nous préconisons d'étendre CL-AntHocNet ([Benyahia et al., 2017](#)) développé précédemment. Le nouveau changement apporté aux contributions précédentes consiste à transformer le protocole CL-AntHocNet en un protocole toujours hybride, mais qui intègre les nouvelles capacités des nœuds tel que développé au niveau de l'étape 1 (premier objectif de cette étude) en vue de permettre un comportement cognitif. En effet, tel que conçu, le protocole CL-AntHocNet exploite l'optimisation inter-couches en s'appuyant sur les critères de qualité de lien tels que le SINR et le FER pour le routage des paquets. La nouvelle approche présentée dans ([Benyahia et al., 2017](#)) a permis de réduire le délai de transmission de bout-en-bout. Néanmoins, la solution est plutôt réactive puisque les erreurs sont détectées et ensuite des événements sont déclenchés au niveau de la couche 3 grâce au principe des architectures inter-couches présenté dans cette première solution.

Le changement que nous apportons dans cette étude se traduit par un nouveau comportement du protocole de routage qui devient cognitif via de nouvelles interactions et les informations qui composent sa table de routage.

4.2.3 Tester notre solution dans un contexte de couche MAC optimale

Pour atteindre le troisième objectif de recherche, nous allons déployer notre solution de routage pour les applications critiques des réseaux VANET dans un contexte MAC optimale. Ceci permet de prendre en considération les liens et impacts des couches 2 et 3. En optimisant les opérations de la couche 2, nous serons en mesure de mieux valider la solution que nous développons au niveau du routage (couche 3). Plus précisément, nous nous appuierons sur les résultats des études empiriques issus de la littérature en vue de sélectionner un protocole MAC qui offre une meilleure performance et l'intégrer à notre solution.

Afin de démontrer comment prendre en considération l'impact des composantes urbaines au niveau des communications véhiculaires pour les applications critiques, nous aborderons dans la section suivante une étude de cas détaillée.

4.3 Étude de cas : tenir compte de l'impact des composantes urbaines au niveau des communications véhiculaires

Dans cette section, nous allons présenter une étude de cas détaillée permettant de traiter l'impact des composantes urbaines au niveau des communications véhiculaires pour les applications critiques des STI qui sont supportées par les réseaux VANET. Pour ce faire, l'étude de cas s'appuiera sur l'exemple soulignant l'affaiblissement du signal par trajets multiples illustré par la figure 2.1 du chapitre 2. L'approche proposée est basée sur les réseaux de neurones artificiels (Artificial Neural Networks, ANN ou RNA).

4.3.1 Choix des RNA

Le choix d'utilisation des RNA (voir annexe A) pour cette phase est justifié par le besoin de définir un modèle prédictif capable de généraliser la prédiction pour des situations nouvelles dans un environnement en ligne (i.e. un couplage bidirectionnel NS-3-SUMO). En effet, étant donné que notre étude est caractérisée par la nécessité de mettre en place un modèle capable d'associer le niveau de dégradation par rapport aux obstacles, il est essentiel de choisir un modèle prédictif apte à généraliser la prédiction en ligne pour des situations nouvelles. En outre, les RNA permettent de mieux modéliser l'hétéroscédasticité (données très volatile et variance non constante) par rapport aux autres techniques de prédictions ([Paul & Vishwakarma, 2017](#)). Par conséquent, le choix des RNA pour la suite de nos travaux est guidé par les besoins décrits précédemment.

4.3.2 Choix de la méthode d'apprentissage

Il existe principalement trois méthodes d'apprentissages machines([Borne, Benrejeb & Haggège, 2007](#); [Singh, Thakur & Sharma, 2016](#)), à savoir : l'apprentissage supervisé, l'apprentissage non-supervisé, et l'apprentissage par renforcement. Leurs utilisations dépendent de la nature des données d'apprentissage. Par exemple, si l'on a des données étiquetées, il s'agit d'un problème d'apprentissage supervisé. Dans le cas où les données ne sont pas étiquetées et qu'il faut leur trouver une structure, il s'agit d'un problème d'apprentissage non-supervisé. Si l'on désire que l'algorithme apprenne de ses propres

erreurs, on fait appel à l'apprentissage par renforcement. Plus précisément, la méthode d'apprentissage est confrontée directement à des choix qui peuvent l'a pénalisé (si elle prend une mauvaise décision) ou l'a récompensé (si elle prend une bonne décision).

Le choix d'une méthode d'apprentissage peut se résumer par les critères présentés dans le tableau 4.1 ci-dessous.

Tableau 4.1 : Choix de la méthode d'apprentissage ([Russell & Norvig, 2010](#))

Nature des données	Choix méthodes d'apprentissage
Données étiquetées	Apprentissage supervisé
Données non étiquetées et recherche de structure	Apprentissage non supervisé
Optimisation de la fonction objective avec interaction de l'environnement (récompense)	Apprentissage par renforcement

Compte tenu de l'importance de considérer les impacts des obstacles sur la qualité du signal des transmissions qui se trouvent à proximité de ces derniers, nous cherchons à quantifier le niveau de l'impact sur la qualité d'une transmission. De ce fait, il nous faut des données inhérentes à la structure géométrique (longueur, largeur, hauteur et profondeur) de l'immeuble ainsi que d'autres informations telles que : le rapport signal sur bruit (SNR) et le nombre de bits à transmettre. Ces données sont collectées par le biais de la simulation. Plus précisément, la simulation est basée sur des modèles urbains réalistes, i.e. que les informations des structures géométriques des composantes urbaines sont extraites d'une base de données géographique du monde réel([Fan, Yang, Zipf & Rousell, 2016](#)). En d'autres termes, nous leur associons des étiquettes. Par conséquent, le type d'apprentissage le plus approprié pour notre étude est l'apprentissage supervisé.

N'ayant pas accès aux données réelles, nous les collectons par le biais de la simulation avec un souci de détails et précisions sur les modèles de notre simulation. Notamment, les informations de la carte routière incluant les composantes urbaines sont

recueillies à partir de la base de données géographique OpenStreetMap (OSM) ([Haklay & Weber, 2008a](#)). Puis, les données concernant la structure géométrique des immeubles sont générées à partir de SUMO, tandis que les données liées au SNR et le nombre de bits à transmettre proviennent de la simulation sous NS-3. La combinaison entre le simulateur NS-3 et SUMO nous permet de disposer d'un environnement de simulation connecté. Ce qui nous procure l'avantage d'obtenir le niveau de dégradation du signal par rapport à l'obstacle en temps réel. Cet aspect est important, d'autant plus que nous nous intéressons aux communications véhiculaires pour les applications critiques des STI qui sont supportées par des réseaux VANET. Nous aborderons en détails l'aspect relatif à la collecte des données à la sous-section de 4.3.3.

4.3.3 Prévion du taux d'erreurs binaires par apprentissage supervisée

Dans cette section, nous expliquons la démarche adoptée en vue d'appliquer l'apprentissage supervisé à notre problématique de recherche. Cette démarche est présentée par la figure 4.1 ci-dessous.

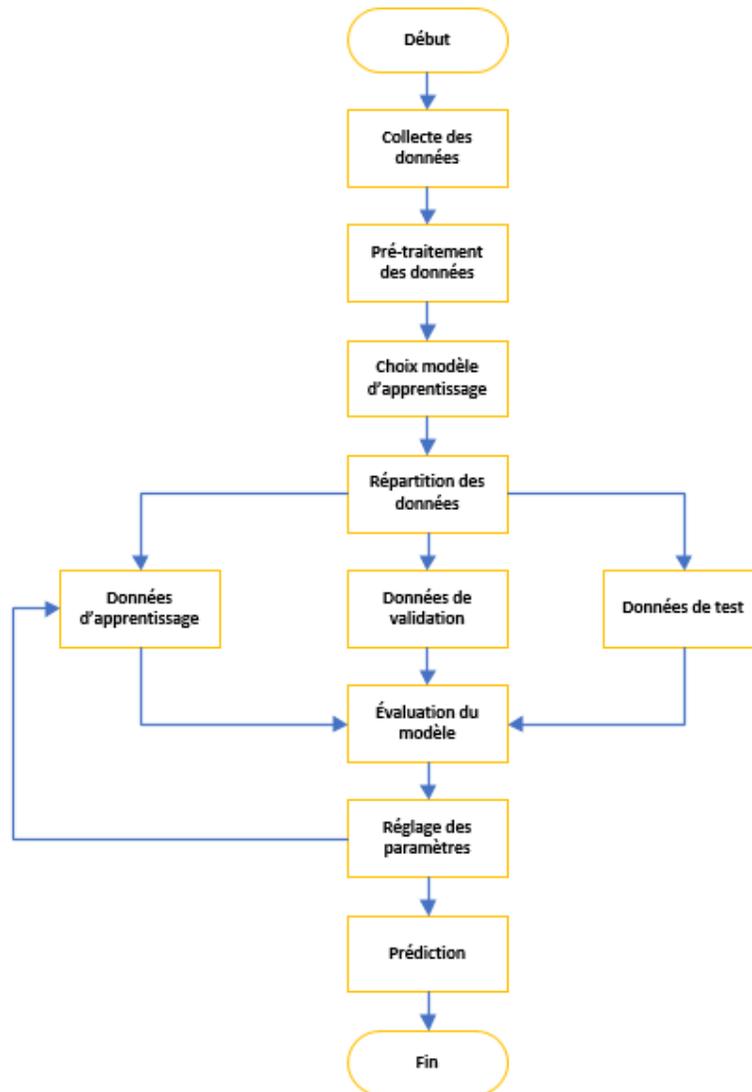


Figure 4.1 : Phases d'apprentissage supervisé

4.3.3.1 Collecte des données d'apprentissage

Cette étape permet la collecte ainsi que la formation d'une base de données de connaissances pour l'apprentissage machine du modèle prédictif. En l'occurrence, les données sont collectées par le biais de la simulation. Ces données représentent la structure géométrique de l'immeuble (longueur, largeur, hauteur, profondeur), le rapport signal sur bruit, le nombre de bits transmis, le nombre de bits reçus, le pourcentage de bits reçus et le taux d'erreurs binaires.

A. Environnement pour la collecte des données

L'environnement de simulation que nous considérons pour la collecte des données est composé de deux simulateurs, à savoir : un simulateur de réseau et un simulateur de trafic. Nous avons choisi d'utiliser NS-3 ([Henderson, Lacage, Riley, Dowell & Kopena, 2008](#)) comme simulateur réseau, car il possède un ensemble de composants réseaux diversifiés qui permettent de mener différentes études concluantes en fonction des protocoles à analyser. L'utilisation de NS-3 est de plus en plus populaire dans le domaine des études liées aux technologies réseaux sans fil ([Arbabi & Weigle, 2010](#)). NS-3 supporte également les normes IEEE 802.11 PHY/MAC, 1609/WAVE et 802.11p ([Bu, Tan, Ding, Liu & Son, 2014](#)). L'ensemble de ces normes permettent de mener des études réalistes sur les réseaux ad-hoc véhiculaires (Vehicular Ad-hoc Network, en abrégé VANet). En outre, NS-3 prend en charge la technologie DSRC (Dedicated Short Range Communication), utilisée en Amérique du Nord pour les tests réels des communications véhiculaires.

Afin de disposer d'un modèle de mobilité conforme aux véhicules, nous avons choisi d'utiliser SUMO (Simulation of Urban MObility) ([Behrisch, Bieker, Erdmann & Krajzewicz, 2011](#)) comme simulateur de trafic. Il est fréquemment utilisé pour les études en rapport avec le trafic dans un réseau routier et est hautement portable. Par ailleurs, SUMO offre la possibilité d'interagir avec une application externe par le biais d'une connexion socket. Celle-ci permet d'effectuer des simulations en mode client/serveur par l'entremise d'une interface de contrôle de trafic dénommée TraCi (Traffic Control Interface). La figure 4.2 ci-dessous illustre notre environnement d'expérimentation pour la collecte des données.



Figure 4.2 : Environnement pour la collecte des données

B. Spécification des scénarios dans SUMO pour la collecte des données

Cette sous-section, présente un bref descriptif de la démarche adoptée pour la spécification des scénarios via le simulateur urbain SUMO. Différents outils incorporés dans SUMO ont été utilisés pour spécifier nos composantes urbaines incluant les obstacles tels que les immeubles. La figure 4.3 ci-dessous illustre le processus de génération des scénarios urbains dans SUMO.

Dans le cadre de notre étude, nous avons d'abord besoin de représenter la structure géométrique du réseau en tenant compte des composantes urbaines tels que : les immeubles. Pour y parvenir, nous utilisons les outils NETCONVERT et POLYCONVERT fournis avec le simulateur de trafic SUMO.

NETCONVERT permet de convertir le fichier (par exemple OSM) qui contient les informations sur la spécification de la structure géométrique des réseaux routiers en provenance de sources différentes vers un format lisible par le simulateur de trafic SUMO. En l'occurrence, pour nos travaux de recherches, nous importons la spécification de la structure géométrique du réseau routier à partir de la plateforme OpenStreetMap ([Haklay & Weber, 2008b](#)).

POLYCONVERT permet de convertir la structure des formes géométriques (immeubles) en provenance de différentes sources en une représentation visualisable par l'interface graphique du simulateur urbain SUMO.

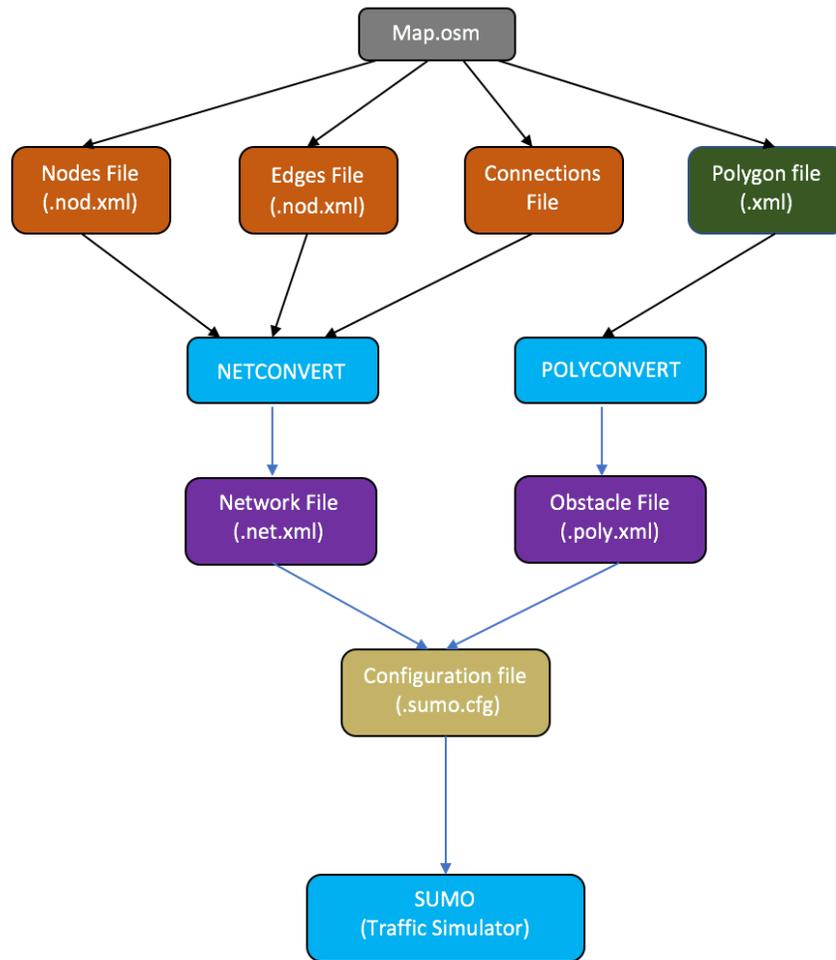


Figure 4.3 : Processus de spécification du scénario urbain via SUMO

Une fois qu'un modèle des composantes urbaines est établi, l'étape suivante consiste à développer un script dans NS-3 en fonction de ce dernier. Bien que les données concernant la structure géométrique des immeubles soient obtenues à partir de SUMO, celles-ci demeurent fiables. Cette fiabilité relève de la faculté de SUMO à extraire les informations directement de la base de données géographique du monde réelle telle qu'OpenStreetMap. Plus spécifiquement, pour chaque composante urbaine de la simulation, les valeurs de la structure géométrique (coin supérieur gauche, coin inférieur gauche, coin supérieur droite et coin inférieur gauche) correspondent aux informations du monde réel. Ainsi, dans NS-3, nous pouvons associer le SNR à une structure géométrique

réelle. Ce qui nous permet de déduire le niveau de dégradation en fonction des composantes urbaines dans notre modèle prédictif.

C. Spécification des composantes urbaines d'un réseau de transport

Dans cette sous-section, nous verrons deux exemples permettant de souligner la spécification des composantes urbaines d'un réseau de transport via SUMO.

a. Dégradation du signal entre deux véhicules dû à la présence d'un immeuble

Dans cette sous-section, nous présentons un exemple descriptif d'un modèle urbain que nous avons spécifié via SUMO. Le scénario considéré se décrit comme suit. À l'instant t_0 deux véhicules V_1 et V_2 (émetteur et récepteur) s'engagent sur un tronçon de route. À $t_0 + \Delta t$, la transmission entre V_1 et V_2 est perturbée par la présence d'un obstacle. Ce scénario met en évidence l'impact d'un obstacle au niveau de la dégradation du signal. Notamment, le scénario permet d'illustrer le problème l'affaiblissement du signal par trajets multiples dû aux obstacles présenté dans le chapitre 2 de cette thèse.

Étant donné que dans notre étude, nous cherchons à quantifier la dégradation du signal en fonction d'un obstacle, nous limitons la transmission à deux véhicules. Ainsi, nous pourrions mieux appréhender l'impact d'un obstacle sur le signal de transmission.

b. Affaiblissement du signal dû à la présence d'un pont

Dans ce scénario, nous supposons qu'à l'instant t_0 deux véhicules V_1 et V_2 s'engagent sur un pont. À $t_0 + \Delta t$, la transmission entre V_1 et V_2 est affectée par la structure géométrique du pont. Ce scénario permet d'illustrer le problème d'affaiblissement par trajets multiples causé par la présence d'un obstacle tel qu'un pont (voir chapitre 2).

Il convient de noter que pour la suite de ce travail, nous appliquons la démarche illustrée par la figure 4.3 pour spécifier tous nos scénarios urbains en modifiant la structure de l'obstacle par le biais du fichier qui contient la description géométrique de celle-ci (fichier.poly.xml). Ces scénarios s'appliquent également dans le contexte de véhicules d'urgence. En l'occurrence, lors de nos travaux de recherche publiés dans ([Benyahia et al.](#),

2017), nous démontrons à l'aide d'exemple l'impact d'un retard lors d'une communication d'urgence.

D. Paramètre de simulation NS-3 pour la collecte des données

Le tableau 4.2 ci-dessous présente un récapitulatif des paramètres de simulation que nous avons utilisés sur NS-3 pour former la base de données de connaissance destinée au système d'apprentissage.

Tableau 4.2 : Paramètres d'expérimentation sur NS-3.

Paramètre	Valeur
Type de données	CBR
Protocole de transport	UDP
Temps de simulation	300 secondes
Nombre de véhicules	2
Modèle de propagation radio	ItuR1411LosPropagationLossModel, Two-Ray Ground
Nombre de bits à transmettre	1500 bits
Protocole Mac	IEEE 802.11p
Portée de transmission	250 m

Le tableau 4.3 ci-dessous présente les résultats issus de la simulation et tient compte des paramètres d'expérimentation présenté précédemment par le tableau 4.2. Sur ce tableau, les informations représentées par les colonnes X_{min} , Y_{min} , X_{max} , Y_{max} spécifient la structure d'un immeuble en termes de coin inférieur gauche, coin supérieur gauche, coin inférieur droit et coin supérieur droit. Les autres informations sont représentées par le nombre de bits à envoyer, le nombre de bits reçus, le pourcentage de bits reçus ainsi que le taux d'erreurs binaire.

Tableau 4.3 : Résultats illustrant la simulation NS-3

#OBS	X_{min}	Y_{min}	X_{max}	Y_{max}	SNR	Nombre de bits envoyés	Nombre de Bits reçus	Pourcentage de bits reçus (en %)	Taux erreurs binaires (en %)
1	762	830	1480	277	25	1500	795	53	47
2	163	497	1435	839	23	1500	851	57	43
3	700	726	131	884	20	1500	936	62	38
4	1786	483	1151	81	17	1500	1021	68	32
5	36	672	908	283	14	1500	1105	74	26
6	1718	2	1107	523	10	1500	1218	81	19
7	744	917	1880	969	8	1500	1274	85	15
8	1994	789	226	459	6	1500	1331	89	11
9	1007	831	1981	325	22	1500	880	59	41
10	688	11	393	437	21	1500	908	61	39
11	898	828	526	830	19	1500	964	64	36
12	109	574	1651	867	18	1500	992	66	34
13	732	40	1997	471	12	1500	1162	77	23
14	176	725	847	43	16	1500	1049	70	30
15	1049	894	1566	477	13	1500	1133	76	24
16	1928	557	204	958	20	1500	936	62	38
17	1417	409	1016	192	7	1500	1303	87	13
18	1532	845	532	156	5	1500	1359	91	9
19	378	849	882	181	4	1500	1387	92	8
20	684	78	484	141	9	1500	1246	83	17
21	1815	26	977	328	27	1500	739	49	51
22	39	264	1227	517	15	1500	1077	72	28
23	1357	96	151	976	33	1500	569	38	62
24	1752	54	1094	244	40	1500	372	25	75
25	1990	667	1013	637	35	1500	513	34	66
26	707	402	147	856	50	1500	90	6	94
27	1634	839	169	213	45	1500	231	15	85

Pour obtenir les résultats du tableau 4.3 présenté précédemment, nous utilisons les informations de la structure géométrique des immeubles à travers notre script de simulation en s'appuyant sur les paramètres de simulation présentés dans le tableau 4.2. Puis, nous collectons les résultats issus de la simulation.

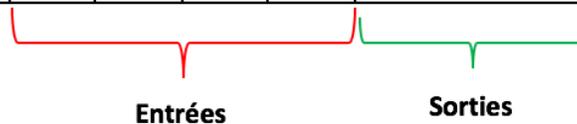
Sur le tableau 4.3 précédent, nous constatons, que chaque structure géométrique d'un immeuble a un impact particulier sur la qualité de transmission. Nous pouvons lire ces informations par les biais des variables qualitatives telles que : le rapport signal sur bruit (SNR), le taux d'erreurs binaires par rapport au nombre de bits transmis, etc.

4.3.3.2 Prétraitements des données collectées par simulations

Lors de cette phase, nous effectuons une transformation des données (paramètres d'entrées et sorties) pour les rendre plus utilisables par l'algorithme d'apprentissage. En l'occurrence, nous effectuons une première transformation consistant à éliminer les informations redondantes telles que : le nombre de bits reçus et le pourcentage de bits reçus. En effet, ces informations dépendent de la variable prédite TEB (Taux Erreur Binaire) et ne sont pas indispensables pour le système d'apprentissage. Ensuite, nous effectuons une seconde transformation consistant à normaliser la représentation de la structure géométrique de l'immeuble par un vecteur composé de quatre variables (X_{min} , Y_{min} , X_{max} , Y_{max}). Cette normalisation permet d'apporter une homogénéité au niveau de la représentation de la variable inhérente à la structure des immeubles. Le tableau 4.4 ci-dessous présente un extrait de données prétraitées tout en spécifiant les données d'entrées et sortie de notre modèle d'apprentissage.

Tableau 4.4 : Extrait des données d'entrées prétraitées pour l'apprentissage

#Obs	L	I	H	P	Taux erreurs binaires (en %)
1	762	830	1480	277	47
2	163	497	1435	839	43
3	700	726	131	884	38
4	1786	483	1151	81	32
5	36	672	908	283	26
6	1718	2	1107	523	19
7	744	917	1880	969	15
8	1994	789	226	459	11
9	1007	831	1981	325	41
10	688	11	393	437	39
11	898	828	526	830	36
12	109	574	1651	867	34
13	732	40	1997	471	23
14	176	725	847	43	30
15	1049	894	1566	477	24
16	1928	557	204	958	38
17	1417	409	1016	192	13
18	1532	845	532	156	9
19	378	849	882	181	8
20	684	78	484	141	17
21	1815	26	977	328	51
22	39	264	1227	517	28
23	1357	96	151	976	62
24	1752	54	1094	244	75
25	1990	667	1013	637	66
26	707	402	147	856	94
27	1634	839	169	213	85



4.3.3.3 Choix de la fonction d'apprentissage pour le TEB

La prochaine étape de notre méthodologie consiste à choisir une fonction d'apprentissage pour notre modèle de prévision du TEB. Pour y parvenir, nous exploitons

l'Erreur Quadratique Moyenne (EQM) comme critère de choix. Le tableau 4.5 ci-dessous présente les résultats du test de différentes fonctions d'apprentissage sur MATLAB.

Tableau 4.5 : Test des fonctions d'apprentissage

N°	Nombre exécution	Algorithmes d'apprentissage	Acronyme	Nombre de neurones	Erreur Quadratique Moyenne (EQM)		
					Min	Max	Moyenne
1	15	Levenberg- Marquardt	LM	10	0.0366	0.0385	0.0377
2	15	Bayesian Regularization	BR	10	0.0826	0.0826	0.0826
3	15	BFGS Quasi- Newton	BFG	10	0.0404	0.0404	0.0404
4	15	Resilient Backpropagation	RP	10	0.0377	0.0426	0.0399
5	15	Scaled Conjugate Gradient	SCG	10	0.0352	0.0409	0.0378
6	15	Conjugate Gradient with Powell/Beale Restarts	CGB	10	0.0328	0.0363	0.0340
7	15	Fletcher-Powell Conjugate Gradient	CGF	10	0.0379	0.0429	0.0397
8	15	Polak-Ribière Conjugate Gradient	CGP	10	0.0363	0.0401	0.0377
9	15	One Step Secant	OSS	10	0.0351	0.0465	0.0400
10	15	Variable Learning Rate Backpropagation	GDX	10	0.0469	0.0601	0.0527

Le tableau 4.5 présente les performances de différentes fonctions d'apprentissage testées sur Matlab. L'erreur quadratique moyenne (Mean Square Error ou MSE) est utilisée comme critère de choix. Nous choisissons la fonction d'apprentissage qui minimise le MSE pour le développement de notre modèle prédictif. En l'occurrence, nous utilisons la

fonction d'apprentissage Conjugate Gradient with Powell/Beale Restarts (CGB) pour notre modèle prédictif, car ce dernier présente la plus basse valeur de MSE.

