

THÈSE

présentée en vue d'obtenir le grade de

Docteur de l'Université de Lille

en

Sciences et technologies de l'information et de la communication

par

Lucas RIVOIRARD

Modèle d'auto-organisation pour les protocoles de routage dans les réseaux ad hoc de véhicules : application à la perception élargie et à la localisation coopératives

Soutenue le 21 septembre 2018 devant le jury d'examen :

Rapporteur	Pr Anthony Busson	Univ. Lyon 1, LIP
Rapporteur	Dr-HDR Selma Boumerdassi	CNAM, CEDRIC
Examineur	Dr-HDR Alain PIROVANO	ENAC
Examineur	Pr Soumaya CHERKAOUI	Univ. Sherbrooke
Invité	Dr Jochen LANGHEIM	STMicroelectronics
Directeur de thèse	DR-HDR Dominique GRUYER	IFSTTAR, COSYS-LIVIC
Directeur de thèse	DR Marion BERBINEAU	IFSTTAR, COSYS-LEOST
Encadrant de thèse	Dr Martine WAHL	IFSTTAR, COSYS-LEOST
Encadrant de thèse	Dr Patrick SONDI	Univ. ULCO, LISIC

Résumé

Automatiser la conduite peut réduire le risque d'accidents et fluidifier le trafic en s'appuyant sur des communications entre véhicules et avec une infrastructure dédiée. Maintenir les services lorsque l'infrastructure faillit nécessite un protocole de routage adapté aux communications véhicule-à-véhicule (V2V).

Cette thèse pose l'hypothèse de l'absence d'infrastructure et de service global de localisation, chaque véhicule connaissant sa seule position. Un modèle d'organisation des communications V2V, *Chain Branch Leaf* (CBL), est proposé pour créer et maintenir une structure supportant les échanges proches, entre véhicules voisins, et lointains à travers le trafic. Il crée une hiérarchie où des groupes dissociés de véhicules voisins (feuilles) sont connectés via un véhicule leader (branche) à une colonne vertébrale (chaîne) formée le long de la route par interconnexion des branches. *Chain Branch Leaf* (CBL) requière la seule connaissance du voisinage à un saut et peut s'intégrer à tout protocole de routage.

Les évaluations sur des scénarios routiers montrent que CBL exploite mieux les contraintes spatiales dues à la route, obtenant une structure plus adaptée à l'optimisation du trafic *broadcast* comparativement à la technique des multipoints relais dans OLSR, mais aussi face aux protocoles DSR, AODV et GRP, pour un trafic basé sur les préconisations IEEE de taille et fréquence de messages en milieu routier. Des évaluations sur des trafics d'applications coopératives telles que l'égo-localisation et la perception élargie montrent que CBL offre un service de communication à latence et taux de pertes faibles selon un compromis voulu entre fréquence de messages et performances.

Mots clés : Réseaux véhiculaires ad hoc, OLSR, protocole de routage, groupement, structuration, réseau de communication, véhicule coopératif, liaison V2V.

Abstract

Automated driving can reduce accidents and improve the traffic flow by using communications between vehicles and dedicated infrastructures. Maintaining services when an infrastructure failed requires a routing protocol adapted to vehicle-to-vehicle (V2V) communications.

This dissertation assumes the absence of dedicated infrastructure and global location service, each vehicle knowing its only own position. A V2V communication organization model, *Chain Branch Leaf* (CBL), is proposed to create and maintain a structure that supports close exchanges between neighbouring vehicles and remote exchanges through the traffic. CBL creates a hierarchy where disjoint groups of neighbouring vehicles (leaves) are connected via a leader vehicle (branch) to a backbone (chain) formed along the road by the interconnection of the branches. CBL requires only the knowledge of the one-hop neighborhood and can be integrated into any routing protocol.

The evaluations on road scenarios using IEEE-based traffic recommended on road environments in terms of size and message frequency show that CBL makes a better use of the spatial constraints due to the road. A structure is obtained that suits better in the optimisation of the broadcast traffic than the technique of multipoint relays in OLSR, but also compared to DSR, AODV and GRP protocols. Evaluations of cooperative application traffic such as ego-localization and extended perception show that CBL offers a communication service with low latency, low packet loss, according to a wanted trade-off between message frequency and performance.

Keywords : Ad hoc vehicle networks, OLSR, routing protocol, clustering, communication network, cooperative vehicle, V2V link.

Remerciements

Les travaux présentés dans ce manuscrit sont le fruit de mes trois années de doctorat réalisées, à Villeneuve d'Ascq, au sein du laboratoire LEOST de l'IFSTTAR.

Je remercie tout d'abord Monsieur Charles Tatkeu, directeur du laboratoire, qui m'a accueilli au LEOST et donné les moyens de réaliser mes déplacements nécessaires à la présentation de mes travaux et à la construction de mon réseau professionnel.

Je voudrais ensuite exprimer ma reconnaissance envers mes deux directeurs de thèse, Madame Marion Berbineau, directrice de recherche au LEOST et Monsieur Dominique Gruyer, directeur du laboratoire LIVIC de l'IFSTTAR, site de Versailles-Satory, pour leur travail de supervision.

Je tiens à adresser ma profonde gratitude à Madame Martine Wahl, chargée de recherche au LEOST, qui m'a encadré et conseillé scientifiquement tout au long de cette thèse. Qu'elle soit également remerciée pour sa gentillesse, sa franchise, sa disponibilité permanente et pour les nombreux encouragements qu'elle m'a prodigués. C'est à ses côtés que j'ai compris ce que rigueur et précision voulaient dire.

Je remercie également Monsieur Patrick Sondi, enseignant-chercheur à l'Université du Littoral Côte d'Opale, pour son expertise et ses remarques. Je suis ravi d'avoir travaillé en sa compagnie car, outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

J'adresse tous mes remerciements à Madame Selma Boumerdassi, enseignant-chercheur HDR au CNAM, ainsi qu'à Monsieur Anthony Busson, Professeur à l'Université Lyon 1, d'avoir accepté d'être rapporteur de cette thèse et pour l'intérêt qu'ils ont porté à mon travail.

J'exprime ma gratitude à Madame Soumaya CHERKAOUI, Professeur à l'Université de Sherbrooke et à Monsieur Alain PIROVANO, enseignant-chercheur à l'ENAC, qui ont bien voulu être examinateurs et je tiens à remercier Monsieur Jochen LANGHEIM, invité, d'avoir accepté de se joindre à mon jury de thèse.

Je remercie également mes nombreux collègues de l'IFSTTAR à Villeneuve d'Ascq qui ont partagé mon quotidien durant cette thèse, en commençant par les plus anciens : Matthieu, Philippe, Antoine, Christophe, Julien, Abel, Pierre, Ni, Anthony, Abderraouf, Mohamed Raouf, Camélia, Imen, Michel, Rémi, Romain, Baptiste, Clément, Corentin, Anne, Enji, Nathan, Céline, Elia, Nicolas, Sébastien, Véréna et Amani. Je les remercie tous pour leur bonne humeur, leur sympathie et les nombreux échanges qui ont contribué de quelque manière que ce soit au bon déroulement de ma thèse. Je remercie aussi toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Au terme de ce parcours, je remercie enfin celles et ceux qui me sont chers et que j'ai quelque peu délaissés ces derniers mois pour achever ce manuscrit. Leurs attentions et encouragements m'ont accompagné tout au long de ces années. Je voudrais aussi exprimer toute ma reconnaissance à mes parents, pour leur encouragement, et j'ai une pensée toute particulière pour ma compagne pour sa patience et son soutien sans faille tout au long de mon travail.

Table des matières

Résumé	3
Remerciements	7
Table des figures	12
Table des tableaux	17
Glossaire	19
Introduction	25
I Vers un protocole de routage véhiculaire adaptatif : CBL	31
1 Protocoles de routage dans les réseaux mobiles ad hoc	33
1.1 Introduction aux réseaux de communication véhiculaires	33
1.1.1 Contraintes physiques inhérentes au médium de communication	35
1.1.2 Contraintes imposées par le contexte routier	35
1.1.3 Passage à l'échelle	35
1.2 Protocoles de routage pour les réseaux mobiles ad hoc	36
1.2.1 Introduction	36
1.2.2 Établissement des liens logiques du réseau	37
1.2.3 Modes de communication	40
1.2.4 Recherche de route par dissémination de requêtes	42
1.3 Stratégies de routage pour le contexte véhiculaire	45
1.3.1 Stratégies et informations contextuelles propres aux systèmes routiers	45
1.3.2 Stratégies de routage et métriques de qualité de service	53
1.4 Conclusion	56
2 Proposition d'organisation structurelle : CBL	59
2.1 Introduction	60
2.2 Définitions et métriques dédiées au schéma CBL	61
2.2.1 Notion de nœuds branche et feuille	61
2.2.2 Notion de chaîne	62
2.2.3 Spécification du sens de propagation d'un message applicatif	63
2.2.4 Définition des métriques utilisées	63
2.2.5 Spécification des tables des voisins à un saut	64
2.2.6 Spécification des variables locales des nœuds	65
2.2.7 Spécification du contenu des messages HELLO	67
2.2.8 Définitions d'un seuil temporel et des coefficients pondérateurs	67
2.3 Algorithmes de construction de la structure CBL	70
2.3.1 Procédure d'émission d'un message HELLO	70
2.3.2 Diagramme fonctionnel de CBL à réception d'un message HELLO	70
2.3.3 Mise à jour de la table des voisins (algorithme 1)	70
2.3.4 Procédure de changement de type, de feuille à branche (algorithme 2)	72

2.3.5	Traitement type feuille (algorithme 3)	72
2.3.6	Traitement type branche (algorithme 4)	75
2.3.7	Procédure de changement de type, de branche à feuille (algorithme 5)	82
2.3.8	Diagrammes d'interactions temporelles	83
2.3.9	Paramétrage des coefficients pondérateurs	87
2.4	Discussion –Propriétés de CBL	90
2.4.1	Élection d'un nœud branche par un nœud feuille (algorithme 3)	90
2.4.2	Gestion des chaînes (algorithme 4)	91
2.5	Mise en œuvre de CBL dans le protocole OLSR : CBL-OLSR	94
2.5.1	Présentation du protocole OLSR	94
2.5.2	Correspondance entre MPRs et nœuds branche	96
2.5.3	Modification du format des paquets	96
2.5.4	Modification du format des messages HELLO	98
2.5.5	Méthode de recherche de route	100
2.5.6	Modification des tables de voisins	100
II Analyse de performances de CBL		103
3 Cahier des charges pour l'évaluation des performances d'un protocole de routage dans le contexte véhiculaire		105
3.1	Introduction	105
3.2	Évaluation structurelle et évaluation système	106
3.3	Simulateurs de réseaux de communication	107
3.3.1	Simulateurs utilisés dans les travaux référencés au chapitre 1	107
3.3.2	Environnements de simulation	108
3.4	Modélisation de la mobilité	109
3.4.1	Modèles de mobilité choisis dans les travaux référencés au chapitre 1	109
3.4.2	Paramétrages du simulateur SUMO	115
3.5	Modèles de propagation des ondes radio	119
3.5.1	Modèles de propagation des ondes radio choisis dans les travaux référencés au chapitre 1	119
3.5.2	Modélisation de la propagation sous OPNET Riverbed Modeler	122
3.6	Modèle de la technologie de transmission de trames	124
3.6.1	Technologies existantes	124
3.6.2	Description du protocole IEEE 802.11p	124
3.6.3	Paramétrage dans Riverbed OPNET Modeler	126
3.7	Modélisation d'applications	127
3.7.1	Classification des applications	127
3.7.2	Besoins de communication	128
3.8	Métriques de performances pour l'évaluation de protocoles de routage	129
4 Analyse structurelle		135
4.1	Analyses de performances de CBL et des <i>Multipoint Relays</i> (MPRs) sous Matlab	135
4.1.1	Évaluation de la structure CBL	135
4.1.2	Évaluation de la méthode des MPRs relativement à CBL version 2	147
4.2	Analyses de performances de CBL-OLSR et OLSR sous OPNET	152
4.2.1	Analyse de l'intégration du schéma CBL sous OPNET	153
4.2.2	Analyse du trafic de routage sous Riverbed OPNET Modeler	154
4.3	Étude de sensibilité de la structure CBL	160
4.3.1	Sensibilité aux coefficients pondérateurs	160
4.3.2	Sensibilité aux scénarios de mobilité	160
4.3.3	Sensibilité aux modèles de propagation	164
4.4	Conclusion	164

5	Analyse système	167
5.1	Évaluations de CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP sous OPNET	167
5.1.1	Présentation des protocoles et spécification de leurs paramétrages	168
5.1.2	Paramétrages de la simulation	170
5.1.3	Métriques de performance	171
5.1.4	Analyse des résultats – scénario applicatif Sa2	171
5.1.5	Analyse des résultats – scénarios applicatifs Sa1-Sa3	178
5.2	Application de localisation distribuée coopérative	180
5.2.1	Intérêt de la localisation pour les applications véhiculaires	180
5.2.2	Présentation de l’application de corrélation cartographique coopérative (CMM) . .	182
5.2.3	Présentation de la station de base dynamique (DDGPS)	183
5.2.4	Adaptation de l’application au schéma CBL et modélisation	184
5.2.5	Évaluation de performances par la simulation	185
5.3	Application de perception élargie	190
5.3.1	Présentation de l’application	190
5.3.2	Besoins en communication	191
5.3.3	Adaptation de l’application au schéma CBL et modélisation	192
5.3.4	Évaluation de performances par la simulation	192
5.4	Conclusion	194
	Bibliographie	201
	Annexes	203
	Références bibliographiques	215
A	Algorithmes de la version 1 de CBL	217
A.1	CBL scheme	217
A.1.1	Algorithm 1. Update the 1-hop neighbor table	218
A.1.2	Algorithm 2. Turn leaf into branch	218
A.1.3	Algorithm 3. Leaf processing	219
A.1.4	Algorithm 4. Branch processing	221
A.1.5	Algorithm 5. Turn branch into leaf	221
B	Proposition d’un codage de la variable “position”	225
C	Petit guide d’utilisation d’OPNET	227
C.1	Fonctionnement d’OPNET Riverbed Modeler	227
C.2	Lancement d’une simulation	228
C.3	Utilisation de l’affectation automatique de trajectoire	229
C.4	Implantation de CBL au sein du protocole OLSR	229
D	Petit guide d’utilisation de SUMO	233
D.1	Création du réseau	233
D.2	Génération du réseau	233
D.3	Demande en véhicules	233
D.4	Simulation	234
E	Résultats de l’étude de sensibilité des coefficients pondérateurs	237
F	Résultats de l’étude de sensibilité de la mobilité	241
G	Tables des valeurs pour les scénarios Sa1, Sa2 et Sa3	245

Table des figures

1	Vue d'artiste de la structure CBL (<i>Chain Branch Leaf</i>) sur autoroute	27
1.1	Vision futuriste des systèmes de transport intelligents	33
1.2	Présentation des différentes architectures de communication V2V et V2I	34
1.3	Décomposition d'un protocole de routage	37
1.4	Schémas des stratégies de routage par relais multipoints et par inondation	38
	(a) Méthode des relais multipoints	38
	(b) Méthode par inondation	38
1.5	Statuts de nœuds d'un protocole de groupement d'un réseau MANET	39
1.6	Topologie hiérarchique de type colonne vertébrale au sein d'un réseau MANET	41
1.7	Modes de communication <i>unicast</i> , <i>broadcast</i> , <i>multicast</i> et <i>anycast</i>	41
1.8	Présentation du fonctionnement global des méthodes proactives	42
1.9	Présentation du fonctionnement global des méthodes réactives	42
1.10	Comparatif du trafic de routage généré entre OLSR et AODV	43
1.11	Présentation du fonctionnement des méthodes hybrides	44
1.12	Classification des informations propres au contexte véhiculaire	45
1.13	Restriction de recherche de route par la définition d'une zone géographique	46
1.14	Restriction géographique locale de la méthode PGB	47
1.15	Choix d'un nœud relais selon l'algorithme de routage glouton	47
1.16	Présentation de la méthode périmètre	48
1.17	Division topologique du réseau par le protocole GRP	48
1.18	Système d'ancre mis en œuvre par le protocole GSR	50
1.19	Variation de puissance en fonction du nombre moyen de voisins	51
	(a) Puissance de transmission faible	51
	(b) Puissance de transmission élevée	51
1.20	Exemple de l'utilisation du protocole VADD dans une intersection	53
2.1	Bus de communication dans les trains	60
2.2	Illustration sur une autoroute à trois voies d'un groupe de véhicules	61
2.3	Illustration de la création d'une chaîne dans un cas autoroutier	62
2.4	Temps de connexion <i>CT</i> en fonction de la vitesse relative et de la portée de transmission	64
	(a) Cas de deux véhicules en sens opposé de circulation	64
	(b) Cas de deux véhicules dans le même sens de circulation	64
2.5	Contenu de certaines variables internes des nœuds CBL	67
2.6	Procédure utilisée lors de la réception d'un message HELLO	71
2.7	Étapes de découverte et d'élection de nœud branche par des nœuds feuille	84
2.8	Choix d'un nœud branche relais par un nœud feuille dans le cas de deux nœuds branche candidats.	85
2.9	Étapes de rétablissement de la chaîne dans le cas d'un nœud branche amont dépassant son nœud branche aval.	86
2.10	Paramétrage du coefficient <i>C1</i> en fonction du pire cas de découverte du voisinage	88
2.11	Paramétrage du seuil <i>C3</i> en fonction du pire cas	89
2.12	Problème de création de chaînes parallèles suite à une rupture d'une liaison inter-branche	92
2.13	En-tête d'un message HELLO défini par le protocole OLSR	96
2.14	En-tête d'un paquet défini par le protocole OLSR	97

2.15	Nœuds et liens fonctionnels de CBL	98
2.16	Modes de communication supportés par le protocole GeoNetworking	98
	(a) Point à point	98
	(b) Point à multi-points	98
	(c) GeoAnycast	98
	(d) GeoBroadcastcast	98
2.17	En-tête d'un message proposé dans le cadre de l'utilisation du schéma CBL	99
2.18	Link code défini par le protocole OLSR	100
2.19	Exemple d'un message HELLO dans le cadre de l'utilisation du schéma CBL	101
3.1	Modélisations nécessaires à l'étude des performances du schéma CBL	106
3.2	Diagramme d'utilisation de MATLAB	109
3.3	Différentes cartes générées par les modèles de type <i>Street-RWP</i>	111
	(a) Carte du type Manhattan	111
	(b) Carte aléatoire	111
	(c) Carte issue du monde réel	111
3.4	Diagramme d'utilisation de SUMO	115
3.5	Densité de probabilité de la vitesse désirée pour les VL et les PL	116
3.6	Réseau A27 importé dans Netedit	117
3.7	Demande de véhicules issue d'une boucle électromagnétique	118
3.8	Demande de véhicules issue d'un radar	118
3.9	Puissance mesurée en environnement autoroutier et modèles associés	122
3.10	Modèle de propagation d'onde intégré par défaut sous OPNET Riverbed Modeler	123
3.11	Puissance moyenne reçue pour le modèle LOS/OLOS et le modèle Nakagami	123
3.12	Exemple de configuration OLOS	124
3.13	Algorithme CSMA/CA de la norme 802.11p	125
	(a) Mode <i>broadcast</i>	125
	(b) Mode <i>unicast</i>	125
3.14	Taxonomie des applications ITS dans les VANETs	127
3.15	Ordonnancement temporel des actions de sécurité dans les ITS	127
3.16	Analyse statistique des paramètres de simulation des articles référencés au chapitre 1	134
4.1	Nombre des nœuds feuille et branche en fonction du temps – Scénario S5, CBL Version 1	136
4.2	État de la structure formée par CBL au temps $t=500$ s – Scénario S5, CBL Version 1	137
4.3	Nombre de chaînes dans le réseau (métrique Ms2) – Scénario S5, CBL Version 1	137
	(a) En fonction du temps	137
	(b) En durée cumulée	137
4.4	État de la structure formée par CBL – Scénario S5, CBL Version 1	138
	(a) Chaînes au temps $t=124$ s	138
	(b) Zoom à $t=124$ s de la chaîne Est-Ouest aux abscisses $X \in \llbracket 1500, 3000 \rrbracket$	138
	(c) Zoom à $t=125$ s de la chaîne Est-Ouest aux abscisses $X \in \llbracket 1500, 3000 \rrbracket$	138
4.5	Ms3 pour le scénario S5 avec CBL version 1	138
4.6	Ms4 pour le scénario S5 avec CBL version 1	139
4.7	Ms5 pour le scénario S5 avec CBL version 1	140
4.8	Ms6 pour le scénario S5 avec CBL version 1	140
4.9	Ms7 pour le scénario S5 avec CBL version 1	141
4.10	Ms8 pour le scénario S5 avec CBL version 1	141
4.11	Ms9 pour le scénario S5 avec CBL version 1	141
4.12	Nombre et type des nœuds – Scénarios (S1 à S9), CBL version 1	142
4.13	État de la structure formée par CBL au temps $T=300$ s – Scénario S1	143
4.14	Histogramme de la durée cumulée (Ms2) – CBL version 1	144
	(a) Scénario S1	144
	(b) Scénario S4	144
	(c) Scénario S7	144
4.15	Histogramme de la durée cumulée (Ms2) – Scénario S6, CBL version 1 et 2	146
	(a) CBL version 1	146

(b) CBL version 2	146
4.16 Nombre et type des nœuds – Scénario S5, méthode des MPRs	148
4.17 Ms11 pour le scénario S5 avec le méthode des MPRs	148
4.18 Ms12 pour le scénario S5 avec le méthode des MPRs	149
4.19 Ms13 pour le scénario S5 avec le méthode des MPRs	149
4.20 Ms14 pour le scénario S5 avec le méthode des MPRs	150
4.21 Ms15 pour le scénario S5 avec le méthode des MPRs	150
4.22 Nombre et type des nœuds – Scénarios S1 à S9, méthode des MPRs	151
4.23 Mr1 pour le scénario S5 cas A avec CBL-OLSR	155
4.24 Mr2 et Mr3 pour le scénario S5 cas A avec CBL-OLSR	155
(a) Messages HELLO (Mr2)	155
(b) Messages <i>Topology Control</i> (TC) (Mr3)	155
4.25 Mr7 pour le scénario S5 cas A avec CBL-OLSR	156
4.28 Comparaison de la charge totale de trafic de routage – Scénario S5, CBL-OLSR et OLSR	157
4.26 Comparaison Mr2 pour le scénario S5 cas A avec CBL-OLSR et OLSR	158
4.27 Comparaison Mr3 pour le scénario S5 cas A avec <i>Chain Branch Leaf - Optimized Link State Routing protocol</i> (CBL-OLSR) et OLSR	158
4.29 Impact observé via Ms1 des coefficients pondérateurs sur la structure CBL pour le scénario S5 cas A avec CBL-OLSR	161
4.32 Comparaison de la charge totale de trafic de routage – Scénario S10 cas A, CBL-OLSR et OLSR	162
4.30 Comparaison Mr2 pour le scénario S10 cas A avec CBL-OLSR et OLSR	163
4.31 Comparaison Mr3 pour le scénario S10 cas A avec CBL-OLSR et OLSR	163
5.1 Ma1 pour le scénario Sa2	172
5.2 Ma2 pour le scénario Sa2	173
5.3 Ma3 pour le scénario Sa2	173
5.4 Ma4 pour le scénario Sa2	174
5.5 Ma5 pour le scénario Sa2	175
5.6 Ma6 pour le scénario Sa2	176
5.7 Ma7 pour le scénario Sa2	176
5.8 Courbe cumulée de la métrique Ma7 pour le scénario Sa2	177
5.9 Ma8 pour le scénario Sa2	177
5.10 Moyenne de Ma1 et Ma2 pour les scénarios Sa1 à Sa3	178
(a) Charge du réseau (Ma1)	178
(b) Débit du réseau (Ma2)	178
5.11 Moyenne de Ma3 et Ma4 pour les scénarios Sa1 à Sa3	178
(a) Trafic de routage envoyé (Ma3)	178
(b) Trafic de routage reçu (Ma4)	178
5.12 Moyenne de Ma5 et Ma6 pour les scénarios Sa1 à Sa3	179
(a) Taux de retransmission (Ma5)	179
(b) Nombre de sauts (Ma6)	179
5.13 Moyenne de Ma7 et Ma8 pour les scénarios Sa1 à Sa3	180
(a) Délai WLAN (Ma7)	180
(b) Délai d'accès au médium (Ma8)	180
5.14 Utilisation de la localisation distribuée coopérative	182
(a) Erreur du GPS	182
(b) Ecart type de l'erreur du GPS	182
5.15 Utilisation de la localisation distribuée coopérative	183
(a) Étape 1	183
(b) Étape 2	183
(c) Étape 3	183
5.16 Format des paquets applicatifs CMM	184
5.17 Format des paquets d'application DDGPS	185
5.18 Séquence d'envoi de messages applicatifs CMM-DDGPS	186

5.19	Ma7 pour le scénario S12 avec l'application de localisation coopérative	187
5.20	Ma8 pour le scénario S12 avec l'application de localisation coopérative	187
5.21	Ma9 pour le scénario S12 avec l'application de localisation coopérative	188
5.22	Ma10 pour le scénario S12 avec l'application de localisation coopérative	188
5.23	Ms5 pour le scénario S12	189
5.24	Ms5 pour le scénario S5	189
5.25	Concept de carte étendue	190
5.26	Pourcentage de découverte du voisinage	191
5.27	Ma7 pour le scénario S12 avec l'application de perception élargie	193
5.28	Ma8 pour le scénario S12 avec l'application de perception élargie	193
5.29	Ma9 pour le scénario S12 avec l'application de perception élargie	194
5.30	Ma10 pour le scénario S12 avec l'application de perception élargie	194
5.31	Proposition d'une structure hybride V2V et V2I : H-CBL	199
A.1	Algorithm applied when receiving a HELLO message	217
C.1	Fonctionnement du logiciel OPNET Riverbed Modeler	227
C.2	Paramétrage des noeuds communicants sous OPNET	228
C.3	Déploiement d'une application sous OPNET	229
C.4	Sélection des statistiques sous OPNET	229
C.5	Lancement d'une simulation sous OPNET	230
C.6	<i>Routing Manager</i> d'OPNET Riverbed Modeler	231
C.7	Modèle de processus de OLSR sous OPNET Riverbed Modeler	231
C.8	Modèle de processus de CBL sous OPNET Riverbed Modeler	231
D.1	Visualisation du réseau créé par NETCONVERT	234
D.2	Visualisation de la simulation générée par SUMO	235
E.1	Impact observé via Ms2 des coefficients pondérateurs sur la structure CBL pour le scénario S5 cas A avec CBL-OLSR	237
E.2	Impact observé via Ms5 des coefficients pondérateurs sur la structure CBL pour le scénario S5 cas A avec CBL-OLSR	238
E.3	Impact observé via Ms6 des coefficients pondérateurs sur la structure CBL pour le scénario S5 cas A avec CBL-OLSR	238
E.4	Impact observé via Ms7 des coefficients pondérateurs sur la structure CBL pour le scénario S5 cas A avec CBL-OLSR	239
F.1	Nombre et type des nœuds – Scénarios S10 à S13, cas A, CBL-OLSR	241

Liste des tableaux

1.1	Métriques de qualité de service	55
1.2	Protocoles de routage classiques de la littérature	56
1.3	Protocoles de routage de groupement et à QoS de la littérature	57
2.1	Exemple de table des voisins à un saut d'un nœud N_i	65
2.2	Valeur à l'initialisation du nœud de ses variables locales (à $t=T$)	66
2.3	Paramétrage par défaut des coefficients pondérateurs	87
2.4	Paramètres du protocole OLSR	96
2.5	Modification pour CBL du "message type" défini dans le protocole OLSR	97
2.6	Modifications proposées du link type dans le cadre de l'utilisation du schéma CBL	100
2.7	Codage de table des voisins à un saut d'un nœud	100
3.1	Simulateurs de communication utilisés dans la littérature	107
3.2	Simulateur de trafic routier microscopique utilisés dans la littérature	112
3.3	Approches de modélisation de la mobilité dans les VANETs	114
3.4	Paramètres utilisés pour le modèle Krauß pour les Véhicules légers (VL) et Poids lourds (PL)	116
3.5	Définition des scénarios et de la valeur de la demande de trafic routier	117
3.6	Scénarios et valeurs de densité de trafic	118
3.7	Nombre de nœuds nécessaires à chaque scénario sous Riverbed OPNET Modeler	119
3.8	Modélisation de la propagation des signaux dans les VANETs	121
3.9	Configuration des nœuds dans OPNET	122
3.10	Valeur des paramètres de la couche MAC du protocole 802.11p	126
3.11	Configuration des nœuds dans OPNET	126
3.12	Besoins en communication des applications de sécurité véhiculaires	130
3.13	Choix de simulation pour l'évaluation des protocoles de routage de groupement de la littérature	131
3.14	Choix de simulation pour l'évaluation des protocoles de <i>backbone</i> de groupement de la littérature	132
3.15	Choix de simulation pour l'évaluation des protocoles de routage à QoS de la littérature	133
4.1	Moyennes des métriques Ms1 à Ms9 sous Matlab – Scénarios S1 à S9, CBL version 1	142
4.2	Moyennes des métriques Ms1 à Ms9 sous Matlab – Scénarios S1 à S9, CBL version 2	145
4.3	Moyennes des métriques Ms10 à Ms15 sous Matlab – Scénarios S1 à S9, méthode des MPRs151	151
4.4	Paramètres des protocoles <i>Optimized Link State Routing protocol</i> (OLSR) et CBL-OLSR pour deux ensembles, cas A et cas B, d'intervalles temporels	152
4.5	Moyennes des métriques Ms1 à Ms3 et Ms5 à Ms7 sous OPNET – Scénarios S1 à S5, S7 et S8, CBL-OLSR	153
4.6	Moyennes des métriques Mr1 à M7 sous OPNET – Scénarios S1 à S5, S7 et S8, CBL-OLSR157	157
4.7	Moyennes des métriques Mr1 à M7 sous OPNET – Scénarios S1 à S5, S7 et S8, OLSR	159
4.8	Moyennes des métriques Ms1 à Ms3 et Ms5 à Ms7 sous OPNET – Scénarios S10 à S13, CBL-OLSR	162
4.9	Moyennes des métriques Mr1 à Mr7 sous OPNET – Scénarios S1 à S3 cas A, CBL-OLSR, modèle de propagation OLOS	165

5.1	Paramètres par défaut du protocole AODV	168
5.2	Paramètres par défaut du protocole DSR	169
5.3	Paramètres par défaut du protocole OLSR	169
5.4	Paramètres par défaut du protocole GRP	170
F.1	Moyennes des métriques Mr1 à M7 sous OPNET – Scénarios S10 à S13, CBL-OLSR . . .	241
F.2	Moyennes des métriques Mr1 à M7 sous OPNET – Scénarios S10 à S13, OLSR	243
G.1	Valeurs des métriques de performances pour le scénario Sa1 avec 10 noeuds sources	246
G.2	Valeurs des métriques de performances pour le scénario Sa2 avec 15 noeuds sources	247
G.3	Valeurs des métriques de performances pour le scénario Sa3 avec 20 noeuds sources	248

Glossaire

- A-STAR** *Anchor based Street and Traffic-Aware Routing.* 49, 50
- AC** *Access Category.* 126
- ACA** *Angle-based Clustering Algorithm.* 52
- ACK** *Acknowledgment.* 126
- AGF** *Advanced Greedy forwarding.* 53
- AH** *Antenna Height.* 55
- AODV** *Ad-hoc On Demand Distance Vector Protocol.* 3, 5, 43, 46, 108, 110, 167–169, 171–175, 178, 194, 198
- AQVA** *Adaptive Qos for Vehicular safety services.* 54, 55
- ARCEP** *Autorité de régulation des communications électroniques et des postes.* 25
- ARP-QD** *Adaptive Routing Protocol Based on QoS and Vehicular Density in Urban VANETs.* 55
- BAHG** *Backbone Assited Hop Greedy.* 54
- BBBR** *BackBone Broadcast Routing.* 49, 55
- BBNC** *Backbone Based routing with inter-flow Network Coding.* 49, 55
- BBR** *BackBone Routing.* 49, 53, 55
- BEDC** *Based on Euclidean distance clustering.* 52
- BW** *Bandwidth.* 55
- C-DRIVE** *clustering based on DIRection in Vehicular Environment.* 52, 108, 112
- CAM** *Cooperative Awareness Message.* 61, 63, 190
- CAR** *Connectivity Aware Routing.* 46, 50, 53
- CBL** *Chain Branch Leaf.* 3, 5, 9, 10, 13–17, 26–29, 59, 61–63, 65, 67–73, 75, 77, 79, 81, 83, 85, 87, 89–101, 106, 108, 109, 123, 124, 129, 135–157, 160, 161, 164, 165, 192, 194, 197–199, 230, 237–239
- CBL-OLSR** *Chain Branch Leaf - Optimized Link State Routing protocol.* 15, 17, 27, 28, 94, 97, 100, 129, 152–160, 162–164, 167, 170–175, 178, 179, 185, 188, 194, 195, 197–199, 230, 241–243
- CC** *Cluster Chain.* 48
- CDS-SVB** *Connected dominating Set for Stable Virtual Backbone.* 52, 108
- CMM** *Cooperative Map Matching.* 181–186, 188, 194
- CORSIM** *CORridor SIMulation.* 111
- CSMA** *Carrier Sense Multiple Access.* 50, 174, 179
- CSMA/CA** *Carrier Sense Multiple Access with Collision Avoidance.* 125, 199
- CT** *Connection Time.* 63, 64
- DBA-MAC** *Dynamic Backbone-Assisted Medium Access Control.* 55, 108
- DBC** *Density Based Clustering.* 50
- DBR** *Destination Based Routing.* 48, 53
- DCA** *Dynamic Clustering Algorithm.* 49

- DCEV** *Distributed Cluster formation for vanet based on End-to-End Relative mobility* . 49
- DDGPS** *Dynamic base station de base Differential GPS*. 181, 183–186, 188, 189, 194
- DEMN** *Decentralized Environmental Notification Message*. 61
- DENM** *Decentralized Environmental Notification Message*. 63
- DIR** *Direction Interdépartementale Des Routes*. 117
- DREAM** *Distance Routing Effect Algorithm for Mobility*. 46
- DSDV** *Destination-Sequenced Distance-Vector Routing*. 36, 37, 42, 110
- DSR** *Dynamic Source Routing*. 3, 5, 42, 110, 167–169, 171–175, 178, 194
- DSRC** *Dedicated Short Range Communication*. 25
- DTSG** *Dynamic Time-Stable Geocast*. 51
- EDCA** *Enhanced distributed coordination function*. 126
- EIRP** *Puissance isotrope rayonnée équivalente*. 125
- EPM** *Environmental Perception Message*. 190–192
- ETSI** *European Telecommunications Standards Institute*. 25, 97, 126, 128
- ETX** *Expected Transmission Count*. 54, 55
- FCD** *Floating Car Data*. 109
- FDM** *Freeway Mobility Model*. 110, 111
- FHWA** *Federal Highway Administration*. 109
- GeoAODV** *Geographic Ad-hoc On Demand Distance Vector Protocol*. 46
- GeoSVR** *Geographic Stateless VANET Routing*. 52
- GPS** *Global Position System*. 109, 110, 113, 180–183, 186
- GPSR** *Greedy Perimeter Stateless Routing for Wireless Networks*. 47, 52, 63
- GRP** *Geographic Routing Protocol*. 3, 5, 47, 108, 110, 167, 169–180, 194
- GSR** *Geographic Source Routing*. 49, 50
- GyTAR** *Greedy Traffic Aware Routing*. 50
- H-CBL** *Hybrid - Chain Branch Leaf*. 199
- HD** *Highest Degree*. 40
- I2V** *Infrastructure to Vehicle*. 128
- IDM** *Intelligent Driving Model*. 111
- IEEE** *Institute of Electrical and Electronics Engineers*. 3, 25, 28, 35, 61, 64, 108, 124, 126, 128, 152, 194, 198
- IP** *Internet Protocol*. 64, 108, 152, 187, 192, 227
- IRQV** *Intersection-based QoS Routing in Vehicular ad hoc networks*. 54
- ISO** *International Standardization Organization*. 40, 124, 227
- ITS** *Intelligent Transportation System*. 25, 26, 28, 33, 34, 127, 181
- ITSA** *Intelligent Transportation Society of America*. 25
- k-CONID** *k-hop CONnectivity ID*. 40
- LAR** *Location-Aided Routing*. 46
- LDM** *Local Dynamic Map*. 190
- LET** *Link Expiration Time*. 54
- LID** *Lowest ID*. 39, 40

- LOS** *Line of Sight*. 120, 121, 123
- LOUVRE** *Landmark Overlays for urban Vehicular Routing Environments*. 51
- LTE** *Long Term Evolution*. 61, 124
- MAC** *Medium Access Control*. 35, 36, 107, 125, 179, 227
- MANET** *Mobile Ad Hoc Network*. 34, 36, 38–40, 42, 45, 60, 110, 197, 198
- MATLAB** *MATrix LABoratory*. 108, 109, 135, 152, 153, 160, 197
- MBS** *Mobility-based Scheme*. 55
- MOBIC** *MObility Based metrIC*. 40
- MOCOPo** *Measuring and Modelling Congestion and Pollution*. 110
- MPDB** *Mobility Prediction-based Directed Broadcasting*. 52
- MPR** *Multipoint Relay*. 10, 37, 38, 55, 94–96, 99, 135, 137, 139, 141, 143, 145, 147–152, 156, 157, 164, 175, 230
- NetStream** *NETwork Simulator for TRaffic Efficiency And Mobility*. 112
- NGSim** *Next Generation Simulation*. 109, 110
- NoH** *Number of Hops*. 54
- NS** *Network Simulator*. 107, 108
- OBU** *On-Board Unit*. 34
- OFDM** *Orthogonal Frequency Division Multiple Access*. 124, 125
- OHNS** *Optimal Next Hop Selection*. 55, 110
- OLOS** *Obstructed Line of Sight*. 121–123
- OLSR** *Optimized Link State Routing protocol*. 3, 5, 17, 27–29, 38, 42, 43, 55, 94–98, 100, 108, 110, 152–154, 156, 157, 159, 160, 162, 164, 167, 169, 171–175, 178, 194, 197, 199, 229, 230, 242, 243
- OMNET++** *Objective Modular Network Testbed in C++*. 108
- OMNeT++** *Objective Modular Network Testbed in C++*. 107, 108
- OPNET** *Optimized Network Engineering Tools*. 108, 109, 122, 124, 126, 152–154, 161, 164, 167, 170, 171, 185, 186, 192, 197, 227–229
- OSI** *Open Systems Interconnection*. 36, 40, 54, 106, 124, 227
- OVB-CAST** *On-Demand Virtual Backbone BroadCAST Protocol*. 55
- PBC** *Position-Based Clustering*. 48, 53
- PDR** *Packet Delivery Ratio*. 129, 186
- PGB** *Preferred Group Broadcasting*. 46
- PL** *Poids lourds*. 17, 115–117, 136
- PPC** *Position-Based Clustering*. 112
- PSO-C** *Particle Swarm Optimization Cluster*. 52, 108
- QOLSR** *QoS - Optimized Link State Routing protocol*. 167, 169, 171, 172, 175, 179, 194
- QoS** *Quality of Service*. 37, 45, 53, 169
- QoS-CBL** *Quality of Service - Chain Branch Leaf*. 198
- QoSBee** *QoS Bee Routing Protocol*. 54
- RAQV** *Road Aware QoS Routing in Vanets*. 50, 54, 55
- ReSCUE** *Relatively Stable Clustering for Unbiased Environments*. 51
- RFC** *Request For Comments*. 152, 229
- RNG** *Relative Neighborhood Graph*. 47

- RPB-MD** *Relative Position Based Message Dissemination*. 53
- RREP** *Route Replies*. 42–44, 168, 171, 172, 198
- RREQ** *Route Requests*. 42–44, 168, 169, 171, 172, 198
- RSU** *Road Side Units*. 34, 49, 199
- RTK** *Real Time Kinematic*. 181
- RWP** *Random Way Point*. 110, 111
- SC** *Stable Clusters*. 49
- SCS** *Stable Clustering Scheme*. 51
- SNR** *Signal Noise Ratio*. 122
- SUMO** *Simulation of Urban MObility*. 28, 107, 108, 112, 113, 115, 117, 197, 233
- TC** *Topology Control*. 15, 62, 94–96, 100, 152, 154–159, 162–164, 172, 230, 243
- TCP** *Transmission Control Protocol*. 108, 152, 227
- TP** *Transmission Power*. 55
- TraNS** *Traffic and Network Simulation Environment*. 107
- TTL** *Time To Live*. 95, 98, 168, 171
- UDP** *User Datagram Protocol*. 108, 152, 227
- V2I** *Vehicle to Infrastructure*. 25, 34, 61, 128, 199
- V2V** *Vehicle to Vehicle*. 3, 5, 25, 26, 34, 61, 109, 120, 124, 128, 140, 199
- VADD** *Vehicle Assisted Data Delivery*. 52
- VANET** *Vehicular Ad Hoc Network*. 34–37, 45–50, 53, 54, 56, 60, 63, 67, 105–113, 119–122, 124, 135, 154, 156, 170, 197–199
- VEINS** *VEhicule In Network Simulation*. 108
- VL** *Véhicules légers*. 17, 115–118, 136
- VMaSC** *Vehicular Multi-hop algorithm for stable clustering cluster*. 52
- VOA** *VANETs' quality Of service by Adaptative protocol*. 54, 55
- VWCA** *Vehicular clustering based on the Weighted Clustering Algorithm*. 52, 108
- WBACA** *Weight Based Adaptive Clustering Algorithm* . 40
- WLAN** *Wireless Local Area Network*. 25, 156, 159, 179, 186, 187, 192, 243

Introduction

Contexte général de la thèse

La téléphonie mobile est entrée dans la vie de tous les jours dans les années 2000 et ne cesse de progresser depuis face à une demande croissante des consommateurs du “toujours plus, toujours plus vite”. Les technologies de communications sans fil sont largement répandues aujourd’hui : de la radio numérique aux bracelets connectés en bluetooth en passant par le téléphone cellulaire 4G ou encore aux systèmes LoRA et SigFox de l’internet des objets. L’objectif de ces technologies est de proposer toujours plus de débit, de bande passante et de connectivité. Aujourd’hui la technologie 5G nous promet des débits de télécommunication de plusieurs gigabits de données par seconde avec des latences très faibles. L’Autorité de régulation des communications électroniques et des postes (ARCEP) prévoit d’autoriser, à partir de juin 2018, les opérateurs français à tester cette nouvelle technologie afin de la déployer dès 2020. De l’autre côté de l’Atlantique, en janvier 2018, la société AT&T annonçait être en mesure de proposer de la 5G à partir de la fin 2018 [1]. Une autre technologie sans fil largement utilisée est le *Wireless Local Area Network* (WLAN) (Wi-Fi) qui s’appuie sur le standard *Institute of Electrical and Electronics Engineers* (IEEE) 802.11 [2]. Le débit proposé par cette technologie a également augmenté de façon exponentielle, passant de 1 Mbit/s en 1999 (norme 802.11b) à 866 Mbit/s en 2014 (dernière norme 802.11ac).

L’un des champs d’applications de ces réseaux de communications concerne le secteur automobile. La possibilité d’une communication inter-véhiculaire partagée avec le gestionnaire de voirie permet d’imaginer un nombre colossal d’applications. En outre, les systèmes d’aide à la conduite, de détection d’obstacles, d’anti-collision, de conduite en convoi sont autant d’applications envisagées pour réduire la congestion routière, le nombre d’accidents et la pollution, en limitant les variations brusques de changement de régime lors de la conduite. Toutes ces applications s’inscrivent dans le cadre des *Intelligent Transportation System* (ITS). De nombreux projets européens sont à l’étude concernant ces systèmes. Les projets SCOOP [3] et InterCor [4] en sont des exemples.

L’automatisation de la conduite va de paire avec l’amélioration des communications entre les entités composant le système (véhicules coopératifs, autoroute intelligente). Les travaux actuels de la littérature reposent en particulier sur la présence d’une infrastructure permettant des communications entre le(s) véhicule(s) et l’infrastructure (système *Vehicle to Infrastructure* (V2I)). Cependant, dans l’hypothèse d’une infrastructure défaillante, un verrou scientifique consiste à concevoir une stratégie de communication inter-véhiculaires (*Vehicle to Vehicle* (V2V)) qui permette, d’une part, les communications proches nécessaires à l’échange de variables et de messages entre véhicules autonomes voisins et, d’autre part, les communications lointaines, typiquement nécessaires à l’échange de messages d’alertes et plus largement la gestion du réseau routier.

Ces systèmes devront utiliser une ou plusieurs technologies permettant les communications sans fil. La forte utilisation des réseaux locaux sans fil (WLAN) a motivé de nombreux travaux de recherche sur le sujet des réseaux véhiculaires depuis les années 2000 [5]. Afin d’adapter les réseaux WLAN, développés initialement pour des réseaux peu mobiles, des évolutions ont été nécessaires pour répondre aux exigences des réseaux véhiculaires. Une normalisation a été proposée par l’*Intelligent Transportation Society of America* (ITS-A) nommée *Dedicated Short Range Communication* (DSRC) qui modifie le standard IEEE 802.11 par l’amendement 6 qui concerne le 802.11p [6]. Cet amendement permet de supporter des communications entre des nœuds dont l’éloignement peut atteindre une distance de 1000 mètres avec une vitesse de 200 km/h. Aux États-Unis, une bande de fréquence (5.850 - 5.925 GHz) a été réservée pour les communications véhiculaires en 1999. En Europe, l’*European Telecommunications Standards Institute* (ETSI) a affecté une bande de fréquence similaire (5,855 - 5,925 GHz) dès 1990. D’après l’article [7],

la NHTSA (*National Highway Traffic Safety Administration*) promeut l'usage de la norme WIFI IEEE 802.11p comme standard pour les communications sans fil à courte et moyenne portée, norme spécialement conçue pour les ITS. Néanmoins, l'article remarque que cette décision ne semble pas avoir été prise en compte par les poids lourds de l'industrie des télécommunications qui proposent des solutions avec des frais d'abonnement, s'appuyant sur des systèmes centralisés cellulaires avec des antennes-relais appelées *Cellular vehicle-to-everything* (C-V2X). Ces systèmes s'appuient actuellement sur les standards cellulaires 4G existants comme le LTE et son option LTE-Direct. Ils autorisent actuellement une communication directe à un saut d'un mobile à un mobile, mais après attribution d'une plage temps-fréquence par une antenne-relais. L'article souligne qu'une lutte fait rage actuellement sur le choix de la technologie à utiliser pour les communications véhiculaires. Ne sachant pas l'issue de ce combat, il est prudent de proposer des approches qui pourront être mises en œuvre, quelle que soit la technologie retenue.

Problématique de la thèse

La puissance des transmissions radio diminue en fonction de la distance, réduisant ainsi la portée maximale des communications. Deux véhicules voulant communiquer entre eux ne sont pas obligatoirement à portée radio l'un de l'autre. Le déploiement d'un réseau véhiculaire à grande échelle nécessite alors le relayage sur plusieurs sauts : des nœuds intermédiaires servent de nœuds de relais. Ainsi, la stratégie qui consiste à trouver l'ensemble des nœuds relais nécessaires à l'acheminement des données est appelée routage. Chaque nœud du réseau doit participer de manière coopérative pour router les messages échangés. Cependant, l'absence d'une centralisation des messages pour le cas des communications V2V rend les algorithmes de routage, pour les réseaux véhiculaires ad hoc distribués, plus complexes que dans le cas centralisé. Ces algorithmes nécessitent la prise en compte des changements continus de la topologie du réseau, du contexte spécifique des réseaux véhiculaires ainsi que des différentes perturbations subies par le médium de communication radio. De plus, compte tenu de la limitation des ressources de communications dans les réseaux ad hoc en termes de bande passante, ces algorithmes doivent réduire le trafic de routage nécessaire à leur strict fonctionnement. Nous focalisons nos travaux de thèse sur la fonction de routage dans les réseaux véhiculaires et nous nous intéressons à l'amélioration des performances du réseau, en réduisant le trafic de routage, dans le but de faire face aux problèmes de passage à l'échelle et de limitation des ressources radios.

Afin de répondre à ces enjeux, nous avons orienté nos travaux sur une méthode de hiérarchisation du réseau de communication via la formation de groupe de nœuds. Cette technique est largement déployée dans les réseaux ad hoc pour accroître les performances et assurer un bon passage à l'échelle. Dans une technique de groupement, certains nœuds du réseau assument des fonctions particulières de routage. L'envoi des messages peut être limité à cet ensemble de nœuds, évitant alors leur diffusion à tous les nœuds du réseau et économisant ainsi des ressources de communications. Dans ce contexte, nous nous sommes intéressés à l'élaboration d'un schéma de groupement nommé *Chain Branch Leaf* (CBL), adapté aux réseaux véhiculaires, indépendant de la technologie sans fil utilisée, et qui permet d'être implanté dans un protocole de routage existant pour supporter des applications des ITS.

Contributions de la thèse

Cette thèse comporte trois principales contributions.

La première contribution de la thèse concerne la proposition d'un schéma d'auto-organisation des réseaux véhiculaires nommé *Chain Branch Leaf* (CBL) (illustration en figure 1). Cette contribution majeure de la thèse est un algorithme de groupement distribué et coopératif qui s'adapte à la mobilité des véhicules, en particulier à leurs contraintes spatiales de mobilité imposées par la configuration des voies routières. *Chain Branch Leaf* (CBL), qui constitue une nouvelle approche du routage dans les réseaux véhiculaires, forme un réseau fédérateur constitué d'une suite ordonnée de véhicules appelée chaîne. En l'absence ou en complément d'une infrastructure, cette approche offre à la couche application les services d'une infrastructure virtuelle "stable dynamiquement".

- Une première version du schéma d'organisation CBL a été réalisé au cours de la thèse qui a mené à une publication en revue [8]. Une deuxième version de l'algorithme CBL a ensuite été formalisée qui corrige certains manquements de la première version, en particulier au moment de l'initialisation

de la structure, mais aussi lors de la gestion des chaînes en évitant leur rupture ou la formation de chaînes isolées suite à un dépassement de véhicules. Ces améliorations ont nécessité notamment l'ajout de seuils temporels ayant pour but de différer certaines décisions de routage et de réaliser un ordonnancement pertinent des conditions d'élection des nœuds de la structure. Une nouvelle métrique temporelle appelée *ChainTime* a également été introduite qui permet un renforcement de la stabilité temporelle de la structure formée par CBL de plus de 50%.

- Nous avons ensuite spécifié l'intégration du schéma de groupement CBL au sein du protocole de routage *Optimized Link State Routing protocol* (OLSR) bien connu des réseaux mobiles ad hoc. La mise en œuvre de cette intégration a alors donné lieu à la création du protocole *Chain Branch Leaf - Optimized Link State Routing protocol* (CBL-OLSR). Les principaux changements réalisés dans le protocole OLSR ont été l'adaptation des différents messages de routage et leur traitement à réception.
- Le schéma CBL comportant de nombreux paramètres intrinsèques nécessaires à son fonctionnement, une étude de sensibilité des performances structurelles de CBL à ces paramètres a été menée. Elle a montré leur faible impact.
- L'évaluation structurelle montre que CBL réduit de 90 à 20% le nombre de nœuds agissant comme relais dans le réseau, et l'évaluation système que CBL-OLSR diminue le trafic de routage induit au bénéfice du trafic applicatif de 3 à 17 fois par rapport à OLSR.

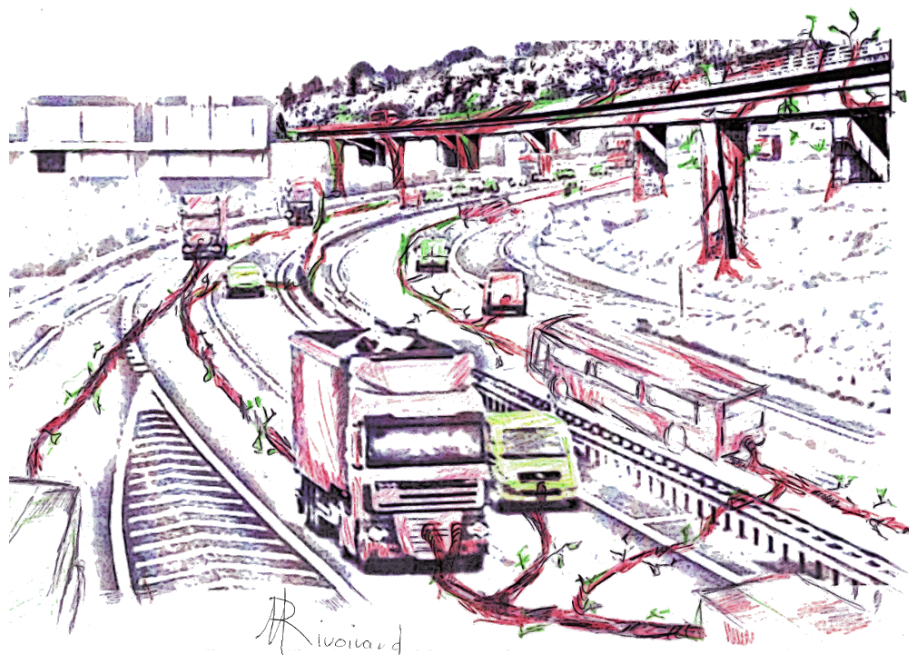


FIGURE 1 – Vue d'artiste de la structure CBL (*Chain Branch Leaf*) sur autoroute

La deuxième contribution de la thèse porte sur la formalisation d'un cahier des charges spécifiant les besoins de simulation nécessaires à l'évaluation des protocoles de routage dans le contexte véhiculaire. En effet, la difficulté des tests sur le terrain de systèmes à grande échelle, tels que le sont les réseaux véhiculaires, fait de la simulation informatique le moyen le plus répandu pour l'évaluation de ces derniers. Les besoins exprimés dans les études concernent :

- un simulateur de communication mettant en œuvre les processus relatifs aux normes de communication en vigueur pour les réseaux ad hoc sans fil ;
- un modèle de propagation d'ondes radio ;
- une modélisation de la mobilité des réseaux véhiculaires ;
- un modèle applicatif pour la définition des messages à transmettre dans le réseaux et celle des exigences de ces applications en termes de délais, débit et pertes ;
- des métriques d'évaluation de protocoles de routage.

À partir de ces besoins, les travaux suivants ont été menés en utilisant les nœuds de communication IEEE 802.11p d'OPNET Riverbed Modeler.

- L'établissement de notre cahier des charges a été réalisé à partir de l'analyse du cahier des charges des études de performances des protocoles de routage de la littérature pour les réseaux véhiculaires. Nous montrons que les ensembles formés par les différents simulateurs de communication et de trafic routier, les modèles applicatifs et de propagation des ondes utilisés dans les articles sont très vastes ; aucun consensus ne se dégage pour recommander l'usage d'un simulateur ou d'un modèle plutôt qu'un autre.
- Nous avons choisi comme simulateur de communication le logiciel MATLAB pour les évaluations structurelles et OPNET Riverbed Modeler pour les évaluations structurelles et système. Nous avons donc développé dans un premier temps un simulateur de trafic de routage simplifié pour l'étude de la construction de la structure CBL et celle d'OLSR sous MATLAB. Puis, nous avons mis en œuvre notre proposition CBL dans des processus d'OPNET Riverbed Modeler.
- Pour la modélisation de la mobilité des nœuds, le logiciel de simulation de trafic microscopique *Simulation of Urban MObility* (SUMO) est utilisé. La modélisation prend en compte différents contextes de mobilités. Des trajectoires de véhicules sont générées pour des densités différentes de véhicules dans des réseaux routiers fictifs simples d'une part, puis dans un réseau réel constitué d'une section de l'autoroute A27 reliant Lille à la Belgique et d'une section de la départementale perpendiculaire D90 d'autre part. Nous montrons alors que la structure CBL se forme de manière semblable sur ces différents réseaux routiers à partir du moment où la densité de véhicules est suffisante pour former une chaîne le long de l'infrastructure routière : un minimum de deux véhicules par sens de circulation à portée radio est nécessaire. Nous verrons que CBL permet le passage à l'échelle car plus la densité de véhicules augmente plus les performances structurelles s'améliorent.
- Enfin, deux modèles de propagation d'ondes sont employés. Le premier est le modèle en espace libre basé uniquement sur la distance entre le nœud émetteur et récepteur. Le second, que nous avons intégré sous OPNET, s'inspire d'un modèle de la littérature qui fut construit à partir d'une campagne de mesure sur autoroute en Suède [9]. Ce modèle distingue les ondes radio reçues en ligne directe de celles partiellement obstruées par un véhicule voisin. Les résultats obtenus sur la formation de la structure CBL avec ces deux modèles ne montrent pas de différences notables.

La troisième contribution de la thèse concerne trois évaluations qui intègrent différents modèles de trafic applicatif dans le but de mesurer les performances système de CBL-OLSR.

- La première évaluation compare notre proposition à cinq protocoles de la littérature avec un même trafic applicatif. Le trafic applicatif choisi a un débit conforme aux recommandations de l'IEEE cités dans [10]. Il correspond à l'envoi d'un message de 300 octets à une fréquence de 10 Hz. Nous montrons que CBL-OLSR, en limitant le nombre de nœuds relais et en diminuant la charge de routage, offre un service de communication qui peut convenir aux applications des ITS. Les taux de retransmission de CBL-OLSR sont inférieurs à 0,93 en présence de 15 nœuds émetteurs et les délais sont en dessous de 200 ms 90% du temps.
- La deuxième évaluation s'intéresse au trafic applicatif nécessaire à la mise en œuvre avec CBL-OLSR de l'application d'égo-localisation des algorithmes de la référence [11]. Selon [11], ces algorithmes ont montré une amélioration de la précision de la localisation des véhicules en fonction de la densité des véhicules communicants voisins avec l'usage conjoint d'un système GPS et de la cartographie de l'infrastructure routière. Nous avons donc analysé ces algorithmes et le trafic nécessaire à leur fonctionnement et les avons traduits en flux de trafic applicatif en tenant compte des particularités structurelles du schéma CBL. Dans l'hypothèse d'une période de transmission des messages de 200 ms cadencée sur la fréquence de 5GHz de rafraîchissement des données de localisation d'un GPS haut de gamme, le taux de pertes obtenu est en moyenne de 6% et les délais de transmission sont inférieurs à 50 ms avec une technologie IEEE 802.11p. Il reste donc du temps aux applications d'égo-localisation pour réaliser leurs traitements de correction.
- La troisième et dernière évaluation s'appuie sur le trafic applicatif d'une application de perception élargie. Ce type d'application défini dans la note d'orientations stratégiques pour l'action publique [12] permettra aux véhicules communicants d'améliorer leur connaissance de l'ensemble des

obstacles potentiels de leur environnement (véhicules non connectés, obstacles fixes...) à partir des informations de perception des véhicules communicants voisins. Comme précédemment, un travail d'analyse des besoins en communication et de leur traduction en flux de trafic applicatif a été mené en tenant compte des particularités structurelles du schéma CBL. Les résultats montrent, pour une fréquence de 5 Hz, un taux de perte de moins de 10% et des délais de 150 ms avec une technologie IEEE 802.11p.

Organisation du manuscrit

Dans ces quelques pages, nous avons décrit le contexte dans lequel s'inscrit cette thèse : les protocoles de routage adaptés aux communications véhiculaires ad hoc. Le reste de ce mémoire est composé de deux parties et de cinq chapitres.

La première partie composée de deux chapitres a pour objectif de décrire notre proposition d'organisation du réseau CBL. Dans le premier chapitre, nous présenterons les notions des réseaux de communication et en particulier les spécificités des réseaux véhiculaires. Nous décrirons le fonctionnement des protocoles de routage dans les réseaux mobiles ad hoc et les différentes stratégies mises en œuvre par les protocoles de routage adaptés aux réseaux véhiculaires. Le deuxième chapitre est consacré à la description de notre proposition d'organisation du réseau : CBL. Nous détaillerons dans ce chapitre les différentes définitions et métriques dédiées au schéma CBL ainsi que les algorithmes régissant le fonctionnement de ce schéma. Nous spécifierons également la mise en œuvre du schéma de groupement CBL au sein du protocole de routage OLSR.

La deuxième partie composée des trois derniers chapitres permet d'analyser les performances de notre proposition sous différents aspects. Dans le troisième chapitre, nous proposerons un cahier des charges permettant l'évaluation d'un protocole de routage dans le contexte véhiculaire. Ce chapitre comportera un état de l'art des différentes études de performances réalisées dans la littérature. Le quatrième chapitre présentera une analyse structurelle du protocole OLSR implémentant le schéma de groupement CBL comparativement à ce protocole sans le schéma CBL. Une première étude sera menée sous Matlab puis sous OPNET Riverbed Modeler. La sensibilité de la structure formée par CBL à la mobilité des nœuds et au modèle de propagation des ondes fera l'objet d'une étude. Le cinquième chapitre consiste à l'analyse système de CBL comparativement à d'autres protocoles de routage de la littérature et à l'étude de deux applications de sécurité pour les réseaux véhiculaires : la localisation coopérative distribuée et la perception élargie.

Nous présenterons nos conclusions dans le dernier chapitre de ce mémoire. Nous y résumerons les contributions de cette thèse, et nous présenterons des pistes à explorer pour compléter et étendre la portée de nos travaux.

Première partie

**Vers un protocole de routage
véhiculaire adaptatif : CBL**

Chapitre 1

Protocoles de routage dans les réseaux mobiles ad hoc

Sommaire

1.1	Introduction aux réseaux de communication véhiculaires	33
1.1.1	Contraintes physiques inhérentes au médium de communication	35
1.1.2	Contraintes imposées par le contexte routier	35
1.1.3	Passage à l'échelle	35
1.2	Protocoles de routage pour les réseaux mobiles ad hoc	36
1.2.1	Introduction	36
1.2.2	Établissement des liens logiques du réseau	37
1.2.3	Modes de communication	40
1.2.4	Recherche de route par dissémination de requêtes	42
1.3	Stratégies de routage pour le contexte véhiculaire	45
1.3.1	Stratégies et informations contextuelles propres aux systèmes routiers	45
1.3.2	Stratégies de routage et métriques de qualité de service	53
1.4	Conclusion	56

1.1 Introduction aux réseaux de communication véhiculaires

Les systèmes de transports intelligents, *Intelligent Transportation System (ITS)*, intègrent de nouvelles technologies de l'information et de la communication aux applications du domaine des transports pour les améliorer. Actuellement, le domaine des ITS connaît une rapide transformation avec de forts enjeux économiques et politiques en vue de construire la ville intelligente de demain. Cette ville intégrera la route de cinquième génération où chaque entité du système sera connectée et intelligente (figure 1.1).

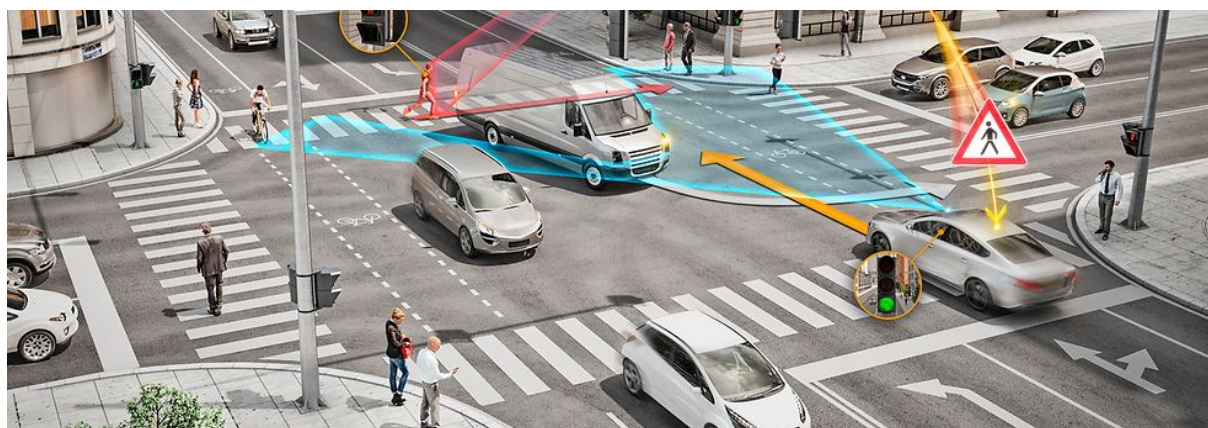


FIGURE 1.1 – Vision futuriste des systèmes de transport intelligents - source[13]

Les réseaux de communication véhiculaires, *Vehicular Ad Hoc Networks* (VANETs), font partie des ITS. Ils sont un cas particulier des réseaux mobiles ad hoc (*Mobile Ad Hoc Network* (MANET)). Leurs nœuds mobiles, aussi appelés *On-Board Unit* (OBU), sont embarqués dans des véhicules qui se déplacent au sein d'un réseau routier. Ils subissent donc les mêmes contraintes de mobilité que leur véhicule hôte, contraintes provenant de la configuration du réseau routier, de ses règles de circulation et des caractéristiques du trafic routier. Les études sur le domaine des VANETs sont nombreuses [14, 15, 16, 17, 18, 19, 20]. Elles considèrent les VANETs comme des *Mobile Ad Hoc Networks* (MANETs) à forte mobilité, sans prendre davantage en compte les contraintes de leur environnement.

La terminologie VANETs regroupe trois architectures différentes, illustrées en figure 1.2.

L'architecture de communication entre véhicules et infrastructure, *Vehicle to Infrastructure* (V2I), utilise des unités de bord de route, appelées *Road Side Units* (RSUs), déployées en bord de chaussée à intervalles réguliers. Ces équipements réalisent l'établissement des communications entre l'infrastructure et les véhicules. C'est une architecture de réseau à gestion centralisée, administrée par le gestionnaire de voirie qui planifie et optimise le trafic de communication entre les équipements connectés. Le coût de déploiement des RSUs d'une architecture *Vehicle to Infrastructure* (V2I) est non négligeable. Leur coût d'installation est estimé à environ 5000\$ par borne selon [21], auxquels il faut ajouter des coûts de maintenance tout au long de leur cycle de vie. Si d'un point de vue des communications, une architecture centralisée peut sembler plus simple à mettre en œuvre et à exploiter, la gestion de l'obsolescence des équipements sur des kilomètres de voirie, l'interopérabilité des matériels d'anciennes générations avec d'autres plus récents, la compatibilité entre les diverses applications des ITS, et la continuité du service de communication V2I en cas de panne de RSUs sont des enjeux logistiques pour maintenir le bon fonctionnement de cette architecture.

L'architecture de communication inter-véhiculaire, *Vehicle to Vehicle* (V2V), repose sur les liens de communication entre les OBUs. Ces OBUs collaborent de manière décentralisée pour former un réseau ad hoc. L'architecture *Vehicle to Vehicle* (V2V) est ainsi modulaire et flexible. Elle peut supporter la panne d'un nœud du réseau, mais elle nécessite toutefois un nombre de nœuds suffisant à portée de communication pour que des communications puissent s'établir. Notons que ce nombre de nœuds correspond au taux de pénétration du système de communication dans le trafic routier. D'autre part, la mise en œuvre de l'architecture V2V fait appel à des algorithmes de routage distribués dont les performances sont, par nature, plus complexes à prédire par rapport à ceux utilisés dans une architecture centralisée. Enfin, la solution V2V, à l'image de la solution précédente V2I, sera confrontée aux évolutions technologiques et, par conséquent, à l'organisation des mises à niveau d'un parc de véhicules communicants technologiquement hétérogènes.

La troisième architecture, appelée hybride, associe à la fois les systèmes V2I et V2V. Dans ce mode de fonctionnement, les deux systèmes devront se compléter et coopérer afin d'assurer une continuité de service. En particulier, le système V2V pourra être une alternative dans les zones non couvertes par les RSU.

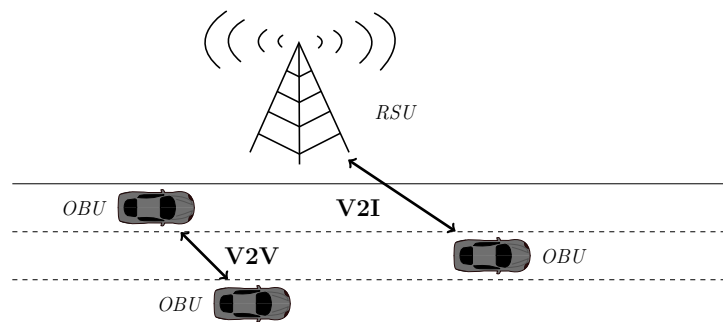


FIGURE 1.2 – Présentation des différentes architectures de communication V2V et V2I

Dans le contexte de cette thèse, nous nous intéressons plus particulièrement aux communications V2V entre nœuds mobiles communicants sans l'intervention d'éléments d'infrastructure. Dans ce modèle, les nœuds sont soumis à des contraintes qui, selon la configuration des routes et la densité du trafic routier,

n'auront pas le même impact sur les communications. Les contraintes sur les réseaux de communication véhiculaires ad hoc sont de trois ordres :

- les contraintes physiques imposées par le médium de communication (section 1.1.1) ;
- celles imposées par le contexte routier (section 1.1.2) ;
- celles auxquelles l'auto-organisation du réseau sera confrontée lors du passage à l'échelle (section 1.1.3).

1.1.1 Contraintes physiques inhérentes au médium de communication

La propagation des ondes radio émises sur un médium est soumise à des phénomènes d'atténuation et de distorsion d'une part, mais aussi de multi-trajets inhérents aux réflexions, réfractions et diffractions sur les obstacles rencontrés par le signal dans des environnements de propagation très variés : urbains, qui comptent de nombreux bâtiments et intersections routières, péri-urbains et autoroutiers, pour lesquels l'infrastructure apporte moins d'obstacles. Ces phénomènes physiques, parfois défavorables à l'établissement des liens de communication, engendrent une dégradation de la qualité et de la puissance du signal radio et sont des causes de pertes de paquets. L'affaiblissement de la puissance du signal est d'autre part responsable d'une portée radio limitée géographiquement à quelques centaines de mètres (de 300 à 1000 mètres pour une technologie *Institute of Electrical and Electronics Engineers* (IEEE) 802.11p [14]).

L'usage du médium de communication pour des applications véhiculaires nécessite l'adoption d'une technologie radio dont la couche *Medium Access Control* (MAC) assure, d'une part le partage des ressources de communication limitées en termes de débit et de bande passante et, d'autre part les tentatives d'accès au médium de communication parfois concurrentes aux différents nœuds du réseau. Ainsi, selon le nombre de nœuds à portée radio et le nombre de demandes d'accès, les conditions de propagation des signaux précédemment cités, les délais de transmission et les pertes de données plus ou moins importants, peuvent être observés dans un VANET.

1.1.2 Contraintes imposées par le contexte routier

Les nœuds des VANETs sont caractérisés par leur forte mobilité. Leurs vitesses varient de 0 km/h (véhicule à l'arrêt dans des embouteillages ou en réponse au code de la route et à la signalisation, par exemple) à parfois plus de 130 km/h en France (sur autoroute avec dépassement de la vitesse limite autorisée) ou sur les autoroutes allemandes qui n'ont pas de limitation de vitesse. Les directions de circulations peuvent être sécantes (aux intersections) ou opposées (en urbain, péri-urbain et sur autoroutes à double sens de circulation). Mais la mobilité des nœuds respecte les règles de circulation imposées par les infrastructures aux véhicules qui les embarquent. Cette mobilité est à l'origine de durées de connexions parfois très courtes entre les véhicules communicants. Ainsi, les nœuds de deux véhicules qui roulent à 130 km/h sur l'autoroute en sens inverse et dont la portée maximale des interfaces de communication est de 500 m n'échangeront des informations que durant au maximum 15 secondes [14, 19, 22, 23].

Les variations de cette mobilité induites par celles de la vitesse au cours du temps et des interactions entre les véhicules [14, 19, 22, 24] sont également responsables des fortes variations topologiques du réseau de communications. Ces variations proviennent également de celles de la densité de trafic et des différents états de vigilance et habitudes de conduite des conducteurs [14, 19, 22, 25, 26].

La variation de densité de véhicules au cours du temps est importante et dépend notamment du type de route et de l'état du trafic. En ville, un véhicule pourra la plupart du temps communiquer sans intermédiaire avec une dizaine voire une centaine de nœuds voisins, avec une portée maximale des interfaces de communication de l'ordre de 500 m. En rase campagne, le manque de véhicules voisins conduira parfois à une impossibilité de communication [14, 25]. Ainsi, des phénomènes de congestion de paquets sont susceptibles de se produire dans le cas d'une densité élevée de véhicules, en raison des ressources de communication limitées du réseau. Une trop faible densité de véhicules engendrera des connexions intermittentes et sporadiques qui pourront empêcher l'information d'être délivrée [19].

1.1.3 Passage à l'échelle

Le nombre de capteurs à bord des véhicules et la pénétration des systèmes communicants au sein du marché automobile vont augmenter d'années en années. De fait, la quantité d'informations collectées et

échangées au sein d'un VANET suivra la même tendance. La conception, la modélisation et l'évaluation par simulation d'un protocole de routage adapté au VANET doit inclure des scénarios permettant d'anticiper le passage à l'échelle en terme de nombre de nœuds connectés et de flux de données échangées, d'évolution des contraintes physiques des technologies de télécommunication en terme de portée, et enfin de l'impact du terrain routier sur la connectivité du système global.

1.2 Protocoles de routage pour les réseaux mobiles ad hoc

1.2.1 Introduction

Le routage est un mécanisme de la couche 3 (couche réseau) du modèle de référence *Open Systems Interconnection* (OSI). Il est en charge de la génération et de la gestion de chemins (ou routes) entre un nœud expéditeur et un ou plusieurs nœuds destinataires délocalisés d'un système de communication. Ces routes sont utilisées pour l'acheminement des informations d'un nœud expéditeur à leur(s) destinataire(s). Une route est un lien virtuel, ou logique, entre des nœuds de communication. Elle identifie une liste de nœuds en capacité de relayer l'information entre au moins deux nœuds distants. Le rôle du protocole de routage est essentiel dès qu'il y a interconnexion de réseaux de communication et échanges d'informations entre des nœuds de ces réseaux. Il est crucial dans les réseaux mobiles ad hoc sans fil, ou MANETs. Dans ces réseaux, les nœuds communicants ne sont pas toujours à portée directe de communication, la portée de transmission maximale étant limitée d'une part, par la puissance de transmission des nœuds et d'autre part, par l'affaiblissement du signal en fonction de la distance (section 3.6.2). Ainsi, lorsqu'un nœud destinataire n'est plus en liaison radio directe du nœud émetteur, l'existence d'une ou de plusieurs routes entre ces nœuds permet de supporter les besoins en communication d'applications distribuées. Une difficulté particulière des réseaux MANETs concerne la mobilité des nœuds. Selon les caractéristiques de cette mobilité, la topologie des nœuds d'un réseau MANET évoluera plus ou moins rapidement et les routes entre les nœuds communicants devront être maintenues ou réactualisées en conséquence.

Le protocole de routage le plus élémentaire est le routage par inondation (ou *blind flooding*) dans lequel chaque nœud du réseau retransmet systématiquement tous les paquets reçus. Ainsi, les paquets de données émis par un nœud seront dans un premier temps retransmis par tous les nœuds de son voisinage à un saut (*i.e.* à portée directe de communication), puis par les voisins de ces nœuds, et ainsi de suite jusqu'à ce que, de proche en proche, tous les nœuds destinataires aient reçu le paquet. Néanmoins, les nœuds du réseau partagent les ressources d'un même médium de communication. Pour éviter les collisions et les interférences entre des transmissions concurrentes, l'envoi d'un paquet par un nœud source impose l'arrêt des transmissions de ses nœuds voisins au niveau de la couche MAC. La méthode de routage par inondation offre alors des performances acceptables pour un réseau de communication constitué d'un nombre très faible de nœuds qui ont relativement peu d'informations à transmettre et une fréquence faible de transmission de données. Dès lors que le nombre de nœuds du réseau augmente, ou le nombre d'informations à transmettre, ou encore la fréquence de transmission de données, le nombre de retransmissions de paquets s'accroît fortement, pouvant mener à un phénomène appelé "tempête de diffusion" (ou *broadcast storm*). Les ressources du médium de communication arrivent alors à saturation et plus aucune transmission n'est possible. Les services de communication du réseau sont défaillants. Si pour un nombre faible de nœuds "peu loquaces", le routage par inondation peut fonctionner. Notons que la majorité des paquets retransmis sont inutiles dans la mesure où ils ne permettront pas d'atteindre le ou les nœuds destinataires. Le *blind flooding* qui surconsomme les ressources limitées en bande passante du réseau, n'est donc pas adapté aux réseaux ad hoc sans fil qui comptent un grand nombre de nœuds, tels que les réseaux véhiculaires.

Historiquement, les protocoles de routage pour les MANETs sont une adaptation de ceux de la communauté des réseaux filaires. Le protocole *Destination-Sequenced Distance-Vector Routing (DSDV)* [27], développé en 1994, est l'un des premiers protocoles adaptés aux MANETs. Il s'appuie sur l'usage de tables de routage, c'est-à-dire de listes contenant les informations de routes entre nœuds. Chaque route inscrite dans une table permet d'atteindre un nœud du réseau. Le protocole DSDV fonctionne ainsi. Chaque nœud possède sa propre table de routage contenant la liste des routes dont il a connaissance. Il la partage périodiquement par diffusion au sein du réseau au moyen de messages de routage. Tout nœud du réseau, à réception d'un message de routage, compare le contenu de sa table de routage à la liste de routes reçues,

puis met à jour sa table, de sorte à ne conserver que les routes les plus récentes offrant le plus faible nombre de nœuds relais vers chaque destination. À son tour, il diffuse sa table de routage ainsi modifiée. Lorsqu'un nœud a un paquet applicatif à transmettre, il cherche, dans sa table de routage, l'ensemble des nœuds relais qui lui permet d'atteindre le nœud destinataire. Les études [28, 29, 30] ont montré que ce protocole n'était pas optimal. Le temps de convergence de l'ensemble des tables de routage vers l'optimum peut s'avérer très long, voire impossible. Il augmente par ailleurs avec le nombre de nœuds du réseau. Dans le cas de nœuds de communication très mobiles, la topologie du réseau est sans cesse modifiée, et les tables de routage ne convergent pas vers l'optimum individuel. Par conséquent, ce protocole semble également inadapté aux VANETs.

L'objectif d'un protocole de routage est de permettre à chaque nœud du réseau d'envoyer des paquets à n'importe quel(s) nœud(s) destinataire(s), tout en évitant le phénomène de *broadcast storm*. Ceci implique qu'un protocole limite le nombre de messages de routage nécessaires à son fonctionnement, ces derniers utilisant des ressources de communication au détriment du trafic applicatif. Pour répondre à cet enjeu, chaque protocole de routage met en œuvre un ensemble de stratégies dont l'objectif est d'optimiser la dissémination des messages de routage et des paquets applicatifs, ce que ne fait pas la méthode *blind flooding*.

Un protocole de routage peut être décomposé en trois phases distinctes, illustrées en figure 1.3. Ces phases sont celles de l'établissement des liens logiques du réseau (section 1.2.2), du choix d'un mode de communication (section 1.2.3) et de la recherche de routes (section 1.2.4). Elles utilisent parfois des métriques spécifiques liées au contexte environnemental des nœuds du réseau, telle que la position géographique des nœuds, la densité des nœuds, mais aussi des métriques de qualité de service – *Quality of Service* (QoS).

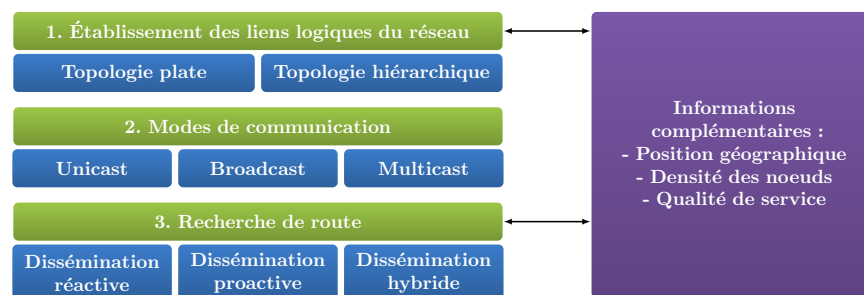


FIGURE 1.3 – Décomposition d'un protocole de routage

1.2.2 Établissement des liens logiques du réseau

Hormis dans les protocoles par inondation et certains protocoles réactifs, afin d'avoir une connaissance des nœuds l'environnant, tout nœud du réseau part, dès son initialisation, à la découverte de son voisinage via l'émission périodique de messages HELLO (également appelés *beacons*). Les réponses des nœuds de son voisinage, dénommés ses voisins, lui permettent de les répertorier dans sa table de routage. Une fois ses voisins connus, le nœud est capable d'établir des routes qui serviront à l'acheminement des paquets vers un (ou des) nœud(s) du réseau. Cette étape de découverte du voisinage est à la base de la formation de la topologie du réseau. Cette topologie peut être horizontale ou hiérarchique.

Une topologie horizontale (*flat topology*) ne privilégie aucun nœud : tout nœud du réseau a le même rôle et les mêmes fonctions que ses nœuds voisins. Cette topologie est celle de la plupart des protocoles de routage. C'est par exemple celle du protocole DSDV précédemment présenté en section 1.2.1.

Une topologie hiérarchique (*hierarchical topology*) attribue des rôles différents à certains nœuds, construisant ainsi des liens logiques hiérarchiques entre les nœuds de communication au-dessus de la topologie physique. Nous présentons dans les sous-sections suivantes une première topologie hiérarchique construite autour de nœuds *Multipoint Relays* (MPRs), une deuxième autour de groupes de nœuds (*clusters*) et une troisième autour d'une structure fédératrice (*backbone*).

1.2.2.1 Topologie hiérarchique construite à partir de nœuds relais multipoints

La notion de nœuds relais multipoints, MPRs, a été développée pour le protocole *Optimized Link State Routing protocol* (OLSR) [31] (section 2.5.1). Ce protocole crée une topologie hiérarchique autour de deux catégories de nœuds : les nœuds élus MPRs et les nœuds non-MPRs. En mode *broadcast* (section 1.2.3), contrairement aux nœuds non-MPRs, les nœuds élus MPR ont pour fonction de relayer à destination de tous les nœuds du réseau, aussi bien les messages de routage (contenant les informations de routes) que les messages applicatifs. Chaque nœud du réseau élit un ensemble minimal de nœuds relais MPRs parmi les nœuds de son voisinage à un saut qui lui permette d'atteindre tout nœud voisin à deux sauts. L'ensemble de tous les MPRs du réseau constitue un ensemble dominant.

La figure 1.4 illustre les mécanismes de retransmission mis en jeu, dans le cas d'un routage par MPRs et dans celui d'un routage par inondation, après la diffusion par un nœud source (représenté en rouge) d'un paquet à destination de tous les nœuds du réseau. Dans le cas d'un routage par MPR, à réception du paquet diffusé, seuls les voisins à un saut qui ont été élus MPR par le nœud source (représentés en vert) rediffusent le paquet. De même, parmi les voisins à deux sauts du message rediffusé, seuls les MPRs de chacun des nœuds ayant rediffusé le paquet (représentés en bleu), le retransmettent, et ainsi de suite jusqu'à ce que tous les nœuds du réseau aient reçu le paquet initial. Dans le cas d'un routage par inondation, tous les nœuds du réseau rediffusent le paquet. Dans cet exemple (figure 1.4a), seuls quatre nœuds relaient le paquet avec la méthode des MPRs lors de la dernière étape de retransmission, contre douze nœuds avec un routage par inondation (figure 1.4b). Le nombre de nœuds total qui relaient le paquet dans les trois étapes est de 9 pour la méthode des MPRs et de 21 pour la méthode *blind flooding*. Le routage par MPRs réduit de plus de moitié l'utilisation de la bande passante. En conclusion, en ne donnant les droits de retransmission qu'à une catégorie de nœuds, un protocole de routage utilisant la méthode des MPRs optimise l'usage des ressources de communication.

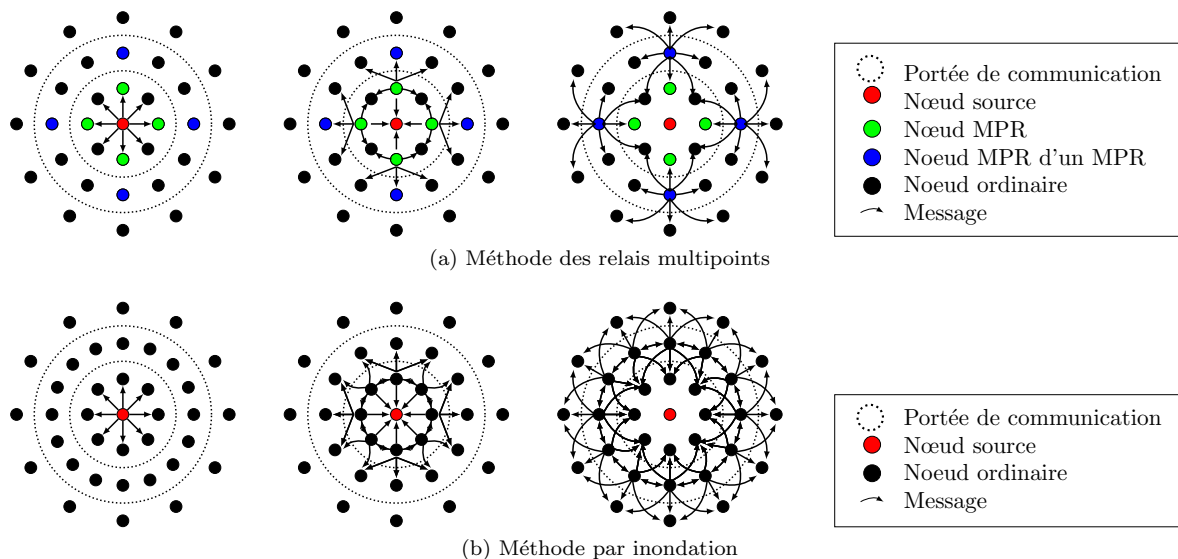


FIGURE 1.4 – Schémas des stratégies de routage par relais multipoints et par inondation

1.2.2.2 Topologie hiérarchique construite à partir de groupes de nœuds

Les méthodes de répartition de nœuds dans des groupes (*clustering*) réalisent un découpage virtuel du réseau de communication conformément à un ensemble de règles définies par le protocole de *clustering*. Lorsqu'elles sont utilisées dans les réseaux MANETs, ces méthodes permettent de contraindre l'ampleur de la diffusion des paquets de données à un ou plusieurs groupes particuliers, de sorte à réduire le nombre de retransmissions inutiles. Le risque de collisions sur le médium est alors réduit, la bande passante est préservée au profit des applications.

La formation des groupes est réalisée autour d'un nœud généralement appelé responsable de groupe (ou *cluster-head*). D'autres nœuds particuliers peuvent également être définis par ces protocoles. Chaque

type de nœud possède alors des propriétés et fonctions différentes, dédiées à certaines tâches protocolaires. La référence [32] répertorie ainsi quatre types de nœuds (figure 1.5) :

- les nœuds ordinaires, qui sont chacun membre d'un seul groupe ;
- les responsables d'un groupe (ou *cluster-heads*), qui sont les responsables de groupe auxquels les nœuds ordinaires sont rattachés ;
- les nœuds passerelles (ou *gateway*), qui sont membres de plusieurs groupes ; ils permettent les transferts d'informations entre les groupes dont ils sont membres ;
- les nœuds indéfinis, qui sont ceux ne possédant, à l'instant considéré, aucun statut (ordinaire, responsable de groupe ou passerelle), ni rattachement à un groupe.

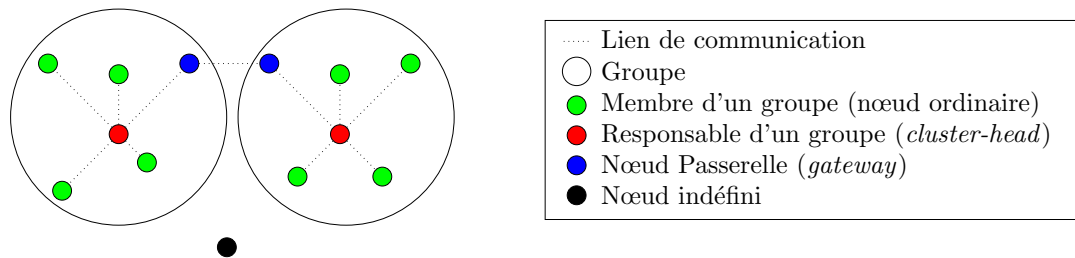


FIGURE 1.5 – Statuts de nœuds d'un protocole de groupement d'un réseau MANET

Les protocoles de *clustering* pour les réseaux MANETs font l'objet de nombreux travaux de recherche. Les études [33, 34, 35] en dressent un état de l'art. La littérature a cherché à classer ces protocoles selon divers critères qui ont l'intérêt de montrer la multitude des solutions conçues. Ces classifications peuvent ainsi reposer sur :

- les critères et métriques utilisés pour la formation et la maintenance des groupes et l'élection de leur responsable ;
- le nombre de sauts qui séparent un nœud ordinaire de son responsable de groupe ;
- la nature dominante des groupes de nœuds. Un groupe peut être un ensemble dominant (*Dominating Set*), dominant connecté (*Connected Dominating Set*) ou dominant faiblement connecté (*Weakly Connected Dominating Set*). Un groupe de nœuds est dominant (ou absorbant) dans un réseau si tout nœud du réseau fait partie de ce groupe ou est à un saut d'un nœud qui en fait partie. Un groupe est dit dominant connecté si le groupe est dominant et si tout nœud du groupe peut atteindre n'importe quel nœud du groupe par une route dont les nœuds intermédiaires sont dans le groupe. Un ensemble dominant faiblement connecté est un ensemble dominant connecté ayant le plus faible nombre de nœuds parmi tous les ensembles dominants connectés du réseau.
- la nature disjointe ou superposable des groupes : des groupes sont disjoints lorsqu'il n'existe aucun nœud qui soit membre de deux groupes (ainsi, de tels groupes n'ont pas de nœuds passerelles) ; des groupes sont susceptibles de se superposer lorsque des nœuds peuvent être des membres de deux groupes (il existe des nœuds passerelles) ;
- la nature active, passive ou hybride de la méthode de *clustering* : une méthode active transmet des messages de routage spécifiques en vue de former et d'organiser les groupes indépendamment des besoins en transmission des applications ; une méthode passive ne crée des groupes qu'au moment où un nœud a une information à transmettre ; une méthode hybride inclue dans des messages de routage couramment envoyés, des informations nécessaires à la formation et la maintenance des groupes.
- le nombre de niveaux hiérarchiques différents (groupes, sous-groupes ...) créés par la méthode de *clustering*.