4.3.3.4 Répartition des données collectées par simulation

Cette étape nous permet de répartir les données en données d'apprentissage, de test et validation. Ci-dessous un exemple de répartition de données :

- 70 % des données pour l'apprentissage.
- 15 % des données pour la validation.
- 15 % des données pour le test.

4.3.3.5 Évaluation du modèle pour la prévision du TEB

L'étape d'évaluation nous permet de tester notre modèle en fonction des données qui n'ont jamais été utilisées pour l'apprentissage. Ceci permet de voir comment le modèle peut fonctionner avec des données dont il n'a pas connaissance. En d'autres termes, l'étape d'évaluation représente la façon dont le modèle pourrait fonctionner dans le monde réel.

4.3.3.6 Calibrage des paramètres

Le calibrage est une étape indispensable pour l'amélioration du modèle prédictif développé. Dans le cas présent, le besoin de calibrage est justifié par la nécessité d'éviter le problème de sur-apprentissage. Ce qui implique de trouver un compromis entre la performance d'apprentissage (capacité de généralisation) et la précision du modèle prédictif. Par conséquent, nous avons agi sur deux paramètres (nombres de neurones cachés et fonction d'apprentissage) afin de calibrer correctement notre modèle prédictif. Nous avons testé différentes fonctions d'apprentissage disponible dans MATLAB en faisant varier le nombre de neurones cachés en vue d'identifier la fonction d'apprentissage adaptée à notre problématique. Les résultats de la figure 4.4 nous montrent une bonne performance de la fonction d'apprentissage CGB pour notre problématique lorsque nous utilisons dix neurones cachés.

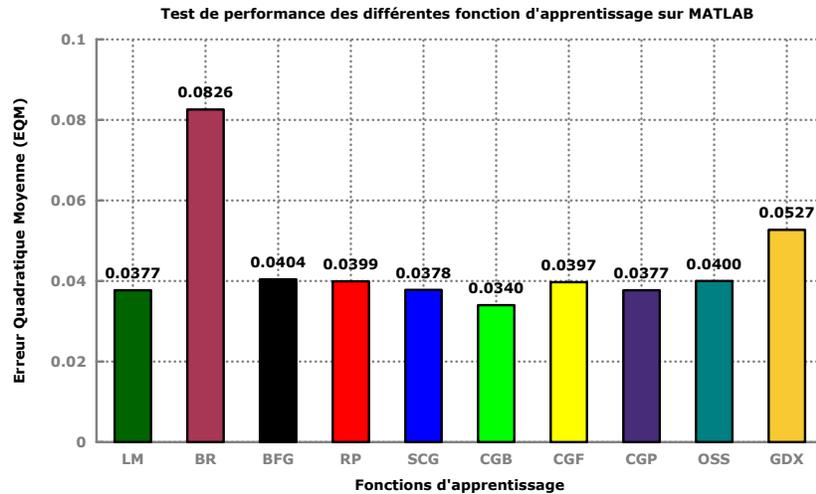


Figure 4.4 : Fonctions d'apprentissage testées sur MATLAB

4.3.3.7 Prédiction du taux d'erreurs binaires

Cette étape permet d'utiliser le modèle ayant appris pour faire une prédiction sur des nouvelles données. En l'occurrence, nous fournissons trois cents nouvelles lignes (X_{min} , Y_{min} , X_{max} , Y_{max}) au modèle et ce dernier nous prédit une sortie correspondant aux taux d'erreurs binaires.

4.3.4 Résultats de l'apprentissage pour la prévision du TEB

Les résultats sont présentés en fonction du nombre de neurones cachés. Ces résultats sont représentés par des tableaux qui indiquent la précision maximale, minimale et moyenne des valeurs du taux d'erreurs binaires prédits. La figure 4.5 illustre le modèle développé pour la prédiction du TEB.

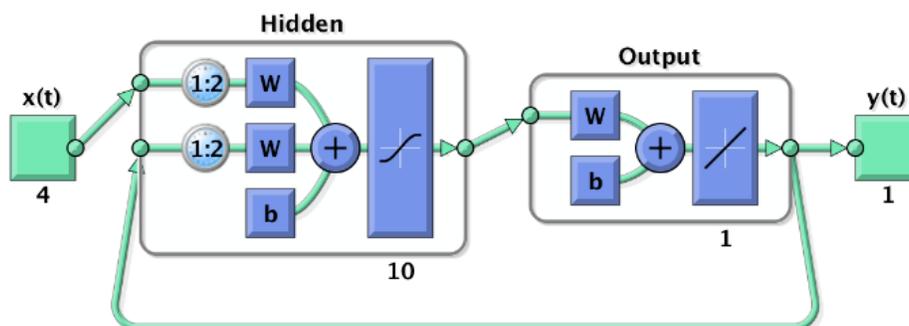


Figure 4.5 : Modèle d'apprentissage pour la prévision du TEB

4.3.4.1 Résultats de l'apprentissage via dix neurones cachés

Cette sous-section présente les résultats de l'apprentissage en fonction de dix neurones cachés. Les détails sont fournis par le tableau 4.6 ci-dessous.

Tableau 4.6 : Précision du modèle basé sur 15 itérations d'apprentissage et 10 neurones cachés

Session apprentissage	Erreur quadratique moyenne (MSE)		
	Minimum	Maximum	Moyenne
1	3,63E-02	3,63E-02	3,63E-02
2	3,43E-02	3,63E-02	3,53E-02
3	3,34E-02	3,63E-02	3,47E-02
4	3,33E-02	3,63E-02	3,43E-02
5	3,32E-02	3,63E-02	3,41E-02
6	3,30E-02	3,63E-02	3,39E-02
7	3,29E-02	3,63E-02	3,38E-02
8	3,28E-02	3,63E-02	3,37E-02
9	3,28E-02	3,63E-02	3,36E-02
10	3,28E-02	3,63E-02	3,35E-02
11	3,28E-02	3,63E-02	3,34E-02
12	3,28E-02	3,63E-02	3,34E-02
13	3,28E-02	3,63E-02	3,33E-02
14	3,28E-02	3,63E-02	3,33E-02
15	3,28E-02	3,63E-02	3,33E-02
Moyenne	3,28E-02	3,63E-02	3,40E-02

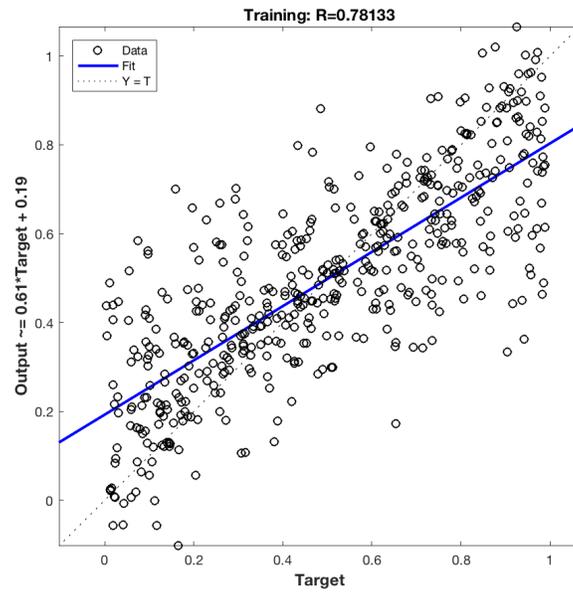


Figure 4.6 : Parcelle de régression du réseau pour l'apprentissage avec 10 neurones

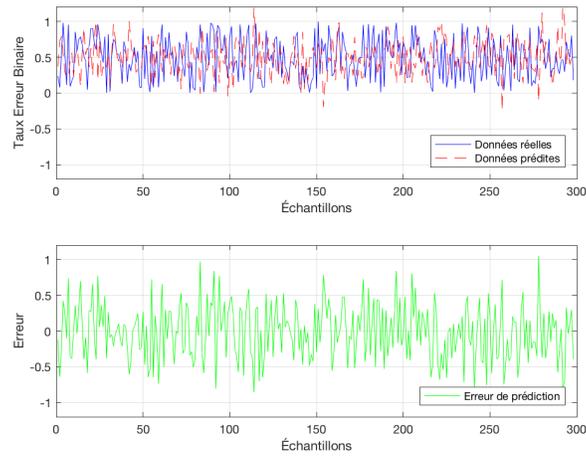


Figure 4.7 : Prédiction du TEB sur 300 échantillons en utilisant 10 neurones

4.3.4.2 Résultats de l'apprentissage via quinze neurones cachés

Cette sous-section présente les résultats de l'apprentissage en fonction de quinze neurones cachés. Les détails sont fournis par le tableau 4.7 ci-dessous.

Tableau 4.7 : Précision du modèle basé sur 15 itérations d'apprentissage et 15 neurones cachés

Session apprentissage	Erreur quadratique moyenne (MSE)		
	Minimum	Maximum	Moyenne
1	0,0240	0,0240	0,0240
2	0,0203	0,0240	0,0222
3	0,0198	0,0240	0,0214
4	0,0191	0,0240	0,0208
5	0,0190	0,0240	0,0205
6	0,0189	0,0240	0,0202
7	0,0186	0,0240	0,0200
8	0,0186	0,0240	0,0198
9	0,0185	0,0240	0,0197
10	0,0185	0,0240	0,0195
11	0,0185	0,0240	0,0194
12	0,0185	0,0240	0,0194
13	0,0185	0,0240	0,0193
14	0,0185	0,0240	0,0193
15	0,0185	0,0240	0,0192
Moyenne	0,0185	0,0240	0,0203

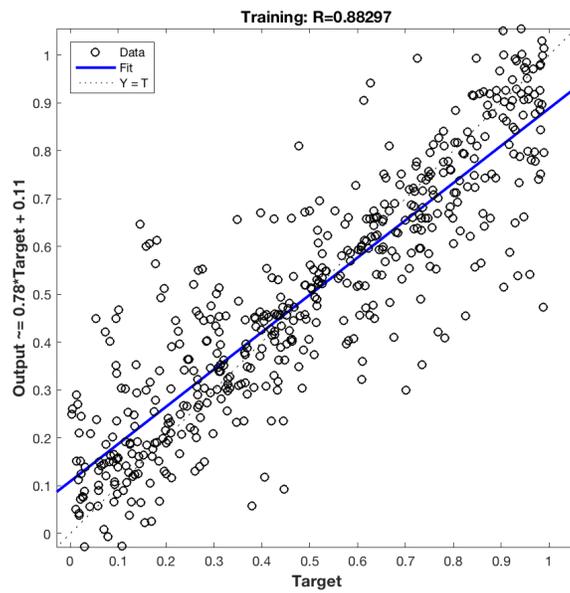


Figure 4.8 : Parcelle de régression du réseau pour l'apprentissage avec 15 neurones

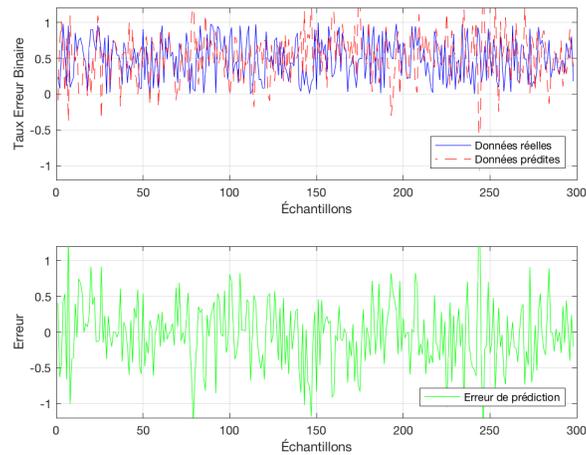


Figure 4.9 : Prédiction du TEB sur 300 échantillons en utilisant 15 neurones

4.3.4.3 Résultats de l'apprentissage via vingt neurones cachés

Cette sous-section présente les résultats de l'apprentissage en fonction de vingt neurones cachés. Les détails sont fournis par le tableau 4.8 ci-dessous.

Tableau 4.8 : Précision du modèle basé sur 15 itérations d'apprentissage et 20 neurones cachés

Session apprentissage	Erreur quadratique moyenne (MSE)		
	Minimum	Maximum	Moyenne
1	0,0137	0,0137	0,0137
2	0,0112	0,0137	0,0125
3	0,0102	0,0137	0,0117
4	0,0096	0,0137	0,0112
5	0,0095	0,0137	0,0108
6	0,0093	0,0137	0,0106
7	0,0093	0,0137	0,0104
8	0,0093	0,0137	0,0103
9	0,0093	0,0137	0,0102
10	0,0093	0,0137	0,0101
11	0,0093	0,0137	0,0100
12	0,0093	0,0137	0,0100
13	0,0093	0,0137	0,0099
14	0,0093	0,0137	0,0099
15	0,0093	0,0137	0,0098
Moyenne	0,0093	0,0137	0,0107

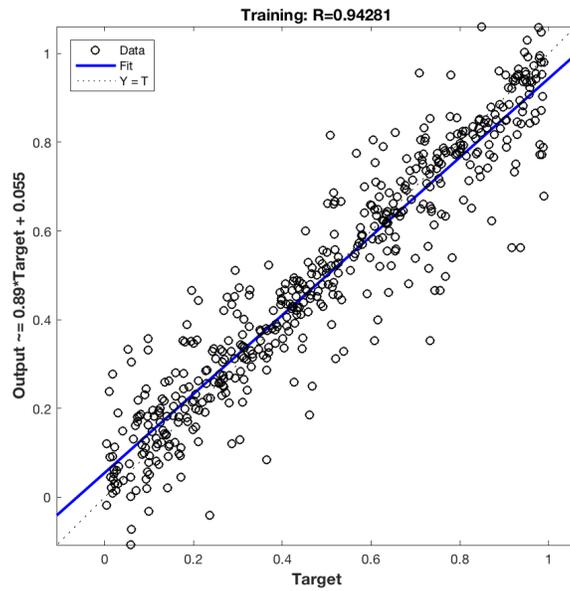


Figure 4.10 : Parcelle de régression du réseau pour l'apprentissage avec 20 neurones

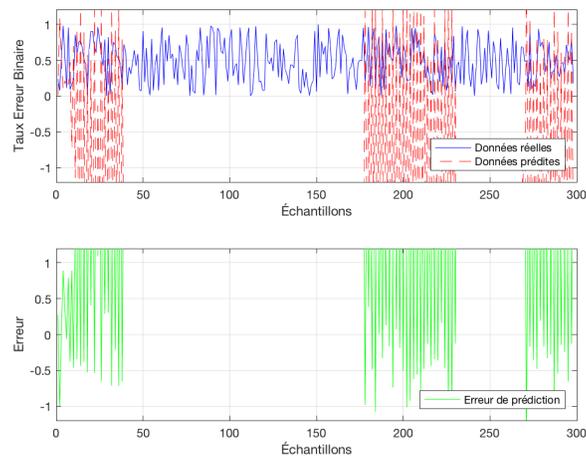


Figure 4.11 : Prédiction du TEB sur 300 échantillons en utilisant 20 neurones

4.3.5 Analyse des résultats

Les figures 4.6, 4.8 et 4.10 présentent les résultats des différentes parcelles de régression pour l'apprentissage avec 10, 15 et 20 neurones. Si le coefficient de corrélation R est proche de 1, l'apprentissage est décrit comme solide, tandis qu'un coefficient de corrélation proche de 0 décrit un apprentissage faible.

Les parcelles de régression affichent les sorties du réseau en fonction des cibles d'un ensemble de données d'apprentissage, de validation ou test. Un ajustement parfait implique une concordance des données le long de la ligne de 45 degrés (les sorties du réseau concordent avec les cibles).

Dans notre cas, l'apprentissage affiche un coefficient de corrélation $R = 0.78$ lorsqu'on utilise 10 neurones, $R = 0.88$ lorsqu'on utilise 15 neurones et $R = 0.94$ lorsqu'on utilise 20 neurones. Ce qui indique une bonne performance au niveau de l'apprentissage.

Les figures 4.7, 4.9 et 4.11 illustrent l'impact de la variation du nombre de neurones cachés par rapport à la prédiction. Ainsi, lorsqu'on utilise 15 ou 20 neurones, le coefficient de corrélation indique une bonne performance d'apprentissage ($R = 0.88$ et $R = 0.94$) par rapport au réseau ayant appris avec 10 neurones. Toutefois, la capacité de généralisation des modèles prédictifs basés sur 15 et 20 neurones se dégradent par rapport à celui qui est basé sur 10 neurones. En effet, sur la figure 4.7, on peut constater que la prédiction est beaucoup plus précise et l'erreur de prédiction est comprise entre -1 et 1, tandis que sur les figures 4.9 et 4.11 les modèles prédictifs perdent leurs précisions et l'erreur de prédiction est beaucoup plus élevée. Cette perte de précision est liée au problème de sur-apprentissage induit par l'augmentation du nombre de neurones cachés. En effet, même si l'augmentation du nombre de neurones cachés permet d'améliorer le processus d'apprentissage, elle induit également à la perte de précision due au sur-apprentissage. En fonction du problème de prédiction étudié, il est crucial de faire un compromis entre performance d'apprentissage et précision du modèle prédictif. Dans le cas de notre étude, les résultats démontrent que le modèle prédictif basé sur 10 neurones cachés présente un meilleur compromis entre performance d'apprentissage et précision au niveau de la prédiction.

En conclusion, dans notre cas, il est préférable de calibrer notre réseau avec 10 neurones cachés pour obtenir une performance de prédiction optimale.

Dans la section suivante, nous introduirons un nouveau type de routage basé sur le modèle de prévision du TEB présenté précédemment.

4.4 Un Routage Cognitif pour les réseaux VANET

Dans cette section, nous présentons le protocole destiné au Routage Cognitif pour VANET que nous appelons "*Cognitive Routing Protocol for VANET*" ou (CRV) dans la suite du document. Celui-ci représente la solution que nous définissons pour atteindre l'objectif 2 de notre travail. CRV est un protocole de routage hybride, inspiré du protocole AntHocNet ([Di Caro, Ducatelle & Gambardella, 2004](#)).

Le fonctionnement du protocole CRV s'appuie sur l'agent intelligent (véhicule autonome). Le véhicule autonome est doté d'une architecture interne basée principalement sur trois modules (perception, prédiction et communication). Ces modules permettent d'étendre les capacités réactives d'un réseau VANET vers un réseau cognitif. De ce fait, dans la suite de ce travail, nous allons décrire les différents aspects cognitifs caractérisant le protocole CRV par le biais des points énumérés ci-dessous :

- L'organisation architecturale du véhicule autonome
- Le rôle des différents modules du véhicule autonome
- La structure dynamique de l'interaction des modules
- La structure de la table de routage du protocole CRV

4.4.1 Organisation Architecturale du Véhicule Autonome

Dans le cadre de ce travail, nous modélisons un véhicule autonome comme une entité comprenant principalement trois modules, à savoir : le module de perception, le module de prédiction et le module de communication. Chaque module permet d'exécuter une fonctionnalité du véhicule autonome. La figure 4.12 ci-dessous présente l'organisation logique de l'architecture du système du véhicule autonome.

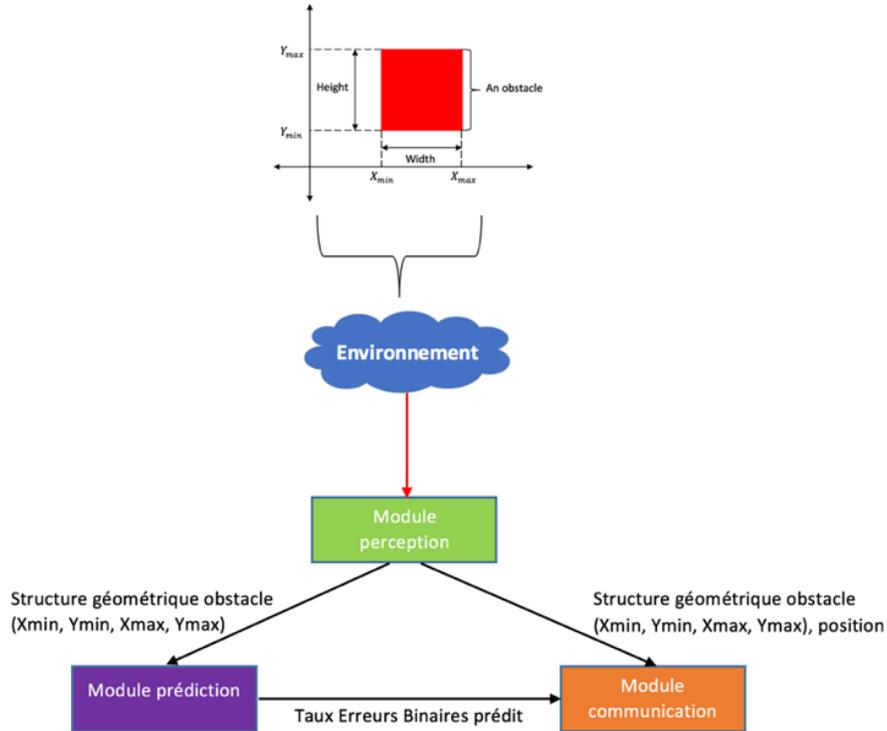


Figure 4.12 : Organisation architecturale du véhicule autonome.

4.4.2 Rôle des différents modules du véhicule autonome

Rôle du module de perception : Ce module perçoit les composantes statiques de l'environnement telles que les immeubles en s'appuyant sur une carte numérique embarquée. La figure 4.13 ci-dessous illustre le processus.

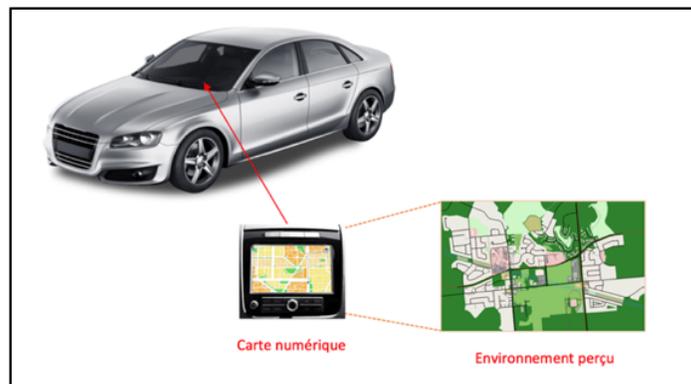


Figure 4.13 : Perception des obstacles statiques dans CRV

Rôle du module de prédiction : Ce module permet de prédire le TEB d'un signal allant d'un nœud V_1 (véhicule courant) vers un autre en fonction des composantes statiques de l'environnement de V_1 tels que les immeubles.

Rôle du module de communication : Ce module met à jour localement la table de liens au niveau de l'agent intelligent. Il envoie ensuite un événement à l'architecture de communication pour interrompre le routage et poursuivre le processus suivant les nouvelles informations de routage reçues via l'agent intelligent.

Dans la prochaine section, nous présentons la structure dynamique permettant d'illustrer l'interaction entre les trois modules décrits.

4.4.3 Structure dynamique de l'interaction des modules

Dans cette section, nous présentons la structure dynamique permettant d'illustrer les interactions entre les modules lors du processus du routage cognitif. En outre, cette structure souligne également notre apport relativement à l'architecture du nœud (véhicule autonome) et celle des communications véhiculaires pour les situations critiques des réseaux VANET. Ainsi, sur la figure 4.14 ci-dessous, la partie illustrée en pointillé vert constitue l'architecture des communications véhiculaires, tandis que la partie représentée en pointillé bleu constitue la nouvelle architecture du véhicule autonome doté de capacités cognitives.

Puisque la composante proactive est responsable de la maintenance et de l'exploration de nouveaux chemins, nous choisissons de transmettre l'événement d'interruption de routage vers cette dernière. Ainsi, la recherche de routes alternatives s'effectue en fonction des informations relatives aux liens valides au niveau de la composante proactive.

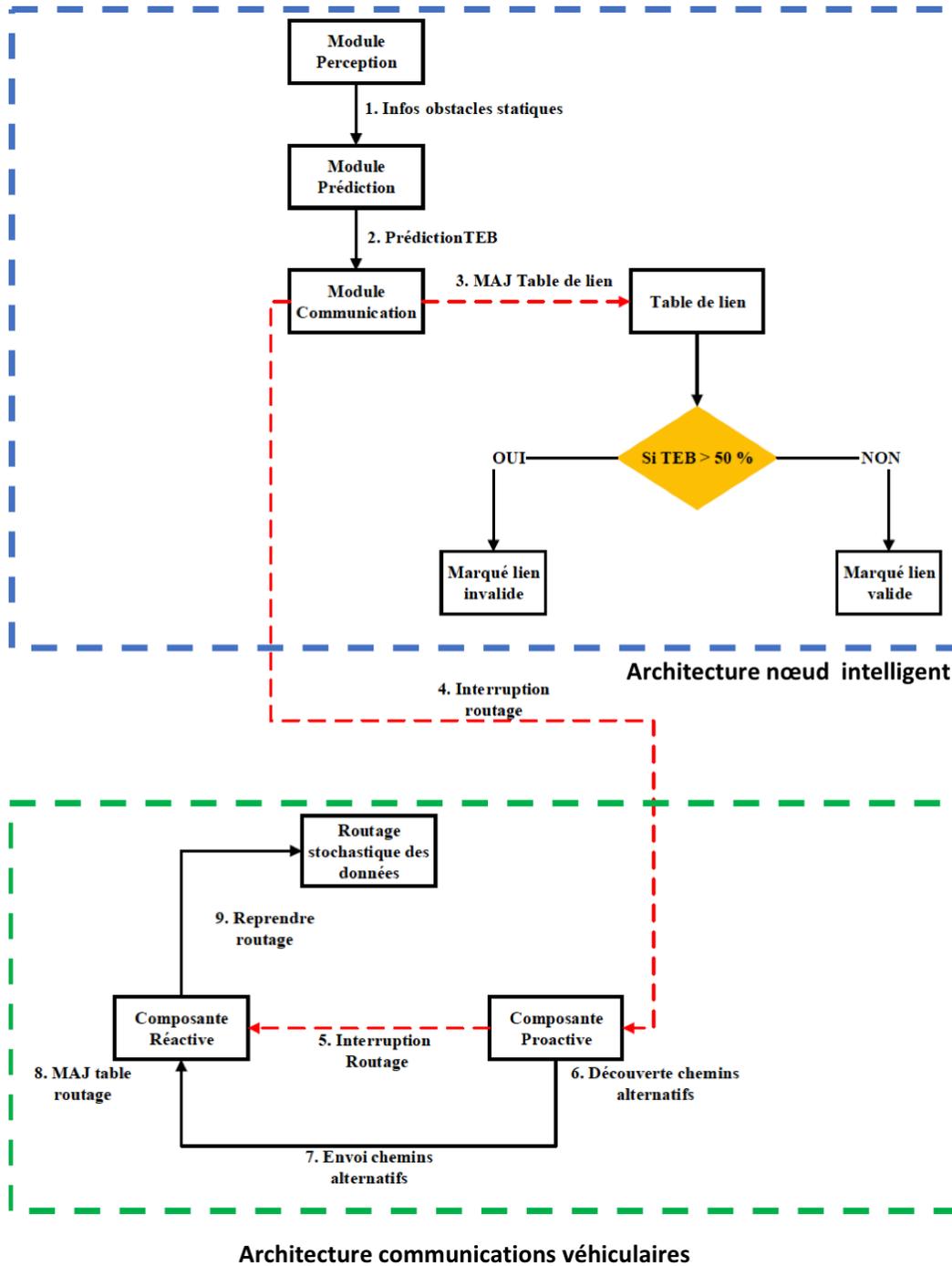


Figure 4.14 : Structure dynamique de l'interaction des modules.

4.4.3.1 Fonctions de la structure dynamique des modules

Sur la figure 4.14 précédente, la structure dynamique de notre système de routage cognitif comprend trois modules, à savoir : le module de perception, le module de prédiction et le module de communication. Dans cette section, nous allons décrire les différentes fonctions de l'architecture interne qui caractérisent ces modules.

A. Fonctions du module de perception

Ce module est caractérisé par une fonction qui consiste à identifier les obstacles présents au niveau de l'environnement physique des véhicules composant notre réseau VANET. L'identification qu'on appelle *localisation* à cause des informations associées aux obstacles statiques est basée sur la perception des composantes urbaines que nous considérons. Le processus de perception s'opère par le biais d'une carte numérique embarquée au niveau du véhicule autonome (voir figure 4.15) et exploite un algorithme d'identification d'obstacles (figure 4.17). Cet algorithme prend en entrée la liste d'obstacles statiques de l'environnement et retourne une liste d'obstacles à proximité du véhicule courant dépendamment de la position ainsi que de la portée de détection de ce dernier.

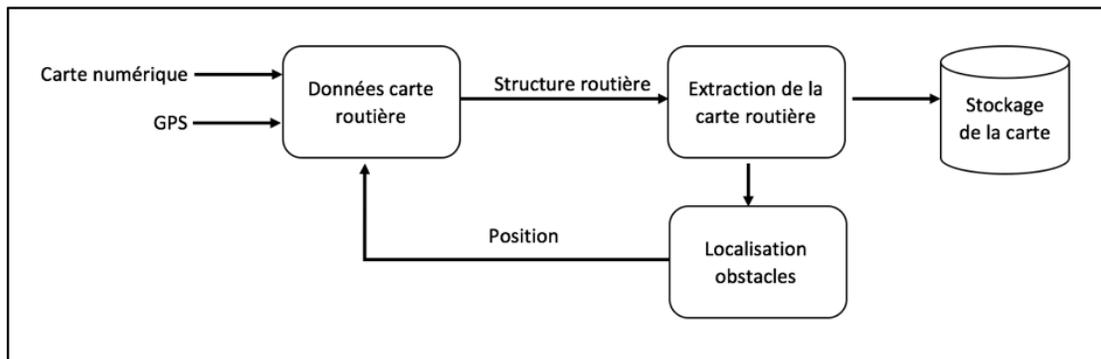


Figure 4.15 : processus d'identification des obstacles statiques.

L'algorithme d'identification d'obstacles statiques par un véhicule autonome s'exécute comme suit dans notre programme :

- le véhicule autonome récupère la liste des obstacles statiques de son environnement par le biais d'une carte numérique embarquée.

- pour chaque obstacle statique de la liste, le véhicule autonome détermine la distance lui séparant de ce dernier.
- Si la distance est inférieure à la portée de transmission, l'obstacle est stocké dans la liste de retour. Dans le cas échéant, l'obstacle est ignoré, car il se trouve en dehors de la portée de transmission du véhicule autonome.
- l'algorithme renvoi la liste des obstacles statiques à proximité du véhicule autonome.

L'organigramme de l'algorithme décrit précédemment est illustré par la figure **4.16** ci-dessous.

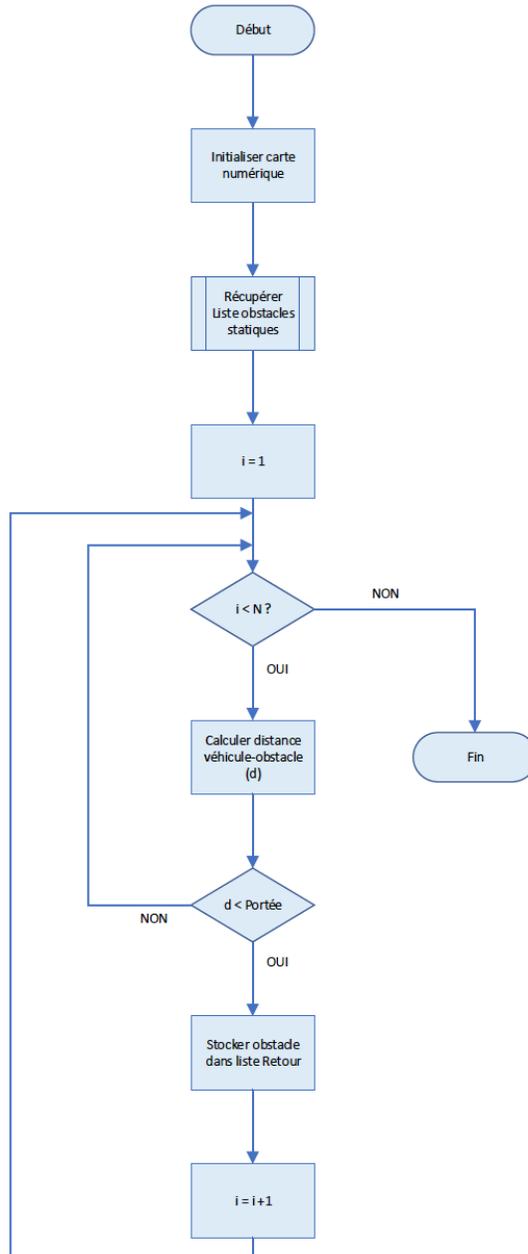


Figure 4.16 : Identification des obstacles statiques par un véhicule autonome.

Sur la figure 4.16 précédente, N représente le nombre d'obstacles statiques.

Algorithme 1: Algorithme identification des obstacles statiques

```
Entrées:
1  ListeObs : string; /* fichier obstacles statiques */
2  posVeh : real; /* Position du véhicule courant */
3  porteeDetection : real; /* Portée de détection d'obstacles statiques */
4
Sorties:
5  listeRetour : list; /* Liste d'obstacles */
6
Données:
7  distance : real;
8  obstacles : list; /* Liste d'obstacles */
9  nombreObs : integer; /* Nombre d'obstacles dans le fichier poly */
10 correspond : integer; /* indice courant de l'obstacle dans la liste */
11 i : integer; /* Compteur */
12 obstacle : Obstacle; /* Instance d'un obstacle */
13
14 début
15   obstacles ← initialiseObstacle(ListeObs);
16   nombreObs ← obstacles.size();
17   porteeDetection ← 50;
18   pour i=1 à nombreObs faire
19     obstacle ← obstacles[i];
20     correspond ← -1;
21     distance ← obstacle.DistanceContourObstacle(posVeh);
22     si (distance < porteeDetection) alors
23       correspond ← i;
24     /* Si indice valide */
25     si (correspond > -1) alors
26       obstacle ← obstacles[correspond]; /* récupérer obstacle indice */
27       listeRetour.ajouter(obstacle); /* Ajouter obstacle à liste de retour */
28     i + 1;
29   return listeRetour; /* retourner liste obstacle identifiés */
30
```

Figure 4.17 : Algorithme d'identification d'obstacles.

L'algorithme de la figure 4.17 précédente est exploité par l'algorithme du module de perception. L'algorithme du module de perception est illustré par la figure 4.18 ci-dessous.

Algorithme 2: Algorithme traitement de la perception

```
Entrées:
1  ObsList : string; /* fichier obstacles statiques */
2  currentNodePos : real; /* position du noeud courant */
3  currentNodeID : integer; /* ID du noeud courant */
4  detectionRange : real; /* Portée de détection d'obstacles statiques */
5
Sorties:
6  obstruction : map; /* Liste obstacles 2D entre noeud courant et voisin */
7
Données:
8  OneHopNeighSet : list; /* ensemble des voisins à un saut */
9  obsSize : integer; /* nombre d'obstacles */
10 neighSize : integer; /* nombre des voisins */
11 oneHopNeigh : node; /* voisin à un saut */
12 neighPos : real; /* position courante du voisin à un saut */
13 neighNodeID : integer; /* ID d'un voisin direct */
14 allBox : map; /* liste structure géométrique obstacle en 2D */
15 box : Box2d; /* structure géométrique obstacle en 2D */
16
17 début
18   detectionRange ← 50;
19   /* Identification obstacles statiques */
20   obstacles ← StaticObsIdent (ObsList, currentPos, detectionRange);
21   obsSize ← obstacles.size(); /* récupérer nombre d'obstacles */
22   neighSize ← NeighborSet.size(); /* récupérer nombre des voisins */
23   pour i=1 à neighSize faire
24     oneHopNeigh ← OneHopNeighSet[i]; /* récupère voisin à un saut */
25     neighPos ← OneHopNeighSet.Pos(); /* récupère position voisin à un saut */
26     /*
27     neighNodeID ← OneHopNeighSet.NodeId(); /* ID du noeud voisin courant */
28     */
29     pour j=1 à obsSize faire
30       string obstacleId ← obstacles[j].getId();
31       box ← obstacles[j].MinAreaBoundingBox(); /* récupère obstacle 2D */
32       allBox[obstacleId] ← box; /* associe obstacle 2D à ID */
33     j ← j+1;
34     pour k=1 à allBox.size() faire
35       Box2d box2d ← allBox[k].second; /* Récupère obstacle 2D de MAP */
36       /* tester si cet obstacle fait obstruction entre noeud courant et voisin */
37       /*
38       si (line.Rect(currentNodePos, neighPos, box2d)) alors
39         Map <String, Box2d> Obs;
40         /* association identifiant obstacle et structure géométrique 2D */
41         Obs [allBox[k].first] = box2d;
42         obstruction[currentNodeID, neighNodeID] = Obs; /* associe ID
43         noeud voisin et noeud courant à liste d'obstruction. */
44       k ← k+1;
45     i ← i+1;
46   return obstruction;
```

Figure 4.18 : Algorithme traitement de la perception.

L'algorithme de la figure 4.18 précédente est utilisé par l'algorithme du module de prédiction.

B. Fonctions du module de prédiction

Le module de prédiction exploite l'apprentissage supervisé basé sur les réseaux de neurones pour évaluer le niveau de dégradation du signal par rapport aux obstacles à proximité de la portée de transmission du véhicule courant (par exemple, le nœud source) et ses voisins directs. Pour ce faire, nous définissons une fonction de prédiction y_j^k à partir d'un ensemble de données d'entraînement (D). D est composé de paires d'exemples (x, y) , où x est un vecteur d'entrées et y le vecteur cible correspondant. Plus spécifiquement, un vecteur d'entrée x est composé d'informations représentant la structure géométrique d'un obstacle sous forme $(x_{min}, y_{min}, x_{max}, y_{max})$, tandis que le vecteur cible y représente le taux d'erreurs binaires (i.e. le nombre de bits perdus) encouru (voir section 4.3.3.2 pour plus de détails). La fonction de prédiction du taux d'erreurs binaires est donnée par la formule ci-dessous :

$$y_j^k = f \left(\sum_{i=1}^n w_{ij}^k x_i^k + b_j^k \right)$$

Avec :

f la fonction d'activation.

w_{ij}^k le coefficient synaptique entre le $i^{\text{ème}}$ neurone de la couche précédente et le neurone considéré.

x_i^k est l'entrée du $i^{\text{ème}}$ neurone de la couche précédente.

b_j^k est le biais associé au neurone courant.

Cette fonction est intégrée au niveau du véhicule intelligent comme nouvelle fonctionnalité permettant de supporter le processus de routage cognitif que nous développons pour cette étude. L'algorithme du module de prédiction intégré au niveau du véhicule intelligent est illustré par la figure 4.19 ci-dessous.

Algorithme 3: Algorithme traitement de la prédiction

```
Entrées:
1  obstruction : map; /* association ID noeud courant, ID voisin, obstructions */
2
Sorties:
3  berList : List; /* Map ID noeud courant, ID noeud voisin, Obstacles, BER */
4
Données:
5  ber : real; /* BER prédite */
6
7 début
8   int somme ← 0;
9   pour i=1 à obstruction.size() faire
10    map <int, int> nodeID; /* map current Node and neighbor ID */
11    map <String, Box2d> ObsList;
12    vector <Box2d> ObsList2;
13    nodeID ← obstruction[i].first; /* association ID noeud courant, ID voisin */
14    ObsList ← obstruction[i].second; /* association ID obs, struct 2D */
15
16    pour j=1 à ObsList.size() faire
17     Box2d box2d;
18     box2d ← j -> second;
19     int x1 ← box2d.min_x();
20     int y1 ← box2d.min_y();
21     int x2 ← box2d.max_x();
22     int y2 ← box2d.max_y();
23     /* Prediction BER obstacle */
24     ber ← predicted_ber(x1, y1, x2, y2);
25     ObsList2.push_back(box2d);
26     somme ← somme + ber;
27     meanBer ← somme/ObsList.size();
28     berList[nodeID] ← make_pair(ObsList2, meanBer);
29     j + 1;
30   i + 1;
31 return berList;
```

Figure 4.19 : Algorithme traitement de la prédiction.

C. Fonctions du module de communication

Le module de communication permet de mettre à jour la table de liens au niveau de l'architecture du nœud intelligent. Cette mise à jour est cruciale, car elle permet d'anticiper les problèmes d'instabilité causés par des liens défaillants lors de la construction d'un chemin de routage par la composante réactive de l'architecture de communication. Ainsi, lorsque le TEB entre le nœud source et l'un des nœuds voisins (directs) est supérieure à 50 %, un événement est transmis par le module de communication du nœud intelligent à l'architecture de communication véhiculaire pour interrompre le processus de routage au niveau de la composante réactive. Le contenu du message transmis par l'événement contient une liste de couples de liens invalides, la source et la destination. Plus précisément, si V_1 est la source et que TEB entre V_1 et V_2 supérieur à 50%, le couple (V_1, V_2) est considéré

comme lien invalide. En conséquence, la composante réactive se charge de mettre à jour la table de routage et construit les chemins entre la source V_1 et la destination D en éliminant les chemins associés au couple de lien (V_1, V_2) (confère figure 4.14).

4.4.4 Structure de la table de routage du protocole CRV

Dans cette section, nous allons décrire la structure de la table de routage du protocole CRV. Contrairement aux protocoles tels que GPSR et AntHocNet, la structure de la table de routage de CRV est composé de huit champs décrits ci-dessous. Plus de détails sur les tables de routage des protocoles reliés à notre solution font l'objet d'une section dans le chapitre 5.

- Le champ source indique l'adresse IP de la provenance du paquet.
- Le champ destination indique l'adresse IP du destinataire.
- Le champ *Prochain Saut* indique l'adresse IP de la prochaine passerelle du paquet.
- Le champ *Phéromone Régulière* est utilisé pour indiquer le chemin par lequel les données doivent transiter pour atteindre la destination d . En effet, il s'agit d'une valeur heuristique permettant d'estimer la qualité de la prochaine passerelle j le long de la route de i à la destination d . Selon les auteurs d'AntHocNet, elle représente la moyenne actuelle de l'inverse du coût en termes de durée estimée et le nombre de sauts de d à n (*voisin*). Elle est exprimée comme suit ([Di Caro et al., 2004](#)) : $T_{nd}^i = \gamma T_{nd}^i + (1 - \gamma)\tau_d^i, \gamma \in [0, 1]$. Avec τ_d^i représentant le paramètre de mise à jour en fonction du temps de parcours de la fourmi et du nombre de sauts.

τ_d^i est déterminé comme suit : $\tau_d^i = \left(\frac{\hat{T}_d^i + hT_{hop}}{2}\right)^{-1}$, Avec \hat{T}_d^i le temps de déplacement estimé par la fourmi, et h le nombre de sauts. T_{hop} est un paramètre (défini à 3×10^{-3} s) représentant le nombre de sauts dans des conditions déchargées.

- Le champ *Phéromone Virtuelle* permet d'indiquer un chemin alternatif. Il est créé ou mis à jour lors de la phase de diffusion des phéromones. Nous conservons ce champ dans la table de routage de CRV pour permettre d'étendre notre travail dans le futur.

- Le champ **Moyenne nombre sauts** indique le nombre de sauts moyen pour atteindre la destination. Dans la version originale d'AntHocNet, ce champ est calculé comme suit : $h_{in}^d = \alpha h_{in}^d + (1 - \alpha)h$, $\alpha \in [0, 1]$ avec h le nombre de sauts parcourus par BANT entre d et i , et α représente un paramètre régulant la rapidité avec laquelle la formule s'adapte aux nouvelles informations. Les auteurs recommandent de définir la valeur de α à **0.7**. Dans notre travail, nous conservons la même approche de calcul pour le protocole CRV.
- Le champ **Pteb** indique le taux d'erreur binaire prédit pour un lien donnée. Il s'agit d'un paramètre important de notre solution. Il permet de déterminer les chemins en anticipant les problèmes de stabilité liens. Lorsque le TEB du lien est supérieur à 50%, l'information est relayée à la composante chargée de construire les chemins entre la source et la destination. Ainsi, les chemins sont construits en supprimant les liens dont le TEB est supérieur à 50 %.
- Le champ **Path Tag** représente une étiquette permettant d'indiquer la validité du chemin de routage en fonction du TEB.

La structure de la table de routage du protocole CRV est illustré par le tableau 4.9 ci-dessous.

Tableau 4.9 : Structure table de routage CRV.

Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (BER)	Path Tag
Source ₁	Destination ₁	Prochain Saut ₁	ψ_1	λ_1	h_1	BER ₁	Tag ₁
Source ₂	Destination ₂	Prochain Saut ₂	ψ_2	λ_2	h_2	BER ₂	Tag ₂
...
Source _n	Destination _n	Prochain Saut _n	ψ_n	λ_n	h_n	BER _n	Tag _n

La principale distinction entre la table de routage du protocole CRV et celle d'AntHocNet est l'utilisation du TEB en vue d'anticiper les problèmes d'instabilité de chemins au niveau du routage.

4.4.5 Principe de fonctionnement généralisé du protocole CRV

Dans cette section, nous allons présenter le principe de fonctionnement du protocole CRV. Pour ce faire, nous décrivons en détails les phases proactives et réactives du protocole CRV, puis nous illustrerons son principe de fonctionnement.

4.4.5.1 Phase réactive

La phase réactive du protocole CRV permet de rechercher une route entre la source et la destination par le biais des fourmis sens avant ou Forward Ant (FANT) et sens arrière ou Backward Ant (BANT) dans le réseau VANET.

L'algorithme de la phase réactive est exécuté selon deux cas de figure, à savoir : initialiser la route pour la première fois ou mettre à jour la table de routage lorsque le TEB est supérieur à 50%. Les deux cas sont détaillés ci-dessous.

Algorithme de la phase réactive : Initialiser la route pour la première fois

Dans ce cas, il n'existe pas encore de route préétablie entre la source et la destination. Dans cette situation, la phase réactive du protocole CRV est identique à la version originale d'AntHocNet([Di Caro et al., 2004](#)). Cette phase est décrite par le paragraphe ci-dessous.

Rappel phase réactive protocole AntHocNet

Lorsqu'un nœud source s débute une session de communication avec un nœud de destination d et n'a pas d'informations de routage sur d , il diffuse une requête réactive $FANT F_d^s$. La tâche de chaque fourmi créée consiste à trouver un chemin qui relie s et d . Chaque nœud effectue une monodiffusion (**unicast**) ou une diffusion (**broadcast**) de la requête en fonction des informations de routage permettant d'accéder au nœud d . Les informations de routage d'un nœud sont représentées dans sa table de phéromone T^i . L'entrée $T_{nd}^i \in \mathbf{R}$ de la table des phéromones indique la valeur de la qualité du chemin estimé passant du voisin i à n pour atteindre la destination d . Si l'information sur la phéromone est connue, la fourmi choisit son prochain saut n avec une probabilité P_{nd} :

$$P_{nd} = \frac{(T_{nd}^i)^\beta}{\sum_{j \in N_d^i} (T_{jd}^i)^\beta}, \beta \geq 1 \quad (1),$$

Où N_d^i est l'ensemble des voisins i pour lesquels le chemin vers d est connu.

β est la valeur du paramètre qui permet de contrôler le comportement d'exploration des fourmis.

Si aucune phéromone n'est disponible pour d , la fourmi est diffusée. Compte tenu de cette diffusion, les fourmis peuvent proliférer rapidement sur le réseau, en suivant des chemins différents vers la destination.

Lorsqu'un nœud reçoit plusieurs fourmis de la même génération, il compare le chemin parcouru par chaque fourmi à celle des fourmis reçues précédemment de cette génération : si son nombre de sauts et le temps de voyage ont un facteur d'acceptation α_1 par rapport à celle de la meilleure fourmi de la génération précédente, il transmettra la fourmi courante. Grâce à cette politique, la surcharge du réseau est limitée en supprimant les fourmis qui empruntent le mauvais chemin. Cependant, la politique permettant de limiter la surcharge du réseau a pour effet de laisser passer d'abord la première fourmi arrivant au niveau du nœud. Les fourmis suivantes doivent donc répondre aux critères de sélection imposés par la meilleure fourmi qui les précède, et sont donc plus susceptibles d'être rejetées. Les fourmis dupliquées qui résultent de la diffusion de la meilleure fourmi avant qu'elles n'atteignent la destination ont des performances proches de la meilleure fourmi et ont plus de chance d'être acceptées ([Di Caro et al., 2004](#); [Ducatelle, 2007](#)).

Chaque **FANT** conserve une liste P des nœuds $[1 ; \dots ; n]$ qu'il a visitée. Lorsqu'elle atteint la destination d , elle se transforme en **BANT**, qui se dirige vers la source en suivant la liste P des nœuds visités (si cela n'est pas possible parce qu'aucun saut suivant n'existe en raison de la mobilité des nœuds, **BANT** est rejetée). **BANT** effectue une estimation incrémentielle du temps \hat{T}_p nécessaire pour qu'un paquet de données se déplace vers la destination via la liste P des nœuds visités par **FANT**. Ce temps est utilisé pour

mettre à jour les tables de routage. \hat{T}_p est la somme des estimations locales \hat{T}_{i+1}^i au niveau de chaque nœud $i \in P$ pour atteindre le prochain saut $i+1$:

$$\hat{T}_p = \sum_{i=1}^{n-1} \hat{T}_{i+1}^i \quad (2)$$

Avec :

\hat{T}_{i+1}^i définit comme étant le produit de l'estimation du temps moyen d'envoi d'un paquet et son équation s'exprime comme suit :

$$\hat{T}_{i+1}^i = (Q_{mac}^i + 1)\hat{T}_{mac}^i, \quad (3)$$

Où \hat{T}_{mac}^i représente le temps pour le nombre de paquets actuel dans la file d'attente (plus un) au niveau de la couche MAC Q_{mac}^i . Il est calculé comme étant la moyenne courante du temps écoulé entre l'arrivée d'un paquet au niveau de la couche MAC et la fin d'une transmission réussie. Si t_{mac}^i est le temps qu'il a fallu pour envoyer un paquet à partir du nœud, alors l'estimation du nœud i mis à jour est définie comme suit :

$$\hat{T}_{mac}^i = \alpha \hat{T}_{mac}^i + (1 - \alpha)t_{mac}^i, \quad (4)$$

Avec $\alpha \in [0, 1]$.

\hat{T}_{mac}^i est calculé à la couche MAC, il comprend des activités d'accès du canal de sorte qu'il représente la congestion locale du support partagé.

Au niveau de chaque nœud $i \in P$, **BANT** détermine un chemin vers la destination d , en créant ou en mettant à jour l'entrée T_{nd}^i dans T^i de la table de phéromone. La valeur de la phéromone T_{nd}^i représente la moyenne courante de l'inverse du coût en termes de temps estimé et du nombre de sauts à réaliser pour se rendre de d au voisin n . Si \hat{T}_d^i est le temps de déplacement estimé par la fourmi, et h le nombre de sauts, la valeur τ_d^i utilisé pour mettre à jour la moyenne courante sera définit comme suit:

$$\tau_d^i = \left(\frac{\hat{T}_d^i + hT_{hop}}{2} \right)^{-1}, \quad (5)$$

Où T_{hop} est un paramètre (défini à 3×10^{-3} s) représentant le nombre de sauts dans des conditions déchargées. Définir τ_d^i de cette façon permet d'éviter de grandes oscillations dans les estimations de temps réalisées par les fourmis (par exemple à cause de la densité du trafic) et permet de tenir compte du délai de bout-en-bout ainsi que du nombre de sauts(Ducatelle, 2007).