Les responsables de groupe sont au cœur des méthodes de *clustering* et leur élection repose sur le choix de métriques. La littérature en a proposé plusieurs de natures différentes. Ainsi, elles peuvent être qualifiées d'arbitraires, de topologiques, d'ordonnées, de type *cross-layer* ou encore de combinées.

Une métrique arbitraire représente une valeur aléatoire non significative pour le réseau de communication. Par exemple, la méthode *Lowest ID* (LID) [36] associe à chaque nœud du réseau "un identifiant

unique et aléatoirement choisi”. Les nœuds voisins à un saut s’échangent la valeur de leur identifiant. Dans un voisinage à un saut, le nœud dont l’identifiant est le plus faible devient un responsable de groupe. Les voisins dont l’identifiant n’est pas le plus faible de leur propre voisinage se rattachent à leur voisin responsable de groupe. Cette méthode simple à mettre en œuvre ne prend en compte ni les particularités du réseau de communication, ni celles de l’environnement dans lequel évoluent les nœuds.

Une métrique est topologique si sa définition dépend de la topologie du réseau. Le “nombre de voisins” en est un exemple exploité dans la méthode *Highest Degree* (HD) [36]. Cette méthode établit la règle suivante : le nœud du réseau élu responsable de groupe est celui qui possède le plus grand nombre de nœuds dans son voisinage à un saut. C’est donc celui-ci qui peut relayer une information à un maximum de voisins.

Les métriques sont dites ordonnées lorsqu’il existe une ou plusieurs métriques secondaires chargées de lever toutes ambiguïtés en cas d’égalité entre plusieurs nœuds suite à l’application de la première métrique. De telles métriques sont mises en œuvre par la méthode *k-hop CONnectivity ID* (k-CONID) [37] qui combine les méthodes LID et HD. Un premier critère identifie le(s) nœud(s) possédant le plus de voisins. Puis, en cas d’égalité de score parmi les nœuds trouvés, un second critère permet de sélectionner celui dont l’identifiant a la plus faible valeur. Ce nœud devient le responsable de groupe.

Une métrique relève de couches croisées (métrique *cross-layer*) lorsqu’elle fait appel à une métrique mesurée au niveau d’une autre couche (au sens du modèle de référence OSI de l’*International Standardization Organization* (ISO)) que celle sur laquelle la métrique est appliquée. Par exemple, la métrique de “puissance de réception des paquets reçus” qui provient de mesures au niveau de la couche 1 (couche physique) est pour la première fois introduite en couche 3 (couche réseau) dans la méthode de groupement *MObility Based metrIC* (MOBIC) [38]. Proche de la méthode LID, l’approche MOBIC remplace le système d’identifiant par un critère de distance des nœuds quantifié par la puissance de réception des paquets.

Enfin, une métrique est combinée lorsqu’elle résulte d’une moyenne pondérée entre plusieurs métriques. Par exemple, la méthode *Weight Based Adaptive Clustering Algorithm* (WBACA) [39] définit une métrique pondérant la puissance de transmission des nœuds, leur nombre de voisins, leur mobilité et l’état de leur batterie.

Un intérêt des méthodes de groupement est, selon [32], qu’elles facilitent le passage à l’échelle. Nous avons vu que les communications peuvent être confinées au sein des groupes. Les responsables de groupe ont alors la charge de superviser et de coordonner les accès au médium de communication pour en optimiser l’utilisation. Ces deux aspects rendent ces méthodes prometteuses pour les réseaux MANETs, tels les réseaux véhiculaires, qui possèdent un grand nombre de nœuds. Les inconvénients du *clustering* relèvent essentiellement du trafic de routage nécessaire à la création et à la maintenance des groupes. En particulier, afin de prendre en compte les changements de topologie du réseau, les messages de routage doivent être envoyés fréquemment lorsque les nœuds du réseau sont fortement mobiles.

1.2.2.3 Topologie hiérarchique construite autour d’une colonne vertébrale (*backbone*)

La structuration logique du réseau autour d’une colonne vertébrale ou *backbone* repose sur l’usage d’une méthode de *clustering*. Une méthode de groupement identifie les responsables de groupe de nœuds et l’approche *backbone* introduit une connexion logique entre ces responsables de groupe (figure 1.6). Dans ce type d’organisation, tous les nœuds sont à moins de k-sauts d’un nœud de la colonne vertébrale. Cette approche a été étudiée par [40] qui propose la création d’un *backbone* selon une topologie en anneau circulaire.

1.2.3 Modes de communication

Lorsqu’un nœud de communication transmet des paquets d’informations dans un réseau MANET, ceux-ci sont diffusés dans l’environnement du nœud sous forme d’ondes radio. Afin d’identifier le ou les destinataires de ces paquets, le système d’adressage de la couche réseau définit différents modes de communication : *unicast*, *broadcast*, *multicast* et *anycast*.

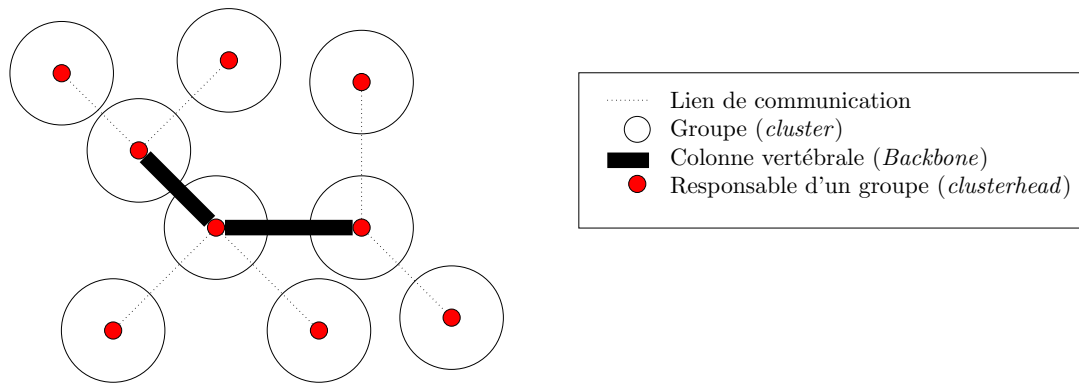


FIGURE 1.6 – Topologie hiérarchique de type colonne vertébrale au sein d'un réseau MANET

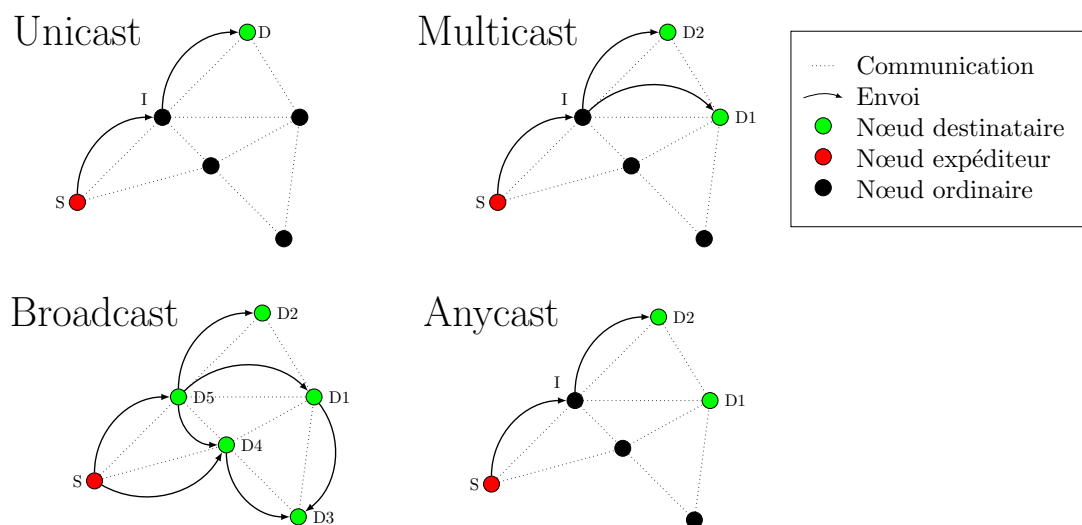
Dans le mode de communication *unicast*, les paquets sont transmis à l'adresse d'un seul nœud. Si ce nœud destinataire est hors portée radio du nœud émetteur, un ou plusieurs nœuds relais les ré-émettent jusqu'à ce qu'ils soient à portée du nœud adressé.

Dans le mode de communication *broadcast*, les paquets sont transmis à destination de tous les nœuds du réseau. Chaque nœud récepteur les rediffuse alors pour qu'ils atteignent l'ensemble des nœuds du réseau.

Dans le mode de communication *multicast*, ils sont transmis à destination d'un groupe de nœuds. Lorsque ce groupe de nœuds récepteurs est défini en fonction de la position géographique, ce mode de communication est appelé *geocast* [41, 42]. Quand les nœuds destinataire sont hors portée radio du nœud émetteur, un ou plusieurs nœuds relais ré-émettent les paquets jusqu'à ce qu'ils soient à portée des nœuds destinataire.

Dans le mode de communication *anycast*, les paquets d'informations du nœud émetteur sont à destination d'un seul nœud choisi parmi les nœuds d'un groupe destinataire. Lorsque ce groupe de nœuds destinataire est défini en fonction de la position géographique de ses nœuds, ce mode de communication est appelé *geoanycast* [41, 42]. Quand le nœud récepteur est hors portée radio du nœud émetteur, un ou plusieurs nœuds relais ré-émettent les paquets jusqu'à ce qu'ils soient à portée du nœud adressé.

La figure 1.7 résume ces quatre principaux modes de communication. Un protocole de routage peut supporter un ou plusieurs modes de communication pour le trafic de paquets applicatifs. Les paquets de routage, permettant le bon fonctionnement du protocole, peuvent également exploiter l'un de ces différents modes.

FIGURE 1.7 – Modes de communication *unicast*, *broadcast*, *multicast* et *anycast*

1.2.4 Recherche de route par dissémination de requêtes

La recherche de routes s'effectue par dissémination de requêtes de routage. Elle a pour objectif de répertorier l'ensemble des nœuds intermédiaires (les nœuds relais) qui relayeront les paquets applicatifs d'un nœud expéditeur à un ou plusieurs destinataires, si celui ou ceux-ci sont hors portée radio. Il existe trois principales méthodes de dissémination de requêtes. Les méthodes proactives, réactives et hybrides se distinguent par le choix de l'instant où elles initient la procédure de recherche de routes [43].

1.2.4.1 Approche proactive

Dans une approche proactive, chaque nœud maintient une table de routage, en plus de sa table de voisins à un saut. Chacun diffuse périodiquement des paquets de routage dans le réseau, indépendamment de toute demande de transmission de paquets. Ces paquets de routage contiennent la liste des routes connues par le nœud. Ces échanges de paquets permettent la gestion des tables de routage. Une table de routage contient à minima l'adresse d'un nœud potentiellement destinataire et l'adresse du premier relais (à un saut) capable de rediffuser l'information vers ce nœud destinataire (figure 1.8). Selon les protocoles proactifs, une métrique de coût peut être associée à chaque route. Une métrique de coût est par exemple le nombre de nœuds relais nécessaires à l'atteinte du destinataire. Lors d'une demande de transmission en mode *unicast* ou *multicast*, un nœud transmet ses paquets à l'adresse du premier nœud relais en direction du ou des nœud(s) destinataire(s). À réception de chaque paquet du message, le nœud relais procède de la même façon en le rediffusant à destination du prochain nœud relais de la route les conduisant vers le ou les nœud(s) destinataire(s), et ainsi de suite jusqu'à destination. DSDV [27] et OLSR [31] sont des exemples de protocoles proactifs pour les MANETs.

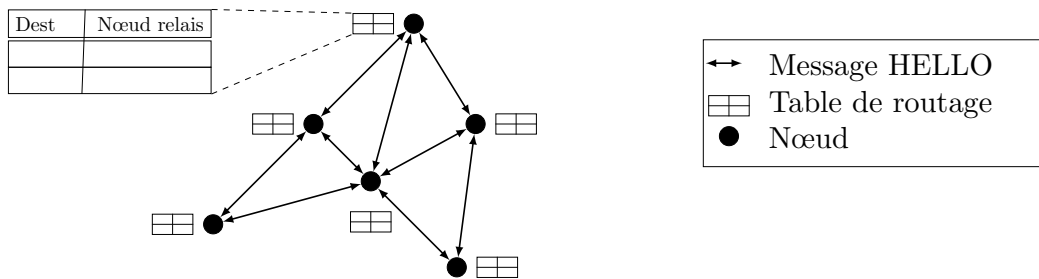


FIGURE 1.8 – Présentation du fonctionnement global des méthodes proactives - figure inspirée de [43]

Les méthodes de dissémination proactive mettent à disposition, à tout moment, une route vers tout nœud du réseau. Cependant, la gestion des tables de routage nécessite la mise en œuvre d'un trafic de routage périodique permanent.

1.2.4.2 Approche réactive

Dans une approche réactive, les recherches de routes sont réalisées à chaque consigne d'émission de paquets applicatifs à l'aide de messages *Route Requests* (RREQ) et *Route Replies* (RREP) (figure 1.9). Un paquet RREQ est envoyé en mode *broadcast* et retransmis de nœud en nœud jusqu'à atteindre le nœud destinataire.

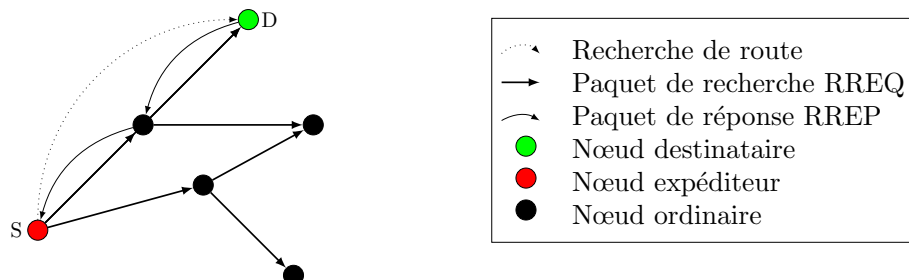


FIGURE 1.9 – Présentation du fonctionnement global des méthodes réactives - figure inspirée de [43]

Dans le cas du protocole *Dynamic Source Routing* (DSR) [44] qui réalise un routage par la source,

avant retransmission, chaque nœud intermédiaire y ajoute son adresse. Le nœud destinataire retourne alors un paquet de routage RREP au nœud source contenant la liste des adresses des nœuds intermédiaires qui lui ont transmis le paquet RREQ. Ce nœud destinataire peut recevoir plusieurs copies du message RREQ qui ont suivi différentes routes. Le nœud destinataire peut ensuite sélectionner une des routes pour construire le paquet de réponse RREP. À réception du paquet RREP le nœud source reconstitue la route nécessaire au transfert de ces paquets applicatifs vers le destinataire.

Dans d'autres protocoles tel que *Ad-hoc On Demand Distance Vector Protocol* (AODV) [44], le paquet RREQ ne contient pas de liste d'adresses des nœuds intermédiaires. Ces adresses intermédiaires sont enregistrées localement dans des tables de routage. À réception d'un paquet RREP ou RREQ, les nœuds mémorisent un ensemble de trois adresses : l'adresse du nœud source à l'origine du paquet de requête, celle du nœud voisin qui a relayé ce paquet et celle du nœud destinataire final. Selon les protocoles réactifs, une métrique de coût peut être associée à chaque route.

Contrairement aux approches proactives, les méthodes de dissémination réactives ne nécessitent pas, pour leur fonctionnement, des envois périodiques de paquets de routage dans le réseau. Cependant, un délai de recherche de route est nécessaire à chaque consigne d'émission de paquets, ce qui retarde le moment de transmission des paquets applicatifs.

La figure 1.10 extraite de la référence [45] (annotée en rouge par nos soins) compare analytiquement le trafic de routage généré par le protocole proactif OLSR et le protocole réactif AODV en fonction du nombre de nœuds dans le réseau pour des paramètres intrinsèques identiques. Seuls deux paramètres τ et θ sont spécifiques. τ quantifie la fréquence d'envoi des messages de contrôle de topologie du protocole OLSR (section 2.5.1) et θ les perturbations des requêtes RREP et RREQ du protocole AODV. Nous observons des points de décrochage pour lesquels le trafic de routage d'AODV devient supérieur à celui d'OLSR à partir d'un certain nombre de nœuds dans le réseau, quel que soit le paramétrage des deux protocoles. OLSR génère moins de trafic de routage pour un grand nombre de nœuds dans le réseau pour des paramètres intrinsèques équivalents.

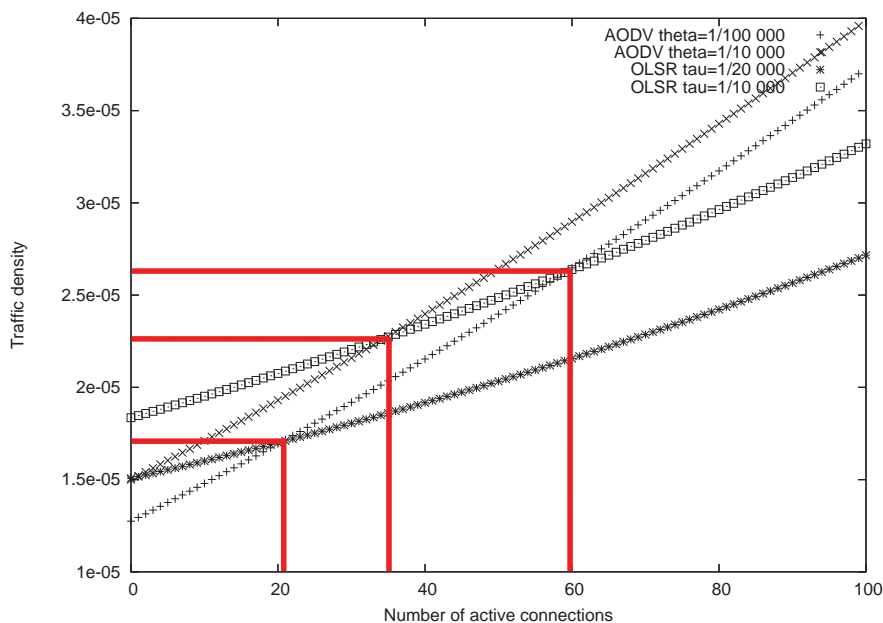


FIGURE 1.10 – Comparatif du trafic de routage généré entre OLSR et AODV - source [45]

1.2.4.3 Approche hybride

Une approche hybride combine les approches réactives et proactives. Selon [43], une approche proactive est adoptée dans une région proche du nœud, et une approche réactive pour les nœuds les plus éloignés. La région proche est définie en nombre de sauts entre les nœuds (soit en nombre de nœuds relais intermédiaires). Au delà de ce nombre de nœuds relais, les nœuds sont considérés comme hors de

la région proche. Selon ce principe, la figure 1.11 montre une région proche définie par un voisinage d'un saut autour du nœud source S. Le nœud destinataire D, à deux sauts, est lui à l'extérieur de cette région proche. Pour connaître les routes vers les nœuds voisins de leur région proche, chaque nœud transmet périodiquement des paquets de routage. Pour atteindre des nœuds hors de cette région, une recherche de route est déclenchée au moment de la consigne d'émission des paquets applicatifs et est réalisée via des échanges de paquets RREQ-RREP.

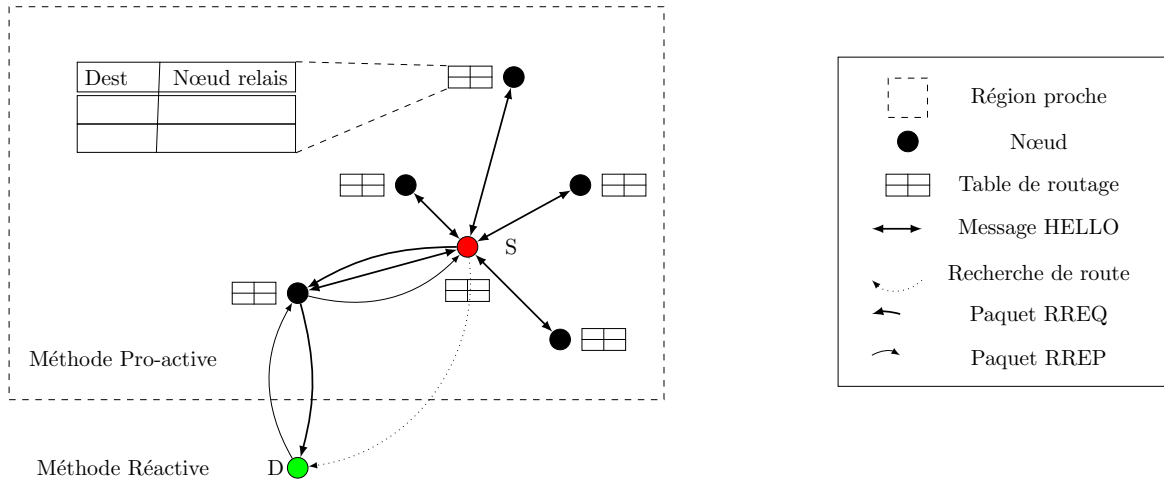


FIGURE 1.11 – Présentation du fonctionnement des méthodes hybrides - figure inspirée de [43]

1.3 Stratégies de routage pour le contexte véhiculaire

L'environnement de travail des MANETs a un impact sur leur topologie et parfois leurs performances. Comme le montre l'abondance des solutions protocolaires de la littérature (tableaux 1.2 et 1.3), il n'existe pas de protocole de routage adapté à l'ensemble des topologies et des différentes contraintes environnementales s'exerçant sur les MANETs. Dans cette section, nous nous intéressons aux contraintes subies par les *Vehicular Ad Hoc Network* (VANET). En particulier en section 1.3.1, nous étudierons les métriques contextuelles liées aux réseaux véhiculaires, ces métriques étant utilisées dans les protocoles de routage pour les VANETs de la littérature, puis en section 1.3.2, aux métriques de *Quality of Service* (QoS) exploitées dans les protocoles de routage pour les VANETs.

1.3.1 Stratégies et informations contextuelles propres aux systèmes routiers

Les différentes stratégies de routage peuvent être dissociées en fonction des données dont elles ont besoin pour fonctionner. Nous proposons de distinguer en trois catégories les données issues du contexte véhiculaire (voir figure 1.12) :

- les caractéristiques internes aux véhicules, par exemple la position, la vitesse ou l'accélération ;
- les caractéristiques externes aux véhicules propres à l'environnement comme le type de route, le nombre de voies ou le réseau routier ;
- les caractéristiques hybrides en interaction entre le véhicule et l'environnement telles que la densité ou l'itinéraire.

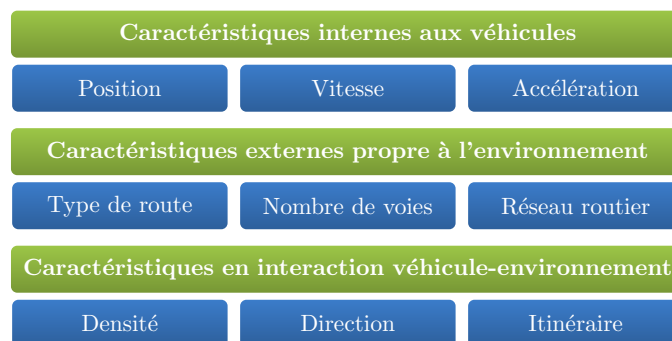


FIGURE 1.12 – Classification des informations propres au contexte véhiculaire

1.3.1.1 Information de position de véhicules

L'information de positionnement provient en général d'un système de localisation satellitaire intégré au véhicule. Le nœud partage cette information périodiquement avec ses voisins à un saut en l'insérant dans ses messages de découverte du voisinage (section 1.2.2). Elle est alors utilisable par les algorithmes de routage des nœuds à portée directe de communication.

Cependant, certains algorithmes de routage nécessitent la connaissance de la position du nœud destinataire. Or, ce nœud peut être hors de la portée de communication. La mise en place d'un "service de localisation" (*Location Service*) est alors nécessaire pour renseigner chaque nœud du réseau sur la position des autres nœuds. Cette position reste approximative compte tenu de la dynamique des réseaux VANETs. L'état de l'art [46] distingue les services de localisation exploitant une méthode par inondation de ceux reposant sur une méthode "sur rendez-vous". Les méthodes par inondation (section 1.2.1), proactives ou réactives, sont supportées par une topologie horizontale. Celles proactives transmettent périodiquement des paquets contenant l'information de position des nœuds, paquets qui sont alors relayés dans le réseau. Celles réactives émettent dans le réseau un paquet de recherche de position du destinataire au moment où cette information est nécessaire. Les méthodes sur rendez-vous attribuent un rôle de "serveur de localisation" à certains nœuds du réseau, établissant ainsi une topologie hiérarchique. L'obtention d'une position d'un nœud du réseau est alors réalisée par l'émission d'une requête au nœud le plus proche qui remplit le rôle de serveur de localisation.

À noter, quelle que soit la méthode d'échanges de l'information de position adoptée (par inondation ou sur rendez-vous), la mise en œuvre d'un service de localisation nécessite un trafic de communication dédié à ces échanges, trafic gourmand en ressources de communication [46]. Cependant, ce trafic n'est généralement pas pris en compte (il est alors négligé) lors de l'évaluation des performances des protocoles de routage qui nécessitent la connaissance de la position des nœuds pour leur fonctionnement.

De nombreuses stratégies de routage exploitent depuis les années 2000 cette information de position des véhicules. Les protocoles de routage géographiques, qui l'utilisent pour leur fonctionnement, montrent de bonnes performances pour les VANETs [47, 48, 49]. Des protocoles de routage mettent en œuvre des stratégies de recherche de route contraintes spatialement à l'intérieur de zones de recherche [50, 51, 52, 53] ou relativement à un axe source-destination de recherche [54, 53] (paragraphe a.). D'autres l'exploitent pour la création de *clusters* et de responsables de groupe. (paragraphe b.)

a. Recherche de route sous contraintes géographiques de surfaces et d'axes

Des restrictions par zones géographiques peuvent être employées de manière globale au réseau lors de la recherche d'une route de manière réactive ou de manière locale à chaque nœud lors du choix du nœud relais suivant.

Les protocoles réactifs *Distance Routing Effect Algorithm for Mobility* (DREAM) et *Location-Aided Routing* (LAR) [50, 51] restreignent globalement le mécanisme de recherche de route à une zone géographique. Celle-ci est délimitée à partir des positions du nœud source et du nœud destinataire. Le protocole LAR définit une zone rectangulaire (figure 1.13), le protocole DREAM une zone triangulaire (figure 1.13). La forme de la zone a un impact non négligeable. Ainsi, dans l'exemple de la figure 1.13, le protocole DREAM ne trouve pas de route entre le nœud expéditeur S et destinataire D en raison de l'absence de nœuds intermédiaires, hormis N1, dans la zone de recherche de route, contrairement au protocole LAR dont la zone rectangulaire comporte plus de nœuds. Le protocole DREAM doit alors élargir la surface de sa zone de recherche triangulaire initiale. Cette restriction géographique de recherche de route introduite dans le protocole LAR a été ensuite reprise dans la version *Geographic Ad-hoc On Demand Distance Vector Protocol* (GeoAODV) [55] du protocole AODV.

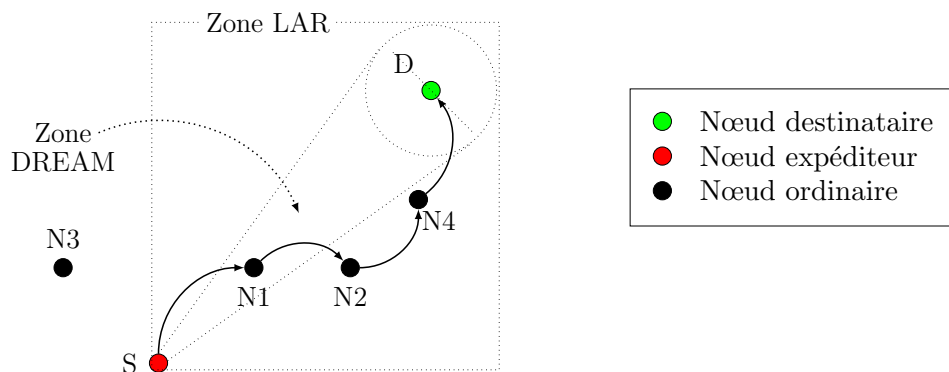


FIGURE 1.13 – Restriction de recherche de route par la définition d'une zone géographique

La méthode *Preferred Group Broadcasting* (PGB) [56], mise en œuvre par le protocole *Connectivity Aware Routing* (CAR) [52], propose quant à elle de restreindre localement le mécanisme de recherche du nœud relais suivant, au moyen d'une zone en anneau définie en périphérie de chaque nœud relais en fonction de la portée de communication (figure 1.14). Les nœuds voisins dont la position est dans la zone PGB comprise entre $3/5$ et $4/5$ de la portée de transmission maximale sont retenus en priorité comme potentiels prochains nœuds relais.

L'algorithme de routage glouton élit localement, saut après saut, chaque nœud relais suivant en minimisant la distance séparant le nœud relais du nœud destinataire. Chaque nouveau nœud relais sélectionné est celui dont la distance au nœud destinataire est la plus courte, relativement à l'axe de recherche défini par les positions du nœud source et du nœud destinataire. Cette optimisation locale de la distance entre les nœuds relais intermédiaires de la route vise à obtenir un nombre minimum de sauts entre les nœuds source et destinataire. Dans l'exemple de la figure 1.15, le nœud source S (point vert) élit N1 comme

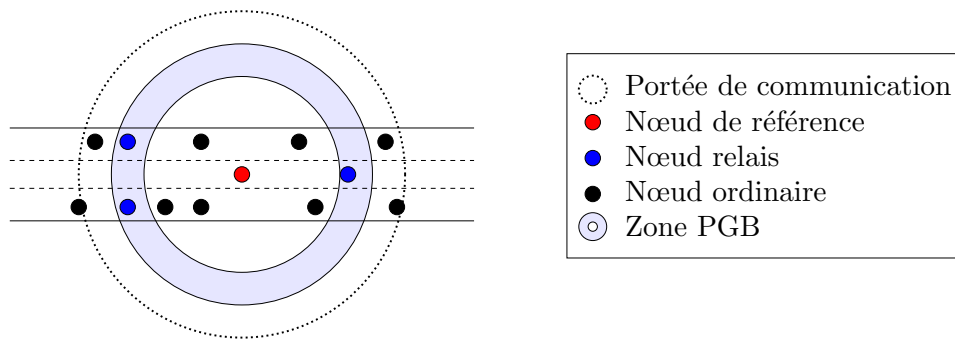


FIGURE 1.14 – Restriction géographique locale de la méthode PGB – schéma inspiré de [16]

nœud relais lors de son envoi d'un paquet au nœud destinataire D (point rouge), N1 étant le plus proche de D selon l'axe défini par les positions géographiques de S et D. Cependant, dans un réseau VANET au sein duquel les nœuds ont une forte mobilité, l'algorithme peut créer des routes dont la suite ordonnée des nœuds relais intermédiaires crée des boucles. De plus, l'algorithme de routage glouton favorise le choix de nœuds relais qui sont en limite de portée de communication, car plus proches du nœud destinataire. En raison de cette forte mobilité, un nœud peut rapidement sortir de la zone de portée du nœud qui l'a choisi, la route devenant caduque [57] et le paquet transmis est alors perdu. Enfin, lorsque la recherche de route atteint un optimum local, par exemple lorsqu'une zone est vide de nœud ou en présence d'un obstacle, la méthode de routage gloutonne est mise en défaut par l'arrêt complet du processus de recherche de route. Ainsi, figure 1.16, avec la seule méthode gloutonne de routage, l'algorithme arrêterait sa recherche au nœud N1, nœud le plus proche géographiquement du nœud destinataire D.

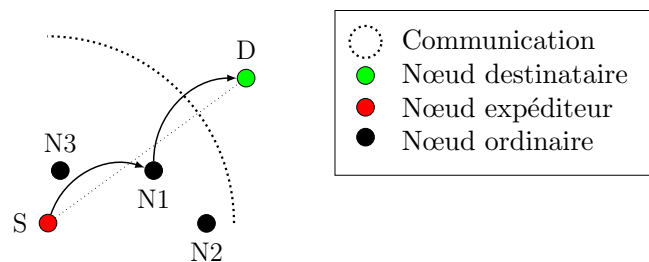


FIGURE 1.15 – Choix d'un nœud relais selon l'algorithme de routage glouton qui minimise la distance entre le relais et le nœud destination selon l'axe (nœud source, nœud destinataire)

Pour empêcher la création de boucles dans une route et l'arrêt d'une recherche de route en présence d'un optimum local, le protocole *Greedy Perimeter Stateless Routing for Wireless Networks* (GPSR) [54] associe une méthode alternative de "routage du périmètre" à celle du routage glouton qui lui permet de contourner des zones vides de nœud en raison, par exemple, de l'existence d'un obstacle. La méthode de "routage du périmètre" comporte deux phases. Lors de la première, l'algorithme construit un graphe de voisinage non orienté, appelé *Relative Neighborhood Graph* (RNG), qui prend en compte toutes liaisons entre deux nœuds pour lesquelles il n'existe pas de troisième nœud plus proche. Ainsi, figure 1.16, même si les nœuds N1 et N4 sont à portée radio, seules les liaisons N1-N2 et N2-N4 sont retenues. Lors de la seconde phase, le nœud relais suivant est élu parmi les candidats selon "la règle de la main droite" qui consiste à minimiser l'angle orienté défini entre deux nœuds. Cette règle permet de parcourir le graphe RNG jusqu'au nœud destinataire.

Ainsi, figure 1.16, la méthode de routage du périmètre du protocole GPSR [54], protocole réactif et à hiérarchie horizontale, permet, à partir du nœud N1, optimum local de l'algorithme de routage glouton, de poursuivre la construction de la route jusqu'au nœud destinataire D par la sélection des nœuds relais intermédiaires N2, N3, N4, N5 et N6. Notons que cette méthode de routage du périmètre n'optimise pas le nombre de nœuds relais.

Le protocole, géographique et proactif, *Geographic Routing Protocol* (GRP) [53] utilise également la méthode de routage gloutonne, mais en association avec une division hiérarchique du réseau en zones, sous-zones, sous-sous-zones... rectangulaires. La première division du réseau correspondant au niveau

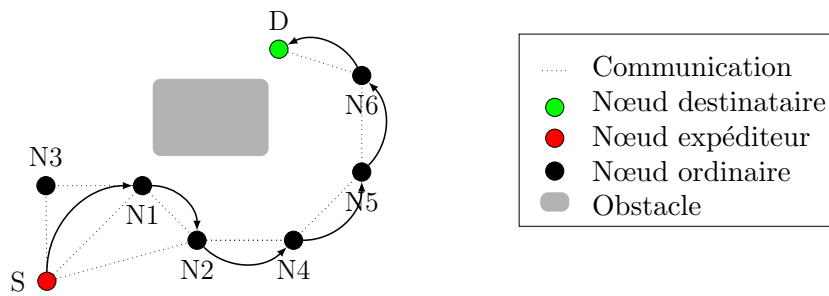


FIGURE 1.16 – Présentation de la méthode périmètre permettant d’outre-passer les zones vide de nœuds (à cause d’un obstacle) lors de l’utilisation du routage glouton. Figure inspirée de [16]

hiérarchique 1. Chaque zone résultante est alors découpée en quatre sous-zones de niveau hiérarchique 2, et ainsi de suite (figure 1.17). La création de cette topologie hiérarchique permet de canaliser la propagation d’un paquet en la cantonnant à une zone limitée géographiquement, elle peut aussi permettre la définition de *clusters* ou encore des communications en mode *geocast*.

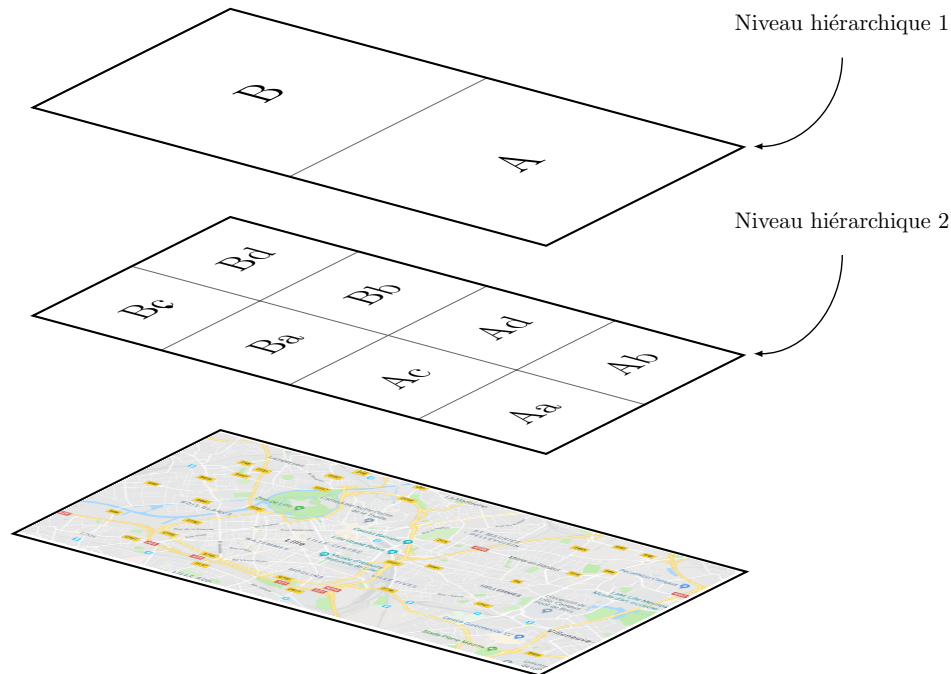


FIGURE 1.17 – Division topologique du réseau par le protocole GRP [53]

b. Création de *clusters* et élection de chef de groupe

L’information de position, qui permet notamment de calculer la distance entre deux nœuds voisins dans le réseau, est fréquemment exploitée par les méthodes de groupement [58, 59, 60, 61]. Par exemple, le protocole *Position-Based Clustering* (PBC) [62] utilise la valeur de la distance pour former des groupes dont la taille est limitée par un seuil. L’information de position est aussi utilisée dans la phase d’élection de responsables de groupe. Ainsi, les protocoles *Cluster Chain* (CC) [61] et *Destination Based Routing* (DBR) [63] recherchent comme nœud responsable de groupe celui qui minimise la valeur de la moyenne de la distance de tous les nœuds voisins. Le responsable de groupe résultant est alors le nœud le plus proche de tous ses voisins ; il se situe au centre géographique du groupement.

1.3.1.2 Information de vitesse

La donnée de vitesse des véhicules est, tout comme la donnée de position, très utilisée par les stratégies des protocoles de routage adaptés aux réseaux VANETs. Cette donnée peut, par exemple, être issue du

capteur vitesse de rotation du moteur. Comme la donnée de position, elle doit être partagée entre les nœuds voisins du réseau. La donnée de vitesse peut être ajoutée au sein des paquets HELLO de découverte du voisinage envoyés périodiquement. Lorsque ce paquet est reçu par un nœud voisin, celui-ci enregistre la vitesse de ce nœud dans ses tables de routage. Cette information est réactualisée lors de la réception d'un nouveau paquet par ce même voisin. Ce fonctionnement permet à tous les nœuds du réseau de connaître la vitesse de chacun de ses voisins à un saut.

La valeur de la vitesse recherchée n'est pas la même selon les protocoles. Certains s'intéressent à la vitesse relative des nœuds du réseau [64, 65, 66, 67, 68], d'autres à la vitesse moyenne [63, 69, 70] et d'autres au nœud ayant la valeur de la vitesse la plus faible parmi leur voisins [71, 72].

Les protocoles [64, 65, 66, 67, 68] utilisent donc la vitesse relative pour la formation de groupes de véhicules communicants avec l'objectif de former la structure la plus stable possible dans le temps. Le protocole *Stable Clusters* (SC) [68], par exemple, crée des groupes de véhicules pour lesquels la vitesse relative entre le responsable de groupe et les membres de ce groupe est faible. Ainsi, la distance séparant les membres d'un groupe de son responsable évolue de manière limitée au cours du temps. Les membres restent à portée de communication, ce qui limite le trafic de routage nécessaire à la maintenance des groupes.

Les protocoles [63, 69, 70] s'intéressent aux vitesses moyennes des nœuds voisins pour élire le responsable du groupe [63, 69] ou former des groupes de taille adaptative [70]. Ainsi, [69] définit un score pour chaque nœud en fonction de l'écart entre sa vitesse et la vitesse moyenne de ces nœuds voisins, son nombre de voisins et la qualité de la liaison avec la station de base (RSU) la plus proche. Le nœud obtenant le meilleur score est alors élu responsable de groupe par ses voisins. Pour former des groupes de taille adaptative, le protocole *Distributed Cluster formation for vanet based on End-to-End Relative mobility* (DCEV) [70] a spécifié une nouvelle métrique de mobilité qui combine la distance relative moyenne et la vitesse relative moyenne entre un nœud et ses nœuds voisins.

Le protocole de routage *BackBone Routing* (BBR) [71] est défini pour des communications en mode *unicast*. Le protocole *BackBone Broadcast Routing* (BBBR) [73] est établi pour des communications en mode *broadcast*, ainsi que leur version améliorée *Backbone Based routing with inter-flow Network Coding* (BBNC) [72] qui utilise la donnée de vitesse des nœuds pour leur fonctionnement. Les algorithmes de sélection des nœuds du *backbone* sélectionnent les nœuds ayant une vitesse la plus faible parmi leurs voisins pour enclencher la formation de la colonne vertébrale.

1.3.1.3 Information d'accélération

La donnée d'accélération est peu utilisée au sein des protocoles de routage. À notre connaissance seul le protocole *Dynamic Clustering Algorithm* (DCA) [74] structure le réseau VANET par un groupement grâce à une métrique nécessitant l'accélération des véhicules. Cette métrique est nommée "dépendance spatiale" et est calculée entre deux nœuds du réseau comme le produit de leur vitesse relative et de leur accélération relative. Chaque nœud calcule la moyenne de la "dépendance spatiale" avec ses voisins. Le nœud ayant la plus faible "dépendance spatiale" devient le responsable de groupe. Ce nœud correspond au nœud ayant une trajectoire la plus proche de ses voisins en termes de vitesse et d'accélération.

1.3.1.4 Information cartographique d'intersection de rues

De nombreux protocoles de la littérature concernent des applications en milieux urbains, par exemple [75, 76, 77] et utilisent un système d'ancre. Ces protocoles considèrent les intersections de rues comme des points particuliers du réseau routier. Les paquets de routage transitent alors par ces points fixes. Ces protocoles nécessitent que chaque nœud de communication ait accès à une cartographie précise et à jour de l'infrastructure routière. Dans une approche d'ancre, un nœud recherche tout d'abord un ensemble de points fixes qui le relie au destinataire des paquets. Ensuite les paquets sont relayés jusqu'au nœud destinataire par un algorithme de routage glouton.

Le protocole *Geographic Source Routing* (GSR) [75] s'appuie sur l'algorithme du plus court chemin (Dijkstra) pour fournir une liste des intersections à franchir, dans le but d'atteindre le nœud destinataire, liste incluse dans les paquets à transmettre. Les nœuds relais utilisent ensuite la méthode de routage gloutonne (figure 1.18). Les protocoles *Anchor based Street and Traffic-Aware Routing* (A-STAR) [76] et

Greedy Traffic Aware Routing (GyTAR) [77] ont par la suite été définis sur la base de GSR. Le protocole A-STAR privilégie, dans l'étape relative au choix des intersections à franchir, les rues empruntées par les bus car elles sont considérées comme des routes à fortes connectivités. Le protocole GyTAR affecte un système de poids dynamique à chaque rue qui est fonction du nombre de nœuds présents dans cette rue. À la différence des protocoles GSR et A-STAR, l'ensemble des intersections par lesquelles le paquet doit transiter est recalculé à chaque passage d'une intersection, prenant ainsi en compte la valeur courante de la densité du trafic.

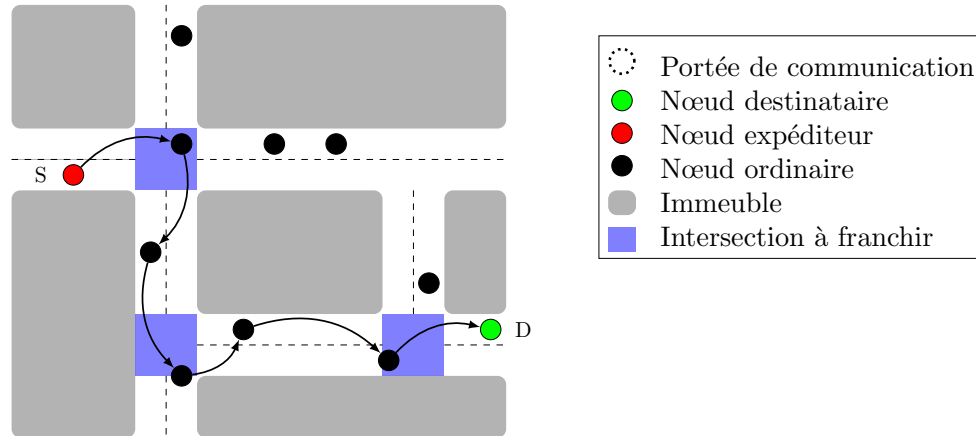


FIGURE 1.18 – Système d'ancrage mis en œuvre par le protocole GSR

1.3.1.5 Information de densité

La densité d'une population est une mesure du nombre moyen d'individus occupant une région donnée. Elle s'obtient en divisant la population considérée totale par la superficie de la région considérée. Ainsi, si on assimile la population à des véhicules, la densité des véhicules sur une route est, en théorie, la mesure du nombre moyen de véhicules sur cette route divisée par la longueur de la section routière considérée (en prenant en compte le nombre de voies). Dans un réseau ad hoc sans fil complètement distribué, pour lequel de nouveaux nœuds peuvent à tout moment s'insérer ou au contraire se retirer du système de communication comme ce serait le cas de réseaux VANET sans contrôle de l'infrastructure, il n'est pas possible, pour un nœud de communication de connaître le nombre exact de nœuds communicants du réseau ad hoc. Chaque nœud n'a connaissance que d'une estimation du nombre de nœuds présents dans son voisinage à un saut, c'est-à-dire dans le voisinage défini par la portée radio de la technologie sans fil utilisée. Pour un nœud donné, une estimation du nombre moyen de véhicules communicants est réalisable par la mesure locale du nombre moyen de voisins de ce nœud de communication. La métrique de densité se réfère donc généralement au nombre de nœuds voisins. Une estimation de la valeur de la densité à un instant t consiste pour un nœud communicant à comptabiliser le nombre de nœuds voisins enregistrés à cet instant t dans sa table de routage. Cette estimation de la densité est cependant limitée géographiquement à la zone de portée de la technologie radio.

La densité a un impact important sur le réseau de communication. À forte densité, le nombre de nœuds à portée de communication est plus élevé que celui à faible densité. Les contraintes de partage du canal et d'accès au médium sont alors plus fortes : un nombre plus important de nœuds entrent en compétition pour l'accès au médium et occasionnent des risques de collision et des délais de préemption accrus dans le cas des protocoles de la couche MAC mettant en œuvre une stratégie *Carrier Sense Multiple Access* (CSMA) ; de plus, la bande passante de la technologie reste constante quel que soit le nombre de nœuds.

Le protocole *Road Aware QoS Routing in Vanets* (RAQV) [78] utilise cette estimation de la densité pour qualifier la faculté d'un nœud relais d'une route à trouver un chemin alternatif en cas de rupture de communication entre deux nœuds relais de cette route. Plus les nœuds relais d'une route ont un nombre moyen élevé de voisins, plus il leur sera facile de trouver une route alternative.

Les protocoles de *CAR* [52] et *Density Based Clustering* (DBC) [79] utilisent cette estimation de la densité pour adapter la fréquence de transmission des messages HELLO en fonction du nombre de nœuds sollicitant le médium de transmission. Ils diminuent la fréquence de transmission en présence d'une forte

densité et l'augmentent à l'occasion d'une densité faible. Les messages HELLO de ces deux protocoles contiennent les valeurs de variables locales de position, de vitesse, et de direction de déplacement du nœud. La variation de la fréquence de transmission a un impact sur celle du rafraîchissement de ces valeurs auprès des nœuds voisins, et par conséquent sur les mises à jour de leur connaissance de la topologie du réseau.

Le protocole *Dynamic Time-Stable Geocast* (DTSG) [80] fait, quant à lui, varier la puissance d'émission en fonction du nombre moyen de voisins dans le but d'optimiser et de faciliter localement l'usage des ressources du médium de communication (figure 1.19). Lorsque le nombre moyen de voisins est faible, c'est à dire lorsque la densité est faible, les nœuds du réseau sont susceptibles d'être plus éparpillés, voire à des distances plus lointaines. Les nœuds accroissent alors leur portée radio en amplifiant leur puissance d'émission, les rendant accessibles à plus de nœuds. Par contre, lorsque la densité est forte, les nœuds diminuent leur portée radio en réduisant leur puissance d'émission. Ils sont alors à portée d'un nombre plus restreint de nœuds avec lesquels ils entrent en concurrence pour les accès au médium et le partage de la bande passante. Les délais d'accès aux ressources canal sont alors localement optimisés. Cependant, la diminution de la portée de communication entraîne une augmentation du nombre de nœuds relais d'une route, provoquant un allongement du délai de transmission des paquets de bout en bout (deux sauts au lieu de un saut dans le cas de la figure 1.19). Par ailleurs, si le protocole est réactif, comme c'est le cas de DTSG, le délai de recherche de route est également augmenté. Enfin, les routes trouvées par le protocole, qu'il soit réactif ou proactif, sont susceptibles de devenir rapidement caduques si la distance entre deux nœuds relais est supérieure à la nouvelle portée de communication.

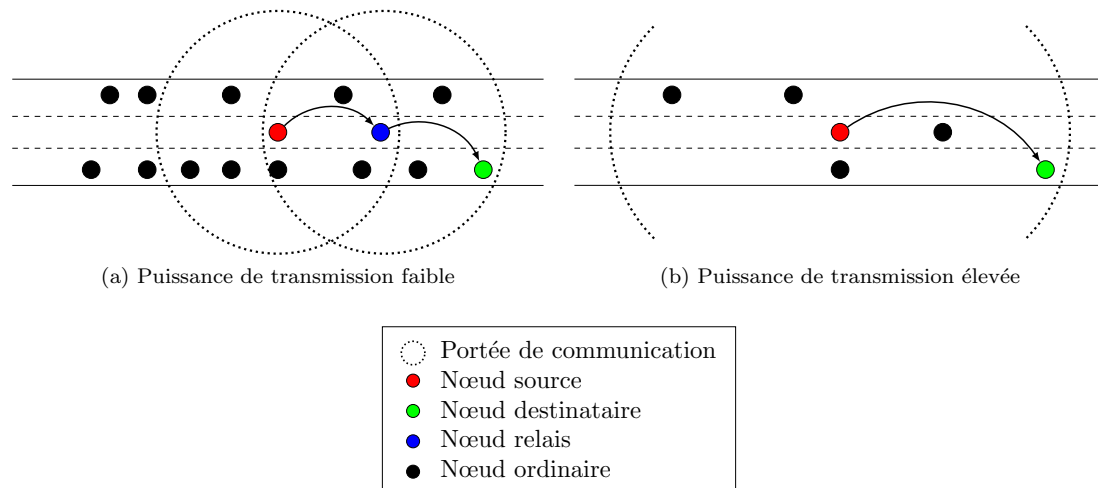


FIGURE 1.19 – Variation de puissance en fonction du nombre moyen de voisins

Les protocoles de groupement *Relatively Stable Clustering for Unbiased Environments* (ReSCUE) [81] et *Stable Clustering Scheme* (SCS) [82] utilisent l'information du nombre moyen de voisins pour l'élection de leurs responsables de groupe, en combinaison avec les variables de position, de vitesse, et d'écart type de la vitesse.

Le protocole *Landmark Overlays for urban Vehicular Routing Environments* (LOUVRE) [83] adapté au milieu urbain calcule de manière différente la donnée de densité. Les nœuds enregistrent l'estimation de la densité de la rue dans laquelle ils se trouvent. Lorsqu'ils sont proches d'une intersection, ils transmettent ces valeurs aux nœuds voisins. Chaque nœud situé à une intersection a connaissance de la densité approximative des rues desservant l'intersection. Lorsque les paquets de recherche d'une route atteignent un nœud relais situé à une intersection de plusieurs rues, le nœud relais situé dans la rue la plus dense est choisi comme nœud relais suivant. Ce choix est motivé par la volonté d'éviter les zones vides de véhicules qui empêchent les paquets de recherche de route d'atteindre le ou les destinataires.

1.3.1.6 Information sur les voies de circulation

En milieu urbain, le flux de véhicules se divise à chaque intersection en fonction de la direction voulue : la ou les voies les plus à droite permettent aux véhicules d'effectuer des tourne-à-droite, les voies centrales de continuer dans la même direction et la ou les voies de gauche d'effectuer des tourne-à-gauche.

Dans ce contexte urbain, *clustering based on DIRection in Vehicular Environment* (C-DRIVE) [58] utilise la détection de la voie des véhicules afin de grouper, en amont d’une intersection, les véhicules allant dans une même direction. Plusieurs groupes peuvent coexister dans une même rue si des véhicules circulent sur des voies distinctes qui donneront lieu à des mouvements directionnels différents à la prochaine intersection.

Le protocole *Connected dominating Set for Stable Virtual Backbone* (CDS-SVB) [67] calcule un coefficient qui est fonction du positionnement du véhicule sur une voie du réseau routier et l’utilise lors de l’élection du responsable de groupe. La valeur de ce coefficient diminue lorsque le véhicule est positionné sur une des voies situées à l’extrême droite ou à l’extrême gauche du réseau routier, laissant présager que le véhicule va effectuer un tourne-à-droite ou un tourne-à-gauche.

Quant au protocole de routage géographique *Geographic Stateless VANET Routing* (GeoSVR) [84], afin d’éviter de construire des routes passant par des nœuds relais situés dans des rues à faible densité, et donc à faible connectivité, il propose d’utiliser la connaissance du nombre de voies d’une rue pour estimer la densité de véhicules pouvant y circuler : plus une rue a un nombre de voies élevées, plus elle est sensée accueillir un trafic de véhicules important et donc une densité de véhicules forte.

1.3.1.7 Information sur la direction de déplacement

L’information de direction de déplacement des nœuds peut être obtenue au niveau d’un nœud de différentes façons. Certains la partagent entre les nœuds via les paquets HELLO [58, 68, 85, 86, 87]. Le protocole *Vehicular clustering based on the Weighted Clustering Algorithm* (VWCA) [88] la recalcule à partir de l’enregistrement de la position antérieure et la connaissance de la position actuelle d’un nœud voisin. Les protocoles *Angle-based Clustering Algorithm* (ACA) [87] et *Mobility Prediction-based Directed Broadcasting* (MPDB) [89] font appel à un seuil angulaire (de 45°) pour définir les différents sens de direction de circulation.

Cette information permet aux protocoles de travailler sur la mesure du temps de communication, d’anticiper les déplacements des véhicules et leur éloignement de la zone de portée des nœuds ou encore de transférer des paquets en aval (dans la direction de circulation) ou en amont du flux routier.

La direction de déplacement renseigne sur le temps de communication attendu. Pour une même valeur absolue de vitesse des véhicules, ce temps sera plus long pour deux véhicules circulant dans la même direction que celui de deux véhicules dans des directions opposées. De nombreux protocoles de *clustering* tels que [58, 68, 85, 86, 87, 88, 89] utilisent ainsi cette information afin de former des groupes avec des véhicules communicants qui restent à portée de communication. L’objectif des protocoles comme *Based on Euclidean distance clustering* (BEDC) [59] et *Vehicular Multi-hop algorithm for stable clustering cluster* (VMaSC) [90] est la formation de groupes pour chaque direction de circulation.

Le protocole *Particle Swarm Optimization Cluster* (PSO-C) [91] autorise les messages à transiter par des *clusters* ayant des directions de circulation opposées. Ceci permet de relayer les messages applicatifs lorsqu’il n’y a pas de nœud relais dans l’une des directions de circulation.

Le protocole GPSR+PRedict [92] propose d’améliorer le protocole GPSR (présenté en section 1.3.1.1) en exploitant l’information de direction de déplacement des nœuds afin de prédire la position future des nœuds voisins et ainsi limiter le nombre de déconnexions entre les nœuds. Cet algorithme de routage recherche une route alternative à une route contenant deux nœuds en limite de portée de communication qui se dirigent dans des directions opposées. Cette idée de prévoir le déplacement est encore repris dans le protocole *Vehicle Assisted Data Delivery* (VADD) [93]. VADD propose deux modes d’élections des nœuds relais, le mode “section courante” (*straightway*), pour lequel le nœud relais candidat au saut suivant est trouvé au moyen d’un algorithme glouton, et le mode intersection. Dans le mode intersection, le nœud relais candidat au saut suivant est soit trouvé par une approche appelée L-VADD, qui vise à minimiser la distance vers le nœud destinataire (ainsi, figure 1.20, le nœud source S choisirait le nœud relais A, plus proche du nœud destinataire D), soit par l’approche D-VADD qui privilégie la direction des nœuds (ainsi, figure 1.20, S choisirait le nœud relais B qui est dans la même direction que D), soit enfin par une approche hybride H-VADD afin d’éviter les boucles potentiellement induites par L-VADD.

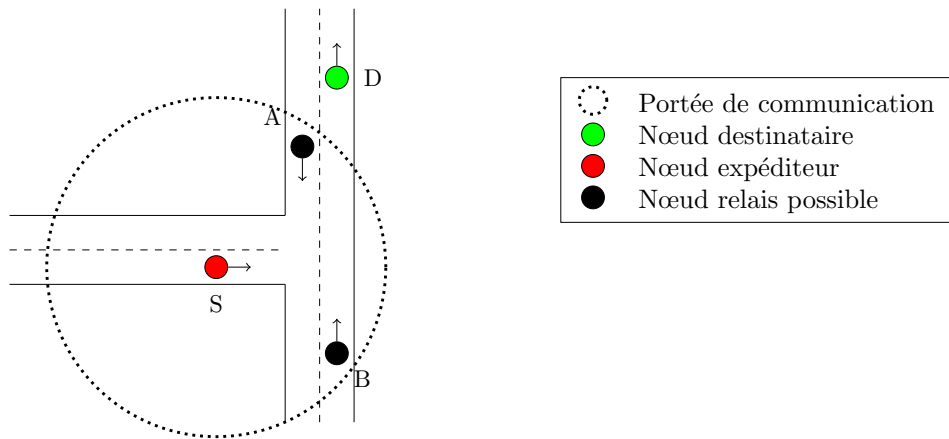


FIGURE 1.20 – Exemple de l'utilisation du protocole VADD dans une intersection [16]

Dans la stratégie de routage *Advanced Greedy forwarding* (AGF) [56] et dans le protocole CAR [52], chaque nœud diffuse, dans son voisinage à un saut, sa direction de déplacement et sa vitesse. Ces informations, combinées avec celles du nœud qui les reçoit, permettent à ce dernier de ne pas élire comme nœud relais d'une route un nœud voisin dont la trajectoire va sortir de sa zone de portée.

La direction de déplacement peut aussi être exploitée pour orienter la transmission des paquets dans une direction de circulation du trafic routier. Le protocole *Relative Position Based Message Dissemination* (RPB-MD) [94] introduit ainsi la notion d'émission de paquets en aval du trafic routier, c'est-à-dire dans la direction de circulation des véhicules. Ce protocole utilise une méthode de routage gloutonne directionnelle basée sur les nœuds en aval du trafic routier qui relaient l'envoi de paquets. Il repose sur l'hypothèse que chaque nœud du réseau est équipé de deux antennes unidirectionnelles et peut alors diviser en deux groupes l'ensemble de ses nœuds voisins selon que ces derniers soient en aval ou en amont. Deux autres protocoles BBR [71] et BBR [73] exploitent également la direction de déplacement pour l'émission de paquets en amont ou en aval du trafic.

1.3.1.8 Information sur l'itinéraire du véhicule

La connaissance de l'itinéraire du véhicule est utilisée par [95] et [62] pour l'estimation du temps de trajet. Ainsi, [95] assure une bonne stabilité des groupes de sa méthode de *clustering* en choisissant pour responsables de groupe ceux qui maximisent la métrique du temps de trajet. Cette métrique, ainsi que l'information d'écart de vitesse, sert également dans l'élection des responsables de groupes du protocole PBC [62].

La connaissance de la destination finale est utilisée pour l'élection des responsables de groupe des protocoles de *clustering* [96, 63]. Elle permet à [96] d'obtenir une bonne stabilité des groupes en privilégiant un responsable de groupe ayant une destination finale éloignée.

Dans le protocole DBR [63], c'est la valeur moyenne de la distance restante à parcourir aux nœuds voisins pour atteindre leur destination finale qui est utilisée dans l'élection des responsables de groupe.

1.3.2 Stratégies de routage et métriques de qualité de service

1.3.2.1 Introduction

Les protocoles de routage qui exploitent des informations relatives aux technologies et algorithmes du système de communication sont les protocoles de routage à *Quality of Service* (QoS). Pour tenir compte des exigences des applications en terme de bande passante, délai de réception ou perte de paquets, ces protocoles tâchent de répondre à des attentes antagonistes : trouver la route qui limite le taux de pertes des paquets, assure un délai minimal de transmission et offre le débit le plus élevé.

Les protocoles à QoS dans les réseaux VANETs doivent tenir compte en particulier des contraintes physiques des ressources de communication en terme de bande passante, de portée de communication et de pertes de messages dues aussi bien à la nature du médium qu'au protocole d'accès au médium. [97] distingue quatre familles de métriques pour les algorithmes des protocoles de routage à QoS dédiés aux

VANETs : les métriques relatives à la topologie du réseau, à la qualité du lien, à la charge du trafic ou encore à la mobilité des liens (tableau 1.1).

Mathématiquement, une métrique est additive, multiplicative, concave ou convexe. Le résultat d'une métrique additive (par exemple le délai) correspond à la somme de la métrique de chaque lien entre nœuds relais. Celui d'une métrique multiplicative (par exemple le taux d'erreurs binaires) correspond au produit de la métrique de chaque lien entre nœuds relais. Celui d'une métrique concave (par exemple la bande passante) correspond au minimum de la métrique pour chaque lien. Enfin, celui d'une métrique convexe (par exemple le taux de perte) correspond au maximum de la métrique de chaque lien (tableau 1.1).

Une métrique se mesure localement ou de façon externe au nœud [98]. Une métrique locale est directement accessible par le nœud qui la mesure. Par exemple, un nœud peut calculer le nombre de voisins à un saut en comptabilisant le nombre de nœuds voisins enregistrés dans sa table des voisins à un saut. Une métrique externe est mesurée par observation passive, active ou indirecte du trafic sur le réseau. Un nœud réalise une mesure passive par l'intermédiaire des paquets qu'il envoie et reçoit. Par exemple, le calcul de la métrique "nombre des paquets reçus d'un nœud voisin" relève d'une observation passive des paquets transmis sur le médium. Une mesure externe active ou indirecte impacte les messages de routage ou les paquets de données transmis par insertion d'informations nécessaires à la mesure. Un nœud réalisant une mesure externe active d'une ou plusieurs métriques transmet des messages dédiés aux calculs de ces métriques. Une telle mesure est intrusive. Enfin, un nœud réalisant une mesure externe indirecte, ajoute cette fois des champs dédiés à cette mesure dans les paquets de routage déjà existants qu'il transmet (tableau 1.1).

1.3.2.2 Métriques de délais

Deux métriques d'évaluation des délais sont en particulier utilisées dans la littérature :

- le délai de bout en bout ou *End-to-End Delay* (EED) ;
- le nombre de sauts ou *Number of Hops* (NoH).

Le délai de bout en bout est le temps mis par un paquet pour atteindre sa destination à partir de l'instant de la création du paquet par la couche applicative du nœud source jusqu'à celui où la couche applicative du nœud destinataire le reçoit. Cette métrique additionne les temps passés par le paquet dans les files d'attente des différentes couches du modèle OSI, le temps de ses traitements, celui de sa (ses) transmission(s) et de sa (ses) propagation(s) entre nœuds. Cette métrique est utilisée, notamment par les protocoles *Adaptive QoS for Vehicular safety services* (AQVA) [99], *QoS Bee Routing Protocol* (QoSBee) [100], *Intersection-based QoS Routing in Vehicular ad hoc networks* (IRQV) [101] ou *VANETs' quality Of service by Adaptive protocol* (VOA) [102], à l'occasion de l'élection des nœuds relais dans leur procédure de recherche de route réactive. Elle intervient aussi comme métrique de coût pour la recherche d'une route en mode glouton dans le protocole *Backbone Assited Hop Greedy* (BAHG) [103].

La métrique du nombre de sauts comptabilise le nombre de nœuds relais entre le nœud source et le nœud destinataire. C'est donc une mesure indirecte du délai de bout en bout d'une route, qui augmente également avec le nombre de nœuds relais d'une route. En minimisant cette métrique, les protocoles RAQV [78] et RQ-AODV [104], cherchent les routes qui réduiront les délais de transfert des paquets.

1.3.2.3 Métriques de pertes

Deux métriques d'évaluation des pertes sont notamment citées dans la littérature :

- le nombre de transmissions attendu ou *Expected Transmission Count* (ETX) ;
- le temps avant expiration du lien ou *Link Expiration Time* (LET).

La métrique du nombre de transmissions attendu correspond au nombre de transmissions requis pour envoyer un paquet entre deux nœuds en incluant les cas de retransmissions consécutifs à une mauvaise réception. Elle est calculée en fonction du taux d'erreur de paquets reçus et envoyés par deux nœuds du réseau. Elle est souvent calculée de manière indirecte via les paquets HELLO. Ces paquets étant envoyés périodiquement, leur nombre important permet d'effectuer des mesures statistiques de la métrique ETX au cours du temps. Le calcul de la métrique ETX par l'usage des paquets HELLO est faussé car ces paquets ont une taille inférieure aux paquets applicatifs. Les paquets HELLO sont moins sujets au problème de

perte que les paquets applicatifs [105]. Cette méthode de mesure permet cependant d’approcher la valeur de la métrique ETX sans envoyer de paquet supplémentaire.

La métrique du temps avant expiration du lien mesure la durée pendant laquelle deux nœuds sont à portée de communication directe l’un de l’autre. Calculée en tenant compte de la mobilité des nœuds, elle est utilisée par un grand nombre de protocoles réactifs dans l’élection du prochain nœud relais (par exemple dans *Adaptive Routing Protocol Based on QoS and Vehicular Density in Urban VANETs* (ARP-QD) [106], RQ-AODV [104], *Optimal Next Hop Selection* (OHNS) [107], VOA [102], *Mobility-based Scheme* (MBS) [108] et RAQV [78]).

1.3.2.4 Métriques de qualité de liaison

Trois métriques d’évaluation de la qualité de la liaison sont notamment référencées dans la littérature :

- l’estimation de la bande passante entre les nœuds ou *Bandwidth (BW)*;
- la puissance de transmission ou *Transmission Power (TP)*;
- la hauteur de l’antenne ou *Antenna Height (AH)*.

La métrique d’estimation de la bande passante entre les nœuds permet d’évaluer la charge du réseau. Pour la mesurer, il est possible d’observer le nombre de fois où le médium de communication est libre par rapport au nombre de fois où il est occupé. Cette métrique est par exemple utilisée par les protocoles réactifs OHNS [107] et AQVA [99] pour sélectionner les nœuds intermédiaires d’une route. Elle sert également au protocole *Dynamic Backbone-Assisted Medium Access Control* (DBA-MAC) [60] pour élire les responsables de groupe. Le protocole QOLSR [109], ses variantes proposées par [110, 111, 112, 113, 114] et le protocole QoS-OLSR [115] utilisent la métrique de bande passante pour améliorer le choix des nœuds MPRs du protocole OLSR.

La métrique de puissance de transmission est utilisée dans les protocoles DBA-MAC [60] et *On-Demand Virtual Backbone BroadCAST Protocol* (OVBCAST) [116] pour privilégier les nœuds ayant le meilleur lien radio lors de l’élection des responsables de groupe. Le protocole VWCA [88] implémente l’algorithme AATR (Adaptive Allocation of Transmission Range) qui permet de faire varier la puissance de transmission en fonction du nombre de nœuds voisins. Lors de l’initialisation du nœud, la puissance est faible. Si peu de nœuds voisins sont détectés alors l’algorithme AATR augmente la valeur de la puissance de transmission jusqu’à ce que le nœud ait un nombre de nœuds voisins suffisant.

La taille de l’antenne est également un indicateur de qualité de la liaison radio. Plus la taille de l’antenne est élevée (typiquement celle d’un bus ou d’un camion), plus la portée de communication et la qualité de réception seront importantes. Cette métrique est considérée dans les protocoles BBR [71], BBR [73] et BBNC dans la fonction de sélection des nœuds de leur *backbone*.

TABLEAU 1.1 – Métriques de qualité de service

Métrique	Optimisation			Classification				Propriétés Mathématiques					Méthode de Mesure				
	Délai	Pertes	Débit	Topologie	Qualité lien	Mobilité	Charge trafic	Additif	Multi	Concaves	Convexes	Objectif	Interne	Externe	Passive	Active	Hybride
ETT	✓				✓			✓				↘		✓		✓	✓
NoH	✓			✓				✓				↘	✓				
ETX		✓			✓		✓			✓		↘		✓	✓	✓	✓
LET		✓			✓	✓				✓		↗		✓			✓
BW			✓				✓			✓		↗		✓	✓	✓	✓
TP			✓		✓					✓		↗		✓	✓	✓	✓
AH			✓		✓					✓		↗	✓		✓	✓	✓

1.4 Conclusion

Dans ce chapitre nous avons présenté les notions des réseaux de communication et en particulier les spécificités des réseaux véhiculaires. Nous avons décrit le fonctionnement des protocoles de routage dans les réseaux mobiles ad hoc et également les différentes stratégies mises en œuvre par les protocoles de routage adaptés aux VANETs.

Cet état de l'art a été réalisé sous l'angle des métriques nécessaires à la mise en œuvre des stratégies de routage. Nous avons vu que ces métriques étaient issues à la fois du contexte véhiculaire et de la qualité de service. Nous avons vu que les protocoles de routage étaient constitués d'une combinaison de stratégies afin de répondre à un besoin spécifique (par exemple être efficace en milieu urbain).

La création de nouveaux protocoles de routage est un processus incrémental. Le fonctionnement d'un nouveau protocole repose sur un déjà existant mais avec l'intégration d'une ou plusieurs stratégies. Nous notons qu'aucun consensus n'existe actuellement pour un protocole de routage unique répondant à toutes les contraintes physiques inhérentes au médium de communication et à celles imposées par le contexte routier et le passage à l'échelle.

Dans le chapitre suivant nous allons spécifier notre contribution sur l'organisation du réseaux de communication véhiculaire ainsi que de sa mise en œuvre dans un protocole de routage déjà existant.

TABLEAU 1.2 – Protocoles de routage classiques de la littérature

	Ref	Nom	Attributs			Informations véhiculaires						Métriques technologiques							
			Mode	Route	Réseau	Position	Vitesse	Densité	Direction	Carte	Itinéraire	Grille	Délai		Pertes		Débit		
											EED	NoH	ETX	LET	BW	PT	AH		
MANET	[44]	AODV	U	R	-														
	[31]	OLSR	UB	P	-														
	[54]	GPSR	U	R	-	✓												∅	
	[51]	LAR	U	R	-	✓													
	[50]	DREAM	U	R	-	✓													
	[117]	ZRP	U	H	-														
	[118]	FSR	U	R	-														
VANET	[76]	A-STAR	U	R	V	✓				✓									
	[77]	GyTAR	U	R	V	✓				✓									
	[56]	PGB	UB	R	-	✓													
	[56]	AGF	U	R	-	✓	✓												
	[119]	GPSRJ+	U	R	V	✓													
	[52]	CAR	UB	R	A/V	✓	✓			✓									∅
	[83]	LOUVRE	U	R	V	✓		✓											
	[93]	VADD	U	R	V	✓	✓			✓									
	[53]	GRP	U	P	V/A	✓				✓									✓
	[80]	DTSG	U	R	A	✓	✓	✓	✓										
	[84]	GeoSVR	U	R	V	✓	✓			✓	✓								
	[92]	GPSR+P	U	R	A/V	✓	✓			✓									
[120]	RPGR	U	R	V	✓		✓			✓									

U, Unicast ; B, Broadcast ; R, Réactif ; P, Proactif ; H, Hybride A, Autoroutier ; V, Ville

TABLEAU 1.3 – Protocoles de routage de groupement et à QoS de la littérature

Ref	Nom	Attributs			Informations véhiculaires						Métriques technologiques						
		Mode	Route	Réseau	Position	Vitesse	Densité	Direction	Carte	Itinéraire	Grille	Délai		Pertes		Débit	
											EED	NoH	ETX	LET	BW	PT	AH
[59]	BEDC	UB	R	A	✓		✓										
[62]	PBC	U		A	✓	✓				✓							
[88]	VWCA	U	-	A	✓	✓	✓	✓									
[68]	SC	U	-	A	✓	✓	✓	✓									
[108]	MBS	UB	-	V	✓	✓		✓						✓			
[63]	DBR	UB	P	V	✓	✓		✓		✓							
[82]	SCS	UB	-	AV	✓	✓	✓	✓					✓			✓	
[58]	C-DRIVE	B	-	V	✓			✓	✓		✓						
[70]	DCEV	U	-	A	✓	✓											
[90]	VMaSC	U	-	-		✓		✓									
[91]	PSO-C	UB	-	A	✓	✓		✓					✓				
[87]	ACA	B	-	A/V	✓	✓	✓	✓									
[89]	MPDB	U	P	A	✓	✓	✓										
[71]	BBR	U	-	AV	✓	✓	✓	✓									✓
[73]	BBBR	B	-	AV	✓	✓	✓	✓									✓
[61]	CC	U	-	A	✓												
[72]	BBNC	B	-	AV	✓	✓	✓	✓									
[103]	BAHG	U	R	V	✓				✓		✓						
[60]	DBA-MAC	B	R	V	✓	✓		✓			✓					✓	
[116]	OVB-CAST	B	R	V	✓	✓		✓								✓	
[67]	CDS-SVB	U	P	V	✓	✓		✓									
[121]	RBF	U	R	A	✓	✓			✓			✓	✓		✓		✓
[101]	IRQV	U	R	V							✓						
[107]	OHNS	U	R	V	✓								✓	✓			
[104]	RQ-AODV	U	R	A			✓						✓	✓			
[99]	AQVA	U	R	V							✓				✓		
[102]	VOA	U	R	A	✓						✓		✓				
[78]	RAQV	U	R	V			✓						✓		✓		
[122]	GVGRID	U	R	V	✓	✓		✓		✓			✓				
[115]	QoS OLSR	UB	P	A	✓	✓	✓			✓		✓			✓		
[114]	QOLSR	UB	P	-							✓	✓			✓		
[100]	QoS-Bee	U	R	V							✓	✓			✓		
[106]	ARP-OD	U	R	V	✓	✓			✓				✓				

U : Unicast ; B : Broadcast ; R : Réactif ; P : Proactif ; A : Autoroutier ; V : Ville

Chapitre 2

Proposition d'organisation structurelle des réseaux véhiculaires : CBL

Sommaire

2.1	Introduction	60
2.2	Définitions et métriques dédiées au schéma <i>Chain Branch Leaf</i> (CBL)	61
2.2.1	Notion de nœuds branche et feuille	61
2.2.2	Notion de chaîne	62
2.2.3	Spécification du sens de propagation d'un message applicatif	63
2.2.4	Définition des métriques utilisées	63
2.2.5	Spécification des tables des voisins à un saut	64
2.2.6	Spécification des variables locales des nœuds	65
2.2.7	Spécification du contenu des messages HELLO	67
2.2.8	Définitions d'un seuil temporel et des coefficients pondérateurs	67
2.3	Algorithmes de construction de la structure CBL	70
2.3.1	Procédure d'émission d'un message HELLO	70
2.3.2	Diagramme fonctionnel de CBL à réception d'un message HELLO	70
2.3.3	Mise à jour de la table des voisins (algorithme 1)	70
2.3.4	Procédure de changement de type, de feuille à branche (algorithme 2)	72
2.3.5	Traitement type feuille (algorithme 3)	72
2.3.6	Traitement type branche (algorithme 4)	75
2.3.7	Procédure de changement de type, de branche à feuille (algorithme 5)	82
2.3.8	Diagrammes d'interactions temporelles	83
2.3.9	Paramétrage des coefficients pondérateurs	87
2.4	Discussion –Propriétés de CBL	90
2.4.1	Élection d'un nœud branche par un nœud feuille (algorithme 3)	90
2.4.2	Gestion des chaînes (algorithme 4)	91
2.5	Mise en œuvre de CBL dans le protocole OLSR : CBL-OLSR	94
2.5.1	Présentation du protocole OLSR	94
2.5.2	Correspondance entre MPRs et nœuds branche	96
2.5.3	Modification du format des paquets	96
2.5.4	Modification du format des messages HELLO	98
2.5.5	Méthode de recherche de route	100
2.5.6	Modification des tables de voisins	100

2.1 Introduction

Historiquement, les protocoles de routage ont été conçus pour les MANETs dans lesquels les nœuds sont répartis aléatoirement dans l'espace sans qu'aucune direction géographique de mouvement ne soit privilégiée. Cependant, dans le cas des VANETs, ces nœuds correspondent à des véhicules auxquels l'infrastructure et la signalisation routières imposent une direction de mouvement et des contraintes en termes de mobilité. Nous estimons que cette réalité de terrain doit être prise en compte dans la conception des algorithmes de routage dédiés aux VANETs. Ainsi, pour notre proposition [8] de structuration du réseau adapté aux communications véhicule à véhicule, nous avons cherché à ce qu'elle réponde à un certain nombre de contraintes qui pourraient représenter de futures exigences pour les véhicules autonomes. Ces contraintes sont d'ordres environnementaux, structurels et applicatifs.

Intégré dans un environnement routier, un système de communication V2V doit répondre au fil de l'eau aux différentes variations de configuration du trafic sur les voies routières. La densité de véhicules est disparate, la vitesse des véhicules est variable et potentiellement élevée, la mobilité est contrainte par le réseau routier. Le système imaginé doit donc être dynamiquement capable de s'adapter à un changement du nombre de nœuds communicants dans le réseau routier (montée ou descente en charge, *scalability*). De plus, bien que les milieux de propagation rencontrés soient divers, cette contrainte environnementale physique n'a pas été prise en compte au niveau de notre proposition. Elle existe cependant, et fait l'objet de nombreux travaux de recherche en particulier au niveau des couches physiques et liaisons de données des systèmes de communication VANETs [123, 105]. En choisissant de ne pas prendre en compte cette donnée, l'algorithme conçu pourra être porté au niveau de la couche réseau, au-dessus des actuelles et futures technologies de communication de niveau physique et liaison de données, qui sont ou seront développées dans les années à venir.

Nous avons considéré que le système devait être adapté structurellement à son environnement, en particulier à la spécificité longitudinale de l'infrastructure routière. Nous nous sommes appuyés sur l'exemple du train [124, 125]. Un train est un ensemble de véhicules (wagons ou voitures) interconnectés entre eux par un réseau fédérateur dont les nœuds sont des relais pour les messages émanant d'un véhicule et à destination d'un autre. Ce réseau fédérateur (*Train bus* ou *Backbone network*) est interconnecté via ses nœuds aux équipements des véhicules. Ces équipements sont capables de communiquer entre eux par diffusion sur un réseau bus. Ils peuvent aussi transmettre ou diffuser de l'information à destination d'un nœud ou d'un bus lointain par l'intermédiaire des nœuds relais du réseau fédérateur (figure 2.1). D'un point de vue applicatif, ce système hiérarchique offre ainsi des services de communication locaux à un groupe d'équipements, et des services de communication lointains pour la communication entre équipements distribués délocalisés. Il offre notamment un service de diffusion de variables (données courtes, souvent temps réel, périodiques ou aperiodiques) et un service de transmissions de messages entre un émetteur et un ou plusieurs destinataires.

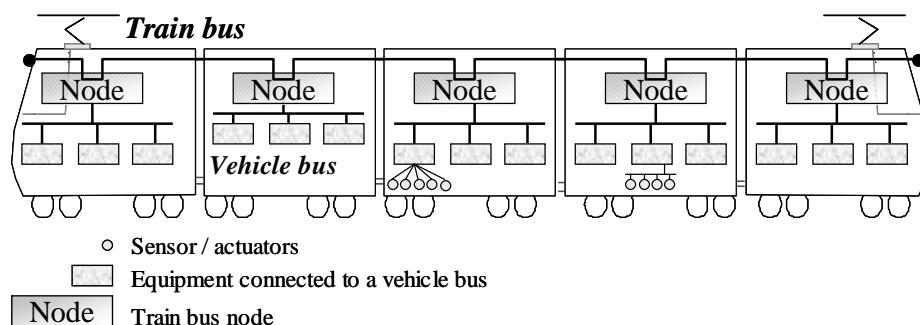


FIGURE 2.1 – Bus de communication dans les trains - extrait de [125]

Nous avons transposé l'idée d'une telle structure hiérarchique en considérant que les nœuds relais et les équipements sont des véhicules communicants de la route. Nous les avons spécifiés de telle sorte que les nœuds relais forment un réseau fédérateur virtuel. Les autres véhicules (les "équipements connectés") peuvent communiquer aussi bien localement par diffusion qu'à destination de nœuds hors de portée radio par l'intermédiaire d'un accès au réseau fédérateur via un des nœuds relais. Pour que cette structure

fédératrice soit “dynamiquement stable”, deux nouvelles métriques qui prennent en compte la portée radio théorique, un invariant de la technologie sous-jacente (IEEE 802.11p, WiMAX, *Long Term Evolution* (LTE) ...) seront introduites en section 2.2.4. En considérant le sens de circulation des véhicules sur le réseau routier lors de la construction des structures fédératrices, nous montrerons dans la section 2.2.2 que ces structures suivent l’aspect longitudinal du réseau routier.

En termes de contraintes applicatives, pour pouvoir naviguer au sein d’un flux d’automobiles en environnement autoroutier ou départemental, chaque véhicule autonome devra partager périodiquement des informations locales avec son voisinage proche (*Cooperative Awareness Message* (CAM) [126]) de sorte à synchroniser ses déplacements conformément à ceux de ses voisins. Il devra, d’autre part, être informé au plus tôt de tout changement dans l’évolution du flux de véhicules (arrêt, ralentissement, accélération de trafic, évitement d’obstacles) de façon à modifier par anticipation sa trajectoire en collaboration avec celles des véhicules de son voisinage. Cette seconde contrainte nécessite la possibilité de transfert de messages apériodiques (*Decentralized Environmental Notification Message* (DEMN) [127]) sur des distances supérieures au voisinage à un saut, entre un nœud du réseau et un ou plusieurs nœuds destinataire. Nous montrerons que la structure fédératrice CBL peut offrir, pour les nœuds mobiles, une solution aux quatre types de scénarios de communication spécifiés dans le standard européen [128], point-à-point, point-à-multipoint, *GeoAnyCast* et *GeoBroadcast*.

Si CBL offre une structure fédératrice similaire à celle obtenue classiquement par une architecture V2I, sa construction repose sur l’échange de messages entre véhicules (V2V). Deux versions ont été réalisées. La première a fait l’objet de la publication [8], la seconde est présentée en section 2.3

2.2 Définitions et métriques dédiées au schéma CBL

Notre proposition s’intéresse à la création de groupes de nœuds communicants à un saut avec un seul niveau de hiérarchie, formant ainsi une topologie hiérarchique. Les membres d’un groupe sont à portée directe de communication avec un responsable du groupe élu, le *cluster-head*. Ainsi, la taille des groupes est de fait limitée à deux fois la portée de transmission radio (P) (voir figure 2.2). Les responsables de groupes forment un groupe dominant connecté dans le réseau (section 1.2.2.2). D’autre part, les groupes formés sont disjoints (section 1.2.2.2).

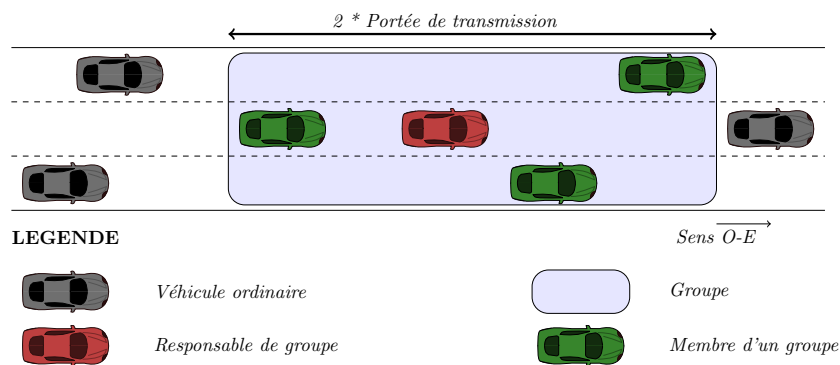


FIGURE 2.2 – Illustration, sur une autoroute à trois voies, d’un groupe (*cluster*) de véhicules créé autour d’un responsable de groupe. La taille du groupe est limitée au voisinage à un saut du responsable, soit à deux fois la portée de transmission de la technologie radio embarquée.

Nous spécifions, dans les paragraphes suivants, les définitions et paramètres nécessaires aux algorithmes du schéma CBL, présentés en section 2.3.

2.2.1 Notion de nœuds branche et feuille

L’algorithme proactif *Chain Branch Leaf* (CBL) repose sur la spécification de deux types de nœuds : les nœuds branche et les nœuds feuille. À un instant donné n’importe quel nœud du réseau est soit un nœud branche soit un nœud feuille.

Un **nœud branche** est un nœud responsable d'un groupe (*cluster*) de nœuds feuille. Chaque nœud branche a un rôle de *cluster-head*. Il est élu par d'autres nœuds (branche ou feuille). Il sert de relais pour les nœuds branche et feuille l'ayant élu. Dans notre proposition CBL, un nœud branche réalise la découverte de son voisinage au moyen de messages périodiques HELLO. Il dispose du droit de relayer les tables de routage au sein du réseau au moyen de messages de contrôle de topologie *Topology Control* (TC). Il participe à la construction d'une chaîne de véhicules relais qui circulent dans une même direction sur la voie routière. Ainsi, acteur de la constitution de la chaîne dont il fait partie, il veille à maintenir une liaison valide avec un nœud branche en amont et un en aval de son sens de circulation et à élire un nœud branche en amont. Le choix de ses nœuds branche, amont et aval, utilise une métrique de temps de chaîne (*ChainTime* - section 2.2.4). Vis-à-vis des messages applicatifs, il est un relais pour la retransmission de messages entre deux nœuds du réseau éloignés de plus d'un saut. Il peut être l'initiateur d'un message applicatif.

Un **nœud feuille** est un nœud ordinaire. Dans le schéma d'organisation du réseau CBL, chaque nœud feuille réalise la découverte de son voisinage au moyen de messages périodiques HELLO. Il établit un lien avec un nœud branche. S'il ne détecte pas de nœud branche dans son voisinage à un saut, il choisit comme nœud branche un des nœuds feuille voisins selon le procédé suivant : 1. C'est un voisin qui circule dans la même direction ; 2. En cas de concurrence, ce voisin est celui ayant la vitesse la plus faible. Tout comme un nœud branche, il peut être l'initiateur d'un message applicatif.

Remarquons que la spécification de nœuds branche et feuille crée des groupes disjoints (section 1.2.2.2) et des liaisons de natures différentes. Chaque nœud étant membre d'un unique groupe identifié par le responsable de groupe (nœud branche), les ensembles de nœuds feuille de deux groupes adjacents d'une même chaîne forment des *clusters* disjoints. Les notions de nœuds branche et feuille nous conduisent à distinguer deux types de liaisons :

- les liaisons entre les nœuds feuille d'un groupe et le responsable branche du groupe ;
- les liaisons deux à deux entre nœuds branche qui définissent un lien entre maillons de chaîne.

2.2.2 Notion de chaîne

Les nœuds branche sont à l'origine de la construction des chaînes. Une chaîne est un réseau fédérateur (*backbone*) virtuel composé d'une séquence ordonnée de nœuds branche qui circulent dans une même direction. Ainsi, une chaîne est créée par sens de circulation dont chaque nœud branche constitue l'un des maillons. Vis-à-vis du routage, les chaînes supportent le trafic des messages de topologie *Topology Control* (TC) transmis par relayage de nœuds branche en nœuds branche. Vis-à-vis des messages applicatifs, les chaînes offrent un service de transfert de messages d'un nœud source à un ou plusieurs nœuds destinataire éloignés de plus d'un saut. La figure 2.3 illustre une formation de chaîne sur une autoroute à trois voies. Remarquons que l'ensemble des nœuds branche forme un groupe dominant connecté car tout nœud du réseau est soit un nœud branche, soit un nœud feuille rattaché directement à un nœud branche et que tout nœud branche peut atteindre n'importe quel nœud par une route dont les nœuds intermédiaires sont des nœuds branche.

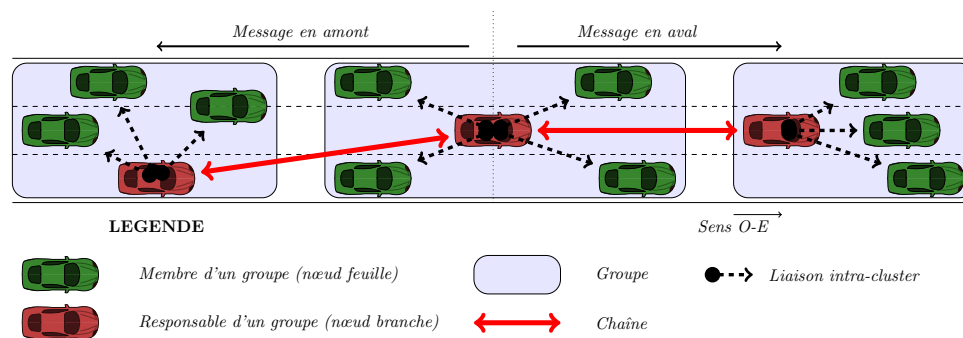


FIGURE 2.3 – Illustration de la création d'une chaîne dans un cas autoroutier avec le concept d'envoi des messages en fonction du sens de circulation, en aval ou en amont du trafic.

2.2.3 Spécification du sens de propagation d'un message applicatif

Pour que les messages applicatifs puissent être transférés en amont ou en aval d'une chaîne, ou encore diffusés à tous les nœuds au voisinage d'une chaîne, un champ contenant une information de direction est ajouté à l'entête des paquets des messages applicatifs. Soit "Dir" le nom de ce champ d'en-tête. "Dir" code les consignes suivantes de façon à permettre pour un paquet donné l'exécution d'une de ces consignes ou d'un ensemble de ces consignes :

- transmettre en aval du trafic ;
- transmettre en amont du trafic ;
- transmettre au voisinage proche ;
- transmettre à tous les nœuds du réseau.