La valeur T_{nd}^i est mise à jour comme suit :

$$T_{nd}^i = \gamma T_{nd}^i + (1 - \gamma) \tau_d^i, \gamma \in [0, 1]. \quad (6)$$

Suite aux expérimentations menées lors des travaux présentés dans (Ducatelle, 2007), les auteurs recommandent de définir les valeurs de α et γ (équation 4) à 0.7.

Si le processus d'initialisation de chemin est un succès, un certain nombre de chemins appropriés entre la source et la destination sont mis à disposition. Le chemin peut se trouver à proximité d'un obstacle. Dans ce cas précis, l'obstacle n'a pas d'impact considérable sur la qualité de transmission, car l'on anticipe la défaillance des liens en prédisant la dégradation des signaux associés à ces derniers. Ainsi, si un lien représente un risque pour la transmission, il est supprimé du chemin de routage. Lorsqu'aucune fourmi de type **BANT** ne revient à la source après un certain laps de temps (pour les expériences prévues à 1s), les données sont temporairement stockées en mémoire tampon et l'ensemble du processus recommence. Ce processus est répété pour un nombre maximum fixé à trois, après quoi, les données stockées en mémoires tampon sont libérées.

La figure 4.20 ci-dessous illustre la phase réactive pour l'initialisation de la route. Le pseudocode présenté pour le protocole CRV est basé sur la version originale d'AnthocNet(Di Caro et al., 2004; Ducatelle, 2007).

Algorithme 4: Phase reactive pour l'initialisation de la route

```
1  $N_s$  crée FANT;  
2 Diffusion FANT;  
3 tant que ( $N_c \neq N_d$ ) faire  
4    $t_{arrive} \leftarrow tempsCourant()$ ;  
5   Ajouter  $t_{arrive}$  à la liste T;  
6   Ajouter  $N_c$  à la liste P;  
7   si ( $pheromoneExiste = true$ ) alors  
8     Sélection Prochain saut via probabilité :  $P_{nd} = \frac{T_{nd}^i}{\sum_{j \in N_{jd}^i} T_{jd}^i}$  ;  
9     Transmission information vers prochain saut;  
10  sinon  
11  | Rediffusion FANT (voisins directes avec un max de deux sauts);  
12  $N_d$  Crée BANT;  
13 Détruire FANT;  
14 tant que ( $N_c \neq N_s$ ) faire  
15 | MAJ table des phéromones via l'équation  $\gamma T_{nd}^i + (1 - \gamma)\tau_d^i$  ;  
16 | Envoyer BANT sur le chemin inverse de P;  
17  $Reception\_reponse(N_s, N_d)$ ;  
18 Détruire BANT;
```

Figure 4.20 : Phase réactive pour l'initialisation de la route du protocole CRV.

Avec :

N_s le nœud source.

N_c le nœud courant.

N_d le nœud de destination.

T liste de temps de parcours des nœuds.

P liste des nœuds visités.

Variante du comportement de l'algorithme pour la phase réactive : MAJ table de routage lorsque TEB supérieur à 50 %

Pour ce cas de figure, l'exécution de l'algorithme est interrompue par un événement en provenance de la composante proactive. Ainsi, la table de routage est mise à jour en fonction des informations issues de liens valides. Tel mentionné dans le chapitre 3, nous retenons les bases de notre travail précédent (variante d'AntHocNet) en vue d'intégrer le comportement cognitif à la phase réactive du protocole CRV. La figure 4.21 illustre le pseudocode de cet algorithme.

Algorithme 5: Réaction lors de la réception d'un événement

```

1 Réception événement Interruption routage; /* composante proactive */
2 Réception Envoi chemins alternatifs; /* composante réactive */
3 Consultation des liens invalides; /* composante réactive */
4 MAJ Table Routage : /* composante réactive */
5 si (TEB > 50%) alors
6   | Marqué chemin associé au lien invalide comme obsolète;
7 sinon
8   | Marqué chemin valide;
9 Enclencher requête Reprendre routage; /* composante réactive */
10

```

Figure 4.21: Réaction lors de la réception d'un événement

4.4.5.2 Phase proactive

Contrairement à CL-AntHocNet, la phase proactive du protocole CRV permet la maintenance et l'exploration des routes alternatives vers la destination en anticipant les risques de défaillance de liens. Pour ce faire, lorsque le TEB prédit au niveau du nœud intelligent est supérieur au seuil de 50 %, ce dernier met à jour la table de liens locale, puis envoie un événement d'interruption de routage vers la composante proactive de l'architecture de communication et qui est structurée selon un processus concurrent. L'événement est relayé par la composante proactive vers la composante réactive via la requête *Interruption Routage*. Ainsi, la composante proactive déclenche le processus découverte de chemins alternatifs sur la base d'informations du TEB qu'elle a reçues du module de communication de l'architecture du nœud intelligent. Les informations inhérentes au TEB indiquent la *source*, la *destination* et la *liste des liens directs* du nœud courant. Chaque lien est associé à une table (voir tableau 4.10) indiquant le TEB ainsi que la validité de ce dernier. Après avoir déterminé les routes alternatives à travers le processus *Découverte chemins alternatifs*, la composante proactive transmet les informations (la liste des liens directs du nœud courant, routes alternatives) à la composante réactive pour qu'elle puisse mettre à jour la table de routage au niveau de l'architecture de communication. Le nœud courant va identifier les liens étiquetés comme invalides à partir de liste, puis il procédera à la suppression des chemins de routage associés à ces liens au niveau de la table de routage. Dans ce qui suit, nous décrivons le processus *Découverte*

chemins alternatifs ainsi que les opérations de la phase proactive en se situant par rapport à la version originale d'AntHocNet.

Tableau 4.10 : Illustration du contenu de la table de liens.

<i>Nœud local</i>	<i>Id Voisin₁</i>		<i>Id Voisin₂</i>		...	<i>Id Voisin_k</i>	
	TEB lien ₁	Validité lien ₁	TEB lien ₂	Validité lien ₂		TEB lien _k	Validité lien _k

Rappel du processus de découverte de chemins alternatifs dans la version originale d'AntHocNet

Dans la version d'origine d'AntHocNet, les fourmis proactives sens avant (**Proactive Forward Ants, PFA**) sont disséminées périodiquement pour redécouvrir ou explorer des chemins alternatifs vers la destination *d*. Dans un premier temps, les PFA sont envoyées en mode monodiffusion et elles choisissent leur prochain saut suivant une probabilité P_{nd} , déterminée comme suit :

$$P_{nd} = \frac{T_{nd}^i}{\sum_{j \in N_{jd}^i} T_{jd}^i}$$

Cependant, si la probabilité est faible au niveau de chaque nœud pour la diffusion de PFA, l'on procède comme suit :

Lorsque le nœud transmet PFA vers la destination sans avoir eu recours à la diffusion :

- Un chemin vers la destination existe.
- Les fourmis sens arrières proactives (**Proactive Backward Ants, PBA**) mettent à jour les valeurs de la phéromone.

Si le nœud a recours à une diffusion pour transmettre PFA vers la destination, il s'agit de l'exploration d'un nouveau chemin. PFA est reçue par tous les voisins directs (situé à deux sauts) du nœud courant ayant effectué la diffusion. S'il n'existe pas de phéromone pour la destination, une nouvelle diffusion est effectuée ([Di Caro et al., 2004](#); [Ducatelle, 2007](#)).

Les messages *hello* sont diffusés périodiquement pour améliorer le chemin de routage vers la destination.

Lorsqu'un nœud diffuse des *messages hello* vers un voisin et s'il ne reçoit pas de réponse, le voisin concerné est supprimé de sa table de routage. Dans le cas échéant, tout nœud qui reçoit un *message hello* d'un nouveau nœud *n* ajoute ce dernier comme destination dans sa table de routage. D'autres nœuds peuvent avoir connaissance de leurs voisins immédiats et obtenir des informations de la phéromone sur leur table de routage.

Défaillance de lien

Chaque nœud doit détecter la perte de lien avec ses voisins. Après un certain laps de temps, s'il n'y a pas de réponse au *message hello*, le nœud concerné est supprimé. Si le voisin du nœud *n* disparaît alors :

- Il est retiré de la liste des nœuds et les entrées qui lui sont associées sont supprimées de sa table de routage.
- Une notification de défaillance de lien est diffusée.
- Une nouvelle estimation du meilleur chemin est diffusée.
- Les voisins mettent à jour leur table de phéromone.

Si ces voisins perdent leur meilleur chemin vers la destination, ils diffusent une notification jusqu'à ce que tous les nœuds concernés soient notifiés.

La figure 4.22 ci-dessous illustre le processus de découverte de chemins alternatifs dans la version originale d'AntHocNet([Di Caro et al., 2004](#); [Ducatelle, 2007](#)).

Algorithme 6: Découverte chemins alternatifs dans AntHocNet

```
1  $N_s$  crée PFA;
2 Disséminer PFA ;
3 tant que ( $N_c \neq N_d$ ) faire
4    $t_{arrive} \leftarrow tempsCourant()$ ;
5   Ajouter  $t_{arrive}$  à la liste T;
6   Ajouter  $N_c$  à la liste P;
7   si ( $pheromoneExiste = true$ ) alors
8     Sélection Prochain saut via probabilité :  $P_{nd} = \frac{T_{nd}^i}{\sum_{j \in N_{jd}^i} T_{jd}^i}$  ;
9     Transmission PFA vers prochain saut;
10  sinon
11    diffusion PFA (voisins directes avec un max de deux sauts);
12  $N_d$  Crée PBA;
13 Détruire PFA;
14 tant que ( $N_c \neq N_s$ ) faire
15   MAJ table des phéromones via l'équation  $\gamma T_{nd}^i + (1 - \gamma)\tau_d^i$  ;
16   Envoyer PBA sur le chemin inverse de P;
17 Reception_reponse ( $N_s, N_d$ );
18 Détruire PBA;
```

Figure 4.22: Processus découverte chemins alternatifs dans AntHocNet ([Di Caro et al., 2004](#)).

N_s le nœud source.

N_d le nœud de destination.

N_c le nœud courant.

PFA proactive Forward Ant

PBA proactive Backward Ant

P_{nd} Probabilité choix prochain saut entre n et la destination d .

T_{nd}^i entrée indiquant la valeur de la phéromone allant du voisin i à n pour atteindre la destination d .

N_d^i l'ensemble des voisins de i pour lequel un trajet vers d est connu.

T liste de temps de parcours des nœuds.

P liste des nœuds visités.

Processus de découverte de chemins alternatifs dans CRV

Le processus de découverte de chemins alternatifs dans CRV est similaire à celui d'AnthocNet à une exception près. En effet, la phase d'exploration se fait en fonction d'informations inhérentes aux liens valides obtenues auprès du module de communication du nœud intelligent. Plus précisément, lorsque le TEB est supérieur à 50 %, le module de communication du nœud intelligent communique avec la composante proactive de l'architecture de communication via un événement. Ainsi, la composante proactive interrompt le processus de routage au niveau de la composante réactive. De ce fait, les nœuds invalides sont retirés de la liste des nœuds et chaque entrée associée à ces derniers est supprimée de la table de routage. Une notification risque de défaillance de lien est diffusée en indiquant les liens invalides. Tous les nœuds voisins qui reçoivent cette information mettent à jour leur table de routage en supprimant les chemins associés aux liens invalides.

La figure suivante illustre le processus de découverte de chemins alternatifs dans CRV. Nous supposons que la composante proactive a reçu l'événement *Interruption Routage*.

Algorithme 7: Découverte chemins alternatifs dans CRV

```
1 Consulter liste de liens invalides;
2 Supprimer chemins associés aux liens invalides;
3  $N_s$  crée PFA; /* On indique la destination voulu ainsi que les liens
   invalides dans la requête */
4 Disséminer PFA ;
5 tant que ( $N_c \neq N_d$ ) faire
6    $t_{arrive} \leftarrow tempsCourant()$ ;
7   Ajouter  $t_{arrive}$  à la liste T;
8   Ajouter  $N_c$  à la liste P;
9   si ( $pheromoneExiste = true$ ) alors
10    Sélection Prochain saut via probabilité :  $P_{nd} = \frac{T_{nd}^i}{\sum_{j \in N_{jd}^i} T_{jd}^i}$  ;
11    Transmission PFA vers prochain saut;
12  sinon
13    diffusion PFA (voisins directes avec un max de deux sauts);
14  $N_d$  Crée PBA;
15 Détruire PFA;
16 tant que ( $N_c \neq N_s$ ) faire
17   MAJ table des phéromones via l'équation  $\gamma T_{nd}^i + (1 - \gamma)\tau_d^i$  ;
18   Envoyer PBA sur le chemin inverse de P;
19 Reception_reponse ( $N_s, N_d$ );
20 Détruire PBA;
```

Figure 4.23: Processus découverte chemins alternatifs dans CRV.

Sur la figure 4.23 ci-dessus, la partie en orange représente les modifications apportées au niveau du processus de découverte de chemins alternatifs dans CRV, tandis que le reste provient de la littérature ([Di Caro et al., 2004](#); [Ducatelle, 2007](#)).

Ces modifications sont importantes, car elles permettent d'anticiper les défaillances de liens en supprimant les chemins de routage associés à ces derniers. Néanmoins lorsque des obstacles sont présents entre le nœud courant et les voisins directs, cela peut affecter le chemin de routage en fonction d'informations (liens invalides) dont dispose la composante proactive. Si la composante proactive à connaissance des liens invalides, dans ce cas le lien sera supprimé de tout chemin de routage. Dans le cas échéant, l'échec de lien est détecté par la composante proactive et le processus de découverte de chemins alternatifs recommence jusqu'à ce que l'on obtention un chemin approprié.

Les étapes de la phase proactive du protocole CRV sont illustrées par la figure ci-dessous.

Algorithme 8: Phase proactive CRV

- 1 Reception événement *Interruption Routage*;
 - 2 Envoi événement *Interruption Routage* à la composante réactive;
 - 3 Recherche chemins alternatifs via *Découverte chemins alternatifs*;
 - 4 Transmission à la composante réactive via *Envoi chemins alternatifs*.
-

Figure 4.24: Algorithme phase proactive protocole CRV.

Sur l'algorithme précédent, l'événement *Interruption Routage* indique les informations telles que : la source, la destination et la liste des liens directs du nœud courant. Plus précisément le format de l'information transmise par le module de communication du nœud intelligent est spécifié comme suit : *<source, destination, liste de liens directs du nœud courant >*.

4.4.5.3 Illustration principe du fonctionnement concurrent de CRV

Dans cette section, nous présenterons le principe de fonctionnement concurrent du protocole CRV selon deux architectures : l'architecture du nœud intelligent (représentant un véhicule intelligent) et l'architecture de communication. Pour ce faire, nous utilisons un digramme de séquence avec une représentation de l'exécution des processus concurrents tel qu'indiqué par les auteurs des travaux présentés dans ([Khandai, Acharya & Mohapatra, 2011](#)). Pour des raisons de lisibilité, nous illustrerons le fonctionnement en fonction de deux diagrammes de séquence qui représenteront respectivement l'architecture du nœud intelligent et l'architecture de communication. La figure 4.25 ci-dessous illustre l'exécution des processus concurrents de l'architecture du nœud intelligent.

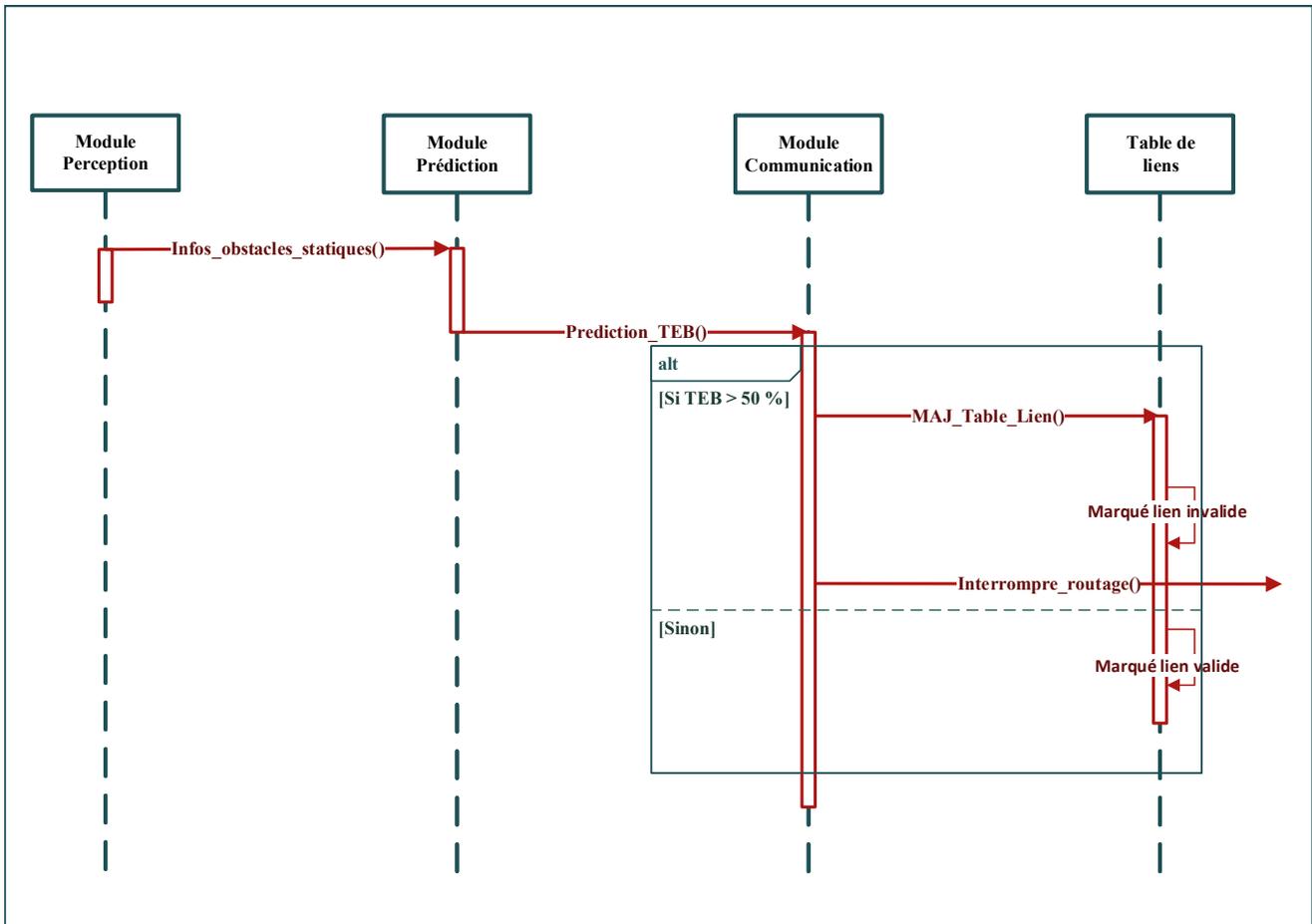


Figure 4.25 : Architecture concurrente du nœud intelligent.

Sur la figure précédente, les processus au niveau des modules de communication s'exécutent parallèlement aux actions de mises à jour au niveau de la table de liens. Dans ce cas précis, il s'agit d'une exécution parallèle conditionnelle (lorsque $TEB > 50\%$) qui déclenche un événement d'interruption de routage vers la composante proactive du module de communication.

Le diagramme de séquence suivant (figure 4.26) illustre le fonctionnement concurrent de l'architecture de communication lorsque ce dernier reçoit l'événement d'interruption de routage au niveau de la composante proactive.

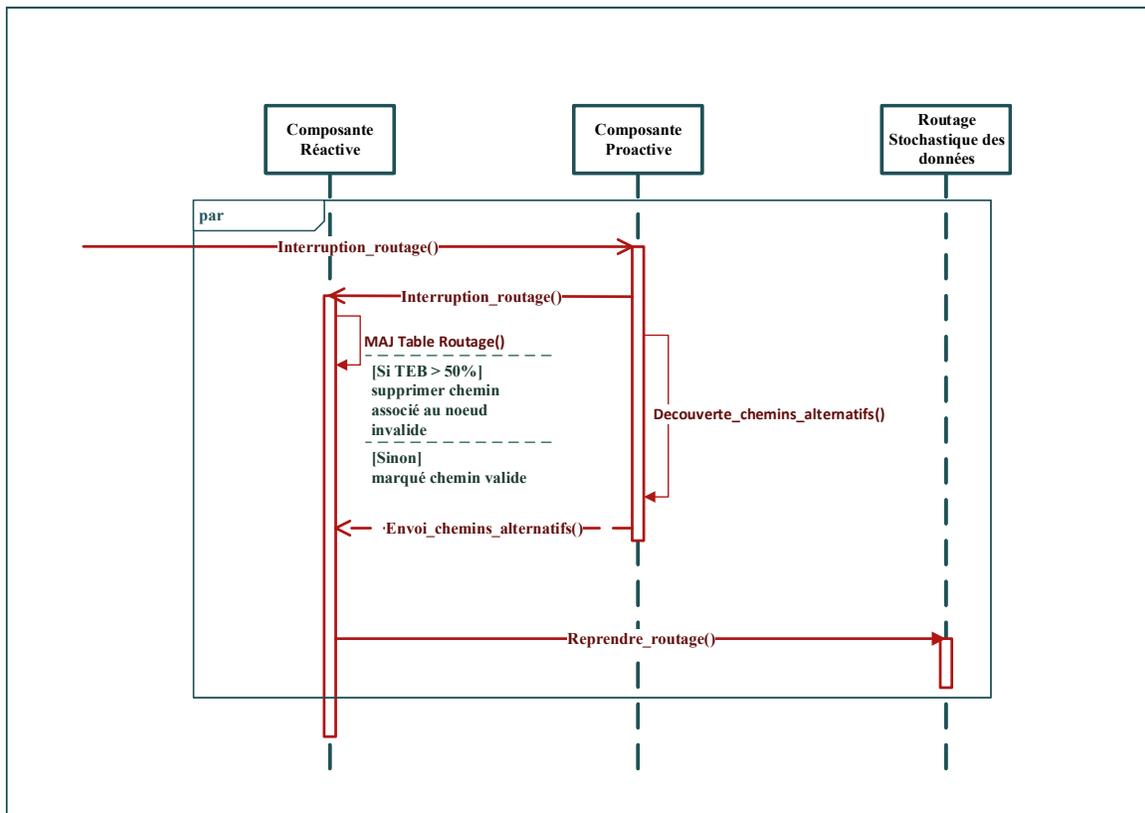


Figure 4.26 : Architecture concurrente des processus de communication.

La figure 4.26 ci-dessus illustre le principe de fonctionnement de l'architecture concurrente des processus de communication lorsque ce dernier reçoit un événement d'interruption de routage au niveau de la composante proactive. Ainsi, nous pouvons constater qu'en cas de réception d'événement d'interruption de routage, la composante proactive transmet l'événement ***Interruption_routage*** vers la composante réactive. Les informations reçues par la composante réactive comprennent : la source, la destination, la liste de liens directs du nœud courant. Parallèlement, la composante proactive enclenche le processus de recherche de chemins alternatifs via l'algorithme ***Decouverte_chemins_alternatifs***. Une fois les chemins alternatifs déterminés, la composante proactive transmet l'information (les chemins alternatifs) à la composante réactive par le biais de la requête ***Envoi_chemins_alternatifs***. Lorsque la requête ***Envoi_chemins_alternatifs*** est reçue par la composante réactive, cette dernière met à jour sa table de routage via ***MAJTableRoutage*** en supprimant le chemin associé aux nœuds

invalides (i.e. les liens), puis elle émet un signal **ReprendreRoutage** vers la composante routage stochastique pour l'indiquer de reprendre le processus de routage en fonction des nouvelles informations (**chemins alternatifs**).

Pour illustrer en détails le principe de fonctionnement de CRV, nous présenterons une étude de cas dans la section suivante.

A. Détails fonctionnement CRV

Cette section présente une étude de cas permettant d'illustrer le fonctionnement en détails du protocole CRV dans un réseau VANET lors d'une situation critique.

1. Initialisation

a. Présentation du réseau avec l'environnement.

Afin d'illustrer le fonctionnement en détails de CRV, nous nous appuyons sur le réseau présenté par la figure 4.27 ci-dessous. Sur cette figure, nous supposons qu'à l'instant t , un accident survient sur une voie. Les intersections adjacentes à la voie accidentée sont immédiatement bloquées par le véhicule de police V_1 . Ensuite, V_1 entame une diffusion de messages d'urgences pour aviser l'ambulance (V_8) afin qu'elle emprunte un détour et quitte la voie accidentée. Les véhicules V_2 , V_3 , V_4 , V_5 et V_6 jouent le rôle de nœuds intermédiaires lors de cette situation.

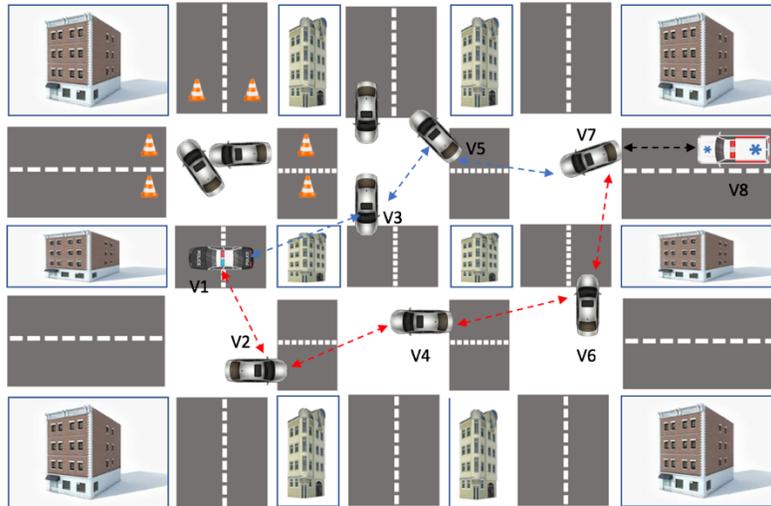


Figure 4.27 : Réseau VANET étudié

b. Architecture de communication

L'architecture de communication de CRV est régie par la nature hybride du processus de routage de ce dernier (voir section 4.4.3). En effet, CRV est basé sur le protocole CL-AntHocNet et utilise les informations de localisation des obstacles pour une prise de décision de routage.

c. Architecture d'un nœud représentant un véhicule autonome (intelligent)

L'architecture d'un nœud (voir fig. 4.14, partie en pointillé bleu) comprend principalement trois modules : **1)** le module d'identification des obstacles (i.e. la localisation, voir la section 4.4.3.1, point A), **2)** le module de prédiction des TEB (voir la section 4.4.3.1, point B) exploite l'apprentissage supervisé en spécifiant comme entrées la structure géométrique (X_{min} , Y_{min} , X_{max} et Y_{max}) d'un obstacle et prédit le TEB encouru en sortie, **3)** le module de communication (voir la section 4.4.3.1, point C) met à jour la table de liens (voir le tableau 4.10) et transmet l'événement d'interruption de routage vers la composante proactive de l'architecture des communications véhiculaires (voir fig. 4.14.).