Avec ce champ, l'algorithme contrôle la propagation d'un message applicatif, limitant les risques d'inondation du VANET (*broadcast storm*). Par exemple, selon le codage de "Dir" dans l'en-tête du message applicatif, il sera possible d'émettre un message d'alerte de risque d'embouteillage (message *Decentralized Environmental Notification Message* (DENM)) en aval du trafic ; ou un message d'alerte de la présence d'un véhicule prioritaire (police, pompier...) en amont ; ou encore un message périodique pour signaler sa présence aux véhicules voisins (message CAM).

2.2.4 Définition des métriques utilisées

La création de la chaîne CBL nécessite l'usage des quatre métriques de décisions suivantes :

- la position des nœuds (voir aussi section 1.3.1.1) ;
- la vitesse des nœuds (section 1.3.1.2) ;
- la direction des nœuds (section 1.3.1.7) ;
- le degré d'un nœud (section 1.3.2.2) ;

Nous travaillons sur un espace à deux dimensions au même titre que la plupart des protocoles de la littérature et contrairement aux travaux de [129] sur le protocole GPSR qui comparent les résultats à deux et trois dimensions.

La position $[X_i, Y_i]$ d'un nœud mobile N_i est utilisée par CBL pour calculer la distance séparant deux nœuds. Elle intervient également dans le calcul de la métrique du temps de connexion et du temps de chaîne (*ChainTime*). Nous posons l'hypothèse que chaque position $[X_i, Y_i]$ d'un nœud mobile N_i est donnée par un système de positionnement satellitaire équipant le véhicule du nœud.

La vitesse des nœuds comprend la vitesse absolue V_i et la valeur de l'angle de braquage σ_i moyennée sur une seconde. CBL utilise la vitesse absolue pour ordonnancer les véhicules. L'angle de braquage permet de les classifier en fonction de leur sens de circulation. La vitesse est d'autre part nécessaire au calcul de la métrique du temps de connexion (*CT*) et du temps de chaîne (*ChainTime*). Les valeurs (V_i, σ_i) peuvent être obtenues au moyen d'un système inertiel et d'un capteur d'angle ou encore être déduites par le suivi des positions successives des nœuds.

Le degré (*Degree*) d'un nœud voisin N_j à un saut de N_i est le nombre de voisins symétriques de ce nœud N_j après exclusion du nœud N_i et des nœuds du voisinage à un saut de N_i . Cette métrique est utilisée afin d'identifier les nœuds voisins à un saut qui apportent la connaissance de nouveaux nœuds voisins.

Le temps de connexion *Connection Time* (CT) est calculé à partir de la position, la vitesse et la direction des nœuds. Il est utilisé par un nœud feuille pour choisir son nœud branche relais. Il est défini comme le temps durant lequel les nœuds pourraient communiquer s'ils gardaient une vitesse constante. Nous avons pris pour temps de connexion la métrique définie dans [130] et [25], conforme à l'équation 2.1.

$$CT(N_i, N_j) = \frac{-(ab + cd) + \sqrt{(a^2 + c^2) * P^2 - (ab - bc)^2}}{a^2 + c^2} \quad (2.1)$$

$$\begin{cases} a = V_i \cos(\sigma_i) - V_j \cos(\sigma_j) & b = X_i - X_j \\ c = V_i \sin(\sigma_i) - V_j \sin(\sigma_j) & d = Y_i - Y_j \end{cases}$$

avec P la portée radio maximale de la technologie réseau utilisée.

La figure 2.4 illustre le temps de connexion entre deux véhicules en fonction de leur inter-distance et de leur vitesse relative lorsque la technologie de réseau utilisée est le IEEE 802.11p. La figure 2.4b montre les résultats obtenus pour deux véhicules circulant dans la même direction, et la figure 2.4a le cas pour deux véhicules évoluant dans des directions opposées. Dans cet exemple, nous observons que les valeurs maximales prises par le CT dans le cas de deux véhicules en sens opposés sont comprises entre 5 et 20 secondes. Pour deux véhicules allant dans le même sens de circulation avec une vitesse relative faible (inférieure à 10 km/h), le temps de connexion peut dépasser 100 secondes.

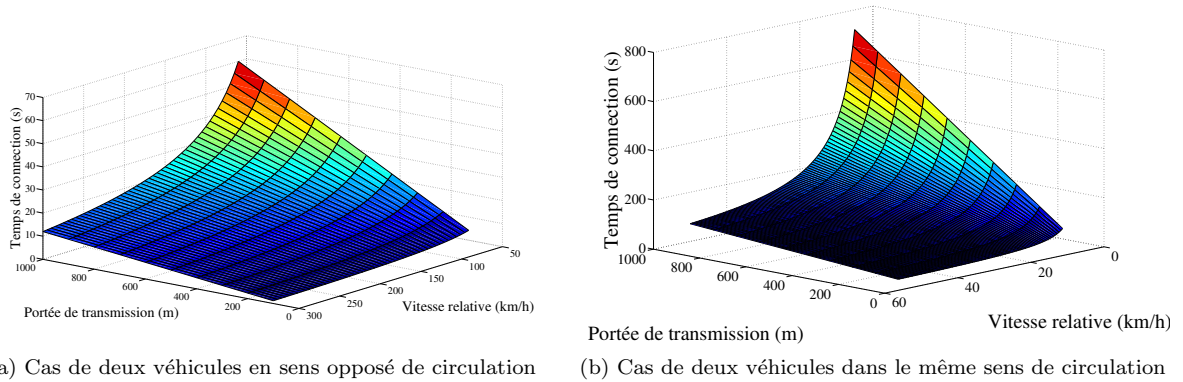


FIGURE 2.4 – Temps de connexion CT en fonction de la vitesse relative et de la portée de transmission

Le temps de chaîne (*ChainTime*) est une nouvelle métrique que nous avons définie pour le choix d'élection des nœuds branche amont et aval d'un nœud branche. Son calcul repose sur celui du temps de connexion *Connection Time* (CT). Il est défini, pour deux nœuds branche d'une chaîne, comme le temps minimum entre leur CT et la durée écoulée avant qu'un des deux nœuds ne dépasse l'autre. *ChainTime* est donné par l'équation 2.2

$$ChainTime(N_i, N_j) = \begin{cases} CT(N_i, N_j) & \text{si } N_i \text{ en aval de } N_j \text{ et } V_j < V_i \\ CT(N_i, N_j) & \text{si } N_i \text{ en amont de } N_j \text{ et } V_j > V_i \\ \min \left[CT(N_i, N_j), \frac{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{|V_i - V_j|} \right] & \text{sinon} \end{cases} \quad (2.2)$$

2.2.5 Spécification des tables des voisins à un saut

Chaque nœud N_i du réseau crée sa table des nœuds voisins qui contient la liste de tous ses voisins à un saut. La liste est mise à jour à partir des informations contenues dans les messages HELLO reçus des nœuds voisins. Pour chaque voisin noté N_j , cette table contient treize champs qui répertorient la position de ce nœud voisin (champs 5 à 8), son type branche ou feuille (champ 9), la nature de sa relation vis-à-vis du nœud propriétaire de la table des voisins (champs 2 et 10) et ses choix de branche (champs 11 à 13). Ces champs sont :

Champ 1. **@Adresse**, l'adresse *Internet Protocol* (IP) du nœud voisin N_j ;

Champ 2. **Lien**, l'état du lien de communication entre le nœud N_i et son voisin N_j :

- Lien=1, le lien est non spécifié (*UNSPEC*) ;
- Lien=2, le lien a été perdu (*LOST*) ;
- Lien=3, le lien est unidirectionnel, i.e. asymétrique (*ASYM*) ;
- Lien=4, le lien est bidirectionnel, i.e. symétrique (*SYM*).

Champ 3. **Date 1**, valeur en secondes de l'horodatage par rapport à une heure de référence de l'instant où la liaison est devenue symétrique ;

- Champ 4. **Date 2**, valeur en secondes de l'horodatage par rapport à une heure de référence de l'instant, au cours duquel a été reçu le dernier HELLO en provenance du nœud voisin N_j .
- Champ 5. **Position X**, la valeur de la position relative en mètre X_j du nœud voisin N_j dans le repère du nœud N_i ;
- Champ 6. **Position Y**, la valeur de la position relative en mètre Y_j du nœud voisin N_j dans le repère du nœud N_i ;
- Champ 7. **Vitesse**, la valeur V_j , en km/h, de la vitesse moyenne de déplacement du véhicule du nœud voisin N_j sur une seconde ;
- Champ 8. **Angle braquage**, la valeur σ_j , en degré, de l'angle de braquage moyen sur une seconde du véhicule du nœud voisin N_j par rapport au nord ;
- Champ 9. **Type**, la variable booléenne spécifiant le type, branche ou feuille, du nœud voisin :
 – Type=1, le voisin N_j est un nœud branche ;
 – Type=0, le voisin N_j est un nœud feuille.
- Champ 10. **Elected**, la variable booléenne identifiant si le nœud voisin l'a élu à titre de nœud relais :
 – Elected=1 le voisin N_j (feuille ou branche) a élu le nœud N_i à titre de nœud relais ;
 – Elected=0 sinon.
- Champ 11. **@N.Br.Choice** (NeighborBranchChoice), l'adresse du nœud branche relais N_r élu par le nœud voisin N_j (nulle si N_j est un nœud branche) ;
- Champ 12. **@N.ChainDO** (NeighborChainDO), l'adresse du nœud branche N_{AR} en amont élu par le nœud voisin N_j (nulle si N_j est un nœud feuille) ;
- Champ 13. **@N.ChainUP** (NeighborChainUP), l'adresse du nœud branche N_{AV} en aval élu par le nœud voisin N_j (nulle si N_j est un nœud feuille).

Le tableau 2.1 est un exemple de table des nœuds voisins à un saut d'un nœud N_i pour lequel le voisin d'adresse 2 est un nœud branche et le voisin d'adresse 3 est un nœud feuille. Le nœud voisin d'adresse 2 a une liaison symétrique avec le nœud N_i (Lien=4) et il a élu N_i au titre de nœud relais (Elected=1). Le nœud voisin d'adresse 3 a une liaison asymétrique avec le nœud N_i (Lien=3) et il a élu le nœud d'adresse 2 au titre de nœud relais (Elected=1).

Champs	1	2	3	4	5	6	7	8	9	10	11	12	3
	Adresse	Lien	Date 1	Date 2	Position X	Position Y	Vitesse	Angle	Type	Elected	N.Br.Choice	N.ChainDO	N.ChainUP
	2	4	50	100	180	10	130	90	1	1	0	0	0
	3	3	-	120	130	20	125	89	0	0	2	0	0

TABLEAU 2.1 – Exemple de table des voisins à un saut d'un nœud N_i

2.2.6 Spécification des variables locales des nœuds

Chaque nœud N_i enregistre et maintient dix-huit variables qui définissent son identité (adresse, type) et ses liens :

- **@Adresse**, l'adresse du nœud N_i ;
- **Type**, le type actuel du nœud N_i :
 – Type=1, le nœud N_i est un nœud branche ;
 – Type=0, le nœud N_i est un nœud feuille.
- **@ChainUP**, l'adresse du nœud branche N_{AV} en aval avec lequel il constitue un maillon de la chaîne. Le champ @ChainUP est nul si le nœud est un nœud feuille ;
- **DegreeChainUP**, le degré du nœud branche N_{AV} en aval ;

- **@ChainDO**, l'adresse du nœud branche N_{AR} en amont avec lequel il constitue un maillon de la chaîne. Le champ @ChainDO est nul si le nœud est un nœud feuille ;
- **DegreeChainDO**, le degré du nœud branche N_{AR} en amont ;
- **@BranchChoice**, l'adresse du nœud choisi comme nœud branche relais N_r . Le champ @BranchChoice est nul si le nœud n'a pas de nœud branche élu ou si le nœud est un nœud branche ;
- **T1**, l'horodatage (en secondes) T1 mémorise :
 - si N_i est un nœud branche, la date de la dernière réception d'un message HELLO en provenance d'un nœud (feuille ou branche) qui l'a élu comme nœud branche relais ;
 - si N_i est un nœud feuille, la date de la dernière réception d'un message HELLO en provenance du nœud branche N_r auquel il est rattaché (@BranchChoice).
- **T2**, l'horodatage T2 (en secondes) mémorise la date de la dernière réception d'un message HELLO en provenance du nœud branche en amont N_{AR} (@ChainDO) ;
- **T3**, l'horodatage T3 (en secondes) mémorise la date de la dernière réception d'un message HELLO en provenance du nœud branche en aval N_{AV} (@ChainUP) ;
- **T4**, l'horodatage T4 (en secondes) mémorise la date de changement de type du nœud N_i de nœud feuille à nœud branche ;
- **T4b**, l'horodatage T4b (en secondes) mémorise la date de changement de type du nœud N_i de nœud branche à nœud feuille
- **T5**, l'horodatage T5 (en secondes) mémorise la date de création du dernier message HELLO par le nœud N_i ;
- **T6**, l'horodatage T6 (en secondes) mémorise l'instant auquel la variable @ChainDO est mise à zéro par le nœud branche N_i . La valeur de T6 est nulle lorsque le nœud N_i est un nœud feuille ou @ChainDO contient une adresse de nœud branche N_{AR} ;
- **T7**, l'horodatage T7 (en secondes) mémorise l'instant auquel la variable @ChainUP est mise à zéro par le nœud branche N_i . La valeur de T7 est nulle lorsque le nœud N_i est un nœud feuille ou @ChainUP contient une adresse de nœud branche N_{AV} ;
- **T8**, l'horodatage T8 (en secondes) mémorise l'instant auquel la variable @BranchChoice est mise à zéro par le nœud N_i . La valeur de T8 est nulle lorsque le nœud N_i est un nœud branche ou @BranchChoice contient une adresse de nœud branche relais N_r ;
- **T9**, l'horodatage T9 (en secondes) mémorise l'instant auquel la variable @ChainDO contient une adresse de nœud branche N_{AR} . La valeur de T9 est nulle lorsque le nœud N_i est un nœud feuille ou @ChainDO est mis à zéro.
- **T10**, l'horodatage T10 (en secondes) mémorise l'instant auquel la variable @ChainUP contient une adresse de nœud branche N_{AV} . La valeur de T10 est nulle lorsque le nœud N_i est un nœud feuille ou @ChainUP est mis à zéro.
- **T11**, l'horodatage T11 (en secondes) mémorise l'instant auquel la variable @BranchChoice contient une adresse de nœud branche relais N_r . La valeur de T11 est nulle lorsque le nœud N_i est un nœud branche ou @BranchChoice est mis à zéro.

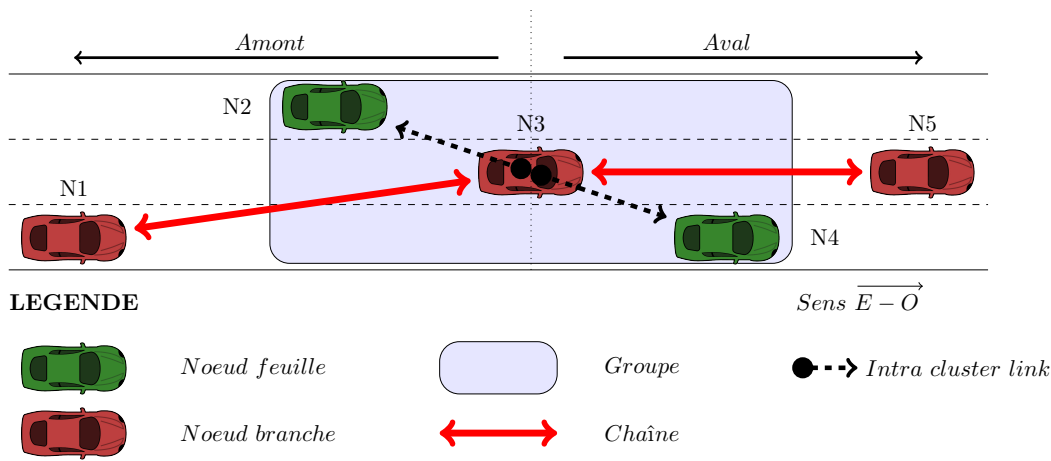
Les valeurs de ces variables locales à l'initialisation du nœud de communication (à $t=T$) sont présentées dans le tableau 2.2.

Variables	Type	@ChainUP	DegreeChainUP	@ChainDO	DegreeChainDO
Valeurs	0	0	0	0	0

Variables	T1	T2	T3	T4	T4b	T5	T6	T7	T8	T9	T10	T11
Valeurs	0	0	0	T	0	0	0	0	T	0	0	0

TABLEAU 2.2 – Valeur à l'initialisation du nœud de ses variables locales (à $t=T$)

La figure 2.5 présente un exemple de topologie du réseau avec certaines variables internes des nœuds.



Les variables internes des noeuds N1, N2, N3, N4 et N5 sont les suivantes :

Noeud N1	Noeud N2	Noeud N3	Noeud N4	Noeud N5
$\left\{ \begin{array}{l} \text{Adresse} = @N1 \\ \text{Type} = 1 \\ @\text{ChainUP} = @V3 \\ @\text{ChainDO} = \emptyset \\ @\text{BranchChoice} = \emptyset \end{array} \right.$	$\left\{ \begin{array}{l} \text{Adresse} = @N2 \\ \text{Type} = 0 \\ @\text{ChainUP} = \emptyset \\ @\text{ChainDO} = \emptyset \\ @\text{BranchChoice} = @N3 \end{array} \right.$	$\left\{ \begin{array}{l} \text{Adresse} = @N3 \\ \text{Type} = 1 \\ @\text{ChainUP} = @V5 \\ @\text{ChainDO} = @V1 \\ @\text{BranchChoice} = \emptyset \end{array} \right.$	$\left\{ \begin{array}{l} \text{Adresse} = @N4 \\ \text{Type} = 0 \\ @\text{ChainUP} = \emptyset \\ @\text{ChainDO} = \emptyset \\ @\text{BranchChoice} = @N3 \end{array} \right.$	$\left\{ \begin{array}{l} \text{Adresse} = @N5 \\ \text{Type} = 1 \\ @\text{ChainUP} = \emptyset \\ @\text{ChainDO} = @V3 \\ @\text{BranchChoice} = \emptyset \end{array} \right.$

FIGURE 2.5 – Contenu de certaines variables internes conformes à un exemple de topologie des noeuds CBL

2.2.7 Spécification du contenu des messages HELLO

CBL est un algorithme proactif dont la découverte des voisins à un saut est réalisée au moyen de la transmission périodique de messages HELLO. Nous noterons $HELLO_INTERVAL$ cette période.

Le contenu des messages envoyés par un nœud N_i contient la liste des adresses des nœuds N_j voisins à un saut dont il a connaissance ainsi que le type de leur liaison. Les messages doivent aussi inclure les huit informations suivantes :

- **@Adresse**, l'adresse du nœud N_i émetteur du message ;
- **Type**, le type actuel du nœud N_i (branche ou feuille) ;
- **Position**, la position absolue (X_i, Y_i) du nœud N_i ;
- **Vitesse**, la valeur V_j , en km/h, de la vitesse moyenne de déplacement du véhicule du nœud voisin N_j sur une seconde ;
- **Angle braquage**, la valeur σ_j , en degré, de l'angle de braquage moyen sur une seconde du véhicule du nœud voisin ;
- **@ChainUP**, l'adresse du nœud branche N_{AV} en aval ;
- **@ChainDO**, l'adresse du nœud branche N_{AR} en amont ;
- **@BranchChoice**, l'adresse du nœud choisi comme nœud branche relais N_r .

2.2.8 Définitions d'un seuil temporel et des coefficients pondérateurs

L'algorithme CBL est un automate exécuté par chaque nœud du réseau à la réception d'un message HELLO de l'un de ces nœuds voisins. CBL cherche en permanence à maintenir et prolonger la chaîne de nœuds branche et des liens feuille-branche. Une des difficultés liée à la forte mobilité des nœuds des VANETs est de garantir une certaine stabilité entre les différents rôles des nœuds du réseau et leurs inter-relations. Il est en effet important d'offrir à la couche application des liens stables le temps des échanges. Cela nécessite en particulier d'atténuer les effets des accélérations et décélérations incessantes des nœuds sur la topologie des VANETs et d'éviter ainsi des permutations intempestives du rôle des nœuds (permutation de type, *i.e.* feuille versus branche) ou encore la recherche par tâtonnement du meilleur lien entre un nœud feuille et des nœuds branche.

Pour garantir cette stabilité, un temporisateur (*watchdog timer*), interne à chaque nœud distribué, a été spécifié. Ce temporisateur, ou temps de validité *Validity time* (ou *Vtime*), intervient pour différer des décisions et vérifier qu'elles sont encore valides à l'issu du délai d'expiration (*timeout*). Il permet aux nœuds de prendre des décisions dans leur référentiel temporel local : chaque datation de réception d'un message est relative à l'horloge du nœud récepteur. Notons que la notion de temps de validité *Validity time* existe dans les protocoles de routage des réseaux ad hoc sur lesquels CBL peut être mis en œuvre (par exemple OLSR).

Validity time ou Vtime indique la période de validité d'une information contenue dans le dernier message HELLO reçu, à partir de sa date de réception. À l'issue de la période d'expiration **Vtime**, l'algorithme déclenche, si les conditions sont satisfaites, les actions suivantes :

- un nœud N_i (feuille ou branche), sans nouvelle réception d'un message HELLO en provenance d'un nœud voisin à un saut N_j , le supprime de sa table des voisins ;
- un nœud feuille N_i , sans nouvelle réception d'un message HELLO en provenance du nœud branche auquel il est abonné (@BranchChoice), se désabonne (@BranchChoice=0) ;
- un nœud branche N_i , sans nouvelle réception d'un message HELLO en provenance du nœud chaîne d'un de ses chaînons N_{AV} ou N_{AR} (@ChainUP ou @ChainDO), supprime l'adresse de ce nœud de ses choix (respectivement @ChainUP=0 ou @ChainDO=0) ;
- un nœud branche N_i , sans nouvelle réception d'un message HELLO en provenance de ses nœuds feuille abonnés et de ses deux nœuds branche aval et amont (N_{AV} ou N_{AR}), prend le statut de feuille ;

Trois coefficients C1, C2 et C3 servent à pondérer l'attente d'observation du voisinage à un saut d'un nœud :

- **C1** est un coefficient multiplicateur de *HELLO_INTERVAL*. À l'issue de la période d'expiration $C1 * HELLO_INTERVAL$, un nœud feuille N_i , n'ayant pas détecté de nœud branche dans son voisinage à un saut, élit un nœud feuille voisin N_j au titre de nœud branche.
- **C2** est un coefficient multiplicateur de *HELLO_INTERVAL*. À l'issue de la période d'expiration $C2 * HELLO_INTERVAL$, un nœud branche N_i , n'ayant pas détecté de nœud branche en amont dans son voisinage à un saut, élit un nœud feuille voisin N_j au titre de nœud branche de sorte à créer un chaînon de chaîne. Pour prévenir les configurations où un nœud branche d'une chaîne ne choisisse par défaut un nœud récemment transformé en branche par un nœud feuille, la période d'observation du nœud branche de la chaîne doit avoir une durée plus courte que celle d'un nœud feuille : $C2 * HELLO_INTERVAL < C1 * HELLO_INTERVAL$, soit $C2 < C1$.
- **C3** est un coefficient multiplicateur de *HELLO_INTERVAL*. $C3 * HELLO_INTERVAL$ est une période de garde qui s'applique à l'instant où un nœud branche devient feuille. Ce nœud feuille ne pourra accepter une requête d'élection, au titre de nœud branche, d'un nœud voisin qu'à expiration de cette période de garde.

Deux autres coefficients L1 et L2, tiennent compte de la portée maximale théorique de communication radio P . Cette portée est un invariant de la technologie de communication utilisée. Un troisième, L3 tient compte de la fréquence d'envoi des messages HELLO. Un quatrième α , est un coefficient de pondération des métriques *ChainTime* et *Degree*.

- **L1** intervient dans le critère de changement d'abonnement d'un nœud feuille N_i à un certain nœud branche relais. Soient N_r le nœud relais actuel de N_i et CT_{ri} son temps de connexion à N_i . Soient N_j un nœud branche du voisinage à un saut de N_i qui a une liaison symétrique avec N_i depuis au moins **Vtime** et CT_{ji} son temps de connexion à N_i . Le critère de changement d'abonnement est défini par la formule 2.3.

$$CT_{ji} - CT_{ri} > \frac{L1 * P}{V_{N_i}} \quad (2.3)$$

Si le critère 2.3 est satisfait, alors le nœud N_i enregistre l'adresse de N_j dans @BranchChoice. Ainsi, le nœud N_i change de nœud branche lorsqu'un nœud branche de son voisinage lui procure un délai supplémentaire en termes de temps de connexion.

- **L2** et α interviennent comme critères de décision dans le choix de l'élection d'un maillon de chaîne par un nœud branche N_i au titre de nœud branche N_{AR} . Ils servent à départager deux nœuds candidats N_j ou N_k . Le critère se décline ainsi :
 - si et seulement si un seul des deux candidats à un *ChainTime* supérieur au temps que mettrait le nœud à parcourir la distance $L2 * P$ proportionnelle à la portée maximale P du nœud dans l'hypothèse où le nœud N_i garde une vitesse constante ($ChainTime > \frac{L2 * P}{V_{N_i}}$), il est choisi ;
 - si chacun des *ChainTime* des deux candidats est tel que $ChainTime > \frac{L2 * P}{V_{N_i}}$, alors le candidat dont le degré (*Degree*) est le plus élevé est choisi ; en cas d'égalité, le candidat dont le message HELLO a été réceptionné par N_i en premier est choisi ;
 - si les valeurs des *ChainTime* des deux candidats sont inférieures ($ChainTime < \frac{L2 * P}{V_{N_i}}$), alors le coefficient α intervient dans le calcul de la métrique de décision M_α (formule 2.4) du choix du maillon. Le candidat dont la valeur M_α est supérieure est choisi ; en cas d'égalité, le candidat dont le message HELLO a été réceptionné par N_i en premier est choisi.

$$M_\alpha(N_j) = \alpha * \frac{ChainTime_{ij}}{\max(ChainTime_{ij}, ChainTime_{ik})} + (1 - \alpha) * \frac{Degree_{ij}}{\max(Degree_{ij}, Degree_{ik})} \quad (2.4)$$

- **L3** intervient dans le critère de changement d'abonnement d'un nœud feuille N_i à son nœud branche relais N_r ou d'un nœud branche N_i à son nœud branche en amont $N_r = N_{AR}$. Soit $CTri$ le temps de connexion de N_r à N_i . La connexion entre N_i et N_r est en voie de devenir caduque si et seulement si $CTri < L3 * HELLO_INTERVAL$. Une recherche proactive est alors enclenchée pour élire avant rupture un meilleur nœud relais.

2.3 Algorithmes de construction de la structure CBL

2.3.1 Procédure d'émission d'un message HELLO

La procédure d'émission est la suivante. À intervalle de durée constante *HELLO_INTERVAL* :

1. Un message HELLO est créé par un nœud N_i .
2. Encapsulé, il est mis dans la file d'attente pour une prochaine transmission.
3. N_i mémorise la date de création du message dans sa variable locale T5.

Le message HELLO de CBL créé contient les valeurs des variables locales du nœud N_i spécifiées en section 2.2.7. Ces valeurs ont été régulièrement mises à jour après chacune des réceptions de messages HELLO en provenance de nœuds voisins qui sont intervenues, dans l'intervalle "*HELLO_INTERVAL*", depuis la précédente création d'un message HELLO par N_i .

2.3.2 Diagramme fonctionnel de CBL à réception d'un message HELLO

À l'initialisation du réseau, aucun groupe n'est formé, tous les nœuds du réseau sont des nœuds feuille. Les messages HELLO échangés par les nœuds enclenchent le mécanisme de création de la topologie hiérarchique. À réception d'un message HELLO en provenance d'un nœud voisin, N_i exécute la procédure illustrée en figure 2.6.

En première tâche, l'algorithme 1 réalise la mise à jour de la table de routage contenant la liste des nœuds voisins à un saut.

Ensuite, selon le type (branche ou feuille) du nœud récepteur N_i , les tâches exécutées seront celles de l'algorithme 5 (si N_i est un nœud branche) ou de l'algorithme 2 (si N_i est un nœud feuille). L'algorithme 5 a pour fonction de réinitialiser le nœud branche N_i en un nœud feuille. L'algorithme 2 a pour fonction de transformer le nœud feuille N_i en un nœud branche si le nœud émetteur du message reçu l'a élu au titre de nœud relais.

À l'issue des algorithmes 2 ou 5, selon la nature résultante du nœud récepteur N_i , feuille ou branche, l'algorithme 3 (si N_i est un nœud feuille) ou l'algorithme 4 (si N_i est un nœud branche) est exécuté. L'algorithme 3 permet au nœud feuille N_i de réaliser, vérifier ou maintenir l'élection de son nœud branche relais associé. L'algorithme 4 permet au nœud branche N_i de réaliser, vérifier ou maintenir l'élection des nœuds branche de sa chaîne, en amont et en aval du sens de circulation du trafic routier.

Chacun des algorithmes cités est maintenant détaillé dans les sections 2.3.3 à 2.3.7.

2.3.3 Mise à jour de la table des voisins (algorithme 1)

N_i met à jour sa table des voisins à la réception d'un message HELLO d'un voisin N_j dans l'algorithme 1. La procédure est la suivante :

- Tout d'abord, N_i recherche son adresse $@N_i$ dans la liste des voisins, contenue dans le message HELLO (lignes 5-9). Si elle y figure, la liaison entre les deux nœuds est symétrique. Dans le cas contraire, la liaison est asymétrique. Si la liaison passe du statut asymétrique à symétrique, N_i mémorise l'instant dans le champ Date 1, associé à N_j , de sa table locale des voisins à un saut (lignes 2-4).
- Ensuite, N_i recherche l'adresse $@N_j$ dans sa table locale des voisins à un saut (lignes 10-14). Si elle y figure, N_i rafraîchit les champs associés à N_j avec la fonction UPDATELOCALNEIGHBORTABLE (positions X_j et Y_j , vitesse V_j , angle de braquage σ_j , Date 2, type du nœud, choix de nœud branche relais @HELLO.BranchChoice, et choix de nœud branche pour la chaîne @HELLO.ChainDO et @HELLO.ChainUP). Sinon N_i ajoute et remplit une nouvelle entrée dans sa table des voisins avec la fonction ADDLOCALNEIGHBORTABLE.
- Enfin, N_i contrôle, pour chacun de ses voisins connus, si la date de réception inscrite (Date 2) n'est pas périmée vis-à-vis de VTime. Lorsque c'est le cas, le voisin concerné est supprimé de la table avec la fonction CLEARLOCALNEIGHBORTABLE (lignes 15-17).



FIGURE 2.6 – Procédure utilisée lors de la réception d’un message HELLO

Algorithm 1 Mise à jour de la table des voisins

```

1: procedure UPDATE_NEIGHBOR_TABLE( $N_i, N_j$ )
2:   if  $LinkType.N_i(N_j) == ASYM$  AND  $@N_i \in HELLO.NeighborTable.N_j$  then
3:      $Date1.N_i(N_j) \leftarrow T$  ▷ La liaison devient symétrique.
4:   end if
5:   if  $@N_i \in HELLO.NeighborTable.N_j$  then
6:      $LinkType \leftarrow SYM$ 
7:   else
8:      $LinkType \leftarrow ASYM$ 
9:   end if
10:  if  $@N_j \in Local.NeighborTable.N_i$  then
11:    UPDATELOCALNEIGHBORTABLE( $N_i, @N_j, HELLO.X_j, HELLO.Y_j, HELLO.V_j, HELLO.\sigma_j,$ 
     $Date\ 2=T, HELLO.type.N_j, @HELLO.BranchChoice, @HELLO.ChainDO, @HELLO.ChainUP$ )
12:  else
13:    ADDLOCALNEIGHBORTABLE( $N_i, @N_j, HELLO.X_j, HELLO.Y_j, HELLO.V_j, HELLO.\sigma_j,$ 
     $Date\ 2=T, HELLO.type.N_j, @HELLO.BranchChoice, @HELLO.ChainDO, @HELLO.ChainUP$ )
14:  end if
15:  while  $\exists N_k \in Local.NeighborTable.N_i$  with  $T > Date2.N_i(N_k) + Vtime$  do
16:    CLEARLOCALNEIGHBORTABLE( $N_i, N_k$ )
17:  end while
18: end procedure

```

SYM : Symétrique, $ASYM$: Asymétrique

2.3.4 Procédure de changement de type, de feuille à branche (algorithme 2)

Le nœud feuille N_i contrôle s'il doit changer de type dans l'algorithme 2. La procédure est la suivante :

- Tout d'abord, N_i vérifie s'il a été branche et qu'il est un nœud feuille depuis une durée supérieure à $C3 * HELLO_INTERVAL$. Si c'est le cas, il recherche son adresse $@N_i$ dans l'un des champs $@HELLO.BranchChoice$, $@HELLO.ChainUP$ ou $@HELLO.ChainDO$ contenus dans le message HELLO. Si l'adresse y figure, N_i devient un nœud branche (lignes 3). Il met à jour ses variables locales (Type, T1, T4, T6, T7, T8, section 2.2.6) (lignes 4-11). Il indique qu'il a été élu branche par ce nœud voisin en mettant à jour la variable "Elected", associé à ce voisin, dans sa table locale des voisins à un saut.
- Par ailleurs, si le voisin est un nœud branche et s'il a élu N_i en tant que nœud amont ($@HELLO.ChainDO$) alors N_i inscrit l'adresse de ce voisin dans la variable locale $@ChainUP$ (lignes 12-16) et met à jour ses variables locales (T3, T7, T10).

Algorithm 2 Changement de type : de feuille à branche

```

1: procedure TURN_LEAF_INTO_BRANCH( $N_i$ )
2:   if [ $T > T4b + C3 * HELLO\_INTERVAL$ ] AND [ $@HELLO.BranchChoice.N_j == @N_i$  OR
    $@HELLO.ChainDO.N_j == @N_i$ ] then
3:      $Type.N_i \leftarrow 1$  ▷  $N_i$  devient branche
4:     [ $T1, T2, T3, T4$ ]  $\leftarrow [T, 0, 0, T]$  ▷ Mise à jour des horodatages
5:     [ $T6, T7, T8, T9, T10, T11$ ]  $\leftarrow [T, T, 0, 0, 0, 0]$ 
6:      $@BranchChoice.N_i \leftarrow 0$ 
7:      $@ChainDO.N_i \leftarrow 0$ 
8:      $DegreeChainDO.N_i \leftarrow 0$ 
9:      $@ChainUP.N_i \leftarrow 0$ 
10:     $DegreeChainUP.N_i \leftarrow 0$ 
11:     $Elected.N_i(N_j) \leftarrow 1$ 
12:    if  $Type.N_j == 1$  AND  $@HELLO.ChainDO.N_j == @N_i$  AND  $RELATIVEPOSITION$ 
   ( $N_i, N_j) == 0$  then
13:       $@ChainUP.N_i \leftarrow @N_j$  ▷ Ajout de  $N_j$  à la chaîne
14:       $DegreeChainUP.N_i \leftarrow NUMBERNEWNEIGHBORS(N_i, N_j)$ 
15:      [ $T3, T7, T10$ ]  $\leftarrow [T, 0, T]$ 
16:    end if
17:  end if
18: end procedure

```

2.3.5 Traitement type feuille (algorithme 3)

Un nœud feuille doit élire le nœud branche auquel il se rattache. L'adresse de ce nœud est enregistrée dans sa variable $@BranchChoice$ et cette information est ensuite partagée avec ses voisins à un saut par l'intermédiaire des messages HELLO qu'il émet. La stratégie proposée dans CBL version 2 est celle-ci : un nœud feuille se rattache au premier nœud branche qu'il détecte. Cependant, il arrive qu'il en détecte plusieurs ou, au contraire, n'en détecte pas.

- S'il en détecte plusieurs, le nœud branche se déplaçant dans la même direction, faisant partie d'une chaîne et offrant le plus long temps de connexion CT est choisi. Ces deux critères de choix garantissent que le nœud relais élu à une date donnée parmi plusieurs nœuds branche détectés sera celui qui aura avec le nœud feuille électeur la durée de liaison radio la plus longue.
- S'il n'en détecte aucun, après une période d'attente de $C1 * HELLO_INTERVAL$, le nœud choisit, pour nœud branche relais, le nœud feuille voisin qui circule dans la même direction et dont la vitesse est la plus faible. Cette élection d'un nœud feuille au titre de nœud branche initie la création de la structure d'une chaîne.

Le nœud feuille N_i exécute l'algorithme 3. La procédure est la suivante :

- Tout d'abord, N_i compare l'adresse du nœud expéditeur avec celle de son champ @BranchChoice. Si elles sont identiques, N_i contrôle dans le message reçu que ce voisin est toujours un nœud branche. Si tel est le cas, il met à jour ses variables locales temporelles (T1 et T8) (lignes 3-5). Sinon, il l'invalidé au titre de nœud branche relais (lignes 6-9).
- Ensuite, N_i vérifie s'il a déjà choisi un nœud branche auquel se rattacher. Dans ce cas, N_i contrôle si les informations concernant son nœud relais ne sont pas obsolètes (variable T1). Si elles le sont, il l'invalidé au titre de nœud branche relais (lignes 6-9).
- Lorsque le nœud émetteur est un nœud branche, si N_i n'a pas déjà de nœud relais, il le choisit et met à jour ses variables locales temporelles (T1, T8 et T11) (lignes 12-15). Sinon, il compare les informations de direction, d'appartenance à une chaîne et de temps de connexion du nœud relais actuel avec celles du nœud émetteur avec la fonction COMPAREBRANCHNODE. Il modifie son choix si le nouveau nœud est un meilleur candidat (lignes 16-21).
- Lorsque le nœud émetteur est un nœud feuille, N_i , une fois la période $C1 * HELLO_INTERVAL$ d'observation du voisinage écoulée, choisit ce nœud émetteur pour relais s'il n'avait pas de nœud relais auparavant. S'il en avait un et n'avait pas déjà informé son voisinage de son choix par l'émission d'un message HELLO ou bien si le nœud élu n'était pas dans sa direction, alors il procède à la comparaison des informations de direction et de vitesse du nœud relais actuel avec celles du nœud émetteur avec la fonction COMPARELEAFNODE. Si le nouveau nœud est un meilleur candidat, N_i modifie son choix (lignes 23-35).

Algorithm 3 Traitement de type feuille

```

1: procédure LEAF_PROCESSING( $N_i, N_j$ )
2:   if @BranchChoice. $N_i \neq 0$  then
3:     if @BranchChoice. $N_i == @N_j$  AND HELLO.Type. $N_j == 1$  then
4:       [T1, T8]  $\leftarrow$  [T, 0]
5:     end if
6:     if  $T > T1 + Vtime$  OR (@BranchChoice. $N_i == @N_j$  AND HELLO.Type. $N_j == 0$ ) then
7:       @BranchChoice. $N_i \leftarrow 0$  ▷ Perte du nœud relais
8:       [T1, T8, T11]  $\leftarrow$  [0, T, 0]
9:     end if
10:  end if
11:  if Type. $N_j == 1$  then ▷  $N_j$  est un nœud branche
12:    if @BranchChoice. $N_i == 0$  then
13:      @BranchChoice. $N_i \leftarrow @N_j$ 
14:      [T1, T8, T11]  $\leftarrow$  [T, 0, T]
15:    else
16:      @ $N_k \leftarrow$  BranchChoice. $N_i$ 
17:      if COMPAREBRANCHNODE(@ $N_j, @N_k == 1$  then
18:        @BranchChoice. $N_i \leftarrow @N_j$  ▷ Changement de nœud relais
19:        [T1, T8, T11]  $\leftarrow$  [T, 0, T]
20:      end if
21:    end if
22:  else ▷  $N_j$  est un nœud feuille
23:    if  $T > T8 + C1 * HELLO\_INTERVAL$  then ▷ Attente d'un délai d'observation du voisinage
24:      @ $N_k \leftarrow$  BranchChoice. $N_i$ 
25:      if @ $N_k == 0$  AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  AND  $T > Date1.N_i(N_j) + Vtime$  then
26:        @BranchChoice. $N_i \leftarrow @N_j$  ▷ Changement de nœud relais
27:        [T1, T8, T11]  $\leftarrow$  [T, 0, T]
28:      end if

```

```

29:         if @Nk! = 0 AND [(Type.Nk == 0 AND T5 < T11) OR (Type.Nk == 1 AND
    |σNj - σNi| < σmax AND |σNk - σNi| >= σmax)] then
30:             if COMPARELEAFNODE(@Nj, @Nk) == 1 then
31:                 @BranchChoice.Ni ← @Nj                                ▷ Changement de nœud relais
32:                 [T1, T8, T11] ← [T, 0, T]
33:             end if
34:         end if
35:     end if
36: end if
37: end procedure

```

```

function FLAG=COMPAREBRANCHNODE(@Nj, @Nk) (Retourne vrai si Nj est un meilleur candidat
au titre de nœud branche que celui @Nk précédemment élu) C
2:     if T < Date1.Ni(Nj) + Vtime then
3:         return False                                ▷ Vérification de la stabilité de la liaison symétrique Ni-Nj
4:     break
5:     end if
6:     if |σNj - σNi| >= σmax AND |σNk - σNi| < σmax then                                ▷ Nj est dans une autre direction
7:         return False
8:     break
9:     end if
10:    if |σNj - σNi| < σmax AND |σNk - σNi| >= σmax then                                ▷ Nk est dans une autre direction
11:        return True                                    ▷ Nj est un meilleur candidat
12:    break
13:    end if
14:    if N.ChainUP.Ni(Nk)==0 AND N.ChainDO.Ni(Nk)==0 then
15:        if N.ChainUP.Ni(Nj)!=0 OR N.ChainDO.Ni(Nj)!=0 then
16:            return True                                ▷ Nj appartient à une chaîne mais pas Nk
17:            break
18:        end if
19:    end if
20:    if CT(Ni,Nk)>L3 * HELLO_INTERVAL then
21:        return False                                ▷ La liaison avec le nœud relais actuel est temporellement stable
22:    break
23:    end if
24:    if CT(Ni,Nj)-CT(Ni,Nk) > L1 * P/VNi then                                ▷ voir formule 2.3
25:        return True                                    ▷ Nj est un meilleur candidat
26:    else
27:        return False
28:    end if
end function

```

```

1: function FLAG=COMPARELEAFNODE(@Nj, @Nk) (Retourne vrai si Nj est un meilleur candidat au
   titre de nœud branche que celui @Nk précédemment élu)
2:   if T < Date1.Ni(Nj) + Vtime then
3:     return False                                ▷ Vérification de la stabilité de la liaison symétrique Ni-Nj
4:     break
5:   end if
6:   if |σNj - σNi| >= σmax AND |σNk - σNi| < σmax then
7:     return False                                ▷ Nj est dans une autre direction
8:     break
9:   end if
10:  if Type.Nk == 1 then
11:    return True                                  ▷ Nj est un meilleur candidat
12:    break
13:  end if
14:  if VNj < VNk then
15:    return True                                  ▷ Nj est un meilleur candidat
16:  else
17:    return False
18:  end if
19: end function

```

2.3.6 Traitement type branche (algorithme 4)

Un nœud branche participe à la création d'une chaîne : il élit, parmi ces nœuds branche voisins circulant dans la même direction, un nœud branche en amont (N_{AR}) et un nœud branche (N_{AV}) en aval de sa position si celui-ci l'a élu. L'adresse de ces nœuds est enregistrée respectivement dans ses variables locales @ChainDO et @ChainUP et ces informations sont ensuite partagées avec ses voisins à un saut par l'intermédiaire des messages HELLO qu'il émet. La stratégie proposée dans CBL est celle-ci : un nœud branche se rattache au premier nœud branche qu'il détecte en amont de sa position. Cependant, il arrive qu'il en détecte plusieurs ou, au contraire, n'en détecte pas.

- S'il en détecte plusieurs et qu'il n'a pas encore averti ses voisins de son choix, alors le nœud branche qui se déplace dans la même direction, qui fait partie d'une chaîne, qui offre le plus long temps de chaîne (*ChainTime*) et dont le degré est le plus élevé est élu.
- S'il n'en détecte aucun, après une période d'attente de $C2 * HELLO_INTERVAL$, le nœud élit, pour nœud branche amont, le nœud feuille qui répond aux mêmes critères que précédemment. Cette élection d'un nœud feuille au titre de nœud branche prolonge la chaîne d'un maillon.

Lorsque le nœud branche a ajouté un maillon à sa chaîne, il ne modifie plus son choix tant que l'ordonnement spatial des véhicules amont et aval relativement à ce nœud branche n'a pas subi de permutations.

La procédure de l'algorithme 4 exécutée par un nœud branche N_i est la suivante :

- Tout d'abord, si N_i a déjà choisi un nœud branche amont ou aval auquel se rattacher, il contrôle que leurs informations ne sont pas obsolètes (respectivement variables T2 ou T3). Si elles le sont, il l'invalide au titre de nœud branche amont respectivement aval (lignes 2-11).
- Puis, N_i compare l'adresse du nœud expéditeur avec celles contenues dans ses champs @ChainDO et @ChainUP. Si cette adresse est identique à @ChainDO (respectivement @ChainUP), N_i vérifie que le voisin qui envoie le message l'a élu dans sa chaîne (@HELLO.ChainUP respectivement, @HELLO.ChainDO). Si le voisin a élu un autre nœud alors N_i l'invalide au titre de nœud branche amont, respectivement aval (lignes 2-11).
- Ensuite, N_i compare l'adresse du nœud expéditeur avec celles contenues dans ses champs @ChainDO et @ChainUP. Si l'une d'elles est identique à un des deux champs, N_i contrôle, via les informations du message reçu avec la fonction CHAINCHANGENODE, si ce nœud voisin est encore un nœud branche, dans sa direction de circulation et que leurs positions relatives n'ont pas permutées (dépassement de l'un des véhicules). Si ces trois conditions sont réunies, N_i met à jour ses variables locales associées respectivement aux nœuds de la chaîne en amont (T2, T6, T9 et DegreeChainDO) ou en aval (T3,

T7, T10 et DegreeChainUP) (lignes 14-21). Sinon, il l'invalide au titre de nœud branche de sa chaîne (lignes 22-38). Si l'adresse du nœud expéditeur est celle du champ @ChainDO et que ce nœud se situe maintenant en aval, N_i l'invalide en tant que maillon de chaîne. Si N_i constate que, dans sa table des voisins, il a connaissance du nœud branche en amont qu'avait choisi le nœud expéditeur, que ce voisin est effectivement un nœud branche et, enfin, que leur liaison est symétrique depuis au moins $Vtime$, alors il le sélectionne au titre de nœud branche amont (lignes 27-31).

- Puis, N_i vérifie si son nœud branche choisi en amont, a un temps de connexion inférieur au seuil $L3 * HELLO_INTERVAL$ (lignes 39-47). Dans ce cas, la liaison avec ce nœud amont va devenir caduque. N_i initie une recherche proactive d'un nouveau nœud relais amont de la chaîne. Si le nœud voisin qui envoie le message HELLO est en amont, dans sa direction et apporte la connaissance de nouveaux voisins, alors N_i le choisit et met à jour ses variables internes (T2, T6, T9 et DegreeChainDO).
- Puis, N_i effectue la mise à jour de la variable locale T1 et du champ "Elected", de sa table des voisins associée à l'adresse du nœud expéditeur si son adresse @ N_i figure dans l'un des champs @ChainDO, @ChainUP ou @BrancheChoice du message HELLO du nœud expéditeur (lignes 48-53). Si son adresse est absente de ces champs la valeur du champs "Elected" est mise à zéro.
- Lorsque le nœud expéditeur N_j du message HELLO est un nœud branche, N_i prend connaissance de la position relative de ce nœud N_j par rapport à lui-même.

Si N_j est en aval et qu'il a sélectionné N_i dans sa chaîne alors N_i le sélectionne en retour comme maillon de chaîne aval (lignes 76-82). Si N_i n'a pas de maillon de chaîne dans la direction de ce nœud en amont, il le choisit comme maillon ; il met alors à jour ses variables locales temporelles (T2, T6 et T9) (lignes 57-60). Si, par contre, N_i avait antérieurement sélectionné un nœud N_k en amont comme maillon et que N_j est également en amont, deux cas de figure se présentent :

- si le nœud N_k est en bout de chaîne et si le nœud expéditeur N_j n'est pas un maillon isolé alors N_i sélectionne N_j comme maillon en amont ; il met ensuite à jour ses variables locales temporelles (T2, T6 et T9) lignes 61-64 ;
- sinon, si N_i n'a pas encore diffusé le choix du nœud N_k , il compare son choix antérieur N_k avec la fonction COMPARECHAINNODE à l'aide des coefficients L2 et α avec le nœud expéditeur N_j conformément à la section 2.2.8. Si N_j est un meilleur candidat, alors N_i sélectionne N_j comme maillon en amont ; il met ensuite à jour ses variables locales temporelles (T2, T6 et T9) (lignes 65-71).
- Lorsque le nœud expéditeur N_j du message HELLO est un nœud feuille, alors, une fois la période $C2 * HELLO_INTERVAL$ d'observation du voisinage écoulée, N_i prend connaissance de la position relative de ce nœud N_j par rapport à lui-même . Ensuite, si N_i n'a pas de maillon de chaîne en amont, il le choisit comme maillon ; il met alors à jour ses variables locales temporelles (T2, T6 et T9). Si, par contre, N_i avait antérieurement sélectionné un nœud feuille N_k comme maillon dans la direction du nœud expéditeur en amont, et s'il n'a pas encore diffusé son choix de nœud N_k , il réalise les actions suivantes : il compare son choix antérieur N_k avec la fonction COMPARECHAINNODE à l'aide des coefficients L2 et α avec le nœud expéditeur N_j conformément à la section 2.2.8. Si N_j est un meilleur candidat, alors N_i sélectionne N_j comme maillon en amont ; il met ensuite à jour ses variables locales temporelles (T2, T6 et T9) (lignes 85-91).

Algorithm 4 Traitement type branche

```

1: procedure BRANCH_PROCESSING( $N_i, N_j$ )
2:   if [ $T > T2 + Vtime$  AND  $@ChainDO.N_i! = 0$ ] OR [ $@ChainDO.N_i == @N_j$  AND
    $@HELLO.ChainUP.N_j! = @N_i$  AND  $@HELLO.ChainUP.N_j! = 0$ ] then
3:      $@ChainDO.N_i \leftarrow 0$  ▷ Perte du nœud branche arrière
4:      $DegreeChainDO.N_i \leftarrow 0$ 
5:      $[T2, T6, T9] \leftarrow [0, T, 0]$ 
6:   end if
7:   if [ $T > T3.N_i + Vtime$  AND  $@ChainUP.N_i! = 0$ ] OR [ $@ChainUP.N_i == @N_j$  AND
    $@HELLO.ChainDO.N_j! = @N_i$  AND  $@HELLO.ChainDO.N_j! = 0$ ] then
8:      $@ChainUP.N_i \leftarrow 0$  ▷ Perte du nœud branche avant
9:      $DegreeChainUP.N_i \leftarrow 0$ 
10:     $[T3, T7, T10] \leftarrow [0, T, 0]$ 
11:  end if
12:  if  $@N_j == @ChainUP.N_i$  OR  $@N_j == @ChainDO.N_i$  then
13:     $[Flag, ChoiceUP, ChoiceDO] \leftarrow CHANGECHAINNODE(N_i, N_j)$ 
14:    if  $Flag == 0$  then ▷ Pas de modification dans la chaîne
15:      if  $@ChainDO.N_i == @N_j$  then
16:         $[T2, T6] \leftarrow [T, 0]$ 
17:         $DegreeChainDO.N_i \leftarrow NUMBERNEWNEIGHBORS(N_i, N_j)$ 
18:      else
19:         $[T3, T7] \leftarrow [T, 0]$ 
20:         $DegreeChainUP.N_i \leftarrow NUMBERNEWNEIGHBORS(N_i, N_j)$ 
21:      end if
22:    else ▷ Modification dans la chaîne
23:      if  $ChoiceDO == 0$  then
24:         $@ChainDO.N_i \leftarrow 0$  ▷ Perte du nœud branche arrière
25:         $DegreeChainDO.N_i \leftarrow 0$ 
26:         $[T2, T6, T9] \leftarrow [0, T, 0]$ 
27:      else
28:         $@ChainDO.N_i \leftarrow ChoiceDO$  ▷ Changement du nœud branche arrière
29:         $DegreeChainDO.N_i \leftarrow 0$ 
30:         $[T2, T6, T9] \leftarrow [T, 0, T]$ 
31:      end if
32:      if  $ChoiceUP == 0$  then
33:         $@ChainUP.N_i \leftarrow 0$  ▷ Perte du nœud branche avant
34:         $DegreeChainUP.N_i \leftarrow 0$ 
35:         $[T3, T7, T10] \leftarrow [0, T, 0]$ 
36:      end if
37:    end if
38:  end if
39:  if  $@ChainDO.N_i! = 0$  AND  $RELATIVEPOSITION(N_i, N_j) == 1$  AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  then
40:    if  $CT(N_i, @ChainDO.N_i) < L3 * HELLO\_INTERVAL$  then ▷ Perte imminente du nœud
    branche arrière
41:      if  $COMPARECHAINNODE(N_j, \emptyset) == 1$  then
42:         $@ChainDO.N_i \leftarrow N_j$ 
43:         $DegreeChainDO.N_i \leftarrow NUMBERNEWNEIGHBORS(N_i, N_j)$ 
44:         $[T2, T6, T9] \leftarrow [T, 0, T]$ 
45:      end if
46:    end if
47:  end if

```

```

48:   if @Ni ∈ {@HELLO.ChainUP.Nj, @HELLO.ChainDO.Nj, @HELLO.BranchChoice.Nj}
      then
49:     Elected.Ni(Nj) == 1
50:     T1 ← T
51:   else
52:     Elected.Ni(Nj) ← 0
53:   end if
54:   if Type.Nj == 1 then
55:     if RELATIVEPOSITION(Ni, Nj) == 1 AND |σNj - σNi| < σmax then
56:       if @HELLO.ChainUP.Nj = 0 OR @Ni == @HELLO.ChainUP.Nj then
57:         if @ChainDO.Ni == 0 then
58:           @ChainDO.Ni ← Nj
59:           DegreeChainDO.Ni ← NUMBERNEWNEIGHBORS(Ni, Nj)
60:           [T2, T6, T9] ← [T, 0, T]
61:         else if @HELLO.ChainDO.Nj != 0 AND @N.ChainDO(ChainDO.Ni) == 0 then
62:           @ChainDO.Ni ← Nj
63:           DegreeChainDO.Ni ← NUMBERNEWNEIGHBORS(Ni, Nj)
64:           [T2, T6, T9] ← [T, 0, T]
65:         else if T5 < T9 then
66:           Nk ← @ChainDO.Ni
67:           if COMPARECHAINNODE(Nj, Nk) == 1 then
68:             @ChainDO.Ni ← Nj
69:             DegreeChainDO.Ni ← NUMBERNEWNEIGHBORS(Ni, Nj)
70:             [T2, T6, T9] ← [T, 0, T]
71:           end if
72:         end if
73:       end if
74:     end if
75:     if RELATIVEPOSITION(Ni, Nj) == 0 AND |σNj - σNi| < σmax then
76:       if @Ni == @HELLO.ChainDO.Nj then
77:         if @ChainUP.Ni == 0 OR @HELLO.ChainUP.Nj == @ChainUP.Ni then
78:           @ChainUP.Ni ← Nj
79:           DegreeChainUP.Ni ← NUMBERNEWNEIGHBORS(Ni, Nj)
80:           [T3, T7, T10] ← [T, 0, T]
81:         end if
82:       end if
83:     end if
84:   else
85:     if RELATIVEPOSITION(Ni, Nj) == 1 AND T > T6 + C2 * HELLO_INTERVAL AND
      [@ChainDO.Ni == 0 OR (@ChainDO.Ni != 0 AND Type.@ChainDO.Ni == 0 AND T5 < T9)] then
86:       if COMPARECHAINNODE(Nj, @ChainDO) == 1 then
87:         @ChainDO.Ni ← @Nj
88:         DegreeChainDO.Ni ← NUMBERNEWNEIGHBORS(Ni, Nj)
89:         [T1, T2, T6, T9] ← [T, T, 0, T]
90:       end if
91:     end if
92:   end if
93: end procedure

```

```

1: function [FLAG, CHOICEUP, CHOICEDO]= CHANGECHAINNODE( $N_i, N_j$ )( Retourne Flag=0 s'il
   n'y a pas de modification dans la chaîne ; sinon, retourne Flag=1 avec les adresses des nœuds branche
   amont et aval respectivement dans ChoiceDO et ChoiceUP.)
2:   ChoiceUP  $\leftarrow$  @ChainUP. $N_i$ 
3:   ChoiceDO  $\leftarrow$  @ChainDO. $N_i$ 
4:   Flag  $\leftarrow$  0
5:   if @ $N_j$ ==@ChainDO. $N_i$  AND RELARIVEPOSITION( $N_i, N_j$ )==0) then  $\triangleright N_i$  a été dépassé par
    $N_j$ 
6:     Flag  $\leftarrow$  1
7:     ChoiceDO  $\leftarrow$  0
8:     @ $N_k$   $\leftarrow$  @HELLO.ChainDO. $N_j$ 
9:     if @ $N_k$   $\in$  NeighborTable. $N_i$  AND Date1. $N_i$ (@ $N_k$ ) + Vtime < T AND @ $N_k$ ! = @ $N_i$  AND
   RELARIVEPOSITION( $N_i, N_k$ )==0) AND Type. $N_k$ ==1 then
10:       ChoiceDO  $\leftarrow$  @ $N_k$ 
11:     end if
12:     break
13:   end if
14:   if @ $N_j$ ==@ChainUP. $N_i$  AND RELARIVEPOSITION( $N_i, N_j$ )==1) then  $\triangleright N_i$  a dépassé  $N_j$ 
15:     Flag  $\leftarrow$  1
16:     ChoiceDO  $\leftarrow$  0
17:     ChoiceUP  $\leftarrow$  0
18:     break
19:   end if
20:   if Type. $N_j$  == 1 AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  then
21:     Flag  $\leftarrow$  0
22:   else
23:     Flag  $\leftarrow$  1
24:     if @ $N_j$ ==@ChainDO. $N_i$  then
25:       ChoiceDO  $\leftarrow$  0
26:     else
27:       ChoiceUP  $\leftarrow$  0
28:     end if
29:   end if
30: end function

```

```

1: function NUMBERNEWNEIGHBORS( $N_i, N_j$ )(Retourne le nombre de nouveaux voisins apportés par
   le nœud  $N_j$  pour le nœud  $N_i$ )
2:   Degree  $\leftarrow$  0
3:   n  $\leftarrow$  0
4:   while n < Size(HELLO.NeighborTable. $N_j$ ) do
5:      $N_k$   $\leftarrow$  HELLO.NeighborTable. $N_j$ (n)
6:     if  $N_k$  !=  $N_i$  then
7:       if LinkType( $N_j, N_k$ )==SYM then
8:         if @ $N_k$   $\notin$  Local.NeighborTable. $N_i$  then
9:           Degree  $\leftarrow$  Degree+1
10:        end if
11:      end if
12:    end if
13:    n  $\leftarrow$  n + 1
14:  end while
15:  return Degree
16: end function

```

```

1: function FLAG=COMPARECHAINNODE( $N_i, N_j, N_k$ )(Retourne vrai si le nœud  $N_j$  est un meilleur
   candidat que  $N_k$  pour le choix de chaîne avant ou arrière.)
2:   if  $T < Date1.N_i(N_j) + Vtime$  then
3:     return False ▷ Vérification de la stabilité de la liaison symétrique  $N_i-N_j$ 
4:     break
5:   end if
6:   if NUMBERNEWNEIGHBORS( $N_i, N_j$ )==0 then
7:     return False ▷  $N_j$  n'apporte pas de nouveau voisin
8:     break
9:   end if
10:  if  $|\sigma_{N_j} - \sigma_{N_i}| \geq \sigma_{max}$  then
11:    return False ▷  $N_j$  est dans une autre direction
12:    break
13:  end if
14:  if  $N_k == 0$  then
15:    return True ▷  $N_j$  est un meilleur candidat
16:    break
17:  end if
18:  if Type. $N_j == 1$  AND Type. $N_k == 0$  then
19:    return True ▷  $N_j$  est un meilleur candidat
20:    break
21:  end if
22:  if Type. $N_k == 1$  AND Type. $N_j == 0$  then
23:    return False
24:    break
25:  end if
26:  if Type. $N_k == 1$  then
27:    if RELATIVEPOSITION( $N_i, N_j$ )==1 then
28:      if N.ChainDO. $N_i(N_j) != 0$  AND N.ChainDO. $N_i(N_k) == 0$  then
29:        return True ▷  $N_j$  est un meilleur candidat
30:        break
31:      end if
32:    else
33:      if N.ChainUP. $N_i(N_j) != 0$  AND N.ChainUP. $N_i(N_k) == 0$  then
34:        return True ▷  $N_j$  est un meilleur candidat
35:        break
36:      end if
37:    end if
38:  end if
39:  if CHAINTIME( $N_i, N_j$ )=>L2*P/V AND CHAINTIME( $N_i, N_k$ )=>L2*P/V then
40:    if NUMBERNEWNEIGHBORS( $N_i, N_j$ )>NUMBERNEWNEIGHBORS( $N_i, N_k$ ) then
41:      return True ▷  $N_j$  est un meilleur candidat
42:    else
43:      return False
44:    end if
45:  else if CHAINTIME( $N_i, N_j$ )=>L2*P/ $V_i$  AND CHAINTIME( $N_i, N_k$ )<L2*P/ $V_i$  then
46:    return True ▷  $N_j$  est un meilleur candidat
47:  else if CHAINTIME( $N_i, N_j$ )<L2*P/ $V_i$  AND CHAINTIME( $N_i, N_k$ )=>L2*P/ $V_i$  then
48:    return False ▷  $N_j$  est un meilleur candidat
49:  else
50:    Compute  $M_\alpha(N_j), M_\alpha(N_k)$  ▷ Voir équation 2.4
51:    if  $M_\alpha(N_j) > M_\alpha(N_k)$  then
52:      return True ▷  $N_j$  est un meilleur candidat
53:    else
54:      return False
55:    end if
56:  end if
57: end function

```

```

1: function FLAG=RELARIVEPOSITION( $N_i, N_j$ )(Retourne vrai si le nœud  $N_j$  est derrière le nœud  $N_i$ )
2:   if  $\cos(\sigma_{N_i}) * (X_j - X_i) + \sin(\sigma_{N_i}) * (Y_j - Y_i) < 0$  then
3:     return True
4:   else
5:     return False
6:   end if
7: end function

```

```

1: function TIME=CHAINTIME( $N_i, N_j$ )(Retourne le temps durant lequel le nœud  $N_j$  peut rester dans
   la chaîne)
2:   if [RELARIVEPOSITION( $N_i, N_j$ )==1] AND  $V_j < V_i$  OR [RELARIVEPOSITION( $N_i, N_j$ )==0]
   AND  $V_j > V_i$  then
3:     return CT( $N_i, N_j$ )
4:   else
5:     return  $\min(\text{CT}(\mathbf{N}_i, \mathbf{N}_j), \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} / |V_i - V_j|)$ 
6:   end if
7: end function

```

```

1: function CT=CT( $N_i, N_j$ )(Retourne le temps de connexion entre le nœud  $N_i$  et  $N_j$ )
2:    $a \leftarrow V_i * \cos(\sigma_i) - V_j * \cos(\sigma_j)$ 
3:    $b \leftarrow X_i - X_j$ 
4:    $c \leftarrow V_i * \sin(\sigma_i) - V_j * \sin(\sigma_j)$ 
5:    $d \leftarrow Y_i - Y_j$ 
6:   return  $\frac{-(ab+cd) + \sqrt{(a^2+c^2)*P^2 - (ab-bc)^2}}{a^2+c^2}$ 
7: end function

```

2.3.7 Procédure de changement de type, de branche à feuille (algorithme 5)

Le nœud branche N_i réalise une permutation du type branche en type feuille dans l'algorithme 5 si au moins l'une des conditions suivantes est satisfaite :

- le temps de validité écoulé depuis la datation (mémorisée dans T1) de la dernière réception d'un message HELLO en provenance d'un nœud l'ayant élu est dépassé (lignes 3-5) ;
- s'il vient de dépasser le nœud qu'il avait choisi comme nœud branche à l'avant de sa chaîne (lignes 6-8) ;
- s'il a détecté qu'un nœud branche en aval de sa position a déjà élu un autre nœud branche qui est en amont de sa position (lignes 9-14).

N_i devient un nœud feuille et met à jour ses variables locales (Type, T1, T2, T3, T4b, T6, T7, T8, @BranchChoice, et @ChainUP, DegreeChainUP, DegreeChainDO section 2.2.6) (lignes 15-27). S'il vient de dépasser le nœud qu'il avait choisi comme nœud branche à l'avant de sa chaîne, alors N_i continue à mémoriser son choix de nœud branche en amont (@ChainDO). Cette information permettra à ce voisin de sélectionner directement ce nœud branche en amont. Dans le cas contraire, il le réinitialise à zéro (lignes 23-25).

Algorithm 5 Changement de type : de branche à feuille

```

1: procedure TURN_BRANCH_INTO_LEAF( $N_i$ )
2:   Change  $\leftarrow$  0
3:   if  $T > T1.N_i + Vtime$  AND  $T > T4.N_i + Vtime$  then
4:     Change  $\leftarrow$  1
5:   end if
6:   if RELARIVEPOSITION( $N_i, N_j$ )==1 AND @ChainUP. $N_i$ ==@ $N_j$  then
7:     Change  $\leftarrow$  1
8:   end if
9:   if RELARIVEPOSITION( $N_i, N_j$ )==0 AND @ChainUP. $N_i$ ==0 AND @HELLO.ChainDO. $N_j$ !=0
AND @HELLO.ChainDO. $N_j$ !=@ $N_i$  AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  then
10:     $N_k \leftarrow$  @HELLO.ChainDO. $N_j$ 
11:    if @ $N_k \in Local.NeighborTable.N_i$  AND Type. $N_k$ ==1 AND RELARIVEPOSITION( $N_i, N_k$ )==1
AND  $|\sigma_{N_k} - \sigma_{N_i}| < \sigma_{max}$  then
12:      Change  $\leftarrow$  1
13:    end if
14:  end if
15:  if Change==1 then
16:    Type. $N_i \leftarrow$  0 ▷  $N_i$  devient feuille
17:    [T1, T2, T3, T4b]  $\leftarrow$  [0, 0, 0, T] ▷ Mise à jour des horodatages
18:    [T6, T7, T8, T9, T10, T11]  $\leftarrow$  [0, 0, T, 0, 0, 0]
19:    @BranchChoice  $\leftarrow$  0
20:    @ChainUP. $N_i \leftarrow$  0
21:    DegreeChainDO. $N_i \leftarrow$  0
22:    DegreeChainUP. $N_i \leftarrow$  0
23:    if  $T > T1.N_i + Vtime$  AND  $T > T4.N_i + Vtime$  then
24:      @ChainDO. $N_i \leftarrow$  0
25:    end if
26:     $\forall N_k \in HELLO.NeighborTable.N_i, Elected.N_i(N_k) \leftarrow$  0
27:  end if
28: end procedure

```

2.3.8 Diagrammes d'interactions temporelles

2.3.8.1 Étapes de découverte et d'élection de nœud branche par des nœuds feuille

Les étapes de la découverte du voisinage et de l'élection des nœuds branche par des nœuds feuille sont présentées dans la figure 2.7. Au départ, les nœuds N1, N2 et N3 sont des nœuds feuille. Au fur et à mesure de leurs échanges de messages HELLO, les nœuds remplissent leur table de voisins à un saut avec les nœuds voisins qu'ils découvrent. Les nœuds actualisent constamment la nature des liaisons (ASYM, SYM) qu'ils entretiennent avec leurs voisins. Dans l'exemple de la figure 2.7, le nœud N2 découvre le nœud N1 ; puis, les nœuds N1 et N3 découvrent le nœud N2 ; enfin, le nœud N2 découvre les nœuds N1 et N3. Après un délai temporel $C1 * HELLO_INTERVAL$, le nœud N1 sélectionne le nœud N2 comme son nœud branche relais et informe ses voisins de ce choix via un message HELLO. En le recevant, le nœud N2 devient un nœud branche, puis en informe ses voisins. Le nœud N3 choisit alors le nœud N2 comme relais. À la fin de ces échanges, les nœuds N1 et N3 sont des nœuds feuille et le nœud N2 est un nœud branche élu par les nœuds N1 et N3.

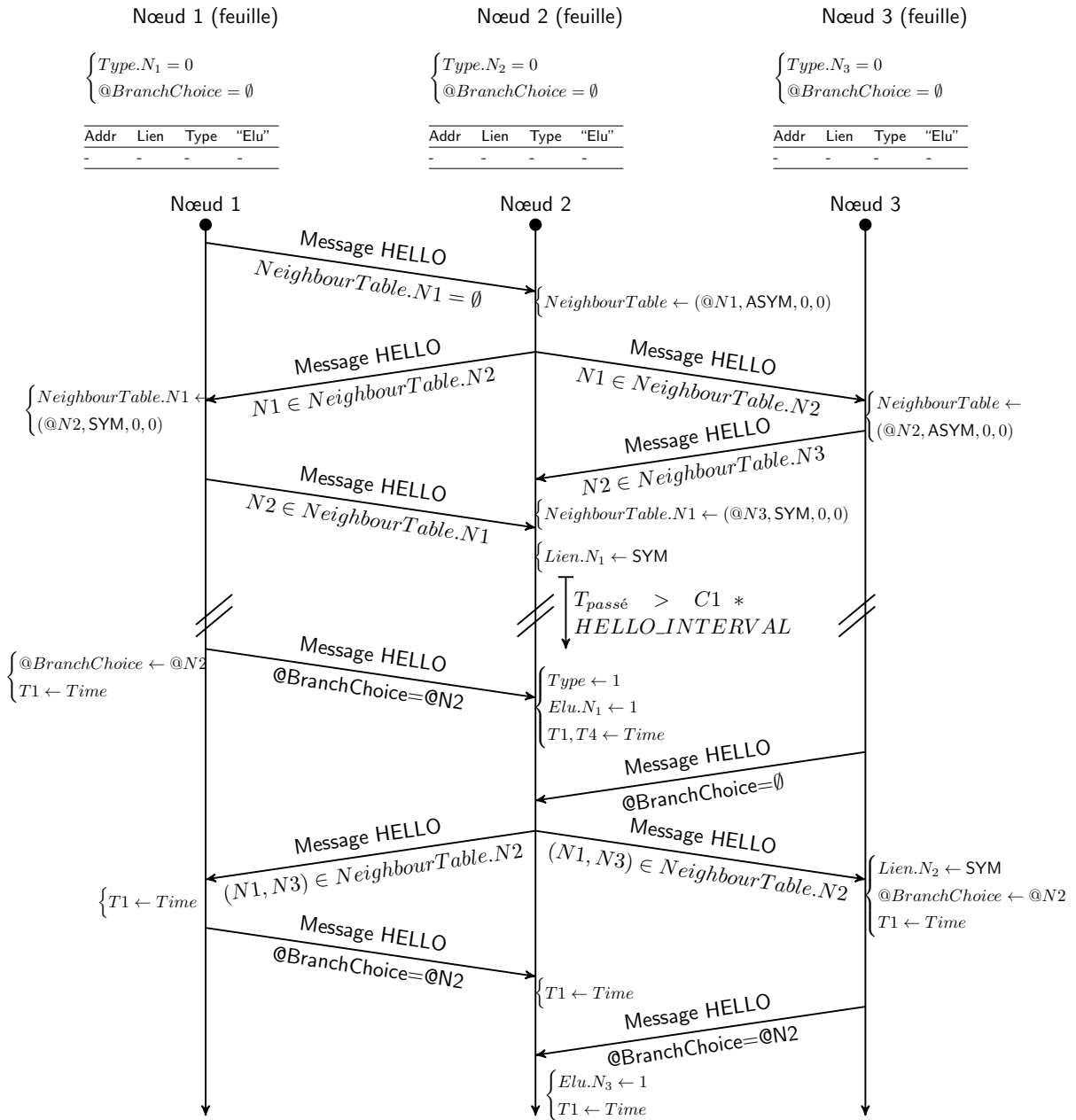
2.3.8.2 Choix d'un nœud branche relais par un nœud feuille dans le cas de deux nœuds branche candidats

La figure 2.8 illustre les étapes du choix d'un nœud branche par un nœud feuille lorsque deux nœuds branche sont candidats potentiels. Au départ, le nœud N2 est un nœud feuille, les nœuds N1 et N3 sont deux nœuds branche. En recevant le message HELLO de N1 informant ses voisins de son état (branche), N2 sélectionne N1 comme son nœud branche relais et informe ses propres voisins. Lors de la réception d'un message HELLO de N3 indiquant qu'il est également un nœud branche, N2 sélectionne N3 comme son nouveau nœud branche (le CT entre N3 et N1 est supérieur à celui entre N2 et N1). En l'absence de messages HELLO de N3 après un délai $Vtime$, N2 supprime N3 de sa table des voisins et de son choix de branche. Après réception d'un message de N1, N2 le sélectionne comme nœud branche relais. En raison de l'absence de réception d'un HELLO de N2, N3 retire N2 de sa table des voisins. À la fin de ces échanges, N1 est le nœud branche élu par le nœud feuille N2, la liaison entre N2 et N3 est rompue.

2.3.8.3 Étapes de rétablissement de la chaîne dans le cas d'un nœud branche amont dépassant son nœud branche aval

La figure 2.9 illustre les étapes liées au choix des nœuds branche aval et amont pour la formation d'une chaîne. Au départ, le nœud N1 est en amont du nœud N2, lui-même en amont du nœud N3. Les nœuds N2 et N3 sont des nœuds branche qui font partie de la même chaîne. Le nœud N1 est un nœud feuille qui n'a pas encore sélectionné de nœud branche relais. Le nœud N1 envoie un message HELLO à ses voisins. N2, n'ayant pas sélectionné de nœud branche en amont de sa chaîne et en supposant que la période de découverte du voisinage est dépassée, sélectionne le nœud N1 comme nœud branche en amont. Puis le nœud N3 fait de même, lorsque ce message est reçu par N1, ce nœud le sélectionne comme nœud branche relais. N2 informe ses voisins qu'il sélectionne N1 comme son nœud branche en amont. À la réception de ce message le nœud N1 devient un nœud branche et sélectionne N2 comme branche en aval. À la réception du HELLO de N1, N2 indique que N1 l'a élu comme nœud branche aval de sa chaîne. Après un certain temps, le nœud N1 dépasse le nœud N2. Lorsque le nœud N1 envoie son message HELLO, le nœud N2 détecte que son nœud branche amont (N1) l'a dépassé. N2 cesse de le sélectionner. À la réception du message HELLO de N2, N1 détecte qu'il a dépassé N2 et redevient un nœud feuille. Il sélectionne alors N2 comme nœud branche relais. Après un certain temps, la liaison N2-N3 devient caduque car le temps de connexion entre ses deux nœuds est inférieur à $L3 * HELLO_INTERVAL$. N3 va alors élire N1 comme nœud branche en amont pour éviter d'atteindre la cassure de la chaîne. En recevant un HELLO de N3, N2 arrête de le sélectionner comme nœud branche en aval, et N1 devient branche et sélectionne N3 comme nœud branche aval de sa chaîne. N1 envoie son HELLO, N2 le sélectionne alors comme nœud branche en aval et N3 indique que N1 l'a élu comme nœud branche de sa chaîne. Enfin, lorsque N2 envoie un HELLO, N1 sélectionne N2 en aval. La chaîne est alors reconstruite dans l'ordre N2-N1-N3.

Au temps courant, les tables de voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :
(N2 est supposé avoir la vitesse la plus faible)



À la fin de ces échanges, la table des voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :

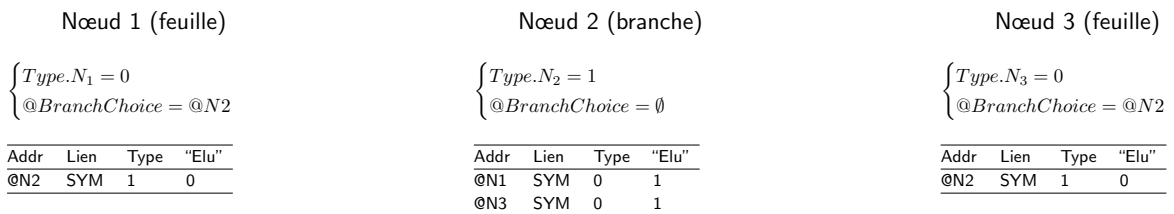
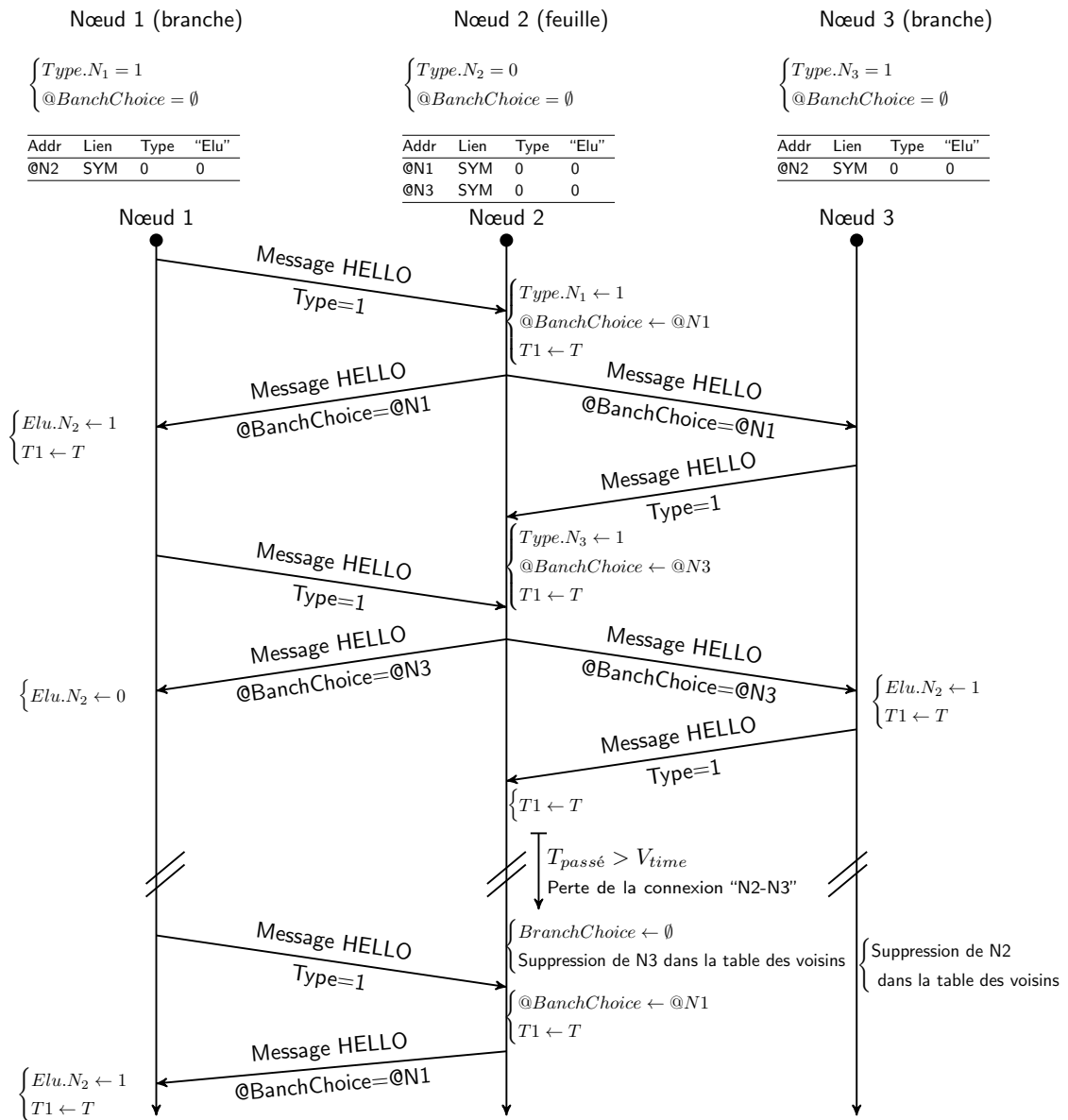


FIGURE 2.7 – Étapes de découverte et d'élection de nœud branche par des nœuds feuille

Au temps courant, les tables de voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :
(le CT entre N3 et N1 est supposé plus grand que celui entre N2 et N1)



À la fin de ces échanges, la table des voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :

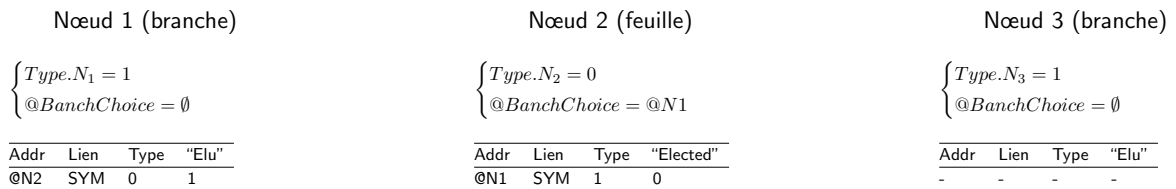
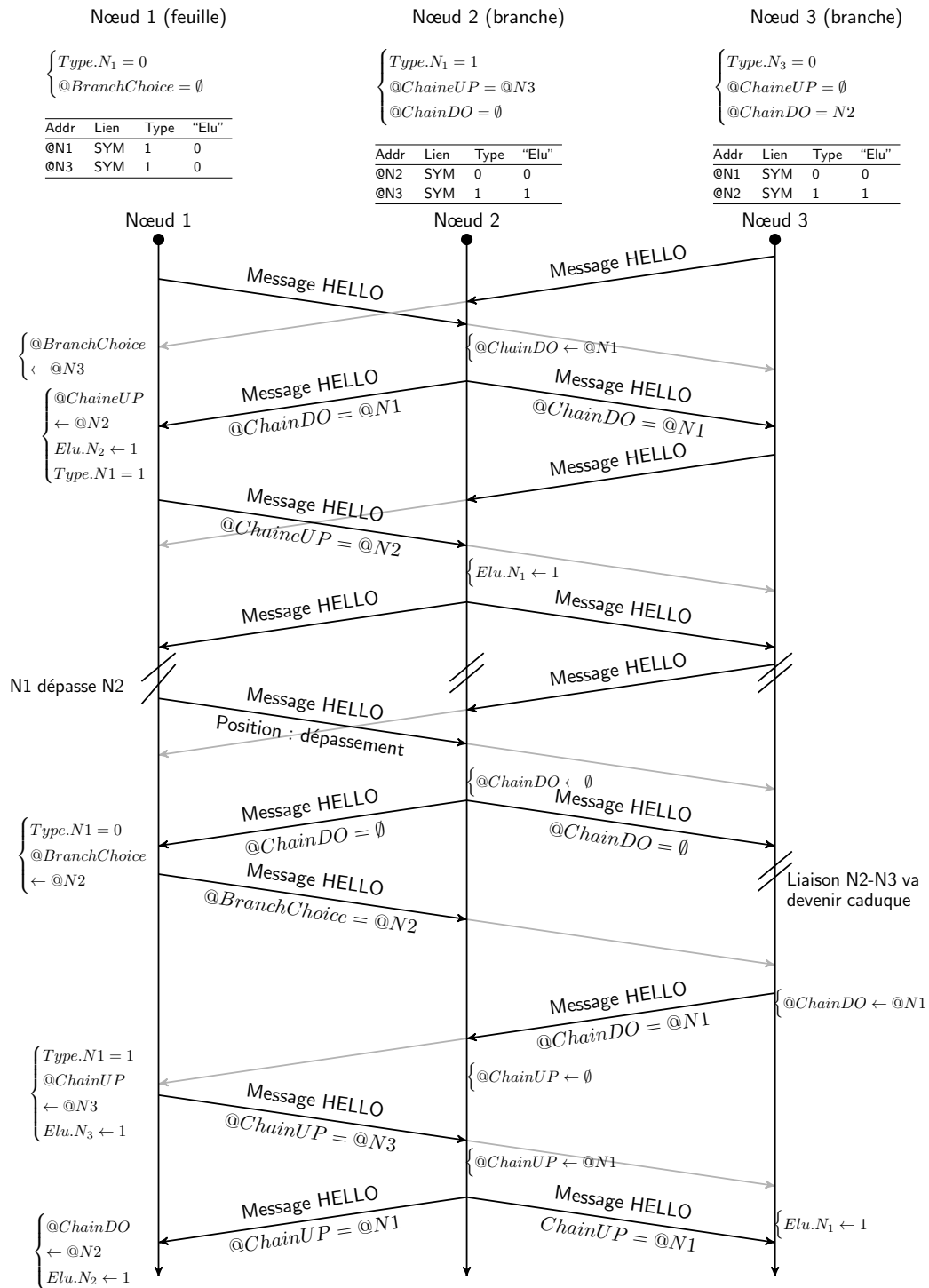


FIGURE 2.8 – Choix d’un nœud branche relais par un nœud feuille dans le cas de deux nœuds branche candidats.

Au temps courant, les tables de voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :
(N1 est en amont de N2 et N2 est en amont de N3)



À la fin de ces échanges, la table des voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :

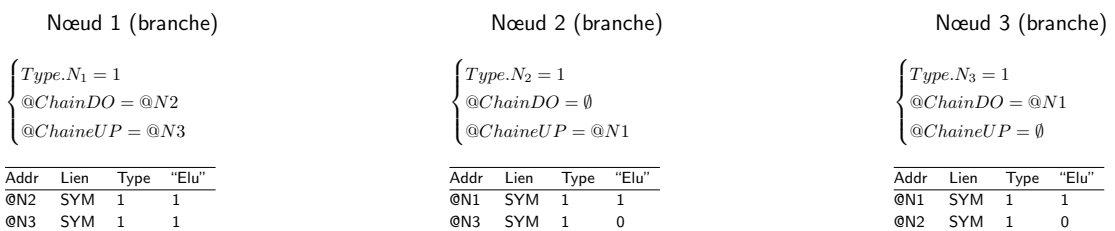


FIGURE 2.9 – Étapes de rétablissement de la chaîne dans le cas d'un nœud branche amont dépassant son nœud branche aval.

2.3.9 Paramétrage des coefficients pondérateurs

La seconde version de CBL présentée ici utilise sept coefficients pondérateurs ($C1$, $C2$, $C3$, $L1$, $L2$, $L3$ et α - voir section 2.2.8). Nous allons ici discuter du paramétrage de certains de ces coefficients.

Le coefficient **C2** intervient pour l'élection d'un nœud branche en amont par un nœud branche. Le nœud branche attend une période d'expiration de $C2 * HELLO_INTERVAL$, à l'issue de laquelle, s'il n'a pas détecté de nœud branche en amont dans son voisinage à un saut, il élit un nœud feuille voisin N_j au titre de nœud branche de sorte à créer un chaînon de chaîne. Pour rendre l'algorithme CBL le plus réactif possible, ce coefficient **C2** est fixé à **1**. Si un nœud branche en amont est présent dans le voisinage, alors un message HELLO sera reçu de ce nœud durant la période $HELLO_INTERVAL$.

En fixant la valeur de $C2$ à 1, le coefficient **C1** intervenant dans la découverte du voisinage d'un nœud branche relais par un nœud feuille peut également être fixé. On rappelle que les nœuds branche ont pour fonction de construire une chaîne.

L'objectif de l'algorithme CBL consiste à donner la priorité aux nœuds branche face aux nœuds feuille pour l'élection d'un nœud feuille au titre de nœud branche. Ainsi, il est nécessaire de laisser une période de découverte de voisinage plus importante pour un nœud feuille que pour un nœud branche. Comme nous l'avons vu en section 2.2.8 le coefficient $C1$ doit vérifier $C1 > C2$. La figure 2.10 illustre le pire cas de découverte du voisinage et permet de fixer la valeur minimale du coefficient $C1$. Pour rendre l'algorithme le plus réactif possible nous avons gardé cette valeur minimale comme valeur par défaut du coefficient $C1$ tel que **C1=C2+3=4**.

Le coefficient **C3** intervient dans le calcul d'une période de garde qui s'applique à l'instant où un nœud branche devient feuille. La figure 2.11 illustre le pire cas où cette période de garde doit être supérieure à $3 * HELLO_INTERVAL$. Nous avons donc choisi de fixer la valeur de **C3** à **3**.

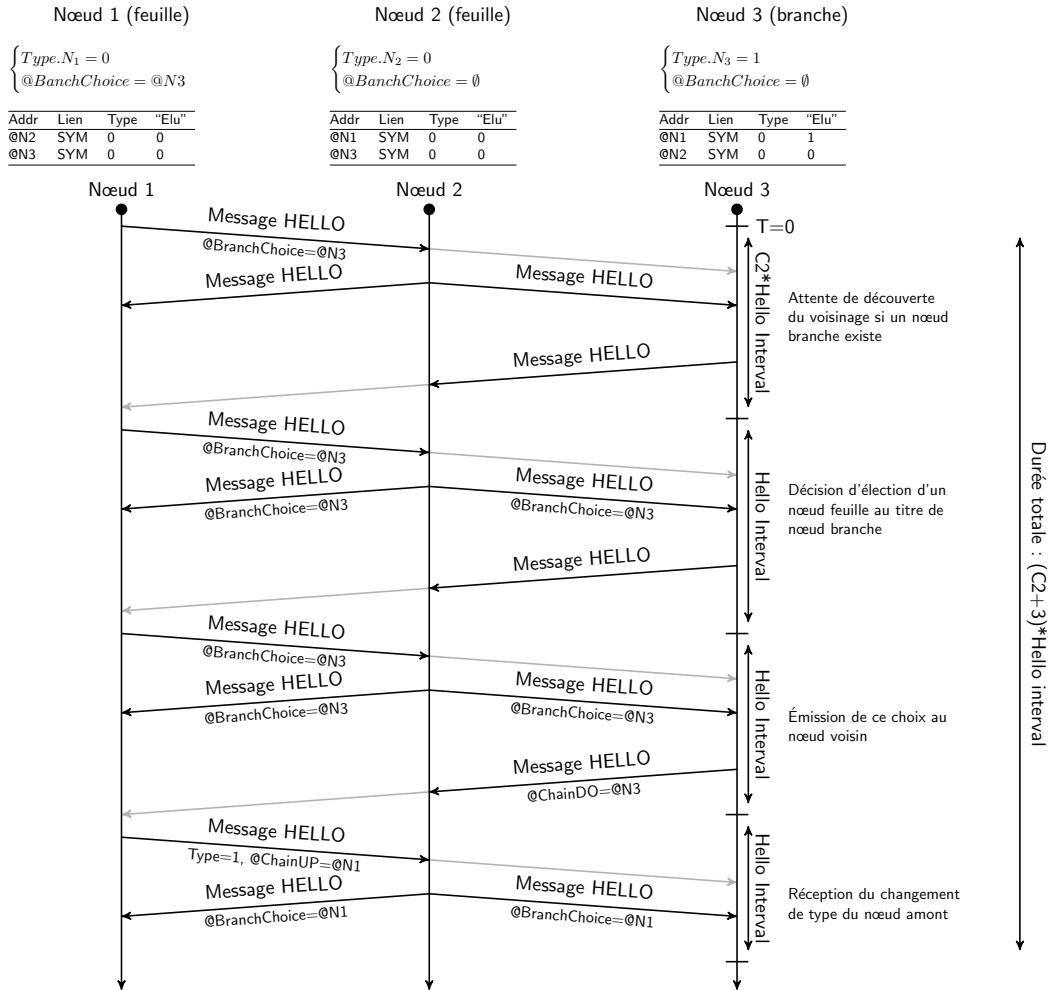
Le dernier coefficient dont nous avons fixé la valeur est le seuil $L3$. Ce seuil intervient dans le critère de changement d'abonnement d'un nœud feuille à son nœud branche relais ou d'un nœud branche à son nœud branche en amont. Nous avons fixé la valeur de **L3** à **3** pour faire correspondre ce critère au seuil temporel **Vtime** défini par le protocole OLSR [31] tel que $3 * HELLO_INTERVAL = Vtime$. Par conséquent, si un nœud élu comme branche relais ou maillon amont a un temps de connexion inférieur à $3 * HELLO_INTERVAL$, on estime que la connexion avec ce nœud va devenir caduque très rapidement et qu'il est nécessaire d'étudier un changement de choix pour un autre nœud voisin.

Nous avons fixé arbitrairement la valeur des trois derniers seuils ($L1$, $L2$ et α). Le tableau 2.3 récapitule ces choix. Dans la section 4.3.1 nous étudierons l'impact de la valeur de ces seuils sur la structure formée par CBL.

TABLEAU 2.3 – Paramétrage par défaut des coefficients pondérateurs

Seuil	C1	C2	C3	L1	L2	L3	α
Valeur fixée	4	1	3	0.5	5	3	0.5

Au temps courant (T=0), le nœud N3 vient d'être élu nœud branche par N1



À la fin des ces échanges, la table des voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :

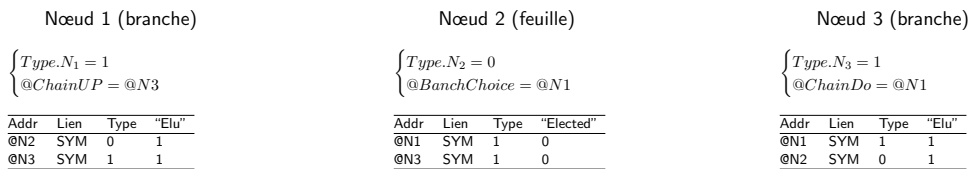
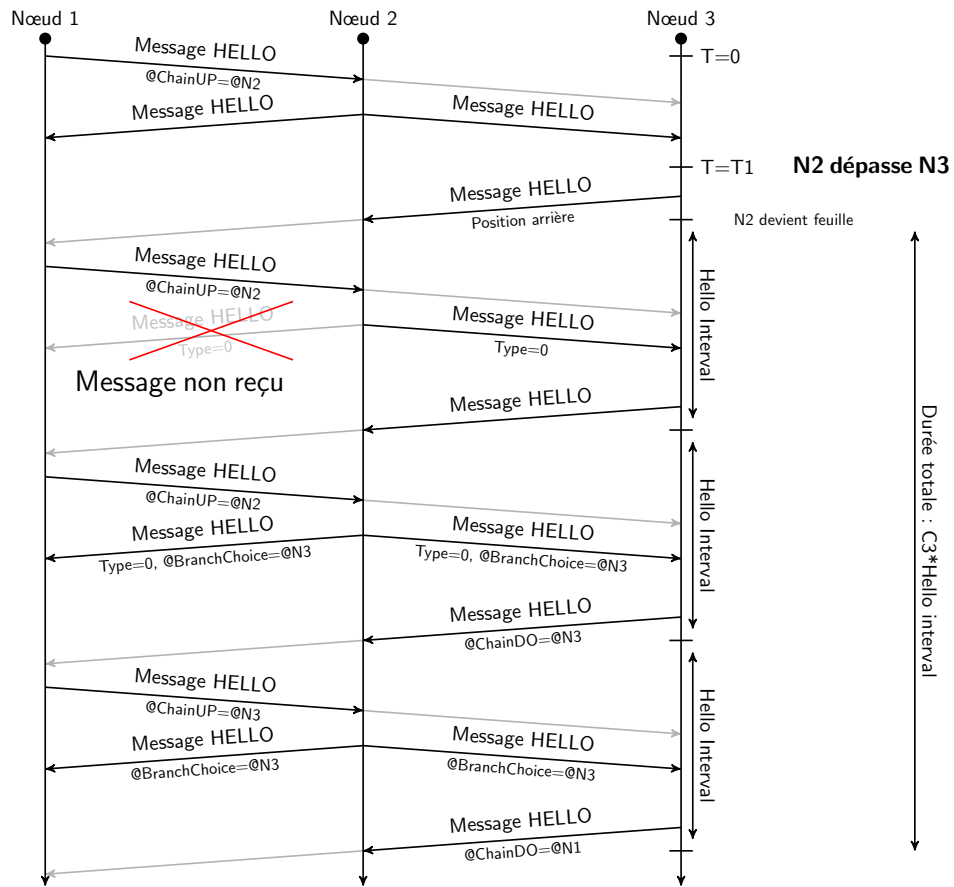


FIGURE 2.10 – Paramétrage du coefficient C1 en fonction du pire cas de découverte du voisinage

Au temps courant ($T=0$), la chaîne est composée des nœuds N1, N2 et N3. À $t=T1$, le nœud N2 dépasse le nœud N3

<p>Nœud 1 (branche)</p> $\begin{cases} Type.N_1 = 1 \\ @ChainUP = @N2 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elu"</th> </tr> </thead> <tbody> <tr> <td>@N2</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> <tr> <td>@N3</td> <td>SYM</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elu"	@N2	SYM	1	1	@N3	SYM	1	0	<p>Nœud 2 (branche)</p> $\begin{cases} Type.N_2 = 1 \\ @ChainDO = @N1 \\ @ChainUP = @N3 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elu"</th> </tr> </thead> <tbody> <tr> <td>@N1</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> <tr> <td>@N3</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elu"	@N1	SYM	1	1	@N3	SYM	1	1	<p>Nœud 3 (branche)</p> $\begin{cases} Type.N_3 = 1 \\ @ChainDO = @N2 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elu"</th> </tr> </thead> <tbody> <tr> <td>@N1</td> <td>SYM</td> <td>1</td> <td>0</td> </tr> <tr> <td>@N2</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elu"	@N1	SYM	1	0	@N2	SYM	1	1
Addr	Lien	Type	"Elu"																																			
@N2	SYM	1	1																																			
@N3	SYM	1	0																																			
Addr	Lien	Type	"Elu"																																			
@N1	SYM	1	1																																			
@N3	SYM	1	1																																			
Addr	Lien	Type	"Elu"																																			
@N1	SYM	1	0																																			
@N2	SYM	1	1																																			



À la fin des ces échanges, la table des voisins à un saut et les variables internes des nœuds 1, 2 and 3 sont :

<p>Nœud 1 (branche)</p> $\begin{cases} Type.N_1 = 1 \\ @ChainUP = @N3 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elu"</th> </tr> </thead> <tbody> <tr> <td>@N2</td> <td>SYM</td> <td>0</td> <td>0</td> </tr> <tr> <td>@N3</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elu"	@N2	SYM	0	0	@N3	SYM	1	1	<p>Nœud 2 (feuille)</p> $\begin{cases} Type.N_2 = 0 \\ @BranchChoice = @N3 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elected"</th> </tr> </thead> <tbody> <tr> <td>@N1</td> <td>SYM</td> <td>1</td> <td>0</td> </tr> <tr> <td>@N3</td> <td>SYM</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elected"	@N1	SYM	1	0	@N3	SYM	1	0	<p>Nœud 3 (branche)</p> $\begin{cases} Type.N_3 = 1 \\ @ChainDo = @N1 \end{cases}$ <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Addr</th> <th>Lien</th> <th>Type</th> <th>"Elu"</th> </tr> </thead> <tbody> <tr> <td>@N1</td> <td>SYM</td> <td>1</td> <td>1</td> </tr> <tr> <td>@N2</td> <td>SYM</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	Addr	Lien	Type	"Elu"	@N1	SYM	1	1	@N2	SYM	0	1
Addr	Lien	Type	"Elu"																																			
@N2	SYM	0	0																																			
@N3	SYM	1	1																																			
Addr	Lien	Type	"Elected"																																			
@N1	SYM	1	0																																			
@N3	SYM	1	0																																			
Addr	Lien	Type	"Elu"																																			
@N1	SYM	1	1																																			
@N2	SYM	0	1																																			

FIGURE 2.11 – Paramétrage du seuil C3 en fonction du pire cas

2.4 Discussion – Propriétés de CBL

Dans les sections 2.2 et 2.3, la méthode de création par le schéma CBL d'une structure dynamique à partir de la spécification de nœuds branche et feuille et du partage périodique d'informations d'état de ces nœuds dans leur voisinage proche via des messages HELLO a été présentée. Il a été en particulier notifié que :

1. D'une part, une structure CBL comporte des nœuds branche qui, rattachés successivement deux à deux construisent des chaînes en fonction de leur direction de circulation.
2. D'autre part, elle inclut des nœuds feuille, chaque nœud feuille sélectionnant dans son voisinage à un saut, un nœud branche auquel il se rattache.

Dans cette organisation, les nœuds branche jouent le rôle de relais lors du routage d'information entre nœuds distants de plus d'un saut. Chacun est le *cluster-head* d'un ensemble vide ou constitué du ou des nœuds feuille qui l'ont élu. Les groupes définis selon le schéma CBL possèdent donc la propriété suivante qui pourra être utilisée au profit de la mise en œuvre d'applications :

“Sachant qu'un nœud feuille ne sélectionne qu'un nœud branche et qu'un nœud branche ne se rattache au plus qu'à deux nœuds branche d'une même chaîne (l'un en amont de sa position et l'autre en aval), chaque groupe de nœuds feuille agrégés autour d'un nœud branche définit un ensemble disjoint de ceux constitués autour des nœuds branche voisins”.

Une première version du schéma CBL a fait l'objet d'une publication en revue [8]. Cette version, dont les algorithmes sont donnés en annexe A à la page 217, diffère en plusieurs points de la version présentée dans ce manuscrit. Nous avons cherché à corriger certains manquements de la version 1 et à renforcer la stabilité temporelle de la structure formée par CBL. Cette section explique les différences entre la version présentée dans [8] et celle expliquée dans ce manuscrit.

2.4.1 Élection d'un nœud branche par un nœud feuille (algorithme 3)

2.4.1.1 Élection d'un premier nœud branche – Initialisation de la structure

Dans la version 1 (annexe A, section A.1.3, algorithme 3, fonction COMPARELEAFNODE, ligne 12), lors de la procédure d'élection d'un nœud branche, après réception d'un message HELLO en provenance d'un nœud feuille circulant dans le même sens, un nœud feuille compare la vitesse de ce nœud avec celle des nœuds déjà enregistrés dans sa table des voisins dont la direction est identique. Il arrive que la table des voisins ait en mémoire un nœud à faible vitesse, mais dont la liaison est asymétrique. Ce voisin enregistré empêche la sélection d'un nouveau nœud relais.

Dans la version 2 (section 2.3.5, algorithme 3, fonction COMPARELEAFNODE ligne 14), les nœuds de la table des voisins ne sont plus pris en compte. À la place, l'algorithme compare la vitesse du nœud candidat avec celle du nœud branche élu s'il existe, sinon il sélectionne le nœud candidat.

Notons que dans la version 1 (annexe A, section A.1.4, algorithme 4, fonction COMPARECHAINNODE, ligne 11), le même biais existe dans la procédure d'élection, par un nœud branche, d'un nœud feuille au titre de nœud branche de la chaîne, où la valeur du temps de connexion du nœud candidat est comparée à celle des nœuds voisins. La valeur du temps de connexion est calculée à partir des informations enregistrées dans la table des voisins (position et vitesse). Dans la version 2 une solution identique a été mise en place qui compare le nœud déjà élu s'il existe avec le nœud candidat dont le message HELLO a été reçu.

2.4.1.2 Date de prise de décision de l'élection d'un nœud feuille au titre de nœud branche

Dans la version 1, (annexe A, section A.1.3, algorithme 3, fonction COMPARELEAFNODE, ligne 2) la décision de l'élection, par un nœud feuille, d'un nœud feuille au titre de nœud branche est conditionnée par la date à laquelle la liaison entre les deux nœuds est devenue symétrique ; cette liaison doit avoir été symétrique depuis au moins $Vtime$.

Un biais de cette méthode est le suivant : il arrive qu'un nœud feuille perde sa liaison avec son nœud branche. Le nœud feuille recherche alors un nouveau nœud branche. S'il reçoit un message HELLO d'un nœud feuille voisin ayant une liaison symétrique depuis au moins $Vtime$, il le sélectionne d'office, sans vérifier au préalable la présence d'un nœud branche dans son voisinage. S'il diffuse alors son choix du nœud feuille élu, ce dernier se transforme en nœud branche. Si ensuite il reçoit un message HELLO

l'informant de l'existence d'un autre nœud branche dans son voisinage avant d'apprendre en retour que le nœud précédemment élu est devenu branche, alors il modifie son choix, privilégiant le nœud déjà branche. Ainsi, un nœud feuille est entre-temps devenu branche sans que cela soit nécessaire à la structure CBL.

Dans la version 2, (section 2.3.5, algorithme 3, ligne 23) la décision de transformation, par un nœud feuille, d'un nœud feuille au titre de nœud branche est différée. Cette version introduit un ensemble d'horodatages consignants les dates de différentes actions du nœud. L'horodatage T8 enregistre l'instant où le nœud n'a plus de nœud branche ($@BranchChoice = 0$). À partir de cette date, le nœud effectue une découverte du voisinage pendant $C1 * HELLO_INTERVAL$. Si durant cet intervalle il détecte un nœud branche, il le sélectionne. Dans le cas contraire, il procède à l'élection d'un nœud feuille au titre de nœud branche. Tant que l'information d'élection n'a pas été transmise dans un message HELLO, le nœud a la possibilité de le modifier si un autre nœud feuille est un meilleur candidat (section 2.3.5, algorithme 3, ligne 29). L'horodatage T5 enregistre la date de création du dernier message HELLO du nœud. L'horodatage T11 enregistre l'instant où le nœud a sélectionné un nœud branche. Ces deux horodatages informent le nœud de l'émission ou non de la consigne de diffusion au voisinage.

Notons que le même biais existait dans la procédure de prise de décision de l'élection, par un nœud branche, d'un nœud feuille au titre de nœud branche de la chaîne. Deux horodatages ont été introduits, qui jouent le même rôle que T8 et T11. T6 enregistre l'instant où le nœud branche n'a plus de nœud branche amont ($@ChainDO = 0$). T9 enregistre l'instant où le nœud a sélectionné un nœud branche amont.

2.4.1.3 Métrique et procédure d'élection d'un nœud branche

Dans la version 1 (annexe A, section A.1.3, algorithme 3, fonction COMPAREBRANCHNODE, lignes 7 et 14), la procédure d'élection d'un nœud branche par un nœud feuille utilise la métrique de distance entre deux candidats potentiels. Il arrive qu'un nœud feuille se situant à équidistance entre deux nœuds branche modifie sans cesse, d'un message HELLO à l'autre, son choix de nœud branche.

Dans la version 2 (section 2.3.5, algorithme 3, fonction COMPAREBRANCHNODE lignes 20-28), la métrique utilisée pour départager deux candidats au titre de nœud branche est le temps de connexion. Une fois un nœud branche sélectionné, il le reste tant que son temps de connexion n'est pas inférieur à $L3 * HELLO_INTERVAL$. S'il est inférieur, l'écart entre le temps de connexion des deux candidats est comparé à un seuil conformément à l'équation 2.3. Cette amélioration augmente la durée de sélection d'un même nœud branche. Elle évite les changements intempestifs de choix.

2.4.2 Gestion des chaînes (algorithme 4)

2.4.2.1 Perte d'information lors de l'élection successive de nœuds branche amont et aval (algorithme 4)

Il arrive qu'un nœud branche d'une chaîne sélectionne un nœud en aval (ou en amont) puis qu'il sélectionne un nœud branche en amont (respectivement en aval) avant d'avoir informé son voisinage du précédent choix.

Or, dans la version 1 (annexe A, section A.1.4, algorithme 4, ligne 47), l'une et l'autre des sélections sont enregistrées dans une même variable @BranchChoice occasionnant la perte de l'information du précédent choix. Cependant, pour se transformer en branche (amont ou aval), les nœuds feuille vérifient la présence de leur adresse dans le champ @BranchChoice du message HELLO reçu. Dans cette configuration où l'un des choix a été écrasé, seul le dernier des deux nœuds sélectionnés se transforme en nœud branche. Un second nœud, du côté opposé de la chaîne, sera effectivement élu avec l'émission du prochain message HELLO.

Dans la version 2, la variable locale @BranchChoice est dédiée à l'élection d'un nœud branche par un nœud feuille, sa valeur est à zéro lorsque le nœud est branche. L'adresse contenue dans chacun des champs @BranchChoice, @ChainUP et @ChainDO des messages HELLO indique alors aux nœuds feuille s'ils doivent se transformer en branche (section 2.3.4, algorithme 2, ligne 2).

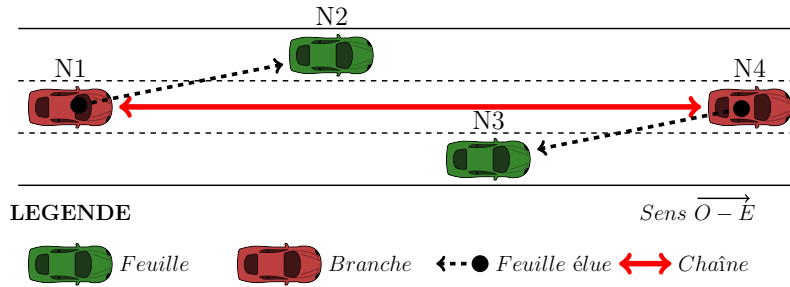
2.4.2.2 Chaînes parallèles

Suppression d'une cause Dans la version 1 (annexe A, section A.1.4, algorithme 4), la création d'une chaîne parallèle à une chaîne déjà existante, pour un même sens de circulation, peut se produire suite à

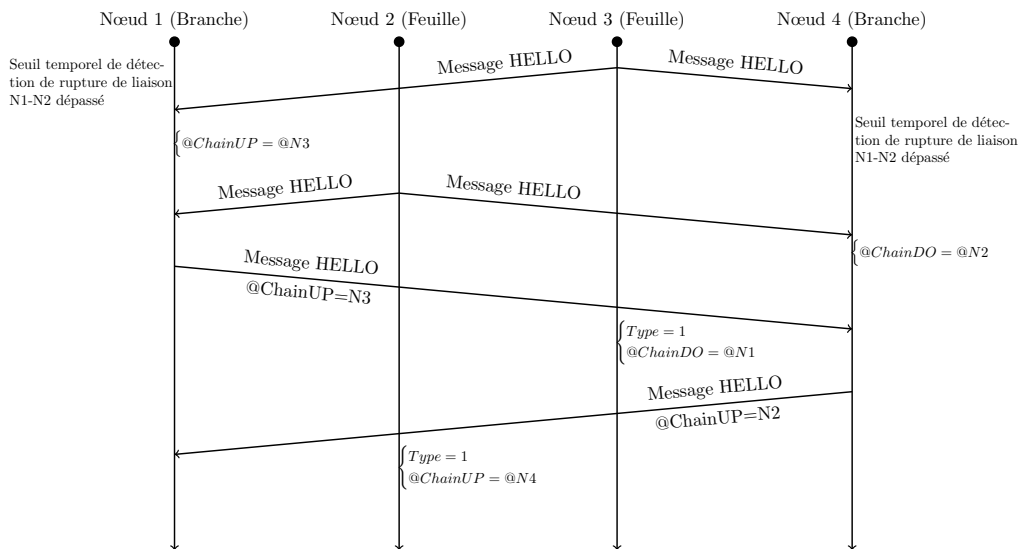
la rupture d'une liaison entre deux nœuds branche de la première chaîne (figure 2.12). Elle se produit lorsque les deux conditions suivantes sont réunies : deux nœuds branche sélectionnent chacun, dans le même laps de temps, un nœud feuille différent comme maillon de chaîne, les nœuds feuille sélectionnés ne forment pas une séquence ordonnée géographiquement. Cette création, non prévue par conception lors de la définition de CBL et donc non reproductible, par exemple au bénéfice d'une redondance, dégrade les performances en termes d'optimisation du trafic de routage.

Dans la version 2, CBL n'autorise l'extension de la chaîne qu'en amont de la direction de circulation (section 2.3.6, algorithme 4, lignes 85-91). En ne permettant plus l'extension de la chaîne concurremment en amont et en aval, une cause de choix conflictuel susceptible d'engendrer des chaînes parallèles est supprimée.

Situation initiale sur la route



Échange de messages après la rupture de la liaison N1-N2



Situation finale sur la route

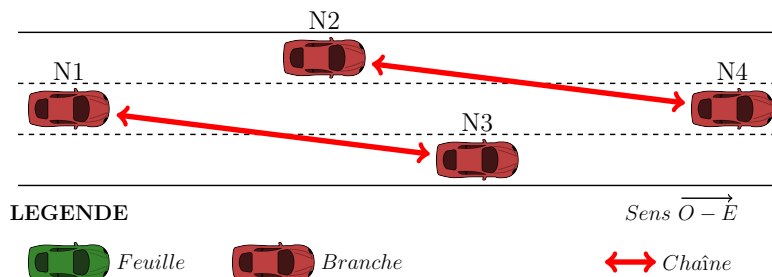


FIGURE 2.12 – Problème de création de chaînes parallèles suite à une rupture d'une liaison inter-branche

Chaînes parallèles persistantes Dans la version 1, les chaînes parallèles qui se forment progressent une fois créées.

Dans la version 2 (section 2.3.7, algorithme 5, lignes 9-14), dans le cas où une chaîne parallèle se créerait malgré la suppression de causes déjà identifiées, la méthode suivante procède à leur déconstruction. Un nœud branche qui n'a pas encore élu de nœud en aval devient feuille si, à réception d'un message HELLO en provenance d'un nœud branche 1) situé en aval de sa position, 2) dans son sens de circulation, 3) et électeur d'un maillon de chaîne situé en amont de sa position,

2.4.2.3 Phénomène de proximité de deux nœuds branche d'une chaîne consécutif à sa rupture par dépassement

Dans la version 1, lors d'une rupture d'une chaîne par dépassement d'un nœud branche par son nœud branche en aval, la reconstruction de la chaîne conserve les nœuds branche de la chaîne en opérant un réajustement de leur ordonnancement. Les deux nœuds concernés par le dépassement sont à proximité l'un de l'autre. Ils ont donc approximativement le même voisinage qui comprend notamment leurs nœuds branche amont et aval.

Dans la version 2, les modifications suivantes ont été introduites pour éviter que deux nœuds branche d'une même chaîne apportent la connaissance d'un même voisinage à la chaîne. Le nœud effectuant le dépassement devient feuille et n'est pas autorisé à se transformer en branche avant une période de garde $C3 * HELLO_INTERVAL$ (section 2.3.4, algorithme 2, ligne 2). Cette période laisse le temps au voisinage de prendre connaissance du fait qu'il soit devenu feuille et qu'il ne fait donc plus parti de la chaîne.

2.4.2.4 Procédure d'élection d'un maillon de chaîne

Métrique tenant compte de rupture de chaînes par dépassement. Dans la version 1 (annexe A, section A.1.4, algorithme 4, fonction COMPARECHAINNODE lignes 10-11), la procédure d'accroissement de la chaîne utilise la métrique du temps de connexion qui mesure le moment où le nœud va être hors de portée radio. Cette métrique ne tient pas compte des ruptures de chaînes dues au phénomène de dépassement entre deux nœuds successifs de la chaîne. L'algorithme privilégie, dans sa recherche d'un nœud en amont (respectivement en aval), le nœud ayant une vitesse supérieure (respectivement inférieure). Ce choix cause une rupture de la chaîne par dépassement, bien avant l'échéance du temps de connexion estimé avec ce nœud. Cependant, un autre nœud candidat feuille avec une vitesse inférieure (respectivement supérieure) aurait pu fournir une durée de connexion plus longue.

Dans la version 2 (section 2.3.6, algorithme 4, fonction COMPARECHAINNODE lignes 39-49), une nouvelle métrique de décision, *ChainTime*, a été créée, qui mesure le temps avant rupture de la chaîne. Son utilisation lors de la procédure d'accroissement retarde la rupture de la chaîne, qu'elle soit causée par une sortie de la zone de portée entre deux nœuds branche ou par dépassement de l'un vis-à-vis de l'autre.

Métriques et leur ordonnancement pour le choix d'un maillon dans un contexte de forte densité véhiculaires. Dans la version 1 (annexe A, section A.1.4, algorithme 4, fonction COMPARECHAINNODE ligne 2-4), la procédure d'accroissement de la chaîne utilise, dans un premier temps, l'apport de la connaissance d'au moins un nouveau voisin et, dans un second temps, le temps de connexion. Dans un contexte de forte densité de nœuds, il arrive qu'un nœud branche élise un nœud feuille qui soit très proche de lui, en ne lui apportant qu'un seul nouveau voisin. Ainsi, de fil en aiguille, la chaîne peut comporter un grand nombre de nœuds branche éloignés d'une faible distance géographique. Cette option contribue à un faible nombre de nœuds feuille par *cluster* et à un nombre de *clusters* supérieur.

Dans la version 2 (section 2.3.6, algorithme 4, fonction COMPARECHAINNODE, lignes 6-9), le critère qualitatif d'apport de nouveaux voisins est conservé et celui quantitatif du degré du nœud candidat (*i.e.* le nombre de nouveaux voisins apportés) est ajouté. La métrique quantitative privilégie les candidats possédant un degré élevé parmi l'ensemble des candidats éligibles. Cependant, plus le degré du candidat est élevé, plus son voisinage est différent du nœud branche électeur et sa distance géographique importante. Le risque est alors d'élire un nœud en limite de portée radio. Pour supprimer ce risque, la métrique de décision *ChainTime* est également prise en compte comme critère d'élection.

2.4.2.5 Gestion des nœuds branche isolés

Il arrive que des nœuds branche isolés apparaissent en présence d'une chaîne déjà construite.

Dans la version 1 (annexe A, section A.1.4, algorithme 4), tant qu'un nœud branche isolé est élu par un nœud feuille au titre de nœud relais, alors : 1. Le nœud reste branche ; 2. Il crée une chaîne parallèle.

Dans la version 2 (section 2.3.5, algorithme 3, fonction COMPAREBRANCHNODE, lignes 14-19), pour supprimer cette cause de création de chaînes parallèles, le critère d'appartenance à une chaîne est pris en compte lors de l'élection du nœud branche par un nœud feuille. Les nœuds feuille privilégient alors les nœuds branche appartenant à une chaîne et un nœud branche isolé est peu à peu effeuillé des anciennes feuille électriques si un nœud branche appartenant à une chaîne est détecté dans leur voisinage. Le nœud branche isolé et effeuillé devient feuille.

Notons que ce même critère est utilisé lors de l'élection des nœuds branche de la chaîne en amont et en aval afin d'éviter la formation de chaînes parallèles (section 2.3.6, algorithme 4, fonction COMPARECHAINNODE, lignes 26-38).

2.5 Mise en œuvre de CBL dans le protocole OLSR : CBL-OLSR

Si CBL est un schéma proactif de structuration d'une topologie de routage, il ne spécifie pas l'intégralité d'un protocole de routage. Par exemple, CBL ne spécifie pas les mécanismes de recherche de route. Ce schéma de *clustering* nécessite donc d'être mis en œuvre dans un protocole de routage existant.

Le fonctionnement de CBL repose exclusivement sur l'envoi périodique de messages HELLO. Toute mise en œuvre de CBL dans un protocole de routage existant nécessite donc, s'ils ne sont pas déjà spécifiés, la définition de l'envoi périodique de messages HELLO.

Parmi les protocoles de routage connus, nous avons présenté en section 1.2.2.1 le protocole OLSR particulièrement adapté au cas d'un réseau avec un grand nombre de nœuds [45]. Une autre raison pour laquelle OLSR est un candidat intéressant pour une implémentation de CBL, provient de ses caractéristiques. OLSR est un protocole de routage ad hoc dont la création de route repose sur l'émission périodique de message HELLO également nécessaire au fonctionnement de CBL et sur la définition de deux types de nœuds, les nœuds MPRs (nœuds relais du protocole) et des nœuds ordinaires comme pour CBL avec les nœuds branche et les nœuds feuille. La mise en œuvre de CBL au sein du protocole OLSR donne lieu à la création du protocole *Chain Branch Leaf - Optimized Link State Routing protocol* (CBL-OLSR). Les informations nécessaires au fonctionnement de CBL seront ajoutées dans les messages HELLO envoyés par OLSR ce qui fait de CBL une méthode de groupement hybride (section 1.2.2.2).

2.5.1 Présentation du protocole OLSR

Le protocole OLSR [31] est un protocole de routage hiérarchique et proactif. Chaque nœud du réseau applique une routine pour calculer le chemin optimal pour transmettre des messages vers un ou plusieurs nœuds destinataire. Ces chemins sont enregistrés au sein de plusieurs bases de données appelées table de routage. Ces tables consistent en la table des voisins à un saut, celle des voisins à deux sauts, celle des nœuds MPR choisis par le nœud et celle des nœuds l'ayant élu comme nœud MPR. Cette routine utilise l'état des liens du réseau pour fonctionner ainsi que deux types de messages de routage : les messages HELLO permettant la découverte des nœuds voisins et les messages *Topology Control* (TC) permettant le partage des tables de routage. La figure 2.13 présente l'en-tête d'un message HELLO défini par le protocole OLSR.

La première étape du protocole OLSR consiste à découvrir les nœuds voisins. Chaque nœud diffuse de manière périodique (paramètre *HELLO_INTERVAL* fixé par défaut à 2 secondes) un message HELLO qui contient les informations sur son voisinage. Ces informations sont l'adresse et le type de liaison des voisins (symétrique, asymétrique, MPR, perdu ou non spécifié). Si la liaison est symétrique cela signifie que le lien entre les deux nœuds est valide et bidirectionnel. Si la liaison est asymétrique, le lien est unidirectionnel, c'est-à-dire que le nœud a reçu un message de son voisin mais que celui-ci ne l'a pas encore validé. Si la liaison est de type MPR, cela signifie que ce nœud voisin est un relais MPR ce qui

entraîne automatiquement une liaison symétrique entre les deux nœuds. En recevant ces messages, les nœuds voisins traitent les informations (en actualisant leurs tables de routage) mais ne relaient pas ce message grâce à un système de compteur appelé *Time To Live* (TTL). Le nœud origine fixe la valeur du compteur TTL. Lorsqu'un message est relayé par un nœud, le compteur TTL est décrémenté de 1. Un nœud qui reçoit un message dont le compteur TTL atteint la valeur de 1 ne retransmet pas ce message. Dans le cas des messages HELLO, le compteur TTL est toujours fixé à 1 par le nœud émetteur du message HELLO. Par ailleurs, lorsqu'un changement de voisinage est détecté, la table de routage est recalculée pour mettre à jour les informations. Si un nœud n'a pas envoyé de message HELLO depuis un temps supérieur au seuil fixé (paramètre *Neighbor hold time*), il est supprimé des tables de routage. Un système de temporisateur est utilisé pour garantir la non obsolescence des informations. Le protocole OLSR transmet les valeurs nécessaires au calcul de ce délai dans le champ *Vtime* de l'en-tête des messages transmis au sein des paquets OLSR. Spécifié sur un octet, les quatre bits de poids forts du champ *Vtime* (section 2.2.8) codent l'entier représentant la mantisse a , les quatre bits de poids faibles codent l'entier représentant l'exposant b . La formule de *Vtime* est donnée selon [31] par :

$$Vtime = C * \frac{1 + a}{16} * 2^b \text{ (s)}$$

où C est une constante dont la valeur définie par défaut à 1/16 de secondes dans [31].

Les paramètres par défaut du protocole OLSR préconisés dans [31] sont présentés dans le tableau 5.3.

Les nœuds sélectionnés comme MPR diffusent des paquets TC à une fréquence fixée par le paramètre *TC interval*. Ces paquets sont envoyés en mode *broadcast*. Ils sont uniquement relayés par les nœuds MPRs. Un paquet TC d'un nœud contient les informations sur les nœuds voisins qui l'ont sélectionné comme nœud MPR. Grâce à la réception des paquets TC, chaque nœud du réseau met à jour ses tables de routage et calcule les chemins vers les autres nœuds du réseau grâce à l'algorithme Dijkstra du plus court chemin en utilisant la métrique du nombre de nœuds relais pour chaque route. Le fonctionnement est le suivant : soit le nœud N_i recevant un message TC initialement envoyé par le nœud N_k . Le message TC contient la liste des nœuds N_j ayant choisi le nœud N_k comme nœud MPR. Il existe une route passant par le nœud N_k pour joindre les nœuds N_j . Si les nœuds N_j sont présents dans les tables de routage du nœud N_i , celui-ci calcule le nombre de nœuds relais actuel lui permettant de joindre les nœuds N_j . Si le nombre de nœuds relais de la route passant par le nœud N_k est plus faible alors le nœud N_i ajoute cette route dans sa table de routage.

Le protocole OLSR effectue une structuration du réseau par l'intermédiaire des nœuds MPRs (section 1.2.2.1) permettant d'économiser les ressources radios. La procédure de sélection des nœuds MPRs est la suivante : chaque nœud N_i du réseau sélectionne un ensemble de voisins à un saut lui permettant de communiquer avec son voisinage à deux sauts. Les nœuds MPR sont utilisés comme nœud relais permettant au nœud de communiquer avec son voisinage à deux sauts. Seuls les nœuds de cet ensemble retransmettent les messages envoyés en *broadcast*. Pour effectuer ce choix, le nœud N_i exécute la routine suivante :

1. Chaque nœud voisin N_j ayant comme paramètre *N_willingness* égal à *WILL_ALWAYS* est automatiquement ajouté à l'ensemble des nœuds MPRs de N_i .
2. Le nœud N_i calcule pour chaque voisin N_j à un saut, le nombre D de voisins à deux sauts que le nœud N_j lui permet de joindre. Si un nœud N_j est l'unique nœud voisin à un saut ayant un lien de communication vers un voisin à deux sauts alors le nœud N_j est ajouté à l'ensemble des MPRs de N_i .
3. La procédure suivante est répétée jusqu'à ce que tous les voisins à deux sauts du nœud N_i aient un lien de communication avec l'un des nœuds MPRs choisi : Pour chaque nœud voisin N_j à un saut, le nœud N_i calcule le nombre D2 de voisins que le nœud N_j lui permet de joindre dans le voisinage à deux sauts de N_i et n'ayant pas encore de lien de communication avec l'un des nœuds MPRs choisi. Le nœud N_i ajoute à l'ensemble de ces MPRs, le nœud voisin à un saut dont la valeur de D2 est non nulle et ayant le paramètre *N_willingness* le plus élevé. Dans le cas de la présence de plusieurs nœuds candidats, c'est le nœud ayant la valeur de D la plus élevée qui est ajouté à l'ensemble de MPRs du nœud N_i .

Attribut	Valeur par défaut
Willingness	défaut
Hello interval	2 secondes
TC interval	5 secondes
Neighbor hold time	6 secondes (3*Hello interval)
Topology hold time	15secondes (3*TC interval)
Duplicate message hold time	30 secondes
Adressing	IPV4

TABLEAU 2.4 – Paramètres du protocole OLSR - issue de RFC 3626

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved																Htime								Willingness							
Link Code								Reserved								Link Message Size															
Neighbor Interface Address 1																															
Neighbor Interface Address 2																															
...																															
Link Code								Reserved								Link Message Size															
Neighbor Interface Address 1																															
Neighbor Interface Address 2																															
...																															

FIGURE 2.13 – En-tête d'un message HELLO défini par le protocole OLSR - source [31]

2.5.2 Correspondance entre MPRs et nœuds branche

Le schéma CBL permet, comme le protocole OLSR, de former une hiérarchie dans le réseau grâce à la distinction de deux types de nœuds. L'implémentation de CBL au sein de OLSR s'est donc naturellement orientée vers l'utilisation des nœuds branche de CBL comme des nœuds MPRs. Les nœuds branche héritent alors des mêmes fonctionnalités que celles des nœuds MPRs. Ce sont des nœuds qui relaient les messages envoyés en mode *broadcast* (comme les messages TC par exemple). Il est à noter que l'implémentation de CBL au sein de OLSR impacte le choix des routes initialement conçues par CBL. En effet le protocole OLSR effectue une recherche de route par l'algorithme du plus court chemin calculé grâce au nombre de nœuds relais d'une route. Il est possible qu'une route contienne des nœuds non-MPRs qui ont alors pour fonction de relayer un message envoyé en mode *unicast* utilisant cette route. Ainsi, comme l'implémentation, proposée ici de CBL au sein de OLSR, consiste à définir les nœuds branche comme les nœuds MPRs et les nœuds feuille comme les nœuds non-MPRs, il est possible qu'un nœud feuille ait pour fonction de relayer un message envoyé en mode *unicast*.

2.5.3 Modification du format des paquets

L'implémentation de CBL au sein du protocole OLSR nécessite une modification afin d'inclure l'information de direction des messages "Dir" (section 2.2.3). Cette information est ajoutée dans l'en-tête des messages de OLSR (figure 2.14) au niveau du type de message qui est codé initialement sur 8 bits. Or, dans la version initiale du protocole OLSR, le type de message ne peut prendre que des valeurs entières comprises entre 1 et 4. L'implémentation proposée utilise les quatre premiers bits pour coder l'information "Dir" (voir tableau 2.5 et figure 2.15). Lorsqu'un message est reçu par un nœud branche, il est retransmis

0									1									2									3								
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1				
Packet Length																		Packet Sequence Number																	
Message Type									Vtime									Message Size																	
Originator Adress																																			
Time To Live									Hop Count									Message Sequence Number																	
Message 1																																			
Message Type									Vtime									Message Size																	
Originator Adress																																			
Time To Live									Hop Count									Message Sequence Number																	
Message 2																																			
...																																			

FIGURE 2.14 – En-tête d’un paquet défini par le protocole OLSR - source : [31]

de la manière suivante :

1. Si la valeur du premier bit est à 1, le message est retransmis à tous les nœuds branche dans le voisinage à un saut (dont les nœuds branche d’un autre sens de circulation). Ce mode de retransmission correspond au mode *broadcast*.
2. Si la valeur du deuxième bit est à 1, le message est retransmis au nœud branche aval ;
3. Si la valeur du troisième bit est à 1, le message est retransmis au nœud branche amont ;
4. Si la valeur du quatrième bit est à 1, le message est retransmis aux nœuds feuille élus.

Message type	1	2	3	4	5	6	7	8
Transmission à tous les nœuds branche (mode <i>broadcast</i>)	1	-	-	-				
Transmission à @ChainUP	-	1	-	-	Conforme			
Transmission à @ChainDO	-	-	1	-	à			
Transmission aux nœuds feuille élus	-	-	-	1	OLSR			

TABLEAU 2.5 – Modification pour CBL du “message type” défini dans le protocole OLSR pour prendre en compte les différents cas de retransmission possible.

L’information de direction des messages “Dir” permet à la structure fédératrice CBL d’apporter une solution aux quatre scénarios de communication définis dans [128] pour les applications automobiles spécifiées par l’*European Telecommunications Standards Institute* (ETSI) : point-à-point, point-à-multipoint, *GeoAnyCast* et *GeoBroadcast* (figure 2.16). Pour une implémentation CBL-OLSR, les scénarios de communication point-à-point et point-à-multipoint sont respectivement réalisés par les mécanismes *unicast* et *multicast* du protocole OLSR. Les scénarios *GeoAnyCast* et *GeoBroadcast* pourraient être mis en œuvre conjointement avec les informations de position contenues dans la table de routage des nœuds voisins, celles de direction des messages “Dir”, avec le codage d’un nouveau type de mode de communication dans le champ “Message type” à l’aide des bits libres restants d’OLSR (tableau 2.5) et en recourant à la structure CBL. Soit une zone d’intérêt géographique vers laquelle un nœud source veut envoyer un message (cercle jaune dans la figure 2.16). Le nœud source identifierait la position relative en amont ou aval de cette zone d’intérêt. Il enverrait le message applicatif (contenant un nouveau champ identifiant la position et la taille de la zone géographique) au nœud de sa chaîne avec l’information de direction “Dir”

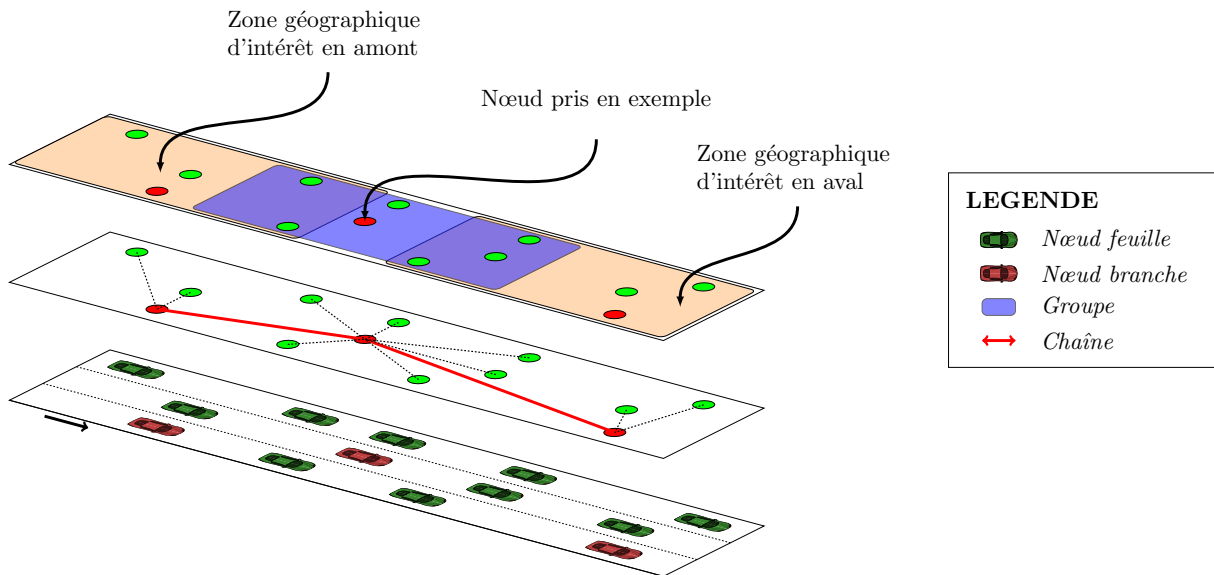


FIGURE 2.15 – Nœuds et liens fonctionnels de CBL : en bas, illustration d'un trafic routier ; au milieu, nœuds CBL correspondant aux véhicules représentés et liens entre nœuds ; en haut, mise en valeur des zones d'intérêt en amont et en aval d'un nœud branche particulier.

paramétrée pour que le message soit propagé dans le sens adéquat. À chaque réception du message par un nœud branche relais, lorsque le champ "Message type" indiquerait une communication selon l'un des scénarios *GeoAnyCast* et *GeoBroadcast*, le nœud vérifierait si la position du nœud branche suivant se situerait dans la zone cible. Dans le cas d'un scénario *GeoAnyCast*, lorsque cette zone serait atteinte, le nœud relais écraserait le champ TTL du message avec une valeur de 1 validant une dernière retransmission. Dans le cas d'un scénario *GeoBroadcast*, lorsque cette zone serait atteinte, le nœud relais écraserait le champ "Dir" du message avec la valeur "1001XXXX" permettant la diffusion locale à tous les nœuds branche et nœuds feuille. Chaque nœud branche recevant ce message le rediffuserait dans le cas où il serait situé dans la zone cible.

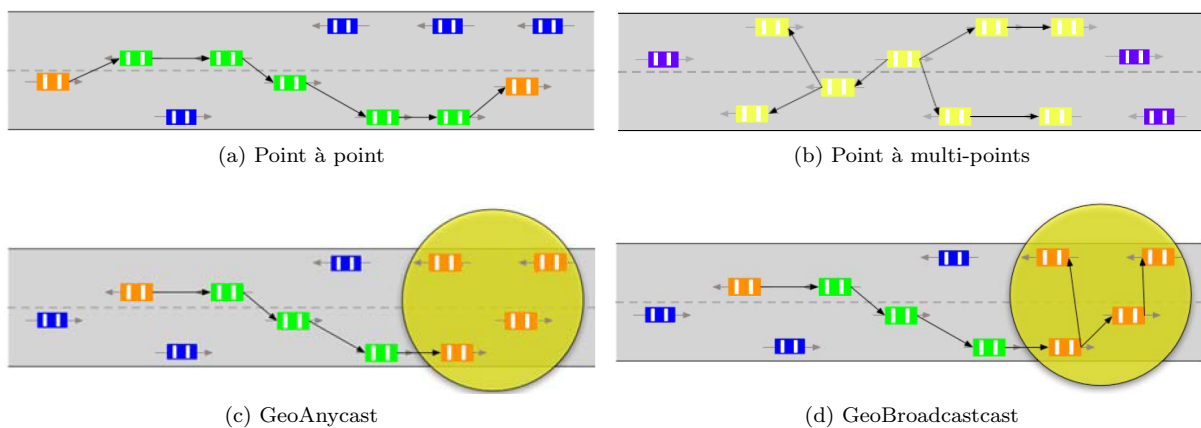


FIGURE 2.16 – Modes de communication supportés par le protocole GeoNetworking - source [128]

2.5.4 Modification du format des messages HELLO

L'implémentation de CBL au sein du protocole OLSR nécessite plusieurs modifications au niveau des messages HELLO pour ajouter les informations de position, de type de nœud (branche ou feuille) et de vitesse, d'angle de direction ainsi que du choix des nœuds concernant la structure de la chaîne (@BranchChoice, @ChainUP et @ChainDO). Un des objectifs de l'implémentation de CBL au sein du protocole OLSR était d'optimiser au maximum la taille requise pour les informations supplémentaires à ajouter

au sein des messages HELLO. Ces messages sont envoyés périodiquement à une fréquence importante et utilisent une part non négligeable des ressources de communication. Seuls 32 bits supplémentaires ont été ajoutés au niveau de l'en-tête du message HELLO par rapport à l'en-tête standard du protocole OLSR.

2.5.4.1 En-tête des message HELLO de CBL

Les variables sont codées ainsi (figure 2.17) :

- **La vitesse** (speed) sur 8 bits de 0 à 256 km/h avec une résolution de 1 km/h.
- **L'angle de braquage** (steering angle) sur 8 bits de 0 à 360° par rapport au Nord, avec une résolution de 1.5°.
- **Longitude** :
 - (Seconde Longitude) sur 10 bits (entier de 0 à 540) permettant de coder les minutes et secondes d'angle de positionnement longitudinal.
 - (dsec Lon) sur 4 bits (entier de 0 à 9) permettant de coder les dixièmes de secondes d'angle de positionnement longitudinale.
- **Latitude** :
 - (Seconde Latitude) sur 10 bits (entier de 0 à 540) permettant de coder les minutes et secondes d'angle de positionnement de la latitude.
 - (dsec Lat) sur 4 bits (entier de 0 à 9) permettant de coder les dixièmes de secondes d'angle de positionnement de la latitude.
- **Type du véhicule (T)** sur 1 bit (0 : nœud feuille, 1 : nœud branche).

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Speed								Steering Angle								Htime								Willingness							
Reserv	T	Seconde Longitude								Seconde Latitude								dsec Lon				dsec Lat									

FIGURE 2.17 – En-tête d'un message proposé dans le cadre de l'utilisation du schéma CBL

Précisions concernant la position : les nœuds destinataire du message HELLO se situent dans le voisinage à un saut du nœud émetteur. Ils sont donc à portée directe de communication. La technologie utilisée dans cette thèse pour les réseaux véhiculaires correspond au IEEE 802.11p [6]. Cette technologie a une portée maximale (P) de 1 km. Compte tenu de cette distance, il est inutile que les nœuds transmettent, au sein des messages HELLO, les valeurs de position de degrés d'angle et de dizaines de minutes d'angle. En effet, ces valeurs sont communes pour tous les nœuds destinataire des messages HELLO. Le codage de l'information de position de longitude et de latitude sur 14 bits est présenté en annexe B à la page 225.

2.5.4.2 Corps des messages HELLO

Une modification au niveau du *link code* a été réalisée pour inclure les informations du choix des nœuds concernant la structure de la chaîne (@BranchChoice, @ChainUP et @ChainDO). Initialement dans OLSR, le *link code* est codé sur 8 bits et les quatre derniers bits permettent de coder le type de lien et le type de voisin comme l'illustre la figure 2.18. Les valeurs prises par les champs "Link Type" et "N.Type" sont les suivantes :

- Link Type=1, le lien est non spécifié (*UNSPEC*);
- Link Type=2, le lien a été perdu (*LOST*);
- Link Type=3, le lien est unidirectionnel, i.e. asymétrique (*ASYM*);
- Link Type=4, le lien est bidirectionnel, i.e. symétrique (*SYM*).
- N.Type=1, le nœud voisin possède au moins un lien symétrique (*SYM_NEIGH*);
- N.Type=2, le nœud voisin possède au moins un lien symétrique et a été sélectionné comme MPRs (*MPR_NEIGH*);

- N.Type=3, le nœud voisin n'a pas encore un lien symétrique (*NOT_NEIGH*);

L'implémentation de CBL utilise 3 bits du *link code* lorsque Link Types=SYM_LINK et N.Types=MPR_NEIGH pour inclure les informations de @BranchChoice, @ChainUP et @ChainDO. Les 3 bits permettent de définir l'ordre d'apparition de ces choix dans la liste des adresses des nœuds voisins qui suivent le *link code* comme illustré dans le tableau 2.6.

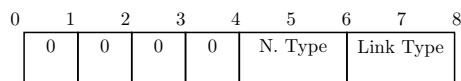


FIGURE 2.18 – Link code défini par le protocole OLSR

Link code	1	2	3	4	5	6	7	8
@ChainUP en 1 ^{ère} position	-	-	-	1				
@ChainUP non élu	-	-	-	0				
@ChainDO en 1 ^{ère} position	-	-	1	0				
@ChainDO en 2 ^{ème} position	-	-	1	1		Conforme		
@ChainDO non élu	-	-	0	-				
@BranchChoice en 1 ^{ère} position	-	1	0	0		à		
@BranchChoice en 2 ^{ème} position	-	1	1	0				
@BranchChoice en 2 ^{ème} position	-	1	0	1		OLSR		
@BranchChoice en 3 ^{ème} position	-	1	1	1				
@BranchChoice non élu	-	0	-	-				

TABLEAU 2.6 – Modifications proposées du link type dans le cadre de l'utilisation du schéma CBL pour inclure l'ordre d'apparition du choix des nœuds relais @BranchChoice, @ChainUP et @ChainDO

La figure 2.19 montre un exemple de message HELLO envoyé par un nœud branche ayant choisi un nœud comme branche en amont (@ChainUP) et en aval (@ChainDO).

2.5.5 Méthode de recherche de route

Le principe de recherche de route du protocole OLSR n'a pas été modifié par la mise en œuvre de CBL-OLSR. Comme dans le protocole OLSR, les messages TC sont envoyés par les nœuds relais. Chaque nœud relais inscrit dans ses messages TC la liste des nœuds voisins l'ayant sélectionnés. Ainsi, les nœuds branches de CBL-OLSR génèrent les messages TC contenant la liste de leurs nœuds feuilles rattachés. Ces messages sont relayés dans tout le réseau en mode *broadcast* avec la chaîne.

2.5.6 Modification des tables de voisins

Cette section 2.5.6 termine la présentation de l'algorithme avec la description des modifications qui ont été apportées à l'en-tête et au corps des messages HELLO lors de la mise en œuvre de CBL dans OLSR. Les tables de routage de chaque nœud ont été modifiées pour ajouter les informations supplémentaires concernant les nœuds voisins. Cette table contient les treize champs présentés dans la section 2.2.5. Le codage de ces champs est présenté dans le tableau 2.7.

Champs	1	2	3	4	5	6	7	8	9	10	11	12	13
	Adresse	Lien	Date 1	Date 2	Position X	Position Y	Vitesse	Angle	Type	Elected	N.Br.Choice	N.ChainDO	N.ChainUP
Nombre de bits utilisées pour le codage	32	2	32	32	16	16	8	8	1	1	32	32	32

TABLEAU 2.7 – Codage de table des voisins à un saut d'un nœud

0				1								2								3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Speed								Steering Angle								Htime								Willingness															
Reserv				1				Seconde Longitude								Seconde Latitude								dsec Lon				dsec Lat											
Link Code								Reserved								Link Message Size																							
Neighbor Interface Address 1																																							
Neighbor Interface Address 2																																							
...																																							
X				X				1				1				LC LSB								Reserved								Link Message Size							
Neighbor Interface Address 1 : @ChainUP																																							
Neighbor Interface Address 2 : @ChainDO																																							
Link Code								Reserved								Link Message Size																							
...																																							

LC LSB : 4 bits de poids faible du Link Code de OLSR

FIGURE 2.19 – Exemple d’un message HELLO dans le cadre de l’utilisation du schéma CBL

Deuxième partie

Analyse de performances de CBL

Chapitre 3

Cahier des charges pour l'évaluation des performances d'un protocole de routage dans le contexte véhiculaire

Sommaire

3.1	Introduction	105
3.2	Évaluation structurelle et évaluation système	106
3.3	Simulateurs de réseaux de communication	107
3.3.1	Simulateurs utilisés dans les travaux référencés au chapitre 1	107
3.3.2	Environnements de simulation	108
3.4	Modélisation de la mobilité	109
3.4.1	Modèles de mobilité choisis dans les travaux référencés au chapitre 1	109
3.4.2	Paramétrages du simulateur SUMO	115
3.5	Modèles de propagation des ondes radio	119
3.5.1	Modèles de propagation des ondes radio choisis dans les travaux référencés au chapitre 1	119
3.5.2	Modélisation de la propagation sous OPNET Riverbed Modeler	122
3.6	Modèle de la technologie de transmission de trames	124
3.6.1	Technologies existantes	124
3.6.2	Description du protocole IEEE 802.11p	124
3.6.3	Paramétrage dans Riverbed OPNET Modeler	126
3.7	Modélisation d'applications	127
3.7.1	Classification des applications	127
3.7.2	Besoins de communication	128
3.8	Métriques de performances pour l'évaluation de protocoles de routage	129

3.1 Introduction

Une fois un nouveau protocole de routage conçu, il est important d'en évaluer les performances et de les comparer à celles des protocoles proposés dans la littérature. Pour cela, s'il n'est pas possible ou aisé de l'expérimenter directement sur le terrain en situation réelle ou dans des conditions proches de la réalité, il convient de choisir un environnement de modélisation et de simulation informatique. Dans le cas de l'évaluation d'un protocole de routage pour les VANETs, équiper des véhicules et réaliser des tests sur le terrain s'avèrent effectivement coûteux en ressources et difficiles à mettre en place, même avec un nombre restreint de véhicules comme cela peut être le cas avec la participation de constructeurs automobiles et d'équipementiers à l'occasion de programmes de recherche européens comme les projets SCOOP [3] et InterCor [4].

La simulation informatique d'un réseau VANET est une solution plus simple à mettre en œuvre et plus économe en ressources. Cependant, cette simulation repose sur la modélisation de phénomènes physiques complexes et sur celle de systèmes informatiques et électroniques dont le fonctionnement est par nature à la fois synchrone et asynchrone. La précision des résultats de simulation obtenus dépend du réalisme des modèles employés. Les environnements de modélisation et de simulation, les modèles mis en œuvre, la nature des contextes routiers ciblés (urbains, autoroutiers) mènent à une littérature abondante en matière de proposition de protocoles et d'évaluation de performances de ceux-ci (tableaux 3.13, 3.14 et 3.15). Cette situation est critiquée par exemple par [131]. En effet il existe un nombre significatif de protocoles de routage pour les VANETs mais il est impossible de connaître leurs performances relatives puisqu'aucune plate-forme commune d'évaluation n'a été mise en place.

Nous nous attachons dans ce chapitre à établir un état des différents outils nécessaires à l'évaluation d'un protocole de routage aussi bien au niveau de la structure formée par le protocole que d'un point de vue système. Pour ce faire, nous avons repris les différentes propositions de protocoles de routage des travaux référencés dans le chapitre 1.

3.2 Évaluation structurelle et évaluation système

Les évaluations des protocoles de routage peuvent avoir deux objectifs : l'analyse de la structure formée par un protocole lorsque ce dernier crée une topologie hiérarchique dans le réseau (section 1.2.2) ou l'évaluation du système global formé par l'ensemble des couches du réseau de communication, dont celle incluant le protocole conçu.

L'évaluation structurelle d'un protocole de routage consiste à en étudier les performances en idéalisant les couches du modèle OSI qui sont en interaction avec la couche réseau supportant le protocole de routage étudié. Dans le cas de l'évaluation du schéma CBL qui est indépendant des technologies radio et des applications, seules les modélisations de la mobilité des nœuds et de la portée radio du canal de propagation sont nécessaires. Les résultats d'évaluation attendus ne concernent que les aspects structurels du protocole de routage tels que les types des nœuds (branche ou feuille), le nombre de structures (chaînes) formées ou encore le temps nécessaire à la formation d'une structure.

Une évaluation système est plus complète. Elle prend en compte l'ensemble des couches (selon la terminologie du modèle OSI) mises en œuvre par le réseau de communication. Ainsi, la modélisation du schéma de *clustering* CBL au sein d'un système de communication pour les VANETs nécessite la modélisation de la couche réseau dans lequel le schéma CBL s'intègre, celle des couches supérieures application et transport, qui supportent notamment les profils applicatifs des utilisateurs et la préparation des paquets de données, et celle des couches inférieures, liaison de donnée et physique de la technologie radio. La couche liaison de donnée modélisera en particulier les accès au médium et la couche physique le modèle de propagation.

Par conséquent, une fois un logiciel de modélisation et de simulation sélectionné, et outre le choix d'une modélisation d'applications pour les VANETs, d'un modèle de propagation, d'une technologie radio appropriée aux VANETs, il convient, pour évaluer le schéma CBL de choisir un protocole de routage où intégrer CBL (section 2.5.1), une modélisation réaliste de la mobilité des nœuds (figure 3.1) et des métriques de performances.

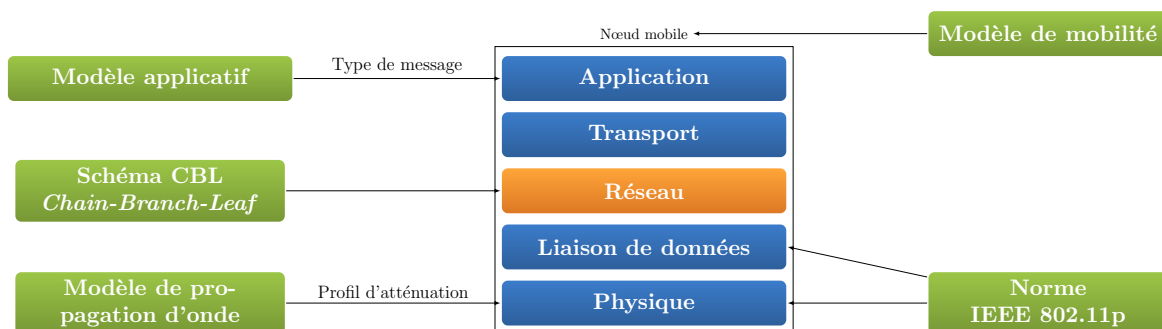


FIGURE 3.1 – Modélisations nécessaires à l'étude des performances du schéma CBL dans un système VANET.

3.3 Simulateurs de réseaux de communication

3.3.1 Simulateurs utilisés dans les travaux référencés au chapitre 1

Il existe deux familles de simulateurs selon qu'ils intègrent ou non la modélisation de la mobilité des nœuds. Les simulateurs de réseaux simples modélisent uniquement les réseaux de communication et nécessitent donc l'usage de traces de véhicules qui ont été préalablement générées avec un autre outil (section 3.4). Les simulateurs intégrés comportent en réalité un simulateur pour la modélisation du réseau de communication et un second pour la modélisation des trajectoires des nœuds. Selon le mode de synchronisation entre ces deux simulateurs, le simulateur intégré est dit unidirectionnel lorsque seul le simulateur de mobilité interagit avec le simulateur de réseau en lui fournissant les trajectoires des nœuds au cours de la simulation ; il est dit bidirectionnel lorsque les deux simulateurs interagissent l'un avec l'autre et qu'ainsi les messages applicatifs reçus par les nœuds du réseau peuvent modifier la mobilité de ces derniers.

Nous nous intéressons ici plus particulièrement aux simulateurs de réseau couramment cités dans les études et évaluations des protocoles de routage référencés dans notre état de l'art du chapitre 1 (voir aussi les tableaux 3.13, 3.14 et 3.15). Certaines des caractéristiques de ces simulateurs sont résumées dans le tableau 3.1.

TABLEAU 3.1 – Simulateurs de communication utilisés dans la littérature

Simulateurs		NS2/NS3 [132]	MATLAB [133]	Omnet ++ [134]	OPNET [135]	NCTUns [136]
Caractéristiques	Simulateur intégré					✓
	Développement en cours	✓	✓	✓	✓	
	Langage	C++/OTCL	C/C++/Java	C++	C++/Java	C
	Multi-plateforme	✓	✓	✓	✓	
	Licence libre	✓				✓
	Interface graphique	✓	✓	✓	✓	✓
	Animation	✓	✓	✓	✓	✓
Usage	Prise en main	–	++	–	+	–
	Exemples fournis	✓			✓	
	Installation	+++	+++	++	+	–
	Facilité d'usage	–	+	–	–	–
Fonction	Rapidité simulation	✓	✓	✓	✓	✓
	Passage à l'échelle	–	++	+	+	++
	Modèle 802.11p	✓		✓	✓	✓

L'outil de développement de réseau *Network Simulator* (NS) est un simulateur à événement discret modulaire à licence libre écrit en C++ qui a été développé pour la recherche sur les réseaux de communication. Avec 20 citations (figure 3.16), ses versions NS2 ou NS3 sont très utilisées dans la communauté scientifique des VANETs. Ce logiciel contient des bibliothèques et des modèles permettant la génération de topologies réseau, des protocoles de routage et de trafics applicatifs. Il offre aussi des outils de visualisation. En particulier, il intègre un modèle de la couche physique et MAC de la norme IEEE 802.11p. Les codes de NS n'étant pas un produit fini mais le résultat de recherche et de développement continus, les modèles proposés doivent être vérifiés avant utilisation par les utilisateurs, afin d'en supprimer les éventuelles bogues [132]. Le simulateur NS2 est le simulateur de réseau de l'outil intégré unidirectionnel *Traffic and Network Simulation Environment* (TraNS). Cet outil, développé en Java, utilise par ailleurs le simulateur de mobilité de véhicules *Simulation of Urban MObility* (SUMO) pour la génération de trajectoire (voir section 3.4.1).

Objective Modular Network Testbed in C++ (OMNeT++) est un simulateur modulaire de réseaux écrit en C++ [134]. Il offre un environnement de développement intégré basé sur la plate-forme Éclipse

avec de nombreux outils et une interface graphique. Le simulateur OMNeT++ est utilisé par le simulateur intégré et bidirectionnel de communication inter-véhiculaire *VEhicule In Network Simulation* (VEINS) en association avec le simulateur de mobilité de véhicules SUMO. VEINS a notamment été utilisé dans les études de performances des protocoles DBA-MAC [60] et CDS-SVB [67].

Optimized Network Engineering Tools (OPNET) Riverbed Modeler [135] (anciennement dénommé “OPNET Modeler[®] Suite”) est un logiciel commercial de conception, de modélisation, et d’analyse des réseaux de communication qui propose des modèles validés par la société éditrice du logiciel des protocoles les plus utilisés (OLSR, AODV, *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP), IP, Ethernet, IEEE 802.11, etc.), et notamment depuis la version 18, une modélisation de l’interface 802.11p. OPNET Riverbed Modeler est un simulateur intégré à événements discrets. Il donne la possibilité de générer manuellement les trajectoires de nœuds ou de faire appel à des modèles de déplacements aléatoires. Avec une seule citation (figure 3.16), il est moins utilisé dans la recherche que NS et *Objective Modular Network Testbed in C++* (OMNET++) à cause du coût de la licence. [114] s’en est servi pour la conception et l’évaluation du protocole QOLSR [114] ou encore [55] pour le développement du protocole GeoAODV.

NCTUns [136] est à l’origine un simulateur de réseau de communication. Néanmoins, il intègre un simulateur de trafic routier à partir des versions supérieures à la v4.0, devenant ainsi un simulateur intégré bi-directionnel. Ce simulateur qui n’est pas une multi-plateforme, nécessite l’utilisation d’un OS Linux. Avec une seule citation (figure 3.16), il a notamment été utilisé dans [58] pour évaluer le protocole C-DRIVE.

MATrix LABoratory (MATLAB) est un environnement de développement et un langage de programmation émulé pour l’analyse des données, le développement d’algorithmes et la création de modèles. Il n’a pas été conçu pour la simulation des environnements des VANETs, ni pour l’étude des réseaux de communication. Fortement utilisé dans le monde académique malgré une licence payante, il est avec 4 citations l’outil le plus utilisé après NS pour la modélisation et l’évaluation de performances de protocoles pour les VANETs, tels que VWCA [88], PSO-C [91] ou encore QoS OLSR [115].

Synthèse et choix retenu

En conclusion, notons que les études [137, 14] réalisent une comparaison plus approfondie des simulateurs de réseaux de communication utilisés pour les VANETs, mais qu’au moment de l’écriture de ce manuscrit, aucun consensus ne se dégage pour recommander l’usage d’un simulateur plutôt qu’un autre. Notre choix s’est porté sur deux outils de développement : OPNET Riverbed Modeler et MATLAB.

3.3.2 Environnements de simulation

3.3.2.1 OPNET Riverbed Modeler

OPNET Riverbed Modeler se décompose en plusieurs domaines présentés dans l’annexe C page 227. Nous avons utilisé ce logiciel pour les évaluations système avec le module WIRELESS car l’IFSTAR travaille avec ce logiciel depuis 2008 et dispose du modèle de nœud 802.11p de Riverbed Modeler. De plus, le laboratoire Léost avait également enrichi ce modèle de nœud d’une option d’affectation automatique d’une trajectoire à chaque nœud du réseau, trajectoires contenues dans des fichiers de données externes. Les comparaisons d’évaluation de protocoles [138, 139, 140, 141, 8, 142, 143, 144] font également appel aux modèles fournis par OPNET tels que ceux de OLSR, AODV et GRP. L’intérêt de ces modèles est d’être commun aux utilisateurs d’OPNET Riverbed Modeler.

3.3.2.2 MATLAB

La version R2012b a été utilisé en complément et préalablement à la mise en œuvre du protocole CBL sous OPNET, pour l’évaluation structurelle de CBL face à celle d’OLSR [145, 8] en réalisant un simulateur simplifié de communication ne prenant en compte que deux aspects. Le premier concerne la mobilité des nœuds et le second concerne la portée de communication qui a été fixée par une distance maximale entre les nœuds.

De plus MATLAB a été utilisé pour plusieurs tâches présentées dans la figure 3.2. En particulier, il a permis d’afficher des animations pour visualiser la progression des nœuds du réseau. Cette visualisation a

notamment aidé à mettre en exergue les différents exemples cités en section 2.4. Ces exemples ont montré les limites de la première version de CBL et ont permis de définir des mécanismes de correction pour atteindre la seconde version de CBL présentée dans ce manuscrit au chapitre 2. MATLAB a également été utilisé pour générer automatiquement les fichiers TRJ, calculer *a posteriori* à partir des fichiers journaux (*log files*) de simulation d'OPNET, le nombre de chaînes formées dans le réseau. Pour finir il nous a été utile pour réaliser les figures de résultats dont l'affichage sous OPNET n'est pas optimal.

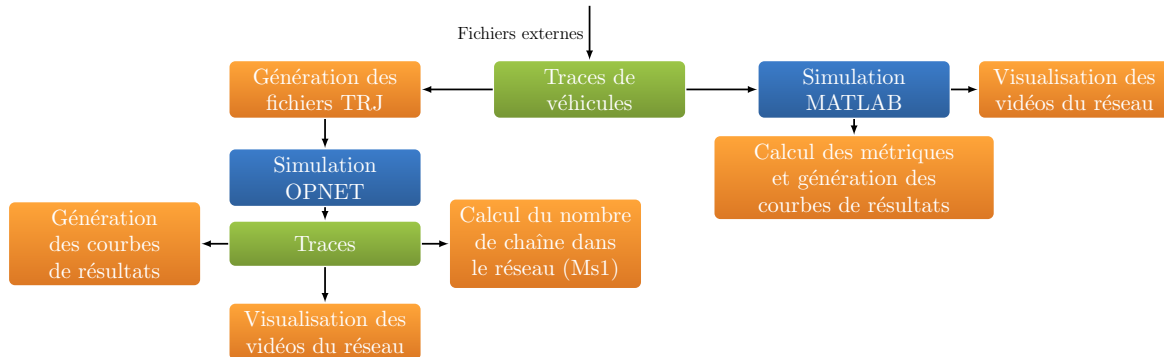


FIGURE 3.2 – Diagramme d'utilisation de MATLAB. Les blocs en orange correspondent aux tâches exécutées avec des scripts MATLAB.

3.4 Modélisation de la mobilité

3.4.1 Modèles de mobilité choisis dans les travaux référencés au chapitre 1

La mobilité des nœuds à bord de véhicules des VANETs possède les mêmes caractéristiques que celles des véhicules. L'évaluation d'un protocole pour les communications V2V impose donc la modélisation des trajectoires des nœuds. L'idéal serait d'avoir à disposition, pour chaque configuration routière étudiée, des ensembles de trajectoires réelles de véhicules enregistrées sur quelques kilomètres de voies routières, suffisamment nombreuses d'une part et précises en nombre de mesures d'autre part pour pouvoir en affecter une à chacun des nœuds du réseau de communication évalué. Pour notre étude, nos besoins en données consistent en au moins 5 km de route départementale ou autoroute, avec ou sans entrées et sorties, à une ou plusieurs voies, et contenant l'ensemble des véhicules qui y circulent à des heures de densité faible, moyenne, et dense.

Les traces réelles mises à la disposition de la communauté scientifique proviennent d'enregistrements de données *Global Position System* (GPS) ou d'enregistrements vidéo. Les traces de véhicules en provenance d'enregistrements de données GPS (ou *Floating Car Data* (FCD)), peuvent facilement être obtenues de nos jours. Ainsi, une base de données est mise à disposition des chercheurs par *Crawdad* [146]. *Crawdad* regroupe plus d'une centaine d'ensembles de traces réelles de flotte de véhicules de type taxis ou bus pour l'étude des VANETs. Par exemple, à l'aide des traces disponibles générées par les GPS de 4 000 taxis de Shanghai, l'étude [130] a pu montrer la limitation temporelle des connexions des communications V2V induite par la forte mobilité des nœuds d'un VANET. Cependant, les flottes de véhicules ne comptent qu'un sous-ensemble du trafic routier d'une zone qui ont de plus une mobilité liée à une utilité professionnelle différente de celle des véhicules de particuliers. Les traces réelles issues de ces enregistrements ne contiennent pas de véhicules proches ou alors n'incluent que très peu de véhicules.

Il est aussi possible d'obtenir des traces réelles de trajectoires de véhicules à l'aide d'enregistrements vidéo. Ces enregistrements sont réalisés avec une caméra installée sur un mât aux abords d'une autoroute ou sur une plate-forme aérienne embarquée [147]. Selon [148], deux grands groupes de jeux de données ont été construits depuis le début des années 2000 : des séries d'ensembles de données résultants d'expériences menées aux Pays-Bas par l'université technologique de Delft (TU Delft) [149] et les jeux de données *Next Generation Simulation* (NGSim) [150] du programme de l'administration fédérale américaine des autoroutes *Federal Highway Administration* (FHWA). Seuls les jeux de données NGSim sont publics. Ce sont des données brutes et non filtrées pour lesquelles [151] propose une méthode de filtrage. Un troisième jeu de données filtrées a été proposé par [148] suite à des enregistrements dans la région grenobloise d'une

zone routière de 500 m, réalisés par voie aérienne à l'aide d'une caméra sur un hélicoptère, à l'occasion du projet PREDIT *Measuring and Modelling Congestion and Pollution* (MOCOPO). Ces données filtrées, permettant de reconstruire les trajectoires des véhicules sur cette zone, ont été utilisées dans [140] pour l'évaluation comparative des performances des protocoles de routage OLSR, AODV, GRP et DSR. Les traces de trajectoires de véhicules issues de l'analyse d'enregistrement vidéo ont pour intérêt de contenir l'ensemble des véhicules qui ont traversé la section couverte par l'enregistrement. Cependant cette section est de très faible longueur : 500 m pour les données MOCOPO et 1 km pour les données NGSIM.

Nous constatons donc que les bases de données de traces réelles actuelles, qu'elles soient réalisées à partir d'enregistrement GPS de flottes ou d'enregistrement vidéo de tous les véhicules d'une zone, ne répondent pas à l'ensemble des besoins de notre cahier des charges. Les premières ne prennent pas en compte l'ensemble des véhicules de la région étudiée, les secondes ne couvrent qu'une zone géographique d'au plus 1 km. En l'absence d'une base de données répondant à nos besoins, une modélisation du trafic est possible. Son objectif est de fournir une reproduction, la plus fidèle possible, du comportement des automobilistes. Les études [152, 138] montrent qu'en fonction du choix de modèle de mobilité, les résultats des performances des protocoles de routage sont variables. L'exemple donné dans [153] à l'occasion de l'étude du protocole DSR confirme également l'analyse de ces études.

Selon le niveau de réalisme souhaité, différentes modélisations mathématiques existent, certaines développées dans un premier temps pour l'étude des MANETS, d'autres plus spécifiquement pour celles des VANETS. Parmi celles-ci figurent les modèles reposant sur un choix stochastique des variables de position de début et de fin de trajectoire d'une part, et de vitesse entre ces positions d'autre part.

Le modèle *Random Way Point* (RWP) est le premier modèle de mobilité aléatoire qui fut initialement créé pour les MANETS, puis repris pour l'étude de VANETS. L'utilisateur de ce modèle définit au préalable la dimension de la surface sur laquelle les nœuds peuvent se déplacer, les intervalles de valeur dans lesquels la vitesse des nœuds se situe, enfin les temps d'attentes à chaque position intermédiaire des nœuds. Le modèle calcule ainsi les trajectoires des nœuds : tout nœud du réseau tire aléatoirement une position d'origine et une destination se situant dans la surface de simulation définie. La valeur de vitesse du nœud, choisie aléatoirement dans les limites d'un intervalle paramétrable, est alors considérée comme fixe et sert à générer les positions successives du nœud entre le point d'origine et celui de destination précédemment tirés selon un parcours en ligne droite. Une fois la destination atteinte, une immobilisation du nœud en ce point est réalisée pendant la durée d'attente programmée (pouvant être nulle). Puis, de nouveau le nœud sélectionne aléatoirement les nouvelles valeurs de destination et de vitesse, et ainsi de suite. Ce modèle, simple à mettre en œuvre, est employé dans de nombreuses évaluations de protocoles telles que [154, 155, 28, 59, 88, 89, 114] ; bien qu'il soit irréaliste dans la mesure où les nœuds mobiles se déplacent sans contraintes dans l'espace défini, son intérêt est de permettre le choix de vitesses de déplacement élevées.

Le modèle *Street-RWP* rajoute au modèle RWP des contraintes de déplacement entre deux positions successives des nœuds. La trajectoire entre deux points n'est plus rectiligne, mais suit des chemins définis sur une carte représentant le réseau routier. Cette carte peut être définie aléatoirement, correspondre à une grille dans le cas d'un "modèle Manhattan" ou encore à un fragment de réseau routier réel (figure 3.3). La trajectoire des véhicules est alors calculée soit à l'aide d'un algorithme du plus court chemin, par exemple de type Dijkstra, soit de manière stochastique. Cette dernière option introduit une probabilité de changement de direction (tourne-à-droite, tourne-à-gauche) à chaque intersection rencontrée. Le modèle *Street-RWP* a été employé notamment pour l'évaluation des performances des protocoles OHNS [107], DSDV [156] et QOLSR [157]. Si le modèle *Street-RWP* prend en compte les contraintes de déplacement au moyen des cartes des contraintes de déplacement des nœuds, il ne considère cependant pas les interactions entre les véhicules. C'est pourquoi avec ce modèle, les véhicules peuvent se percuter.

Un troisième modèle de mobilité stochastique, *Freeway Mobility Model* (FDM) [158], est utilisé dans [71, 73, 72]. Il ajoute au modèle *Street-RWP* la notion d'interaction entre véhicules qui lui manquait et impose comme carte routière celle d'une autoroute à deux fois deux voies : chaque nœud qui se déplace sur cette autoroute est contraint de garder une distance de sécurité minimum avec le nœud qui le précède. La vitesse de chaque nœud à un instant t est dépendante de sa vitesse à l'instant précédent $t - 1$ et d'une valeur tirée aléatoirement. Cependant, lorsque deux nœuds se suivent sur une même voie, le déplacement

du nœud qui suit le premier est bridé par la contrainte de la distance de sécurité minimum qu'il doit garder avec le nœud qui le précède.

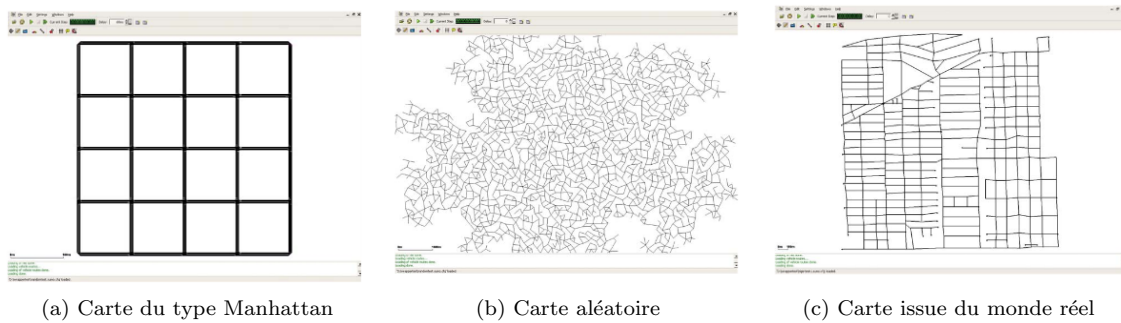


FIGURE 3.3 – Différentes cartes générées par les modèles de type *Street-RWP* - source : [159]

Malgré les améliorations contextuelles apportées au modèle initial RWP, les modèles *Street-RWP* et FDM ne reflètent toujours pas la réalité de terrain. Stochastiques, ils sont construits à partir de positions et vitesses aléatoirement choisies. En outre, la dispersion significative des valeurs de vitesses au cours du temps et de l'espace que nous avons signalée comme une caractéristique importante des réseaux VANETs en section 1.1.2 n'est pas prise en compte.

La modélisation de la mobilité des nœuds dans les VANETs peut encore être réalisée à l'aide d'un simulateur de trafic routier microscopique. [137, 160] en dressent un état de l'art. Ces simulateurs microscopiques génèrent la trajectoire de chaque véhicule à l'aide de modèles d'interaction entre les véhicules voisins. Gourmands en ressources mémoire et menant à de longues simulations, ils nécessitent de restreindre celles-ci à des portions limitées du réseau routier (quelques rues). Deux autres catégories de simulateurs de trafic routier existent, mais qui ne permettent pas de générer des trajectoires individuelles de véhicules. Ce sont, d'une part, les simulateurs macroscopiques, rapides à mettre en œuvre et adaptés aux simulations à large échelle (taille d'une ville) qui modélisent l'évolution de flots du trafic routier selon les principes de la mécanique des fluides et, d'autre part, les simulateurs mésoscopiques, qui combinent les propriétés des modèles macroscopiques et microscopiques. Ces derniers ne considèrent pas les mouvements et vitesses des véhicules de manière individuelle car leurs calculs se basent sur la valeur de la vitesse moyenne d'un lien du réseau.

Différents simulateurs de trafic microscopiques ont été utilisés lors des évaluations des protocoles de routage présentés au chapitre 1. Le tableau 3.2 en établit la liste. Ces simulateurs n'ont pas été initialement conçus pour la génération de traces de véhicules en vue de l'évaluation de protocoles de routage pour les réseaux VANETs. Cependant certains d'entre eux le permettent. En utilisant un simulateur de trafic, il est nécessaire de prendre en compte la complexité de ces logiciels et leur lenteur concernant le calibrage des modèles régissant les interactions entre véhicules. Ces modèles nécessitent l'ajustement de nombreux paramètres afin d'obtenir des trajectoires de véhicules réalistes.

Le simulateur VanetMobiSim [161] a été conçu à partir du modèle CanuMobiSim avec l'objectif d'améliorer le réalisme des simulations. Sa gestion de la topologie routière et celle des caractéristiques et structures des routes s'appuie sur un modèle macroscopique. Un modèle microscopique est également intégré qui donne accès à plusieurs modèles intelligents de suivi de véhicules basé sur le modèle *Intelligent Driving Model* (IDM) [166] et ses variantes (IDM-IM pour la gestion des intersections et IDM-LC pour la gestion multi-voies). Ce simulateur, paramétrable grâce à des fichiers XML, a été validé dans des scénarios de communication réels. Avec cinq citations (figure 3.16), le simulateur VanetMobiSim [161] est très utilisé pour les études sur les VANETs [115, 100, 104, 101, 106]. Cependant, il n'est plus maintenu et sa documentation est limitée.

CORridor SIMulation (CORSIM) [162] est un simulateur commercial développé par le département américain des transports. Il combine deux modèles microscopiques : NETSIM pour la gestion des routes urbaines et FRESIM pour celle des autoroutes. Ce simulateur propose des modèles de suivi de véhicules, la

TABLEAU 3.2 – Simulateur de trafic routier microscopique utilisés dans la littérature

Simulateurs	VanetMobiSim [161]	CORSIM [162]	SUMO [163]	NetStream [164]	NCTUns [165]	
Caractéristiques	Première version	2006	1997	2002	1997	2002
	Version actuelle	2007	2012	2017	1997	2010
	Développement en cours		✓	✓		✓
	Multi-plate-forme			✓		
	Licence libre	✓		✓		✓
	Interface graphique	✓	✓	✓	✓	✓
	Facilité d'utilisation	+++	+	++	-	+
Réseau	Carte aléatoire	✓		✓	✓	
	Carte définie par l'utilisateur	✓	✓	✓		✓
	Import carte réel	✓	✓	✓	✓	✓
Fonctionnalités	Modèle de poursuite	✓	✓	✓	✓	✓
	Multi-voies	✓	✓	✓		✓
	Dépassement	✓	✓	✓		✓
	Contrainte vitesse	✓	✓	✓		✓
	Gestion intersection	✓	✓	✓		✓
	Type de véhicule différents	✓	✓	✓		✓

gestion des voies routières multiples avec changement de file et la gestion des intersections. Ce simulateur a notamment été utilisé pour l'évaluation du protocole *Position-Based Clustering* (PPC) [62].

Le simulateur *Simulation of Urban MObility* (SUMO) [163] permet la modélisation des systèmes de trafic inter-modaux, y compris les véhicules routiers, les transports en commun et les piétons. Il propose l'importation du réseau routier de manière manuelle, de manière aléatoire ou par l'import d'un réseau routier réel. SUMO est capable de gérer les voies routières multiples, les vitesses maximales par voies, ainsi que les intersections grâce à un système de stop ou de feux de signalisation. Ce simulateur permet également de modéliser plusieurs types de véhicules. Il propose différents algorithmes de poursuite de véhicules, notamment l'algorithme Krauß[167] qui décrit les mouvements de chaque véhicule en fonction des positions et vitesses des véhicules qui se trouvent à proximité. Dans ce modèle, chaque véhicule essaie d'atteindre sa "vitesse désirée" tout en maintenant une distance de sécurité parfaite avec le véhicule à l'avant. Ce simulateur, sous licence libre, est codé en C++, est paramétrable grâce à des fichiers XML, comme l'est le simulateur VanetMobiSim. Depuis 2007, une extension de SUMO nommé MOVE permet de générer des traces de véhicules compatibles avec les logiciels de simulation de réseau de communication. Avec douze citations (figure 3.16), ce simulateur est le plus utilisé dans les études de protocole de routage dans les VANETs [68, 108, 63, 82, 70].

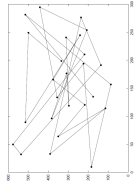
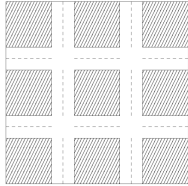
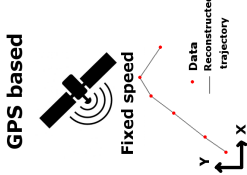
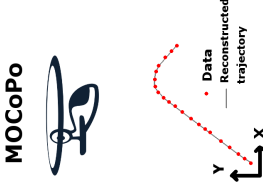

Le simulateur *NETwork Simulator for TRaffic Efficiency And Mobility* (NetStream) [164] a été développé par Toyota Central R&D Labs. Il permet de simuler la mobilité des véhicules tels que les changements de voie ou la vitesse limite. Il a été utilisé pour évaluer le protocole GVGRID [122]. Cependant, avec une seule citation (figure 3.16), ce simulateur de trafic est peu utilisé. De plus, sa dernière version est ancienne (1997) et il n'est plus maintenu depuis.

Le simulateur NCTUns [165], déjà présenté en section 3.3.1, gère les multi-voies et les intersections avec un système de feux tricolores. Avec une seule citation (figure 3.16), il est peu utilisé par la communauté académique, mais il a notamment servi à évaluer le protocole C-DRIVE [58].

Synthèse et choix retenu

Le tableau 3.3 présente une synthèse des différentes approches présentées ci-avant. Étant donnée notre choix précédent de simulateur de communication, nous avons besoin des traces de véhicules représentant la mobilité des nœuds du VANET. Les modèles basés sur l'aléatoire ne sont pas assez réalistes pour simuler la mobilité des véhicules. Les traces réelles issues de capteurs GPS ou de vidéos ne sont pas adaptées, car elles ne contiennent soit pas assez de véhicules proches ou soit un réseau trop restreint. La dernière option s'offrant à nous est l'utilisation d'un simulateur microscopique. Notre choix s'est orienté vers le simulateur de trafic intégrant un modèle microscopique SUMO, très utilisé par la communauté scientifique. Il a l'avantage d'être libre, paramétrable et assez facile d'utilisation. Il ne faut pas perdre de vue que l'utilisation des simulateurs de trafic introduit un biais dans les résultats des performances des protocoles de routage. Comme nous l'avons montré dans [138], il est difficile de reproduire des comportements réels d'automobilistes avec un simulateur de trafic. Les traces utilisées pour la mobilité des nœuds restent donc des traces synthétiques. Cependant, cette limite sera levée lorsque le taux de pénétration des véhicules autonomes dans le parc de véhicules sera conséquent. Dans ce cas, la mobilité des véhicules sera issue d'algorithmes implantés dans les ordinateurs de bord des véhicules qui pourront être fidèlement retranscrits au sein des simulateurs de trafic microscopique.

TABLEAU 3.3 – Approches de modélisation de la mobilité dans les VANETs

Approche	Simple	Aléatoire	GPS	Traces réelles	Simulateur
Définition	Mobilité définie aléatoirement par une vitesse et une destination	Avec contraintes géographiques Mobilité définie aléatoirement par une vitesse et une destination en fonction des contraintes géographiques	Traces de mobilité réelles pour une situation donnée	Vidéo Traces de mobilités réelles pour une situation donnée	Microscopique Mobilité influencée par celle des nœuds voisins soumis à des modèles de suivi de véhicules
Exemple	RWP	Manhattan	Crowdad	MOCOPo	SUMO
Avantages	Mise en œuvre simple	Mise en œuvre simple	Réaliste et réutilisable	Réaliste et réutilisable	Réutilisable
Inconvénients	Irréaliste	Non prise en compte des interactions entre véhicules	Densité faible de véhicules ou véhicules éloignés	Section courte	Paramétrage difficile pour atteindre un comportement humain réaliste
Traces					

3.4.2 Paramétrages du simulateur SUMO

3.4.2.1 Utilisation de SUMO

L'utilisation de SUMO se décompose en plusieurs phases de configuration (voir figure 3.4) par le biais de fichier XML. Tout d'abord, il est nécessaire de définir le réseau routier sur lequel les véhicules vont circuler. Pour cela, il est possible de le définir manuellement ou à partir d'un réseau routier réel. Un outil de visualisation permet de vérifier le réseau routier ainsi construit. Ensuite, il convient de définir la demande en trafic sur ce réseau routier. Enfin, il faut choisir les traces que l'on souhaite obtenir en lançant la simulation. Un outil de visualisation permet de vérifier l'ensemble de la simulation. Cette procédure est détaillée dans l'annexe D à la page 233. Dans cette thèse, des traces de véhicules ont été générées par SUMO pour deux types de réseaux routiers différents. Le premier type correspond à des réseaux fictifs très simple (section 3.4.2.3) et le second à un réseau réel composé de l'autoroute A27 (section 3.4.2.4).