Initialisation du contenu de la table de liens de V_1

Étant donné que le processus réactif fonctionne sur demande (requête de route), lorsqu'il n'y a pas de requête de route, la table de liens entre V_1 , et ses voisins directs (V_2 et V_3) est initialisée comme suit :

Tableau 4.11: Table de liens de V_1 à l'initialisation

V_1	V_2		V_3	
	TEB V_1V_2	Validité V_1V_2	TEB V_1V_3	Validité V_1V_3

Nous constatons sur la table précédente que les informations relatives aux TEB ainsi qu'à la validité des liens sont inconnues. Ceci s'explique par l'inexistence d'une requête de route. À ce stade, chaque nœud détecte uniquement ses voisins directs via des *messages hello*.

Initialisation de la table de routage V_1

En absence de requête de route, V_1 initialise sa table de routage ainsi que celle de ses voisins directs (V_2 et V_3) comme suit :

Tableau 4.12 : Table de routage de V_1 à l'initialisation

Table de routage de V_1							
IP Source	IP Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
-	-	V_2		-	1	-	-
-	-	V_3		-	1	-	-

Sur le tableau 4.12, en absence d'une requête de routage, le nœud V_1 détecte uniquement ses voisins directs (V_2 et V_3) via des messages hello. Ces voisins sont indiqués comme des passerelles potentielles pour V_1 (champ *Prochain Saut*) dans la table de routage.

2. Lancement du processus proactif du protocole de routage

Pour cette phase, nous supposons que les modules du nœud intelligent ainsi que le processus réactif du protocole de routage ne sont pas actifs. Le processus est déclenché tous

les Δt via l'algorithme décrit dans la section 4.4.5.2. Nous illustrons les tables de routage de V_1 , V_3 et V_5 lors de cette phase.

Tableau 4.13 : Table de routage de V_1 lors du processus proactif avant interruption

Table de routage de V_1							
IP Source	IP Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
-	-	V_2	1	-	1	-	-
-	-	V_3	1	-	1	-	-

Tableau 4.14 : Table de routage de V_3 lors du processus proactif avant interruption

Table de routage de V_3							
IP Source	IP Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
-	-	V_1	1	-	1	-	-
-	-	V_5	1	-	1	-	-

Tableau 4.15 : Table de routage de V_5 lors du processus proactif avant interruption

Table de routage de V_5							
IP Source	IP Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
-	-	V_3	1	-	1	-	-
-	-	V_7	1	-	1	-	-

Sur les tables de routage précédentes, les voisins directs des nœuds V_1 , V_3 et V_5 sont détectés via les messages hello. En outre, l'algorithme détermine la valeur des phéromones uniquement entre les voisins directs, puisqu'aucun nœud n'a initié une requête de routage.

3. Occurrence d'un événement et requête de transmission de messages

En s'appuyant sur le réseau présenté à la figure 4.27, nous allons illustrer la transmission des messages entre V_1 et V_8 .

Dans cette situation, de nouveaux traitements sont déclenchés parallèlement au processus proactif du routage. Plus précisément, les modules de localisation et de prédiction du nœud sont exécutés en parallèle au processus réactif qui est déclenché comme réaction à la requête de transmission. Les algorithmes associés (fig. 4.21 & 4.22) sont exécutés. Les résultats d'exécution de ces algorithmes se reflètent sur la table des liens et celle de routage des nœuds V_1 , V_3 , V_7 et V_8 .

a) Illustration processus réactif

Dans cette section, nous illustrons l'impact du processus réactif sur les tables de routage de V_1 , V_3 , V_7 et V_8 .

Tableau 4.16 : Table de routage de V_1 après exécution des modules de prédiction et localisation

Table de routage de V_1							
Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_1	V_3	V_3	0.179	-	1	65	invalide
V_1	V_5	V_3	0.209	-	2	-	invalide
V_1	V_7	V_3	0.262	-	3	-	invalide
V_1	V_8	V_3	0.35	-	4	-	invalide
V_1	V_2	V_2	0.057	-	1	5	valide
V_1	V_4	V_2	0.067	-	2	-	valide
V_1	V_6	V_2	0.137	-	3	-	valide
V_1	V_7	V_2	0.266	-	4	-	valide

V_1	V_8	V_2	0.473	-	5	-	valide
-------	-------	-------	-------	---	---	---	--------

Tableau 4.17 : Table de routage de V_3 après exécution des modules de prédiction et localisation

Table de routage de V_3

Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_3	V_1	V_1	0.229	-	1	65	invalide
V_3	V_5	V_5	0.223	-	1	15	valide
V_3	V_7	V_5	0.272	-	2	-	valide
V_3	V_8	V_5	0.276	-	3	-	valide

Tableau 4.18 : Table de routage de V_7 après exécution des modules de prédiction et localisation

Table de routage de V_7

Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_7	V_5	V_5	0.19	-	1	45	valide
V_7	V_3	V_5	0.132	-	2	-	valide
V_7	V_1	V_5	0.326	-	3	-	invalide
V_7	V_8	V_8	0.352	-	1	35	valide

Tableau 4.19 : Table de routage de V_8 après exécution des modules de prédiction et localisation

Table de routage de V_8							
Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_8	V_1	V_7	0.25	-	4	-	invalide
V_8	V_3	V_7	0.271	-	3	-	valide
V_8	V_5	V_7	0.259	-	2	-	valide
V_8	V_7	V_7	0.22	-	1	35	valide

Les tableaux précédents illustrent le contenu des tables de routage associés aux nœuds V_1 , V_3 , V_7 et V_8 suite aux traitements effectués par les modules de prédiction et localisation. Ainsi, la table de routage de V_1 indique un taux de TEB supérieur à 50 % pour le lien V_1V_3 , ce qui implique un risque d'échec pour la transmission des messages. Par conséquent, le lien V_1V_3 est marqué comme invalide dans la table de liens (voir tableau 4.20) de V_1 et tout chemin de routage (par exemple, le chemin représenté par $V_1 - V_3 - V_5 - V_7 - V_8$) associé à ce dernier sera supprimé de la table de routage du nœud V_1 .

Quelques exemples d'instanciation de valeurs des tables de liens sont fournis à la section suivante.

b) Instanciation de valeurs au niveau des tables de liens

Dans cette section, nous présentons des exemples d'instanciation de valeurs des tables de liens pour les nœuds V_1 , V_3 , V_5 et V_7 .

Tableau 4.20 : Table de liens V_1

V_1	V_2		V_3	
	TEB V_1V_2	Validité V_1V_2	TEB V_1V_3	Validité V_1V_3
	5%	valide	65%	invalide

Tableau 4.21 : Table de liens V_3

V_3	V_1		V_5	
	TEB	Validité	TEB	Validité
	V_3V_1	V_3V_1	V_3V_5	V_3V_5
	65%	invalide	15%	valide

Tableau 4.22 : Table de liens V_5

V_5	V_3		V_7	
	TEB	Validité	TEB	Validité
	V_5V_3	V_5V_3	V_5V_7	V_5V_7
	15%	valide	45%	valide

Tableau 4.23 : Table de liens V_7

V_7	V_5		V_8	
	TEB	Validité	TEB	Validité
	V_7V_5	V_7V_5	V_7V_8	V_7V_8
	45%	valide	35%	valide

Tableau 4.24 : Table de liens V_8

V_8	V_7	
	TEB	Validité
	V_8V_7	V_8V_7
	35%	valide

En tenant compte des détails explicatifs précédents, nous soulignons notre apport scientifique dans le domaine des réseaux VANET suivants deux axes :

- Nouveauté au niveau de la mise à jour de la table de routage : contrairement aux protocoles classiques tels que GPSR et AntHocNet, le protocole CRV met à jour sa table de routage en fonction du TEB prédit. Notamment, les chemins associés aux liens (voir tableau 4.20 à 4.24) invalides sont marqués comme obsolètes (voir tableau 4.16 à 4.19). Ce mécanisme procure l'avantage d'anticiper les chemins à risques (chemins dont le lien est invalide, c.-à-d. le TEB du lien est supérieur à 50 %) en les éliminant de la table de routage en amont d'une transmission.

- Nouveauté dans le contexte du routage et les véhicules autonomes : CRV se diffère des protocoles classiques de la littérature par son approche de routage innovante basée sur les nouvelles capacités (perception, localisation des obstacles statiques et estimation du TEB entre le nœud courant et ses voisins directs) d'un véhicule autonome. En effet, le protocole CRV se base sur l'estimation du TEB en amont d'une transmission en utilisant un algorithme d'identification d'obstacles statiques en considérant une carte numérique embarquée au niveau du véhicule autonome (nœud du réseau). Grâce à cette stratégie, CRV assure une meilleure fiabilité en termes de chemin de routage par rapport aux protocoles classiques.

L'organigramme présenté par la figure 4.28 ci-dessous illustre le fonctionnement d'un véhicule autonome pour différents cas (i.e. lorsque $TEB < 50\%$ et lorsque $TEB > 50\%$), tout en soulignant la nouveauté mentionnée précédemment pour la mise à jour (MAJ) de la table de routage dans CRV.

Hypothèse : Nous nous appuyons sur la figure 4.27 et supposons qu'à l'instant t_0 le véhicule V_1 (source) veut initier une transmission vers V_8 (destination).

Les étapes sur la figure 4.28 illustrant le fonctionnement du véhicule autonome se résument comme suit :

- Initialisation de la carte numérique : cette étape permet de charger la carte numérique de l'environnement ;
- Identification (localisation) des obstacles statiques : cette étape permet de localiser les obstacles statiques via un algorithme (figure 4.17 & 4.18) en s'appuyant sur la carte numérique de l'environnement ;
- Prédiction du taux d'erreurs binaires : cette étape permet d'estimer le TEB (voir fig. 4.19) entre le véhicule autonome (nœud courant) et ses voisins directs en fonction de la structure géométrique des obstacles.
- Mise à jour de la table de liens : cette étape permet au véhicule autonome de mettre à jour sa table de liens en fonction de la valeur du TEB. Sur la figure 4.25

précédente, nous illustrons l'exemple de mise à jour de la table de liens. Plus précisément, lorsque le TEB entre le véhicule autonome (nœud courant) et l'un des voisins directs est supérieur à 50%, le lien est indiqué comme invalide dans la table de liens (voir tableau 4.20 à 4.24), dans le cas échéant, le lien est marqué comme valide. Ainsi, la mise à jour de la table de routage selon les deux cas précédents est illustrée par le tableau 4.16 présenté précédemment.

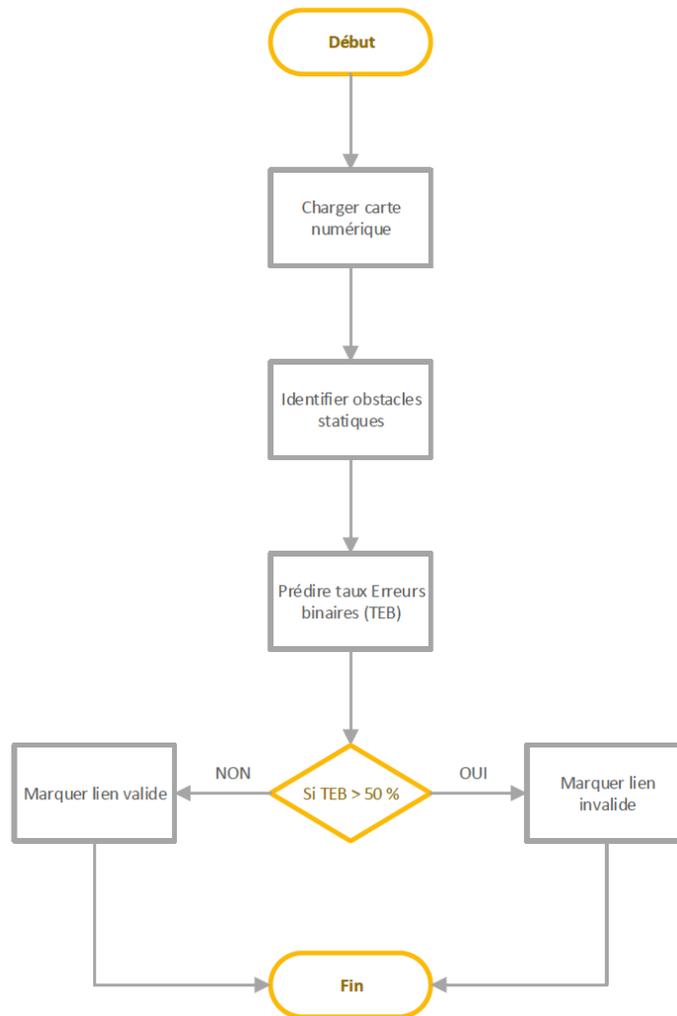


Figure 4.28 : Processus fonctionnement d'un véhicule autonome

Nous constatons sur le tableau 4.16 précédent que le véhicule source (V_1) a identifié deux chemins ($[V_1 - V_3 - V_5 - V_7 - V_8]$ et $[V_1 - V_2 - V_4 - V_6 - V_8]$) pour accéder à la destination. Toutefois, le TEB prédit sur le lien V_1V_3 est supérieur à 50 %, par conséquent, le chemin ($[V_1 - V_3 - V_5 - V_7 - V_8]$) associé à ce dernier est marqué comme invalide et sera supprimé de la table de routage du véhicule autonome (V_1). Par contre, comme V_1 dispose d'un chemin alternatif ($[V_1 - V_2 - V_4 - V_6 - V_8]$) valide (i.e. TEB inférieur à 50%), les données seront acheminées via celui-ci.

4.5 Conclusion

Dans ce chapitre, nous avons introduit le protocole CRV, un nouveau protocole de routage pour les réseaux VANET capable de considérer les capacités qui caractérisent un véhicule autonome (agent intelligent). En effet, CRV exploite l'optimisation par colonies de fourmis ainsi que l'identification des obstacles (localisation) en s'appuyant sur deux sous-systèmes, à savoir : l'architecture du nœud intelligent caractérisé par des capacités cognitives et l'architecture de communication du protocole. Ces deux architectures sont interdépendantes et interagissent entre elles par le biais d'événements. Les capacités cognitives du nœud intelligent permettent à CRV d'identifier la position des obstacles statiques et de prédire le TEB encouru par rapport à ces derniers sur le chemin de routage. Le mécanisme décrit précédemment permet d'éliminer les chemins potentiellement problématiques de la table de routage. Ce mécanisme est illustré sur les figures 4.25 et 4.26 précédentes. Nous pouvons constater sur ces figures que la solution proposée comprend deux architectures représentant les interactions entre le comportement d'un nœud (véhicule intelligent) et le fonctionnement du protocole de routage. Par exemple, lorsque les liens présentent un TEB supérieur à 50 %, ceux-ci sont éliminés avec les chemins qui les contiennent de la table de routage, ce qui permet d'anticiper l'instabilité des nœuds en améliorant le processus de transmission des données entre la source et la destination dans un réseau VANET.

Ainsi, dans ce chapitre, nous soulignons notre contribution dans le domaine des réseaux VANET en fonction de deux axes :

- 1) Contribution par rapport aux protocoles de routage pour les réseaux VANET : cette contribution se situe au niveau de la nouvelle structure de la table de routage et sa mise à jour qui se fait par anticipation aux dégradations potentielles aperçues par les nœuds mobiles du réseau.
- 2) Contribution par rapport aux véhicules autonomes et routage VANET : nous avons introduit une nouvelle architecture de communications de réseaux ad-hoc basée sur le principe du cross layer avec de nouvelles capacités d'intégration à l'architecture des nœuds représentant des agents intelligents.

Les deux architectures définissent un nouveau système de communication avec des interactions en temps réel faisant circuler des flux d'informations basées sur la prédiction du sous-système représentant des nœuds intelligents. Cette nouvelle intégration permet de faire la mise à jour de la table de routage selon la nouvelle structure présentant la première contribution du travail. (Voir les figure **4.25** et **4.26**).

Dans le chapitre suivant, nous procéderons à la validation de la performance du nouveau protocole CRV. Cette validation se fera au regard des protocoles CL-AntHocNet et GPSR en s'appuyant sur le protocole MAC STDMA ([Bilstrup, K. et al., 2009](#); [TR, 2012](#)). En effet, outre la validation de la performance du protocole CRV, la plus-value de ce chapitre consiste également à optimiser la performance du protocole CRV suivant un fonctionnement global dans un réseau VANET en exploitant un protocole MAC adapté pour les transmissions des données en temps réel.

Chapitre 5

Routage Cognitif pour les réseaux VANET :

Analyse de la performance et apports

Sommaire du chapitre 5

5.1	Introduction.....	132
5.2	Environnement de simulation	133
5.2.1	Choix de NS-3 comme simulateur réseaux.....	133
5.2.2	Choix de SUMO comme simulateur de mobilité urbaine.....	133
5.3	Étude du mécanisme de mise à jour de la table de routage de CRV	134
5.3.1	Scénario de simulation	134
5.3.2	Séquences de transmission.....	135
5.3.3	Discussion.....	140
5.4	Comparaison de CRV par rapport à GPSR et CL-AntHocNet.....	140
5.4.1	Scénarios de comparaison.....	141
5.4.2	Optimisation de la couche 2.....	143
5.4.3	Calibrage des paramètres de simulation NS-3	148
5.4.4	Paramètres de simulation NS-3.....	149
5.4.5	Métriques d'évaluation de performance	149
5.4.6	Résultats de la simulation	150
5.4.7	Analyse des résultats.....	152
5.5	Conclusion	157

5.1 Introduction

Dans le chapitre 4, nous avons passé en revue les solutions proposées en vue de mettre en œuvre le routage cognitif dans les réseaux VANET. Par conséquent, nous avons introduit une approche de routage basée sur des nouvelles capacités (perception de l'environnement, identification d'obstacles statiques et estimation du TEB) de véhicules autonomes pour faire face aux problèmes d'évanouissements par trajets multiples ainsi que celui du retard de transmission liés directement aux composantes urbaines pour les applications critiques des STI supportées par les réseaux VANET.

Le présent chapitre présente des tests et expérimentations des solutions présentées dans le chapitre 4 en considérant plus particulièrement le nouveau protocole de routage CRV. Les résultats numériques sont alors discutés avec des analyses de leurs performances. Nous soulignons par la suite les nouveaux apports. Pour ce faire, nous étudions la performance du protocole CRV par rapport à CL-AntHocNet et GPSR compte tenu de leurs contributions tel que présenté dans le chapitre 3. Divers scénarios issus de la problématique présentée dans le chapitre 2 sont considérés. Plus particulièrement, les effets de la latence de transmission pour les applications critiques des STI qui sont supportées par les réseaux VANET font l'objet de nos observations et analyses. Nos critères d'évaluations des performances du routage des réseaux VANET s'appuient sur deux métriques de performance, à savoir : le délai de bout-en-bout (DBB) et le taux de livraison paquets de données (TLPD).

Le reste du chapitre est structuré comme suit : la section 5.2 présente l'environnement de simulation. Une étude permettant de souligner l'efficacité du mécanisme de mise à jour de la table de routage de CRV est présentée à la section 5.3. La section 5.4 aborde l'analyse de performance du protocole CRV par rapport à GPSR et CL-AntHocNet. Enfin, nous clôturons ce chapitre par une conclusion qui fait l'objet de la section 5.5.

5.2 Environnement de simulation

De nombreux outils ont été développés pour permettre d'étudier les performances des réseaux, à l'instar d'OPNET ([Chang, 1999](#)), OMNET++ ([Varga & Hornig, 2008](#)), Network Simulator 2 (NS-2) ([Issariyakul & Hossain, 2009](#)), Network Simulator 3(NS-3) ([Henderson et al., 2008](#)), QualNet ([Siraj, Gupta & Badgujar, 2012](#)), etc. Notre choix s'est porté sur NS-3 comme outil de simulation réseau et SUMO comme outil de modélisation du trafic urbain.

Dans les sous-sections suivantes, nous ferons un rappel par rapport aux choix de NS-3 et SUMO comme outils de simulation pour nos travaux de recherche.

5.2.1 Choix de NS-3 comme simulateur réseaux

NS-3 (cf. chapitre 4) est un simulateur Open Source destiné à l'étude de performances des réseaux. Le choix de NS-3 comme simulateur réseau se justifie par sa popularité pour l'étude des réseaux VANET auprès de la communauté scientifique et par sa souplesse puisque son code source est facilement modifiable en fonction du besoin. En outre, NS-3 intègre la technologie DSRC ainsi que le protocole MAC 802.11p pour l'étude des réseaux VANET.

5.2.2 Choix de SUMO comme simulateur de mobilité urbaine

SUMO (cf. chapitre 4) est un outil de simulation de la mobilité urbaine. L'utilisation de SUMO comme simulateur de mobilité urbaine se justifie par la nécessité de disposer d'un modèle de mobilité réaliste pour les véhicules. Outre sa portabilité pour différentes plateformes (Windows, Linux, Mac OS), SUMO permet également d'interagir directement avec NS-3 par le biais de son API TraCi.

5.3 Étude du mécanisme de mise à jour de la table de routage de CRV

Dans cette section, nous analysons le mécanisme de mise à jour de la table de routage du protocole CRV via plusieurs séquences de transmission en s'appuyant sur les protocoles CL-AntHocNet et GPSR. Cette analyse permettra de souligner l'apport en termes de fonctionnalités au niveau du protocole CRV pour les applications critiques des STI qui sont supportées par les réseaux VANET. Le choix d'AntHocNet et GPSR comme protocoles de référence pour notre analyse est dû à la nature hybride (combinaison entre l'optimisation par colonies de fourmis et géolocalisation) du protocole CRV.

5.3.1 Scénario de simulation

Le scénario d'analyse s'apparente à celui de la figure 2.2 (cf. chapitre 2). Toutefois, nous formulons les hypothèses ci-dessous pour ce dernier :

- Le scénario se déroule dans un environnement urbain.
- L'événement déclencheur (accident, perturbations dues aux travaux, etc.) du scénario survient à l'instant t_0 .
- Les véhicules V_1 et V_8 représentent respectivement la source et la destination.
- Les véhicules $V_2, V_3, V_4, V_5, V_6,$ et V_7 représentent les nœuds intermédiaires.
- Le véhicule V_1 initie le message d'avertissement vers V_8 à l'instant t_1 .
- Chaque nœud du scénario est représenté par un véhicule autonome qui possède des capacités telles que : la perception, l'identification d'obstacles statiques et l'estimation du TEB.

Dans la sous-section suivante, nous présentons le mécanisme de mise à jour de la table de routage des différents protocoles présenté précédemment à travers diverses séquences de transmission. Le scénario étudié est illustré par la figure 4.27 (cf. chapitre 4).

5.3.2 Séquences de transmission

Dans cette section, nous présentons les séquences de communications inter-véhiculaires pour souligner l'efficacité du processus de mise à jour de la table de routage de CRV par rapport à GPSR et CL-AntHocNet.

5.3.2.1 Séquence de communication inter-véhiculaire via GPSR

Dans cette sous-section, nous présentons une séquence de transmission inter-véhiculaire permettant d'illustrer le mécanisme de mise à jour de la table de routage via GPSR. Cette séquence de transmission se déroule entre le nœud source (V_1) et le nœud de destination (V_8). Le figure 5.1 ci-dessous illustre le principe de fonctionnement de GPSR.

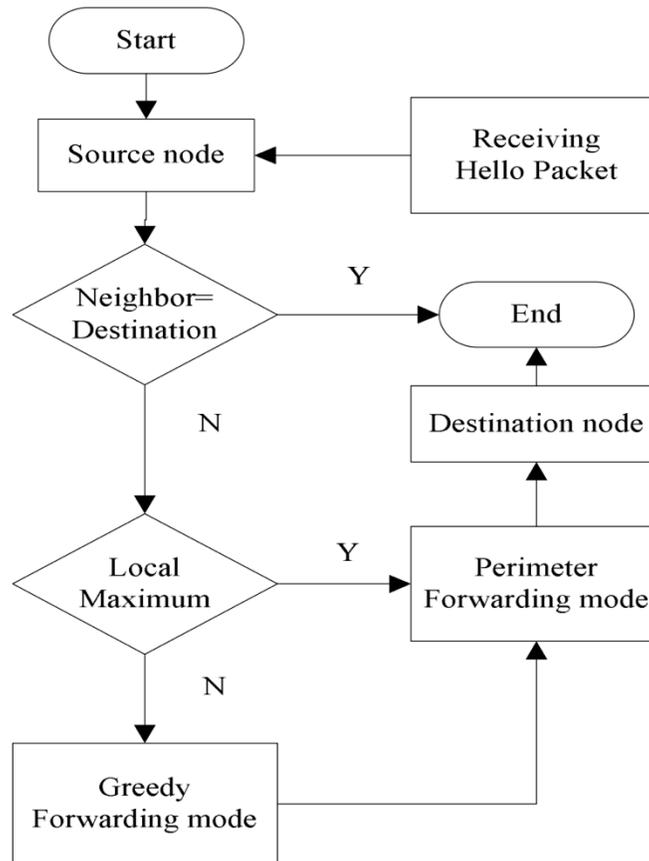


Figure 5.1 : Fonctionnement protocole GPSR (Hu, Ding & Shi, 2012)

Les tableaux 5.1 et 5.2 ci-dessous illustrent le mécanisme de mise à jour de la table de routage via GPSR entre V_1 et V_8 à différents instants.