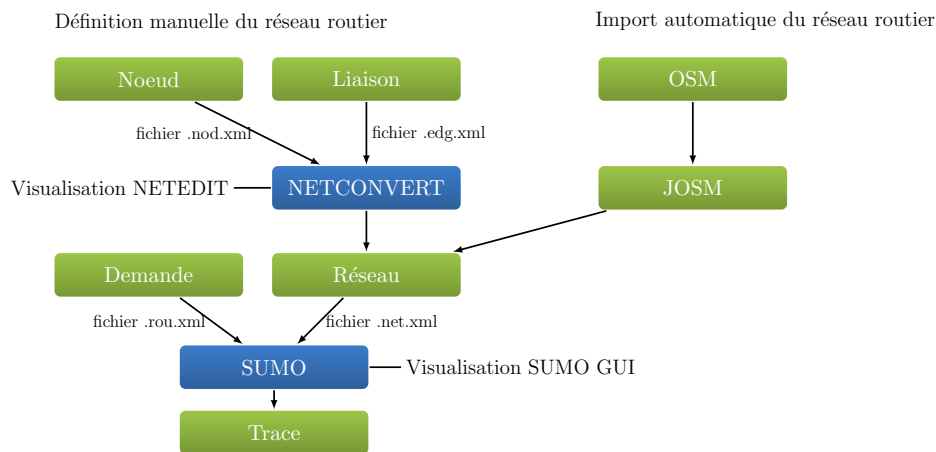


FIGURE 3.4 – Diagramme d'utilisation de SUMO

3.4.2.2 Modèle de poursuite utilisé dans la simulation SUMO

Nous avons utilisé le modèle Krauß [167] implémenté dans SUMO. La limite de vitesse est fixée à 130 km/h pour les réseaux fictifs et le réseau de l'autoroute A27 (respectivement section 3.4.2.3 et 3.4.2.4). Cette vitesse correspond à la limite de vitesse légale des autoroutes en France. Nous définissons deux types de véhicules : les voitures (Véhicules légers (VL)) et les poids lourds (Poids lourds (PL)). Les véhicules sont modélisés par un ensemble de paramètres (les valeurs sont données dans le tableau 3.4) :

- Acc : la capacité d'accélération des véhicules ;
- Dcc : la capacité de décélération des véhicules ;
- L : la longueur du véhicule ;
- σ : l'imperfection du pilote (entre 0 et 1) ;
- τ : le temps inter-véhiculaire souhaité par le conducteur ;
- MinGap (*MG*) : l'espace minimal entre deux véhicules en cas de congestion ;
- MaxSpeed (*MS*) : la vitesse maximale du véhicule ;
- SpeedFactor (*SF*) : le facteur multiplicateur concernant la vitesse maximale ;
- SpeedDev (*SD*) : l'écart type de SpeedFactor.

Pour simuler un comportement de conduite automobile réaliste [138], il est nécessaire d'utiliser des distributions de vitesse pour la vitesse désirée. Sans l'utilisation de ces distributions, tous les véhicules auraient la même vitesse désirée et il n'y aurait pas de dépassement de véhicules. Cette situation serait irréaliste. Par conséquent, deux autres paramètres ont été introduits afin d'utiliser les distributions de vitesse dans SUMO : *speedFactor* et *speedDev*. Par exemple, l'utilisation de *speedFactor* = 1 et *speedDev* = 0,1 entraînera une distribution de vitesse pour laquelle 95% des véhicules conduisent à une vitesse allant de 80% à 120% de la limite de vitesse légale (figure 3.5).

TABLEAU 3.4 – Paramètres utilisés pour le modèle Krauß pour les VL et PL

	Acc	Dcc	L	σ	τ	MG	MS	SF	SD
Unités	m/s^2	m/s^2	m	-	s	m	km/h	-	-
VL	2	3	5	0,5	1	2,5	150	1	0,1
PL	1	2	15	0,5	1	5	130	0,84	0,1

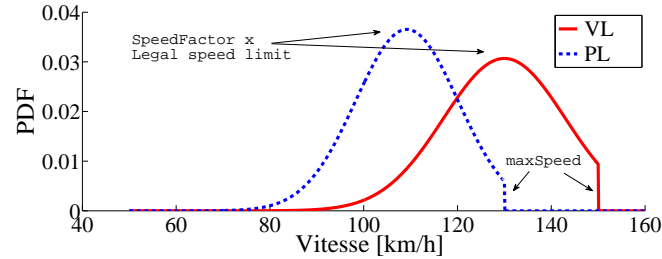


FIGURE 3.5 – Densité de probabilité de la vitesse désirée pour les VL et les PL dans le cas d'une vitesse maximale autorisée à 130 km/h.

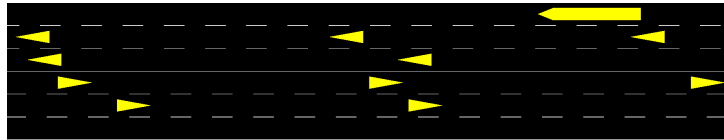
3.4.2.3 Présentation des réseaux fictifs - scénarios S1-S9

Nous avons défini trois réseaux fictifs très simples :

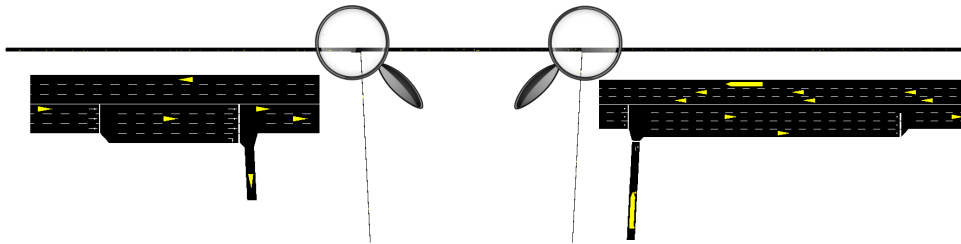
- R1 : une autoroute à sens unique à trois voies sur 5 km de long ;



- R2 : une autoroute à double sens à trois voies sur 5 km de long ;



- R3 : une autoroute à deux fois trois voies avec une entrée et une sortie d'autoroute. La sortie est située à 1,8 km du début de la section, tandis que l'entrée se trouve à 3 km.



Trois cas de densité de trafic différents sont pris en compte. Dans chaque cas, un ratio de 1/6 camions et 5/6 voitures est considéré. Les différentes densités de trafic utilisées sont listées dans le tableau 3.6. Dans le réseau R3, 25% des véhicules arrivent par l'entrée de l'autoroute, 25% des véhicules prennent la sortie et 50% des véhicules traversent toute la section de la route.

3.4.2.4 Réseau réel, exemple de l'A27 - scénarios S10-S13

Deux réseaux réels R4 et R5 ont été définis. Le premier se compose d'une section de 5 km de l'autoroute A27 et d'une section de 1 km de la départementale RD90. Le second ne contient que la section de l'autoroute A27. Cette autoroute relie Lille à la Belgique. La figure 3.6 illustre l'import du réseau A27 issue de Open Street Map grâce à l'outil JOSM dans NETEDIT.

TABLEAU 3.5 – Définition des scénarios et de la valeur de la demande de trafic routier

Densité	Trafic VL (<i>veh/h/direction</i>)	Trafic PL (<i>veh/h/direction</i>)	R1	R2	R3
Faible	500	100	S1	S4	S7
Moyenne	2000	400	S2	S5	S8
Haute	4000	800	S3	S6	S9

Où S1 à S9 sont les scénarios.

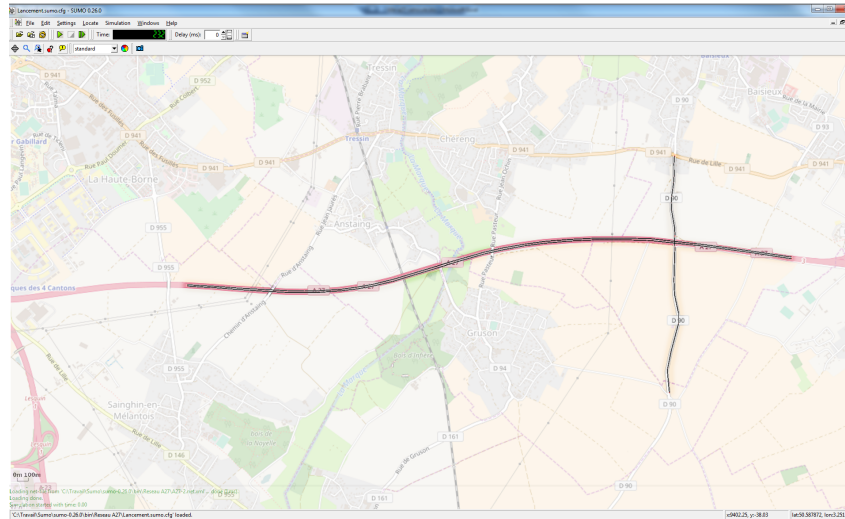


FIGURE 3.6 – Réseau A27 importé dans Netedit

Nous avons pu nous procurer des données de trafics routiers concernant l'autoroute A27. Ces données ont été fournies par la Direction Interdépartementale Des Routes (DIR) Nord et sont issues de boucles électromagnétiques placées sous la chaussée. Nous avons pu obtenir les données de débit de véhicules pour un jour de semaine hors vacances scolaires : le jeudi 6 avril 2017. Ces données sont agrégées heure par heure.

La figure 3.7 présente le débit de véhicules sur l'autoroute A27 pour le 6 avril 2017. La courbe bleue représente le débit de VL et la courbe noire le débit de PL. On observe une forte variabilité du débit en fonction de l'instant de la journée.

Concernant le sens Belgique-France, le trafic PL est faible la nuit de 20h à 4h puis augmente jusqu'à 400 veh/h et se maintient dans la journée. Le trafic VL est faible la nuit de 23H à 5H puis on observe un pic de débit à l'heure de pointe du matin (6H-8H) jusqu'à 1500 veh/h, ensuite une stagnation autour de 1000 veh/h dans la journée jusqu'à 18H, puis une chute de débit entre 18H et 23H. On peut en déduire au vu du pic de débit à l'heure de pointe du matin, que de nombreuses personnes de Belgique viennent travailler en France.

Concernant le sens France-Belgique le trafic PL est semblable à celui observé dans le sens Belgique-France. Le trafic VL est faible la nuit de 23H à 5H puis on observe un petit pic de débit lors de l'heure de pointe du matin (6H-8H) jusqu'à 700 veh/h, ensuite une chute de débit de 8H à 9H suivi d'une augmentation entre 9H et 17H pour atteindre un pic de débit à 1700 veh/h à 17H. Le débit diminue ensuite entre 18H et 23H. On peut en déduire au vu du pic de débit à l'heure de pointe du soir, que de nombreuses personnes reviennent de leur travail de France vers leur domicile en Belgique.

Par la suite nous avons pu obtenir des données plus précises agrégées sur une minute à partir de 12 :35 issues d'un capteur radar. La figure 3.8 présente ces données. On observe une variabilité du débit minute par minute en fonction du sens de circulation et du type de véhicules. Par le biais de ces données plus précises, nous avons pu générer des traces de véhicules avec le simulateur SUMO correspondant à une demande réaliste en distinguant les VL et les PL.

Quatre nouveaux scénarios ont été définis (S10-S13) en fonction de la densité de trafic considérée et de la prise en compte de la RD90 ou non. Dans le premier scénario (S10), la densité de trafic de l'autoroute

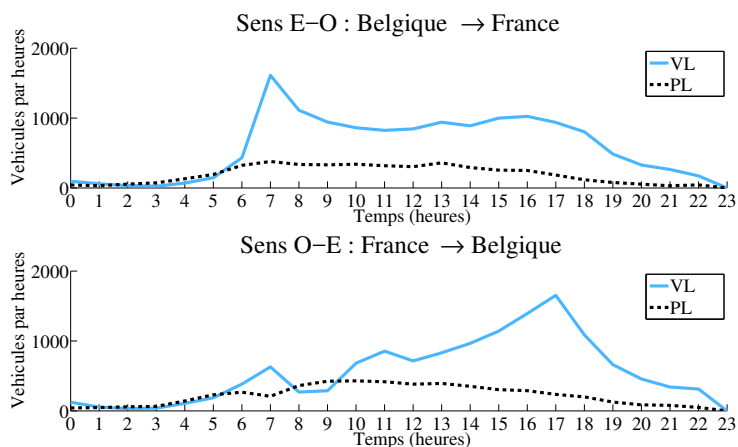


FIGURE 3.7 – Demande de véhicules sur une boucle électromagnétique placée sous la chaussée pour la journée du 06-04-2017. En haut de la figure, dans le sens Belgique-France, en bas de la figure dans le sens France-Belgique.

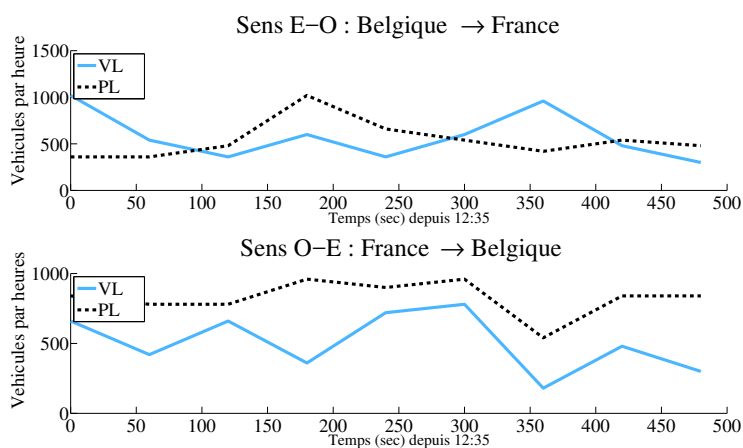


FIGURE 3.8 – Demande de véhicules issue d'un capteur radar pour la journée du 06-04-2017 à partir de 12 :35. En haut de la figure, dans le sens Belgique-France, en bas de la figure dans le sens France-Belgique.

A27 reflète la situation du jeudi 6 avril 2017 à partir de 12 :35 (figure 3.8). La départementale RD90 est prise en compte dans ce scénario avec un trafic VL de 200 véhicules par heure et par sens. Le deuxième scénario (S11) est identique au premier excepté le fait qu'il ne prend pas en compte la RD90. Le troisième (S12) et le quatrième scénario (S13) ne prennent pas en compte la RD90. La densité de véhicules sur l'autoroute A27 pour le troisième scénario correspond à celle observée durant le pic du matin de 7H à 8H ; et celle du quatrième scénario correspond à la faible demande de trafic de 6H à 7H du matin figure 3.7.

TABLEAU 3.6 – Scénarios et valeurs de densité de trafic

Scénario	Réseau		Densité A27 (veh/h)				Densité RD90 (veh/h)				
			Sens O-E		Sens E-O		Sens O-E		Sens E-O		
	A27	RD90	VL	PL	VL	PL	VL	PL	VL	PL	
S10	R4	✓	✓	Densité variable - Fig. 3.8				250	0	250	0
S11	R5	✓		Densité variable - Fig. 3.8				0	0	0	0
S12	R5	✓		630	208	1620	380	0	0	0	0
S13	R5	✓		380	260	430	320	0	0	0	0

3.4.2.5 Limites et solutions de mise en œuvre des simulations sous OPNET

Dans le cadre de cette thèse, compte tenu des densités de véhicules de chaque scénario, le nombre de nœuds dans le réseau est très élevé. Il dépasse les capacités de l'architecture matérielle dont nous disposons pour la simulation avec OPNET. Par exemple, le nombre total de nœuds nécessaires au lancement d'une simulation sous Riverbed OPNET Modeler de 500 secondes est d'environ 666 nœuds pour le scénario S5 qui comporte une densité moyenne de véhicules (2000 VL/heure/direction et 400 PL/heure/direction). Ce nombre de 666 nœuds s'est avéré trop élevé pour lancer une simulation (*crash* du logiciel).

Pour nos scénarios routiers, il est également nécessaire d'initialiser chaque nœud du réseau au moment où il entre dans la zone de simulation. Or, par défaut, les nœuds d'une simulation sous OPNET sont initialisés dès le début de la simulation et stationnent à leur position initiale un laps de temps paramétrable. Si, avant d'entrer sur la section du réseau, le nœud est initialisé à une position donnée, il émet des messages de découverte de voisinage trop tôt. Il va ainsi modifier les statistiques collectées durant la simulation affectant les résultats finaux.

Pour résoudre ces deux difficultés, nous avons réduit le nombre de nœuds en les réutilisant à différents instants de la simulation avec des trajectoires de véhicules qui ne sont pas présents simultanément dans le réseau routier. Ces trajectoires sont compilées pour un nœud donné dans le fichier ".trj" affecté au nœud (annexe C page 227). Au lancement de la simulation, tous les nœuds attendent dans une zone tampon éloignée du réseau routier. Ces nœuds sont ensuite initialisés lorsqu'ils sortent de la zone tampon. Lorsqu'un nœud a terminé le parcours d'une première trajectoire, il retourne immédiatement dans la zone éloignée du réseau où il est réinitialisé dans un état d'attente. Dès qu'il sortira de cette zone pour parcourir une nouvelle trajectoire, il sera initialisé de nouveau. Cette optimisation a permis de réduire le nombre de nœuds nécessaires au lancement des simulations (tableau 3.7) d'une part et, d'autre part, de ne pas impacter les statistiques des métriques par les émissions des nœuds en attente. Cependant, malgré cette solution, les scénarios S6 et S9 n'ont pu être lancés en raison du nombre élevé de véhicules présents simultanément dans le réseau.

TABLEAU 3.7 – Nombre de nœuds nécessaires à chaque scénario sous Riverbed OPNET Modeler

Scénario	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Nombre de nœuds	62	262	567	138	532	999	123	522	1454	380	328	357	175

3.5 Modèles de propagation des ondes radio

3.5.1 Modèles de propagation des ondes radio choisis dans les travaux référencés au chapitre 1

Les études de performances [168, 169, 170, 171, 172] montrent que le choix d'un modèle de propagation des ondes radio a un fort impact sur les évaluations de performances des protocoles de routage. Un état de l'art complet des modèles disponibles à ce jour est réalisé dans [173].

L'onde radio est une onde électromagnétique qui interagit avec son environnement de propagation, ce qui rend difficile une modélisation de la réalité de terrain. La propagation de cette onde électromagnétique est en effet sujette à un phénomène d'affaiblissement (*path-loss*) fonction de la distance entre l'émetteur et le récepteur et causé par la dispersion de sa puissance dans l'espace. En plus de cet affaiblissement, l'onde subit une atténuation (aussi appelée *large scale fading*) lors de sa rencontre avec des obstacles amenant à la création de zones d'ombre (*shadowing*), par exemple causées par des dispersions du signal sur un feuillage, des réflexions du signal sur le sol ou des bâtiments, des réfractions du signal sur des bâtiments. Enfin, la puissance de l'onde mesurée à sa réception montre des fluctuations rapides (appelées *small scale fading*) dues au retard induit par des multi-trajets et par l'effet Doppler.

Différents modèles de propagation non spécifiques aux VANETs sont souvent utilisés pour la modélisation de ces réseaux (tableau 3.8) : les modèles déterministes d'atténuation en espace libre (*Free-Space Path-Loss Model* ou *Unit-Disc*) et à deux rayons (*Two-Ray*) d'une part et les modèles probabilistes à zones d'ombre (*Probabilistic Shadowing Model*) et Nakagami d'autre part. Les formules mathématiques

de ces modèles, qui relie la puissance émise par un nœud à celle reçue par un autre, sont disponibles dans l'un et l'autre des articles [174, 175].

Le modèle d'atténuation en espace libre (*Free-space*), très couramment utilisé pour l'étude des VANETs (tableaux 3.13, 3.14 et 3.15), simule le cas où les deux nœuds, émetteur et récepteur, sont en visibilité directe (*Line of Sight* (LOS)). Il repose sur l'hypothèse d'une relation inversement proportionnelle au carré de la distance entre la puissance reçue et celle émise et considère qu'en-dessous d'un seuil minimal de puissance de réception, le nœud récepteur n'est plus en capacité de décoder les signaux transmis. Par conséquent, au-dessus d'une certaine distance fixe entre l'émetteur et le récepteur (appelée portée radio), le modèle considère que le récepteur ne reçoit plus le signal émis.

Le modèle de propagation à deux rayons (*Two-ray*), choisi par [78, 91, 108, 59] pour l'évaluation de protocoles de routage, considère à la fois le signal LOS et le signal réfléchi par le sol. L'affaiblissement de puissance est calculé en fonction de la taille des antennes des nœuds émetteurs et récepteurs et de la distance qui les sépare. Cependant ce modèle ne peut pas être utilisé dans le cas d'une distance trop faible entre les deux nœuds car la combinaison du rayon direct avec celui réfléchi donne alors de mauvais résultats. C'est pourquoi [16] recommande l'utilisation hybride de ce modèle avec le modèle *Free-Space* : pour une distance d importante ($d > d_c$) entre les deux nœuds, le modèle à deux rayons est utilisé et pour une distance plus faible ($d < d_c$) le modèle *Free-Space* est employé. L'étude [176] montre que ce modèle *Two-Ray* n'est pas adapté à l'étude de la technologie WIFI 802.11p. En effet, en tenant compte de la fréquence des signaux émis par la technologie WIFI 802.11p et en considérant que les antennes sont positionnées sur le toit des véhicules, [176] a calculé une distance d_c de 886,6 m. Or, selon [177], la portée radio maximale expérimentée avec cette technologie pour une communication V2V est de 300 m à 850 m. Ainsi, la distance d_c de 886,6 m n'est jamais atteinte et l'utilisation de la solution hybride proposée par [176] devient équivalente à celle du modèle *free-space*.

Le modèle probabiliste à zones d'ombre (*Probabilistic Shadowing Model*) prend en compte la variabilité temporelle de la puissance modélisant par une approche stochastique les multi-trajets et les phénomènes de *fading*. La fonction mathématique de ce modèle à deux parties comporte deux paramètres, β et θ , calculés de manière empirique en fonction de l'environnement. La première partie du modèle considère en effet la moyenne de la puissance reçue en fonction du ratio entre la distance et les deux nœuds (émetteur et récepteur) et une distance de référence, ratio exprimé par une fonction puissance et un coefficient β dont l'objectif est de déterminer la rapidité de la décroissance de la puissance reçue. La deuxième partie considère une variable aléatoire de loi gaussienne, de moyenne nulle et de variance θ . Selon [171], les valeurs typiques de β sont comprises entre 2, pour un environnement ouvert, et 2,7 à 5, pour un environnement urbain ; celles de θ sont généralement comprises entre 4 et 12.

Enfin, le modèle probabiliste Nakagami, qui n'a été utilisé que récemment pour l'évaluation de performances des protocoles de routage [120, 71, 73, 72], modélise de façon plus réaliste le comportement des ondes. Ajoutant deux seuils de distance au précédent modèle, il applique des paramètres β et θ différents en fonction de la distance qui sépare deux nœuds.

En conclusion de leur étude de performances de deux protocoles de routage en fonction des modèles de propagation des ondes, les auteurs de [174] recommandent de choisir un modèle de propagation adapté au milieu routier dans lequel évoluent les véhicules (autoroutier, urbain ou rural). Par exemple dans le cas d'un milieu urbain, le modèle Nakagami est plus adapté que le modèle *Two rays* qui ne prend pas en compte les multi-chemins. À contrario, pour les milieux autoroutiers ou ruraux, [178] préconise l'utilisation du modèle à deux rayons. [175] propose de s'appuyer sur des tests terrain pour modéliser la propagation des signaux.

Plusieurs études ont donc eu pour objectif d'adapter, de calibrer ou de créer des modèles de propagation des ondes électromagnétiques plus réalistes en vue de l'étude de protocoles de routage pour les VANETs. Ces modèles capitalisent l'analyse des puissances enregistrées en réception lors de campagnes de mesure réalisées sur le terrain à l'aide de véhicules équipés de nœuds de communication sans fil (tableau 3.8).

Le lecteur intéressé trouvera dans [179] une analyse des résultats de mesures de la puissance reçue à une intersection en milieu urbain pour des émissions réalisées à des distances de 30 m et 50 m, et dans [180], une étude sur des mesures réalisées en environnement urbain et autoroutier où les auteurs

TABLEAU 3.8 – Modélisation de la propagation des signaux dans les VANETs

Modèle	Type d'analyse		Contexte de propagation				Environnement adapté			Modèle calibré avec des mesures				
	Modèle	Mesure	LOS	NLOS	OLOS	Fadding	Autoroute	Urbain	Rural	Free Space	Two Rays	PSM	Nakagami	Autre
Général	Espace libre	✓	✓				✓		✓					
	Two Rays	✓	✓	✓			✓		✓					
	PSM	✓		✓	✓	✓	✓		✓					
	Nakagami	✓		✓	✓	✓								
VANET	[179]		✓					✓						
	[180]		✓	✓	✓		✓	✓						
	[181]	✓	✓	✓		✓	✓							A
	[182]	✓	✓	✓	✓	✓	✓	✓	✓		R	AU		
	[9]	✓	✓	✓		✓	✓	✓						AUR

A : Autoroute, U : Urbain, R : Rural.

s'intéressent à l'impact des véhicules voisins sur la propagation des signaux et notamment aux cas avec obstruction des signaux.

[182] propose un modèle de propagation d'ondes différent pour chaque milieu routier. Il emploie et calibre, à l'aide des résultats obtenus lors d'une campagne de mesures, le modèle *two rays* en milieu rural, et le modèle probabiliste à zones d'ombre pour les autres milieux. [181] tient compte de l'impact des véhicules voisins sur la propagation des signaux dans le modèle autoroutier qu'il propose.

Le modèle proposé par [9] (figure 3.9) s'appuie sur une campagne de mesures menée sur autoroute et en milieu urbain dans le sud de la Suède entre les villes de Lund et Malmö avec des véhicules équipés d'antennes omnidirectionnelles placées sur le centre du toit des véhicules. Plusieurs routes ont été parcourues avec des environnements différents en termes de densité de trafic et d'environnements routiers (obstacles, immeubles...). Avec le recueil des mesures expérimentales, les auteurs de [9] ont proposé de distinguer les signaux reçus en ligne directe et ceux reçus en ligne directe obstruée, ou partiellement obstruée par un véhicule voisin (*Obstructed Line of Sight* (OLOS)). Les auteurs ont proposé deux paramétrages différents d'un modèle doubles pentes (*dual-slope*) pour décrire les données expérimentales, un pour le cas urbain et un autre pour le cas autoroutier. Celui adapté à l'environnement autoroutier est présenté en figure 3.9. Les mesures pour le cas de ligne directe LOS sont représentées par des points bleus tandis que ceux du cas OLOS sont en vert. Le modèle pour le cas LOS correspond à la courbe noire tandis que la courbe rouge représente celui pour le cas OLOS. On remarque qu'aucune mesure n'a enregistré pour le cas OLOS pour une distance entre l'émetteur et le récepteur inférieure à 80 m. En effet pour que le cas OLOS se produise il faut qu'un autre véhicule se place entre l'émetteur et le récepteur ce qui est rare sur autoroute lorsque l'émetteur et le récepteur sont à moins de 80 m de distance.

Synthèse et choix retenu

En conclusion, la complexité des phénomènes de propagation des ondes radio a mené à la création de différents modèles, adaptés ou non aux VANETs (tableau 3.8). La variété des configurations routières (infrastructures, mobilité, densité de véhicules, etc.) est telle qu'il est optimiste de penser trouver un modèle représentatif d'un trafic routier particulier. Cela nécessiterait en effet une campagne de mesures d'envergure au cours de laquelle seraient notamment enregistrées les trajectoires des véhicules sur le réseau routier étudié, ainsi que, pour chaque couple de véhicules à portée de communication, les profils d'atténuation de la puissance en réception en fonction du temps et de leur distance relative. Les pertes de communication pour cause de masquage par exemple, pourraient alors seulement être représentatives d'une réalité de terrain. Or, à la difficulté de mettre en œuvre une telle campagne de mesures, s'ajoute celle d'un parc de véhicules non encore connectés.

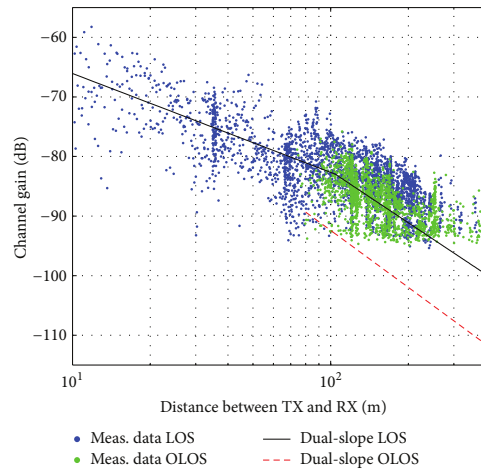


FIGURE 3.9 – Puissance mesurée en environnement autoroutier et modèles associés [9]

En l'absence d'une telle connaissance, nous avons choisi d'utiliser deux modèles de propagation :

- le modèle de propagation en espace libre ; actuellement très utilisés dans les études d'évaluation de protocoles de routage pour les VANETs (voir tableaux 3.13, 3.14 et 3.15), il est simple à mettre en œuvre. La portée de transmission est son seul paramètre à régler.
- le modèle de propagation [9] construit à partir de mesures sur autoroute ; ce modèle prend en compte les effets de masquage partiel (OLOS) du signal en présence de véhicules intermédiaires. Le seuil de réception est son seul paramètre à régler.

3.5.2 Modélisation de la propagation sous OPNET Riverbed Modeler

3.5.2.1 Description du modèle de propagation en espace libre

Le modèle de propagation d'ondes, implanté par défaut sous OPNET, est celui d'atténuation en espace libre. Pour cela, OPNET met en œuvre quatre processus. Un premier calcule, à chaque transmission et en fonction de la distance entre l'émetteur et le receveur, le taux de puissance reçue à partir des valeurs de gains de l'antenne de transmission et de celle de réception et du modèle de l'affaiblissement de la propagation du signal en espace libre. Un deuxième détermine le rapport du signal sur bruit (*Signal Noise Ratio* (SNR)) au niveau de la réception du signal en fonction de la puissance de réception, du bruit thermique et du bruit des interférences causées par des collisions de paquets. Un troisième processus évalue le taux d'erreur binaire en utilisant une table de modulation telle que celle du modèle défini par défaut illustrée en figure 3.10. Une table de modulation donne le taux d'erreur binaire en fonction du taux défini en échelle logarithmique du SNR ($SNR = Eb/No$ avec Eb l'énergie par bit reçu en joules et No la densité du bruit de la puissance spectrale (Watt/hertz)). Un quatrième et dernier processus vérifie, à réception du signal, si les paquets reçus ont un nombre de bits en erreur inférieur à un seuil (ici égal à 0,20). Si ce n'est pas le cas, les signaux ne peuvent pas être décodés.

Les paramètres utilisés sous OPNET Riverbed Modeler sont repris dans le tableau 3.9. La puissance a été fixée à 0,005 W et le seuil de réception à -95 dBm permettant ainsi d'obtenir une portée maximale de 500 mètres qui correspond à celle attendue par la technologie WIFI 802.11p.

TABLEAU 3.9 – Configuration des nœuds dans OPNET

Attributs	Valeur
Fréquence minimale	5,855 GHz
Bande passante	10 Mhz
Puissance de transmission	0,005 W (i.e. 7 dBm)
Seuil de réception	-95 dBm
Portée maximale	500 m

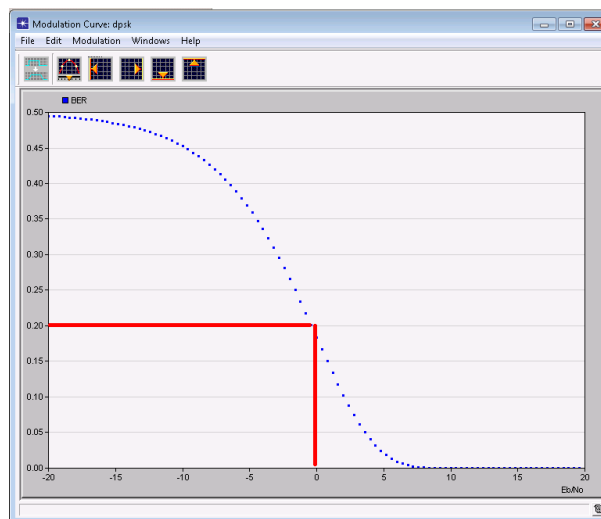


FIGURE 3.10 – Modèle de propagation d’onde intégré par défaut sous OPNET Riverbed Modeler

3.5.2.2 Mise en œuvre du modèle construit à partir de mesures réelles

Le second modèle utilisé [9] est ainsi mis en œuvre. La figure 3.11 fournit la valeur du seuil de la puissance reçue pour lequel la portée de communication est égale à 500 m dans le cas LOS (courbe bleue) : -85 dBm. Ce seuil diffère de celui choisi pour le modèle de propagation en espace libre (tableau 3.9). En effet, ce seuil dépend en particulier de la puissance de transmission qui est de 20 dBm dans l’étude [9] et de 7 dBm dans notre précédente configuration. Nous avons cherché à garder la configuration des nœuds du tableau 3.9. La valeur du seuil de puissance reçue de -85 dBm donne dans le cas du modèle OLOS (courbe verte) une distance maximale de communication entre deux nœuds de 210 m.

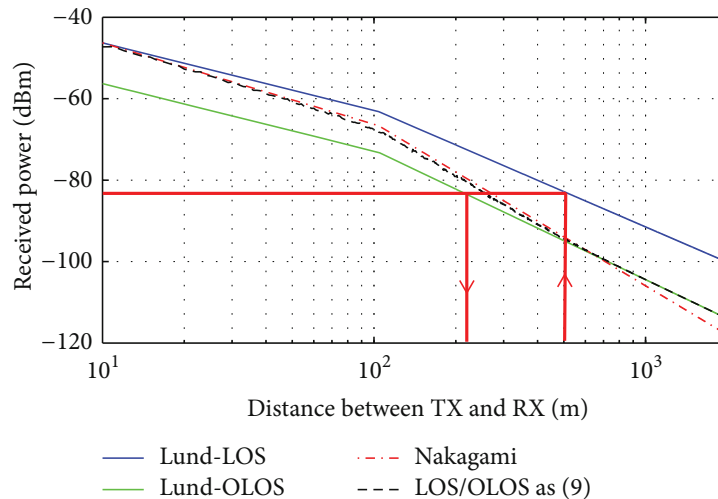


FIGURE 3.11 – Puissance moyenne reçue pour le modèle LOS/OLOS et le modèle Nakagami [9]

La prise en compte de ce second modèle a été réalisée par programmation dans le modèle du processus CBL développé, donc au niveau de la couche réseau. À chaque réception d’un paquet, le nœud récepteur évalue la distance le séparant du nœud source du paquet. Si cette distance est inférieure à 210 m, il considère que le contenu du paquet peut être lu quelle que soit la configuration de propagation (LOS ou OLOS) et accepte donc sa réception. Si au contraire cette distance est supérieure à 210 m, le nœud récepteur vérifie la configuration de propagation [183] (figure 3.12). Si au moins un nœud voisin se situe dans le rectangle de quatre mètres de large reliant le nœud source du nœud récepteur, le paquet est considéré comme non lisible et le nœud récepteur le rejette. Si la configuration est en lien direct (LOS), le paquet en provenance de la couche physique est considéré comme lisible et il est accepté au niveau CBL.

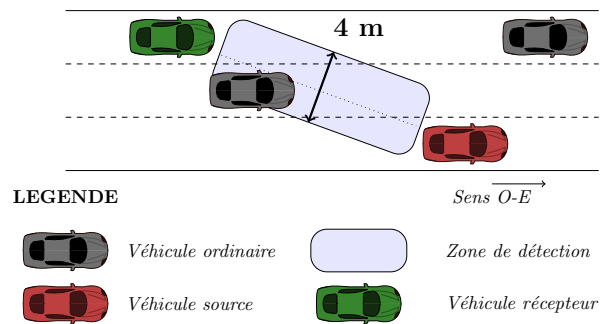


FIGURE 3.12 – Exemple de configuration OLOS – simplification de l’ellipse de Fresnel par un rectangle de quatre mètres de large

3.6 Modèle de la technologie de transmission de trames

3.6.1 Technologies existantes

Nous avons présenté au chapitre 2, le fonctionnement du schéma de *clustering* CBL. Ce schéma, qui se greffe dans un protocole de routage, est indépendant de la couches physique et liaison de données de la technologie radio utilisée. La structure “dynamiquement stable” construite par CBL ne prend en effet en compte que l’invariant de la portée radio théorique de la technologie employée. Cependant, une évaluation complète de ce protocole dans des conditions proches du terrain nécessite de simuler cette structure en prenant en compte une technologie sous-jacente permettant un accès physique au médium ainsi que de vérifier qu’elle offre les services nécessaires au transport de trafic applicatif.

Le modèle IEEE 802.11p est actuellement le seul standard conçu pour les VANETs qui permet une communication directe entre véhicules. Selon la modulation choisie, cette technologie offre des débits théoriques de 3 à 27 Mbps.

Quant à la version 13 de la technologie LTE défendue par l’Europe, elle permet également des communications *Device-to-Device* (D2D), mais à un saut, et réservées à des fins de sécurité publique d’une part et sous coordination d’une borne statique appelée eNodeB pour l’obtention d’un accès au médium de deux composants dans une même cellule ou entre un composant de la cellule et l’autre en bordure, d’autre part. La version commerciale du LTE R13 n’autorise que la phase de découverte du D2D. Les communications directes sans intervention d’un eNodeB ne sont pas possibles.

Si la génération 4G (comme les 2G et 3G) était centralisée sur l’usager humain, la technologie 5G à venir a pour vocation d’être plus flexible et de permettre à terme les communications de véhicules à véhicules ([184]). La technologie LTE est donc en pleine évolution, en particulier pour ce qui concerne les communications V2V à plus d’un saut, qui ne devraient être prises en compte qu’à partir de la seconde phase de développement de la 5G.

Notre choix s’est donc porté sur la technologie IEEE 802.11p. Nous avons utilisé pour l’ensemble des simulations réalisées sous OPNET Riverbed Modeler le modèle IEEE 802.11p disponible avec la licence WIRELESS d’OPNET Riverbed Modeler.

3.6.2 Description du protocole IEEE 802.11p

Le protocole de communication IEEE 802.11p est défini sur les couches 1 et 2 du modèle de référence OSI de l’ISO pour un médium de communication sans fil, soit en espace libre.

La norme prévoit un codage des signaux sur le médium selon la technique “d’Accès multiple par division de fréquences orthogonales”, *Orthogonal Frequency Division Multiple Access* (OFDM) [185]. La bande de fréquence de 5,850 à 5,925 GHz aux États Unis (5,855 à 5,925 GHz en Europe) est divisée en sept canaux de 10 Mhz avec un premier canal de contrôle et six canaux de service [6]. Le canal de contrôle est réservé à la transmission des messages de gestion du réseau. Les six autres canaux sont dédiés à la transmission des données applicatives annoncées sur le canal de contrôle. Cette méthode s’appuie sur la

norme 802.11a qui utilise des canaux de 20 Mhz. Pour le 802.11p, la taille des bandes de fréquence des canaux, et donc le débit, a été divisée par deux pour améliorer la prise en compte des signaux réfléchis et limiter l'effet Doppler. Jusqu'à huit débits de transfert de données (3 ; 4,5 ; 6 ; 9 ; 12 ; 18 ; 24 et 27 Mbit/s) sont possibles avec la méthode OFDM selon les méthodes de modulation et de codage utilisées. La portée de transmission de la norme 802.11p s'étend jusqu'à 1 km. Quatre catégories de puissance sont autorisées Puissance isotrope rayonnée équivalente (EIRP) allant de 23 dBm à 44,8 dBm pour les USA et de 23 dBm à 33 dBm en Europe.

L'accès distribué au médium de communications est géré par la couche MAC par un algorithme, de type compétition d'accès multiple par détection de porteuse avec évitement de collision appelé (*Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA)). Chaque nœud souhaitant préempter le médium l'écoute durant une durée T_{AIFS} (*Arbitration Interframe Space*). À l'issue de ce temps d'écoute, si le médium est libre, le nœud émet (figure 3.13), sinon il diffère son émission d'un nombre d'intervalles de temps (*Time Slot*) aléatoirement choisi. Ce nombre aléatoire appelé *backoff* est tiré dans une fenêtre de contention (*Contention Window*). Lorsque le nœud doit différer son émission, il écoute de nouveau la porteuse durant T_{AIFS} . Si le médium est libre à l'issue de cette période, il diffère encore sa transmission du nombre de *Time Slot* aléatoirement tiré préalablement à sa tentative de préemption du médium. Cependant, si lors de l'un de ces *Time Slot*, le médium n'est plus libre, le nœud gèle la décrémentation du nombre *backoff* jusqu'à la libération de la porteuse, puis le temps inter-trame T_{AIFS} . C'est seulement à l'issue de ce décompte, que le nœud pourra émettre si le médium est toujours libre.



FIGURE 3.13 – Algorithme CSMA/CA de la norme 802.11p d'après [6, 186]

Le délai inter-frame est ainsi calculé :

$$T_{AIFS}(AC) = SIFS + AIFSN(AC) * SlotTime$$

Où *Short Interframe Space (SIFS)* est la durée requise pour que l'interface sans fil traite la trame reçue et émette une trame de réponse et *AIFSN* ou *Arbitration Interframe Space Number*.

Les valeurs par défaut sont :

$$\begin{cases} SIFS = 32 \mu s \\ SlotTime = 13 \mu s \end{cases}$$

T_{AIFS} est la durée écoulée entre deux transmissions de trames. Sa valeur peut être différente selon la catégorie d'accès ou priorité, *Access Category (AC)*. Elle repose sur le mode *Enhanced distributed coordination function (EDCA)* de l'amendement IEEE 802.11e. La norme 802.11p distingue quatre catégories d'accès correspondant chacune à un type de trafic : AC0 qui donne une priorité *background* quand le trafic est faible, AC1 qui donne une priorité appelée *best-effort* aux applications standards, AC2 pour une priorité à la vidéo (*video*) et AC3 pour une priorité donnée à la voix (*voice*) [185]. En fonction de la catégorie, les valeurs de certains paramètres sont modifiées [187] (tableau 3.10).

Catégorie	$CW_{min}(AC)$	$CW_{max}(AC)$	AIFSN(AC)	$T_{AIFS}(AC)(\mu s)$
AC0	$aCW_{min} = 15$	$aCW_{max} = 1023$	9	149
AC1	$aCW_{min} = 15$	$aCW_{max} = 1023$	6	110
AC2	$(aCW_{min} + 1)/2 - 1 = 7$	$aCW_{min} = 15$	3	71
AC3	$(aCW_{min} + 1)/4 - 1 = 3$	$(aCW_{min} + 1)/2 - 1 = 7$	2	58

TABLEAU 3.10 – Valeur des paramètres de la couche MAC du protocole 802.11p en fonction de la catégorie d'accès

Lorsque la communication s'effectue en mode *unicast*, le nœud destinataire doit émettre en retour de la bonne réception de la trame un accusé de réception (trame *Acknowledgment (ACK)*). Si le nœud source ne le reçoit pas, il augmente de deux la largeur de sa fenêtre de contention de $CW(AC)$ avant de tirer une nouvelle valeur aléatoirement *backoff* avant de retenter l'émission de la trame perdue. À l'initialisation, $CW(AC)$ a pour valeur $CW_{min}(AC)$ (avec $aCW_{min} = 15$) qui est fonction du type de trafic. La valeur de $CW(AC)$ est plafonnée par $CW_{max}(AC)$ (avec $aCW_{max} = 1023$).

3.6.3 Paramétrage dans Riverbed OPNET Modeler

La version de Riverbed OPNET Modeler utilisée est la 17.1 qui ne contenait pas initialement le modèle de nœud 802.11p. Ce modèle a donc été créé par rétro-ingénierie à partir de celui de la licence WIRELESS de la version 18.0 en prévision de cette thèse. Il avait été vérifié que les résultats de simulation obtenus pour des scénarios identiques étaient les mêmes avec OPNET Riverbed Modeler 17.1 et Riverbed Modeler 18.0. Le paramétrage du modèle de nœud 802.11p est présenté dans le tableau 3.11. Seul le premier canal de 10 Mhz de la bande de fréquence (5,855 - 5,925 GHz) alloué en Europe par ETSI est utilisé. Le débit de transmission est fixé à 12 Mbits/s et les messages de routage sont envoyés en catégorie d'accès AC0. Les messages applicatifs sont transmis en catégorie d'accès AC1.

TABLEAU 3.11 – Configuration des nœuds dans OPNET

Attributs	Valeur
Fréquence minimale	5,855 GHz
Bande passante	10 Mhz
Débit de transmission	12 Mbits/s
Puissance de transmission	0,005 W (i.e. 7 dBm)

3.7 Modélisation d'applications

3.7.1 Classification des applications

L'état de l'art proposé dans [188] classe les applications ITS en trois catégories (figure 3.14) : confort, optimisation du trafic et sécurité.

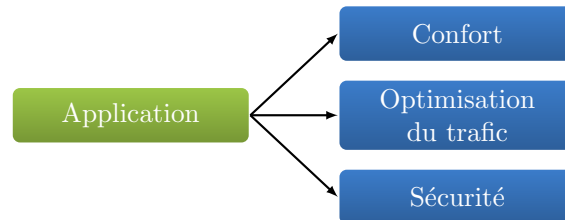


FIGURE 3.14 – Taxonomie des applications ITS dans les VANETs

Les “applications de confort” ont pour objectif de rendre plus agréable le voyage des automobilistes en leur fournissant un accès à Internet leur permettant d’accéder à des applications de divertissements (chargement de musique ou accès à des jeux distribués) ou des informations contextuelles relatives à leur déplacement (informations touristiques : restaurants, réservation de parking...).

Les “applications d’optimisation de trafic” visent à améliorer les conditions de circulation dans le but de réduire les embouteillages et la pollution. Elles s’intéressent à la gestion des flux de trafic et à la surveillance des véhicules et des conditions sur la route. Ces applications peuvent par exemple contribuer à améliorer le suivi d’itinéraires, apporter une assistance à l’insertion dans un flux de véhicules sur une autoroute, donner une vitesse recommandée au niveau des feux tricolores ou favoriser la conduite coopérative.

Les “applications de sécurité” ont pour objectif d’améliorer la sécurité des usages sur le réseau routier. Ces applications peuvent être classées en trois catégories, représentées en figure 3.15, en fonction du délai restant avant un probable accident à compter de leur date d’activation [189].

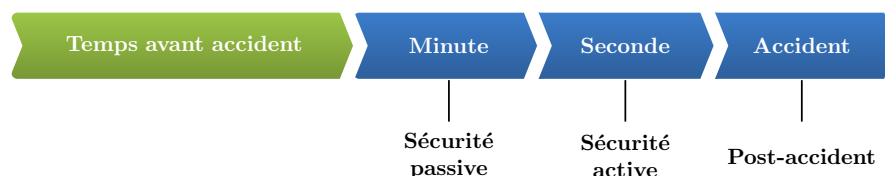


FIGURE 3.15 – Ordonnancement temporel des actions de sécurité dans les ITS

La sécurité passive intervient quelques minutes à quelques secondes avant un incident en prévention et en avertissement. Les applications de sécurité passive informent le conducteur du véhicule d’une situation potentiellement dangereuse, mais n’interfèrent pas avec l’action de conduite. Le conducteur garde le contrôle du véhicule. Ces applications de prévention et d’avertissement peuvent par exemple informer le conducteur d’une route verglacée, de risques de collisions dans une intersection sans visibilité ou de l’approche d’un véhicule d’urgence prioritaire.

La sécurité active intervient quelques secondes à quelques millisecondes avant un incident et son système prend le contrôle de l’action de conduite. Les applications de sécurité active interfèrent donc avec l’action de conduite. Ces applications peuvent concerner l’évitement de collision par changement de direction ou freinage d’urgence par exemple, le maintien du véhicule dans une voie et la conduite coopérative.

La sécurité post-accident intervient pendant et après un incident afin d’en limiter l’impact. Les applications de sécurité post-accident concernent l’avertissement d’accident et l’appel de détresse.

3.7.2 Besoins de communication

Ces différentes applications de sécurité (passive, active et post-accident) peuvent nécessiter des communications V2V, V2I ou *Infrastructure to Vehicle* (I2V). Le tableau 3.12 issu de [190] donne le type de communication requise (*i.e.* quel V2X) pour un ensemble d'applications de sécurité relatives à l'évitement de collision en intersection, à la sécurité publique, à l'extension de la signalisation ou encore nécessitant des informations en provenance d'autres véhicules. Pour chacune des applications listées, les besoins en termes de périodicité de transmission (transmission périodique ou événementielle), l'ordre de grandeur en millisecondes du délai de latence attendue entre la transmission de l'information et la prise en compte de l'information transmise et la portée maximale de la communication sont notifiés.

[190] montre ainsi que les applications de sécurité actives nécessitent des délais plus courts que les autres, d'environ 100 ms, avec des portées de transmission de l'ordre de 300 m. Cependant deux d'entre elles ont des contraintes temporelles plus critiques : les applications visant à former des convois de véhicules et celles de prévention des risques de collisions requièrent des délais inférieurs à 20 ms avec des distances de communication plus courtes, respectivement de l'ordre de 100 m et de 50 m. Les applications de sécurité passive, post-accident ou encore celles relatives à l'optimisation du trafic ont des contraintes temporelles moins restrictives avec des délais de bout-en-bout attendus inférieurs à 500 ms, voire 1 s.

D'autres exemples de la littérature définissent les exigences en communication d'applications. L'étude [23] par exemple spécifie une application d'information sur le trafic nécessitant un délai de transfert de messages inférieur à 100 ms. Dans une étude plus récente, [191] précise que les applications de conduite coopérative ont besoin d'un délai de bout en bout inférieur à 5 ms.

Les organismes de normalisation ETSI et IEEE proposent dans [10] d'utiliser un trafic applicatif d'une fréquence de 2 Hz avec des paquets de 800 bits (ETSI) et d'une fréquence de 10 Hz avec des paquets de 300 bits (IEEE).

Synthèse et choix retenu

En conclusion, choisir une application à modéliser signifie définir le type de communication à mettre en œuvre, le mode de communication (*broadcast, unicast...*), la taille et la fréquence d'envoi des messages applicatifs, enfin les délais maximaux de bout-en-bout autorisés par l'application.

Dans cette thèse, nous avons choisi de modéliser des applications de sécurité V2V ayant des exigences critiques en matière de délai de transmission des messages de bout-en-bout et de bandes passantes requises. Ces besoins constituent un véritable challenge pour les réseaux de communication ad hoc V2V. Trois applications et leurs besoins de communication ont ainsi été modélisés qui sont expliqués au chapitre 5 :

- une application utilisant des besoins applicatifs standardisés par l'IEEE (section 5.1) ;
- une application de localisation distribuée coopérative (section 5.2) ;
- une application de perception élargie (section 5.3).

3.8 Métriques de performances pour l'évaluation de protocoles de routage

Une métrique permet de mesurer de manière quantitative les performances d'un protocole de routage comparativement à un autre. Certaines métriques de performances pour l'évaluation de protocoles de routage sont communes aux métriques utilisées par les protocoles de routage présentées en section 1.3.2 comme par exemple le délai de bout en bout ou le nombre de sauts. D'autres métriques diffèrent, par exemple en ce qui concerne l'occupation de la bande passante : si elle est utilisée par le protocole de routage, elle permet d'évaluer la charge du réseau tandis que si elle est utilisée pour évaluer les performances d'un protocole de routage elle permet de mesurer l'occupation du trafic de routage. Les métriques usuelles utilisées pour évaluer et comparer des protocoles de routage sont les suivantes :

- Débit du trafic de routage : débit en bits/s des messages de routage envoyés sur le réseau ;
- Délai de bout en bout : délai de transmission de bout-en-bout d'un paquet ;
- Taux de paquets reçus (*Packet Delivery Ratio* (PDR)) : nombre de paquets reçus par rapport au nombre de paquets émis dans le réseau.

D'autres métriques de performances sont spécifiques aux protocoles de routage de *clustering*. Ces métriques sont :

- le nombre de *cluster* (Nb cluster) formés dans le réseau ;
- la durée d'élection du *cluster-head* (Durée CH : Cluster-Head) ;
- la durée de rattachement à un groupe par un nœud ordinaire (Durée CM : *Cluster-member*) ;
- le nombre de changements de la topologie du réseau.

Les tableaux 3.13, 3.14 et 3.15 référencent les métriques de performances utilisées dans les évaluations de protocole de routage de la littérature citées dans le chapitre 1.

Pour évaluer notre proposition d'organisation du réseau, nous avons défini différents types de métriques de performances : des "métriques structurelles" permettant d'évaluer la structure formée par CBL, des "métriques de routage" permettant d'évaluer le protocole de routage CBL-OLSR et des "métriques applicatives" permettant d'évaluer des scénarios applicatifs. Nous allons définir ces différentes métriques dans les chapitres 4 et 5 au cours des évaluations.

TABLEAU 3.13 – Choix de simulation pour l'évaluation des protocoles de routage de groupement de la littérature

Ref	Nom	Comm		Réseau - mobilité				Technologie			Application			Performance										
		Simulateur	Durée (s)	Simulateur	Longueur (km)	Nombre de voies	Densité (veh)	Vitesse (km/h)	Propagation	Portée (m)	BP (Mbits/s)	Type	Taille (Bytes)	Fréquence (Hz)	Comparaison avec les autres protocoles	Délat	PDR	Débit	Nb cluster	Durée CH	Durée CM	Changement		
[59]	BEDC	NS2	-	-	2	2*2	20-120	50-90	TR	-	-	FTP	1000	-	AODV	✓	✓	✓	✓	✓	✓	✓	✓	
[62]	PPC	NS2	-	CORS	10	2*2	100	-	U	250	-	Hello	75	-	LID	✓	✓	✓	✓	✓	✓	✓	✓	
[88]	VWCA	Matlab	600	Rand	2,5	2*3	10-350	70-120	U	100-1000	-	Hello	300	0,5	LID, HD, WCA	✓	✓	✓	✓	✓	✓	✓	✓	
[68]	SC	C++	650	SUMO	15	5	400	70, 90, 110	U	150-300	6	Hello	1000	10	WB, BB	✓	✓	✓	✓	✓	✓	✓	✓	
[108]	MBS	NS2	300	SUMO	15	2*2	50-100	30-140	TR	200	-	Hello	-	1	LID VMaSC	✓	✓	✓	✓	✓	✓	✓	✓	
[63]	DBR	NS2	100	SUMO	-	-	-	36-180	-	-	-	Hello	1500	-	CBR Flooding	✓	✓	✓	✓	✓	✓	✓	✓	
[82]	SCS	OMNET	180	SUMO	5	2*2	90, 180, 270	60-120	U	300	6	Hello	100	-	QoS OLSR, DC	✓	✓	✓	✓	✓	✓	✓	✓	
[58]	C-DRIVE	NCT	100-1000	NCT	3km ²	2	20-100	30-50	U	200	-	Hello	-	1	-	✓	✓	✓	✓	✓	✓	✓	✓	
[70]	DCEV	-	300	SUMO	-	4	30	30-120	U	300	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	
[89]	VMaSC	NS3	600	SUMO	1km ²	-	100	30-100	U	100-300	-	Hello	-	5	N-hop	✓	✓	✓	✓	✓	✓	✓	✓	✓
[89]	MPDB	NS2	200	Rand	-	4	0-50	0-120	U	250	2	CBR	512	0,5	V-TRADE	✓	✓	✓	✓	✓	✓	✓	✓	
[87]	ACA	NS2	80	SUMO	3	-	20-300	120-150	TR	310	6	UDP	512	-	HD, AWCP, CPM	✓	✓	✓	✓	✓	✓	✓	✓	
[91]	PSO-C	Matlab	50	-	0,7	3	10-50	100	U	150	-	-	-	-	MA-DSDV	✓	✓	✓	✓	✓	✓	✓	✓	

U : UnitDisc; TR : *Two-ray ground model*; S : *Shadowing model*; N : Nakamy; CBR : *Constant Bit Rate*, CORS : *CORSIM*, Rand : *Random mobility*, NCT : NCTUns, VM : VanetMobility, BP : Bande passante, PDR : Packet Delivery Ratio, Nb Cluster : Nombre de groupe, CH : Cluster-head, CM : Cluster member

TABLEAU 3.14 – Choix de simulation pour l'évaluation des protocoles de backbone de groupement de la littérature

Ref	Nom	Comm		Réseau - mobilité				Technologie			Application			Performance								
		Simulateur	Durée (s)	Simulateur	Longueur (km)	Nombre de voies	Densité (veh)	Vitesse (km/h)	Propagation	Portée (m)	BP (Mbits/s)	Type	Taille (Bytes)	Fréquence (Hz)	Comparaison avec les autres protocoles	Délat	PDR	Débit	Nb cluster	Durée CH	Durée CM	Changement
[71]	BBR	NS2	150	IMP	2	4	100-600	145	N	-	-	CBR	512	10	Flooding, MPR, Wp-persistence	✓	✓					
[73]	BBBR	NS2	500	IMP	2	4	80	145	N	-	-	30 CBR	512	2	AODV, AODV-L	✓	✓	✓				
[61]	CC	NS2	300	IDM MO-BIL	5	8	30-150	30-180	U	300-700	-	-	-	10	AODV, AODV-L				✓	✓	✓	✓
[72]	BBNC	NS2	500	IMP	2	4	80	-	N	600	-	10 - 30 CBR	512	2,5	AODV, AODV-L CDS-SVB	x	✓	✓				✓
[103]	BAHG	NS2	700	SUMO	3km ²	-	600	15-130	U	300	2	10 CBR	-	0,5-5	GPCR, GyTar	✓	✓	✓				
[60]	DBA-MAC	OMNET - VEINS		SUMO	12km ²	-	60-500	-	U	300	-	Alert	32	0,25	Flooding	✓	✓	✓				
[67]	CDS-SVB	OMNET - VEINS	6000	SUMO	12km ²	-	120-500	90	U	300	-	Alert	32	0,25	DBA-MAC	✓	✓	✓				✓
[121]	RBF	Matlab	100	?	100km	3	150-350	0-80/100/120	U	1000	-	CBR	1000	4	PassCar	✓	✓	✓	✓			✓

U : UnitDisc; TR : *Two-ray ground model*; S : *Shadowing model*; N : Nakamy; CBR : *Constant Bit Rate*, CORS : *CORSIM*, Rand : *Random mobility*, NCT : *NCTUns*, VM : *VanetMobility*, IMP : *IMPORTANT*, BP : *Bande passante*, PDR : *Packet Delivery Ratio*, Nb Cluster : *Nombre de groupe*, CH : *Cluster-head*, CM : *Cluster member*

TABLEAU 3.15 – Choix de simulation pour l'évaluation des protocoles de routage à QoS de la littérature

Ref	Nom	Comm		Réseau - mobilité				Technologie		Application			Performance											
		Simulateur	Durée (s)	Simulateur	Longueur (km)	Nombre de voies	Densité (veh)	Vitesse (km/h)	Propagation	Portée (m)	BP (Mbits/s)	Type	Taille (Bytes)	Fréquence (Hz)	Comparaison avec les autres protocoles	Délai	PDR	Débit	Nb cluster	Durée CH	Durée CM	Changement		
[101]	IRQV	NS2	6000	VM	$7km^2$	-	0,01-0,03veh/m	30-90	U	250	3	CBR	512	1	GSR CAR	✓	✓	✓						
[107]	OHNS	NS2	200	Manh	-	-	1-10	10-50	U	250	2	CBR FTP	512k	-	GPSR	✓	✓	✓						
[104]	RQ-AODV	NS2	300	VM	3	2*4	50	55-160	U	250	-	CBR	512	-	AODV ✓	✓	✓	✓						
[99]	AQVA	NS2	1000	VM	$1km^2$	-	5-50	40-100	U	100	2	5-40 CBR	500	5-40	DYMO, FSR, AODV	✓	✓	✓						
[102]	VOA	NS2	300	-	-	3	25	120	U	250	-	-	-	-	DSR, AODV	✓	✓	✓						
[78]	RAQV	NS2	300	-	$2km^2$	-	250	18-125	TR	-	-	CBR	512	-	AODV	✓	✓	✓						
[122]	GVGRID	Net	600	Net	$1,5km^2$	-	360-1080	30-60	U	200	-	-	-	-	GPSR, GPCR	✓	✓	✓						
[115]	QoS OLSR	Matlab	420	VM	-	-	-	60-120	S	150-300	-	-	-	1000	QOLSR	✓	✓	✓						
[114]	QOLSR	OPNET	600	Rand	$1km^2$	-	50	3-30	U	50-300	2	CBR	?	?	OLSR	✓	✓	✓						
[120]	RPGR	NS2	500	SUMO	$4km^2$	-	100-350	40-70	N	300	3	CBR	512	2	GeoSVR, SDR, CMGR	✓	✓	✓						
[100]	QoS-Bee	NS2	600	VM	$2,4km^2$	-	30	-	-	-	-	15 CBR	-	-	DSDV AODV	✓	✓	✓						
[106]	ARP-OD	NS2	1000	VM	$10km^2$	-	50-250	30-100	U	250	2	CBR	512	0,5-3	GPSR, ROM-SGP AODV	✓	✓	✓						

U : UnitDisc; TR : *Two-ray ground model*; S : *Shadowing model*; N : Nakamya; CBR : *Constant Bit Rate*, CORS : *CORSIM*, Rand : *Random mobility*, NCT : NCTUns, VM : *VanetMobility*, Manh : *Manhattan*, Net : *NetStream*, BP : *Bande passante*, PDR : *Packet Delivery Ratio*, Nb Cluster : *Nombre de groupe*, CH : *Cluster-head*, CM : *Cluster member*

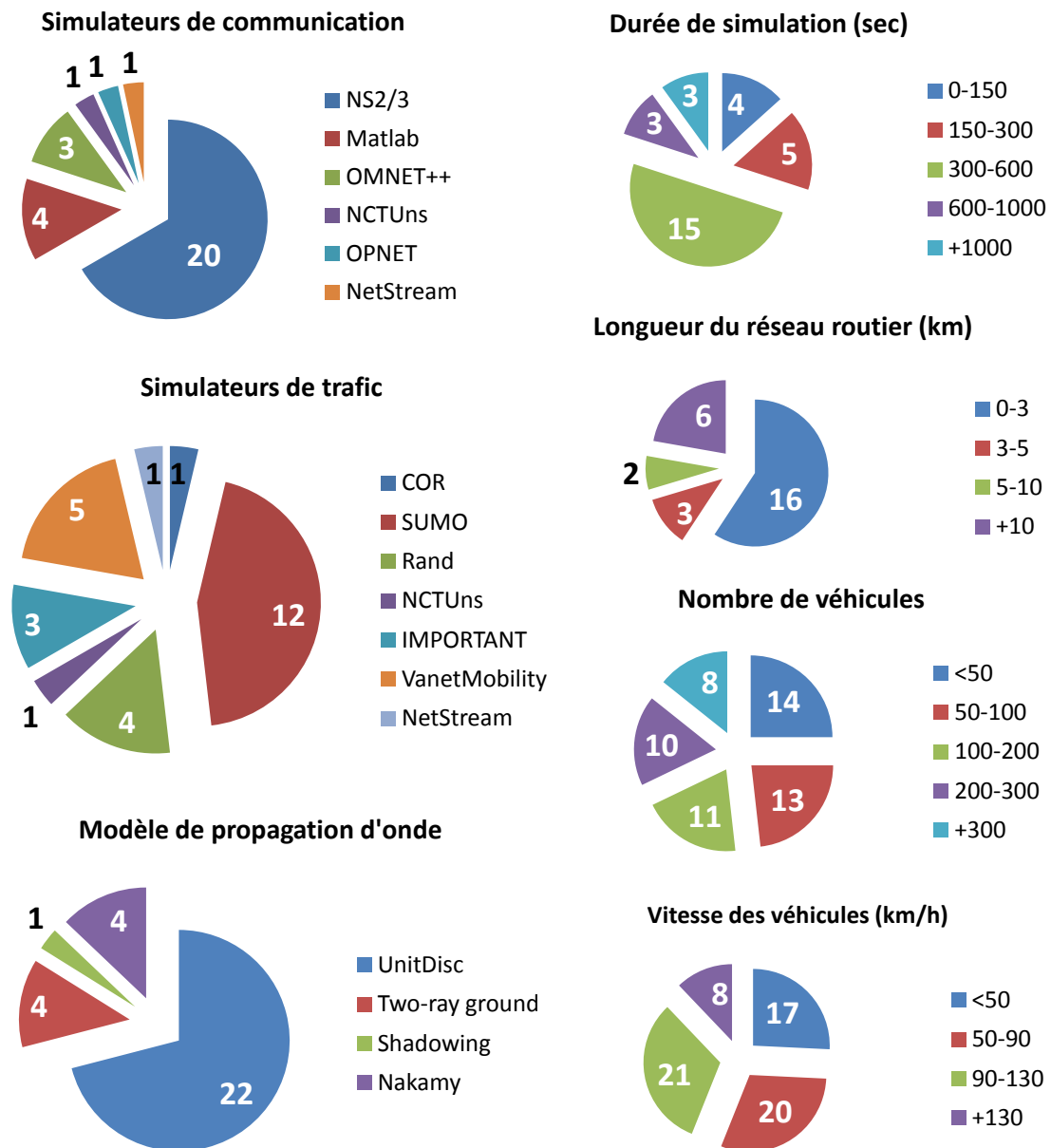


FIGURE 3.16 – Analyse statistique des paramètres de simulation des articles référencés au chapitre 1

Chapitre 4

Analyse structurelle

Sommaire

4.1	Analyses de performances de CBL et des MPRs sous Matlab	135
4.1.1	Évaluation de la structure CBL	135
4.1.2	Évaluation de la méthode des MPRs relativement à CBL version 2	147
4.2	Analyses de performances de CBL-OLSR et OLSR sous OPNET	152
4.2.1	Analyse de l'intégration du schéma CBL sous OPNET	153
4.2.2	Analyse du trafic de routage sous Riverbed OPNET Modeler	154
4.3	Étude de sensibilité de la structure CBL	160
4.3.1	Sensibilité aux coefficients pondérateurs	160
4.3.2	Sensibilité aux scénarios de mobilité	160
4.3.3	Sensibilité aux modèles de propagation	164
4.4	Conclusion	164

4.1 Analyses de performances de CBL et de la méthode des MPRs sous Matlab

Les simulations réalisées avec le logiciel MATLAB ont pour objectif une évaluation structurelle de CBL et de la méthode des MPRs. Leur durée a été fixée à 500 secondes. Cette durée est celle usuellement adoptée dans les études de la littérature. Seul le trafic de messages HELLO est modélisé. Les nœuds VANET ont été paramétrés pour envoyer un message HELLO toutes les secondes. Le seuil V_{time} (section 2.2.8) est réglé à 3 secondes. Enfin, le modèle de propagation en espace libre est utilisé et la portée de transmission maximale a été fixée à 500 mètres.

4.1.1 Évaluation de la structure CBL

L'objet de l'évaluation présentée dans cette section est la structure réalisée par CBL [8] dans une constellation de véhicules. Elle est effectuée en trois temps à l'aide des métriques d'évaluation des performances définies en section 4.1.1.1 :

- Dans un premier temps, section 4.1.1.2, l'évaluation des performances de CBL version 1 est réalisée sur le scénario de mobilité S5 ;
- Dans un second temps, section 4.1.1.3, cette étude est étendue à l'ensemble des scénarios de mobilité S1 à S9 présentés en section 3.4.2.3 ;
- Enfin, section 4.1.1.4, l'évaluation de CBL version 2 est réalisée comparativement à la version 1 sur ces même scénarios.

4.1.1.1 Métriques de performances

Neuf métriques de performances ont été définies pour évaluer la structure formée par le schéma de *clustering* CBL. Ces "métriques structurelles" (Ms) sont les suivantes :

- Ms1 : Feuille/Réseau, le pourcentage de nœuds feuille dans le réseau ;
- Ms2 : Nb_Chaîne, le nombre de chaînes dans le réseau ;
- Ms3 : Branche/Chaîne, le nombre de nœuds branche par chaîne ;
- Ms4 : Voisins/Direction, le nombre de voisins à un saut par sens de circulation du trafic routier ;
- Ms5 : Feuille/Branche, le nombre de nœuds feuille rattachés à un nœud branche ;
- Ms6 : Durée_Branche, la durée pendant laquelle un nœud reste dans l'état de nœud branche ;
- Ms7 : Durée_rattachement_Branche, la durée pendant laquelle un nœud feuille reste attaché au même nœud branche ;
- Ms8 : Nb_relais/nœud, le nombre de nœuds relais choisis par un nœud ;
- Ms9 : Nb_relais/Broadcast, le nombre de nœuds relais impliqués dans une transmission en mode *broadcast*.

4.1.1.2 CBL version 1 – scénario de mobilité S5

Nous présentons dans cette partie l'évaluation de la structure créée par la première version de CBL dans le cas du scénario routier S5 qui est celui le plus fréquemment rencontré sur une section courante d'autoroute (sans entrée-sortie). Ce scénario se compose d'une section de 5 km d'autoroute à double sens et de trois voies par sens avec une densité moyenne de véhicules définie à 2000 VL/h/direction et 400 PL/h/direction. Cette évaluation de CBL version 1 (dont les algorithmes sont en annexe A page 217) a été publiée dans [8].

L'évolution du nombre de nœuds dans l'espace de simulation est représentée en figure 4.1. La zone du réseau routier simulée, vide de véhicules à l'initialisation du système, se charge peu à peu, au fur et à mesure de l'arrivée des véhicules dans l'espace de simulation. Cette montée en charge dure environ 150 s (fin de période signalée sur la figure par une ligne pointillée verticale). Le réseau se stabilise alors avec en moyenne environ 200 nœuds sur le réseau à chaque temps qui se répartissent en 150 nœuds feuille (75% des nœuds du réseau) et 50 nœuds branche (25% des nœuds). Rappelons que les nœuds branche sont les seuls autorisés à retransmettre les messages à travers le réseau de communication. Ainsi, moins ils sont nombreux, moins les ressources de communication sont sollicitées lors des retransmissions. Avec une moyenne d'environ 50 nœuds branche pour le scénario S5, un message qui ne devrait être retransmis que dans un seul des sens de circulation (soit sur trois des six voies) le serait donc par seulement 12,5% des nœuds du réseau.

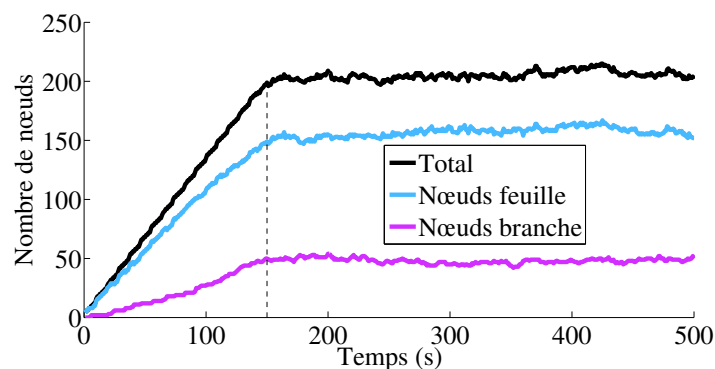


FIGURE 4.1 – Nombre des nœuds feuille et branche en fonction du temps – Scénario S5, CBL Version 1

Nous allons maintenant nous intéresser à la répartition des nœuds branche dans des chaînes. Par exemple, la figure 4.2 montre l'état de la structure formée par CBL à la date $T=500$ s de fin de simulation du scénario S5. À cette date, deux chaînes distinctes ont été créées, une dans chaque sens de circulation. Ce résultat est conforme à ce que nous attendons du schéma de *clustering* de CBL.

Cependant, la figure 4.3a révèle que le nombre de chaînes existantes à tout instant de la simulation du scénario S5 n'est pas toujours de deux chaînes (une chaîne par sens de circulation). Le nombre de chaîne augmente sporadiquement de deux chaînes à trois, quatre, voire cinq sur l'intervalle de temps 100-400 s. Ces augmentations ponctuelles ont deux causes possibles. Une première est l'occurrence d'une rupture

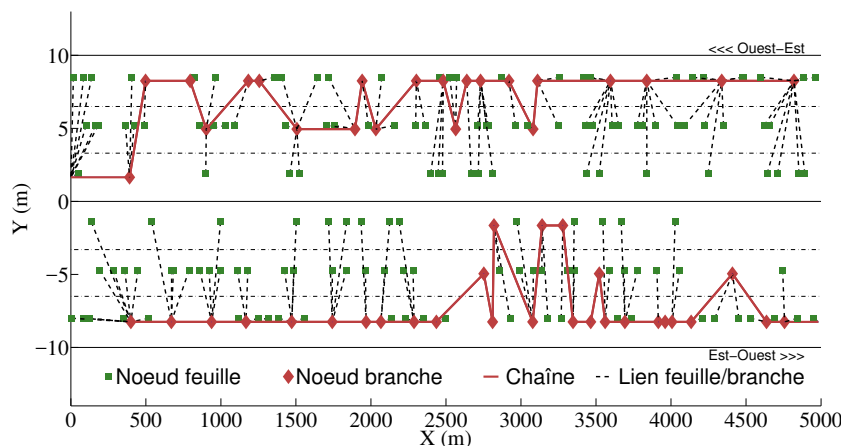


FIGURE 4.2 – État de la structure formée par CBL au temps $t=500$ s – Scénario S5, CBL Version 1 sur ces même scénarios.

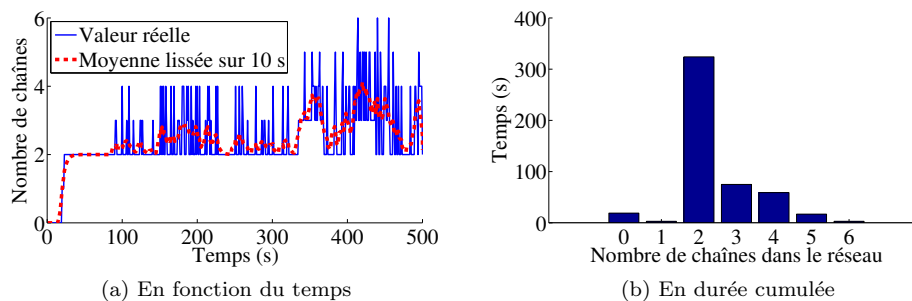


FIGURE 4.3 – Nombre de chaînes dans le réseau (métrique Ms2) – Scénario S5, CBL Version 1

de chaîne suite au dépassement d'un nœud branche par un de ses nœuds branche voisin de la chaîne. La seconde est la sortie d'un nœud branche de la chaîne de la portée de communication de son nœud branche amont ou aval. Ce second cas de rupture est illustré en figure 4.4. À $T=124$ s, la chaîne de la direction Est-Ouest est rompue à l'abscisse 2200 m lorsque deux nœuds branche voisins deviennent trop éloignés. Une seconde après, à $T=125$ s, l'élection d'un nouveau nœud branche intermédiaire permet la reconstruction de la chaîne. Le prompt retour à deux chaînes suite à une rupture de chaîne témoigne d'une rapide reconstruction des chaînes par CBL.

Cependant, au delà de 400 s, entre environ 400 et 450 s, la figure 4.3a montre une formation stable de trois chaînes au lieu de deux, avec des phénomènes de ruptures précédemment expliqués qui font varier ce nombre de quatre, cinq à rarement six. Puis de nouveau entre environ 450 et 500 s, le nombre de chaîne revient à deux. Conformément à nos explications en sous-section 2.4.2.2 et en figure 2.12, cet épisode est dû à la création d'une chaîne parallèle lorsque, suite à la rupture de la liaison entre deux nœuds branche, ces derniers sélectionnent dans un même laps de temps un nœud feuille différent comme maillon de la chaîne. Le retour à deux chaînes correspond à la sortie de la section des nœuds branche impliqués dans la création de cette chaîne supplémentaire. Il résulte de ces ruptures de chaînes ponctuelles et de la création d'une chaîne parallèle un nombre de chaînes supérieur à deux chaînes pendant 30% du temps de la simulation (figure 4.3b). Ainsi, malgré ce défaut de la version 1 de CBL, une chaîne par sens de circulation a bien été formée pendant 70% de temps.

Selon la figure 4.5, le nombre maximal de nœuds branche par chaîne fluctue autour de 25 nœuds. Le réseau routier faisant une longueur totale de 5 km, chaque nœud branche d'une chaîne est donc éloigné de 200 m en moyenne de chacun de ses deux nœuds branche voisins. Néanmoins, il arrive que le nombre de nœuds branche par chaîne chute fortement, aux instants où au moins l'une des deux chaînes du réseau est rompue. Trois chaînes ou plus coexistent ainsi durant un faible laps de temps et leur nombre de nœuds branche devient inférieur à la situation optimale de deux chaînes comportant chacune environ 25 nœuds

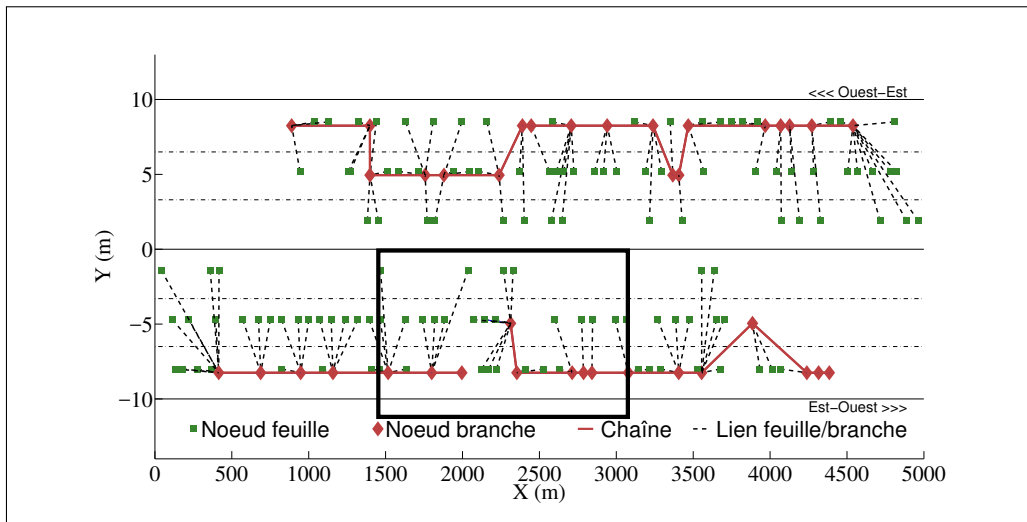
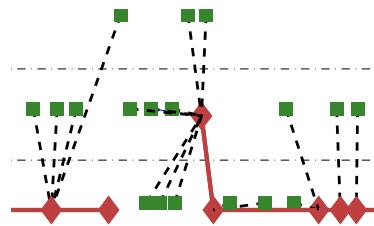
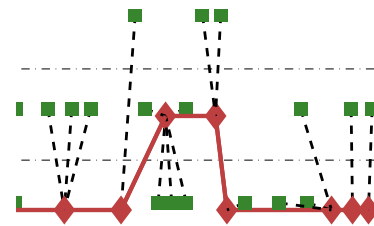
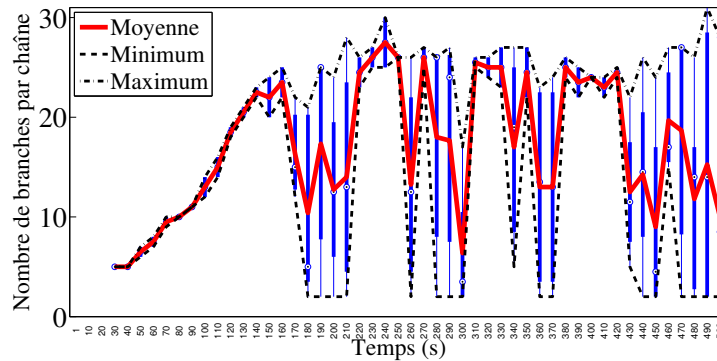
(a) Chaînes au temps $t=124$ s(b) Zoom à $t=124$ s de la chaîne Est-Ouest aux abscisses $X \in [1500, 3000]$ (c) Zoom à $t=125$ s de la chaîne Est-Ouest aux abscisses $X \in [1500, 3000]$

FIGURE 4.4 – État de la structure formée par CBL – Scénario S5, CBL Version 1

branche.

FIGURE 4.5 – Nombre de nœuds branche par chaîne (métrique Ms_3) – Scénario S5, CBL Version 1

La figure 4.6 trace le nombre de voisins à un saut de chaque nœud du réseau. Elle montre qu'une moyenne de 40 nœuds voisins est atteinte à l'issue de la durée de 150 s de montée en charge du réseau (voir aussi la figure 4.1).

En effet, après l'accroissement rapide du nombre de nœuds voisins en début de simulation, la courbe subit un premier ralentissement autour de 15 à 20 nœuds entre $T=20$ s et $T=70$ s. Celui-ci correspond au temps nécessaire aux nouveaux entrants pour réaliser une découverte complète de leur voisinage, via leurs messages HELLO, et atteindre la vingtaine de voisins. Après 70 s environ de simulation, les premiers nœuds entrants du réseau se situent à mi-parcours de la section autoroutière et se rapprochent des premiers nœuds arrivant en sens opposé. Le nombre moyen de voisins à un saut de chaque nœud du réseau grimpe alors, entre les dates approchées de 70 s et 150 s, d'une vingtaine à une quarantaine

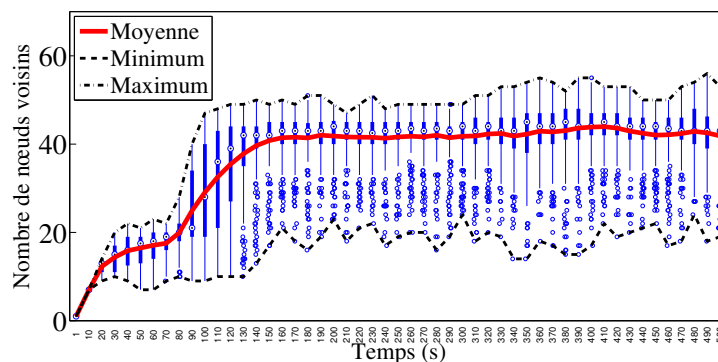


FIGURE 4.6 – Nombre total de nœuds voisins à un saut (deux fois la métrique Ms4, Ms4 : Voisins/Direction) – Scénario S5, CBL Version 1

de nœuds. Ainsi, lorsque le réseau est complètement stabilisé, après environ 150 secondes de simulation, chaque nœud du réseau compte 40 nœuds voisins à un saut, soit une vingtaine de nœuds par direction de circulation (métrique Ms4). L’observation des premiers et troisièmes quantiles représentés toutes les dix secondes sur la figure 4.6 met en relief des valeurs proches de la moyenne, ce qui met en évidence l’uniformité du trafic. Ceci est conforté par les valeurs extrêmes mesurées. Celles minimales de 20 nœuds voisins correspondent aux nœuds n’ayant pas terminé la découverte de leur voisinage. Quant à celles maximales de 45 nœuds, elles restent relativement proches de la valeur moyenne.

La courbe du nombre de nœuds feuille par nœud branche, figure 4.7, montre, qu’après un pic initial et une fois le réseau stabilisé, un *cluster* compte en moyenne cinq nœuds feuille et qu’ainsi 25% des voisins dans un même sens de circulation choisissent le même nœud branche.

Certains nœuds branche sont choisis par 15 nœuds (75% de leurs voisins à 1 saut), d’autres par un seul nœud. Le pic initial, qui intervient environ après 20 secondes de simulation, est dû à l’arrêt brusque de l’augmentation du nombre de feuille rattaché aux deux premiers nœuds branche élus (un dans chaque sens de circulation). Témoin du schéma de création d’une chaîne par CBL, il correspond au délai avant l’initialisation d’une chaîne. Ce délai est causé par l’entrée progressive des nœuds sur le réseau routier. En effet, en début de simulation, chaque nœud (initialisé feuille) s’insérant dans la section du réseau routier simulé réalise une découverte de son voisinage. Un de ces nœuds feuille qui a la vitesse la plus faible est élu nœud branche par l’un de ses voisins. Alors seul nœud branche élu, les autres nœuds feuille à portée de communication se rattachent à lui. À chaque rattachement, la valeur de la métrique Ms5 augmente. Cette situation se perpétue tant que le nœud branche n’élit pas un nœud feuille au titre de nœud branche. Finalement, la chute du pic intervient lorsque le premier nœud branche élu dans un des sens de circulation est en mesure, suite à son avancée sur le réseau routier et l’entrée de nouveaux nœuds feuille plus lointains, de détecter un nœud feuille qui lui apporte un nouveau voisin à deux sauts de lui qu’il élit alors au titre de nœud branche “maillon de leur chaîne”. Ainsi, après cet épisode d’une vingtaine de secondes, une chaîne par sens de circulation a été créée. Les nœuds feuille jusqu’alors rattachés à un seul nœud branche se répartissent en fonction de leur proximité et du lien de communication autour des deux nœuds de la “jeune” chaîne. Et la métrique Ms5 du nombre de nœuds feuille attachés à un même nœud branche, qui jusqu’alors avait augmenté, décroît jusqu’à la moyenne de cinq nœuds feuille par *cluster*.

Selon les résultats de la métrique Ms6 reportés figure 4.8, la durée moyenne d’un nœud dans l’état de nœud branche est de 70 s une fois le réseau stabilisé. Cette valeur est à considérer au regard de la durée moyenne de traversée de la section qui est de 138 s : c’est la moitié de la durée moyenne de traversée de la section. Elle correspond aussi, pour un véhicule roulant à une vitesse faible de 80 km/h, à 1,5 km parcourus, soit à une distance de trois fois la portée radio maximale. Cette distance est significative même sur une autoroute. Certains nœuds restent branche durant toute la traversée de la section d’autoroute comme le révèle les valeurs maximales à plus de 150 s atteintes par la métrique Ms6. Ainsi, si la section du réseau était plus longue, il est probable que la valeur moyenne de Ms6 augmenterait. Le premier quantile atteint la valeur de 25 s lorsque le réseau est stable et le troisième quantile est supérieur à 100 s.

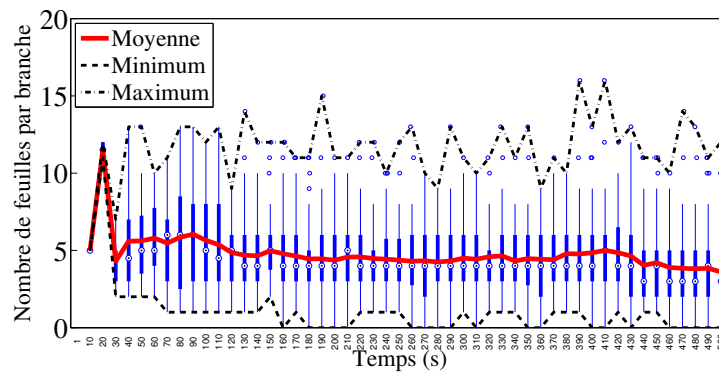


FIGURE 4.7 – Nombre de nœuds feuille attachés à un même nœud branche (Ms5 : Feuille/Branche) – Scénario S5, CBL Version 1

Les valeurs minimales observées, proches de quelques secondes, correspondent aux nœuds branche qui viennent d’être élus au moment où la simulation se termine.

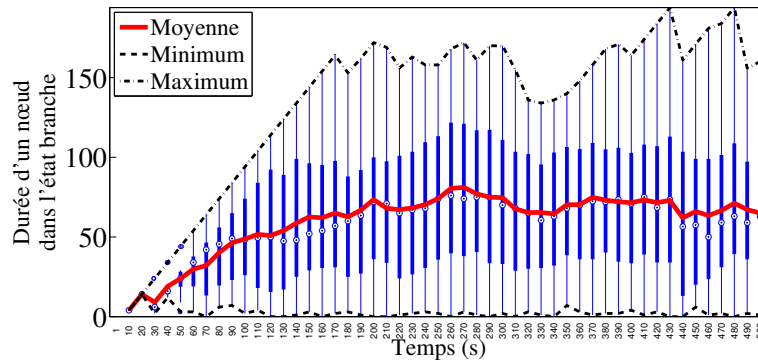


FIGURE 4.8 – Durée pendant laquelle un nœud reste dans l’état de nœud branche (Ms6 : Durée_Branche) – Scénario S5, CBL version 1

La figure 4.9 montre que la durée moyenne de rattachement d’un nœud feuille à un même nœud branche est de 20 s, une fois la période de stabilisation du réseau passée. Pour des applications V2V de sécurité routière dont la périodicité de transmission des messages irait de 50 ms à 500 ms, ce qui correspond au cahier des charges de la plupart des applications envisagées [192], ce temps représente jusqu’à 40 alertes transmises par un nœud feuille et relayées par le même nœud branche sur l’ensemble du réseau. Remarquons que certains nœuds feuille restent attachés à un même nœud branche durant toute leur traversée : les valeurs maximales de la métrique Ms7 dépassent en effet 120 s.

Le nombre de nœuds relais choisis par un nœud (métrique Ms8), figure 4.10, est son nombre de nœuds branche élus, soit un ou deux nœuds selon le type et la position du nœud. Il est imposé par l’algorithme CBL conformément aux définitions spécifiées en section 2.2. Sa moyenne, après stabilisation du réseau, est de 1,2 nœuds.

Le nombre de nœuds relais impliqués dans une transmission en mode *broadcast* (métrique Ms9) est illustré figure 4.11. Cette métrique atteint une valeur moyenne de 47,8 après stabilisation du réseau. Ainsi si un message ne doit être envoyé que dans un sens de circulation, il sera relayé par moins de 24 nœuds.

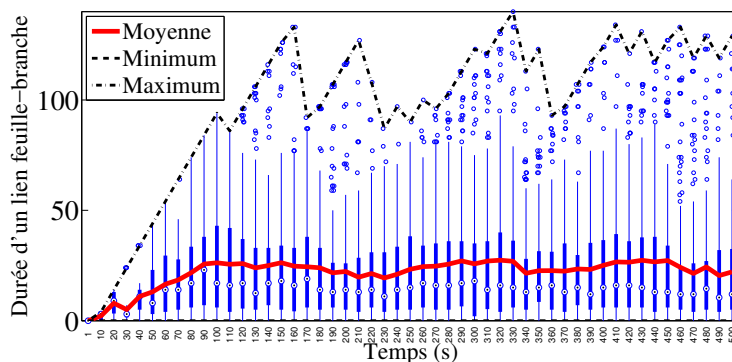


FIGURE 4.9 – Durée pendant laquelle un nœud feuille reste attaché au même nœud branche (Ms7 : Durée_rattachement_Branche) – Scénario S5, CBL Version 1

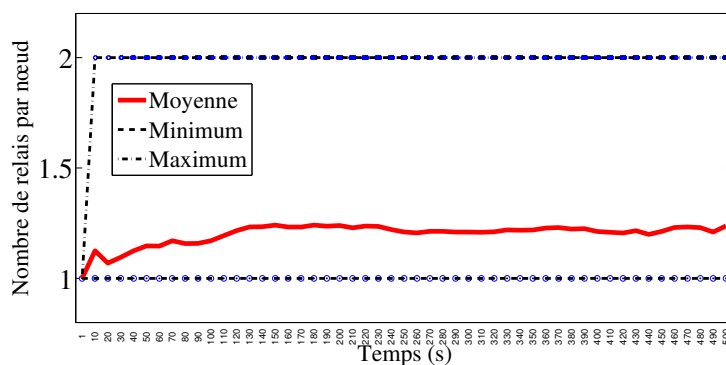


FIGURE 4.10 – Nombre de nœuds relais choisis par un nœud (Ms8 : Nb_relais/nœud) – Scénario S5, CBL Version 1

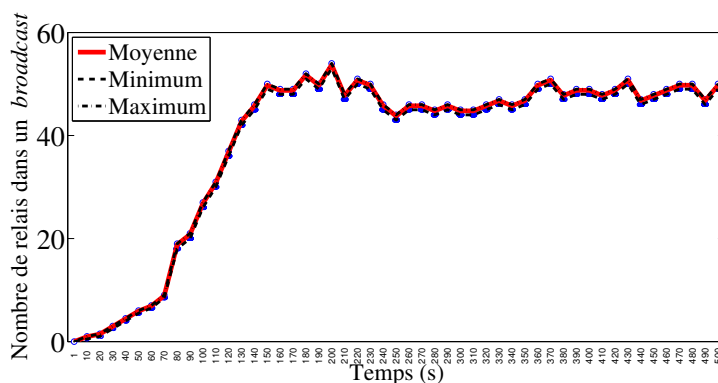


FIGURE 4.11 – Nombre de nœuds relais impliqués dans une transmission en mode *broadcast* (Ms9 : Nb_relais/Broadcast) – Scénario S5, CBL Version 1

Ainsi, l'étude des premiers résultats de simulations du protocole CBL Version 1 sur le scénario S5 montre que CBL forme une structure stable dans le temps, qui se répare rapidement mais qui souffre parfois de la création de chaînes parallèles dans le même sens de circulation.