Tableau 5.1 : Table de routage du nœud V_1 à Δt_1

V_1						
<i>IP Destination</i> (V_8)	<i>PosDestination</i>		<i>Nœud Voisin</i>			<i>DistanceSrcDest</i>
	<i>X</i>	<i>Y</i>	<i>IdNoeud</i>	<i>PosNoeud</i>		
				<i>X</i>	<i>Y</i>	
V_8	107.5	74.3	V_2	17.1	32.7	99,5
	-	-	V_3	30.2	58.6	78,8

Sur le tableau 5.1 précédent, V_1 veut transmettre des données vers V_8 . Étant donné qu'il dispose de plusieurs voisins intermédiaires lui permettant d'accéder au nœud de destination, V_1 utilise le mécanisme de routage **Greedy Forwarding**. Ce mécanisme lui permet de déterminer la prochaine passerelle la plus proche du nœud de destination V_8 en termes de distance euclidienne. V_1 identifie le nœud V_3 comme étant son voisin le plus proche de la destination et débute la transmission de l'information vers ce dernier.

Tableau 5.2 : Table de routage du nœud V_7 à Δt_4

V_7						
<i>IP Destination</i> (V_8)	<i>PosDestination</i>		<i>Nœud Voisin</i>			<i>DistanceSrcDest</i>
	<i>X</i>	<i>Y</i>	<i>IDNoeud</i>	<i>PosNoeud</i>		
				<i>X</i>	<i>Y</i>	
V_8	107.5	74.3	V_5	45.3	68.1	62,5
	-	-	V_6	60.1	48.3	54,06
	-	-	V_8	107.5	74.3	0

Sur le tableau 5.2 précédent, les paquets des données arrivent au niveau du nœud intermédiaire V_7 . Il consulte sa table de routage et constate qu'il dispose de plusieurs voisins (V_5 , V_6 , V_8) lui permettant d'atteindre la destination. V_7 utilise le mécanisme Greedy Forwarding en sélectionnant le nœud voisin V_8 comme prochaine passerelle et débute la transmission des paquets de données vers ce dernier.

5.3.2.2 Séquence de communication inter-véhiculaire via CL-AntHocNet

Cette sous-section présente le processus de mise à jour de la table de routage à divers instants lorsque CL-AntHocNet est utilisé comme protocole de routage entre le nœud source V_1 et le nœud destination V_8 .

Les tableaux 5.3 et 5.4 ci-dessous illustrent la mise à jour des tables de routage via CL-AntHocNet.

Tableau 5.3 : Table de routage du nœud V_1 à Δt_1

V_1					
<i>IP Source</i>	<i>Destination</i>	<i>Nœud voisin</i>			
		<i>Prochain Saut</i>	<i>Nbre Sauts</i>	<i>Table Phéromone</i>	
				<i>Régulière</i>	<i>Virtuelle</i>
V_1	V_3	V_3	1	0.75	-
	V_5	V_3	2	0.12	-
	V_7	V_3	3	0.08	-
	V_8	V_3	4	0.05	-

Sur le tableau 5.3 précédent, le véhicule V_1 , dissémine les fourmis sens avant (Forward Ants) pour la découverte de route. Lorsque celles-ci atteignent la destination V_8 , elles sont transformées en fourmis sens arrière (Backward Ants) et renvoyer vers la source (V_1). À l'issue du mécanisme d'établissement de la route entre V_1 et V_8 , le véhicule V_1 sélectionne la prochaine passerelle (V_3) en fonction de la valeur de la phéromone la plus élevée, puis les paquets des données sont transmises vers celle-ci. Lorsqu'un bris de lien survient dans le réseau, le cross-layering est exploité pour relayer l'information directement vers la couche réseau.

Tableau 5.4 : Table de routage du nœud V_7 à Δt_5

V_8					
<i>IP Source</i>	<i>Destination</i>	<i>Nœud voisin</i>			
		<i>Prochain Saut</i>	<i>Nbre Saut</i>	<i>Table Phéromone</i>	
				<i>Régulière</i>	<i>Virtuelle</i>
V_1	V_5	V_5	1	0.25	-
	V_3	V_5	2	0.12	-
	V_1	V_5	3	0.13	
	V_8	V_8	1	0.50	

Sur le tableau 5.4 précédent, les paquets de données arrivent au niveau du nœud intermédiaire V_7 . Il consulte sa table de routage et constate qu'il dispose de plusieurs voisins lui permettant d'atteindre la destination. En s'appuyant sur la valeur de phéromone (la plus élevée) et le nombre de sauts (le plus petit), V_7 sélectionne le nœud V_8 comme prochaine passerelle et débute la transmission des données vers ce dernier.

5.3.2.3 Séquence de communication inter-véhiculaire via CRV

Dans cette sous-section, nous présentons diverses séquences de communication inter-véhiculaire via CRV. Les séquences de transmission se déroulent entre le nœud source V_1 et le nœud de destination V_8 . Les tableaux 5.5 et 5.6 ci-dessous illustrent le mécanisme de mise à jour de la table de routage via CRV.

Tableau 5.5 : Table de routage du nœud V_1 à Δt_1

Table de routage de V_1							
Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_1	V_3	V_3	0.179	-	1	65	invalide
V_1	V_5	V_3	0.209	-	2	-	invalide
V_1	V_7	V_3	0.262	-	3	-	invalide

V_1	V_8	V_3	0.35	-	4	-	invalide
V_1	V_2	V_2	0.057	-	1	5	valide
V_1	V_4	V_2	0.067	-	2	-	valide
V_1	V_6	V_2	0.137	-	3	-	valide
V_1	V_7	V_2	0.266	-	4	-	valide
V_1	V_8	V_2	0.473	-	5	-	valide

Sur le tableau 5.5 précédent, le véhicule autonome V_1 perçoit son environnement (cf. chapitre 4) via une carte numérique embarquée et identifie les positions des obstacles statiques avoisinantes. Puis, il estime le taux d'erreurs binaires en fonction des obstacles perçus et initie le processus de requête de route vers V_8 par le biais des fourmis sens avant. À l'issue du processus réactif, V_1 établit deux chemins ($[V_1 - V_3 - V_5 - V_7 - V_8]$ et $[V_1 - V_2 - V_4 - V_6 - V_8]$) permettant d'accéder à V_8 . Cependant le lien V_1V_3 présente un TEB supérieur à 50 %, par conséquent le chemin de routage associé à ce lien sera supprimé de la table de routage de V_1 . Ainsi, le nœud V_1 sélectionnera le chemin alternatif formé par les nœuds ($[V_1 - V_2 - V_4 - V_6 - V_8]$) pour l'acheminement des données vers V_8 .

Ci-dessous une illustration de la table de routage au niveau du nœud V_7 .

Tableau 5.6 : Table de routage du nœud V_7 à Δt_5

Source	Destination	Prochain Saut	Valeur Régulière Ph (ψ)	Valeur Virtuelle Ph (λ)	Nombre Moyen de sauts (h)	Pteb (%)	Path Tag
V_7	V_5	V_5	0.19	-	1	45	valide
V_7	V_3	V_5	0.132	-	2	-	valide
V_7	V_1	V_5	0.326	-	3	-	invalide
V_7	V_8	V_8	0.352	-	1	35	valide

Sur le tableau 5.6 précédent, les fourmis sens avant arrivent au niveau du nœud V_7 . Il consulte sa table de routage et s'aperçoit qu'il dispose de plusieurs voisins directs (V_5 et V_8) permettant d'atteindre la destination. Puisque le nœud de destination V_8 est le voisin

direct de V_7 , ce dernier vérifie la validité du lien V_7V_8 ainsi que sa valeur de la phéromone. Si le lien V_7V_8 est valide et que sa valeur de la phéromone existe, alors V_7 transmet l'information directement vers V_8 . Dans le cas échéant, V_7 dissémine une requête *DecouverteCheminsAlternatifs* par le biais des fourmis proactives en vue de déterminer des routes alternatives.

5.3.3 Discussion

En analysant le processus de mise à jour des tables de routage ci-dessus, nous constatons que le protocole CRV s'appuie sur les nouvelles capacités (perception, identification, estimation du TEB) des véhicules autonomes pour mettre en œuvre le routage cognitif dans les VANET. En effet, le mécanisme de mise à jour de la table de routage de CRV est basé sur l'estimation du TEB, par conséquent, les chemins à risque sont supprimés de la table de routage de ce dernier. Plus précisément, lorsque le TEB est supérieur entre le nœud courant et un voisin direct, le lien est marqué comme invalide. Les chemins associés à ce lien sont donc supprimés de la table de routage. Ainsi, le protocole CRV offre une meilleure stabilité des routes par rapport à CL-AntHocNet et GPSR pour les applications critiques des STI qui sont supportées par les réseaux VANET.

5.4 Comparaison de CRV par rapport à GPSR et CL-AntHocNet

Dans cette section, nous allons réaliser un ensemble de simulation en vue d'évaluer les performances du protocole CRV par rapport à GPSR et CL-AntHocNet pour les applications critiques en termes de délai de bout-en-bout (DBB) et taux de livraison de paquets de données (TLPD). En outre, afin d'améliorer la performance du protocole CRV, nous effectuons une étude comparative entre les protocoles MAC STDMA([Bilstrup, K. et al., 2009](#)) et IEEE 802.11p. Ainsi, nous optimisons la couche 2 en intégrant le meilleur protocole MAC à notre solution.

Dans cette analyse, nous utilisons l'environnement de simulation décrit à la section 5.2 précédente. La section 5.4.1 introduit les scénarios de comparaison. Dans la section

5.4.2, nous abordons la question d'optimisation de la couche 2 pour améliorer la performance du protocole CRV. Le calibrage des paramètres de simulation est abordé à la section 5.4.3, tandis que les détails sur les paramètres de simulation retenus sont fournis à la section 5.4.4. Nous définissons les métriques d'évaluation dans la section 5.4.5. Ensuite, nous étudions les performances du protocole CRV et analysons les résultats obtenus dans la section 5.4.6. Finalement, nous terminons ce chapitre par une conclusion qui fait office de récapitulatif.

5.4.1 Scénarios de comparaison

Les scénarios élaborés pour évaluer la performance du protocole CRV dans le cadre des applications critiques des STI qui sont supportées par les réseaux VANET prennent en considération trois types d'infrastructures urbaines :

- La grille de Manhattan ;
- Le pont ;
- Le tunnel.

L'objectif du test consiste à évaluer la performance de CRV pour les applications critiques en fonction de différentes infrastructures urbaines. En effet, les structures géométriques d'obstacles varient dépendamment du type d'infrastructure. Ce qui implique un impact sur la qualité de transmission (problème d'évanouissement par trajets multiples évoqué au chapitre 2 de cette thèse).

Dans la suite de cette sous-section, nous présentons les scénarios de simulation considérée.

5.4.1.1 Scénario 1 : Impact des immeubles sur la qualité du signal dans une grille de Manhattan

Ce scénario se déroule dans une infrastructure urbaine de type de grille de Manhattan en présence d'obstacles multiples tels que les bâtiments. La grille de Manhattan représente une partie de la Ville de Gatineau à la périphérie du boulevard Montclair. La

nécessité d'analyser la performance de CRV pour ce scénario se justifie par le besoin de mesurer l'impact d'obstacles de type immeuble sur la qualité de la transmission pour les applications critiques des STI supportées par les réseaux VANET. Pour ce faire, nous considérons plusieurs densités (base, moyenne, haute) routières pour notre évaluation. Le tableau 5.7 ci-dessous présente les cas de figures élaborés en fonction de la densité.

Tableau 5.7 : Configuration des densités pour le scénario 1

Scénario 1	Densité	Nombre de véhicules	Vitesse maximale (Km/h)	Temps de Simulation
Grille de Manhattan	Basse	49	30	300
Grille de Manhattan	Moyenne	80	30	340
Grille de Manhattan	Haute	198	50	494

5.4.1.2 Scénario 2 : Impact d'un pont sur la qualité du signal

Ce scénario se déroule sur un pont. Les métaux constituant la structure géométrique du pont peuvent affecter la qualité de la transmission différemment des obstacles tels que les bâtiments. Par conséquent, nous analysons la performance de CRV pour les applications critiques des STI supportées par les réseaux VANET pour cette situation. Le tableau 5.8 ci-dessous présente les cas de figures élaborés en fonction des densités considérées.

Tableau 5.8 : Configuration des densités pour le scénario 2

Scénario 2	Densité	Nombre de véhicules	Vitesse maximale (Km/h)	Temps de Simulation
Pont	Basse	39	30	203
Pont	Moyenne	69	30	259
Pont	Haute	163	50	399

5.4.1.3 Scénario 3 : Impact d'un tunnel sur la qualité du signal

Ce scénario se déroule dans un tunnel. La constitution du tunnel peut affecter la transmission différemment des obstacles tels que le pont ou les bâtiments. Par conséquent, il est important d'étudier la performance de CRV pour les applications critiques des STI supportées par les réseaux VANET dans cette situation. Le tableau 5.9 ci-dessous présente les cas de figures élaborés en fonction des densités considérées.

Tableau 5.9 : Configuration des densités pour le scénario 3

Scénario 3	Densité	Nombre de véhicules	Vitesse maximale (Km/h)	Temps de Simulation
Tunnel	Basse	55	30	160
Tunnel	Moyenne	77	30	232
Tunnel	Haute	220	50	580

5.4.2 Optimisation de la couche 2

En vue d'améliorer la performance de CRV pour le routage cognitif VANET, il est essentiel d'optimiser la couche MAC de notre solution. Pour ce faire, nous réalisons une étude empirique en fonction de deux protocoles MAC, à savoir : STDMA([Bilstrup, K. et al., 2009](#)) et IEEE 802.11p. Ainsi, nous pourrions identifier le protocole MAC approprié permettant d'améliorer les performances de notre solution. Pour chaque protocole MAC identifié précédemment, nous évaluons la performance du routage de paquets de données en fonction des protocoles CL-AntHocNet, GPSR et CRV.

Le scénario élaboré pour l'étude empirique est basé sur la grille de Manhattan (scénario 1). Les informations liées à la carte routière ont été extraites d'OpenStreetMap et utilisées dans SUMO pour la génération l'environnement urbain. Une densité urbaine de 200 véhicules a été générée via SUMO. Nous faisons varier progressivement le nombre de véhicules intermédiaires entre le nœud source et le nœud de destination, ce qui permet de

mesurer la qualité de la transmission en fonction de la probabilité de collisions des paquets (PCP).

Le tableau **5.10** ci-dessous présente les paramètres de simulation utilisés pour l'étude empirique. Pour certains paramètres (par exemple, les protocoles de routage), plusieurs valeurs seront testées.

Tableau 5.10 : Paramètres de simulation dans NS-3 pour l'étude empirique

Paramètre	Valeur
Type de données émis	UDP
Protocole MAC	IEEE 802.11p, STDMA
Protocole Routage	CL-AntHocNet, GPSR, CRV
Durée de Simulation	200 seconds
Nombre de véhicules intermédiaires	5, 10, 15, 20
Nombre de véhicules	200
Fréquence de transmission	10 HZ
Modèle de Propagation Radio	Two-ray Ground, Winner Loss model
Nombre des paquets	5000
Taille des paquets	400
Portée de transmission	250
Nombre d'émetteurs	1
Nombre de récepteurs	1

5.4.2.1 Résultats numériques de la simulation des protocoles MAC

Dans cette sous-section, nous allons présenter les résultats numériques issus de la simulation des protocoles MAC. La métrique de performance utilisée pour mesurer la qualité au niveau de la simulation est représentée par la probabilité de collisions des paquets (PCP). Le choix de cette métrique se justifie par la nécessité de mesurer la résistance aux collisions des deux protocoles MAC et d'en sélectionner le meilleur pour

soutenir notre solution de routage pour les applications critiques des STI supportées par les réseaux VANET.

Le scénario de simulation prend en considération le nombre de nœuds intermédiaires entre la source et la destination, ce qui permet d'évaluer leur impact sur la transmission. Pour chaque scénario, nous effectuons la simulation dix fois et considérons la moyenne des PCP en fonction du protocole. Les tableaux ci-dessous présentent la moyenne des PCP en fonction du nombre de nœuds intermédiaires.

Tableau 5.11 : PCP CL-AntHocNet STDMA VS PCP CL-AntHocNet 802.11p

Nombre des véhicules intermédiaires	Moyenne Probabilité collisions Paquets (%)	
	CL-AntHocNet STDMA	CL-AntHocNet 802.11p
5	0	15,72
10	3,38	79,66
15	19,7	88,66
20	79,6	95,64

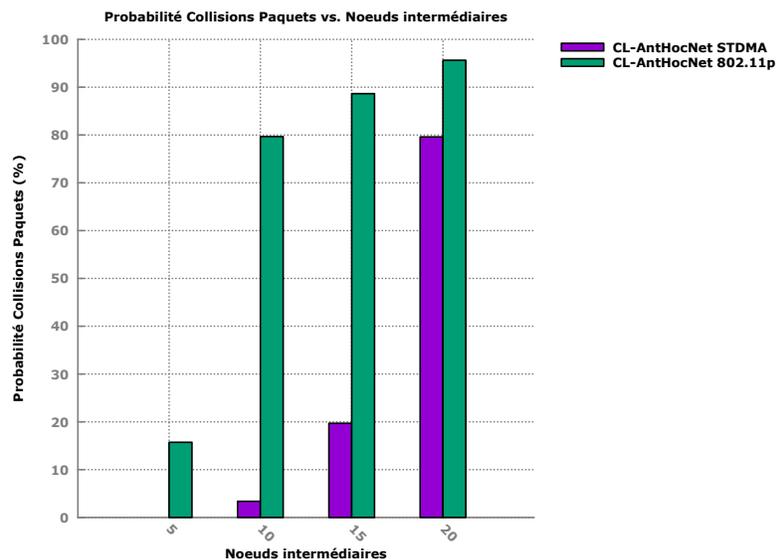


Figure 5.2 : PCP CL-AntHocNet vs. Nœuds intermédiaires

Tableau 5.12 : PCP GPSR STDMA VS. PCP GPSR IEEE 802.11p

Nombre des véhicules intermédiaires	Moyenne Probabilité collisions Paquets (%)	
	GPSR STDMA	GPSR 802.11p
5	0	15,86
10	4,96	80,38
15	16,22	80,44
20	80,08	95,62

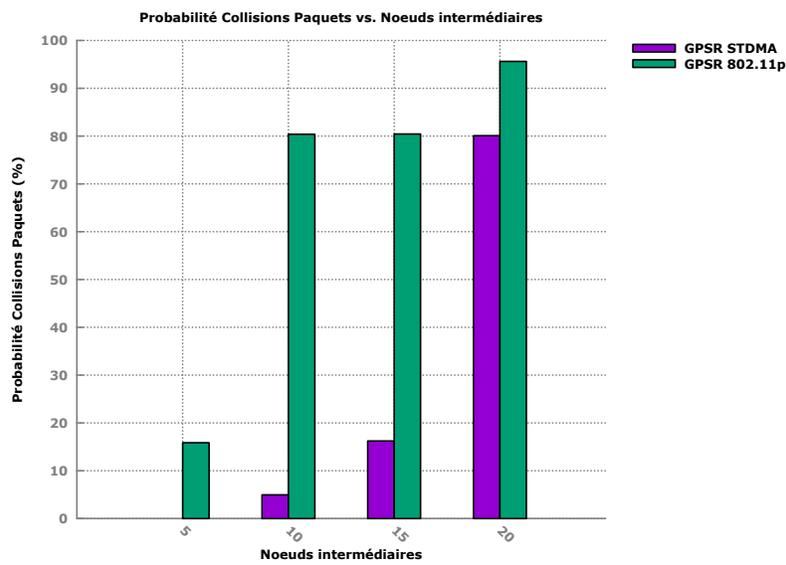


Figure 5.3 : PCP GPSR vs. Nœuds intermédiaires

Tableau 5.13 : PCP CRV STDMA VS. PCP CRV 802.11p

Nombre des véhicules intermédiaires	Moyenne Probabilité collisions Paquets (%)	
	CRV STDMA	CRV 802.11p
5	0	12,34
10	9,56	80,08
15	33,18	85,54
20	67,62	95,2

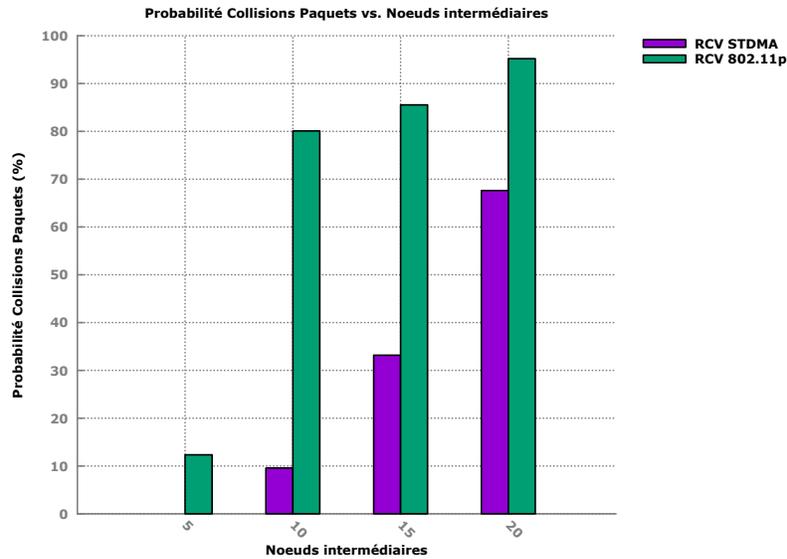


Figure 5.4 : PCP CRV vs. Nœuds intermédiaires

5.4.2.2 Discussion

Les figures 5.3 à 5.5 présentent les résultats issus de l'analyse de performance des protocoles CL-AntHocNet, GPSR et CRV en fonction des protocoles MAC STDMA et IEEE 802.11p. Nous faisons varier le nombre des nœuds intermédiaires afin de mesurer l'impact d'interférences sur la collision des paquets. En effet, les transmissions des nœuds intermédiaires peuvent générer suffisamment du bruit pouvant interférer avec la réception des paquets envoyés par le nœud source, ce qui peut occasionner des collisions de paquets au niveau du nœud de destination. Ainsi, nous constatons sur les figures précédentes qu'une augmentation graduelle des nœuds intermédiaires entraîne une forte PCP au niveau du nœud de destination pour tous les protocoles. Toutefois, lorsque le protocole MAC STDMA est utilisé comme support d'accès au canal, la PCP est moins élevée en comparaison au protocole MAC 802.11p. Cette différence peut s'expliquer par le fait que le protocole MAC STDMA supporte les applications temps réel dans les réseaux VANET ([Bilstrup, K. et al., 2009](#); [Khairnar & Kotecha, 2013](#)). Par conséquent, la robustesse de STDMA induit une résistance plus élevée aux collisions de paquets par rapport au protocole MAC 802.11p. Étant donné que STDMA présente une meilleure résistance aux collisions, nous le choisissons pour la suite de nos travaux.

5.4.3 Calibrage des paramètres de simulation NS-3

Le calibrage est une phase essentielle de notre simulation, car celle-ci nous a permis d'ajuster les paramètres du simulateur NS-3 afin d'obtenir la performance idéale du protocole CRV pour les applications critiques des STI qui sont supportées par les réseaux VANET. Plus précisément, nous avons calibré les paramètres du simulateur NS-3 en s'inspirant de la méthode *essai-erreur*. En effet, nous calibrons notre script de simulation NS-3 en essayant plusieurs paramètres de simulation, ce qui nous a permis de déterminer ceux qui sont appropriés. Le tableau 5.14 ci-dessous présente les valeurs des différents paramètres testés lors du calibrage dans NS-3 en suivant la méthode *essai-erreur*.

Tableau 5.14 : Calibrage des paramètres de simulation NS-3.

Paramètre	Description	Choix possible	Choix retenus	Raison du choix
<i>interval</i>	Temps inter paquet	Dépend du scenario	30 MilliSecondes	Meilleure fluidité des paquets
<i>nodes</i>	Nombre des véhicules	Dépend du scenario	50, 80, 200	Simuler différentes densité
<i>totaltime</i>	Temps de simulation	Dépend du scenario	600 Secondes	Permettre une meilleure analyse
<i>txp</i>	Puissance de transmission	Dépend du scenario	20 dB	Permet une meilleure réception
<i>mobility</i>	Model de mobilité	RandomWaypoint, Trace réel des véhicules	Trace reel des véhicules	Simulation plus proche de la réalité
<i>phyMode</i>	Mode Wifi	Dépend du scenario	OdfmRate6MbpsBW10MHz	Adapté pour la simulation VANET dans NS-3
<i>winner</i>	Permet de choisir le modèle de propagation	WINNER-II, Two-ray Ground	WINNER-II	Prends en considération les obstacles de type bâtiments
<i>usingWifi</i>	Choix du protocole MAC	STDMA, 802.11p	STDMA	Fourni des meilleurs résultats et support les applications temps réel dans les VANET

5.4.4 Paramètres de simulation NS-3

À l'issue du calibrage précédent, nous avons retenu certains paramètres pour la suite de la simulation. Le tableau 5.15 ci-dessous présente les paramètres de simulation retenus. Il est similaire au tableau 5.10 à une exception près : nous ne faisons pas varier les nœuds intermédiaires, le nombre véhicules ainsi que le temps de simulation sont plus élevés.

Tableau 5.15 : Paramètres de simulation dans NS-3.

Paramètre	Valeur
Type de données	CBR
Protocole Transport	UDP
Protocole MAC	STDMA
Protocole Routage	CL-AntHocNet, GPSR, CRV
Durée de Simulation	600 seconds
Nombre de véhicules	50, 80, 200
Modèle de Propagation Radio	Two-ray Ground, Winner Loss model
Nombre des paquets	5000
Taille des paquets	400
Portée de transmission	250

5.4.5 Métriques d'évaluation de performance

Afin d'étudier les performances du protocole CRV pour les applications critiques des STI qui sont supportées par les réseaux VANET, nous définissons les métriques suivantes :

- **Taux de livraison des Paquets de Données (TLPD)** : cette métrique est importante pour les applications critiques des STI qui sont supportées par les réseaux VANET, car il est crucial que tous les véhicules présents dans une zone problématique (par exemple une zone accidentée) reçoivent correctement les informations d'alerte diffusées par les véhicules en situation anormale. Elle représente le ratio entre le nombre de paquets de données reçus par le véhicule de destination et le nombre de paquets de données transmis par le véhicule source. Sa formule est donnée par l'équation ci-dessous :

$$TLPD = \left(\frac{\sum_{i=1}^n D_i}{\sum_{j=1}^n S_j} \right) * 100$$

D_i : Paquets de données reçus par le véhicule de destination.

S_i : Paquets de données transmis par le véhicule source.

- **Délai de Bout-en-Bout (DBB)** : c'est une métrique importante pour les applications critiques des STI qui sont supportées par les réseaux VANET comme par exemple l'application « *Avertissement d'une ambulance pour l'évitement d'une voie congestionnée* ». Cette métrique indique le délai entre la transmission des paquets de données et leur réception par la destination. Sa formule est donnée par l'équation ci-dessous :

$$DBB = T_R - T_S$$

T_R : Temps de réception des paquets de données par le véhicule de destination

T_S : Temps d'émission des paquets de données par le véhicule source.

5.4.6 Résultats de la simulation

Dans cette section, nous présentons les résultats numériques pour les différents scénarios de simulation décrits précédemment. Nous évaluons la performance du protocole CRV en tenant compte des obstacles présents au niveau des infrastructures urbaines. En outre, pour chaque scénario, la performance de CRV est mesurée suivant une densité véhiculaire graduelle (50, 80, 200) en répétant la simulation dix fois. Les tableaux ci-dessous présentent la valeur moyenne des performances de CRV par rapport à CL-AntHocNet et GPSR en termes de DBB et TLPD.

5.4.6.1 Résultats numériques scénario 1

Pour le scénario 1, les valeurs moyennes des TLPD ainsi que des DBB sont présentées respectivement par les tableaux 5.16 et 5.17 ci-dessous.

Tableau 5.16 : TLPD pour le scénario 1 (Grille de Manhattan)

Densité	Véhicules	TLPD CL-AntHocNet	TLPD GPSR	TLPD CRV
Basse	50	25.32	20.57	78.11
Moyenne	80	51.35	41.64	83.53
Forte	200	85.90	49.0	97.39

Tableau 5.17 : DBB pour le scénario 1 (Grille de Manhattan)

Densité	Véhicules	DBB CL-AntHocNet	DBB GPSR	DBB CRV
Basse	50	3.7573	1.779975	1.13875
Moyenne	80	4.587	1.435028	1.1448
Forte	200	5.14108	1.96374	1.64934

5.4.6.2 Résultats numériques scénario 2

Pour le scénario 2, les valeurs moyennes des TLPD ainsi que des DBB sont présentées respectivement par les tableaux 5.18 et 5.19 ci-dessous.

Tableau 5.18 : TLPD pour le scénario 2 (Pont)

Densité	Véhicules	TLPD CL-AntHocNet	TLPD GPSR	TLPD CRV
Basse densité	50	53.9624	83.0294	89.0294
Moyenne Densité	80	54.973	55.6852	94.3217
Forte densité	200	85.3495	67.0989	98.7

Tableau 5.19 : DBB pour le scénario 2 (Pont)

Densité	Véhicules	DBB CL-AntHocNet	DBB GPSR	DBB CRV
Basse densité	50	2.91159	1.686954	0.88223
Moyenne Densité	80	3.02992	1.984534	1.592
Forte densité	200	5.9267	2.53147	2.0019

5.4.6.3 Résultats numériques scénario 3

Pour le scénario 3, les valeurs moyennes des TLPD ainsi que des DBB sont présentées respectivement par les tableaux 5.20 et 5.21 ci-dessous.

Tableau 5.20 : TLPD pour le scénario 3 (tunnel)

Densité	Véhicules	TLPD CL-AntHocNet	TLPD GPSR	TLPD CRV
Basse densité	50	24.44	16.9633	74.3826
Moyenne Densité	80	30.5929	21.2526	57.6891
Forte densité	200	77.6966	25.8619	97.0606

Tableau 5.21 : DDB pour le scénario 3 (tunnel)

Densité	Véhicules	DBB CL-AntHocNet	DBB GPSR	DBB CRV
Basse densité	50	8.17621	1.396033	0.8192573
Moyenne Densité	80	6.14131	1.227466	1.0
Forte densité	200	10.46917	1.166326	1.02

5.4.7 Analyse des résultats

Dans cette section, nous analysons les résultats de la simulation pour chaque scénario.

5.4.7.1 Analyse résultats scénario 1

La figure 5.5 illustre le critère de performance TLPD pour le scénario 1 en fonction de la densité véhiculaire. Sur cette figure, nous constatons une augmentation graduelle du TLPD pour tous les protocoles au fur et à mesure que la densité véhiculaire croît. Cette augmentation s'explique par le fait qu'une forte densité dans une grille de Manhattan entraîne une forte connectivité pour la transmission des données. Le protocole CRV présente le meilleur taux de TLPD par rapport à CL-AntHocNet et GPSR pour les différentes densités véhiculaires. En effet, contrairement à CL-AntHocNet et GPSR, CRV exploite un nouveau mécanisme de routage basé sur la perception de l'environnement et

l'estimation du taux d'erreur binaire en amont de la transmission. De cette manière, CRV détermine les chemins les plus fiables en fonction des obstacles présents dans son environnement pour acheminer les paquets de données.

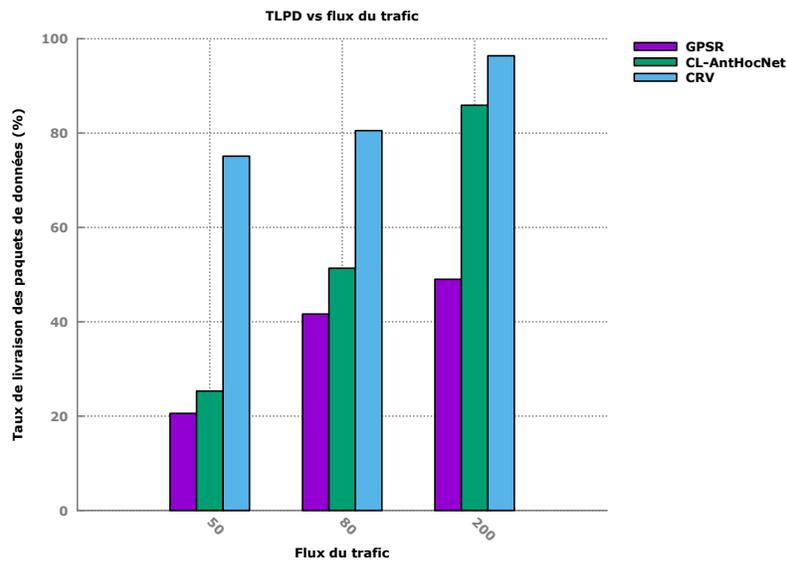


Figure 5.5 : TLPD vs. Flux du trafic (Scénario 1)

La figure 5.6 illustre les résultats en utilisant le critère de performance DBB pour le scénario 1 en fonction de la densité véhiculaire. Sur cette figure, nous constatons une augmentation progressive du DBB en fonction de la densité pour tous les protocoles. En effet, lorsque le nombre de véhicules augmente, cela permet une meilleure connectivité dans le réseau. Toutefois, l'augmentation des véhicules entraîne des interférences qui peuvent provoquer la perte des paquets de données. En cas de perte de paquets de données, tous les protocoles essaient de retransmettre l'information perdue, ce qui peut causer une augmentation du délai. Cependant, comme CRV est basé sur une architecture qui permet aux processus de s'exécuter en parallèle, le temps de traitement est optimisé. Par conséquent, le DBB de CRV est comparativement meilleur par rapport à GPSR et CL-AntHocNet.

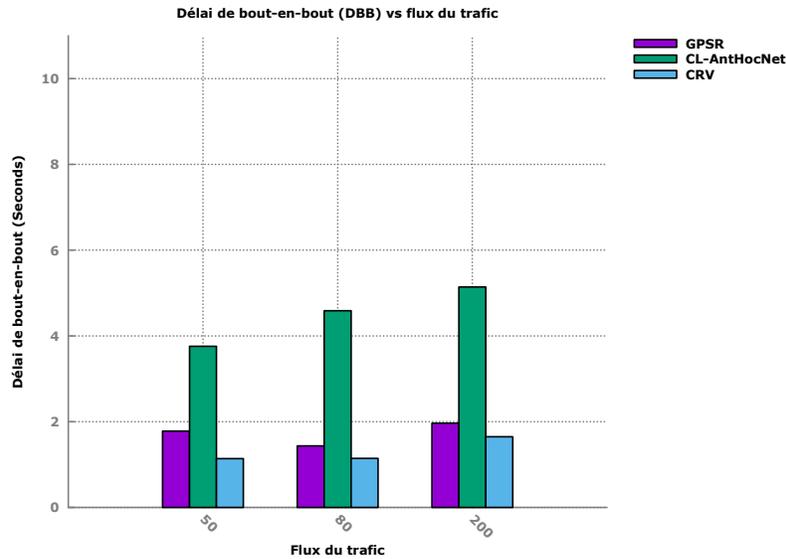


Figure 5.6 : DBB vs. Flux du trafic (Scénario 1)

5.4.7.2 Analyse résultats scénario 2

La figures 5.7 présente les résultats en fonction de la métrique de performance TLPD pour le scénario 2 par rapport à la densité véhiculaire. Ainsi, nous constatons une forte croissance du TLPD pour tous les protocoles en comparaison au scénario 1. En effet, dans ce cas de figure, la trajectoire des véhicules suit un mouvement rectiligne uniforme (c.-à-d. que les véhicules se déplacent en ligne droite) et le principal obstacle est formé par des métaux qui constituent le pont. Par conséquent, en cas de perte de paquets, l'information est retransmise rapidement et avec succès. Néanmoins, CRV qui est plus prédictif présente une performance nettement supérieure par rapport à CL-AntHocNet et GPSR.

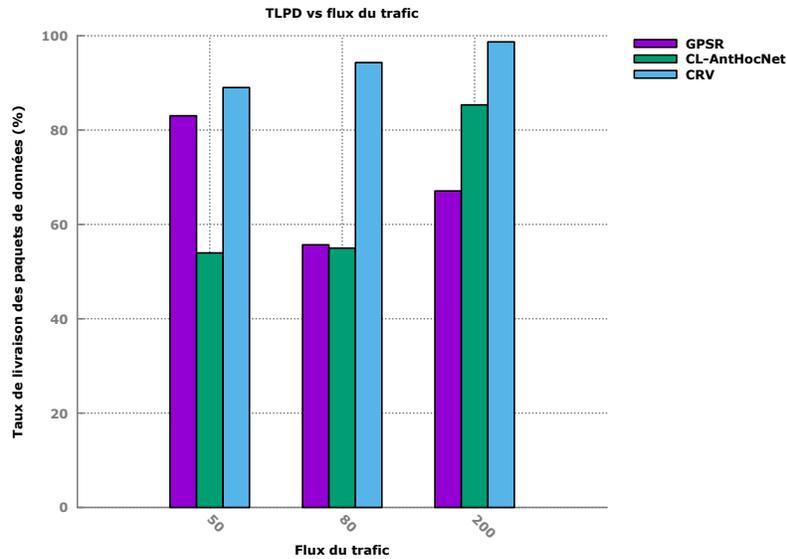


Figure 5.7 : TLPD vs. Flux du trafic (Scénario 2)

La figure 5.8 illustre le critère de performance DBB pour le scénario 2 en fonction de la densité véhiculaire. En comparaison au scénario 1, le DBB est plus élevé pour tous les protocoles. Cette croissance de délai est proportionnelle au nombre de paquets de données transmis avec succès vers la destination. En somme, une croissance du TLPD entraîne également celle du délai de transmission.

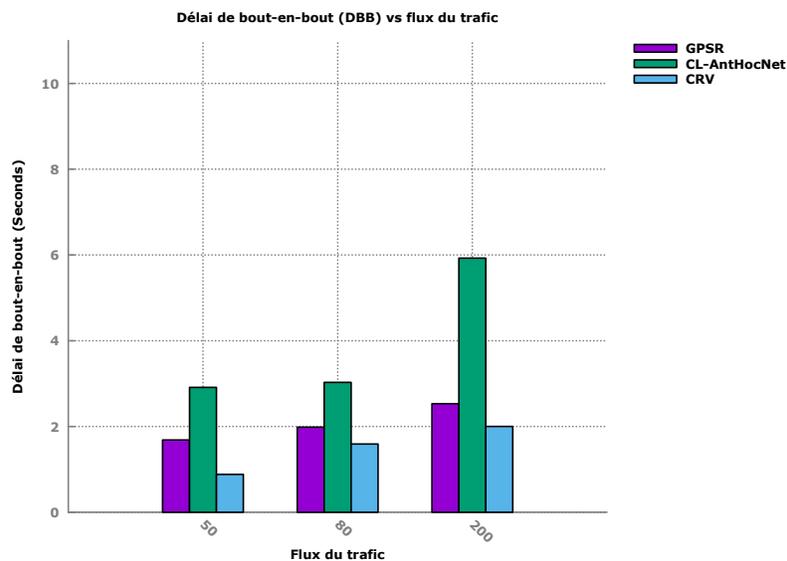


Figure 5.8 : DBB vs. Flux du trafic (Scénario 2)

5.4.7.3 Analyse résultats scénario 3

La figure 5.9 illustre le critère de performance TLPD pour le scénario 3 en fonction de la densité véhiculaire. En comparaison aux scénarios 1 et 2, nous constatons une baisse du TLPD. En effet, étant donné que les véhicules traversent un tunnel, la perte des paquets de données est beaucoup plus prononcée par rapport aux scénarios 1 et 2. Plus précisément, le signal est fortement atténué en raison de la structure souterraine du tunnel, affectant ainsi la qualité de transmission. Néanmoins, le TLPD de CRV est plus élevé par rapport à CL-AntHocNet et GPSR.

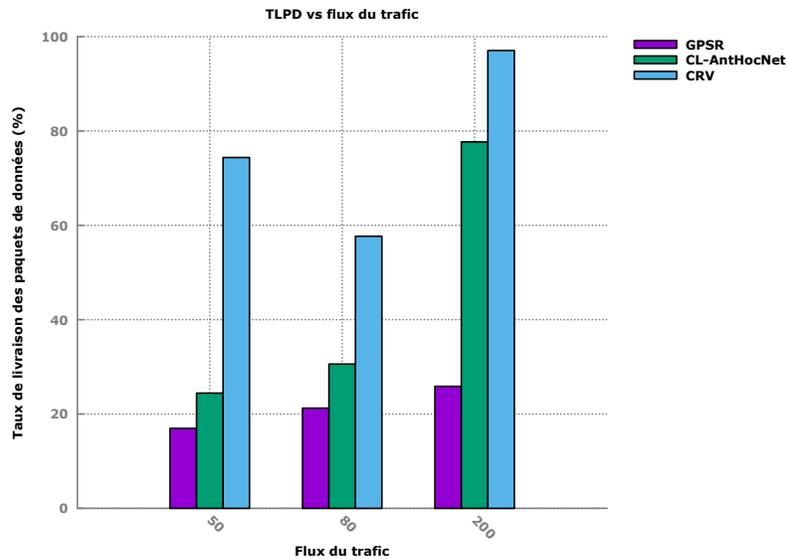


Figure 5.9 : TLPD vs. Flux du trafic (Scénario 3)

La figure 5.10 illustre le critère de performance DBB pour le scénario 3 en fonction de la densité véhiculaire. Nous constatons que le protocole CRV a le plus petit délai par rapport à CL-AntHocNet et GPSR. Cela s'explique par le fait que le délai de traitement des processus est optimisé, car ils s'exécutent en parallèle au niveau de l'architecture de CRV (voir chapitre 4).

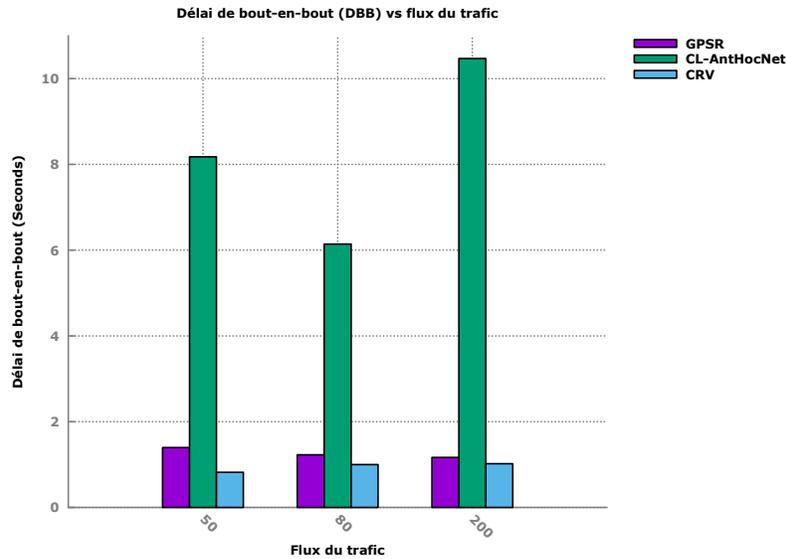


Figure 5.10 : DBB vs. Flux du trafic (Scénario 3)

5.5 Conclusion

Dans ce chapitre, nous avons analysé la performance de la solution (CRV) proposée pour le routage cognitif dans VANET par rapport à CL-AntHocNet et GPSR. Pour ce faire, nous avons élaboré des scénarios de simulation représentant les situations issues de la problématique de la thèse tel que présenté dans le chapitre 2. En outre, le choix de (voir section 5.3) GPSR et CL-AntHocNet pour analyser la performance de notre solution est guidé par la nature hybride de ce dernier.

L'analyse de performance a été effectuée suivant deux volets. Le premier volet, nous a permis de souligner l'impact des composantes urbaines sur le processus de routage. Pour ce faire, nous étudions le mécanisme de mise à jour de la table de routage de chaque protocole en analysant une séquence de communication inter-véhiculaire entre la source et la destination durant une période de temps donné. Nos expérimentations ont révélé que le mécanisme de mise à jour de la table du routage du protocole CRV permettait de réduire considérablement les délais de transmissions en évitant les liens potentiellement problématiques.

Dans le second volet de notre analyse de performance, nous avons évalué l'efficacité de notre solution par rapport à CL-AntHocNet et GPSR suivant trois scénarios, à savoir la grille de Manhattan, le pont et le tunnel. Pour chaque scénario, nous avons analysé le TLPD suivi du DBB pour les applications critiques des STI qui sont supportées par les réseaux VANET. Nous avons constaté une nette augmentation du TLPD dans tous les scénarios pour le protocole CRV par rapport à CL-AntHocNet et GPSR. Cette performance est due à la nature cognitive et proactive du protocole CRV. En effet, grâce à la cognition, le protocole CRV est en mesure d'estimer le taux d'erreur binaire en fonction des obstacles perçus en amont d'une transmission, ce qui permet d'éviter certains liens et de sélectionner le meilleur chemin de routage (celui qui présente le plus petit taux d'erreurs binaires) en améliorant le TLPD. En outre, nous avons constaté qu'une croissance du TLPD entraîne également une augmentation du délai pour tous les protocoles dans tous les scénarios étudiés. Cependant, dans le cas du protocole CRV, le délai reste meilleur par rapport aux autres protocoles, car ce dernier est basé sur une architecture permettant d'exécuter les processus parallèlement. Ce qui permet d'optimiser le délai de transmission.

En somme, le protocole de routage CRV s'avère plus prometteur relativement aux situations issues d'applications critiques des STI. Ces dernières nécessitent des réactions en temps réel et le fait d'optimiser les critères de transmissions, va participer grandement à des opérations en temps réel de grande ampleur.

Chapitre 6

Conclusions et perspectives de recherche

Sommaire du chapitre 6

6.1	Conclusions.....	159
6.2	Perspectives de recherche	161
6.2.1	Publications arbitrées dans des actes de colloque.....	162
6.2.2	Communication arbitrée.....	163

6.1 Conclusions

L'émergence des Systèmes des Transports Intelligents (STI) est inhérente au besoin de disposer des routes plus sûres et sécurisées avec une mobilité accrue. Parmi les solutions avancées, les STI s'appuient sur les réseaux VANET afin de permettre le partage d'informations en temps réel entre les véhicules ou entre les véhicules et l'infrastructure. Les réseaux VANET disposent d'une panoplie d'applications relatives à la sécurité routière, la gestion du trafic, à la gestion des incidents, etc. Les protocoles de routage jouent un rôle important pour le cheminement des informations entre les véhicules. Dans le cadre de cette thèse, nous nous sommes intéressés aux mécanismes de dissémination de l'information pour les applications ITS critiques. La gestion des incidents via des véhicules d'urgence connectés est le principal objet de cette étude. La dissémination des messages d'alertes présente comme principal objectif l'évitement des retards des véhicules d'urgence selon divers contextes et événements qui ont lieu dans un réseau routier.

En effet, la congestion des réseaux VANET a un impact sur les performances des applications d'échange de messages d'urgence sensibles aux délais. Plus précisément dans

les situations d'urgence, telles que la priorité de voie pour une ambulance, il est essentiel de garantir une fiabilité élevée et un faible délai de transmission, ce qui entraîne une complexité du problème. Cette complexité provient de la nature hautement dynamique des réseaux VANET et également de l'atténuation du signal due à diverses causes et particulièrement la présence d'obstacles au niveau de l'environnement.

La solution que nous proposons souligne l'importance de la perception des composantes urbaines et la capacité d'utiliser les conséquences qui en découlent. Pour ce faire, nous avons considéré le contexte des technologies des STI avec des véhicules connectés. Ainsi, les nœuds du réseau VANET sont présentés par des véhicules intelligents. Ces nœuds ont pour rôle les capacités de perception des composantes urbaines avec les possibilités de prédiction du niveau des dégradations des signaux de transmissions. Cette capacité des nœuds de VANET se traduit par des mises à jour des tables de routage des réseaux VANET pour éviter au maximum les temps de corrections et de retransmissions. Le nouveau protocole de routage que nous définissons est hybride. Il combine des caractéristiques des métaheuristiques en se basant sur une variante du protocole AntHocNet et aussi des paramètres géographiques en intégrant les localisations des obstacles. Notre protocole de routage qui se situe au niveau de la couche 3 de l'architecture de communications du réseau VANET présente des capacités de communications avec les nœuds représentés par des véhicules intelligents. Le protocole de routage que nous développons (CRV) utilise les résultats des traitements cognitifs d'un nœud intelligent (perception, prédiction du taux d'erreurs binaires). En plus de maintenir une architecture de cross layer, le fait d'établir des communications entre les nœuds et des processus qui définissent le nouveau protocole de routage, il est possible de réagir rapidement aux nouveaux changements des tables de routage. Ces mises à jour sont déclenchées par des événements déclenchés par les résultats des actions de perception et de prédiction des véhicules intelligents. Plusieurs tests basés sur différents scénarios de fonctionnement des réseaux VANET intégrant notre solution ont été simulés. Les résultats des simulations et leurs analyses de performance ont permis de souligner l'efficacité du protocole CRV par rapport aux protocoles GPSR et CL-AntHocNet soulignés dans notre revue de la littérature.

Les principales contributions qui découlent de cette étude se définissent selon les deux points suivants :

- 1) Structure de table de routage : élaboration d'une nouvelle structure de la table de routage pour un protocole de routage pour les réseaux VANET. Les nouvelles mises à jour des tables de routage dépendent des actions de perception et de prédiction des niveaux d'affaiblissement des signaux qui sont liés directement aux composantes urbaines. Les anticipations des dégradations des signaux se traduisent par les valeurs des tables de routage du nouveau protocole que nous définissons dans cette étude.
- 2) Véhicules autonomes et routage des réseaux VANET : élaboration d'une nouvelle architecture de communication des réseaux ad-hoc en intégrant au concept de cross-layering de nouvelles capacités d'intégration définies par les interactions entre les nœuds (véhicules autonomes) et la couche 3 de l'architecture de communications. Ces nouvelles capacités peuvent être considérées dans divers contextes et catégories de protocoles de routage.

Il est important de noter que les résultats numériques de nos expérimentations soulignent clairement les apports du point de vue délais de réponse. Ces résultats présentent les avantages et valeurs ajoutées des communications véhiculaires.

6.2 Perspectives de recherche

L'objet de l'étude que nous avons apporté pour cette thèse se situe dans un contexte qui intègre des sciences et des technologies en constante évolution. Les tendances peuvent conduire à des extensions de notre étude avec de nouveaux défis et des opportunités de progrès aux deux niveaux. Parmi les orientations futures associées à nos travaux et aux résultats obtenus, nous considérons les suivantes.

- Un module de perception d'obstacles composant un réseau routier : Le module de perception d'obstacles développé actuellement est capable de détecter certaines formes géométriques, à savoir les immeubles, les ponts ou les tunnels. Une extension de cette étude permettra de percevoir et reconnaître des caractéristiques d'objets plus complexes comme des arbres et tout autre construction qui représente

un obstacle aux signaux de transmission. Ces nouvelles capacités au niveau des véhicules autonomes permettront d'apporter de nouvelles améliorations au niveau du processus de routage basé sur le protocole CRV.

- Modéliser l'ensemble d'obstacles en 3D pour rajouter des dimensions qui ne sont pas considérées dans l'étude actuelle. Les nouveaux algorithmes de prédiction seront reconsidérés et de nouvelles métriques de qualité doivent être reconsidérées compte tenu de la diversité et quantité de données d'apprentissage.
- Considérer l'infrastructure du réseau routier selon différentes catégories et configurations. L'étude prendra en considération une combinaison entre les obstacles que nous soulignons en plus des impacts de la géométrie de l'infrastructure (autoroutes, diverses intersections, etc.).
- Étude d'une architecture intégrée et normalisée pour le routage des véhicules autonomes : L'étude actuelle est basée sur des composantes homogènes au niveau de l'architecture des véhicules autonomes et l'architecture de communication du réseau VANET (comme les couches formant le DSRC). L'extension que nous proposons consistera à définir une architecture d'API pour les communications entre le module de communications d'un véhicule autonome et l'un des processus concurrents du protocole de routage pour la mise à jour temps réel des tables de routage.

Finalement, cette étude peut être reprise avec une collecte de données basée sur un banc d'essai réel plutôt qu'une simulation. Des analyses sur les statistiques des résultats seront requises pour un déploiement réel en maintenant la méthodologie définie dans cette thèse.