4.1.1.3 CBL Version 1 – scénario de mobilité S1 à S9

Nous étendons maintenant notre évaluation de la structure CBL version 1 aux neuf scénarios de mobilités S1 à S9 afin de mesurer l'impact du réseau routier et de la densité de véhicules sur cette structure. Le tableau 4.1 rassemble les valeurs moyennes des métriques Ms1 à Ms9 de chacun des neuf scénarios. Ces valeurs moyennes ont été calculées une fois le réseau stabilisé, soit à compter de la date T=150 s de simulation des scénarios et jusqu'à la date T=500 s de fin des simulations.

TABLEAU 4.1 – Valeurs moyennes des métriques de performances Ms1 à Ms9 calculées dans l'intervalle de temps de 150 à 500 s – Scénarios S1 à S9, CBL Version 1, simulation sous Matlab

Réseaux	R1			R2			R3			
	1	2	3	4	5	6	7	8	9	
Scénarios										
Densité	-	+	++	-	+	++	-	+	++	
Nombre de nœuds	25	102	198	50	205	369	46	189	390	
Métriques	Ms1 : Feuille/Réseau (%)	28	76	85	25	77	84	46	67	76
	Ms2 : NB_Chains	1,65	1,96	2,04	3,29	2,68	3,63	3,36	6,36	9,29
	Ms3 : Branche/Chaîne	13,7	15,9	18,9	12,1	20,4	17,8	9,4	10,4	11,3
	Ms4 : Voisins/Direction	3,9	18,8	38,6	4,8	21,2	38,2	4,1	17,7	37,8
	Ms5 : Feuille/Branche	1,94	4,75	7,03	1,94	4,89	6,84	2,31	3,43	4,42
	Ms6 : Durée_Branche (s)	64,4	71,9	70,0	65,1	73	73,5	59,4	53,6	57,7
	Ms7 : Durée_rattachement_Branche (s)	18,8	20,6	21,3	15,6	22,5	22,6	8,6	23,8	16,5
	Ms8 : Nb_relais/nœud	1,60	1,20	1,13	1,62	1,20	1,14	1,41	1,26	1,20
	Ms9 : Nb_relais/Broadcast	17,6	24,1	29,5	36,9	47,8	56,6	25,3	61,5	94,1

Définition des métriques en section 4.1.1.1.

La métrique Ms1 donne le pourcentage de nœuds feuille dans le réseau. Plus celui-ci est élevé, moins le réseau compte de nœuds branche et, par conséquent, moins les ressources du système de communication seront sollicitées pour la retransmission de paquets. Le tableau 4.1 montre, pour chacun des trois réseaux (R1, R2, R3), une augmentation de Ms1 avec celle de la densité en nombre de véhicules du réseau. Ce résultat est confirmé graphiquement par la représentation des nombres moyens de nœuds feuille et branche par scénario en figure 4.12.

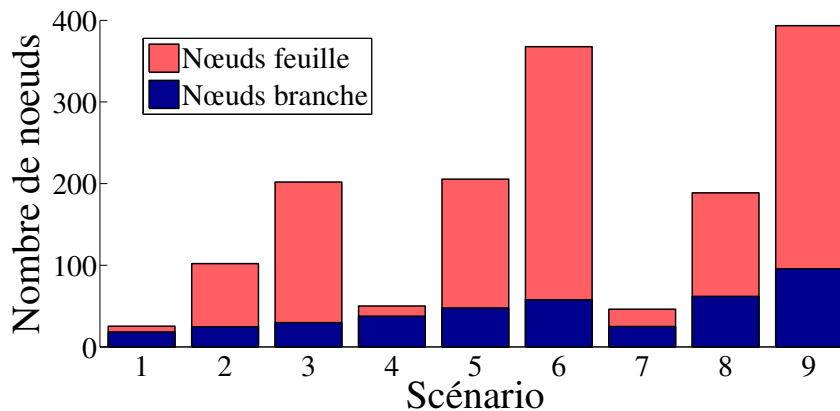


FIGURE 4.12 – Nombre moyen de nœuds feuille et branche dans le réseau lorsque le réseau est stable (150-500s) – Scénarios S1 à S9, CBL Version 1

Le pourcentage des nœuds branche ne dépasse pas 30% du nombre total de nœuds du réseau pour les scénarios de moyenne (S2, S5, S8) et de forte (S3, S6, S9) densités. De moins bons résultats sont obtenus avec les scénarios de faible densité S1, S4, et S7 pour lesquels 54% à 75% des nœuds sont des nœuds branche. Ceci s'explique par les contraintes imposées par la construction de la chaîne. Dès lors qu'un nœud est élu nœud branche, il doit élire un nœud branche en amont de sa position et un autre en aval. Or, lorsque le réseau compte peu de nœuds, dispersés le long réseau routier, l'algorithme CBL

de construction de la chaîne force la majorité des nœuds à devenir nœud branche comme l'illustre la représentation de la structure CBL formée au temps $t=300$ s dans le cas du scénario S1 en figure 4.13. Enfin, au sein d'un même réseau routier, le nombre de nœuds branche varie peu contrairement au nombre de nœuds feuille. En effet, une fois que le nombre de nœuds branche nécessaires à la construction d'une chaîne est atteint, les autres nœuds du réseau restent à l'état feuille.

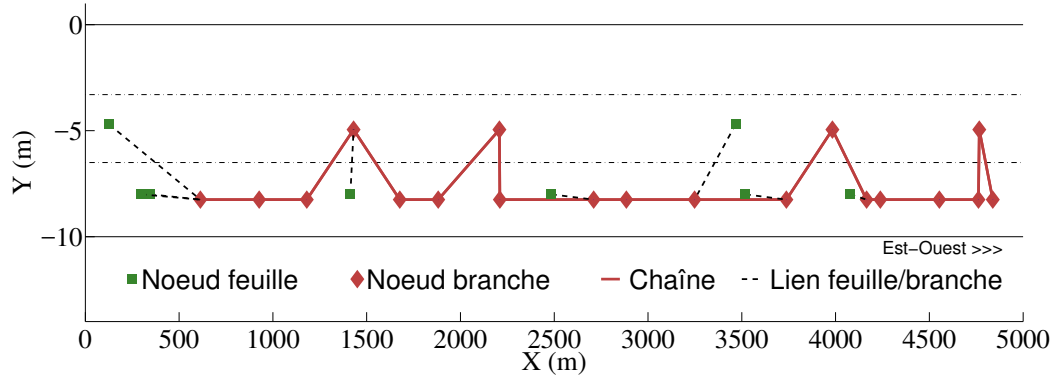


FIGURE 4.13 – État de la structure formée par CBL au temps $T=300$ s – Scénario S1

La métrique Ms_2 , qui informe du nombre de chaînes dans le réseau, varie en fonction de la densité du trafic et de la configuration du réseau routier. Une chaîne est attendue dans le réseau routier R1 à un seul sens de circulation, deux chaînes dans celui à deux sens de circulation R2 et quatre dans le réseau routier R3 à deux sens de circulation avec la présence d'une entrée et d'une sortie :

- dans le réseau R1 : la métrique Ms_2 varie autour d'une moyenne de 1,65 pour le scénario S1 et 2,04 pour le scénario S3 ;
- dans le réseau R2 : Ms_2 varie autour de 2,68 pour le scénario S5 et de 3,3 pour le scénario S6 ;
- dans le réseau R3 : Ms_2 varie autour de 3,36 pour le scénario S7 et de 9,29 pour le scénarios S9.

Hormis pour le scénario S4, nous observons que lorsque la densité de véhicules augmente dans un même réseau routier, la valeur moyenne de Ms_2 augmente également. En effet, l'augmentation de la densité est accompagnée de celle de la probabilité qu'un véhicule du réseau a une vitesse "désirée" plus élevée que celle du véhicule qui le précède. Ainsi, les contraintes de mobilité entre les véhicules se font plus fortes et leur trajectoire est impactée : afin de garder une distance de sécurité entre les véhicules tout en tâchant d'atteindre la vitesse "désirée" paramétrée, ceux-ci tendent à dépasser les véhicules plus lents qui les précèdent. Cependant, les dépassements entraînent des ruptures de chaînes lorsqu'ils concernent des nœuds branche du réseau de communication. Par conséquent, la métrique Ms_2 augmente et sa valeur moyenne dépasse le nombre attendu de chaînes quel que soit le scénario en raison des ruptures de chaînes.

Pour les scénarios de faibles mobilités (S1, S4 et S7), la raison principale des ruptures de chaînes vient d'un nombre insuffisant de nœuds, lesquels sont de plus éparpillés dans le réseau. Dans un même sens de circulation, l'absence de nœuds entre deux nœuds branche séparés d'une distance supérieure à la portée de la communication radio empêche leur chaîne de fusionner. Ainsi, la valeur moyenne de la métrique Ms_2 est supérieure à un pour le scénario S1. Pour le scénario S4 cette valeur est doublée car le réseau possède deux sens de circulation. Si la valeur moyenne de Ms_2 du scénario S7 est similaire à celle du scénario S4, la distribution en nombre de chaînes est très différente (figure 4.14). Pour le scénario S7, cette distribution est étendue sur un plus grand nombre de chaînes (de une à huit chaînes contre deux à cinq dans le scénario S4). L'objectif de quatre chaînes pour le réseau R3 est observé le plus fréquemment. Néanmoins, à certains instants le réseau compte jusqu'à huit chaînes. Les ruptures de chaînes dans le scénario S7 sont dues aussi bien au nombre faible de nœuds dans le réseau qu'à la présence d'une entrée et d'une sortie qui accentuent encore la dispersion des nœuds. Les résultats de Ms_2 pour les scénarios S8 et S9 de moyenne et forte densité confortent cette hypothèse d'une augmentation du nombre de ruptures de chaîne dans le réseau R3 occasionnée par la présence d'une entrée et d'une sortie, et expliquent également la formation de chaînes parallèles.

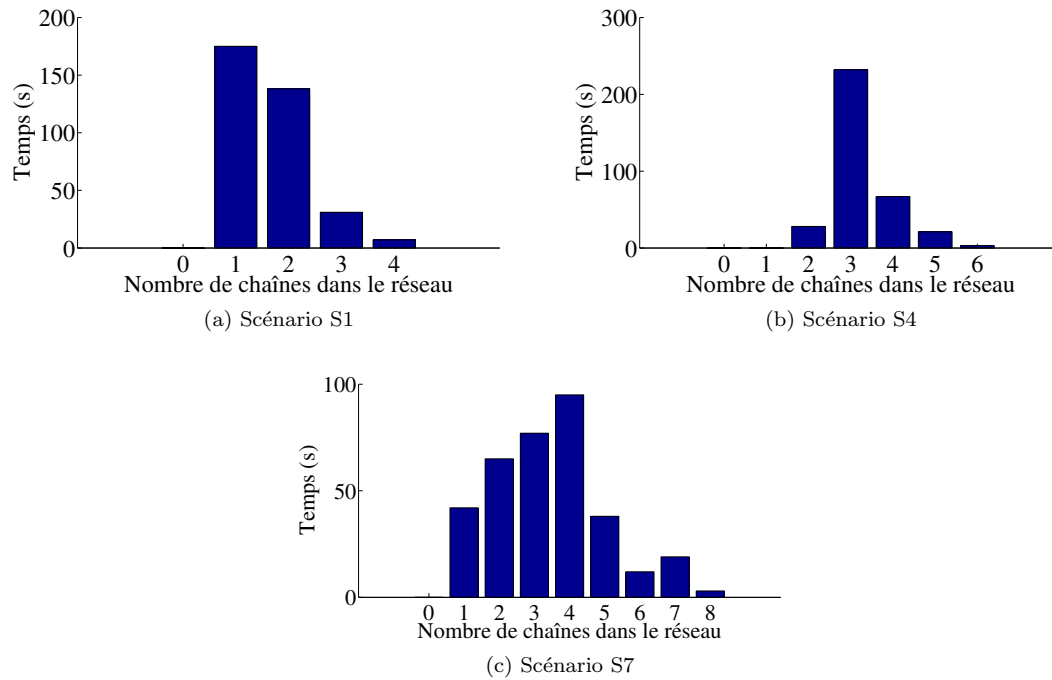


FIGURE 4.14 – Histogramme de la durée cumulée du nombre de chaînes (Ms2) entre les dates 150 et 500 s – Scénario S1, S4 et S7 CBL Version 1

Le nombre moyen de nœuds branche par chaîne (métrique Ms3) garde le même ordre de grandeur quelque soit la densité des nœuds pour un même réseau routier. Il est légèrement plus élevé pour les scénarios de densités moyennes et fortes de véhicules. Cette métrique Ms3 est corrélée avec Ms1 et Ms2.

Les résultats du nombre de voisins à un saut par direction (métrique Ms4) sont corrélés avec la densité de nœuds par direction. Ainsi, pour une densité faible par direction (scénarios S1, S4, et S7), la moyenne de la métrique Ms4 varie entre 3,9 et 4,8. Pour une densité moyenne (scénarios S2, S5, et S8), elle varie entre 17,7 et 21,2. Enfin, pour une densité élevée (scénarios S3, S6, et S9), elle varie entre 37,8 et 38,2.

Le nombre de feuilles par branche (métrique Ms5) est également corrélé avec le nombre de nœuds par direction, car chaque feuille se rattache à un nœud branche circulant dans la même direction. Ainsi, en moyenne deux nœuds feuille se rattachent à un même nœud branche pour un scénario de faible densité ; pour un scénario de moyenne densité, de trois à quatre nœuds feuille ; pour un scénario de forte densité, de quatre à sept feuilles.

La durée d'existence d'un nœud en tant que nœud branche (métrique Ms6) varie entre 64,4 s et 73,5 s pour les scénarios des réseaux routiers R1 et R2 sans entrée-sortie. Cette durée est plus faible, de 53,6 s à 59,4 s, pour ceux du réseau R3 avec entrée-sortie. Avec R3, les permutations d'états des nœuds sont plus fréquentes : un nœud branche dans la section d'entrée passe à l'état de feuille lorsqu'il vient s'insérer dans la section courante de l'autoroute ; inversement, un nœud feuille engagé dans la voie de sortie de la section courante pourra être élu branche par son voisinage. Ces permutations en cours de trajet induisent une diminution de la valeur de la métrique Ms6.

La durée de rattachement d'un nœud feuille à un nœud branche (métrique Ms7) est corrélée à la métrique Ms6 pour les réseaux R1 et R2. Pour les scénarios où la durée moyenne d'existence d'un nœud à l'état de nœud branche est inférieure (S1 et S4), la valeur de la métrique Ms7 est légèrement inférieure. Elle dépend aussi du nombre de nœuds branche (figure 4.12) : pour chaque nœud feuille, la possibilité d'élire un meilleur nœud relais augmente avec le nombre de nœuds branche dans le réseau, ce qui entraîne naturellement une diminution de la durée de rattachement d'une feuille à un même nœud branche. C'est pourquoi la métrique Ms7 est plus faible pour le scénario S9 que pour S8. Quant au scénario S7, il souffre d'une trop faible densité de nœuds.

Le nombre de relais par nœuds (métrique Ms8) dépend de la métrique Ms1 et Ms2. Un nœud feuille n'a qu'un seul nœud relais, son nœud branche. Un nœud branche a un à deux nœuds relais selon sa

position : en bout (un seul relais) ou milieu de sa chaîne (un relais amont, un relais aval, soit deux relais élus par un nœud branche).

Le nombre de nœuds relais intervenant dans une transmission en mode *broadcast* (métrique Ms9) dépend de la métrique Ms1 et du nombre de nœuds branche dans le réseau. Ce nombre de relais est égal, pour un nœud branche, au nombre de nœuds branche du réseau hormis lui-même. Il est égal, pour un nœud feuille, au nombre de nœuds branche du réseau.

Ainsi, le schéma CBL reste stable lors de l'augmentation de la densité des nœuds mobiles du réseau de communication : il résiste donc bien au passage à l'échelle.

4.1.1.4 CBL version 2 – scénarios de mobilité S1-S9

Intéressons-nous maintenant à la seconde version de CBL que nous avons décrite en section 2.3. Les différents paramètres utilisés lors de cette simulation (*Vtime*, fréquence des messages HELLO, modèle de propagation...) sont identiques à ceux utilisés précédemment. Les coefficients pondérateurs sont paramétrés selon leur définition par défaut (section 2.3.9).

L'objectif de cette évaluation est d'analyser les métriques structurelles de la seconde version de CBL présentée dans ce manuscrit. Cette version a fait l'objet d'améliorations notables concernant les métriques et les procédures d'élection d'un premier nœud branche, d'un maillon de chaîne mais également de la gestion des chaînes pour éviter les chaînes parallèles (section 2.4). Les résultats de cette version sont présentés dans le tableau 4.2. Ce tableau regroupe les valeurs moyennes des métriques lorsque le réseau est stable (entre 150 s et 500 s) pour tous les scénarios et la comparaison avec les résultats de CBL version 1.

TABLEAU 4.2 – Valeurs moyennes des métriques de performances Ms1 à Ms9 calculées dans l'intervalle de temps de 150 à 500 s – Scénarios S1 à S9, CBL version 2, comparaison CBL version 1, simulation Matlab

	Réseaux			R1			R2			R3		
	Scénarios	1	2	3	4	5	6	7	8	9		
Densité	-	+	++	-	+	++	-	+	++			
Nombre de nœuds	25	102	198	50	205	369	46	189	390			
Ms1 : Feuille/Réseau (%)	38,2	81,5	90,1	33,7	79,0	87,2	37,1	74,0	84,5			
<i>Diff</i>	+10,2	+5,5	+5,1	+8,7	+2	+3,2	-8,9	+7	+8,5			
Ms2 : NB_Chains	1,68	1,33	1,34	3,46	2,85	2,90	4,88	4,90	5,30			
<i>Diff</i>	+0,03	-0,63	-0,7	+0,17	+0,17	-0,73	+1,52	-1,46	-3,99			
Ms3 : Branche/Chaîne	9,4	14,1	14,4	9,6	14,8	15,8	5,7	9,9	11,1			
<i>Diff</i>	-4,3	-1,8	-4,5	-2,5	-5,6	-2	-3,7	-0,5	-0,2			
Ms4 : Voisins/Direction	3,9	18,8	37,8	4,6	20,4	37,1	3,9	17,1	36,5			
<i>Diff</i>	+0	+0	-0,8	-0,2	-0,8	-1,1	-0,2	-0,6	-1,3			
Ms5 : Feuille/Branche	2,38	6,17	10,84	2,29	5,51	8,48	2,12	4,55	7,13			
<i>Diff</i>	+0,44	+1,42	+3,81	+0,35	+0,62	+1,64	-0,19	+1,12	+2,71			
Ms6 : Durée_Branche (s)	56,3	54,0	55,3	57,6	53,4	53,7	52,3	44,4	40,5			
<i>Diff</i>	-8,1	-17,9	-14,7	-7,5	-19,6	-19,8	-7,1	-9,2	-17,2			
Ms7 : Durée_rattach_Branche (s)	29,4	39,4	42,9	30,0	36,3	40,2	23,6	32,0	31,5			
<i>Diff</i>	+10,6	+18,8	+21,6	+14,4	+13,8	+17,6	+15	+8,2	+15			
Ms8 : Nb_relais/nœud	1,50	1,16	1,09	1,52	1,18	1,11	1,42	1,21	1,13			
<i>Diff</i>	-0,1	-0,04	-0,04	-0,1	-0,02	-0,03	+0,01	-0,05	-0,07			
Ms9 : Nb_relais/Broadcast	15,0	18,7	19,4	32,7	42,9	47,2	28,4	48,8	59,9			
<i>Diff</i>	-2,6	-5,4	-10,1	-4,2	-4,9	-9,4	+3,1	-12,7	-34,2			

Où *Diff* est la différence "valeurs de CBL version 2 moins celles de CBL version 1".

Définition des métriques structurelles en section 4.1.1.1.

Avec la version 2 de CBL, le pourcentage de nœuds feuille dans le réseau (métrique Ms1) augmente pour tous les scénarios, à l'exception du scénario S7 qui subit une diminution de 8,9%. Cette augmentation

est différente selon la densité du réseau. Pour les scénarios de plus faible densité (scénarios S1 et S4), elle est de l'ordre de 10%. Pour ceux de densité plus élevée, les augmentations varient entre 2% et 8,5%. Certes, ce pourcentage est faible, mais une fois rapporté au nombre de nœuds dans le réseau cela concerne un nombre de nœuds non négligeable qui conservent alors un état de feuille au lieu de se transformer en branche. Par exemple, comparativement à CBL version 1, quatre nœuds supplémentaires restent à l'état de nœud feuille pour le scénario S5 où la valeur moyenne de Ms1 passe de 77% à 79% (2% de plus seulement).

Le nombre de chaînes (métrique Ms2) présente des résultats similaires pour les deux versions dans le cas des scénarios de faible densité S1 et S4. Par contre, le scénario S7 atteint l'objectif de quatre chaînes avec une valeur moyenne de 4,88 (qui inclut les ruptures de chaîne), ce qui n'était pas le cas de la version 1 de CBL. Cette augmentation du nombre de chaînes entraîne une augmentation du nombre de nœuds branche dans le réseau et donc une diminution du pourcentage de nœuds feuille, d'où la diminution de 8,9% de Ms1 pour ce scénario S7. La métrique Ms2 diminue pour les scénarios S2, S3, S6, S8 et S9 entre 20% et 34% en raison de la diminution des ruptures de chaînes et la disparition des chaînes parallèles. La figure 4.15 souligne cet écart entre les deux versions pour le scénario S6 du réseau R2 : l'objectif de deux chaînes est plus souvent atteint avec CBL version 2 (48% du temps avec CBL version 2 contre 14% avec CBL version 1) et le nombre maximum de chaîne dans le réseau est moindre puisqu'il passe de huit à six.

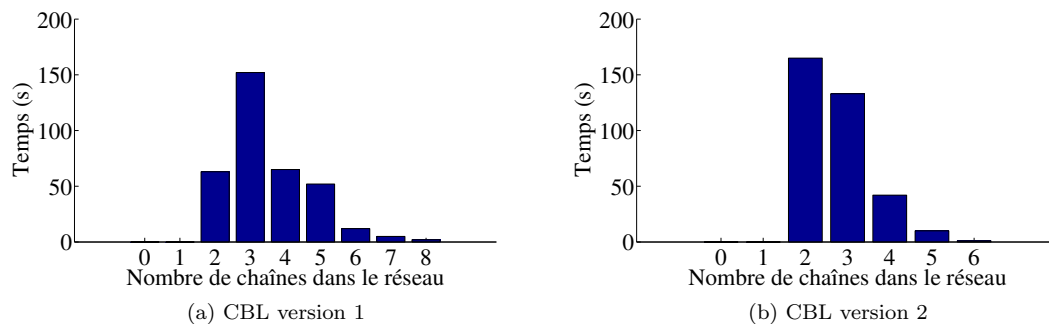


FIGURE 4.15 – Histogramme de la durée cumulée du nombre de chaînes (Ms2) entre $t=150$ s et $t=500$ s – Scénario S6, CBL Version 1 et 2

Pour CBL version 2 et dans le cas des réseaux R1 et R2 sans entrée-sortie, le nombre de chaînes dans le réseau reste quasiment constant pour les scénarios de moyenne et de forte densité d'un même réseau (S2, S3, d'une part et S5, S6 d'autre part) : il ne varie plus en fonction de la densité. Par conséquent, CBL version 2 améliore le passage à l'échelle par une meilleure gestion des chaînes.

Les nombres de nœuds branche par chaîne (métrique Ms3), de nœud relais par nœud (Ms8) et aussi de nœuds relais intervenant dans une transmission en mode *broadcast* (Ms9) dépendent des métriques Ms1 et Ms2.

Le nombre de voisins par direction (métrique Ms4) reste identique, car la mobilité des nœuds n'a pas changé.

Le nombre de nœuds feuille par nœuds branche (métrique Ms5) augmente avec la version 2 de CBL, sauf pour le scénario S7 comme nous l'avons précédemment expliqué. L'augmentation de Ms5 est liée à la diminution du nombre de nœuds branche dans le réseau, restreignant ainsi les possibilités de choix des nœuds feuille. Par exemple, Ms5 augmente de 3,81 pour le scénario S3, avec en moyenne 7,03 nœuds feuille par branche pour la version 1 et 10,84 pour la version 2.

La durée moyenne pendant laquelle un nœud reste à l'état de nœud branche (métrique Ms6) diminue, entre 7,1 et 19,8 s de la version 1 à la version 2, en raison des modifications apportées à l'algorithme 5 sur les conditions de permutation de type de branche à feuille, en particulier dans le cas d'une permutation due à un dépassement de véhicule. En effet, dans la version 1 de CBL, l'ordre des nœuds était simplement modifié dans la chaîne sans permutation de type. Par contre, dans la version 2, le nœud qui dépasse l'autre devient feuille dans l'objectif d'éviter des nœuds branche trop proches.

La durée de rattachement d'un nœud feuille au même nœud branche (métrique Ms7) augmente nettement avec la seconde version de CBL. Cette augmentation est supérieure 8,2 s pour tous les scénarios. Elle atteint même 15 s pour le scénario S7. Cette amélioration provient de la nouvelle procédure d'élection des nœuds branche par les nœuds feuille qui tient compte, non plus de la distance entre un nœud branche et un nœud feuille, mais de leur temps de connexion.

En résumé, la version 2 de CBL augmente la proportion du nombre de nœuds feuille relativement au nombre de nœuds branche. Il en résulte une diminution globale du nombre de nœuds branche relais dans une transmission, en particulier pour une transmission en mode *broadcast*. La diminution du nombre de chaînes montre une meilleure gestion de la construction de la structure CBL. Le nombre de nœuds feuille par branche a augmenté, ce qui correspond, d'un point de vue système, à une augmentation de la taille des *clusters*. Si la durée d'existence d'un nœud à l'état de nœud branche a légèrement diminué, celle de rattachement d'un nœud feuille à un nœud branche a augmenté. Cette amélioration de la durée de connexion entre les nœuds feuille et branche montre une stabilisation des liaisons entre nœuds propice à une amélioration des échanges de messages.

En conclusion, la version 2 de CBL présente de meilleures performances structurelles que la version 1. C'est pourquoi, à partir de la section 4.1.2 de ce chapitre 4, les simulations et évaluations relatives à CBL ne prendront en compte que la version 2 de CBL.

4.1.2 Évaluation de la méthode des MPRs relativement à CBL version 2

Nous présentons ci-après une évaluation structurelle de la méthode des MPRs effectuée en deux temps à l'aide des métriques d'évaluation des performances définies en section 4.1.2.1. Dans un premier temps, section 4.1.2.2, l'évaluation des performances de la méthode des MPRs est réalisée sur le scénario de mobilité S5. Dans un second temps, section 4.1.2.3, cette étude est étendue à l'ensemble des scénarios de mobilité S1 à S9. Nous comparons à chaque fois les résultats avec ceux de CBL version 2 une fois le réseau stabilisé, après 150 s de simulation. Nous indiquerons au fil de l'eau lorsque les métriques de performances de ces deux méthodes sont équivalentes et lorsqu'elles ne le sont pas.

4.1.2.1 Métriques de performances

Cinq métriques de performances tenant compte des spécificités de la méthode des MPRs ont été spécifiées dans le but d'évaluer la structure formée par cette méthode. Ces métriques structurelles sont :

- Ms10 : Non_MPR/Réseau, le pourcentage de nœuds non MPR dans le réseau ;
- Ms11 : Nb_MPR_Selector, le nombre de nœuds qui ont sélectionné un même nœud comme MPR, autrement dit le cardinal du MPR selector set du nœud sélectionné ;
- Ms12 : Nb_MPR_Set, le nombre de nœuds élus en tant que MPR par un nœud, autrement dit le cardinal du MPR set de ce dernier ;
- Ms13 : Durée_MPR, la durée pendant laquelle un nœud reste dans l'état de nœud MPR ;
- Ms14 : Durée_élection_MPR, la durée pendant laquelle un nœud continue à élire un même nœud MPR ;
- Ms15 : MPR/Broadcast, le nombre de nœuds relais MPR impliqués dans une transmission de type *broadcast*.

4.1.2.2 Méthode des MPRs – scénario de mobilité S5

Le nombre de nœuds MPR et non-MPR évolue dans l'espace de simulation selon les courbes de la figure 4.16. Le réseau compte 205 nœuds, 93% sont MPR et 7% non-MPR (métrique Ms10).

Nous ne pouvons pas comparer ces valeurs à celles obtenues avec CBL. Les métriques Ms1, pourcentage de nœuds feuille, et Ms10, pourcentage de nœuds non-MPR, concernent les nœuds des méthodes CBL et MPR qui n'ont pas la capacité de relayer des messages. Pour autant elles ne sont pas équivalentes car la définition des nœuds branche et feuille de CBL est globale à la structure alors que celle des nœuds MPR et non-MPR est locale à chaque nœud. Un nœud est branche pour l'ensemble des nœuds du réseau tandis qu'un nœud est MPR pour les seuls nœuds qui l'ont élu. Par conséquent, lors d'une transmission en mode *broadcast*, tous les nœuds branche de CBL relaient le message ; dans le cas de la méthode des MPRs, ce

message est retransmis par le nœud source, ses MPRs, les MPRs de ses MPRs, et ainsi de suite jusqu'à réception du message par chacun des nœuds du réseau. Néanmoins, concernant le trafic de routage, les nœuds branche et les nœuds MPR sont à l'origine de la génération des messages TC. Ainsi, il est possible de comparer les Ms1 et Ms10 sur ce point.

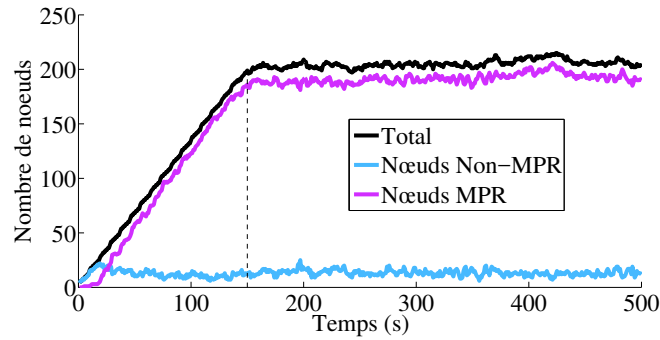


FIGURE 4.16 – Nombre de nœuds MPR et non MPR – Scénario S5, méthode des MPRs

Le nombre de nœuds sélectionnant un même nœud MPR (métrique Ms11) est en moyenne de 11,2 nœuds, figure 4.17. La valeur maximale est de 40 nœuds ce qui correspond à un nœud MPR élu par l'ensemble de son voisinage. La valeur minimale de 1 nœud correspond au cas minimal d'élection d'un nœud comme MPR.

L'aspect global de CBL et local des MPRs biaise la comparaison entre les métriques Ms11 et Ms5 (nombre de nœuds feuille par branche). Cependant les applications automobiles nécessiteront de diffuser des messages à des groupes de nœuds. Ainsi, nous pouvons utiliser ces métriques pour évaluer le nombre de nœuds abonnés à un même nœud relais. De ce point de vue, CBL réduit la taille des groupes de 51%. Les groupes de CBL sont disjoints deux à deux contrairement aux "groupes" de la méthode MPR qui peuvent avoir des intersections importantes.

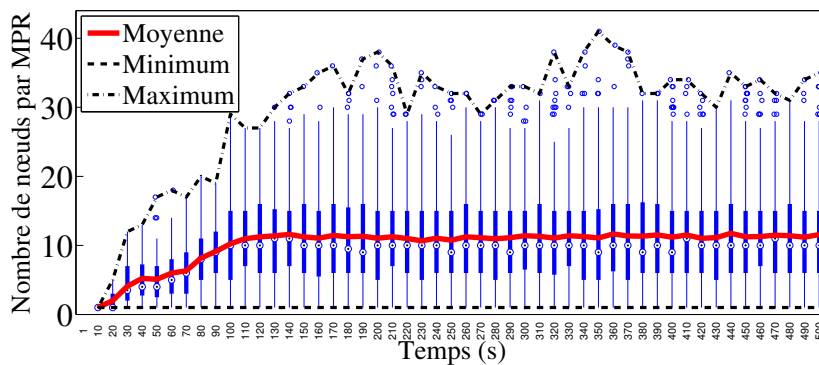


FIGURE 4.17 – Nombre de nœuds sélectionnant un nœud comme MPR (Ms11 : Nb_MPR_Selector) – Scénario S5, méthode des MPRs

Selon la figure 4.18, le nombre de nœuds élus en tant que MPR par un nœud (métrique Ms12) est d'environ 5, les valeurs maximales et minimales sont respectivement de 10 et 1 nœud. Les valeurs des premier et troisième quantile sont très proches, ce qui indique une homogénéité du nombre de nœuds MPR élus par chaque nœud.

Les métriques Ms8 (nombre de relais par nœud) et Ms12 mesurent toutes deux le nombre de nœuds relais impliqués lors de la première transmission du message par un nœud. CBL contraint le nombre de nœuds relais à un ou deux nœuds en fonction du type de nœuds et réduit ainsi cette valeur de 77% par rapport à la méthode des MPRs qui n'impose pas de limite en nombre de nœuds et pour laquelle un message peut couvrir une zone géographique plus étendue lors de cette première transmission.

La figure 4.19 indique une durée moyenne d'un nœud à l'état MPR de 53,8 s (métrique Ms13). Les valeurs maximales de 150 s et plus correspondent au cas de figure où le nœud reste à l'état MPR tout au

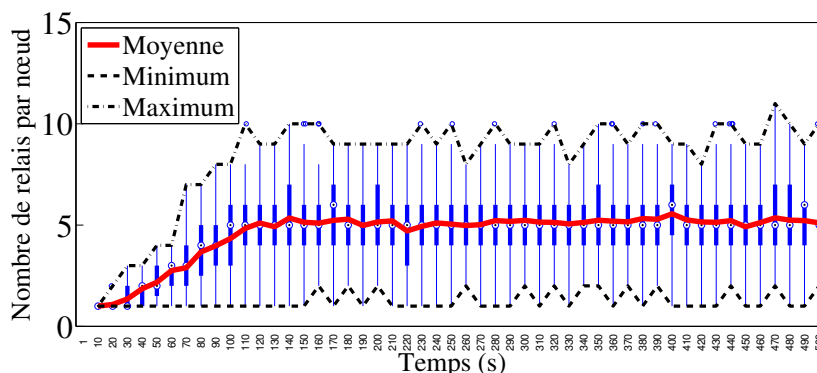


FIGURE 4.18 – Nombre de nœuds dans le MPR MPR Set (Ms12) – Scénario S5, méthode des MPRs

long de son parcours.

La durée moyenne pendant laquelle un nœud sélectionne le même MPR (métrique Ms14) est de 8 s, figure 4.20. Elle est d'environ le double de la valeur médiane (proche de 4 s) en raison des 20% de valeurs singulières supérieures à 13 s.

Les métriques Ms6 (durée d'un nœud à l'état branche) et Ms13 ainsi que les métriques Ms7 (durée de rattachement d'un nœud feuille à un même nœud branche) et Ms14 peuvent être comparées moyennant le biais structurel précédemment évoqué. Les résultats obtenus pour Ms6 et Ms13 sont similaires (53,4 s avec CBL). Par contre, la métrique Ms7 est très supérieure à Ms14 avec 36,5 s en moyenne. La structure CBL fournit donc un rattachement beaucoup plus stable entre un nœud feuille et son nœud relais, comparativement aux résultats obtenus avec la technique des MPRs.

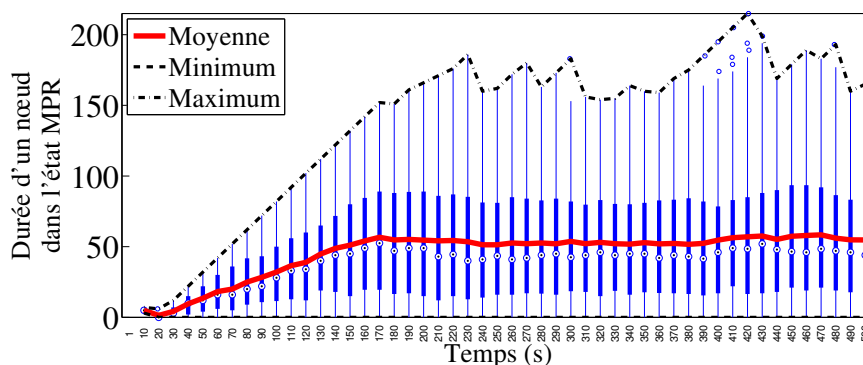


FIGURE 4.19 – Durée pendant laquelle un nœud reste un MPR (Ms13) – Scénario S5, méthode des MPRs

Le nombre de nœuds relais MPR impliqués dans une transmission de type *broadcast* (métrique Ms15) est en moyenne de 176 nœuds soit 85% des nœuds du réseau, figure 4.21. Un peu moins du nombre total de MPR du réseau (93,5% des nœuds) est donc sollicité lors d'une telle transmission.

Seules les métriques Ms9 (nombre de nœud relais impliqués dans une transmission en mode *broadcast*) et Ms15 sont comparables sans biais. Avec près de 43 nœuds participants à la transmission d'un message de type *broadcast*, CBL minimise fortement le nombre de retransmissions, libérant ainsi davantage de bande passante pour les applications.

4.1.2.3 Méthode des MPRs – scénarios de mobilité S1-S9

Les résultats de la méthode des MPRs pour tous les scénarios sont rassemblés dans le tableau 4.3 avec une comparaison de ces différentes métriques entre la méthode des MPRs et CBL version 2.

La figure 4.22 qui représente le nombre de nœuds MPR et non-MPR pour chacun des neuf scénarios confirme les résultats précédemment présentés pour le scénario S5 : la proportion des nœuds non-MPR n'excède jamais 23%.

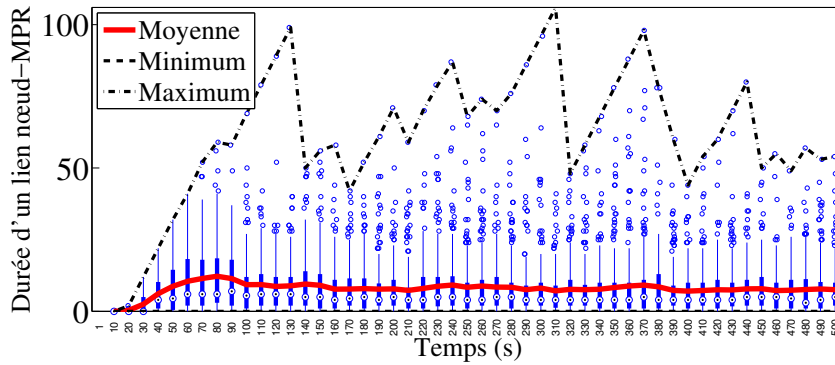


FIGURE 4.20 – Durée pendant laquelle un nœud a sélectionné le même MPR (Ms14) – Scénario S5, méthode des MPRs

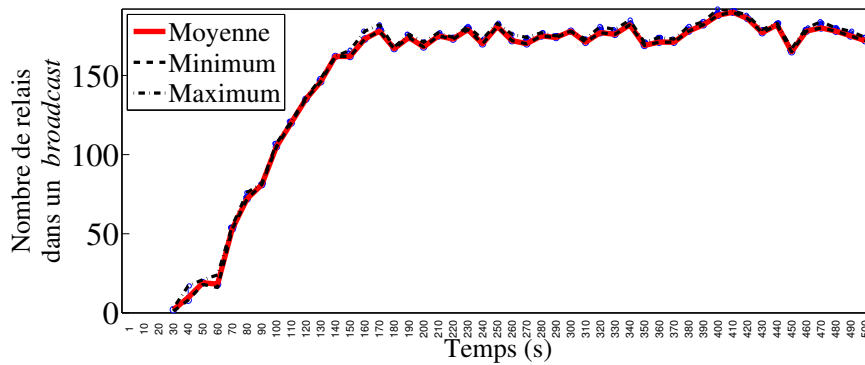


FIGURE 4.21 – Nombre de nœuds relais MPR impliqués dans une transmission de type *broadcast* (Ms15) – Scénario S5, méthode des MPRs

Le nombre de nœuds sélectionnant un même nœud MPR (métrique Ms11) est lié à la densité et au réseau routier. Pour une densité faible (scénarios S1, S4, et S7), la moyenne du nombre de nœuds sélectionnant un même nœud MPR se situe entre 3 et 5,6 nœuds ; pour une densité moyenne (scénarios S2, S5, et S8) entre 6,8 et 11,2 nœuds ; enfin, pour une densité élevée (scénarios S3, S6, et S9) entre 9 et 14,3 nœuds. Cette moyenne augmente donc avec la densité du réseau : un constat similaire avait été fait avec CBL. Pour un même réseau routier, cette métrique augmente de manière identique lorsque la densité augmente. Entre le réseau routier R1 (scénarios S1 à S3) et le réseau routier R2 (scénarios S4 à S6), un second sens de circulation est ajouté. Comme la méthode des MPRs ne fait aucune distinction sur le sens de circulation, les résultats entre ces deux réseaux routiers, pour une même densité, sont doublés. Les résultats obtenus entre le réseau routier R2 et R3 sont proches.

Le nombre de nœuds abonnés à un nœud relais est plus faible avec CBL qu'avec la méthode des MPRs, en particulier pour les scénarios de faible et moyenne densité car les nœuds feuille se rattachent de façon unique à un nœud branche. Une fois le nombre de nœuds branche atteint pour la formation de la chaîne CBL, les nœuds résiduels sont des nœuds feuille ; lorsque la densité augmente, le nombre de nœuds feuille abonnés par nœud branche augmente alors. C'est pourquoi nous constatons, dans le cas du réseau R1 à une direction de circulation, que le nombre de nœuds abonnés devient supérieur pour CBL par rapport à la méthode des MPRs (augmentation de 2,21 nœuds). Dans le cas des réseaux R2 et R3, l'écart des deux méthodes diminue avec l'augmentation de la densité même si le nombre d'abonnés dans le cas de CBL reste moindre par rapport à la méthode des MPRs pour ces réseaux.

Le nombre de nœuds élus en tant que MPR par un nœud (métrique Ms12) augmente avec la densité du réseau (1,8 et 6,7 nœuds). La structure CBL imposant un nombre restreint de nœud relais à un ou deux nœuds, la métrique Ms8 est plafonnée à 2 et tend vers 1 avec l'augmentation de la densité en raison de l'augmentation du nombre de nœuds feuille. CBL limite donc le nombre de nœuds relais impliqués lors de la première retransmission d'un message.

TABLEAU 4.3 – Valeurs moyennes des métriques de performances Ms10 à Ms15 calculées dans l’intervalle de temps de 150 à 500 s – Scénarios S1 à S9, méthode des MPRs, comparaison CBL version 2, simulation Matlab

Réseaux		R1			R2			R3		
Scénarios		1	2	3	4	5	6	7	8	9
Densité		-	+	++	-	+	++	-	+	++
Nombre de nœuds		25	102	198	50	205	369	46	189	390
Métriques	Ms10 : Non_MPR/Réseau (%)	22,1	7,4	10,2	6,9	6,5	7,5	11,5	7,8	8,0
	<i>Diff avec Ms1</i>	-16,1	-74,1	-79,9	-26,8	-72,5	-79,7	-25,6	-66,2	-76,5
	Ms11 : Nb_MPR_Selector	3,0	6,8	9,0	5,6	11,2	13,9	5,1	10,8	14,3
	<i>Diff avec Ms5</i>	+0,62	+0,63	-1,84	+3,31	+5,69	+5,42	+2,98	+6,25	+7,17
	Ms12 : Nb_MPR_Set	1,8	3,30	3,3	3,0	5,2	6,2	2,8	5,2	6,7
	<i>Diff avec Ms8</i>	+0,3	+2,14	+2,21	+1,48	+4,02	+5,09	+1,38	+3,99	+5,57
	Ms13 : Durée_MPR (s)	39,1	41,6	34,6	53,4	53,8	53,7	43,9	48,3	50,7
	<i>Diff avec Ms7</i>	+9,7	+2,2	-8,3	+23,4	+17,5	+13,5	+20,3	+16,3	+19,2
	Ms14 : Durée_élection_MPR (s)	29,0	13,0	9,4	10,9	8,0	6,6	10,1	8,4	6,0
	<i>Diff avec Ms6</i>	-27,3	-41	-45,9	-46,7	-45,4	-47,1	-42,2	-36	-34,5
	Ms15 : MPR/Broadcast	13,9	66,4	117,9	37,2	175,9	317,2	31,4	155,7	308,7
	<i>Diff avec Ms9</i>	-1,1	+47,7	+98,5	+4,5	+133	+270	+3	+106,9	+248,8

Où *Diff* est la différence “valeurs de la méthode des MPRs moins celles de CBL version 2”.

Définition des métriques structurales en section 4.1.1.1.

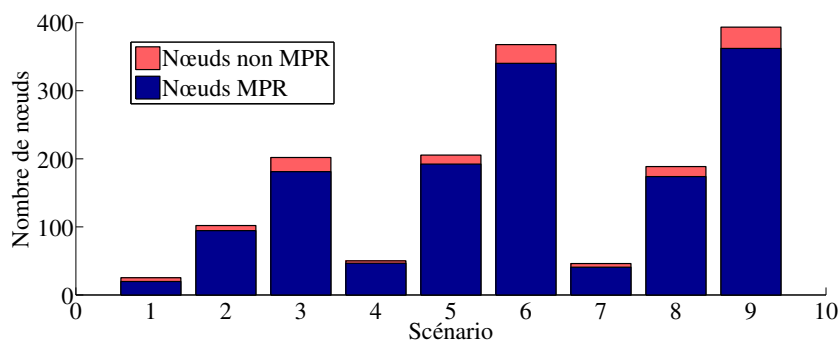


FIGURE 4.22 – Nombre de nœuds MPRs et non-MPRs dans le réseau – Scénarios S1 à S9, méthode des MPRs

La durée durant laquelle un nœud reste un nœud MPR (métrique Ms13) varie en fonction des scénarios entre 34,6 et 53,8 s. La métrique Ms6 est du même ordre de grandeur pour CBL, légèrement plus stable autour de 50 s.

La durée d’élection du même nœud MPR (métrique Ms14) varie entre 6,0 s et 29,0 s en fonction de la densité. Les meilleurs résultats sont obtenus pour une densité faible : chaque nœud ayant moins de nœuds voisins, la probabilité de modification des choix de nœuds MPR est plus faible. Pour CBL, lorsque le nombre de nœuds branche est suffisant pour ne pas avoir de rupture de chaînes due à des défauts de nœuds (densité moyenne et forte), cette durée est stabilisée autour de 40 s pour R1 et R2 (sans entrée-sortie) et 30 s pour R3. Ms6 est le double voire le quintuple de Ms14 en fonction des scénarios, confirmant le constat d’une plus grande stabilité de la structure CBL établie avec le scénario S5.

Le nombre de nœuds relais MPR impliqués dans une transmission de type *broadcast* (métrique Ms15) augmente avec la densité du réseau (entre 13,9 et 312 nœuds). Le nombre de ces nœuds par rapport au nombre de nœuds total varie en fonction du réseau routier. Il est compris entre 56 et 65% pour le réseau R1, de 74 à 85% pour le réseau R2 et de 67 à 82% pour le réseau R3. Contrairement à la méthode des MPRs, CBL permet de limiter ce pourcentage de nœuds relais qui ne dépasse pas 15% des nœuds pour les scénarios de forte densité, 25% pour les scénarios de moyenne densité, et 70% pour les scénarios de faible densité où CBL est en dessous de ses performances moyennes, mais toujours plus efficace que la

méthode des MPRs.

4.2 Analyses de performances des protocoles CBL-OLSR et OLSR sous OPNET

L'analyse structurelle précédente avec MATLAB ne prend en compte ni la technologie de communication radio, ni l'émission des messages TC. L'utilisation du logiciel OPNET Modeler, édité par Riverbed, permet de modéliser intégralement un nœud au niveau des ses fonctionnalités de communication. Ainsi, CBL version 2 (section 2.5) a été intégré à un processus OLSR pour donner CBL-OLSR. Chaque "nœud CBL-OLSR" met alors en œuvre le protocole de routage CBL-OLSR, la pile de protocole TCP/UDP/IP et la technologie radio IEEE 802.11p. Les paramètres du logiciel utilisé lors des simulations ont été présentés en section 3.3.2.1 et dans le tableau 3.9.

Cette partie comporte une présentation en deux temps. Dans un premier temps, section 4.2.1, une analyse structurelle de CBL version 2 intégré à CBL-OLSR est réalisée sur les scénarios de mobilité S1 à S5, S7 et S8. Les simulations des scénarios S6 et S9 n'ont pu être lancées en raison d'un nombre trop élevé de nœuds nécessaires à la modélisation de l'ensemble des trajectoires des véhicules de la date 0 à celle 500 s : 999 nœuds pour S6 et 1454 nœuds pour S9 (tableau 3.7) ; ce nombre est bien supérieur au nombre de nœuds dans le réseau supporté par OPNET à un instant donné. L'objectif est de valider la mise en œuvre de CBL sous OPNET Riverbed Modeler par comparaison avec les résultats précédemment obtenus avec MATLAB. Les métriques de performances structurelles utilisées sous OPNET Riverbed Modeler sont celles précédemment spécifiées et utilisées sous MATLAB en section 4.1.1.1. Elles ont été programmées sous forme de statistiques lors de l'intégration du protocole CBL à OLSR, à l'exception des métriques Ms4, Ms8 et Ms9.

Dans un second temps, section 4.2.2, une analyse du trafic de routage (messages HELLO et TC) a été menée, tout d'abord sur le scénario S5, puis sur l'ensemble des scénarios S1 à S9. Elle utilise les mêmes conditions de simulations pour les modélisations comportant les nœuds CBL-OLSR et celles les nœuds OLSR.

Pour l'ensemble de ces simulations sous Riverbed OPNET Modeler, les valeurs prises pour les paramètres du trafic de routage des protocoles OLSR et CBL-OLSR sont résumées dans le tableau 4.4. Deux ensembles d'intervalles temporels, nommés cas A et cas B, sont considérés dans l'objectif de tenir compte de l'aspect dynamique des réseaux véhiculaires. Il s'agit d'étudier l'impact de la fréquence d'envoi des messages HELLO sur la structure formée par la méthode de *clustering* de CBL. Les intervalles de temps du cas A ont pour valeur la moitié de ceux du cas B. Les intervalles de temps du cas B correspondent aux valeurs des paramètres définis par défaut dans le document *Request For Comments* (RFC) 3626 [31]. Ainsi, la périodicité d'envoi d'un message HELLO de découverte de son voisinage est de deux secondes pour le cas B et d'une seconde pour le cas A (tableau 4.4).

En outre, lors de l'exécution de chaque scénario, chaque simulation est exécutée avec dix valeurs de *seed* différentes. Ces valeurs permettent d'initialiser le générateur de nombre aléatoire du logiciel Riverbed OPNET Modeler. L'usage de plusieurs *seed* sert à éviter de privilégier une situation particulière, potentiellement singulière, de simulation qui serait induite par une séquence unique de nombres aléatoires favorable à l'un ou l'autre protocole.

TABLEAU 4.4 – Paramètres des protocoles OLSR et CBL-OLSR pour deux ensembles, cas A et cas B, d'intervalles temporels

Attributs	Cas A	Cas B
Intervalle d'attente entre chaque message Hello (Hello interval)	1 s	2 s
Intervalle d'attente entre chaque message TC (TC interval)	2,5 s	5 s
Neighbor hold time : Vtime (3*Hello interval)	3 s	6 s
Topology hold time (3*TC interval)	7,5 s	15 s
Duplicate message hold time	15 s	30 s

4.2.1 Analyse de l'intégration du schéma CBL dans le modèle du processus OLSR d'OPNET

Les valeurs moyennes des métriques structurelles calculées une fois le réseau stabilisé et obtenues avec la version CBL-OLSR du schéma CBL version 2 sont reportées au tableau 4.5 pour les deux ensembles de valeurs des intervalles de temps (cas A et cas B). Une comparaison des valeurs des métriques Ms1 à Ms3 et Ms5 à Ms7 obtenues sous OPNET pour le protocole CBL-OLSR avec celles précédemment obtenues pour CBL version 2 sous MATLAB (tableau 4.2) y a aussi été ajoutée (lignes “*Diff*” du tableau). Seul le cas A est considéré pour la comparaison, les simulations sous MATLAB ayant été menées pour une fréquence d’envoi des messages HELLO de 1 Hz, soit un *HELLO Interval* de 1 s.

TABLEAU 4.5 – Valeurs moyennes des métriques de performances Ms1 à Ms3 et Ms5 à Ms7 calculées dans l’intervalle de temps de 150 à 500 s – Scénarios S1 à S5, S7 et S8, CBL-OLSR, simulation de CBL version 2 sous OPNET

Réseaux		R1			R2		R3		
Scénarios		1	2	3	4	5	7	8	
Densité		-	+	++	-	+	-	+	
Nombre de nœuds		25	102	198	50	205	46	189	
Métriques	Ms1 : Feuille/Réseau (%)	cas A	35,5	84	91,3	43,6	82,4	54,5	74,2
		<i>Diff</i>	-2,7	+2,5	+1,2	+9,9	+3,4	+17,4	+0,2
		cas B	45,8	84,5	91,7	45,2	81,9	58,3	75,7
		<i>Diff</i>							
	Ms2 : NB_Chains	cas A	1,36	1,40	1,52	3,01	3,24	3,11	3,97
		<i>Diff</i>	-0,32	+0,07	+0,18	-0,45	+0,39	-1,77	-0,93
		cas B	1,53	1,68	1,73	3,30	3,62	2,92	3,20
		<i>Diff</i>							
	Ms3 : Branche/Chaîne	cas A	10,4	11,7	11,7	8,35	8,62	5,78	9,52
		<i>Diff</i>	+1	-2,4	-2,7	-1,25	-6,18	+0,08	-0,38
		cas B	8,49	8,68	8,75	6,57	8,34	5,26	10,5
		<i>Diff</i>							
	Ms5 : Feuille/Branche	cas A	0,542	5,29	10,7	0,775	4,74	1,07	2,88
		<i>Diff</i>	-1,838	-0,88	-0,14	-1,515	-0,77	-1,05	-1,67
	cas B	0,79	5,41	11	0,797	4,47	1,12	3,04	
	<i>Diff</i>								
Ms6 : Durée_Branche (s)	cas A	66	61	60,5	61,2	57,5	50,6	46,5	
	<i>Diff</i>	+9,7	+7	+5,2	+3,6	+4,1	-1,7	+2,1	
	cas B	60,4	59,6	62,1	59,5	57,2	49	45,9	
	<i>Diff</i>								
Ms7 : Durée_rattachement_Branche (s)	cas A	28	40,5	41,8	35,9	31,4	28	27,7	
	<i>Diff</i>	-1,4	+1,1	-1,1	+5,9	-4,9	+4,4	-4,3	
	cas B	33,8	39	42,2	30,9	26,4	25,4	24,6	
	<i>Diff</i>								

Où *Diff* est la différence “valeurs de CBL avec MATLAB moins celles de CBL avec OPNET”.

Définition des métriques structurelles en section 4.1.1.1.

L’étude de la hausse et de la baisse des métriques structurelles de performance obtenues sous OPNET comparativement à celles obtenues sous MATLAB montre peu de variations (tableau 4.5, lignes “*Diff*”). Les résultats des deux simulations sont proches avec quelques différences à la marge qui peuvent être attribuées à la prise en compte notamment de la technologie radio et des files d’attente dans le traitement en émission et réception des messages de routage au niveau des couches basses et du modèle de CBL-OLSR. Nous pouvons donc considérer que l’intégration du schéma de *clustering* CBL dans le processus OLSR de Riverbed OPNET Modeler est correcte.

L’étude de l’impact de la fréquence d’envoi des messages HELLO sur le protocole CBL-OLSR (lignes cas A et cas B de chaque métrique) révèle là encore peu de différences.

Le nombre de nœuds feuille dans le réseau (métriques Ms1) augmente de quelques pourcentages dans tous les scénarios sauf le scénario S5.

Le nombre de chaînes dans le réseau (métrique Ms2) augmente également entre les cas A et B, car la fréquence des messages HELLO est divisée par deux, ce qui implique un délai supplémentaire pour

la reconstruction des chaînes lors d'une rupture. Seul le scénario S8 ne suit pas cette tendance avec un nombre de chaînes qui diminue en moyenne de 3,97 à 3,20.

Le nombre de nœuds branche par chaîne (métrique Ms3) et le nombre de nœuds feuille par branche (métrique Ms5) sont corrélés avec les métriques Ms1 et Ms2. Ainsi, Ms3 est plus faible dans le cas B que dans le cas A, sauf pour le scénario S8. Ms5 augmente pour tous les scénarios entre le cas A et B.

Globalement, les résultats du tableau 4.5 ne permettent pas sur les scénarios testés de conclure à une amélioration ou une dégradation des résultats obtenus. Les résultats restent stables, avec pour certains scénarios une légère diminution du nombre moyen de nœuds branche par chaîne du cas A au cas B pour le réseau R1 à un sens de circulation, mais qui n'est pas confirmé dans les scénarios des réseaux R2 et R3 à double sens de circulation. D'autres variations à la baisse ou à la hausse sont constatées pour chacune des autres métriques. **La structure CBL reste stable pour ces deux cas de simulation.**

4.2.2 Analyse du trafic de routage sous Riverbed OPNET Modeler

4.2.2.1 Métriques d'évaluation du trafic de routage

L'analyse de performances avec Riverbed OPNET Modeler permet l'ajout de nouvelles métriques de routage. Ces métriques de performances (Mr - Métriques de routage) sont identiques pour les protocoles CBL-OLSR et OLSR :

- Mr1 (Charge_routage) : débit en bits/s des messages de routage envoyés sur le réseau ;
- Mr2 (HELLO_total) : débit en bits/s des messages HELLO envoyés sur le réseau ;
- Mr3 (TC_total) : débit en bits/s des messages TC envoyés et retransmis sur le réseau ;
- Mr4 (HELLO/Charge_routage) : pourcentage d'occupation des messages HELLO envoyés par rapport à la charge de routage ;
- Mr5 (TC_total/Charge_routage) : pourcentage d'occupation des messages TC envoyés et retransmis par rapport à la charge de routage ;
- Mr6 (TC_relayé/TC_envoyé) : pourcentage entre les messages TC relayés par rapport aux messages TC envoyés et relayés ;
- Mr7 (Délai WLAN) : délai en ms entre les deux nœuds au niveau de la couche LAN.

4.2.2.2 CBL-OLSR version 2 – scénario de mobilité S5, cas A

La figure 4.23 trace la charge de routage du réseau VANET (métrique Mr1) au cours du temps pour le cas A du scénario S5. Cette charge de routage augmente au fur et à mesure de l'initialisation des nœuds dans le réseau. Après stabilisation, Mr1 atteint en moyenne un débit de 463 kbit/s. La variabilité autour de cette moyenne fluctue entre 350 kbit/s et 510 kbit/s et correspond aux 10 simulations (10 valeurs de *seed* différentes). Cette variabilité s'explique par l'ordre de réception des messages HELLO qui peut être différent dans chaque configuration. Cet ordre de réception impacte la structure formée par CBL et indirectement la métrique Mr1. Par exemple, dans une simulation un nœud est de type branche alors qu'il est de type feuille dans une autre. Ce nœud émet et relaie les messages de routage TC dans la première situation mais pas dans la seconde. Ces différences expliquent la variabilité de la charge de trafic de ces deux simulations. Notons que l'occupation du trafic de routage par rapport à la bande passante totale (entre 3 Mbit/s et 27 Mbit/s) est limité : respectivement de 13% à 1,5% .

La charge du trafic de routage des messages HELLO et TC (métriques Mr2 et Mr3) est illustrée figure 4.24. Après une montée en charge, le trafic de messages TC représente en moyenne 83,2 kbit/s soit environ 20% du trafic de routage. Le trafic des messages HELLO représente quant à lui 379 kbit/s soit les 80% restants.

Le délai WLAN (métrique Mr7) moyen est de 0,18 ms (figure 4.25). Ce délai est très faible indiquant que le médium de communication n'est pas saturé. Les points extrêmes de pics de délai dépassant les 0,2 ms sont observés pour l'une des configurations (une valeur de *seed* particulière). Ces valeurs peuvent s'expliquer par un accès au médium plus long à cause, par exemple, d'une interférence entre deux nœuds ayant envoyé un paquet simultanément.

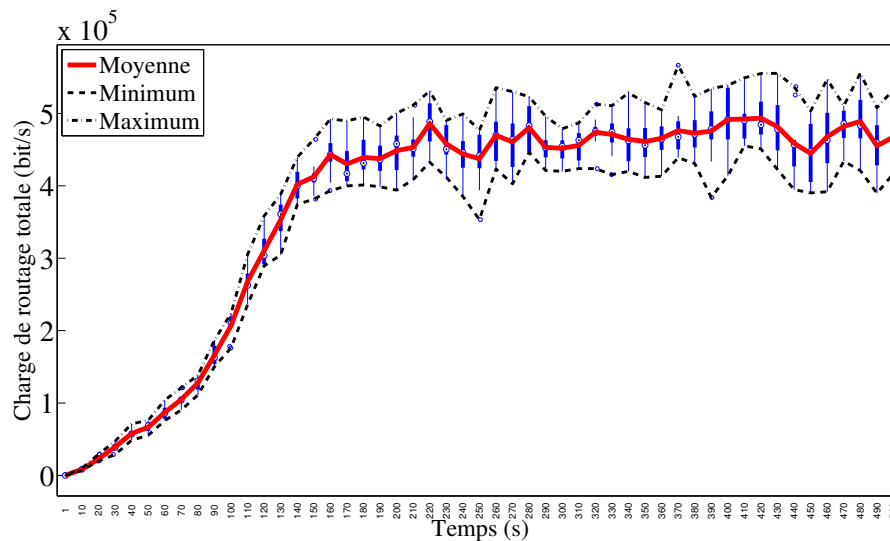
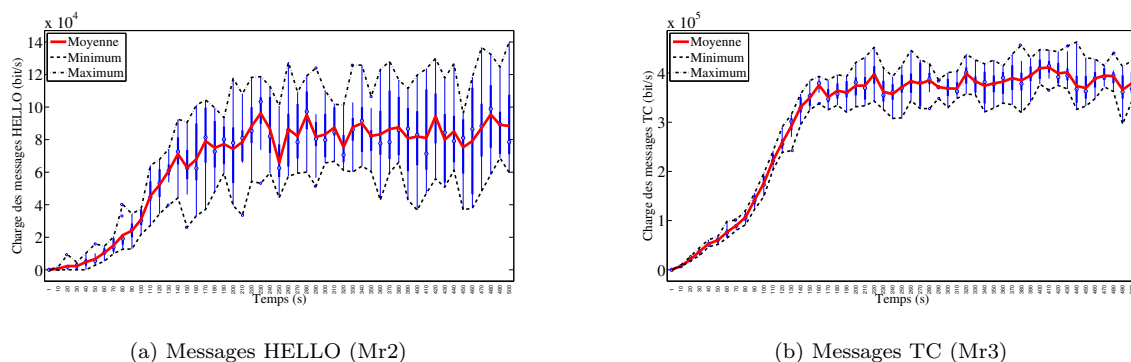


FIGURE 4.23 – Charge de routage totale envoyée (Mr1) – Scénario S5 cas A, CBL-OLSR



(a) Messages HELLO (Mr2)

(b) Messages TC (Mr3)

FIGURE 4.24 – Trafic de routage envoyé – Scénario S5 cas A, CBL-OLSR

4.2.2.3 CBL-OLSR version 2 – scénarios de mobilité S1-S5, S7 et S8

Nous étendons notre analyse du trafic de routage de CBL-OLSR aux autres scénarios de mobilité S1 à S5, S7 et S8. Le tableau 4.6 regroupe les valeurs moyennes des métriques de routage une fois le réseau stabilisé (entre 150 s et 500 s).

La charge du trafic de routage global (métrique Mr1) augmente lorsque la densité du réseau augmente. Plus il y a de nœuds dans le réseau, plus le nombre et la taille des messages HELLO et TC augmentent. Cette augmentation n'est pas linéaire. Par exemple, entre le cas A des scénarios S2 et S3, le nombre de nœuds simultanés dans le réseau double quasiment passant de 102 à 198 nœuds. La charge de routage est quant à elle multipliée par un facteur de 2,8. D'une part, la charge de routage des messages HELLO est multipliée par un facteur de 3,3 passant en moyenne de 93,2 kbit/s à 309 kbit/s, car le nombre et la taille des messages HELLO augmentent (les nœuds possèdent plus de voisins à un saut).

D'autre part, la charge de routage des messages TC entre les scénarios S2 et S3 est multipliée par un facteur de 1,7 passant en moyenne de 33,2 kbit/s à 48,9 kbit/s. Elle suit l'augmentation du nombre de nœuds feuille par nœud branche dans le réseau. Un message TC contenant les adresses de chaque nœud feuille rattaché à son nœud branche, la taille d'un message TC croît avec le nombre de feuille rattachés au nœud branche qui le génère.

La charge de routage entre les cas A et B est en grande partie divisée par 2. Ceci s'explique par le fait que la fréquence des messages de routage est divisée par deux entre les cas A et B.

On remarque que plus la densité du réseau augmente, plus la part d'utilisation des ressources par les messages HELLO augmente par rapport à celle des messages TC. Par exemple, pour le scénario S1

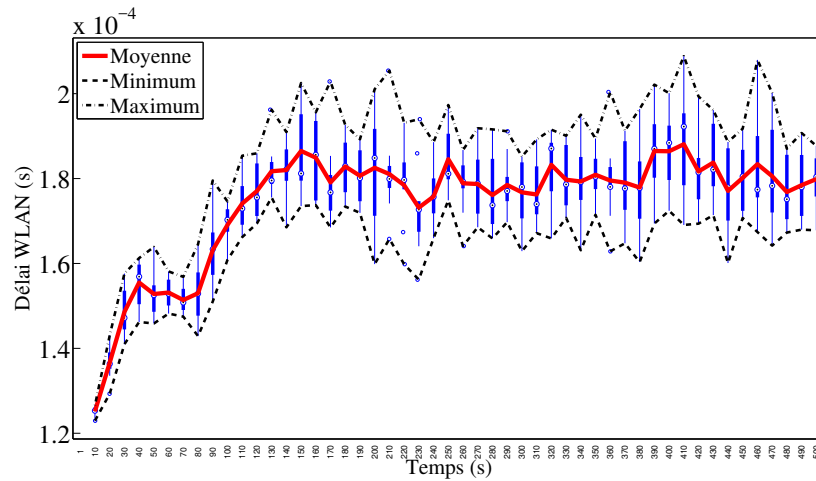


FIGURE 4.25 – Délai WLAN (Mr7) – Scénario S5 cas A, CBL-OLSR

comptabilisant très peu de nœuds (25 nœuds), le trafic de routage s'équilibre de moitié entre les messages HELLO et les messages TC. Pour le scénario S2 comptant 102 nœuds en moyenne, 75% d'utilisation de la charge de routage est dédiée aux messages HELLO et pour le scénario S3 comportant une forte densité de nœuds, 85% est utilisée pour ces mêmes messages contre seulement 25% et 15% pour les messages TC, respectivement. Cette analyse est identique pour le réseau R2 qui possède un double sens de circulation, mais elle n'est pas valide pour le réseau R3 avec une entrée et une sortie. En effet, dans ce réseau le pourcentage d'occupation des messages TC augmente lorsque la densité de nœuds augmente.

La part de messages TC relayés est très importante par rapport aux messages TC émis (la proportion varie de 81,5% à 92,2%). Les messages TC sont relayés par les nœuds branche dans tout le réseau, ce qui explique ce pourcentage important.

Concernant la métrique Mr7 mesurant les délais *Wireless Local Area Network* (WLAN), le nombre de nœuds fait légèrement augmenter ce délai, mais il reste relativement faible. Ce qui indique que, même avec le scénario S5 comportant 200 nœuds simultanés dans le réseau, le médium de communication n'est pas saturé.

4.2.2.4 OLSR – scénario de mobilité S5

Cette section a pour objectif de présenter les résultats du scénario S5 obtenus avec OLSR à l'aide des métriques de routage comparativement à celles obtenues avec CBL-OLSR.

La charge totale du trafic de routage (métrique Mr1) atteint en moyenne un débit de 4180 kbit/s et 2280 kbit/s pour OLSR respectivement pour le cas A et B. Ce trafic se décompose entre le débit des messages HELLO en moyenne de 369 kbit/s pour le cas A et 207 kbit/s cas B (figure 4.26) et celui des messages TC de 3810 kbit/s pour le cas A et 2070 kbit/s cas B (figure 4.27).

La charge de trafic des messages HELLO (métrique Mr2) est quasiment équivalente entre CBL-OLSR et OLSR pour un même cas (A ou B). Cette charge de trafic est directement liée au nombre de nœuds présents dans le réseau VANET (ici 205 nœuds). La différence de débit en faveur du protocole OLSR par rapport à CBL-OLSR s'explique par la taille des messages HELLO de ces deux protocoles. CBL-OLSR définit, en plus des champs déjà présents dans les messages HELLO d'OLSR, des champs dédiés aux informations de position et de type de nœuds. Ces ajouts augmentent la taille de l'en-tête des messages HELLO de CBL-OLSR expliquant la différence présente sur la figure 4.26. Entre le cas A et B la charge de trafic des messages HELLO est divisée par deux, car la fréquence d'envois des messages est également divisée par deux.

La charge de trafic des messages TC (métrique Mr3) est illustrée figure 4.27. La différence entre CBL-OLSR et OLSR est remarquable. OLSR génère de 3 à 17 fois plus de trafic TC que CBL-OLSR. Les messages TC sont relayés en mode *broadcast*. Comme nous l'avons vu en section 4.1.1.4 et 4.1.2.3, les performances structurelles entre CBL et la méthode des MPRs sont très différentes concernant les

TABLEAU 4.6 – Valeurs moyennes des métriques de performances Mr1 à Mr7 calculées sur l'intervalle de temps de 150 à 500 s – Scénarios S1 à S9, CBL-OLSR

Réseaux		R1			R2		R3		
Scénarios		1	2	3	4	5	7	8	
Densité		-	+	++	-	+	-	+	
Nombre de nœuds		25	102	198	50	205	46	189	
Métriques	Mr1 : Charge_routage (kbit/s)	cas A	26,8	126	358	58,4	463	44,2	371
		cas B	11,3	61,3	179	29,6	232	23,3	165
	Mr2 : HELLO_total (kbit/s)	cas A	10,7	93,2	309	32,5	379	27,1	291
		cas B	5,36	47,5	158	17,7	195	14,7	151
	Mr3 : TC_total (kbit/s)	cas A	16,1	33,2	48,9	25,9	83,2	17,1	80,2
		cas B	5,91	13,7	20,5	11,8	37,2	8,6	14
	Mr4 : HELLO/Charge_routage (%)	cas A	43,8	74,7	86,1	57,1	82,4	63,9	79,2
		cas B	52,3	78,6	88,3	61,4	84,3	66	91,5
	Mr5 : TC_total/Charge_routage (%)	cas A	56,2	25,3	13,9	42,9	17,6	36,1	20,8
		cas B	47,7	21,4	11,7	38,6	15,7	34	8,5
	Mr6 : TC_relayé/TC_envoyé (%)	cas A	88,3	91,1	91	88,1	92,2	82,8	91,3
		cas B	83,4	87,5	87,6	86,4	91	81,5	89,1
	Mr7 : Délai WLAN (s)	cas A	0,111	0,145	0,196	0,119	0,181	0,114	0,132
		cas B	0,112	0,147	0,2	0,12	0,185	0,115	0,133

Définition des métriques de routage en section 4.2.2.1.

métriques Ms9 et Ms15 qui comptabilisent le nombre de nœuds relais impliqués dans une transmission en mode *broadcast*. Pour le cas du scénario S5, seuls 47 nœuds en moyenne étaient nécessaires avec CBL pour relayer un message dans tout le réseau, tandis que la méthode des MPRs nécessitait au moins 175 nœuds. Ces résultats structurels expliquent donc les résultats obtenus pour la métrique Mr3.

La figure 4.28 récapitule le nombre total de bits envoyés concernant le trafic de routage durant toute la durée de la simulation (500 s) pour les protocoles CBL-OLSR et OLSR dans le cas du scénario S5. L'optimisation apportée par le schéma de *clustering* CBL est bien visible puisque le protocole CBL-OLSR envoie moins de 500 Mbits pour les cas A et B alors que le protocole OLSR a besoin pour son fonctionnement de 2000 Mbits et 3200 Mbits respectivement pour les cas B et A.

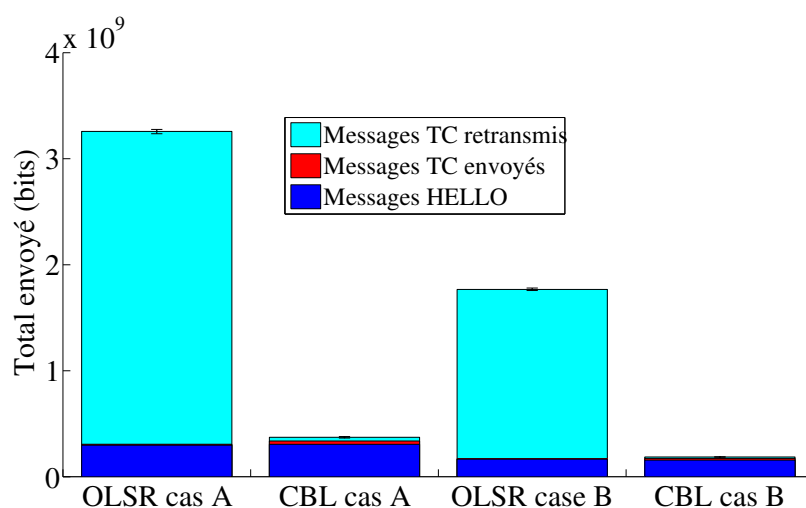


FIGURE 4.28 – Comparaison de la charge totale de trafic HELLO et TC – Scénario S5, protocoles CBL-OLSR et OLSR

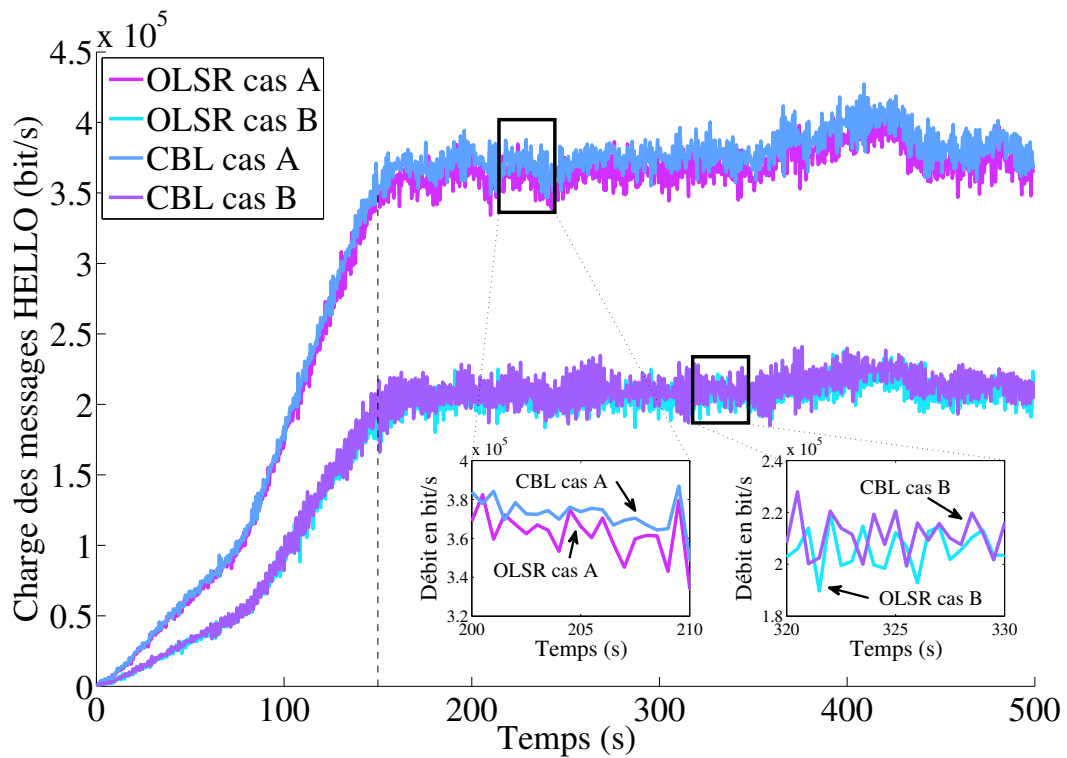


FIGURE 4.26 – Comparaison du trafic HELLO (Mr2) – Scénario S5 cas A, CBL-OLSR et OLSR

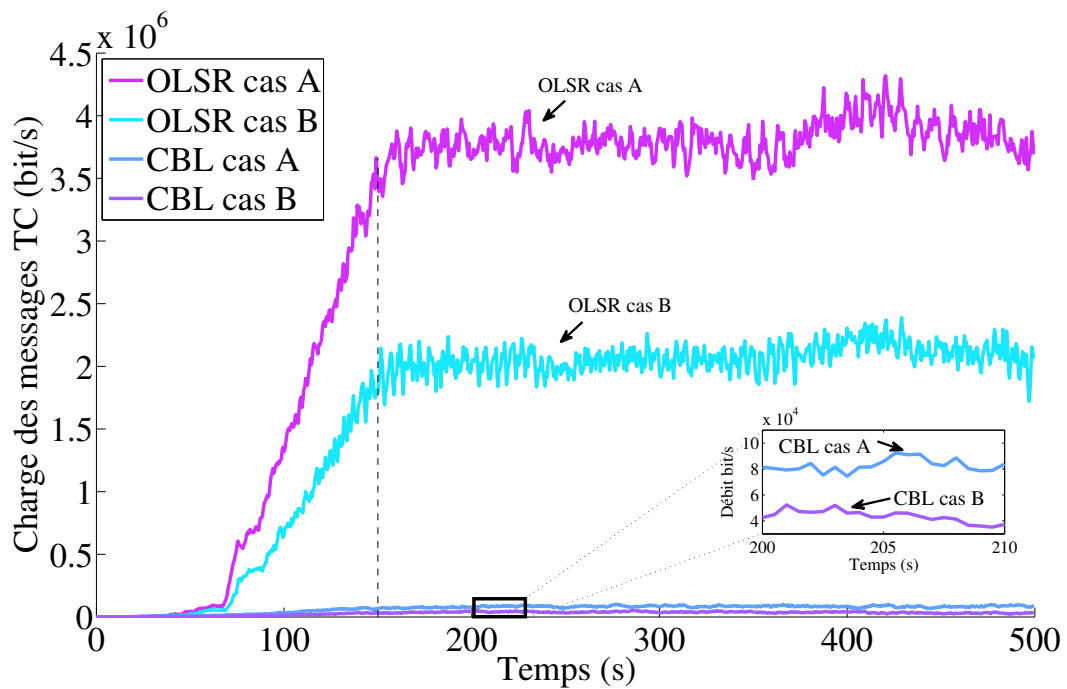


FIGURE 4.27 – Comparaison du trafic TC (Mr3) – Scénario S5 cas A, CBL-OLSR et OLSR

4.2.2.5 OLSR – scénarios de mobilité S1-S5

Le tableau 4.7 présente les résultats obtenus avec le protocole OLSR pour les scénarios S1 à S5 avec une comparaison des résultats obtenus avec CBL-OLSR.

TABLEAU 4.7 – Valeurs moyennes des métriques de performances Mr1 à Mr7 calculées sur l'intervalle de temps de 150 à 500 s – Scénarios S1 à S9, OLSR, comparaison avec CBL-OLSR

Réseaux		R1			R2		
Scénarios		1	2	3	4	5	
Densité		-	+	++	-	+	
Nombre de nœuds		25	102	198	50	205	
Métriques	Mr1 : Charge_routage (kbit/s)	cas A	28	461	1680	175	4180
		<i>Diff</i>	+1,2	+335	+1322	+116,6	+3717
	cas B	20,5	275	1090	95,9	2280	
		<i>Diff</i>	+9,2	+213,7	+911	+66,3	+2048
	Mr2 : HELLO_total (kbit/s)	cas A	9,86	90	303	30,9	369
		<i>Diff</i>	-0,84	-3,2	-6	-1,6	-10
	cas B	4,98	45,9	155	16,9	207	
		<i>Diff</i>	-0,38	-1,6	-3	-0,8	+12
	Mr3 : TC_total (kbit/s)	cas A	18,2	371	1380	144	3810
		<i>Diff</i>	+2,1	+337,8	+1331,1	+118,1	+3726,8
	cas B	15,6	229	940	78,9	2070	
		<i>Diff</i>	+9,69	+215,3	+919,5	+67,1	+2032,8
	Mr4 : HELLO/Charge_routage (%)	cas A	39,8	21,8	20,4	19,1	9,29
		<i>Diff</i>	-4	-52,9	-65,7	-38	-73,11
	cas B	28,8	18,5	16,1	19,4	9,64	
		<i>Diff</i>	-23,5	-60,1	-72,2	-42	-74,66
	Mr5 : TC_total/Charge_routage (%)	cas A	60,2	78,2	79,6	80,9	90,7
		<i>Diff</i>	+4	+52,9	+65,7	+38	+73,1
	cas B	71,2	81,5	83,9	80,6	90,4	
		<i>Diff</i>	+23,5	+60,1	+72,2	+42	+74,7
Mr6 : TC_relayé/TC_émis (%)	cas A	88,8	97,6	98,6	96,5	99,3	
	<i>Diff</i>	+0,5	+6,5	+7,6	+8,4	+7,1	
cas B	88,5	98	98,9	96,5	99,3		
	<i>Diff</i>	+5,1	+10,5	+11,3	+10,1	+8,3	
Mr7 : Délai WLAN (ms)	cas A	0,112	0,128	0,184	0,116	0,362	
	<i>Diff</i>	+0,001	-0,017	+0,184	-195,884	+0,243	
cas B	0,11	0,12	0,156	0,115	0,204		
	<i>Diff</i>	-0,002	-0,027	-0,044	-0,005	+0,019	

Où *Diff* est la différence “valeurs de OLSR moins celles de CBL-OLSR”.

Définition des métriques de routage en section 4.2.2.1.

De la même manière qu’avec CBL-OLSR, la charge de routage totale (métrique Mr1) augmente pour OLSR en fonction de la densité de nœuds du scénario. Les valeurs de cette métrique sont toujours supérieures pour OLSR par rapport à CBL-OLSR. Plus la densité de nœuds est élevée, plus l’optimisation du protocole CBL-OLSR est visible. Par exemple, dans le scénario S1 (faible densité), le protocole OLSR génère 28 kbit/s de trafic de routage, tandis que CBL-OLSR n’en envoie que 26,8 kbit/s, soit une réduction de 4,2%. Pour le scénario S3 (forte densité), le protocole OLSR génère 1680 kbit/s de trafic de routage, tandis que CBL-OLSR n’en envoie que 358 kbit/s, soit une réduction d’un facteur 4,7.

Pour OLSR, contrairement à CBL-OLSR, le pourcentage d’occupation des messages TC (métrique Mr5) augmente en proportion par rapport aux messages HELLO lorsque le nombre de nœuds du réseau augmente. Par exemple, entre les scénarios S1 et S3 du cas A, ce pourcentage passe de 60,2% à 79,6%. Le pourcentage des messages TC relayés par rapport aux messages TC émis (métrique Mr6) est toujours supérieur à 88% quelque soit le scénario, ce qui est proche des résultats obtenus avec CBL-OLSR.

Les délais WLAN sont tous inférieurs à 0,3 ms ce qui montre que le médium n’est pas saturé.

4.3 Étude de sensibilité de la structure CBL

L'objet des évaluations présentées dans cette section correspond à l'étude de sensibilité de la structure CBL. Dans un premier temps, section 4.3.1, l'évaluation porte sur les coefficients pondérateurs de la structure CBL. Cette évaluation est menée sous MATLAB avec le scénario de mobilité S5. Dans un second temps, section 4.3.2, nous nous intéressons à l'étude de la structure et du trafic de routage de CBL-OLSR avec d'autres scénarios de mobilité plus réalistes prenant en compte des densités réelles de véhicules. Enfin, l'évaluation de CBL-OLSR sous Riverbed OPNET Modeler avec les scénarios S1 à S3 et un modèle de propagation d'ondes qui tient compte d'obstructions partielles par des véhicules voisins est réalisée en section 4.3.3.

4.3.1 Sensibilité aux coefficients pondérateurs

Le schéma de *clustering* CBL fait appel à sept coefficients pondérateurs (L1, L2, L3, C1, C2, C3 et α). Comme nous l'avons vu en section 2.3.9, les seuils C1, C2, C3 et L3 ont été fixés de sorte à améliorer la réactivité de l'algorithme (section 2.3.9). La valeur des autres seuils (L1=0,5, L2=5 et $\alpha=0,5$) a été choisie de manière arbitraire. Dans cette section, nous nous intéressons à une étude de sensibilité du schéma CBL à ces trois coefficients pondérateurs, L1, L2 et α .

À cette fin, une analyse de performances des métriques structurelles de CBL a été réalisée sous MATLAB pour laquelle nous avons fait varier L1 dans l'intervalle de valeurs [0,25 ; 0,5 ; 0,75 ; 1 ; 2 ; 4], L2 dans celui [3 ; 5 ; 7] et α dans celui [0,25 ; 0,25 ; 0,75]. Le scénario de mobilité S5 a été utilisé pour cette analyse et la durée de chaque simulation a été fixée à 300 s.

La figure 4.29 représente les valeurs moyennées de la métrique structurelle Ms1 calculées lorsque le réseau est stabilisé (entre $t=150-300$ s) en fonction de la valeur des coefficients L1, L2 et α pris deux à deux. Elle est composée d'un tableau de 3 lignes et 3 colonnes contenant six graphiques d'une part (deux par colonne) et les noms des trois coefficients inscrits chacun dans une des cellules de la diagonale d'autre part. L'axe des ordonnées des graphiques donne la valeur de la métrique Ms1, l'axe des abscisses celle du coefficient correspondant à la colonne du tableau, les courbes de couleurs des graphiques donne la valeur du coefficient inscrit dans la ligne du tableau. Par exemple, sur le graphique en haut à droite, le point de la courbe bleue d'abscisse 0,25 et d'ordonnée 79,6 correspond au paramétrage suivant : L1=0,25 et $\alpha=0,25$. La valeur de 79,6% de la métrique Ms1 est le résultat du calcul de la moyenne des valeurs de Ms1 obtenues pour les simulations réalisées avec L2=3, L2=5 et L2=7.

Les résultats de la métrique Ms1 ne montrent pas d'évolution notable quel que soit le paramétrage des coefficients pondérateurs. Les valeurs de la métrique Ms1 varient entre 79,2% et 79,6%. Ces écarts ne sont pas significatifs et ne permettent pas de mettre en évidence un paramétrage optimal pour cette métrique de performances.

L'annexe E page 237 fournit les résultats de l'analyse de sensibilité aux coefficients pondérateurs des métriques structurelles Ms2, Ms5, Ms6 et Ms7 de la structure CBL. Les autres métriques Ms3, Ms4, Ms8 et Ms9 étant corrélées avec la métrique Ms1, elles n'ont pas été retenues pour cette étude. Une analyse similaire peut être effectuée : les métriques Ms2, Ms5, Ms6 et Ms7 varient peu, respectivement entre les intervalles [2,74 ; 2,82], [5,5 ; 5,7], [50,5 ; 53] et [37,5 ; 39,5].

En conclusion, le schéma CBL n'est pas très sensible aux coefficients pondérateurs. De plus, le paramétrage de l'ensemble de ces coefficients pour optimiser la structure formée par CBL ne serait valable que pour une situation de mobilité donnée. Ainsi, il nous a paru adéquat de garder un paramétrage par défaut pour effectuer l'ensemble de nos simulations.

4.3.2 Sensibilité aux scénarios de mobilité

Le schéma de *clustering* CBL doit être en mesure de s'adapter à différentes situations de mobilité. Dans cette section, nous avons étudié l'impact de nouveaux scénarios de mobilité (S10-S13), d'une part, sur les aspects structurels du schéma de *clustering* CBL et, d'autre part, sur le trafic de routage nécessaire au fonctionnement du protocole CBL-OLSR comparativement à celui d'OLSR.