Une liste des publications sur les contributions liées à cette étude est présentée ci-dessous :

6.2.1 Publications arbitrées dans des actes de colloque

- I. Benyahia, **B. Muktar.**, J.F. Perrot, "A Variant of the AnthocNet Routing Protocol: Empirical Study with Application to Communications between

Emergency Vehicles," presented at the ITS World congress, Montréal, Canada, du 29 octobre au 2 novembre 2017. **Publiée**

- I. Benyahia, **B. Muktar**, J-F. Perrot, "Cognition for autonomous vehicles: predicting and correcting the fading of transmissions", Soumis à 26 th ITS World Congress, Singapore 2019. **Accepté**
- **B. Muktar**, Ilham Benyahia. Congestion Avoidance Based on Multi-Agents Systems (MAS). TAC-ITS 2019 Joint Conference & Exhibition, Halifax, du 22 au 25 septembre 2019. **Publiée**

6.2.2 Communication arbitrée

- Njoh Njoh, Claude, Ilham Benyahia, **B. Muktar**. Processus de choix de méthode de prédiction de paramètres du trafic routier basé sur la simulation du comptage de véhicules. ACFAS. Gatineau, du 27 au 31 mai 2019.
- **B. Muktar**, I. Benyahia, "Capacités réactives d'un réseau VANet cognitif, "Concours ma thèse en 180 seconde, 15 Mars 2019, UQO, Gatineau (QC).
- **B. Muktar**, I. Benyahia, "la fin des embouteillages ?", Salon international de l'Auto, 18 janvier 2020, Palais des congrès, Montréal (QC).

Annexe A

Réseaux de Neurones Artificiels (RNA)

Les Réseaux de Neurones Artificiels (RNA) sont des modèles d'apprentissages statistiques. Ils sont inspirés des réseaux neuronaux biologiques (systèmes nerveux du cerveau humain), utilisés dans l'apprentissage automatique. Ils sont représentés comme étant des systèmes de "neurones" interconnectés qui s'échangent des messages entre eux par le biais des synapses. Les connexions au sein du réseau peuvent être systématiquement ajustées en fonction des entrées et des sorties, ce qui les rend idéales pour l'apprentissage supervisé. Un réseau de neurones comprend trois parties principales : la couche d'entrée, la couche cachée et la couche de sortie. Il peut être vu comme une boîte noire représentée par la figure A.1 ci-dessous.

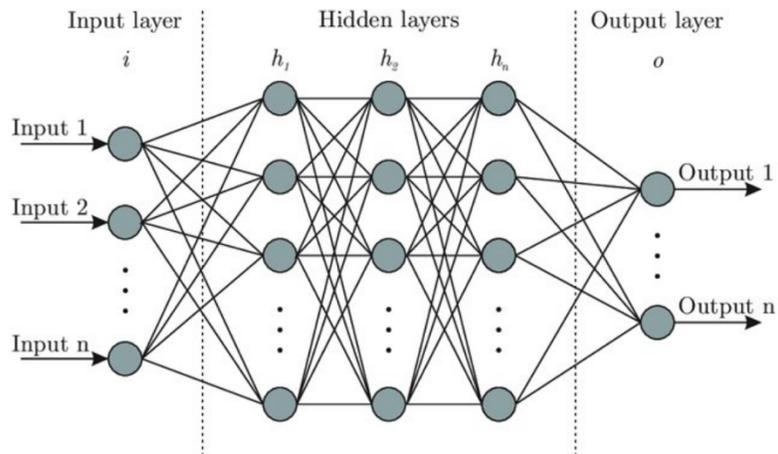


Figure A.1 : composantes d'un réseau de neurones artificiels ([Bre, Gimenez & Fachinotti, 2018](#))

L'utilisation des RNA présente un certain nombre d'avantages pour les problèmes de prédiction, à savoir :

- la capacité de traiter des données bruitées, incomplètes et incohérentes ;
- l'abstraction du processus automatisé de traitement des données fournies en entrées ;

- la capacité du réseau à apprendre sur la base d'exemples ;
- la capacité de correction d'erreur, etc.

Toutefois, même si l'utilisation des RNA présente des avantages pour les problèmes de prédiction, ils présentent un inconvénient majeur inhérent au temps d'apprentissage lorsque la taille des données fournies en entrées est conséquente. En outre, ils ne fournissent aucune garantie de résultats optimaux exacts et fiables([Colombaroni & Fusco, 2014](#); [Zhang et al., 2008](#)).

Annexe B

Fonction apprentissage pour la prédiction du TEB

```
double predicted_ber (double x_min, double x_max, double y_min, double y_max)
{
    double scaled_x_min=2*(x_min-3)/(2000-3)-1;
    double scaled_x_max=2*(x_max-1)/(1000-1)-1;
    double scaled_y_min=2*(y_min-2)/(1998-2)-1;
    double scaled_y_max=2*(y_max-1)/(999-1)-1;

    double y_1_1=tanh(-2.703277826
        -2.44797964*scaled_x_min
        +1.067063914*scaled_x_max
        -2.360853815*scaled_y_min
        -0.3630118013*scaled_y_max);

    double y_1_2=tanh(-0.9479089506
        +0.44921128*scaled_x_min
        +0.5291882471*scaled_x_max
        +0.0305373794*scaled_y_min
        -0.07022701411*scaled_y_max);

    double y_1_3=tanh(4.315352822
        -2.718258506*scaled_x_min
        -7.96017991*scaled_x_max
        +0.8713338684*scaled_y_min
        +2.747578926*scaled_y_max);

    double y_1_4=tanh(2.547918051
        +2.209096796*scaled_x_min
        -1.098283905*scaled_x_max
        +2.087270244*scaled_y_min
        +0.8039651905*scaled_y_max);

    double y_1_5=tanh(4.928593294
        -2.272996759*scaled_x_min
        +5.54545336*scaled_x_max
        +0.09691599923*scaled_y_min
        -3.272111479*scaled_y_max);

    double y_1_6=tanh(3.074688368
        -2.155991162*scaled_x_min
        -5.654603904*scaled_x_max
        +0.6175308324*scaled_y_min
        +1.860935224*scaled_y_max);

    double y_1_7=tanh(8.636460577
        -4.326967919*scaled_x_min
        +3.87784001*scaled_x_max
        +4.170355071*scaled_y_min
        -1.790268339*scaled_y_max);

    double y_1_8=tanh(-5.245498372
        +2.611644525*scaled_x_min
        -6.109567119*scaled_x_max
        +0.08050662348*scaled_y_min
        +3.48634392*scaled_y_max);
}
```

```

double y_1_9=tanh(-6.104096413
-2.858728581*scaled_x_min
+1.358650993*scaled_x_max
-4.857144454*scaled_y_min
-7.152253874*scaled_y_max);

double scaled_taux_erreurs_binaires=(0.4748772525
-1.939808969*y_1_1
+1.567195224*y_1_2
-2.891623341*y_1_3
-2.231970958*y_1_4
-3.362651919*y_1_5
+3.39470935*y_1_6
+0.5064867482*y_1_7
-3.179045567*y_1_8
-0.3212741461*y_1_9);

double taux_erreurs_binaires = (0.5*(scaled_taux_erreurs_binaires+1.0)*(100-
0)+0);

return taux_erreurs_binaires;
}

```

Annexe C

Classe perception des obstacles en 2D

Fichier obstacle-perception.h

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
#ifndef OBSTACLE_PERCEPTION_H
#define OBSTACLE_PERCEPTION_H

#include <vector>
#include "vec2d.h"
#include "obstacle.h"
#include "obstacle_loader.h"

namespace NS-3
{
    namespace obstaclePerception
    {
        class Perception
        {
        public:
            Perception();
            ~Perception();

            std::vector <Obstacle> obstacles_handler (std::string polyFile, Vec2d
CurrentVehiclePos, float detectionRange);

        private:
        };
    }
}

#endif /* OBSTACLE_PERCEPTION_H */
```

Fichier obstacle-perception.cc

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
#include "obstacle-perception.h"

namespace NS-3
{
    namespace obstaclePerception
    {
        Perception::Perception()
        {
        }

        Perception::~Perception()
        {
        }

        std::vector <Obstacle>
        Perception::obstacles_handler (string polyFile, Vec2d CurrentVehiclePos, float
detectionRange)
```

```

{
    //PerceptionObstacles perception_obstacles;
    //GetProtoFromFile(polyFile, &perception_obstacles);
    std::vector <Obstacle> obstacles;
    ObstacleLoader loader;
    obstacles = loader.loadObstacle (polyFile);

    std::vector <Obstacle> matched;
    for (int i = 0 ; i < obstacles.size(); i++)
    {
        Obstacle obs = obstacles[i];

        int match = -1;
        float d = obs.DistanceToBoundary (CurrentVehiclePos);
        if (d < detectionRange)
        {
            match = i;
        }

        if (match > -1)
        {
            Obstacle obs = obstacles[match];
            matched.push_back(obs);
        }
    }

    return matched;
}
}

```

Annexe D

Modélisation des obstacles dans NS-3

Fichier obstacle.h

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
#ifndef OBSTACLE_H
#define OBSTACLE_H

#include "polygon2d.h"

namespace NS-3
{
    namespace obstaclePerception
    {
        class Obstacle : public Polygon2d
        {
        public:
            Obstacle(std::string id, const Box2d &box);
            Obstacle(std::string id, std::vector<Vec2d> points);

            std::string getId () const ;
            void setId (std::string id);

        private:
            std::string m_id;
        };

        std::ostream & operator<<( std::ostream &out, Obstacle const & obs);
    }
}
#endif /* OBSTACLE_H */
```

Fichier obstacle.cc

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
#include "obstacle.h"

namespace NS-3
{
    namespace obstaclePerception
    {
        Obstacle::Obstacle(std::string id, const Box2d &box) : Polygon2d (box),
        m_id(id)
        {
        }

        Obstacle::Obstacle(std::string id, std::vector<Vec2d> points) : Polygon2d
        (points), m_id(id)
        {
        }

        std::string
        Obstacle::getId () const
        {
            return m_id;
        }
    }
}
```

```

}

void
Obstacle::setId(std::string id)
{
    m_id = id;
}

std::ostream & operator<<( std::ostream &out, Obstacle const & obs)
{
    /*std::vector<Vec2d> AllVertices = obs.GetAllVertices();

    out << "id : " << obs.getId() << "\n"
    << "Shape of obstacle : " << "\n"
    << "[" << "\n";
    for (int i = 0; i != AllVertices.size(); i++)
    {
        Vec2d t = AllVertices[i];
        std::cout <<" X : " << t.GetX () <<" Y : " << t.GetY () << std::endl;
    }
    return out << "]" ;*/
    out << "id = " << obs.getId() << "\n"
    << obs.DebugString() << "\n"
    << "\n";
    return out;
}
}
}

```

Bibliographie

- [1] Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37, 380-392.
- [2] Arbabi, H., & Weigle, M. C. (2010). *Highway mobility and vehicular ad-hoc networks in ns-3*. Proceedings of the Winter Simulation Conference (p. 2991-3003).
- [3] Balaji, S., Sureshkumar, S., & Saravanan, G. (2013). Cluster based ant colony optimization routing for vehicular ad hoc networks. *International Journal of Scientific & Engineering Research*, 4(6), 26-30.
- [4] Bandi, A. (2015). *Parameters tuning of OLSR routing protocol with metaheuristic algorithm for VANET*. IEEE International Advance Computing Conference (IACC) (p. 1207-1212).
- [5] Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). *SUMO—simulation of urban mobility: an overview*. Proceedings of SIMUL, The Third International Conference on Advances in System Simulation.
- [6] Benyahia, I., Bappa, M., & Perrot, J. F. (2017). *A Variant of the AnthocNet Routing Protocol: Empirical Study with Application to Communications between Emergency Vehicles*. ITS World congress (p. 10).
- [7] Bilal, S. M., Bernardos, C. J., & Guerrero, C. (2013). Position-based routing in vehicular networks: A survey. *Journal of Network and Computer Applications*, 36(2), 685-697.
- [8] Bilstrup, K., Uhlemann, E., Ström, E., & Bilstrup, U. (2009). On the ability of the 802.11 p MAC method and STDMA to support real-time vehicle-to-vehicle communication. *EURASIP Journal on Wireless Communications and Networking*, 2009(1), 902414.

- [9] Bilstrup, K. S., Uhlemann, E., & Strom, E. G. (2010). *Scalability issues of the MAC methods STDMA and CSMA of IEEE 802.11 p when used in VANETs*. 2010 IEEE International Conference on Communications Workshops (p. 1-5).
- [10] Bitaghsir, S. A., & Hendessi, F. (2011). An intelligent routing protocol for delay tolerant networks using genetic algorithm. Dans *Smart Spaces and Next Generation Wired/Wireless Networking* (p. 335-347). Springer.
- [11] Bitam, S., Mellouk, A., & Zeadally, S. (2013). HyBR: A hybrid bio-inspired bee swarm routing protocol for safety applications in vehicular ad hoc networks (VANETs). *Journal of Systems Architecture*, 59(10), 953-967.
- [12] Blum, J. J., Eskandarian, A., & Hoffman, L. J. (2004). Challenges of intervehicle ad hoc networks. *IEEE transactions on intelligent transportation systems*, 5(4), 347-351.
- [13] Booyesen, M. J., Zeadally, S., & Van Rooyen, G.-J. (2011). Survey of media access control protocols for vehicular ad hoc networks. *IET communications*, 5(11), 1619-1631.
- [14] Borgonovo, F., Capone, A., Cesana, M., & Fratta, L. (2002). *RR-ALOHA, a Reliable R-ALOHA broadcast channel for ad-hoc inter-vehicle communication networks*. Proceedings of Med-Hoc-Net (vol. 2002).
- [15] Borne, P., Benrejeb, M., & Haggège, J. (2007). *Les réseaux de neurones: présentation et applications* (vol. 15). Editions OPHRYS.
- [16] BoussaïD, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82-117.
- [17] Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 1429-1441.
- [18] Bu, J., Tan, G., Ding, N., Liu, M., & Son, C. (2014). *Implementation and evaluation of WAVE 1609.4/802.11 p in ns-3*. Proceedings of the 2014 Workshop on ns-3 (p. 1).

- [19] Cavalcante, E. S., Aquino, A. L., Pappa, G. L., & Loureiro, A. A. (2012). *Roadside unit deployment for information dissemination in a VANET: An evolutionary approach*. Proceedings of the 14th annual conference companion on Genetic and evolutionary computation (p. 27-34).
- [20] Chang, X. (1999). *Network simulations with OPNET*. WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038) (vol. 1, p. 307-314).
- [21] Colombaroni, C., & Fusco, G. (2014). Artificial neural network models for car following: experimental analysis and calibration issues. *Journal of Intelligent Transportation Systems*, 18(1), 5-16.
- [22] Daizo, H., Iwahashi, T., Bandai, M., & Watanabe, T. (2004). *An inter-vehicle communication MAC protocol supported by roadside communication and its extension*. Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (p. 89-90).
- [23] Darwish, T., & Bakar, K. A. (2016). Traffic aware routing in vehicular ad hoc networks: characteristics and challenges. *Telecommunication systems*, 61(3), 489-513.
- [24] Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C., & Harley, R. G. (2008). Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on evolutionary computation*, 12(2), 171-195.
- [25] Di Caro, G., Ducatelle, F., & Gambardella, L. M. (2004). *AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks*. International Conference on Parallel Problem Solving from Nature (p. 461-470).
- [26] Dimitrakopoulos, G., & Demestichas, P. (2010). Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1), 77-84.
- [27] Ducatelle, F. (2007). *Adaptive routing in ad hoc wireless multi-hop networks*, Università della Svizzera italiana].

- [28] Englund, C., Chen, L., Vinel, A., & Lin, S. Y. (2015). Future applications of VANETs. Dans *Vehicular ad hoc Networks* (p. 525-544). Springer.
- [29] Fan, H., Yang, B., Zipf, A., & Rousell, A. (2016). A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data. *International Journal of Geographical Information Science*, 30(4), 748-764.
- [30] Gaugel, T., Mittag, J., Hartenstein, H., Papanastasiou, S., & Ström, E. G. (2013). *In-depth analysis and evaluation of self-organizing TDMA*. 2013 IEEE vehicular networking conference (p. 79-86).
- [31] Gong, J., Xu, C.-Z., & Holle, J. (2007). *Predictive directional greedy routing in vehicular ad hoc networks*. 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07) (p. 2-2).
- [32] Granelli, F., Boato, G., & Kliazovich, D. (2006). *MORA: A movement-based routing algorithm for vehicle ad hoc networks*. IEEE Workshop on Automotive Networking and Applications (AutoNet), San Francisco, USA (p. 1-7).
- [33] Gupta, D., & Kumar, R. (2014). *An improved genetic based routing protocol for VANETs*. 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence) (p. 347-353).
- [34] Haklay, M., & Weber, P. (2008a). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12-18.
- [35] Haklay, M., & Weber, P. (2008b). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12-18.
- [36] Härri, J., Bonnet, C., & Filali, F. (2008). Kinetic mobility management applied to vehicular ad hoc network protocols. *Computer Communications*, 31(12), 2907-2924.
- [37] Hartenstein, H., Bochow, B., Ebner, A., Lott, M., Radimirsch, M., & Vollmer, D. (2001). *Position-aware ad hoc wireless networks for inter-vehicle communications: the Fleetnet project*. Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (p. 259-262).

- [38] Hartenstein, H., & Laberteaux, L. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications magazine*, 46(6), 164-171.
- [39] Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., & Kopena, J. (2008). Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14), 527.
- [40] Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2), 88-105.
- [41] Hu, L., Ding, Z., & Shi, H. (2012). *An improved GPSR routing strategy in VANET*. 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing (p. 1-4).
- [42] Issariyakul, T., & Hossain, E. (2009). Introduction to network simulator 2 (NS2). Dans *Introduction to network simulator NS2* (p. 1-18). Springer.
- [43] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. (2001). *Optimized link state routing protocol for ad hoc networks*. Proceedings. IEEE International Multi Topic Conference. IEEE INMIC. Technology for the 21st Century. (p. 62-68).
- [44] Jerbi, M., Senouci, S.-M., Rasheed, T., & Ghamri-Doudane, Y. (2009). Towards efficient geographic routing in urban vehicular networks. *IEEE Transactions on Vehicular Technology*, 58(9), 5048-5059.
- [45] Kaiwartya, O., & Kumar, S. (2014). *Geocasting in vehicular adhoc networks using particle swarm optimization*. Proceedings of the International Conference on Information Systems and Design of Communication (p. 62-66).
- [46] Kalambe, K., Deshmukh, A., & Dorle, S. (2015). Particle swarm optimization based routing protocol for vehicular ad hoc network. *Int. J. Eng. Res. General Sci*, 3(1), 1375-1382.
- [47] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3), 459-471.
- [48] Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1), 687-697.

- [49] Karp, B. (2001). *Challenges in geographic routing: Sparse networks, obstacles, and traffic provisioning*. Presentation at the DIMACS Workshop on Pervasive Networking (vol. 588).
- [50] Karp, B., & Kung, H.-T. (2000). *GPSR: Greedy perimeter stateless routing for wireless networks*. Proceedings of the 6th annual international conference on Mobile computing and networking (p. 243-254).
- [51] Kenney, J. B. (2011). Dedicated short-range communications (DSRC) standards in the United States. *Proceedings of the IEEE*, 99(7), 1162-1182.
- [52] Khairnar, V. D., & Kotecha, K. (2013). Performance of vehicle-to-vehicle communication using IEEE 802.11 p in vehicular ad-hoc network environment. *arXiv preprint arXiv:1304.3357*.
- [53] Khandai, M., Acharya, A. A., & Mohapatra, D. P. (2011). *A novel approach of test case generation for concurrent systems using UML Sequence Diagram*. 2011 3rd International Conference on Electronics Computer Technology (vol. 1, p. 157-161).
- [54] Kumar, S., & Verma, A. K. (2015). Position based routing protocols in VANET: A survey. *Wireless Personal Communications*, 83(4), 2747-2772.
- [55] [#154 utilise un type de document non défini dans ce style].
- [56] Lee, K. C., Härri, J., Lee, U., & Gerla, M. (2007). *Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios*. *IEEE Globecom workshops* (p. 1-10).
- [57] Lee, K. C., Lee, U., & Gerla, M. (2010). Survey of routing protocols in vehicular ad hoc networks. Dans *Advances in vehicular ad-hoc networks: Developments and challenges* (p. 149-170). IGI Global.
- [58] Li, F., & Wang, Y. (2007). Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular technology magazine*, 2(2), 12-22.
- [59] Lobiyal, D., Katti, C., & Giri, A. (2015). Parameter value optimization of ad-hoc on demand multipath distance vector routing using particle swarm optimization. *Procedia Computer Science*, 46, 151-158.

- [60] Lochert, C., Hartenstein, H., Tian, J., Fussler, H., Hermann, D., & Mauve, M. (2003). *A routing strategy for vehicular ad hoc networks in city environments*. IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683) (p. 156-161).
- [61] Lochert, C., Mauve, M., Füßler, H., & Hartenstein, H. (2005). Geographic routing in city scenarios. *ACM SIGMOBILE mobile computing and communications review*, 9(1), 69-72.
- [62] Lochert, C., Scheuermann, B., Wewetzer, C., Luebke, A., & Mauve, M. (2008). *Data aggregation and roadside unit placement for a vanet traffic information system*. Proceedings of the fifth ACM international workshop on VehiculAr Inter-NEtworking (p. 58-65).
- [63] Majumdar, S., Prasad, P. R., Kumar, S. S., & Kumar, K. S. (2016). *An efficient routing algorithm based on ant colony optimisation for VANETs*. IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (p. 436-440).
- [64] Mak, T. K., Laberteaux, K. P., & Sengupta, R. (2005). *A multi-channel VANET providing concurrent safety and commercial services*. Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks (p. 1-9).
- [65] Mo, Z., Zhu, H., Makki, K., & Pissinou, N. (2006). *MURU: A multi-hop routing protocol for urban vehicular ad hoc networks*. Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (p. 1-8).
- [66] Musliu, N. (2010). Problem Solving and Search in Artificial Intelligence. *Database and Artificial Intelligence Group, Institut für Informations systeme, TU-Wien*.
- [67] Parker, R., & Valaee, S. (2006). *Vehicle localization in vehicular networks*. IEEE Vehicular Technology Conference (p. 1-5).
- [68] Patzold, M. (2001). *Mobile fading channels: Modelling, analysis and simulation*. John Wiley & Sons, Inc.
- [69] Paul, C., & Vishwakarma, G. K. (2017). Back propagation neural networks and multiple regressions in the case of heteroskedasticity. *Communications in Statistics-Simulation and Computation*, 46(9), 6772-6789.

- [70] Rana, H., Thulasiraman, P., & Thulasiram, R. K. (2013). *MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET*. IEEE congress on evolutionary computation (p. 2948-2955).
- [71] Rappaport, T. S. (1996). *Wireless communications: principles and practice* (vol. 2). prentice hall PTR New Jersey.
- [72] Russell, S., & Norvig, P. (2010). *Intelligence artificielle: Avec plus de 500 exercices*. Pearson Education France.
- [73] Saleet, H., Langar, R., Basir, O., & Boutaba, R. (2009). *Adaptive message routing with QoS support in vehicular ad hoc networks*. GLOBECOM IEEE Global Telecommunications Conference (p. 1-6).
- [74] Schmidt, R. K., Köllmer, T., Leinmüller, T., Böddeker, B., & Schäfer, G. (2009). Degradation of Transmission Range in VANETS caused by Interference. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, 32(4), 224-234.
- [75] Seet, B.-C., Liu, G., Lee, B.-S., Foh, C.-H., Wong, K.-J., & Lee, K.-K. (2004). *A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications*. International Conference on Research in Networking (p. 989-999).
- [76] Series, M. (2010). Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band. *Recommendation ITU*, 1371-1375.
- [77] Sharma, S., Giri, A., & Singhal, N. (2014). Finding optimal configuration of DSDV using particle swarm optimization. *International Journal of Computer Applications*, 104(4).
- [78] Singh, A., Thakur, N., & Sharma, A. (2016). *A review of supervised machine learning algorithms*. 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (p. 1310-1315).
- [79] Siraj, S., Gupta, A., & Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(4), 199-206.

- [80] Sommer, C., & Dressler, F. (2007). *The DYMO routing protocol in VANET scenarios*. IEEE 66th Vehicular Technology Conference (p. 16-20).
- [81] Srivastava, V., & Motani, M. (2005). Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine*, 43(12), 112-119.
- [82] Tian, D., Shafiee, K., & Leung, V. C. (2009). *Position-based directional vehicular routing*. GLOBECOM IEEE Global Telecommunications Conference (p. 1-6).
- [83] TR, E. (2012). *102 861 Intelligent Transport Systems (ITS); STDMA Recommended Parameters and Settings for Cooperative ITS; Access Layer Part*. Technical Report, ETSI.
- [84] Varga, A., & Hornig, R. (2008). *An overview of the OMNeT++ simulation environment*. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (p. 60).
- [85] Wang, K.-P., Huang, L., Zhou, C.-G., & Pang, W. (2003). *Particle swarm optimization for traveling salesman problem*. Proceedings of the international conference on machine learning and cybernetics (IEEE cat. no. 03ex693) (vol. 3, p. 1583-1585).
- [86] Weiland, R. J., & Purser, L. B. (2000). *Intelligent transportation systems. Transportation in the new millennium*.
- [87] Xu, Q., Mak, T., Ko, J., & Sengupta, R. (2004). *Vehicle-to-vehicle safety messaging in DSRC*. Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (p. 19-28).
- [88] Yuan, Y., Chen, H., & Jia, M. (2005). *An optimized ad-hoc on-demand multipath distance vector (AOMDV) routing protocol*. Asia-Pacific Conference on Communications (p. 569-573).
- [89] Zhang, L., Wang, J., Li, K., Yamamura, T., Kuge, N., & Nakagawa, T. (2008). *Driver car-following behavior modeling using neural network based on real traffic experimental data*. 15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual MeetingITS AmericaERTICOITS JapanTransCore.
- [90] Zhao, J., & Cao, G. (2008). VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE transactions on vehicular technology*, 57(3), 1910-1922.