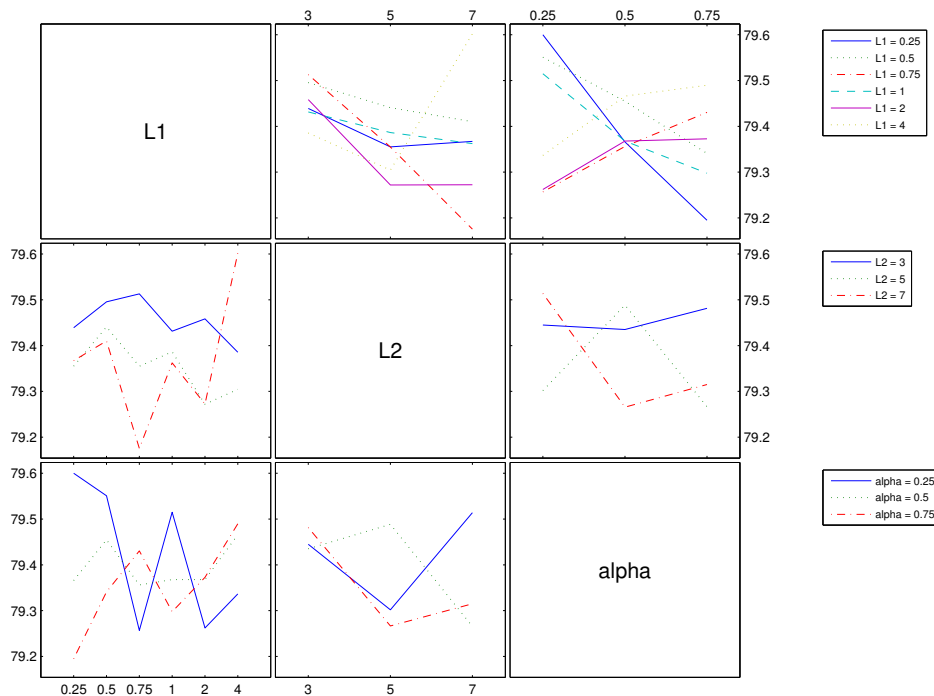


FIGURE 4.29 – Impact observé via Ms1 entre $t=150-300$ s des coefficients pondérateurs L1, L2 et alpha sur la structure CBL – Scénario S5 cas A (axe des ordonnées en %)

4.3.2.1 Résultats structurels de CBL-OLSR

Les scénarios S10 à S13 (définis dans la section 3.4.2.4) sont composés des réseaux routiers R4 et R5 comprenant l'autoroute A27 et, pour le réseau R4, la RD90. Le paramétrage du logiciel Riverbed OPNET Modeler est identique à celui des scénarios S1 à S9. Le tableau 4.8 récapitule les valeurs moyennes des métriques structurelles obtenues pour ces nouveaux scénarios routiers.

Le pourcentage de nœuds feuille dans le réseau (Ms1) est proche de celui obtenu lors des simulations avec les scénarios S1-S9. Il varie de 49,7% à 71,6% en fonction de la densité de nœuds. La figure F.1 (en annexe G page 245) illustre le nombre de nœuds branche et feuille dans chaque scénario. Nous remarquons qu'environ 50 nœuds sont nécessaires pour construire les quatre chaînes attendues dans le réseau R4 et environ 40 nœuds pour le réseau R5. Après ce seuil, tous les autres nœuds sont des nœuds feuille.

Pour le scénario S10 qui comporte l'autoroute A27 et la RD 90, le nombre de chaînes dans le réseau devrait être de 4, tandis que dans les autres scénarios il devrait être de 2. Cependant, la valeur de la métrique Ms2 du nombre de chaînes est supérieure à l'objectif attendu du fait des ruptures de chaînes. Le nombre de chaînes le plus faible est obtenu pour le scénario S12 qui comporte une forte densité de nœuds. Les scénarios ayant une densité plus faible de nœuds, et en particulier le scénario S13, souffrent de l'absence localisée de nœuds ne permettant parfois pas de construire la chaîne comme c'était le cas avec les scénarios S1 et S4. Les résultats obtenus avec le cas B sont moins bons que ceux du cas A puisque la fréquence des messages HELLO permettant le bon fonctionnement du protocole a été divisée par deux.

Les résultats des métriques Ms3, Ms5, et Ms6 sont proches de ceux obtenus pour les scénarios de mobilité avec une faible ou moyenne densité de nœuds étudiés précédemment. La métrique Ms7 montre des résultats en deçà de ceux obtenus dans le cas des scénarios de mobilité S1-S9 : le temps de rattachement au même nœud branche est compris entre 27,3 s et 8,83 s.

Pour conclure, les résultats structurels du protocole CBL-OSLR pour un nouveau cas de mobilité sont proches de ceux obtenus avec les scénarios S1-S9.

TABLEAU 4.8 – Valeurs moyennes des métriques de performances Ms1 à Ms7 pour les scénarios S10 à S13 calculées dans l'intervalle de temps de 150 à 500 s pour le protocole CBL-OLSR

Réseaux		R4		R5		
Scénarios		10	11	12	13	
Densité		+	+	++	-	
Nombre de nœuds		128	116	137	67	
Métriques	Ms1 : Feuille/Réseau (%)	cas A	62,4	65,3	70,8	49,7
		cas B	63,8	67	71,6	51,4
	Ms2 : NB_Chains	cas A	4,43	3,63	2,29	3,21
		cas B	5,06	3,69	3,30	3,86
	Ms3 : Branche/Chaîne	cas A	10,83	10,41	17,33	11,07
		cas B	9,15	8,36	11,72	10,35
	Ms5 : Feuilles/Branche	cas A	1,66	1,89	2,46	1,01
		cas B	1,72	1,99	2,51	1,08
	Ms6 : Durée_Branche (s)	cas A	57,7	61,3	64,5	69,5
		cas B	56,1	59,7	63,7	71,4
	Ms7 : Durée_rattachement_Branche (s)	cas A	17,8	17,5	27,3	24,2
		cas B	9,22	8,83	11,7	11,6

Définition des métriques structurelles en section 4.1.1.1.

4.3.2.2 Analyse du trafic de routage de CBL-OLSR versus OLSR – scénario de mobilité S10

Les résultats concernant les métriques de routage des protocoles CBL-OLSR et OLSR pour le scénario de mobilité S10 sont présentés figure 4.30 et 4.31. L'analyse de ces résultats est identique à celle faite en section 4.2.2.4. Le trafic des messages HELLO est identique entre CBL-OLSR et OLSR modulo la taille du paquet. La charge de trafic des messages TC est réduite de 7 à 17 fois (respectivement pour le cas A et B) avec CBL-OLSR par rapport à OLSR. La figure 4.32 récapitule la charge de routage envoyée durant toute la simulation. Le protocole CBL-OLSR envoie moins de 200 Mbits pour les cas A et B tandis qu'OLSR envoie au moins 550 Mbits.

Des résultats analogues sont présentés en annexe F à la page 241 pour les scénarios S11 à S13 définis en section 3.4.2.4.

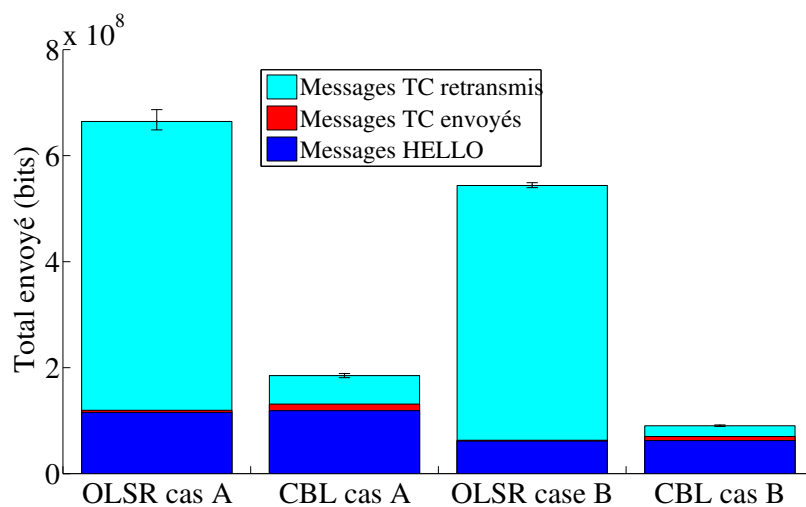


FIGURE 4.32 – Comparaison de la charge totale de trafic HELLO et TC – Scénario S10 cas A, protocoles CBL-OLSR et OLSR

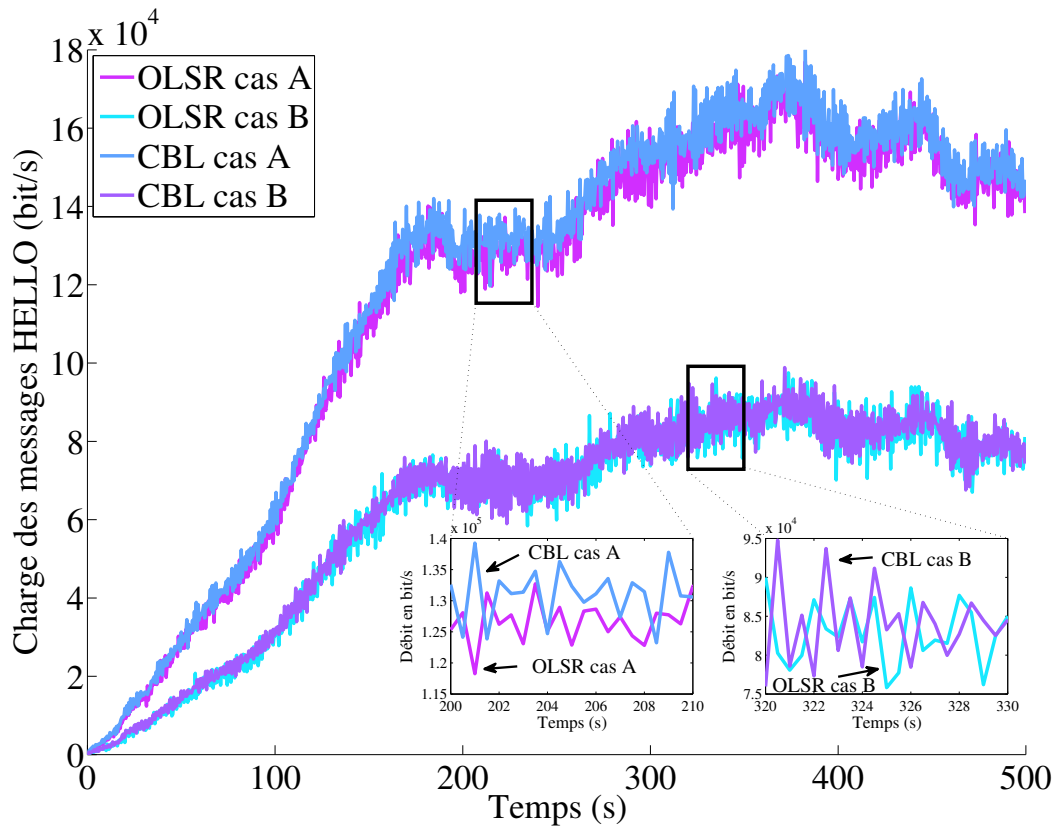


FIGURE 4.30 – Comparaison du trafic HELLO (Mr2) – Scénario S10 cas A, CBL-OLSR et OLSR

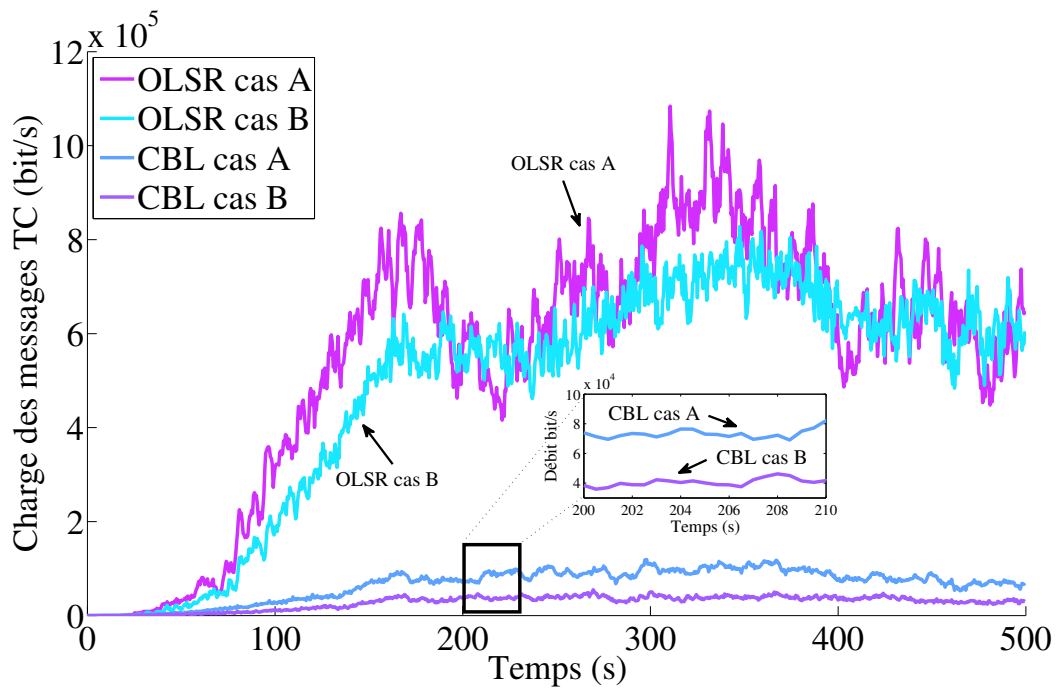


FIGURE 4.31 – Comparaison du trafic TC (Mr3) – Scénario S10 cas A, CBL-OLSR et OLSR

4.3.3 Sensibilité aux modèles de propagation

Le modèle d'atténuation en espace libre a jusqu'à présent été utilisé dans les simulations menées pour l'évaluation structurelle du schéma CBL. Afin d'évaluer la sensibilité de CBL-OLSR au modèle de propagation, nous utilisons cette fois le modèle de propagation de [9] dont l'implémentation sous OPNET et les paramètres de simulation ont été présentés en section 3.5.2.1. Ce modèle considère qu'en ligne directe, les paquets d'un véhicule voisin peuvent être reçus avec succès sur une distance de 500 m. Par contre, lorsqu'un ou plusieurs voisins se trouvent entre l'émetteur et le récepteur, ce dernier recevra avec succès les paquets jusqu'à une distance de 210 m. Ce modèle a été appliqué dans les simulations de CBL avec trois scénarios de mobilité différentes : S1, S2 et S3. Classés de densité faible à forte, ces scénarios permettent de mesurer l'impact du modèle de propagation d'onde dans des situations où les distances inter-véhiculaires varient.

Les résultats de simulation des métriques structurelles Ms1 à Ms7 et de routage Mr1 à Mr7, calculées sur l'intervalle de temps de 150 à 500 s et obtenues avec le modèle de propagation [9], sont reportés au tableau 4.9 avec une comparaison avec ceux réalisés précédemment avec le modèle de propagation en espace libre sous forme d'augmentation ou de décroissance exprimées en pourcentage.

Le tableau 4.9 reporte des résultats structurels similaires pour les scénarios S2 et S3. En effet, l'inter-distance entre les nœuds branche est de l'ordre de 200 m pour ces deux scénarios de mobilité (section 4.1.1.2) alors que le modèle de propagation [9] n'a un impact sur la réception d'un paquet qu'au-delà de 210 m de distance.

Pour le scénario S1, seules les métriques Ms1, Ms5, Ms7, Mr2 et Mr3 sont modifiées de manière significative (écart supérieur à 20%). Dans ce scénario les nœuds sont éloignés les uns des autres (voir figure 4.13), si un nœud feuille est présent entre deux nœuds branche, il empêche la communication. Ce nœud feuille sera alors transformé en branche faisant augmenter la métriques Ms1. Dans les scénarios S2 et S3, les nœuds ont plus de choix pour élire un nœud branche, limitant ainsi l'impact du modèle de propagation choisi. De plus, la métrique du nombre de nœuds branche par chaîne explique ce phénomène. Cette métrique est de 9,4 pour le scénario S1 et respectivement de 14,1 et 14,4 pour les scénarios S2 et S3.

4.4 Conclusion

Ce chapitre a eu pour objet l'étude structurelle des versions 1 et 2 de CBL et de la méthode des MPRs.

Les résultats sur la version 1 de CBL ont mis en évidence de plus fortes ruptures de chaînes qu'avec la version 2 conçue de sorte à supprimer certaines causes de ruptures. Les nœuds dans la version 1 restent à l'état branche durant plus de 50 s et la durée moyenne d'un lien entre une feuille et son nœud branche est de l'ordre de 15 s. Dans la version 2, la durée dans l'état branche diminue d'environ 10 s mais la durée moyenne d'un lien entre une feuille et son nœud branche augmente de plus de 50% ce qui assure une meilleure stabilité de la structure.

Les résultats structurels entre CBL version 2 et la méthode des MPRs ont mis en avant l'intérêt de la structure de la chaîne. Le nombre de nœuds relais MPRs constitue plus de 80% des nœuds du réseau alors que le pourcentage de nœuds branche ne dépasse pas 25% pour les scénarios de moyenne et forte densité et 70% pour les scénarios de faible densité. Pour ces scénarios de faible densité, CBL est en dessous de ses performances moyennes, mais reste plus efficace que la méthode des MPRs. Le nombre de nœuds relayant le trafic en mode *broadcast* ne dépasse jamais 50 nœuds pour CBL, même pour des réseaux de forte densité comptabilisant 369 nœuds et plus, alors qu'il dépasse les 300 nœuds pour la méthode des MPRs.

L'étude du trafic de routage généré par CBL-OLSR et OLSR sous OPNET Riverbed Modeler a montré la diminution du trafic des messages TC d'un facteur de 3 et 17 en fonction du scénario. Cette diminution est liée au faible nombre de nœuds relais de CBL (branche) par rapport à celui des nœuds MPRs qui génèrent ces messages TC et relaient le trafic en mode *broadcast*.

Pour finir, plusieurs études de sensibilité ont été réalisées. La première a exposé la faible sensibilité de CBL à ses paramètres intrinsèques de fonctionnement. La deuxième étude sur la mobilité des nœuds,

TABLEAU 4.9 – Valeurs moyennes des métriques de performances Ms1 à Ms7 et Mr1 à Mr7 calculées sur l'intervalle de temps de 150 à 500 s – Cas A, scénarios S1 à S3, CBL-OLSR, modèle de propagation OLOS

Réseau		R1		
Scénarios		1	2	3
Densité		-	+	++
Nombre de nœuds		25	102	198
Structure	Ms1 : Feuille/Réseau (%)	44,6	83,1	90,4
	<i>Diff</i>	+ 6,4	+1,6	+0,3
	Ms2 : NB_Chains	1,38	1,45	1,75
	<i>Diff</i>	-0,3	+0,12	+0,41
	Ms3 : Branche/Chaîne	9,82	12,2	11,1
	<i>Diff</i>	+0,42	-1,9	-3,3
	Ms5 : Feuille/Branche	0,793	4,98	9,62
<i>Diff</i>	-1,587	-1,19	-1,22	
Ms6 : Durée_Branche (s)	61,6	60,6	58,3	
<i>Diff</i>	+5,3	+6,6	+3	
Ms7 : Durée_rattachement_Branche (s)	34,9	36,5	35,6	
<i>Diff</i>	+5,5	-2,9	-7,3	
Routing	Mr1 : Charge_routage (kbit/s)	23,9	126	354
	<i>Diff</i>	-2,9	+0	-4
	Mr2 : HELLO_total (kbit/s)	10,6	93,6	309
	<i>Diff</i>	-0,1	+0,4	+0
	Mr3 : TC_total (kbit/s)	13,2	32,5	45,1
	<i>Diff</i>	-2,9	-0,7	-3,8
	Mr4 : HELLO/Charge_routage (%)	49,3	75	87,3
<i>Diff</i>	+5,5	+0,3	+1,2	
Mr5 : TC_total/Charge_routage (%)	50,7	25	12,7	
<i>Diff</i>	-5,5	-0,3	-1,2	
Mr6 : TC_relayé/TC_envoyé (%)	86,6	90,8	89,7	
<i>Diff</i>	-1,7	-0,3	-1,3	
Mr7 : Délai WLAN (ms)	0,112	0,145	0,195	
<i>Diff</i>	+0,001	+0	-0,001	

Où *Diff* est la différence en pourcentages des “valeurs de CBL version 2 avec le modèle de propagation OLOS moins celles de CBL version 2 sans le modèle de propagation OLOS”.

Définition des métriques en section 4.1.1.1 et 4.2.2.1.

réalisée sur un ensemble de scénarios de réseaux routiers, a révélé l'impact de la densité de véhicules sur la structure CBL. Nous avons montré qu'il était nécessaire d'avoir une densité minimale de véhicules pour former la chaîne : deux véhicules par zone de portée et par sens de circulation. Lorsque cette densité dépasse ce seuil minimal, les autres nœuds restent à l'état de feuille, ce qui améliore d'autant plus les performances du schéma de *clustering* CBL. La troisième et dernière étude de sensibilité sur le modèle de propagation a montré que ces modèles n'ont qu'un impact limité sur la structure formée par CBL. En effet, ils modifient en particulier la zone de portée et donc le seuil minimal de densité nécessaire à la construction de la chaîne de CBL.

Chapitre 5

Analyse système

Sommaire

5.1 Évaluations de CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP sous OPNET	167
5.1.1 Présentation des protocoles et spécification de leurs paramètres	168
5.1.2 Paramétrages de la simulation	170
5.1.3 Métriques de performance	171
5.1.4 Analyse des résultats – scénario applicatif Sa2	171
5.1.5 Analyse des résultats – scénarios applicatifs Sa1-Sa3	178
5.2 Application de localisation distribuée coopérative	180
5.2.1 Intérêt de la localisation pour les applications véhiculaires	180
5.2.2 Présentation de l’application de corrélation cartographique coopérative (CMM)	182
5.2.3 Présentation de la station de base dynamique (DDGPS)	183
5.2.4 Adaptation de l’application au schéma CBL et modélisation	184
5.2.5 Évaluation de performances par la simulation	185
5.3 Application de perception élargie	190
5.3.1 Présentation de l’application	190
5.3.2 Besoins en communication	191
5.3.3 Adaptation de l’application au schéma CBL et modélisation	192
5.3.4 Évaluation de performances par la simulation	192
5.4 Conclusion	194

Dans ce chapitre, trois évaluations système de CBL-OLSR sont menées sous OPNET Riverbed Modeler qui font appel à différents trafics de données applicatives. La première évaluation, section 5.1, compare les résultats de simulation du protocole CBL-OLSR avec ceux de cinq protocoles dont nous avons une modélisation sous OPNET et ceci dans le cas de trois scénarios applicatifs. En section 5.2, une analyse des applications d’ego-localisation décrites dans [11] et l’évaluation de leur mise en œuvre avec CBL-OLSR est réalisée. Enfin, en section 5.3, la troisième analyse système est menée avec un trafic applicatif conforme à ce qu’il pourrait être pour une application de perception étendue.

5.1 Évaluations des protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP sous OPNET

Dans cette section, nous présentons tout d’abord les cinq protocoles utilisés pour notre comparaison système avec CBL-OLSR. Quatre de ces protocoles ont déjà une modélisation développée par OPNET Riverbed Modeler. Ce sont deux protocoles réactifs, *Ad-hoc On Demand Distance Vector Protocol* (AODV) [44] et *Dynamic Source Routing* (DSR) [193], et deux protocoles proactifs, *Optimized Link State Routing protocol* (OLSR) [31] (section 2.5.1) et *Geographic Routing Protocol* (GRP) [53] (section 1.3.1.1). Par ailleurs, la modélisation sous OPNET du protocole *QoS - Optimized Link State Routing protocol* (QOLSR) de l’article [114], notre cinquième protocole de comparaison, a été mise à notre disposition

par son auteur. Nous expliquons alors les scénarios de communication créés pour cette évaluation comparative, puis les métriques applicatives de performance avant de produire notre analyse des résultats obtenus.

5.1.1 Présentation des protocoles et spécification de leurs paramétrages

5.1.1.1 AODV

Le protocole AODV [44] est un protocole réactif. Ses paramètres par défaut sont repris dans le tableau 5.1.

Les nœuds de communication AODV envoient des messages HELLO afin d'effectuer une maintenance des routes actives à une fréquence définie par le paramètre *Hello interval*. Si aucune route n'est active, aucun message HELLO n'est envoyé (pas de détection du voisinage). Lorsqu'un nœud n'a pas reçu de message HELLO d'un nœud voisin durant une période supérieure à *Hello loss*, il supprime ce dernier de sa table de routage. Le système de recherche de route par l'envoi de paquets RREQ/RREP est utilisé (section 1.2.4.2). Une optimisation est effectuée quant à l'envoi du paquet de recherche de route RREQ. Le paquet est d'abord envoyé au voisinage proche avec un TTL (section 2.5.1) fixé par le paramètre *TTL start*. Si le nœud source ne reçoit pas de réponse après une durée *RingTraversalTime*, il incrémente la valeur du TTL de *TTL increment* (dans la limite du seuil maximal de *TTL threshold*, sinon, en cas de dépassement de ce seuil, le TTL est fixé par le paramètre *Net diameter*), puis il renvoie un paquet de recherche de route. La durée *RingTraversalTime* est définie par l'équation 5.1 :

$$RingTraversalTime = 2 * NodeTraversalTime * (TTL + Timeoutbuffer) \quad (5.1)$$

où *NodeTraversalTime* est la durée de traitement du paquet et *Timeoutbuffer* un temporisateur qui permet de prendre en compte le temps supplémentaire en cas de congestion du médium de communication.

Une limite d'envoi du même paquet de recherche de route est fixée par le paramètre *Route request retries*. Pour éviter la saturation du médium, le débit maximum des paquets de recherche de routes est limité par le paramètre *Route error rate limit*. Lorsqu'un nœud est un relais intermédiaire d'une route, il enregistre la route trouvée vers le destinataire dans une table de routage. Cet enregistrement est supprimé au bout d'une durée de route inactive *Active route timeout*. Enfin, si l'option *Local repair* est active, un nœud intermédiaire se rendant compte que la liaison avec le nœud intermédiaire suivant est rompue peut envoyer des paquets de recherche de route pour retrouver une route alternative. Si cette option n'est pas activée, le nœud intermédiaire envoie un paquet au nœud source pour le prévenir de la rupture de la liaison et dans ce cas, le nœud source cherchera une route alternative.

TABLEAU 5.1 – Paramètres par défaut du protocole AODV [44]

Attributs	Valeurs	Attributs	Valeurs
Route request retries	5	TTL start	1
Route request rate limit	10 paquets/s	TTL increment	2
Active route timeout	3 s	TTL threshold	7
Hello interval	uniform(1 ; 1,1 s)	Local repair	active
Packet queue size	infinity	Hello loss	2 s
Net diameter	35 sauts	Addressing mode	IPV4
Node traversal time	0,04 s	Timeout buffer	2
Route error rate limit	10 paquets/s		

5.1.1.2 DSR

Le protocole DSR [44] est un protocole de routage réactif. Son paramétrage par défaut est présenté dans le tableau 5.2. Le système de recherche de route est très proche de celui de AODV, excepté le fait que chaque nœud intermédiaire relayant un paquet de recherche de route RREQ l'enrichit de son identifiant. Avec cette liste, le nœud destinataire connaît l'ensemble des nœuds relais intermédiaires par lesquels le paquet RREQ est passé pour l'atteindre. Le protocole DSR inclut cette liste dans le paquet RREP à

destination du nœud source. Comme AODV, il offre également une option de réparation de route locale et d'envoi de paquets RREQ limité au voisinage proche. Ces deux options sont désactivées par défaut.

TABLEAU 5.2 – Paramètres par défaut du protocole DSR [193]

Attributs	Valeur	Attributs	Valeur
Découverte de routes :		Maintenance de routes :	
Request table size	64 nœuds	Max buffer size	50 paquets
Req. table identifiens	16	Hold off time	0,25 s
Req. retransmission.	16	Maint. Retransmis.	2
Max. Req. period	10 s	Ack. timer	0,5 s
Non-propagating Req.	désactivé	Send Buffer :	
GratuitousReply timer	1 s	Max. Buffer size	infini
Enregistrement de routes :		Buffer timeout	30 s
Max Cached routes	infinity		
Route Cache timeout	300 s	Packet salvaging	active
Route Cache export	désactivé	Broadcast jitter	uniform(0; 0,01) s

5.1.1.3 OLSR

Le protocole proactif OLSR [31] ayant déjà été décrit en section 2.5.1, nous ne rappelons ici que les paramètres du protocole utilisés par défaut au moyen du tableau 5.3.

Attribut	Valeur par défaut
Willingness	défaut
Hello interval	2 s
TC interval	5 s
Neighbor hold time	6 s (3*Hello interval)
Topology hold time	15 s (3*TC interval)
Duplicate message hold time	30 s
Adressing	IPV4

TABLEAU 5.3 – Paramètres par défaut du protocole OLSR [31]

5.1.1.4 QOLSR

Le protocole QOLSR [114] est une adaptation de OLSR qui prend en compte la qualité des liaisons entre les nœuds du réseau à l'aide des métriques de délais, débit et pertes de paquets. Chaque nœud ajoute, dans les messages HELLO envoyés par le protocole OLSR, la date courante, qui sert aux calculs des métriques de délai et de débit, et sa position géographique, qui intervient dans la détermination du taux de perte de paquets qui est pour QOLSR fonction de la distance entre les nœuds. Ces trois métriques à QoS entrent en compte dans le choix des nœuds MPR : les nœuds qui maximise la fonction de score de [114] définie comme une moyenne pondérée du délai, du débit, du taux de perte des paquets et du *Willingness* sont prioritairement choisis comme MPR. Les autres paramètres de ce protocole sont identiques à ceux d'OLSR.

5.1.1.5 GRP

Le protocole proactif géographique GRP [53], déjà évoqué en section 1.3.1.1, utilise la méthode de routage gloutonne en association avec une division hiérarchique du réseau en zones rectangulaires. Ses paramètres utilisés par défaut sont rappelés dans le tableau 5.4.

Pour ce protocole GRP [53], les tables de routage des nœuds d'une même zone comportent la position exacte des nœuds de cette zone. Pour les nœuds voisins situés hors de cette zone, seul le numéro de la zone du nœud de niveau hiérarchique supérieur est enregistré dans ces tables de routage. Ainsi, si un

TABLEAU 5.4 – Paramètres par défaut du protocole GRP [53]

Attributs	Valeur	Attributs	Valeur
Hello interval	5 s	Backtrack	activé
Neighbor expiry time	10 s	Route export	désactivé
Distance moved	1000 m	Number of initial floods	1
Position request timer	5 s	Quadrant size	2 km

nœud de la zone Aa1 a des voisins en zones Aa2, Aa3 et Aa4, le nœud de la zone Aa1 mémorise dans sa table de routage le fait que chacun de ses voisins est situé en zone Aa.

Les paquets HELLO transmis contiennent la position des nœuds. Pour limiter le trafic de routage des paquets HELLO, un nœud n'envoie sa nouvelle position que dans l'un des cas suivants : 1) il a parcouru une certaine distance (paramètre *Distance moved*), 2) il a changé de zone, 3) il n'a plus envoyé sa position depuis une certaine durée définie par un seuil temporel (paramètre *HELLO interval*). Selon le cas, la portée géographique de la transmission de ces paquets est ainsi limitée. Dans les cas 1) et 2), l'information est propagée aux nœuds de la zone de niveau hiérarchique supérieur commune à l'ancienne et à la nouvelle zone ; dans le cas 3), sa portée est limitée aux voisins immédiats (à un saut). La suppression d'un nœud des tables de routage des nœuds d'une zone est par ailleurs réalisée lorsqu'un nœud voisin n'a plus émis de paquets durant une période (paramètre *Neighbour expired time*).

Au moment de l'envoi d'un paquet applicatif, lorsque le nœud destinataire est connu, dans la table de routage du nœud émetteur, comme faisant partie de la même zone (Ba1 par exemple), alors la méthode de routage gloutonne est utilisée. S'il est situé dans une autre zone que celle du nœud émetteur, alors ce dernier transmet le paquet à un voisin de la zone de niveau hiérarchique supérieur (Ba dans cet exemple). Ce nœud voisin sert alors de serveur de localisation. Lorsque le paquet atteint un optimum local, il est renvoyé au nœud intermédiaire précédent si l'option *Backtrack* est active. Le nœud cherche alors un autre nœud du réseau pour atteindre la destination et s'il n'en trouve pas, il retourne le paquet à l'expéditeur.

5.1.2 Définition du trafic applicatif, des scénarios de communication et du paramétrage de OPNET Riverbed Modeler

Notre objectif est de comparer notre proposition CBL-OLSR aux protocoles de routage mentionnés ci-dessus. Nous nous intéressons en particulier aux applications de sécurité (section 3.7) relatives aux véhicules autonomes qui sont un véritable challenge pour les VANETs en termes de délais et de bande passante. Dans ce type d'applications de sécurité, les fonctions embarquées (capteurs, géolocalisation, perception élargie, etc.) nécessitent l'envoi périodique de variables (telles la vitesse, l'accélération et la position) pour leur fonctionnement. L'application présentée ici est conforme aux préconisations de l'IEEE pour les communications en environnement véhiculaire [10]. Elle se décline ainsi : des paquets de 300 octets sont envoyés à une fréquence de 10 Hz.

À partir de cette définition applicative, nous avons défini trois scénarios d'évaluation (Sa1, Sa2 et Sa3) qui utilisent tous les trois le réseau routier et la densité de véhicules précédemment définis pour le scénario S5 (tableau 3.6). Sa1, Sa2 et Sa3 se distinguent par le nombre de nœuds concernés par l'envoi et la réception du trafic applicatif en mode de communication *multicast* (section 1.2.3). Ces nœuds sont choisis aléatoirement une fois avant l'exécution d'un scénario d'évaluation donné. Ainsi, pour un scénario, les simulations des six protocoles sont réalisés sur le même panel de nœuds, émetteurs et récepteurs. Dans le scénario Sa1, 10 nœuds source sont choisis correspondant à un taux de pénétration de 5% des véhicules communicants sur un total de 205 nœuds ; dans le scénario Sa2, 15 nœuds sont choisis soit un taux de pénétration de 7,5% ; et enfin dans le scénario Sa3, 20 nœuds soit un taux de pénétration de 10%. Chaque scénario a été exécuté 10 fois avec des valeurs différentes de *seed* pour l'initialisation du générateur de nombre aléatoire du logiciel OPNET Riverbed Modeler pour éviter qu'une séquence particulière de nombres aléatoires ne favorise un des protocoles.

Les simulations de cette section commencent à la date $t'=200$ s pour que le réseau soit complètement chargé en véhicules. Nous avons déclenché l'émission du trafic applicatif 30 s après l'initialisation du réseau (soit à $t'=230$ s) pour laisser le temps aux nœuds d'effectuer la découverte de leur voisinage. Puis,

le trafic applicatif est stoppé 30 s après son déclenchement (soit à la date $t'=260$ s). Dans la suite de la section 5.1, nous fixerons comme référence temporelle $t=0$ à la date de $t'=200$ s de manière à ce que l'activation du trafic applicatif intervienne sur les courbes de résultats à $t=30$ s.

Nous gardons pour ces simulations les valeurs des paramètres d'OPNET Riverbed Modeler telles que définies dans les tableaux 3.11 et 3.9.

5.1.3 Métriques de performance

Les métriques de performance applicatives (Ma - Métriques applicatives) utilisées pour cette évaluation sont les suivantes :

- Ma1 : Charge du réseau, le débit en bit/s des messages envoyés sur le réseau ;
- Ma2 : Débit du réseau, le débit en bit/s des messages reçus sur le réseau ;
- Ma3 : Trafic de routage envoyé, le débit en bit/s des messages de routage envoyés et retransmis sur le réseau ;
- Ma4 : Trafic de routage reçu, le débit en bit/s des messages de routage reçus sur le réseau ;
- Ma5 : Retransmission, le taux de retransmission des messages ;
- Ma6 : Nb_saut, le nombre de sauts effectués par un message (nombre de nœuds relais intermédiaires) ;
- Ma7 : Délai WLAN, le délai de transmission de bout-en-bout d'une trame en ms ;
- Ma8 : Délai d'accès au médium en ms.

Ces métriques sont communes à l'étude des protocoles de routage considérés dans ce chapitre. Pour la section 5.1, la valeur retenue pour chaque métrique est la moyenne des valeurs de la métrique obtenues pour chacune des 10 simulations réalisée avec un *seed* différent.

5.1.4 Analyse des résultats – scénario applicatif Sa2

Après activation du trafic applicatif, à $t=30$ s, la charge du réseau (Ma1) augmente de manière importante pour tous les protocoles de routage (figure 5.1). Cette charge de trafic inclut les messages de routage et d'application envoyés par les 15 nœuds source du scénario Sa2 ainsi que les messages retransmis par les nœuds relais. Durant l'envoi des messages applicatifs (entre $t=30$ s et $t=60$ s), le protocole OLSR a la charge la plus faible avec une moyenne de 4,8 Mbit/s (tableau G.2 en annexe G), puis viennent les protocoles AODV et CBL-OLSR avec une moyenne respective de 8 et 8,2 Mbit/s. Ensuite vient le protocole GRP avec une charge moyenne de 9,9 Mbit/s. Enfin, les deux protocoles QOLSR et DSR présentent la charge la plus élevée avec des moyennes respectivement de 15 Mbit/s et 18 Mbit/s. Notons que la charge du réseau est supérieure pour les protocoles QOLSR et DSR à la limite de la bande passante fixée dans les simulations (12 Mbit/s). En effet, cette limite n'est valable que localement à une zone de portée donnée, ici de 500 m, et non à l'échelle du réseau routier simulé de 5 km. Pour un même trafic applicatif, les protocoles de routage n'ont pas la même gestion de la charge du réseau, passant du simple au triple. Notre proposition CBL-OLSR se classe en troisième position par rapport à la charge du réseau pour le scénario Sa2.

Le profil de l'évolution de la charge du réseau au cours du temps de simulation est également intéressant à étudier. Les protocoles proactifs OLSR et CBL-OLSR présentent une augmentation de la charge entre $t=30$ s et $t=35$ s due au délai d'accès au médium qui augmente lorsque l'application est activée (métrique Ma8, figure 5.9). Puis nous observons une petite diminution et une stabilisation entre $t=35$ s et $t=60$ s. Les protocoles QOLSR et GRP ont une charge très stable durant l'envoi des messages applicatifs. La charge des deux protocoles réactifs AODV et DSR est moins stable étant donné l'utilisation des requêtes RREQ/RREP. Pour le protocole DSR, on remarque un délai de 5 s à 10 s avant stabilisation de la charge à sa valeur moyenne. Ce délai est lié à la recherche de route réactive. Le système TTL du protocole AODV permet de limiter ce délai. De plus, pour ce protocole, le pic de charge maximale lors de l'envoi des requêtes RREQ/RREP est limité à l'aide du paramètre *Route request rate limit*. Après l'arrêt de l'envoi des messages applicatifs, la charge du réseau devrait être égale au trafic de routage envoyé et ainsi revenir à l'état avant $t=30$ s. En regardant conjointement la figure 5.1 et 5.3, nous observons que c'est le cas pour tous les protocoles proactifs (CBL-OLSR, OLSR, QOLSR et GRP). Ce n'est pas le cas pour les

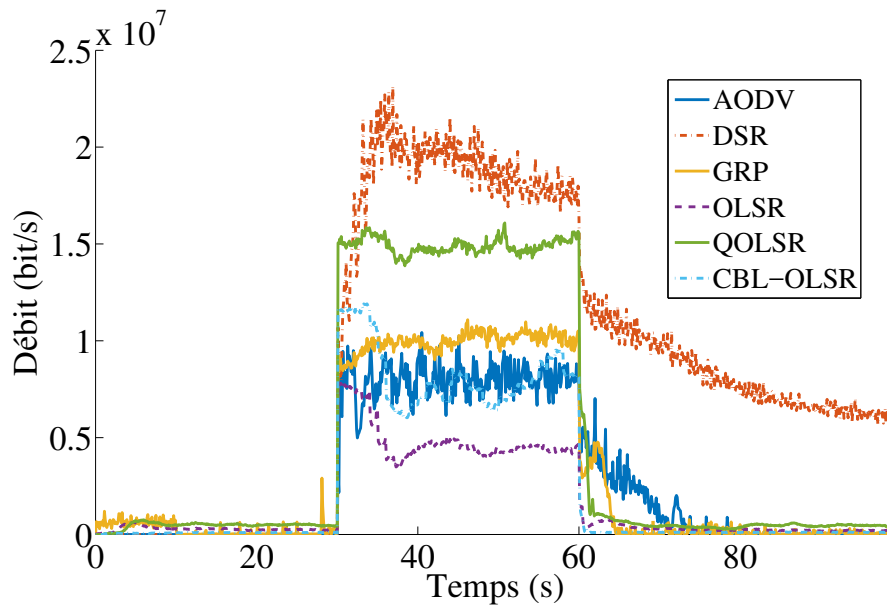


FIGURE 5.1 – Charge du réseau (Ma1) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

deux protocoles réactifs AODV et DSR. Cela signifie qu'ils n'ont pas encore envoyé la totalité du trafic applicatif, des paquets sont encore présents dans les files d'attente. Ceci suggère, comme nous le verrons par la suite, des délais d'envoi plus longs (figure 5.7 et 5.8) et des paquets qui subissent davantage de retransmissions avant d'atteindre les nœuds destinataire (figure 5.5).

Après $t=30$ s, le débit du réseau (Ms2) augmente fortement pour tous les protocoles de routage (figure 5.2). Ce débit inclut les messages de routage et d'application reçus par les 15 nœuds source ainsi que les messages reçus par les nœuds relais. Entre $t=30$ s et $t=60$ s, les protocoles DSR, OLSR et GRP montrent les débits les plus faibles avec respectivement des moyennes de 4,5 Mbit/s, 4,6 Mbit/s et 4,9 Mbit/s, puis vient notre proposition CBL-OLSR avec une moyenne de 8,8 Mbit/s et enfin les protocoles AODV et QOLSR affichant respectivement des débits de 10 Mbit/s et 15 Mbit/s. Nous observons que le profil de débit dans le réseau est le même que celui de la charge pour les protocoles QOLSR, OLSR et CBL-OLSR, ce qui signifie que peu de paquets sont perdus par ces protocoles. Ceci n'est pas le cas des protocoles GRP, AODV et DSR qui ont un débit bien plus faible que leur charge.

Les figures 5.3 et 5.4 présentent respectivement le trafic de routage envoyé et reçu (Ma3). Ces deux figures sont similaires. Les protocoles proactifs OLSR, QOLSR, CBL-OLSR et GRP émettent un trafic de routage avant l'activation de l'application. Ce trafic périodique correspond à l'envoi des messages HELLO de découverte du voisinage et pour OLSR, QOLSR, CBL-OLSR des messages TC permettant le calcul des routes enregistrées dans les tables de routage. Le trafic de routage envoyé par le protocole GRP est constitué exclusivement des messages HELLO et n'est pas périodique. Des pics d'utilisation de la bande passante sont observés (par exemple à $t=28$ s avec une charge de routage de 3 Mbit/s). Ce trafic correspond au passage des nœuds dans une autre zone géographique dont la taille a été fixée à 1 km. Les deux autres protocoles de routage réactifs n'envoient pas de trafic de routage avant l'activation de l'application. Le trafic de routage est en moyenne de 0,14 Mbit/s pour OLSR, 0,07 Mbit/s pour CBL-OLSR, 0,19 Mbit/s pour GRP et 0,25 Mbit/s pour QOLSR et nul pour AODV et DSR. Ce trafic de routage avant l'activation de l'application utilise peu de ressources de communication du réseau.

À $t=30$ s, le trafic de routage pour les protocoles de routages proactifs reste stable, voire diminue, car les messages applicatifs sont prioritaires sur les messages de routage (catégorie d'accès AC1). Le trafic de routage augmente de manière très forte pour les protocoles réactifs AODV et DSR. Ce trafic soudain correspond à l'envoi des requêtes RREQ/RREP permettant de trouver une route vers les nœuds destinataire pour les messages applicatifs. Le trafic de routage du protocole AODV se stabilise durant cette période d'activation de l'application à une moyenne de 1,6 Mbit/s. Celui du protocole DSR est de 2,4 Mbit/s en moyenne. Les protocoles réactifs utilisent pour leur fonctionnement les ressources de communication du réseau au détriment du trafic applicatif au moment même où ce trafic doit être envoyé.

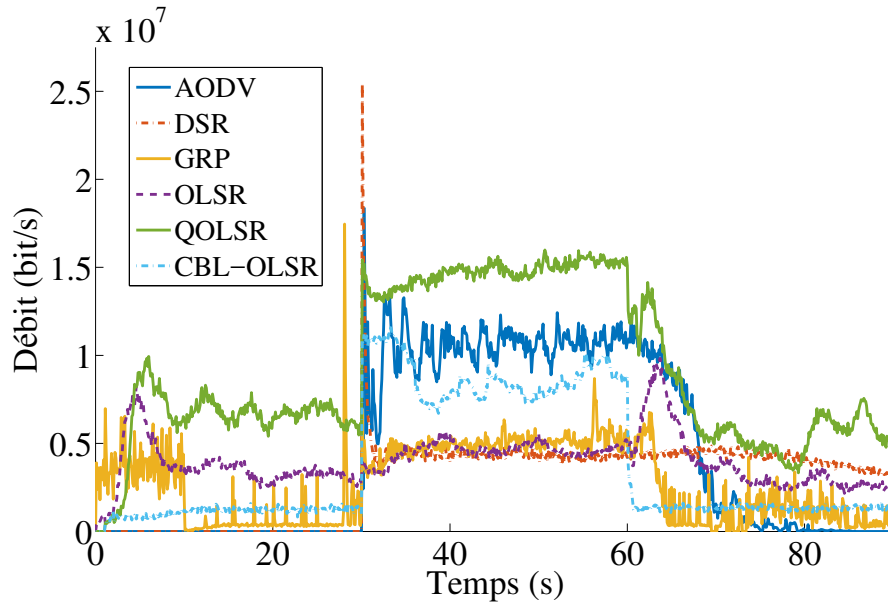


FIGURE 5.2 – Débit du réseau (Ma2) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

Après $t=60$ s, on remarque que la charge de routage du protocole DSR ne diminue pas, alors même que l'application est désactivée car ce protocole ne parvient pas à trouver l'ensemble des routes vers les nœuds destinataire et délivrer l'ensemble des paquets applicatifs. Le trafic de routage du protocole AODV diminue après l'arrêt de l'application, même s'il redevient nul qu'à partir de $t=70$ s, car des messages applicatifs n'ont pas pu être transmis avant. Les profils du trafic de routage des protocoles proactifs reviennent au niveau de ceux avant l'activation de l'application.

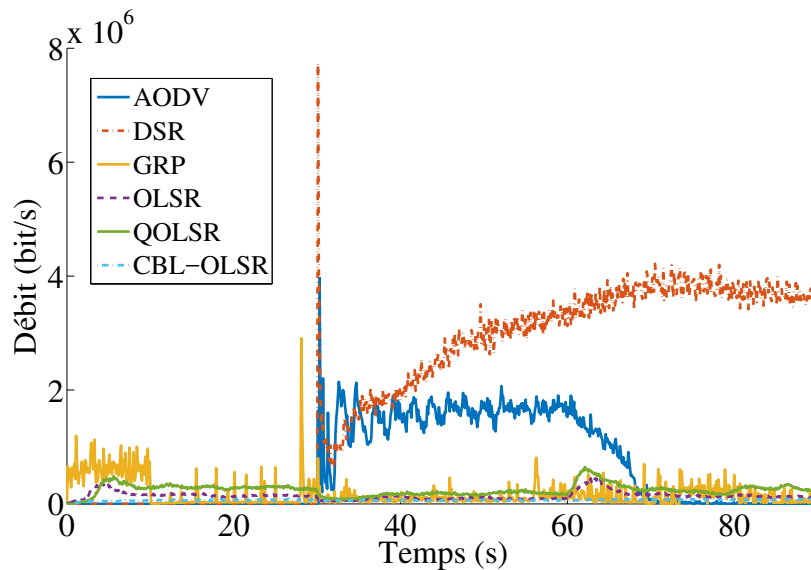


FIGURE 5.3 – Trafic de routage envoyé (Ma3) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

Le taux de retransmission de paquets (Ma5) est nul avant l'activation de l'application pour tous les protocoles (figure 5.4) sauf pour GRP où l'on note qu'un paquet de routage a été retransmis une fois (à $t=28$ s). Durant l'activation de l'application, le taux de retransmission de paquet est stable pour tous les protocoles. Il est en moyenne de 0,58 pour CBL-OLSR, de 0,69 pour QOLSR, 1,8 pour DSR et 2, 2,1, 2,4 respectivement pour AODV, OLSR et GRP. Notre proposition CBL-OLSR montre donc le plus faible taux de retransmissions, ce qui contribue à limiter le taux de pertes.

La figure 5.6 présente le nombre de sauts (Ma6) qui est nul avant l'activation de l'application pour tous

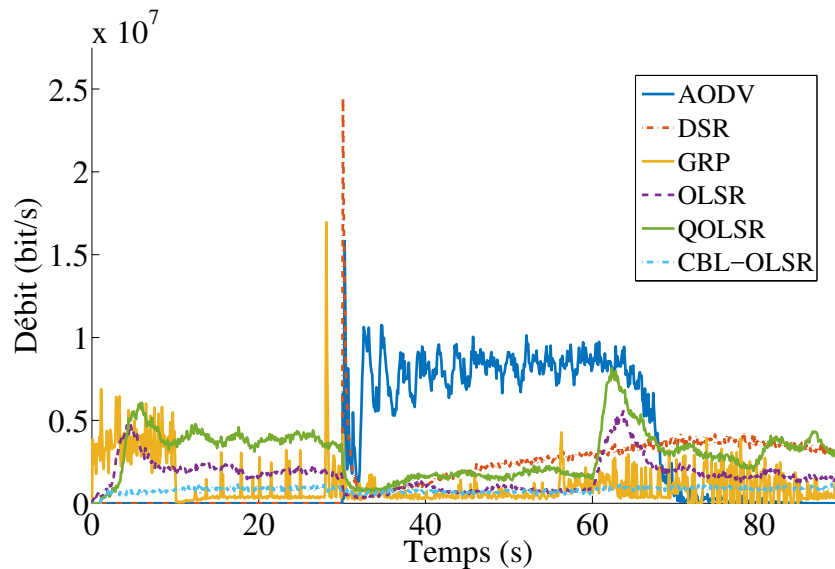


FIGURE 5.4 – Trafic de routage reçu (Ma4) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

les protocoles de routage sauf pour GRP qui est égal à un car les messages HELLO sont retransmis à tous les nœuds d'une zone hiérarchique paramétrée à 1 km^2 . Durant l'activation de l'application ce nombre de sauts est stable pour chaque protocole. Il est en moyenne de 1,8 (AODV), 2,9 (DSR), 1,7 (GRP), 1,8 (OLSR), 3 (QOLSR) et 2,3 (CBL-OLSR). Le nombre de sauts permet la mesure des performances du protocole de routage concernant la route trouvée entre le nœud source et le nœud destinataire. Notons que le nombre de sauts avec notre proposition CBL-OLSR est légèrement supérieur à celui des protocoles AODV, GRP et OLSR. En effet, notre proposition contraint les messages à utiliser la chaîne de nœuds branche. Cependant, comme nous l'avons vu, cela ne pénalise pas les taux de retransmission de paquets. Les protocoles classiques de la littérature cherchent à optimiser le nombre de sauts des paquets applicatifs, ce qui mène à sélectionner des liaisons entre des nœuds intermédiaires très éloignés, presque en limite de portée de communication. Ces liaisons ne sont pas de bonne qualité et induisent des pertes de paquets lorsque la route trouvée devient caduque [114].

Remarquons qu'à la toute fin de l'activation de l'application, le nombre de sauts augmente soudainement avant de revenir à sa valeur initiale pour les protocoles OLSR, QOLSR, CBL-OLSR et GRP en raison du délai de réception des derniers paquets applicatifs transmis à $t=60 \text{ s}$. En effet, les nœuds destinataire à portée de transmission (donc à 1 saut) reçoivent directement les paquets. Les paquets sont ensuite relayés une première fois. À partir de cet instant, la statistique du nombre de sauts n'enregistre plus de valeur de saut égale à 1 car tous les nœuds destinataire à portée de communication du nœud source ont reçu les paquets. Les nœuds destinataire situés à 2 sauts du nœud source reçoivent leur paquet, puis les paquets sont relayés une deuxième fois. À partir de cet instant, la statistique du nombre de sauts n'enregistre plus de valeur de saut égale à 2 car tous les nœuds destinataire à deux sauts du nœud source ont reçu les paquets. Et ce phénomène se répète jusqu'à ce que tous les nœuds destinataire aient reçu leur paquet. La statistique ne fait donc qu'augmenter jusqu'à ce que tous les nœuds destinataire aient reçu leur paquet. Elle redevient nulle ensuite.

Les figures 5.7 et 5.8 présentent respectivement le délai WLAN (Ma7) et sa courbe cumulée. Ce délai est très faible pour notre proposition CBL-OLSR (en moyenne inférieur à $0,11 \text{ s}$). Nous remarquons, au début de l'activation de l'application entre $t=30 \text{ s}$ et $t=40 \text{ s}$, une hausse du délai correspondant à une augmentation du délai d'accès au médium (voir figure 5.9). Cette augmentation est une conséquence de l'algorithme CSMA/CA exécuté lors des tentatives d'accès au médium des nœuds de technologie IEEE 802.11p. À $t=30 \text{ s}$, l'activation simultanée du trafic applicatif pour l'ensemble des 15 nœuds source de Sa2 génère des conflits d'accès induisant des collisions. L'algorithme d'évitement de collision se déclenche alors localement au nœud, et chacun va différer ses tentatives de transmission selon son propre délai de *backoff* aléatoirement choisi. Lorsque les premiers paquets ont été envoyés un "ordre d'envoi" a été trouvé et a convergé vers l'optimum individuel entre les nœuds source et ainsi le délai d'accès au

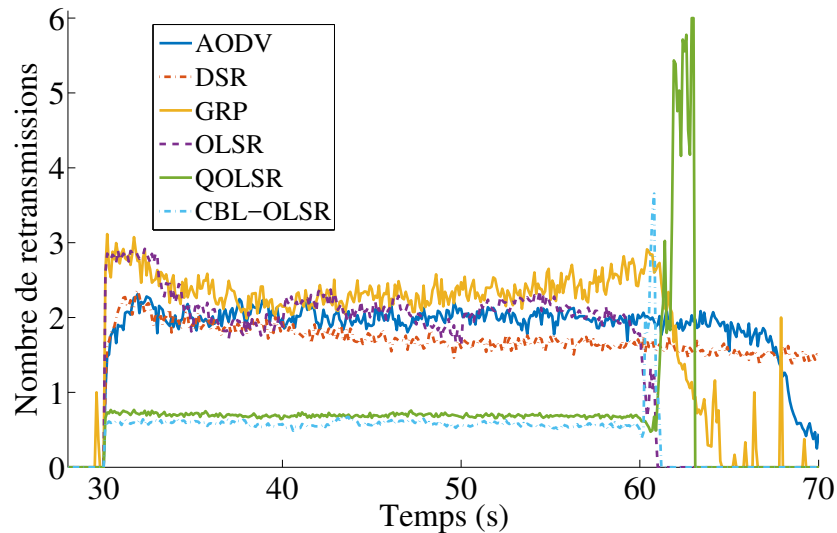


FIGURE 5.5 – Taux de retransmission (Ma5) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE entre $t=30$ s et $t=60$ s

médium diminue. Les protocoles QOLSR et OLSR montrent des moyennes de délai WLAN respectives de 0,23 s et 0,27 s. Puis les protocoles GRP et AODV ont des moyennes respectives de 0,49 s et 0,68 s. Le protocole DSR a un délai WLAN moyen de 2,9 s qui ne fait qu'augmenter (pas de stabilisation), ce qui montre ses faibles performances.

La courbe de la figure 5.8 récapitule les performances des différents protocoles. La valeur à 95% correspond au 0,95 quantile. Notre proposition CBL-OLSR montre un délai inférieur à 0,16 ms 95% du temps de la simulation tandis que QOLSR, le deuxième protocole montrant les meilleures performances en termes de délai, a un délai inférieur à 0,43 ms 95% du temps de la simulation. L'amélioration apportée par le protocole QOLSR prenant en compte des délais de réception des messages et choisissant des nœuds MPRs en fonction de ce critère est perceptible sur la métrique Ma7.

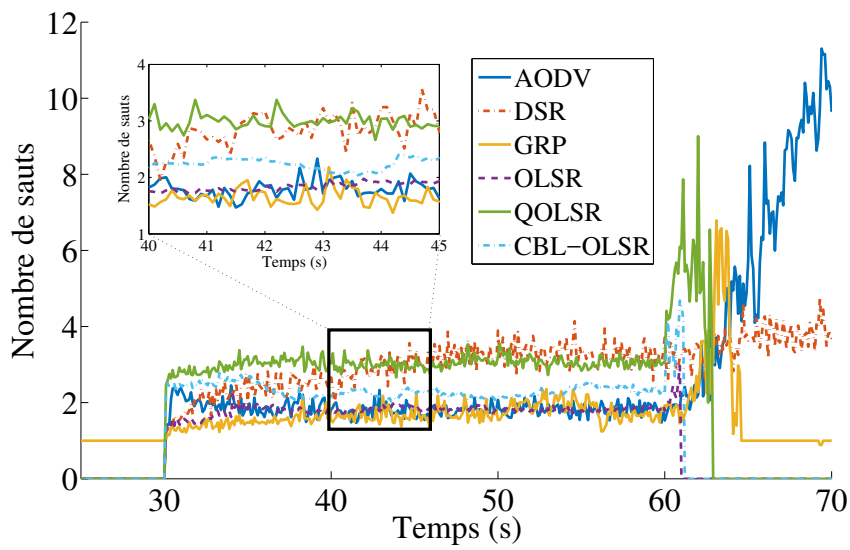


FIGURE 5.6 – Nombre de sauts (Ma6) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

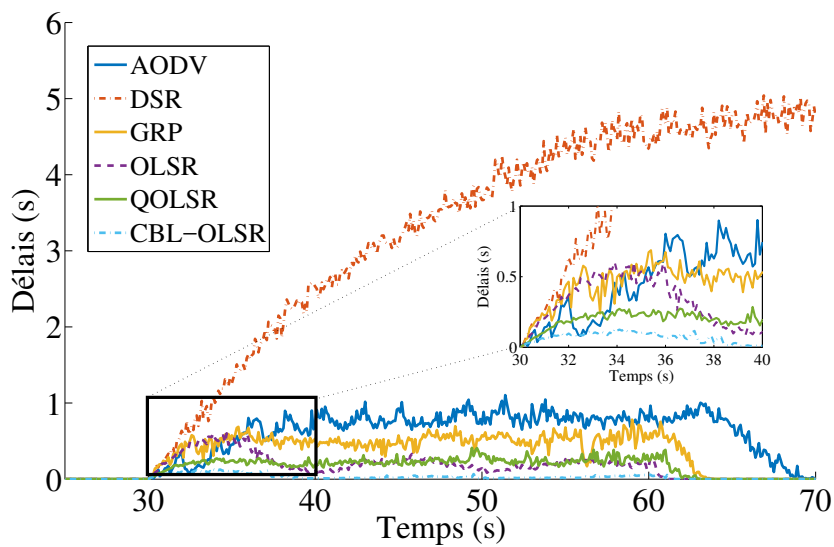


FIGURE 5.7 – Délai WLAN (Ma7) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

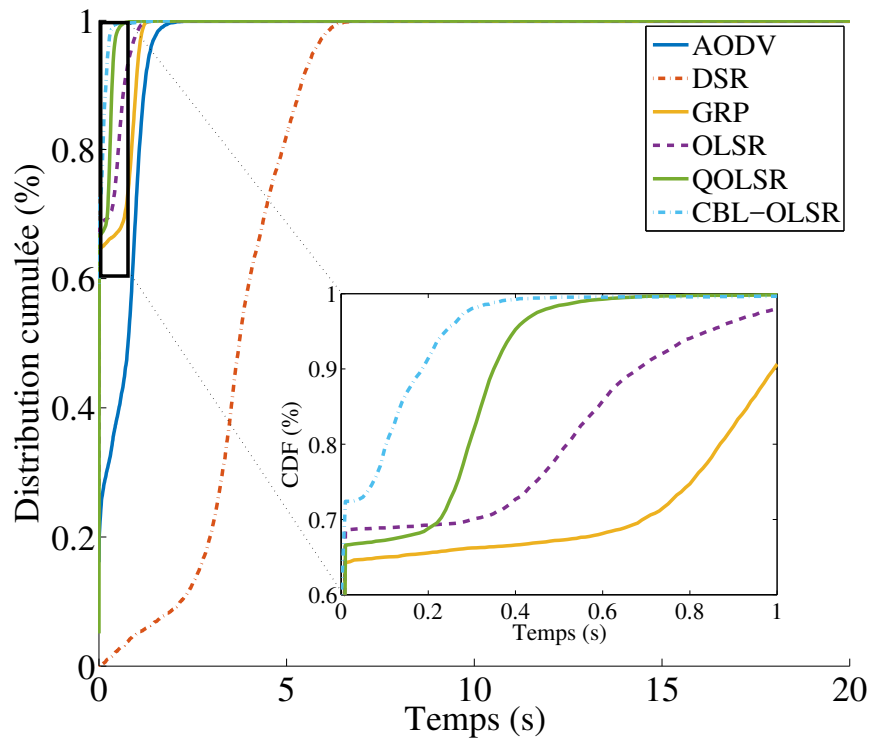


FIGURE 5.8 – Courbe cumulée du délai WLAN (Ma7) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

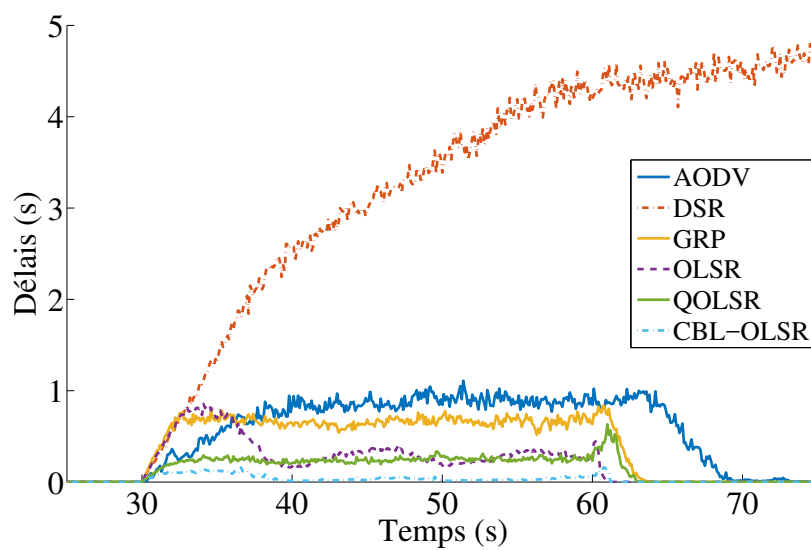


FIGURE 5.9 – Délai d'accès au médium (Ma8) dans le cas du scénario Sa2 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

5.1.5 Analyse des résultats – scénarios applicatifs Sa1-Sa3

Les valeurs minimales, maximales et moyennes des métriques Ma1 à Ma8 obtenues pour les scénarios Sa1, Sa2, et Sa3 sont présentées dans les tableaux G.1, G.2 et G.3 en annexe G.

Les figures 5.10a et 5.10b présentent la moyenne globale de la charge et du débit au cours de l'échange en fonction du scénario pour chacun des 6 protocoles. On remarque que la charge du réseau augmente en fonction du nombre de nœuds source de chaque scénario. Cette augmentation est quasiment linéaire : la charge du scénario Sa3 (20 nœuds source) a quasiment doublé par rapport à la charge du scénario Sa1 (10 nœuds source) pour tous les protocoles. Contrairement à la charge, le débit augmente en fonction du scénario pour les protocoles QOLSR, CBL-OLSR, AODV et DSR. Cependant cette augmentation n'est pas linéaire en fonction du nombre de nœuds source. Pour les protocoles OLSR et GRP le débit reste le même quel que soit le nombre de nœuds source envoyant du trafic applicatif.

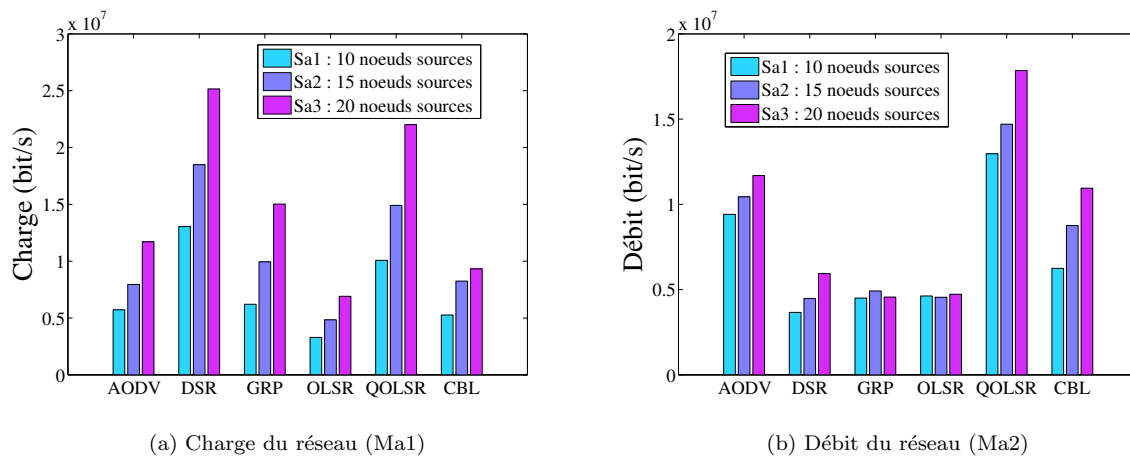


FIGURE 5.10 – Moyenne de Ma1 et Ma2 entre t=30 s et t=60 s dans le cas des scénarios Sa1 à Sa3 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

Les figure 5.11a et 5.11b présentent les trafics de routage envoyés (Ma3) et reçus (Ma4) au cours de l'échange applicatif en fonction du scénario pour chacun des 6 protocoles. On remarque que le trafic de routage augmente pour les protocoles réactifs en fonction du scénario, ce qui montre un impact direct du nombre de nœuds source. En ce qui concerne les protocoles de routage proactifs, le trafic de routage envoyé reste stable.

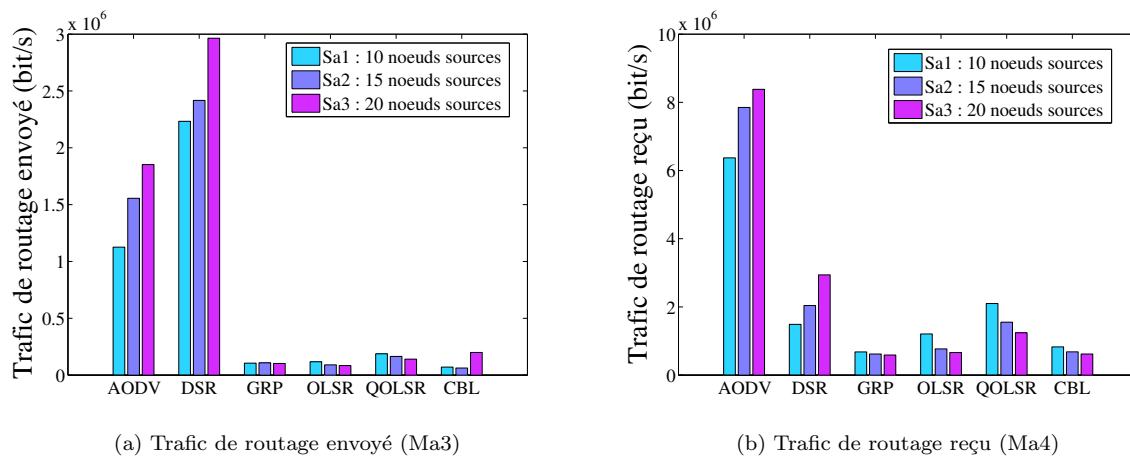


FIGURE 5.11 – Moyenne de Ma3 et Ma4 entre t=30 s et t=60 s dans le cas des scénarios Sa1 à Sa3 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

La figure 5.12a présente le taux de retransmission des messages (Ma5) au cours de l'échange applicatif en fonction du scénario pour chacun des 6 protocoles. Ce taux est globalement stable en fonction des scénarios. Les protocoles QOLSR et CBL-OLSR montrent les meilleures performances avec un taux de retransmission d'environ 0,5, les autres protocoles ont tous un taux supérieur à 1,5.

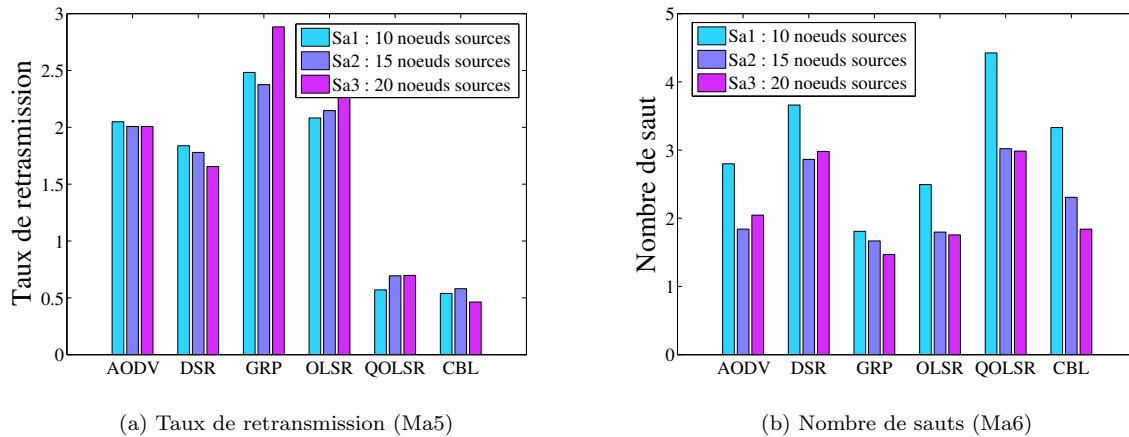


FIGURE 5.12 – Moyenne de Ma5 et Ma6 entre $t=30$ s et $t=60$ s dans le cas des scénarios Sa1 à Sa3 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

La figure 5.13b présente le nombre de sauts des messages (Ma6) au cours de l'échange applicatif en fonction du scénario pour chacun des 6 protocoles. Cette métrique dépend fortement du résultat du choix aléatoire des nœuds source et destinataire des messages applicatifs. Pour le scénario comportant 10 nœuds source, les nœuds sont plus éloignés les uns des autres ce qui explique que la métrique Ma6 soit la plus élevée des trois scénarios pour chacun des 6 protocoles de routage. La configuration de distance entre les nœuds source pour les scénarios Sa2 et Sa3 est proche puisque la métrique Ma6 est quasiment identique dans ces deux scénarios.

La figure 5.13a présente le délai WLAN (Ma7) au cours de l'échange applicatif en fonction du scénario pour chacun des 6 protocoles. Le délai WLAN augmente avec le nombre de nœuds source pour tous les protocoles. En effet plus le nombre de nœuds source augmente, plus les ressources de communication sont demandées et plus les délais WLAN et d'accès au médium augmentent.

Une analyse identique peut être faite sur la figure 5.13a qui présente le délai d'accès au médium (Ma8) au cours de l'échange applicatif en fonction du scénario pour chacun des 6 protocoles. Le délai WLAN est composé du délai d'accès au médium, d'envoi des trames MAC à la couche physique, de propagation des signaux, de réception des signaux et d'envoi et de reconstruction des trames MAC au niveau du nœud récepteur. Nous remarquons que le délai WLAN est composé quasi exclusivement du délai d'accès au médium. Ce phénomène montre que quel que soit le protocole de routage les optimisations nécessaires pour améliorer le délai de bout-en-bout se situent au niveau de la couche MAC et en particulier au niveau de l'algorithme d'accès au médium CSMA/CA qui introduit un délai d'attente aléatoire avant que les nœuds ne puissent transmettre leurs messages.

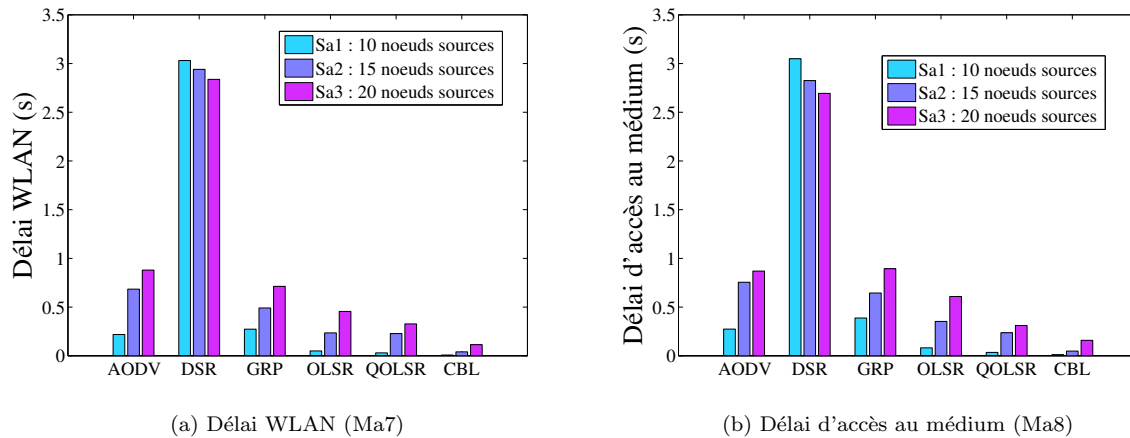


FIGURE 5.13 – Moyenne de Ma7 et Ma8 entre $t=30$ s et $t=60$ s dans le cas des scénarios Sa1 à Sa3 pour les protocoles CBL-OLSR, OLSR, QOLSR, AODV, DSR et GRP avec une application type IEEE

5.2 Application de localisation distribuée coopérative

5.2.1 Intérêt de la localisation pour les applications véhiculaires

La fonction de localisation est l'une des plus importantes pour les réseaux véhiculaires. Elle est nécessaire pour de nombreuses applications allant de la navigation au guidage autonome du véhicule [12].

En ce qui concerne les applications de sécurité passives ou actives, un véhicule devra localiser finement les véhicules voisins afin de détecter de potentielles situations dangereuses. Selon le délai restant estimé avant l'accident, le système devra soit avertir le conducteur, soit assurer une prise de contrôle efficace du véhicule. En sécurité post-accident, la localisation précise de l'accident permettra une intervention rapide des secours permettant de minimiser l'impact de l'accident pour les blessés et pour la sécurité des autres usagers de la route.

La mesure de positionnement par le GPS est réalisée à l'aide de la mesure du temps écoulé entre l'instant de l'émission d'une onde électromagnétique par un satellite GPS et l'instant de sa réception par le récepteur GPS embarqué dans un véhicule. Le récepteur compare l'heure d'émission incluse dans le signal reçu avec l'heure de sa réception. Après multiplication par la vitesse de l'onde électromagnétique (vitesse de la lumière), la valeur obtenue fournit une pseudo-distance. Cette pseudo-distance est assimilable à une distance, mais elle est entachée à la fois par l'erreur de synchronisation entre l'horloge atomique du satellite et celle du récepteur GPS, mais aussi par les erreurs dues aux dégradations subies par l'onde lors de sa propagation entre le satellite et le récepteur, en particulier lors de sa traversée de l'atmosphère.

Le calculateur du récepteur GPS peut obtenir la position absolue du véhicule à partir des pseudo-distances obtenues avec quatre satellites GPS dont il connaît les positions à l'heure d'émission des signaux. Ces mesures lui permettent de résoudre un système d'équations à quatre inconnues : la position du récepteur en trois dimensions, X, Y et Z, et la dérive de son horloge par rapport à celle des satellites. Un calcul en mode dégradé est cependant possible avec seulement trois satellites si l'altitude du véhicule est connue. Lorsque plus de quatre satellites sont visibles, le système d'équations à résoudre est surabondant, rendant possible l'estimation des erreurs de positionnement. Le point délicat du positionnement par GPS provient de la difficulté à synchroniser les horloges atomiques des satellites et celle, moins précise, du récepteur. Une erreur d'un milliardième de seconde provoque un décalage de 300 mètres sur la position. Le récepteur ne peut bien entendu pas bénéficier d'une horloge atomique comme c'est le cas des satellites en raison du coût de fabrication. Il doit néanmoins disposer d'une horloge assez stable, mais dont l'heure n'est a priori pas synchronisée avec celle des satellites.

Différents facteurs viennent affecter la précision de l'information de localisation fournie par un récepteur GPS :

- la traversée des couches de l'atmosphère induit un délai supplémentaire à la propagation du signal,

notamment lors du passage dans l'ionosphère et la troposphère. Ce délai est sensiblement identique pour deux récepteurs proches l'un de l'autre et cause donc une erreur appelée erreur commune ;

- les multi-trajets de l'onde ajoutent un délai qui est spécifique au récepteur appelé erreur non-commune. Lorsque le récepteur GPS se situe dans un canyon (gorges, montagne ou milieu urbain), une réflexion de l'onde électromagnétique contre une surface (typiquement un immeuble en milieu urbain) augmente le temps de propagation du signal jusqu'au récepteur et par conséquent la pseudo-distance mesurée ;
- les bruits de réception, spécifiques au récepteur, affectent également le positionnement. Ces erreurs sont incluses dans l'erreur non-commune.

Bien que le GPS offre un service honorable pour la navigation du véhicule, sa précision n'est pas suffisante. L'erreur moyenne de positionnement est d'environ 10 mètres [194]. L'étude [195] confirme ce résultat en comparant les erreurs de positionnement d'un système GPS traditionnel avec un GPS *Real Time Kinematic* (RTK) utilisé en tant que référence et fournissant une position précise de l'ordre de quelques centimètres. En écartant 5% de données aberrantes, les résultats ont montré une erreur de positionnement du GPS traditionnel de 10 m. Ainsi, cette erreur de positionnement ne permet pas, par exemple, à un système embarqué de déterminer sur quelle voie de circulation se trouve le véhicule. Les applications des ITS auront besoin d'une localisation plus précise des véhicules. La question se pose alors de mettre en œuvre des techniques pour améliorer la précision du GPS.

[196] propose un état de l'art des techniques permettant l'amélioration du positionnement en utilisant d'autres informations sur les équipements :

- Il existe des systèmes de navigation à l'estime (*dead reckoning systems*) qui couplent la mesure de la position GPS aux mesures d'autres capteurs (odomètre, gyroscope...). Cependant, ces systèmes ne peuvent être utilisés que sur de courtes périodes (typiquement en tunnel) car leurs erreurs de mesure dérivent avec le temps. Par exemple, dans le cas d'un véhicule circulant à 100 km/h, une erreur de plus de 20 mètres est à prévoir avec un système de navigation à l'estime utilisé durant 30 secondes.
- L'utilisation de stations de base permet d'améliorer le positionnement des véhicules. Ces stations ont une position fixe connue par les véhicules. Chaque véhicule calcule la distance qui le sépare des stations de base grâce au temps écoulé entre l'émission des signaux par les stations de base et leur réception par le véhicule. Dans sa zone de portée de communication, la présence de trois stations de base permet à un véhicule de connaître sa position grâce à une simple triangulation. Cependant cette solution est coûteuse car il faudrait équiper les bords de route avec un nombre important de stations de base (trois stations par zone de portée soit au minimum trois stations pour 1 km²).
- La localisation grâce à l'usage de vidéos ou d'images peut être une solution en particulier pour obtenir le positionnement au niveau des voies de circulation des véhicules.
- La corrélation cartographique (*map matching*) peut être utilisée si le véhicule circule sur une route en combinant les données de positionnement du véhicule à celles de la cartographie de l'infrastructure routière afin de repositionner le véhicule sur la route la plus proche.
- Le positionnement coopératif distribué permet d'améliorer le positionnement d'un véhicule en utilisant la connaissance de la position des autres véhicules [197, 198].

Parmi les propositions, [11] a mis au point une nouvelle approche pour améliorer le positionnement des véhicules à l'aide des applications *Cooperative Map Matching* (CMM) et *Dynamic base station de base Differential GPS* (DDGPS). Cet algorithme comporte deux étapes :

- dans la première étape, la partie de l'application locale au véhicule intègre les contraintes de l'infrastructure routière des véhicules coopératifs voisins à un saut sur son propre algorithme de corrélation cartographique CMM ;
- dans la seconde étape, les parties de l'application locales aux véhicules partagent entre elles leurs corrections des valeurs de pseudo-distance reçues par le GPS compensant ainsi l'erreur commune d'un satellite GPS. Les véhicules sont dans cette étape assimilables à des "stations de base dynamiques" DDGPS.

Les travaux de [11] ont montré que l'usage de véhicules voisins en tant que satellites virtuels permettait de réduire l'erreur de positionnement si la densité de véhicules est suffisante (figure 5.14). Plus il y a de véhicules voisins, plus l'erreur de positionnement et son écart type diminuent. Par exemple, avec 8 véhicules exécutant l'algorithme CMM, l'erreur de positionnement passe à 7 m et l'écart type à 3 m. L'écart type de l'erreur de positionnement est une donnée importante en termes de sûreté de fonctionnement du système véhiculaire. Le risque d'intégrité sur la position est d'autant plus réduit que l'écart type est faible.

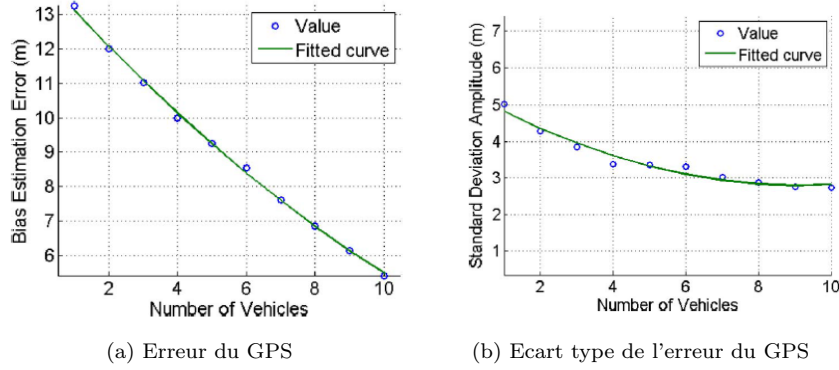


FIGURE 5.14 – Utilisation de la localisation distribuée coopérative en fonction du nombre de véhicules exécutant l'application CMM - source : [11]

Les méthodes CMM et DGPS sont résumées ci-après. Les variables nécessaires à ces méthodes sont présentées, notamment celles distribuées qui nécessitent un échange de leurs valeurs entre les véhicules communicants.

5.2.2 Présentation de l'application de corrélation cartographique coopérative (CMM)

La figure 5.15 illustre les différentes étapes de la méthode CMM selon le point de vue du véhicule blanc. Pour chaque véhicule, la position calculée par le récepteur GPS au moyen des pseudo-distances est signalée par un point rouge, la position réelle l'est par un point vert. L'ellipse rouge représente l'incertitude sur la position calculée. L'algorithme CMM se déroule ainsi.

Soit $\rho_j^{(i)}$ la pseudo-distance associée au satellite j et mesurée par le véhicule i . $\rho_j^{(i)}$ est entachée d'erreurs :

$$\rho_j^{(i)} = D_j^{(i)} - c\delta t^{(i)} + \varsigma_j^{(i)} + \eta^{(i)} \quad (5.2)$$

avec

$$\left\{ \begin{array}{l} \rho_j^{(i)} : \text{ la pseudo-distance associée au satellite } (j) \text{ et mesurée par l'application du véhicule } i \\ D_j^{(i)} : \text{ la distance réelle entre le satellite } (j) \text{ et le véhicule } (i) \\ c : \text{ la célérité de l'onde électromagnétique} \\ \delta t^{(i)} : \text{ l'erreur de synchronisation des horloges} \\ \varsigma_j^{(i)} : \text{ les erreurs communes} \\ \eta^i : \text{ les erreurs non communes} \end{array} \right.$$

Une fois que l'algorithme du véhicule blanc (i) a mesuré la pseudo-distance $\rho_j^{(i)}$ de chaque satellite visible (j), l'algorithme CMM calcule la position GPS ($\tilde{X}^{(i)} = [\tilde{x}^{(i)}, \tilde{y}^{(i)}, \tilde{z}^{(i)}]$) et la covariance de cette position. Il limite l'incertitude de la position GPS en appliquant les contraintes routières à cette position (algorithme de *map matching*). La position du véhicule blanc est alors incluse dans la zone hachurée en blanc de l'illustration (a) de la figure 5.15.

L'application exploite ensuite la propriété suivante : “les erreurs communes $\varsigma_j^{(i)}$ liées à chaque satellite visible (j) conduisent au même biais de la position GPS pour chaque véhicule (i) dans un même voisinage” [11].

En recevant, via une liaison V2V, les mesures de pseudo-distance du véhicule bleu (v) et en calculant à la fois la position GPS du véhicule bleu $\tilde{X}(v)$ et la covariance de sa position, l'algorithme du véhicule blanc (i) restreint à nouveau l'incertitude de sa position en appliquant les contraintes de la cartographie routière de ce véhicule à sa propre position (figure 5.15b). Les contraintes cartographiques des autres véhicules s'appliquent au véhicule blanc (i) “seulement si le même ensemble de satellites est utilisé dans le calcul de la position de tous les véhicules” [11]. De cette façon, la position estimée $\hat{X}^{(i)}$ du véhicule blanc (i) est améliorée à partir de la connaissance de la position des véhicules à proximité (figure 5.15c).

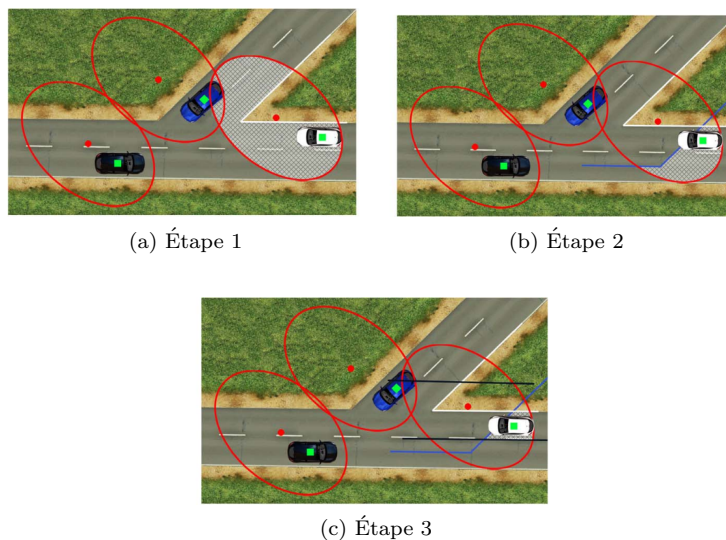


FIGURE 5.15 – Utilisation de la localisation distribuée coopérative - source [11]

Par conséquent, en termes de besoins de communication, la fonction CMM exige que chaque véhicule (i) diffuse ses mesures de pseudo-distance $\rho_j^{(i)}$ pour chaque satellite visible (j) à son voisinage proche.

5.2.3 Présentation de la station de base dynamique (DDGPS)

Selon [11], la fonction DDGPS permet à chaque véhicule de corriger son estimation de position sur la base de la connaissance partagée des informations de correction de biais d'erreurs pour chaque satellite visible. À cette fin, chaque véhicule calcule une correction de pseudo-distance $\Delta\rho_j^{(i)}$ pour chaque satellite visible (j) (équation 5.3).

$$\Delta\rho_j^{(i)} = \hat{D}_j^{(i)} - \rho_j^{(i)} \quad (5.3)$$

où $\hat{D}_j^{(i)}$ est la distance géométrique entre la position du satellite (j) et l'estimation de position $\hat{X}^{(i)}$ par (i); $\hat{X}^{(i)}$ peut avoir été précédemment obtenu avec l'application CMM ou tout autre système de positionnement. Par conséquent, chaque véhicule récepteur (r) peut générer un ensemble de corrections différentielles de pseudo-distances $\rho_{j,corr}^{(i)}$ pour chaque satellite visible j ($\rho_{j,corr}^{(i)} = \rho_j^{(r)} + \Delta\rho_j^{(i)}$).

Cependant, il peut arriver qu'un véhicule (r) reçoive plusieurs corrections de pseudo-distances de différents véhicules pour un même satellite (j). Dans ce cas, il doit fusionner les corrections de pseudo-distances suivant si ces données sont indépendantes ou non. Par conséquent, l'application DDGPS fournit et transmet avec chaque valeur de $\Delta\rho_j^{(i)}$ deux autres informations : d'une part, la variance des corrections de pseudo-distances $\sigma_j^{(i)}$ qui décrit le niveau de confiance de la valeur $\Delta\rho_j^{(i)}$ et, d'autre part, la liste des identifiants des véhicules qui ont été impliqués dans le calcul de chaque correction d'erreur.

Ainsi, lorsqu'un véhicule reçoit plusieurs ensembles de corrections de pseudo-distances, il peut les fusionner après avoir vérifié que ces données soient indépendantes : si les listes des identifiants de véhicules sont disjointes, les données sont indépendantes. Dans ce cas, le véhicule récepteur (r) utilise une moyenne pondérée pour fusionner les corrections reçues, il calcule la nouvelle variance (voir [11]) et il fusionne les listes d'identifiants. Lorsque les listes ne sont pas disjointes, les données ne sont pas indépendantes. L'algorithme DDGPS ne conserve que la correction de pseudo-distance $\Delta\rho_j^{(i)}$ qui a la plus petite valeur de $\sigma_j^{(i)}$.

Donc, en termes de besoins de communication, la fonction DDGPS exige que chaque véhicule (i) envoie un message contenant les identifiants des satellites (j) ; la valeur $\Delta\rho_j^{(i)}$; la valeur $\sigma_j^{(i)}$; et la liste des identifiants des véhicules impliqués dans la correction de cette erreur.

5.2.4 Adaptation de l'application au schéma CBL et modélisation

Il est important de noter que la définition des nœuds branche et des nœuds feuille du schéma CBL génère, à un moment donné, des groupes disjoints de nœuds. Chaque groupe est identifiable par son nœud branche. Nous proposons d'utiliser cette indépendance des groupes de nœuds feuille, formés autour des nœuds branche, pour l'application DDGPS afin de réduire la taille des messages applicatifs en omettant la liste des identifiants des véhicules impliqués dans la correction d'erreur. Par conséquent, nous fixons que :

- seuls les nœuds feuille (i) envoient leurs mesures de pseudo-distances $\rho_j^{(i)}$ pour chaque satellite visible (j) à destination de leur nœud branche relais qui exécute la fonction CMM ;
- les nœuds branche (i) exécutent à la fois les applications CMM et DDGPS pour calculer les données de correction pour chaque satellite visible j ;
- seuls les nœuds branche sont autorisés à relayer, vers les nœuds branche en amont de leur chaîne et de leurs nœuds feuille attachés, les messages DDGPS contenant un ensemble de corrections de pseudo-distances $\Delta\rho_j^{(i)}$ et $\sigma_j^{(i)}$.

Dans notre implémentation des applications CMM et DDGPS, les nœuds branche jouent le rôle de station de base dynamique. Les messages DDGPS sont relayés uniquement par les nœuds branche en amont de la chaîne. Ainsi la structure CBL permet de garantir l'indépendance des données de correction d'erreur sans avoir à transférer la liste des identifiants de véhicules impliqués dans le processus de correction. Cela permet de restreindre la taille des messages à envoyer et, ainsi, l'usage des ressources de communication du médium limitées en bande passante.

Par conséquent, les messages du DDGPS ne contiendront que les informations suivantes pour chaque correction : l'identifiant du satellite j , $\sigma_j^{(i)}$, $\Delta\rho_j^{(i)}$.

5.2.4.1 Format des paquets de l'application CMM adapté à CBL

Les paquets de l'application CMM sont définis en figure 5.16. Ils se composent de plusieurs couples de champs de 32 bits contenant chacun un identifiant d'un satellite (j) et la valeur de pseudo-distance reçue par le véhicule (i) de ce satellite ($\rho_j^{(i)}$). Ainsi, 64 bits codent chaque satellite visible.

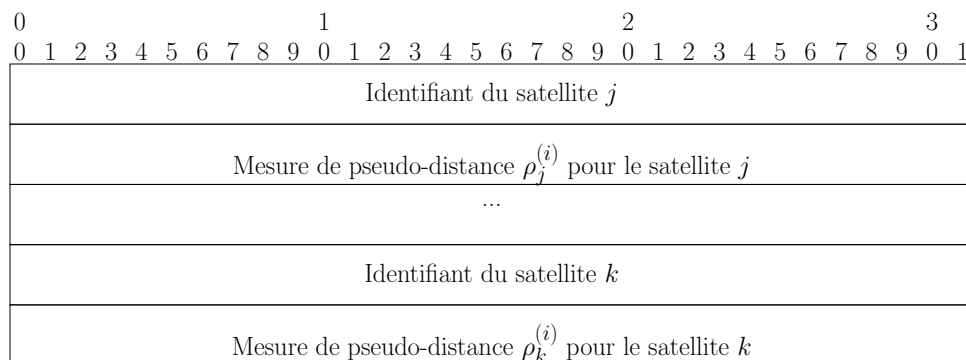


FIGURE 5.16 – Format des paquets applicatifs CMM transmis par les nœuds feuille

0																																1																																2																																3																															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9																																																																																								
Identifiant du satellite j																																																																																																																															
Variance de la correction de pseudo-distance $\sigma_j^{(i)}$ pour le satellite j																																																																																																																															
Correction de pseudo-distance $\Delta\rho_j^{(i)}$ pour le satellite j																																																																																																																															
...																																																																																																																															
Identifiant du satellite k																																																																																																																															
Variance de la correction de pseudo-distance $\sigma_k^{(i)}$ pour le satellite k																																																																																																																															
Correction de pseudo-distance $\Delta\rho_k^{(i)}$ pour le satellite k																																																																																																																															

FIGURE 5.17 – Format des paquets d’application DDGPS envoyés par les nœuds branche à leurs nœuds feuille et au nœud branche en amont.

Pour réaliser l’analyse des performances de l’implémentation de CMM sur le schéma CBL-OLSR, les messages CMM seront envoyés en mode *unicast* par les nœuds feuille à leur nœud branche relais (voir 5.2.4.3).

5.2.4.2 Format des paquets de l’application DDGPS adapté à CBL

Les paquets de l’application DDGPS sont définis figure 5.17. Ils se composent de plusieurs ensembles de trois champs de 32 bits contenant chacun l’identifiant d’un satellite visible (j), la variance de correction ($\sigma_j^{(i)}$) et la correction de pseudo-distance $\Delta\rho_j^{(i)}$. Ainsi, 96 bits codent chaque satellite visible.

Pour réaliser l’analyse des performances de la mise en œuvre de l’application DDGPS sur le schéma CBL, les messages DDGPS seront envoyés en mode *multicast* par les nœuds branche à leur nœud feuille élus et à leur nœud branche en amont de la chaîne.

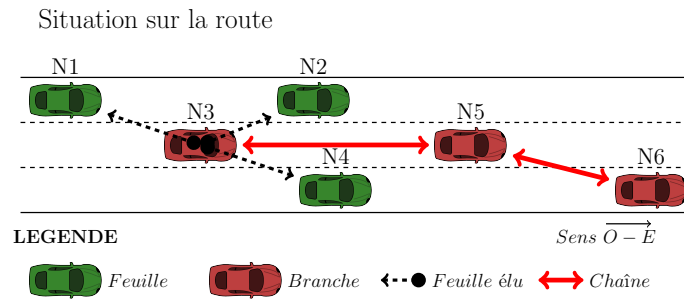
5.2.4.3 Diagramme temporel de fonctionnement des applications CMM et DDGPS

La figure 5.18 illustre une séquence d’envoi de messages applicatifs CMM et DDGPS dans un réseau VANET à six nœuds. Les nœuds N3, N5 et N6 sont trois nœuds branche successifs d’une même chaîne ; le nœud N3 est en aval du nœud N5 qui est en aval du nœud N6. Les nœuds N1, N2 et N4 sont des nœuds feuille ayant élu comme nœud branche relais le nœud N3.

Les nœuds feuille N1, N2 et N4 envoient chacun un message applicatif CMM contenant leurs mesures de pseudo-distance avec les identifiants de satellites à destination de leur nœud branche relais (nœud N3). Ensuite, le nœud N3, après avoir exécuté l’application CMM, puis l’application DDGPS, envoie un message applicatif DDGPS à ses nœuds feuille (nœuds N1, N2 et N4) et au nœud branche amont (N5) avec comme valeurs $\Delta\rho_j^{(3)}$ et $\sigma_j^{(3)}$ et $\sigma_j^{(3)}$. Les nœuds récepteurs calculent de nouveau et affinent leur position avec ces informations. Enfin, le nœud branche N5, qui n’a pas de nœuds feuille élus dans cet exemple, retransmet simplement un message applicatif DDGPS avec la mise à jour de la correction et de la variance de la pseudo-distance à son nœud branche amont N6. Cette mise à jour prend en compte les propres mesures de pseudo-distance du nœud N5 et ses contraintes routières.

5.2.5 Évaluation de performances par la simulation

L’évaluation de performances de l’application CMM-DDGPS avec notre proposition de protocole de routage CBL-OLSR fait l’objet de cette section. Pour cette évaluation menée sous OPNET Riverbed Modeler, nous avons choisi un scénario de mobilité réaliste comprenant l’autoroute A27. Nous avons choisi le scénario S12 qui comporte le plus grand nombre de nœuds afin d’estimer si le réseau de communication mettant en œuvre notre proposition supporte bien le trafic applicatif CMM-DDGPS.



Échange de messages CMM et DDGPS (S correspond à l'ensemble des satellites visibles)

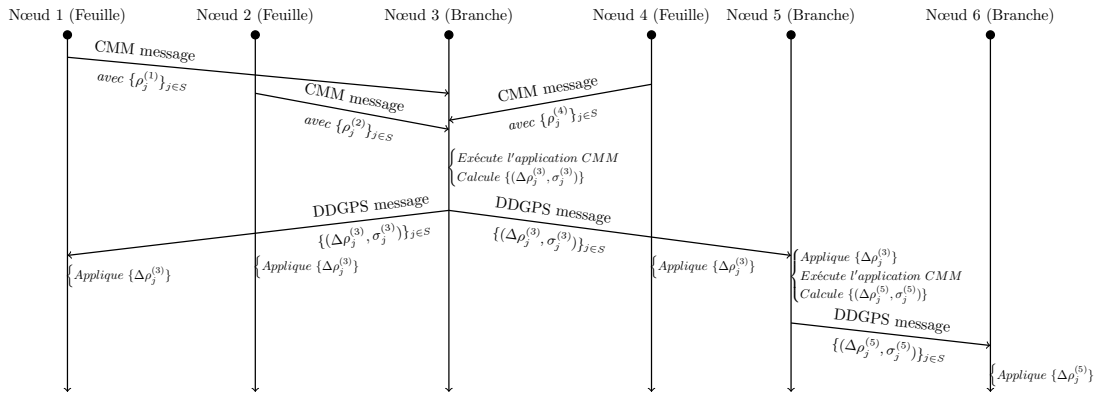


FIGURE 5.18 – Séquence d’envoi de messages applicatifs CMM-DDGPS pour un VANET composé de 6 nœuds

La configuration du simulateur OPNET Riverbed Modeler est identique à celle présentée en section 3.3.2.1 et dans le tableau 3.9. Concernant le trafic de routage, nous avons paramétré les nœuds avec le cas A présenté dans le tableau 4.4. Les simulations sont exécutées durant 300 s, mais les statistiques ne sont enregistrées qu’à partir de 200 s.

Les applications CMM et DDGPS sont exécutées en parallèle. Nous avons fait varier la fréquence d’envoi des messages applicatifs CMM et DDGPS de 2 Hz à 20 Hz avec un pas de 2 Hz (soit un intervalle d’attente entre deux messages de 50 ms à 500 ms). Nous avons fixé le nombre de satellites visibles à 12 satellites (le nombre minimal, maximal et moyen de satellites GPS visibles est respectivement de 4, 24 et 10 [199]). Ainsi, avec un en-tête de 128 bits défini par le protocole OLSR, les nœuds feuille envoient un paquet CMM de 896 bits et les nœuds branche un paquet DDGPS de 1280 bits. Pour chaque fréquence d’envoi de messages applicatifs configurée, la simulation est exécutée 10 fois avec des valeurs différentes de *seed* pour l’initialisation du générateur de nombre aléatoire du logiciel OPNET Riverbed Modeler afin d’éviter une configuration du réseau particulière.

Les métriques de performances applicatives (Ma - Métriques applicatives) utilisées pour cette évaluation sont les suivantes :

- Ma7 : Délai WLAN, le délai de transmission de bout-en-bout d’une trame en ms ;
- Ma8 : Délai d’accès au médium en ms ;
- Ma9 : PDR, le taux de paquets reçus en ms ;
- Ma10 : délai IP, le délai de transmission de bout-en-bout d’un message applicatif en ms.

La figure 5.19 présente le délai WLAN (Ms7) et la figure 5.20 le délai d’accès au médium (Ms8) pour toutes les configurations. Ces délais n’augmentent pas, quel que soit l’intervalle de temps entre deux messages applicatifs, sauf lorsque cet intervalle est le plus faible (soit de 50 ms ce qui est équivalent à une fréquence d’envoi de 20 Hz). En particulier, le délai d’accès au médium passe alors en moyenne de 0,03 ms à 0,05 ms. La valeur minimale du délai WLAN est de 0,2 ms. Les valeurs maximales de délai varient et ne sont représentatives que d’une situation particulière (par exemple lors d’une transmission concurrente de paquets engendrant l’augmentation du temps de *backoff*). En effet, ces points maximums

se trouvent largement en dehors de la zone bleu-claire représentant 99% des valeurs de délai. Par exemple, pour l'intervalle d'envoi des paquets applicatifs de 300 ms, un point extrême de délai WLAN à 0,4 ms est observé qui s'explique par la présence à ce même intervalle d'un point extrême de délai d'accès au médium à 0,25 ms.

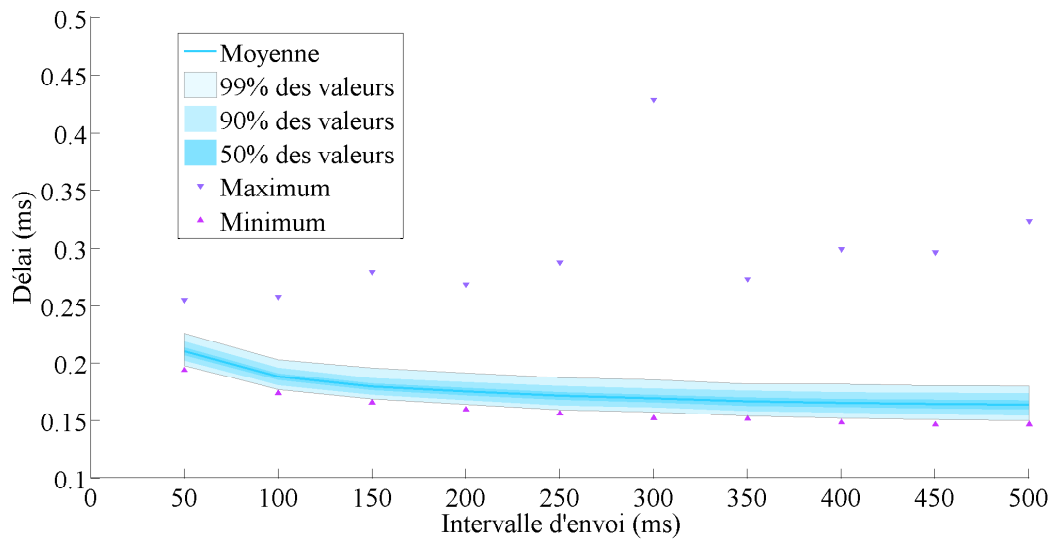


FIGURE 5.19 – Délai WLAN (Ma7) dans le cas du scénario S12 avec l'application de localisation coopérative

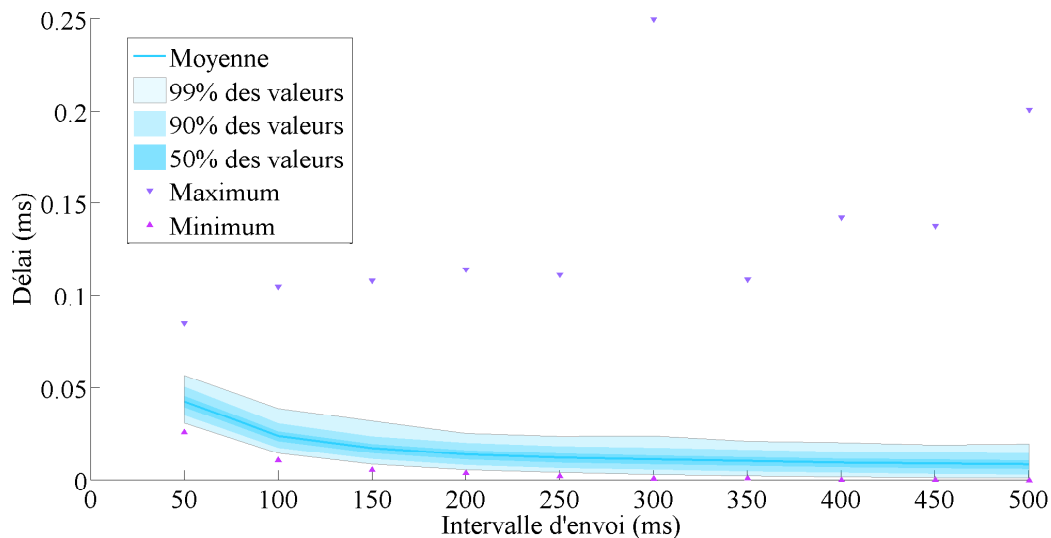


FIGURE 5.20 – Délai d'accès au médium (Ma8) dans le cas du scénario S12 avec l'application de localisation coopérative

La figure 5.21 montre le taux de paquets délivrés (Ma9). Ce taux augmente lorsque l'intervalle d'attente entre deux paquets applicatifs augmente. En effet, dans ce cas les ressources de communication sont moins utilisées. On remarque que la métrique Ma9 est toujours supérieure à 93%. Elle est en moyenne égale à 95% lorsque l'intervalle d'attente de l'envoi des messages applicatifs est supérieur à 200 ms.

La figure 5.22 illustre le délai IP de bout en bout du réseau (Ma10). Inversement au taux de paquets délivrés, le délai IP diminue lorsque l'intervalle d'envoi augmente. En effet, plus l'intervalle d'envoi est grand, moins le nombre de paquets applicatifs est important ce qui libère des ressources en bande passante. On remarque que les délais sont toujours inférieurs à 300 ms. Tout comme le délai WLAN, des points extrêmes maximums sont visibles (pour le même exemple du cas dans lequel l'intervalle d'envoi est de 300 ms avec un maximum de délai IP à 150 ms alors que la moyenne est de 50 ms et que 99% des valeurs

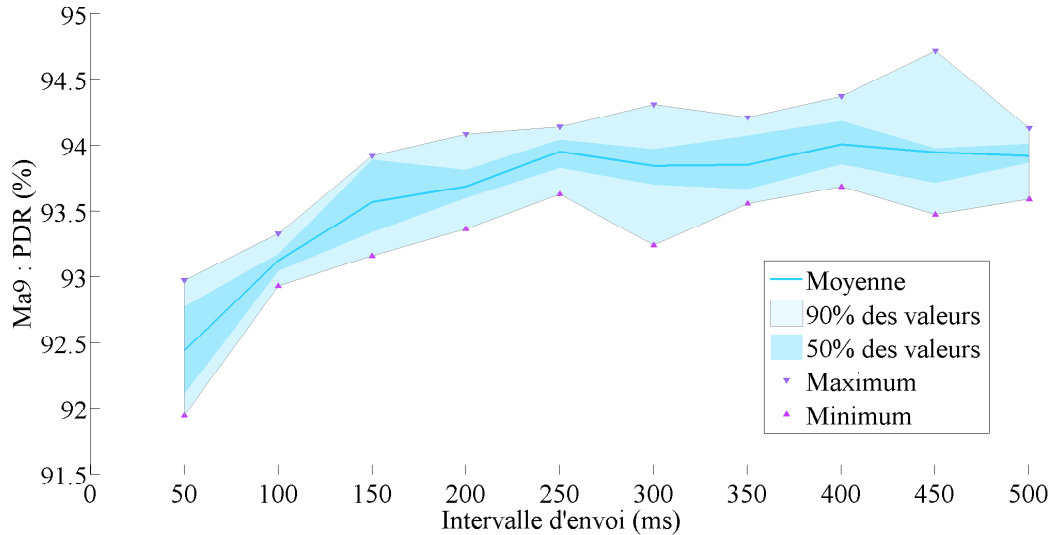


FIGURE 5.21 – PDR (Ma9) dans le cas du scénario S12 avec l'application de localisation coopérative

de délais sont inférieures à 60 ms). Ces valeurs aberrantes sont dues à des situations singulières et ne sont pas représentatives. En traçant la droite de la fonction identité, on peut identifier l'intervalle minimal d'attente entre deux paquets applicatifs afin que le délai moyen de bout en bout du réseau soit inférieur à cet intervalle. Cette valeur est de 125 ms. Ainsi dans la configuration d'étude, en envoyant les paquets applicatifs CMM et DDGPS toutes les 125 ms, on s'assure que ces paquets soient reçus au bout d'un délai inférieur à 125 ms, c'est à dire avant l'envoi d'un nouveau paquet contenant la mise à jour des données intégrées à ce paquet.

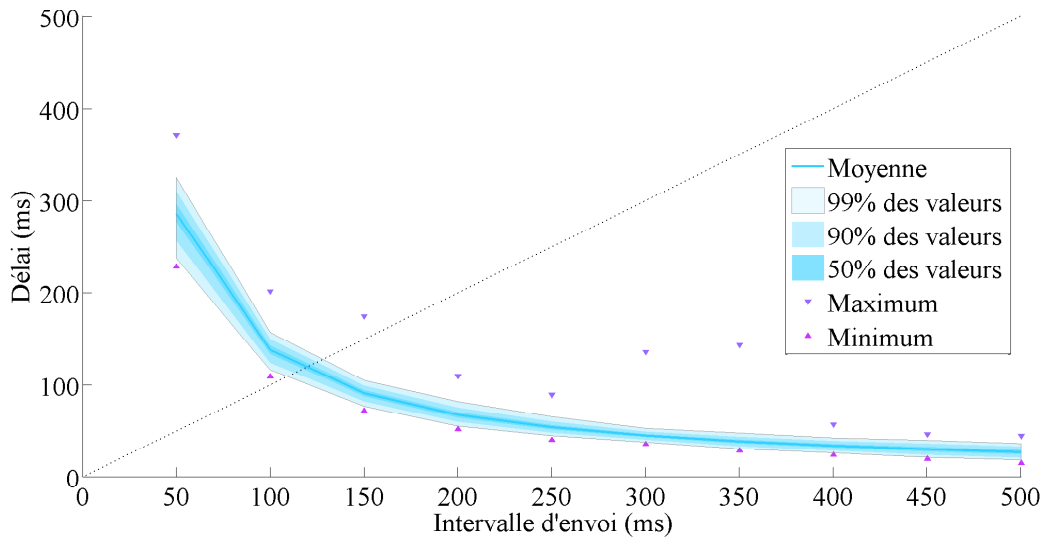


FIGURE 5.22 – Délai applicatif (Ma10) dans le cas du scénario S12 avec l'application de localisation coopérative

Concernant l'application, les résultats structurels du scénario S12 ont déjà été présentés dans le tableau 4.8. On rappelle que la structure formée par CBL-OLSR consiste, dans ce scénario S12, à une moyenne de 2,29 chaînes. L'objectif de 2 chaînes dans ce réseau est donc presque atteint aux ruptures de chaînes près. Le nombre de nœuds feuille attachés à un même nœud branche est en moyenne de 1,72. La figure 5.23 montre la variabilité au cours du temps de la simulation (pour rappel dans notre évaluation les statistiques présentées ci-avant ne sont enregistrées qu'entre les temps $t=200$ s et $t=300$ s). Ainsi chaque nœud branche reçoit des corrections CMM de 1 à 3 nœuds feuille. L'amélioration de la précision des positions des nœuds augmente avec le nombre de nœuds dans chaque groupe. Les nœuds branche avec

un nombre plus élevé de nœuds feuille élus fournissent à l'ensemble du système de meilleures corrections grâce aux messages DDGPS. Notons que le scénario routier R4 étudié comporte l'autoroute A27 qui est une deux fois deux voies. Pour des réseaux autoroutiers tels que le réseau R2 qui comporte une voie de plus dans chaque sens de circulation et avec une densité moyenne de véhicules (scénario S5), nous avons vu que le nombre de nœuds feuille par nœud branche est en moyenne de 5,51 (tableau 4.2) avec des maximales à plus de 19 nœuds par groupe (figure 5.24). Dans ce cas, l'amélioration de la précision des positions des nœuds en sera d'autant améliorée.

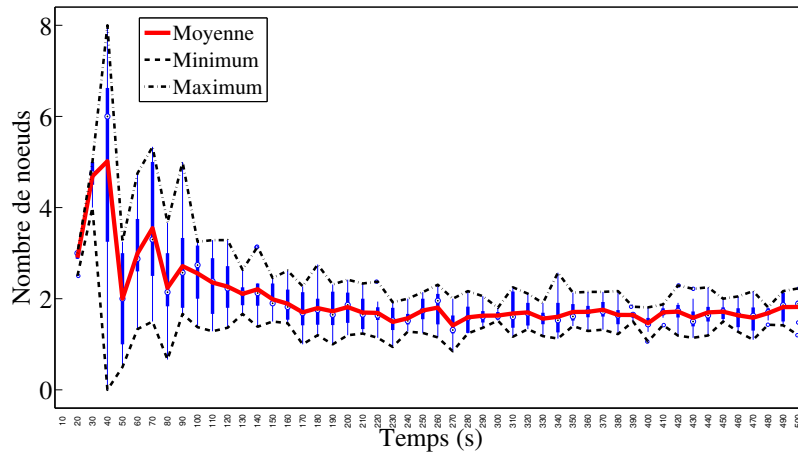


FIGURE 5.23 – Nombre de nœuds feuille attachés au même nœud branche (Ms5) dans le cas du scénario S12

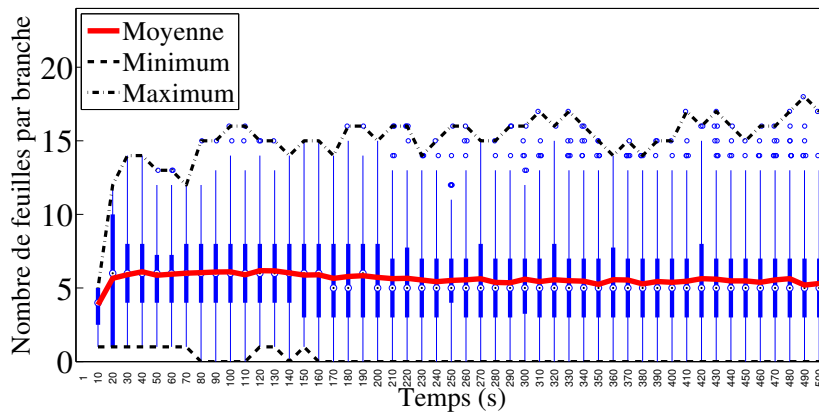


FIGURE 5.24 – Nombre de nœuds feuille attachés au même nœud branche (Ms5) dans le cas du scénario S5

5.3 Application de perception élargie

5.3.1 Présentation de l'application

La perception élargie coopérative est une application de sécurité routière visant à augmenter la perception de l'environnement autour des véhicules. Grâce à l'échange coopératif des positions et des vitesses des objets perçus par les capteurs d'un véhicule, les véhicules voisins peuvent améliorer leur perception de l'environnement. La note d'orientations stratégiques pour l'action publique sur le développement des véhicules autonomes [12] détaille cette application :

“Les véhicules automatisés se repèrent en croisant une perception de l'environnement (signalisation horizontale, dispositifs de retenue, signalisation verticale) par des capteurs natifs (caméras, radars, lidars) et un repérage satellitaire lié à une cartographie embarquée. Les capteurs actuellement ont une portée maximale de 100 à 200 m, ce qui paraît insuffisant pour l'anticipation de certaines situations ou certains événements critiques affectant la sécurité routière [12]. Pour élargir et affiner l'horizon de perception du véhicule, des compléments permettent d'enrichir le modèle de perception et d'asseoir dans un premier temps, des procédures d'alerte du conducteur et des demandes de reprise en main. Ultérieurement, il est envisageable, selon la qualité (latence, fiabilité, précision) de ces données, que l'asservissement du véhicule s'appuie aussi sur ces compléments, permettant alors d'aborder ces situations ou événements critiques en mode automatisé.”

Nous distinguons alors :

- la connaissance locale de l'environnement d'un véhicule qui est donnée par ces capteurs embarqués ;
- la connaissance distribuée de l'environnement qui concerne les informations que le véhicule a reçues de ses voisins.

Cette perception élargie coopérative distribuée permet aux véhicules de construire une carte locale dynamique (*Local Dynamic Map* (LDM)) [200] comme l'illustre la figure 5.25. Ces cartes locales dynamiques permettent de positionner l'ensemble des objets (dynamiques ou statiques) que rencontrent le véhicule durant son parcours. Ces objets sont détectés grâce aux capteurs embarqués dans les véhicules que ce soit des caméras, des radars ou des lidars. L'ensemble des informations issues de ces capteurs doit être fusionnées afin de former cette carte locale dynamique. Avec l'application de perception élargie, cette carte est enrichie à l'aide des informations envoyées par les véhicules voisins.

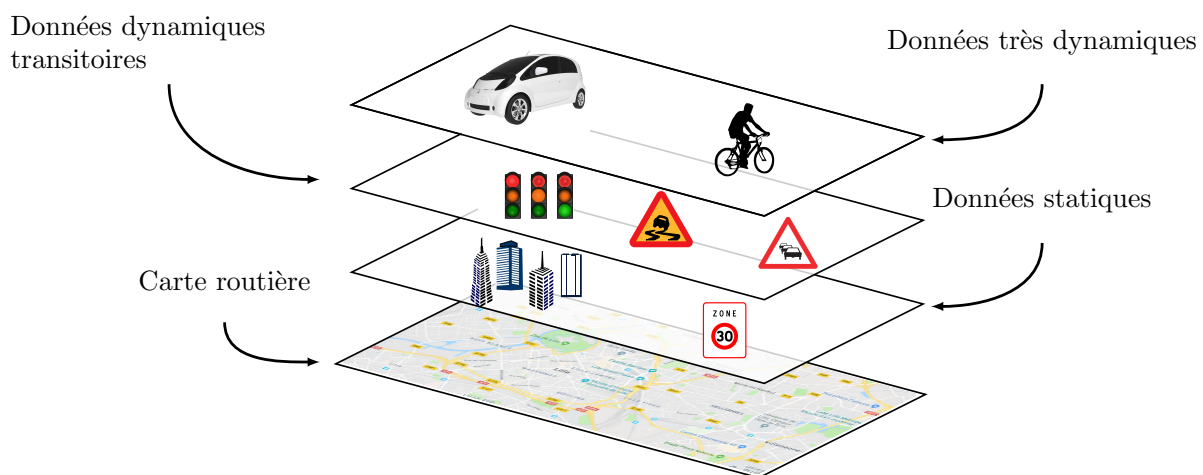


FIGURE 5.25 – Concept de carte étendue - figure inspirée de [201]

L'article [202] montre l'intérêt de la perception coopérative en fonction du taux de pénétration des véhicules communicants au sein du marché automobile. Si le taux de pénétration est de 100%, c'est à dire que tous les véhicules sont communicants, alors l'envoi périodique de message HELLO (ou CAM) est suffisant pour construire une carte locale dynamique comprenant tous les véhicules (*awareness ratio* égal à 1) dans la zone de portée de transmission (300 m dans l'article [202]). Si le taux de pénétration est inférieur alors, l'application de perception élargie, par l'envoi de messages *Environmental Perception*

Message (EPM) montre un intérêt majeur à la détection du voisinage. Un véhicule équipé d'un capteur radar de portée 80 m et d'angle maximal de 60° ne détectera que 20% des véhicules de son voisinage (*awareness ratio* égal à 0,2). Avec par exemple un taux de pénétration de 40%, s'il utilise de plus l'application de perception élargie, alors il en détecte plus de 60% (*awareness ratio* égal à 0,6). En utilisant uniquement les messages CAM (ou HELLO) il n'en détecte que 45% (*awareness ratio* égal à 0,4). Les résultats de [202] montrent bien l'intérêt de l'application de perception élargie. Cependant il ne faut pas oublier qu'un système global aura besoin de redondance. Ainsi, même si le taux de pénétration des véhicules est de 100%, l'application de perception élargie est primordiale. Dans ce cas, elle permet d'une part d'améliorer la fiabilité des informations de la carte locale dynamique ; et d'autre part de partager des informations sur des objets statiques ou des objets dynamiques non communicants. Par exemple, avec ses propres capteurs, un véhicule ne peut détecter un animal traversant la chaussée et qui est masqué par un autre véhicule. Avec une application de perception élargie, cette information pourra lui être communiquée, notamment par le véhicule détectant l'obstacle pour lui permettre de réagir en conséquence.

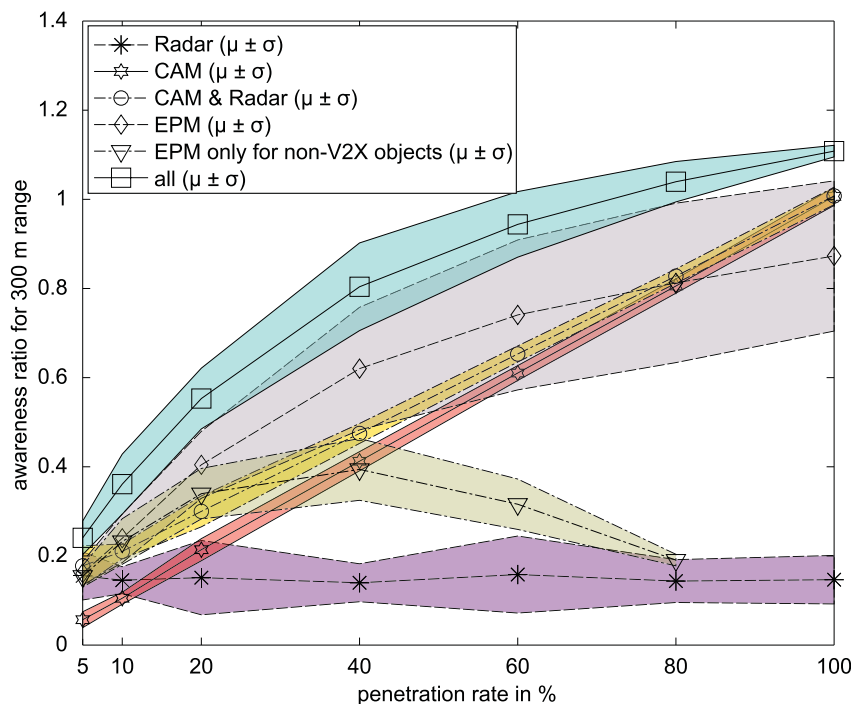


FIGURE 5.26 – Pourcentage de découverte du voisinage dans un rayon de 300 m en fonction du taux de pénétration des véhicules communicants - source : [202]

5.3.2 Besoins en communication

Les besoins de communications en termes de fréquence, de débit et de latence de l'application de perception élargie sont mal définis dans la littérature. Certains auteurs [202] proposent d'envoyer des messages EPM toutes les secondes.

Dans sa thèse, un auteur [203] propose des messages concernant la perception élargie envoyés à une fréquence de 10 Hz contenant des informations sur le nœud source (20 octets pour la position, la vitesse, les variances associées), sur les obstacles (22 octets pour l'identifiant, la position, la vitesse, les variances associées), sur l'infrastructure routière (9 octets pour l'attribut, le type et la confiance), sur les conditions météorologiques (4 octets pour le type, la visibilité, la distance de sécurité), sur la signalisation routière (8 octets pour la position, le type, l'information) et sur des informations additionnelles (3 octets pour le niveau de risque, le danger et le mode de fonctionnement). Les résultats de [203] montrent qu'une telle application ne peut être supportée si la densité de véhicule excède 100 véhicules/km sur l'autoroute.

Dans leurs travaux [204], certains auteurs définissent des messages EPM contenant des informations sur les obstacles voisins. Ces messages ont une taille variable entre 37 et 709 octets. Dans [205], les objets

dynamiques sont caractérisés par 24 octets (date, position, vitesse, accélération, taille) et les objets statiques par 18 octets (date, position, taille).

5.3.3 Adaptation de l'application au schéma CBL et modélisation

En ce qui concerne cette application de perception élargie, nous avons choisi de prendre les valeurs issues de l'article [205]. Notre adaptation du schéma CBL à l'application de perception élargie se fait selon le schéma suivant :

- Les nœuds feuille envoient un message contenant l'ensemble des objets (statiques et dynamiques) qu'ils ont détecté par leurs propres capteurs à destination de leur nœud branche relais.
- Les nœuds branche se chargent de fusionner les informations provenant de leurs nœuds feuille puis ils retransmettent ces informations aux nœuds feuille et au nœud branche en amont de la chaîne permettant ainsi à ces nœuds d'améliorer leur perception.

5.3.4 Évaluation de performances par la simulation

Pour cette évaluation menée sous OPNET Riverbed Modeler, nous avons choisi le même scénario de mobilité que celui utilisé pour l'application de localisation coopérative (section 5.2) : le scénario S12. La configuration du simulateur OPNET Riverbed Modeler est identique à celle présentée en section 3.3.2.1 et dans le tableau 3.9. Le trafic de routage est celui du cas A présenté dans le tableau 4.4. Les simulations sont lancées durant 300 s mais les statistiques ne sont enregistrées qu'à partir de 200 s lorsque le réseau est stabilisé. Nous avons fait varier la fréquence d'envoi des messages applicatifs EPM de 2 Hz à 20 Hz (soit un intervalle d'attente entre deux messages de 50 ms à 500 ms). Chaque nœud feuille calcule le nombre de nœuds voisins dont il a connaissance dans un rayon de 100 m. Il multiplie cette valeur par 24 octets et ajoute une valeur aléatoire comprise entre 0 et 10 multipliée par 18 octets représentant les obstacles fixes. Le résultat de ce calcul donne la taille des paquets à envoyer. Pour calculer la taille des paquets envoyés par les nœuds branche (qui ont fusionné les informations reçues de leurs nœuds feuille), nous avons fait l'hypothèse selon laquelle la taille des paquets était doublée. Pour chaque configuration, constituée d'un scénario et d'une fréquence d'envoi des messages applicatifs, la simulation est exécutée 10 fois avec des valeurs différentes pour l'initialisation du générateur de nombre aléatoire du logiciel OPNET Riverbed Modeler afin d'éviter une configuration du réseau particulière. Le taux de pénétration de l'application de perception élargie a été choisi afin de maximiser les contraintes du médium de communication soit à un taux de 100%, c'est à dire lorsque tous les véhicules sont communicants.

Les métriques de performances applicatives (Ma - Métriques applicatives) utilisées pour cette évaluation sont identiques à celles utilisées pour l'application de localisation coopérative (section 5.2).

La figure 5.27 présente le délai WLAN (Ms7) et la figure 5.28 montre le délai d'accès au médium (Ms8) pour toutes les configurations. Ces délais augmentent lorsque l'intervalle de temps entre deux messages applicatifs diminue. Par exemple, pour un intervalle d'envoi de 50 ms, le délai d'accès au médium est de 0,5 ms et pour un intervalle d'envoi de 500 ms, le délai d'accès au médium est de 0,1 ms. La valeur minimale du délai WLAN est de 0,2 ms. De même qu'avec l'application de localisation coopérative, les valeurs maximales de délai varient et ne sont représentatives que d'une situation particulière. Par exemple, pour l'intervalle d'envoi de paquet applicatif de 450 ms, un point extrême de délai WLAN à 0,28 ms est observé qui s'explique par la présence au même intervalle d'un point extrême de délai d'accès au médium à 0,02 ms.

La figure 5.29 montre le taux de paquets délivrés (Ma9). Ce taux augmente lorsque l'intervalle d'attente entre deux paquets applicatifs augmente. On remarque que la métrique Ma9 est toujours supérieure à 88%. Les valeurs sont plus faibles que celles de l'application de localisation coopérative, puisque les ressources de communication sont plus utilisées. Le taux de paquets reçus est en moyenne égal à 91% lorsque l'intervalle d'attente de l'envoi des messages applicatifs est supérieur 350 ms.

La figure 5.30 illustre le délai IP de bout en bout du réseau (Ma10). Inversement au taux de paquets délivrés, le délai IP diminue lorsque l'intervalle d'envoi augmente. En effet plus l'intervalle d'envoi est grand, moins le nombre de paquets applicatifs est important ce qui libère les ressources de la bande passante. On remarque que les délais sont toujours inférieurs à 600 ms. La zone représentant 99% des valeurs de délais est plus étendue que celle de l'application de localisation ce qui montre l'utilisation

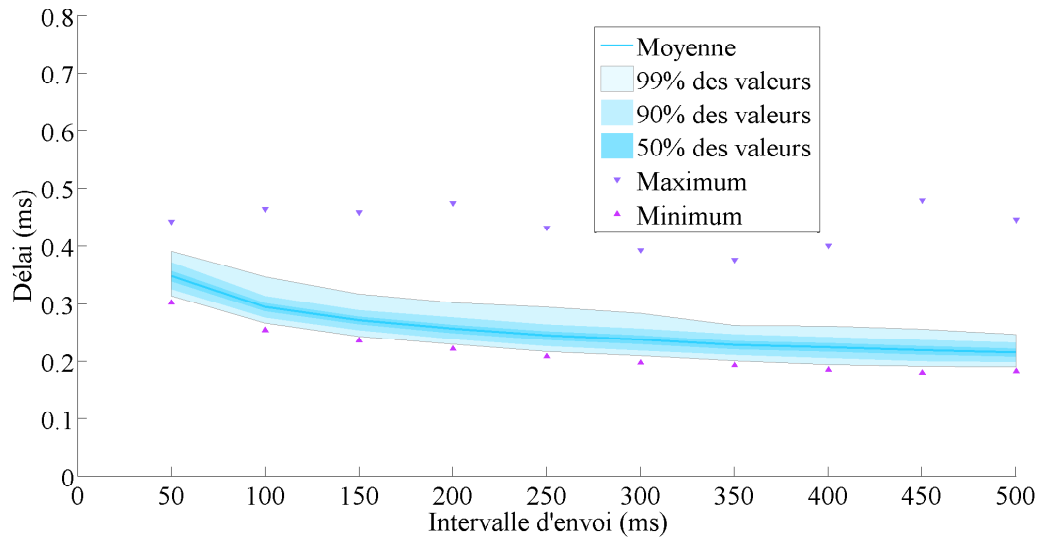


FIGURE 5.27 – Délai WLAN (Ma7) dans le cas du scénario S12 avec l'application de perception élargie

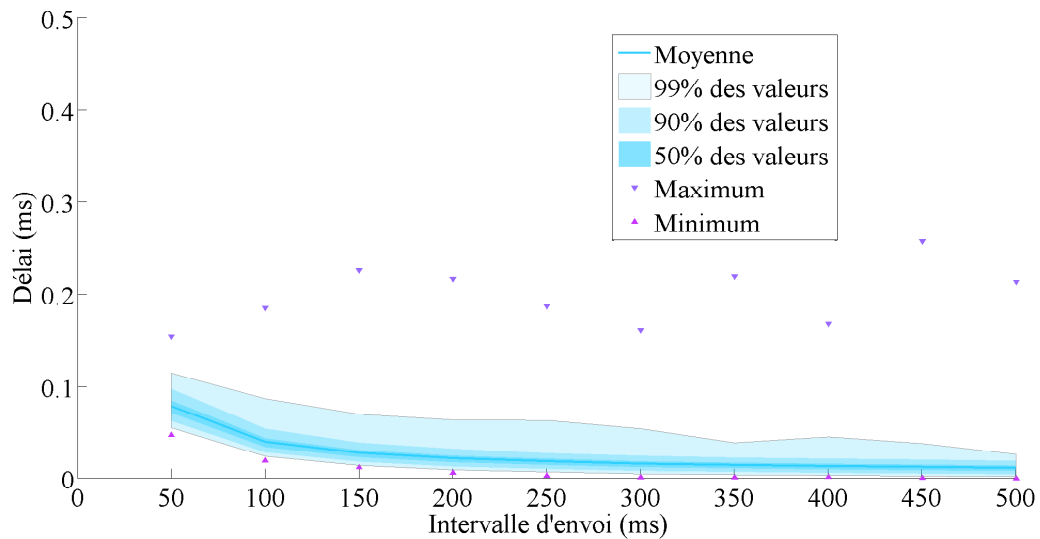


FIGURE 5.28 – Délai d'accès au médium (Ma8) dans le cas du scénario S12 avec l'application de perception élargie

accrue des ressources de communication pour l'application de perception élargie. En traçant la droite de la fonction identité (ligne pointillée), on peut identifier l'intervalle minimal d'attente entre deux paquets applicatifs afin que le délai moyen de bout en bout du réseau soit inférieur à cet intervalle. Cette valeur est de 175 ms. Ainsi dans la configuration d'étude, en envoyant les paquets applicatifs EPM toutes les 175 ms, on s'assure que ces paquets soient reçus à la fin d'un délai inférieur à 175 ms, c'est à dire avant l'envoi d'un nouveau paquet.

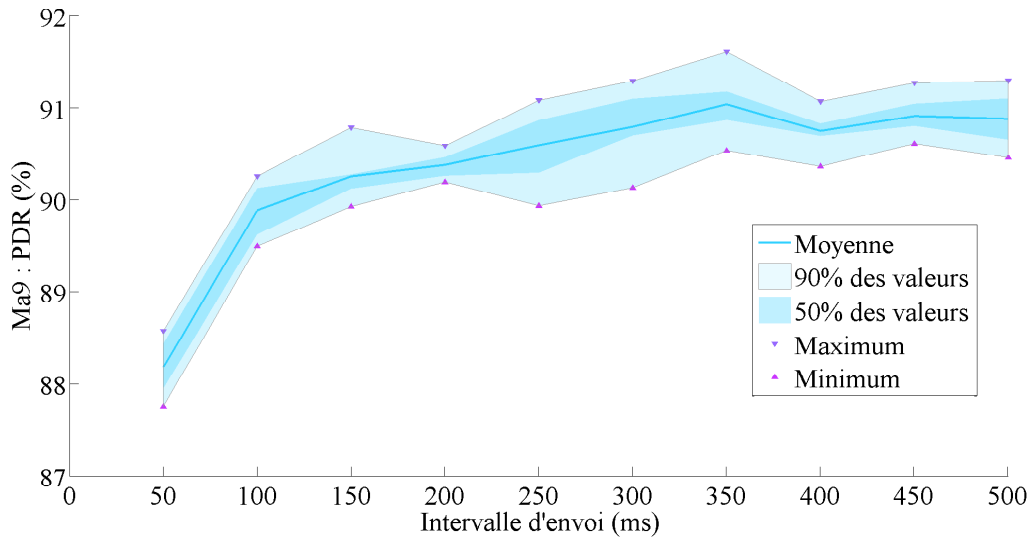


FIGURE 5.29 – PDR (Ma9) dans le cas du scénario S12 avec l'application de perception élargie

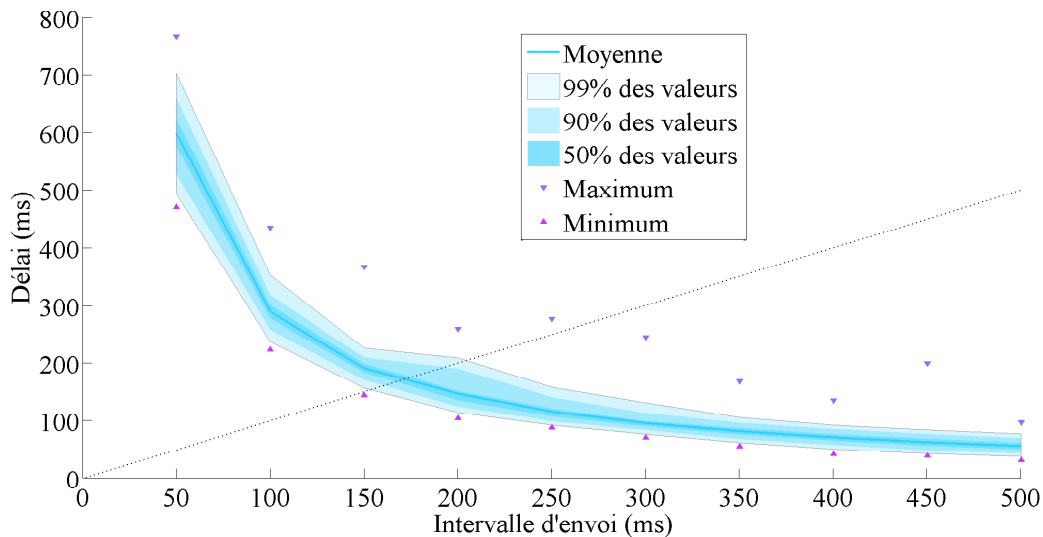


FIGURE 5.30 – Délai applicatif (Ma10) dans le cas du scénario S12 avec l'application de perception élargie

5.4 Conclusion

Ce chapitre a eu pour objet trois évaluations qui intègrent différents modèles de trafic applicatif dans le but de mesurer les performances système de CBL-OLSR.

La première évaluation s'est focalisée sur l'étude comparative de CBL-OLSR aux protocoles AODV, DSR, OLSR, QOLSR et GRP de la littérature avec un trafic applicatif dont le débit est conforme aux recommandations de l'IEEE : des messages de 300 octets envoyés à une fréquence de 10 Hz. Notre proposition CBL-OLSR montre les meilleures performances en termes de délai WLAN et d'accès au médium, ainsi que sur le nombre de retransmissions des messages. Elle limite le trafic de routage envoyé et reçu en libérant de la bande passante au profit du trafic applicatif.

La deuxième évaluation s'est intéressée à une application de localisation distribuée. Nous avons traduit les besoins de communication en flux de trafic applicatif en tenant compte des particularités structurelles du schéma CBL. Les nœuds branche sont les seuls nœuds de la structure à exécuter l'application CMM et DDGPS et à recevoir les informations de leur(s) nœud(s) feuille ainsi qu'à les retransmettre. Les résultats de l'évaluation de performance, réalisée avec un scénario de trafic routier réel et la densité maximale de véhicules observées sur une journée, montrent que le délai de bout en bout fourni à l'application distribuée est raisonnable, même avec une charge de trafic élevée, et que le taux de paquet reçu reste supérieur à

93% même avec un intervalle d'envoi des messages applicatifs très contraint de l'ordre de 50 ms. Dans la configuration testée, si l'on souhaite recevoir les paquets applicatifs avant la création d'un nouveau paquet, il faut choisir un intervalle d'envoi à partir de 125 ms.

La troisième et dernière évaluation s'appuie sur le trafic applicatif d'une application de perception élargie dont nous avons également traduit les besoins de communication en flux de trafic applicatif. Nous avons adapté l'application de perception élargie à la structure formée par notre proposition CBL-OLSR afin de limiter au maximum l'envoi des messages applicatifs. Les nœuds branche de la structure sont les seuls à fusionner les informations reçues de leur nœud feuille et à les retransmettre. Nous avons vérifié que notre proposition CBL-OLSR adapté au contexte véhiculaire supporte le trafic applicatif de la perception élargie, pour un intervalle d'envoi des paquets à partir de 175 ms. Dans le cas contraire, le taux de paquets reçus est inférieur à 90% et les délais de bout en bout peuvent atteindre des valeurs supérieures à la fréquence d'envoi.

Conclusion et perspectives

Conclusion

Avec le futur déploiement des véhicules autonomes et communicants, les besoins en communication deviendront de plus en plus importants. Les contraintes des applications véhiculaires, la nature du réseau VANET et ses caractéristiques intrinsèques exigeront de disposer d'un réseau VANET réactif et adaptatif aux changements fréquents de topologie. Ainsi, les travaux de recherche que nous avons menés et présentés dans ce manuscrit s'attachent à spécifier et évaluer une nouvelle proposition d'organisation du réseau permettant d'aider les décisions de routage en assurant un service de communication adéquat.

Dans ce manuscrit, nous avons tout d'abord réalisé un état de l'art des différentes stratégies utilisées par les protocoles de routage adaptés aux VANETs. De l'intérêt porté aux communications véhiculaires émergent de nombreux défis auxquels la communauté scientifique fait face en proposant des solutions innovantes. Nous avons vu que beaucoup d'efforts ont été réalisés ces dernières années afin de développer des protocoles de routage adaptés aux réseaux véhiculaires. Un nombre conséquent de protocoles a vu le jour. La solution que nous proposons dans cette thèse n'est pas de définir un nouveau protocole de routage, mais plutôt de proposer une organisation du réseau qui pourra être mise en œuvre au sein de protocoles déjà existants, quelle que soit la technologie de communication sans fil utilisée.

Puis, nous avons cherché à améliorer la stabilité temporelle de la structure proposée. Le critère de temps de connexion entre les nœuds du réseau nous a permis d'atteindre cet objectif. Nous avons ainsi spécifié une nouvelle méthode d'organisation du réseau nommée *Chain Branch Leaf* (CBL), basée sur la distinction de deux types de nœuds : les nœuds branche et les nœuds feuille ainsi que sur la formation d'une chaîne de véhicules par sens de circulation du trafic routier. Cette chaîne de véhicules permet de limiter la propagation des messages, soit en amont du trafic routier, soit en aval. Cette proposition a fait l'objet d'une deuxième version, permettant de corriger certains manquements et de renforcer la stabilité de la structure en améliorant la gestion des chaînes. CBL a fait l'objet d'une intégration dans le protocole de routage OLSR, bien connu des MANETs. Cette intégration nous a permis de spécifier un nouveau protocole de routage adapté aux réseaux VANETs : CBL-OLSR.

Ensuite, nous avons décrit les constituants d'un cahier des charges dédié à l'étude de performances d'un protocole de routage pour les VANETs, en comparant les méthodes adoptées dans les études publiées. Nous avons montré qu'aucun consensus à l'heure actuelle n'existait sur les modèles ou simulateurs à adopter en vue d'évaluer un protocole de routage. Les résultats des évaluations des différents protocoles de la littérature peuvent donc difficilement être comparés entre eux de manière rigoureuse. Il est très difficile d'évaluer un protocole de routage par rapport à un autre. Nous avons choisi un cahier des charges pour les évaluations en concordance avec nos besoins. Nous souhaitons évaluer CBL dans le cas d'un réseau autoroutier. Nous avons utilisé en premier lieu le simulateur de mobilité microscopique SUMO permettant de générer des traces de véhicules pour des scénarios autoroutiers. Nous avons employé par la suite le logiciel MATLAB pour une analyse structurelle simplifiée de notre proposition. Nous avons ensuite utilisé le logiciel OPNET Riverbed Modeler pour une modélisation globale du système de communication.

Nous avons alors cherché à comparer les performances structurelles de CBL-OLSR et de OLSR. L'analyse des résultats structurels de CBL-OLSR montre clairement l'efficacité de ce protocole pour les réseaux véhiculaires. Dans les scénarios avec une densité forte de véhicules, seul 20% des nœuds du réseau sont des

nœuds branche relayant le trafic. Les nœuds restent dans l'état branche plus de 50 secondes en moyenne, tandis que les liens feuille-branche perdurent en moyenne durant 35 secondes dans le cas d'une durée de traversée d'une section autoroutière de 150 secondes en moyenne. Ainsi, en implémentant notre solution adaptée au contexte des VANETs à un protocole initialement conçu pour les MANETs, nous arrivons à des résultats permettant de proposer un service de communication effectivement stable.

L'évaluation des performances de CBL-OLSR avec un trafic applicatif a été décomposée en deux parties. Dans la première partie nous avons étudié CBL-OLSR comparativement à cinq autres protocoles de la littérature en utilisant des besoins applicatifs standardisés par l'IEEE. Nous avons montré la pertinence de notre proposition pour diminuer la charge de trafic, le taux de retransmission et le délai. Dans la seconde partie nous avons étudié en détail deux applications requérant l'usage de VANETs : la localisation coopérative et la perception élargie. Nous avons évalué leurs besoins en communication afin de les adapter aux services de transmission offerts par notre proposition CBL-OLSR. Cette adaptation a permis de limiter à la fois le nombre et la taille des messages applicatifs à envoyer. Nous avons étudié les performances fournies par CBL-OLSR en termes de délai et de taux de paquets reçus pour différentes fréquences d'envoi des messages dans un intervalle de 2 Hz à 20 Hz, avec un scénario de mobilité composé d'un réseau routier réel comportant une section de l'autoroute A27 et d'une densité de véhicules relevée durant l'heure de pointe du matin. Nous avons montré que le délai de bout en bout était inférieur à 300 ms à une fréquence d'envoi de 20 Hz pour l'application de localisation coopérative et inférieur à 100 ms si la fréquence est inférieure à 5 Hz. Pour cette application, le taux de paquets reçus est toujours supérieur à 93%. L'application de perception élargie utilisant plus de ressources de communication, les résultats sont moins performants. Le délai est inférieur à 200 ms pour une fréquence d'envoi de 5 Hz tandis que le taux de paquets reçus n'est que de 88% pour une fréquence de 20 Hz. Il augmente à 91% dès que la fréquence d'envoi est supérieure à 3 Hz.

Perspectives

Les travaux présentés dans ce manuscrit ouvrent de nombreuses perspectives. Des pistes d'amélioration ou d'évolution de CBL pourraient être les suivantes :

Une troisième version de CBL pourrait être envisagée qui prendrait en compte des métriques de qualité de service dans les choix d'élection des nœuds branche et des nœuds de la chaîne. Cette proposition pourrait être appelée *Quality of Service - Chain Branch Leaf* (QoS-CBL). L'algorithme de structuration du réseau pourrait redonder les choix de nœuds branche aval et amont de la chaîne ainsi que le choix du nœud branche élu par un nœud feuille. Cette redondance permettrait de faire face à la rupture d'une liaison entre deux nœuds ou l'arrêt total d'un nœud communicant (panne...). Cette version pourrait également être adaptée au milieu péri-urbain voire urbain en tenant compte des futurs changements de directions d'un nœud à l'approche d'une intersection. Cette prise en compte pourrait être effectuée en connaissant l'itinéraire du véhicule (et sa destination finale). Cette information pourrait être ajoutée au sein des messages HELLO. Elle pourrait être intégrée au calcul prédictif du temps de connexion et du temps de chaîne. Ainsi un nœud changera son choix de nœud branche amont ou aval s'il reçoit l'information que ce nœud ne va pas dans la même direction que lui à la prochaine intersection.

Une autre piste concerne la mise en œuvre de CBL au sein d'autres protocoles de routage, que ce soit des protocoles de routage adaptés aux MANETs ou aux VANETs. Cette intégration est même envisageable pour des protocoles de routage réactif. Pour ces protocoles, s'ils ne sont pas déjà spécifiés, il sera nécessaire de définir l'envoi périodique de messages HELLO nécessaire au fonctionnement de la structure CBL. Par exemple, pour le protocole de routage conçu pour les MANETs et bien connu, AODV pourrait utiliser la structure de chaîne pour l'envoi des requêtes RREQ/RREP. Ces requêtes ne seraient plus envoyées par tous les nœuds, mais uniquement relayées par les nœuds branche, limitant ainsi le trafic de routage. Cette proposition pourrait être appelée CBL-AODV. Pour les protocoles de routage géographiques, les nœuds branche pourraient servir de serveur de localisation. Si ces protocoles utilisent l'envoi de requêtes pour connaître la position du ou des nœuds destinataires, la structure de chaîne permettrait de limiter le relayage à tout le réseau.

Une autre idée concerne l'étude d'une architecture hybride V2V et V2I avec la structure CBL. Dans ce cas, les bornes RSUs déployées le long des routes seraient des nœuds branche. En l'absence de bornes, la structure CBL continuerait de fonctionner tel que nous l'avons définie dans ce manuscrit. Cette proposition pourrait être appelée *Hybrid - Chain Branch Leaf* (H-CBL) (figure 5.31).

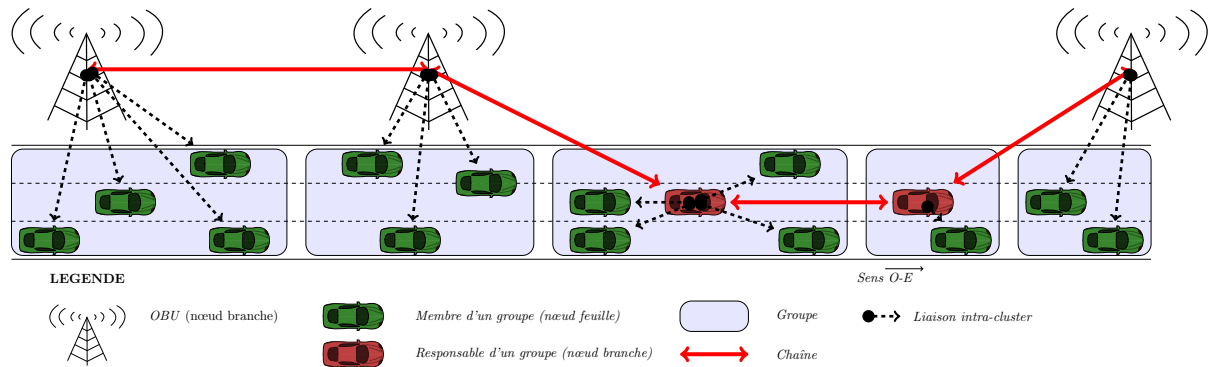


FIGURE 5.31 – Proposition d'une structure hybride V2V et V2I : H-CBL

Par ailleurs, la conception modulaire de CBL permet d'imaginer l'utilisation d'autres technologies que le 802.11p. En effet, nous avons vu que cette technologie pourrait être améliorée en particulier au niveau de son algorithme d'accès au médium (CSMA/CA) qui, avec une forte densité de nœuds, génère des temps d'accès au médium très importants.

Également, nous souhaiterions étudier les performances de CBL avec d'autres applications des VANETs en les adaptant à la structure de chaîne, car le système global devra supporter un ensemble d'applications caractérisées par des attentes en communication différentes. Nous pourrions également étudier les performances de CBL avec d'autres situations de mobilité, d'autres modèles de propagation plus réalistes et avec un simulateur de communication bi-directionnel permettant d'impacter la trajectoire des véhicules en fonction des messages reçus.

Les aspects qualité de service ont été en partie explorés. Les résultats de comparaisons intégrant QOLSR ont montré que cette proposition antérieure apporte réellement une amélioration de la qualité des liaisons dans le cadre du protocole OLSR. La démonstration faite que CBL pouvait être avantageusement intégrée à OLSR permet d'envisager une intégration aussi immédiate de CBL dans QOLSR pour supporter des applications imposant des exigences fortes en termes de débit et de délai. Ces travaux seront menés dans un avenir proche.

Pour finir, les aspects relatifs à la sécurité de l'information dans le protocole de routage n'ont pas été étudiés durant cette thèse. CBL héritera automatiquement des mécanismes de sécurité du protocole auquel il est intégré. Néanmoins, des travaux menés parallèlement à ma thèse au sein de l'Université du Littoral et Côte d'Opale s'intéressent à la preuve formelle des propriétés des protocoles de communications dédiés aux systèmes de transport, notamment celles liées à la qualité de service et à la sécurité. Patrick Sondi, co-encadrants de ma thèse, a entrepris l'application des mécanismes développés à la mise en œuvre de la sécurité dans CBL-OLSR. Au stade actuel de ces travaux, l'étude [206], réalisée dans le contexte du travail d'Emna Chebbi, en thèse à l'ULCO, a été menée sur une modélisation de la version 1 de notre proposition CBL-OLSR. Ils se poursuivront par la mise en œuvre effective de mécanismes de sécurité et l'étude des preuves formelles associées, toujours en collaboration avec l'IFSTTAR.

Bibliographie

ACL : Articles dans des revues internationales ou nationales avec comité de lecture répertoriées

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “Chain-Branch-Leaf : a Clustering Scheme for Vehicular Networks Using Only V2V Communications”, *Journal, Ad Hoc Networks*, vol. 68, pp. 70-84, January 2018, disponible en ligne depuis le 10/10/2017, doi : 10.1016/j.adhoc.2017.10.007.

ACLN : Articles dans des revues avec comité de lecture non répertoriées

Patrick Sondi, Martine Wahl, **Lucas Rivoirard** and Ouafae Cohin, “Performance Evaluation of 802.11p-Based Ad Hoc Vehicle-to-Vehicle Communications for Usual Applications Under Realistic Urban Mobility”, *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7 no. 5, Jan 2016.

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “Using Real-World Car Traffic Dataset in Vehicular Ad Hoc Network Performance Evaluation”, *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 12, pp. 390–398, Dec 2016.

ACTI : Communications avec actes dans un congrès international

Christine Buisson, Daniel Villegas, and **Lucas Rivoirard**, “Using Polar Coordinates to Filter Trajectories Data without Adding Extra Physical Constraint”, 95th Transportation Research Board, 15 p, 10-14 Jan 2016, Washington, United States (TRB).

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “Performance evaluation of AODV, DSR, GRP and OLSR for VANET with real-world trajectories”, 15th International Conference on ITS Telecommunications, 29-31 May 2017, Warsaw, Poland (ITST).

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “CBL : A Clustering Scheme for VANETs”, the Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications, 24-27 Jul 2017, Nice, France (VEHICULAR).

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “From Multipoint Relaying to Chain-Branch-Leaf : Improving the Clustering in OLSR for Vehicular Ad hoc Networks”, the 24th IEEE SCVT symposium, 14 Nov 2017, Leuven, Belgium (SCVT).

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “A new clustering structure for VANET”, the 7th Transport Research Arena, 16-19 Apr 2018, Vienna, Austria (TRA).

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Dominique Gruyer, and Marion Berbineau, “A Cooperative Vehicle Ego-localization Application Using V2V Communications with CBL Clustering”, 2018 IEEE Intelligent Vehicles Symposium, 26-29 Jun 2018, Changshu, China (IV).

Emna Chebbi, Patrick Sondi, Eric Ramat, **Lucas Rivoirard**, Martine Wahl, “Simulation of a Clustering Scheme for Vehicular Ad Hoc Networks Using a DEVS-based Virtual Laboratory Environment”, The 9th International Conference on Ambient Systems, Networks and Technologies, May 8-11, 2018, Porto, Portugal (ANT).

Patrick Sondi, **Lucas Rivoirard**, Martine Wahl, “Performance Evaluation of Vehicle-to-Vehicle Communications for a Collective Perception Application in Vehicular Ad Hoc Networks”, IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 9-12 Sep 2018, Bologna, Italy (PIMRC).

ACTN : Communications avec actes dans un congrès national

Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer, “Proposition d’une structuration dynamique d’un réseau de communication intervéhiculaire pour les ITS”, Rencontres de la Mobilité intelligente, 24-25 Jan 2018, Paris, France (ATEC ITS).

Références bibliographiques

- [1] AT&T. <https://policyforum.att.com/innovation/5g/>. [En ligne], (consulté le 11 juin 2018).
- [2] Ieee standards for local and metropolitan area networks : Overview and architecture : Ieee 802-1990, 1990. <https://ieeexplore.ieee.org/document/467387/>.
- [3] Ministère de la Transition écologique et solidaire. Projet scoop. <http://www.scoop.developpement-durable.gouv.fr/>, 2018.
- [4] Commission Européenne. Projet intercor. <http://intercor-project.eu/>, 2018.
- [5] Holger Füßler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, and Dieter Vollmer. A comparison of routing strategies for vehicular ad hoc networks. *Technical reports*, 2, 2002. <https://doi.org/10.1145/173682.165177>.
- [6] *IEEE standard for Information technology - Telecommunications and information exchange between system - Local and metropolitan area networks - Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6 : Wireless Access in Vehicular Environments* : IEEE std 802.11p-2010, july 2010. <http://standards.ieee.org/findstds/standard/802.11n-2009.html>.
- [7] Lou Frenzel. The battle over V2V Wireless Technologies. <http://www.mwrf.com/systems/battle-over-v2v-wireless-technologies>, 2018.
- [8] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. Chain-branch-leaf : A clustering scheme for vehicular networks using only v2v communications. *Ad Hoc Networks*, 68 :70–84, 2018. <https://doi.org/10.1016/j.adhoc.2017.10.007>.
- [9] Taimoor Abbas, Katrin Sjöberg, Johan Karedal, and Fredrik Tufvesson. A measurement based shadow fading model for vehicle-to-vehicle network simulations. *International Journal of Antennas and Propagation*, 2015, 2015. <https://doi.org/10.1155/2015/190607>.
- [10] ETSI Technical Committee Intelligent Transport System. Intelligent Transport Systems (ITS); STDMA recommended parameters and settings for cooperative ITS; Access Layer Part. Technical Report ETSI TR 102 861 V1.1.1 (2012-01), 2012. http://www.etsi.org/deliver/etsi_tr/102800_102899/102861/01.01.01_60/tr_102861v010101p.pdf.
- [11] Mohsen Rohani, Denis Gingras, and Dominique Gruyer. A novel approach for improved vehicular positioning using cooperative map matching and dynamic base station dgps concept. *IEEE Transactions on Intelligent Transportation Systems*, 17(1) :230–239, 2016. <https://doi.org/10.1109/tits.2015.2465141>.
- [12] IDRAC Anne-Marie. Développement des véhicules autonomes - orientations stratégiques pour l'action publique. Technical report, Ministère de la transition écologique et solidaire, 2018. <http://www.ladocumentationfrancaise.fr/rapports-publics/184000287/index.shtml>.
- [13] Continental. Ces (consumer electronics show) - conférence janvier 2018. <https://www.continental-corporation.com/en/press/press-releases/ces-press-conference-118210>. [En ligne], (consulté le 11 juin 2018).
- [14] Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular Ad Hoc network. *Journal of Network and Computer Applications*, 37 :380–392, 2014. <https://doi.org/10.1016/j.jnca.2013.02.036>.
- [15] Nadia Haddadou. *Réseaux ad hoc véhiculaires : vers une dissémination de données efficace, coopérative et fiable*. PhD thesis, Paris Est, June 2014. <https://tel.archives-ouvertes.fr/tel-01124319/>.

- [16] Abhinay Mukunthan. *Routing and propagation in urban vehicular ad hoc networks*. Thèse, University of Wollongong, Australie, January 2015. <http://ro.uow.edu.au/theses/4558/>.
- [17] Baraa T. Sharef, Raed A. Alsaqour, and Mahamod Ismail. Vehicular communication ad hoc routing protocols : A survey. *Journal of Network and Computer Applications*, 40 :363–396, April 2014. <https://doi.org/10.1016/j.jnca.2013.09.008>.
- [18] Amit Dua, Neeraj Kumar, and Seema Bawa. A systematic review on routing protocols for Vehicular Ad Hoc Networks. *Vehicular Communications*, 1(1) :33–52, January 2014. <https://doi.org/10.1016/j.vehcom.2014.01.001>.
- [19] Felipe Domingos da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio A. F. Loureiro. Data Communication in VANETs : A Survey, Challenges and Applications. Technical Report 8498, INRIA Saclay ; INRIA, March 2014. <https://doi.org/10.1016/j.adhoc.2016.02.017>.
- [20] G. Mary Valentina and S. Jayashri. A Systematic Survey of Vanet Routing Protocols based on Transmission Strategies. *Indian Journal of Science and Technology*, 9(22), June 2016. <https://doi.org/10.17485/ijst/2016/v9i22/54714>.
- [21] Jinqi Zhu, Ming Liu, Yonggang Wen, Chunmei Ma, and Bin Liu. Parking Backbone : Toward Efficient Overlay Routing in VANETs. *International Journal of Distributed Sensor Networks*, 2014 :1–13, 2014. <https://doi.org/10.1155/2014/291308>.
- [22] J. Jakubiak and Y. Koucheryavy. State of the Art and Research Challenges for VANETs. pages 912–916, January 2008. <https://doi.org/10.1109/ccnc08.2007.212>.
- [23] J. Chennikara-Varghese, Wai Chen, O. Altintas, and Shengwei Cai. Survey of Routing Protocols for Inter-Vehicle Communications. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems : Networking Services*, pages 1–5, July 2006. <https://doi.org/10.1109/MOBIQ.2006.340432>.
- [24] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. Using real-world car traffic dataset in vehicular ad hoc network performance evaluation. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(12) :p390–398, 2016. <https://doi.org/10.14569/IJACSA.2016.071251>.
- [25] Ines Ben Jemaa. *Multicast Communications for Cooperative Vehicular Systems*. PhD thesis, Mines ParisTech, 2014. <https://hal.inria.fr/tel-01101679>.
- [26] Sun Xi and Xia-Miao Li. Study of the Feasibility of VANET and its Routing Protocols. pages 1–4. IEEE, October 2008. <https://doi.org/10.1109/WiCom.2008.1021>.
- [27] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *ACM SIGCOMM computer communication review*, volume 24, pages 234–244. ACM, 1994. <https://doi.org/10.1145/190314.190336>.
- [28] S Mohapatra and P Kanungo. Performance analysis of aodv, dsr, olsr and dsdv routing protocols using ns2 simulator. *Procedia Engineering*, 30 :69–76, 2012. <https://doi.org/10.1016/j.proeng.2012.01.835>.
- [29] Shouzhi Xu, Pengfei Guo, Bo Xu, and Huan Zhou. Qos evaluation of vanet routing protocols. *Journal of Networks*, 8(1) :132–139, 2013. <https://doi.org/10.4304/jnw.8.1.132-139>.
- [30] Mayada Abdelgadir, Rashid A Saeed, and Abuagla Babiker. Mobility routing model for vehicular ad-hoc networks (vanets), smart city scenarios. *Vehicular Communications*, 9 :154–161, 2017. <https://doi.org/10.1016/j.vehcom.2017.04.003>.
- [31] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol (olsr). Technical Report RFC 3626, 2003. <https://doi.org/10.17487/RFC3626>.
- [32] Véronique Vèque and Colette Johnen. Hiérarchisation dans les réseaux ad hoc de véhicules. In Rouillon-Couture N. & al. Roose Ph., editor, *8èmes journées francophones Mobilité et Ubiquité, UBIMOB 2012*, pages 45–52. CEPADUES, June 2012. <https://hal.archives-ouvertes.fr/hal-00781267>.

- [33] Rasmeet Bali, Neeraj Kumar, and Joel J.P.C. Rodrigues. Clustering in vehicular ad hoc networks : Taxonomy, challenges and solutions. *Vehicular Communications*, 1(3) :134–152, June 2014. <https://doi.org/10.1016/j.vehcom.2014.05.004>.
- [34] Saleha Mubarak AlMheiri and Hend Saeed AlQamzi. MANETs and VANETs clustering algorithms : A survey. In *8th IEEE GCC Conference and Exhibition (GCCCE)*, pages 1–6, February 2015. <https://doi.org/10.1109/IEEEGCC.2015.7060048>.
- [35] Craig Cooper, Daniel Franklin, Montserrat Ros, Farzad Safaei, and Mehran Abolhasan. A comparative survey of vanet clustering techniques. *IEEE Communications Surveys & Tutorials*, 19(1) :657–681, 2017. <https://doi.org/10.1109/COMST.2016.2611524>.
- [36] Mario Gerla and Jack Tzu-Chieh Tsai. Multicenter, mobile, multimedia radio network. *Wireless networks*, 1(3) :255–265, 1995. <https://doi.org/10.1007/BF01200845>.
- [37] Fabian Garcia Nocetti, Julio Solano Gonzalez, and Ivan Stojmenovic. Connectivity based k-hop clustering in wireless networks. *Telecommunication systems*, 22(1-4) :205–220, January 2003. <https://doi.org/10.1109/HICSS.2002.994183>.
- [38] Prithwish Basu, Prithwish Basu, Naved Khan, and Thomas D. C. Little. A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks. *International workshop on wireless networks and mobile computing*, pages 413–418, 2001. <https://doi.org/10.1109/CDCS.2001.918738>.
- [39] Sanjay K Dhurandher and GV Singh. Weight based adaptive clustering in wireless ad hoc networks. In *Personal Wireless Communications, 2005. ICPWC 2005. 2005 IEEE International Conference on*, pages 95–100. IEEE, 2005. <https://doi.org/10.1109/ICPWC.2005.1431309>.
- [40] Orhan Dagdeviren and Kayhan Erciyes. A distributed backbone formation algorithm for mobile ad hoc networks. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 219–230. Springer, December 2006. https://doi.org/10.1007/11946441_24.
- [41] ETSI Technical Committee Intelligent Transport System. Intelligent Transport Systems (ITS) ; Vehicular Communications ; GeoNetworking ; Part 6 : Internet Integration ; Sub-part 1 : Transmission of IPv6 Packets over GeoNetworking Protocols. Technical Report ETSI TS 102 636-6-1 V1.1.1 (2011-03), 2011. https://archive.org/details/etsi_ts_102_636-6-1_v01.01.01.
- [42] ETSI Technical Committee Intelligent Transport System. Intelligent Transport Systems (ITS) ; Vehicular Communications ; GeoNetworking ; Part 5 : Transport Protocols ; Sub-part 1 : Basic Transport Protocol. Technical Report ETSI TS 102 636-5-1 V1.1.1 (2011-02), 2011. https://archive.org/details/etsi_ts_102_636-5-1_v01.01.01.
- [43] Stephan Olariu and Michele C. Weigle. *Vehicular Networks : From Theory to Practice*. Chapman & Hall/CRC, 1 edition, 2009. ISBN : 9781138116597.
- [44] David Johnson. The dynamic source routing protocol (dsrc) for mobile ad hoc networks for ipv4. Technical report, 2007. <https://doi.org/10.17487/RFC4728>.
- [45] Dmitri LEBEDEV. *Ad Hoc Networks : Study of Protocol Behaviour*. PhD thesis, X - Polytechnique, 2006. http://www.lix.polytechnique.fr/Labo/Dmitri.Lebedev/lebedev_these.pdf.
- [46] Marwane Ayaida, Hacène Fouchal, Lissan Afilal, and Yacine Ghamri-Doudane. A comparison of reactive, grid and hierarchical location-based services for vanets. In *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pages 1–5. IEEE, 2012. <https://doi.org/10.1109/VTCFall.2012.6398920>.
- [47] LABIOD Houada and BEYLOT André-Luc. *Réseaux véhiculaires : Modèles et algorithmes*. Lavoisier, September 2013. ISBN : 978-2-7462-8290-2.
- [48] J.J. Blum, A. Eskandarian, and L. Hoffman. Challenges of intervehicle ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4) :347–351, December 2004. <https://doi.org/10.1109/TITS.2004.838218>.
- [49] Souaad Boussoufa-Lahlah, Fouzi Semchedine, and Louiza Bouallouche-Medjkoune. Geographic routing protocols for vehicular ad hoc networks (vanets) : A survey. *Vehicular Communications*, 11 :20 – 31, 2018. <https://doi.org/10.1016/j.vehcom.2018.01.006>.
- [50] Stefano Basagni, Imrich Chlamtac, Violet R Syrotiuk, and Barry A Woodward. A distance routing effect algorithm for mobility (dream). In *Proceedings of the 4th annual ACM/IEEE international*

- conference on Mobile computing and networking, pages 76–84. ACM, 1998. <https://doi.org/10.1145/288235.288254>.
- [51] Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless networks*, 6(4) :307–321, 2000. <https://doi.org/10.1023/A:1019106118419>.
- [52] Valery Naumov and Thomas R Gross. Connectivity-aware routing (car) in vehicular ad-hoc networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1919–1927. IEEE, 2007. <https://doi.org/10.1109/INFCOM.2007.223>.
- [53] Li Zhiyuan. Geographic routing protocol and simulation. In *Computer Science and Engineering, 2009. WCSE'09. Second International Workshop on*, volume 2, pages 404–407. IEEE, 2009. <https://doi.org/10.1109/WCSE.2009.840>.
- [54] Brad Karp and Hsiang-Tsung Kung. Gpsr : Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000. <https://doi.org/10.1145/345910.345953>.
- [55] Vasil Hnatyshin and Hristo Asenov. Design and implementation of an opnet model for simulating geoadv manet routing protocol. In *Proc. of the OPNETWORK 2010 International Conference, Session : Wireless Ad Hoc and Wireless Personal Area Networks, Washington DC*, 2010.
- [56] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119. ACM, 2006. <https://doi.org/10.1145/1132905.1132918>.
- [57] Deling Huang, Yusong Yan, Chang Su, and Guangxia Xu. Prediction-based geographic routing over vanets. *Technical Journal of the Faculty of Engineering*, 39(2) :157–164, 2016. <https://doi.org/10.1016/j.comnet.2017.10.009>.
- [58] Nitin Maslekar, Mounir Boussedjra, Joseph Mouzna, and Houda Labiod. C-drive : clustering based on direction in vehicular environment. In *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, pages 1–5. IEEE, 2011. <https://doi.org/10.1109/NTMS.2011.5720653>.
- [59] Daxin Tian, Yunpeng Wang, Guangquan Lu, and Guizhen Yu. A vanets routing algorithm based on euclidean distance clustering. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, volume 1, pages V1–183. IEEE, 2010. <https://doi.org/10.1109/ICFCC.2010.5497808>.
- [60] Marco Di Felice, Luca Bedogni, and Luciano Bononi. Dynamic backbone for fast information delivery in vehicular ad hoc networks : an evaluation study. In *Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 1–8. ACM, 2011. <https://doi.org/10.1145/2069063.2069065>.
- [61] Wang Xiaonan and Qian Huanyan. Constructing a vanet based on cluster chains. *International Journal of Communication Systems*, 27(11) :2497–2517, 2014. <https://doi.org/10.1002/dac.2484>.
- [62] Zhigang Wang, Lichuan Liu, MengChu Zhou, and Nirwan Ansari. A position-based clustering technique for ad hoc intervehicle communication. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2) :201–208, 2008. <https://doi.org/10.1109/TSMCC.2007.913917>.
- [63] Vivek Sethi and Narottam Chand. A destination based routing protocol for context based clusters in vanet. *Communications and Network*, 9(03) :179, 2017. <https://doi.org/10.4236/cn.2017.93013>.
- [64] Véronique Vèque, Florent Kaisser, Colette Johnen, and Anthony Busson. Convoi : un protocole de formation de clusters pour les réseaux de véhicules. *Traité IC2*, pages 1–15, July 2013. <https://hal.inria.fr/hal-00821741>.
- [65] Emna Daknou, Mariem Thaalbi, and Nabil Tabbane. A fast clustering algorithm for vanets. In *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, pages 195–202. ACM, 2015. <https://doi.org/10.1145/2837126.2837147>.

- [66] Ruchi Mehra, Rasmeeet S Bali, and Prabhsharan Kaur. Efficient clustering based olsr routing protocol for vanet. In *Colossal Data Analysis and Networking (CDAN), Symposium on*, pages 1–7. IEEE, 2016. <https://doi.org/10.1109/CDAN.2016.7570915>.
- [67] Mohammed Amine Togou, Abdelhakim Hafid, and Pratap Kumar Sahu. A stable minimum velocity cds-based virtual backbone for vanet in city environment. In *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*, pages 510–513. IEEE, 2014. <https://doi.org/10.1109/LCN.2014.6925829>.
- [68] Zaydoun Y Rawashdeh and Syed Masud Mahmud. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *EURASIP Journal on Wireless Communications and Networking*, 2012(1) :15, 2012. <https://doi.org/10.1186/1687-1499-2012-15>.
- [69] Hamid Reza Arkian, Reza Ebrahimi Atani, and Saman Kamali. Fcvca : A fuzzy clustering-based vehicular cloud architecture. In *Communication Technologies for Vehicles (Nets4Cars-Fall), 2014 7th International Workshop on*, pages 24–28. IEEE, 2014. <https://doi.org/10.1109/Nets4CarsFall.2014.7000907>.
- [70] Meysam Azizian, Soumaya Cherkaoui, and Abdelhakim Senhaji Hafid. Dcev : A distributed cluster formation for vanet based on end-to-end relative mobility. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 287–291. IEEE, 2016. <https://doi.org/10.1109/iwcmc.2016.7577072>.
- [71] Celimuge Wu, Satoshi Ohzahata, and Toshihiko Kato. Can we generate efficient routes by using only beacons? backbone routing in vanets. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 2929–2934. IEEE, 2013. <https://doi.org/10.1109/pimrc.2013.6666648>.
- [72] Celimuge Wu, Satoshi Ohzahata, Yusheng Ji, and Toshihiko Kato. How to utilize interflow network coding in vanets : A backbone-based approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(8) :2223–2237, 2016. <https://doi.org/10.1109/tits.2016.2516027>.
- [73] Celimuge Wu, Satoshi Ohzahata, and Toshihiko Kato. Data dissemination with dynamic backbone selection in vehicular ad hoc networks. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–6. IEEE, 2013. <https://doi.org/10.1109/vtcfall.2013.6692152>.
- [74] Wei Fan, Yan Shi, Shanzhi Chen, and Longhao Zou. A mobility metrics based dynamic clustering algorithm for vanets. pages 752–756, 2011. <https://doi.org/10.1049/cp.2011.0769>.
- [75] Tsu-Wei Chen and Mario Gerla. Global state routing : A new routing scheme for ad-hoc wireless networks. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 1, pages 171–175. IEEE, 1998. <https://doi.org/10.1109/icc.1998.682615>.
- [76] Boon-Chong Seet, Genping Liu, Bu-Sung Lee, Chuan-Heng Foh, Kai-Juan Wong, and Keok-Kee Lee. A-star : A mobile ad hoc routing strategy for metropolis vehicular communications. In *International Conference on Research in Networking*, pages 989–999. Springer, 2004. https://doi.org/10.1007/978-3-540-24693-0_81.
- [77] Moez Jerbi, Rabah Meraihi, Sidi-Mohammed Senouci, and Yacine Ghamri-Doudane. Gytar : improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 88–89. ACM, 2006. <https://doi.org/10.1145/1161064.1161080>.
- [78] Zubair Amjad and Wang-Cheol Song. Road aware qos routing in vanets. In *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*, pages 133–138. IEEE, 2015. <https://doi.org/10.1109/apnoms.2015.7275415>.
- [79] Slawomir Kuklinski and Grzegorz Wolny. Density based clustering algorithm for vanets. In *Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pages 1–6. IEEE, 2009. <https://doi.org/10.1109/tridentcom.2009.4976256>.
- [80] Hamidreza Rahbar, Kshirasagar Naik, and Amiya Nayak. Dtsgr : Dynamic time-stable geocast routing in vehicular ad hoc networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, pages 1–7. IEEE, 2010. <https://doi.org/10.1109/medhocnet.2010.5546872>.

- [81] Muhammed Nur Avcil and Mujdat Soyuturk. Rescue : Relatively stable clustering for unbiased environments in vanets. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 1049–1055. IEEE, 2015. <https://doi.org/10.1109/iwcmc.2015.7289227>.
- [82] Hamid Reza Arkian, Reza Ebrahimi Atani, Atefe Pourkhalili, and Saman Kamali. A stable clustering scheme based on adaptive multiple metric in vehicular ad-hoc networks. *Journal of Information Science & Engineering*, 31(2) :361–386, 2015.
- [83] Kevin C Lee, Michael Le, Jerome Harri, and Mario Gerla. Louvre : Landmark overlays for urban vehicular routing environments. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE, 2008. <https://doi.org/10.1109/vetecf.2008.447>.
- [84] Yong Xiang, Zheng Liu, Ruilin Liu, Weizhen Sun, and Wei Wang. Geosvr : A map-based stateless vanet routing. *Ad Hoc Networks*, 11(7) :2125–2135, 2013. <https://doi.org/10.1109/anthology.2013.6784953>.
- [85] Tao Song, Weiwei Xia, Tiecheng Song, and Lianfeng Shen. A cluster-based directional routing protocol in vanet. In *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, pages 1172–1175. IEEE, 2010. <https://doi.org/10.1109/icct.2010.5689132>.
- [86] Yuyi Luo, Wei Zhang, and Yangqing Hu. A new cluster based routing protocol for vanet. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, volume 1, pages 176–180. IEEE, 2010. <https://doi.org/10.1109/NSWCTC.2010.48>.
- [87] Mohamed Hadded, Paul Muhlethaler, Anis Laouiti, and Leila Azzouz Saidane. A novel angle-based clustering algorithm for vehicular ad hoc networks. In *Vehicular Ad-Hoc Networks for Smart Cities*, pages 27–38. Springer, 2017. https://doi.org/10.1007/978-981-10-3503-6_3.
- [88] Ameneh Daeinabi, Akbar Ghaffar Pour Rahbar, and Ahmad Khademzadeh. Vwca : An efficient clustering algorithm in vehicular ad hoc networks. *Journal of Network and Computer Applications*, 34(1) :207–222, 2011. <https://doi.org/10.1016/j.jnca.2010.07.016>.
- [89] Si-Ho Cha, Jong-Eon Lee, and Minwoo Ryu. Directed broadcasting with mobility prediction for vehicular sensor networks. *International Journal of Distributed Sensor Networks*, 12(7), 2016. <https://doi.org/10.1177/1550147716657930>.
- [90] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. Vmasc : Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 2381–2386. IEEE, 2013. <https://doi.org/10.1109/wcnc.2013.6554933>.
- [91] Samira Harrabi, Ines Ben Jaffar, and Khaled Ghedira. Novel optimized routing scheme for vanets. *Procedia Computer Science*, 98 :32–39, 2016. <https://doi.org/10.1016/j.procs.2016.09.008>.
- [92] Zineb Squalli Houssaini, Imane Zaimi, Mohammed Oumsis, and Saïd El Alaoui Ouatik. Gpsr+ predict : An enhancement for gpsr to make smart routing decision by anticipating movement of vehicles in vanets. *Adv. Sci. Technol. Eng. Syst. J.*, 2(3) :137–146, 2017. <https://doi.org/10.25046/aj020318>.
- [93] Jing Zhao and Guohong Cao. Vadd : Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE transactions on vehicular technology*, 57(3) :1910–1922, 2008. <https://doi.org/10.1109/infocom.2006.298>.
- [94] Congyi Liu and Chunxiao Chigan. Rpb-md : Providing robust message dissemination for vehicular ad hoc networks. *Ad Hoc Networks*, 10(3) :497–511, 2012. <https://doi.org/10.1016/j.adhoc.2011.09.003>.
- [95] Evandro Souza, Ioanis Nikolaidis, and Pawel Gburzynski. A new aggregate local mobility (alm) clustering algorithm for vanets. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010. <https://doi.org/10.1109/icc.2010.5501789>.
- [96] Mildred M Caballeros Morales, Choong Seon Hong, and Young-Cheol Bang. An adaptable mobility-aware clustering algorithm in vehicular networks. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1–6. IEEE, 2011. <https://doi.org/10.1109/apnoms.2011.6077004>.

- [97] Mounir Frikha. *Réseaux ad hoc : routage, qualité de service et optimisation*. Performance des réseaux. Hermes Science Publications, 1 edition, January 2010. ISBN : 9782746221239.
- [98] Cheikh Sarr. *De l'apport d'une évaluation précise des ressources pour la Qualité de Service des réseaux ad hoc basés sur IEEE 802. 11*. PhD thesis, Villeurbanne, INSA, 2007. <http://www.theses.fr/2007ISAL0046>.
- [99] Ramesh Babu Kalivaradhan and Arunkumar Thangavelu. A simulated modeling approach towards providing adaptive qos for vehicular safety services over vanet. *International Journal of Research and Reviews in Computer Science*, 1(4) :110, 2010.
- [100] Salim Bitam and Abdelhamid Mellouk. Qos swarm bee routing protocol for vehicular ad hoc networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011. <https://doi.org/10.1109/icc.2011.5963424>.
- [101] Guangyu Li, Lila Boukhatem, and Steven Martin. An intersection-based qos routing in vehicular ad hoc networks. *Mobile Networks and Applications*, 20(2) :268–284, 2015. <https://doi.org/10.1007/s11036-015-0577-4>.
- [102] Gongjun Yan, Danda B Rawat, and Bhed Bahadur Bista. Provisioning vehicular ad hoc networks with quality of service. *International Journal of Space-Based and Situated Computing*, 2(2) :104–111, 2012. <https://doi.org/10.1504/ijssc.2012.047467>.
- [103] Pratap Kumar Sahu, Eric Hsiao-Kuang Wu, Jagruti Sahoo, and Mario Gerla. Bahg : Back-bone-assisted hop greedy routing for vanet's city environments. *IEEE Transactions on Intelligent Transportation Systems*, 14(1) :199–213, 2013. <https://doi.org/10.1109/tits.2012.2212189>.
- [104] Jingqing Zheng, Yi Wu, Zhexinx Xu, and Xiao Lin. A reliable routing protocol based on qos for vanet. In *Advanced Infocomm Technology (ICAIT), 2014 IEEE 7th International Conference on*, pages 21–28. IEEE, 2014. <https://doi.org/10.1109/icaait.2014.7019524>.
- [105] Tiguiane Yélémou. *Utilisation du taux d'erreur binaire pour améliorer la qualité de service dans les réseaux ad hoc*. Thèse de doctorat, Université de Poitiers. UFR des sciences fondamentales et appliquées, France, 2012. www.theses.fr/2012POIT2317.
- [106] Yongmei Sun, Shuyun Luo, Qijin Dai, and Yuefeng Ji. An adaptive routing protocol based on qos and vehicular density in urban vanets. *International Journal of Distributed Sensor Networks*, 11(6) :631092, 2015. <https://doi.org/10.1155/2015/631092>.
- [107] Yen-Wen Lin and Guo-Tang Huang. Optimal next hop selection for vanet routing. In *Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on*, pages 611–615. IEEE, 2012. <https://doi.org/10.1109/chinacom.2012.6417556>.
- [108] Mengying Ren, Lyes Khoukhi, Houda Labiod, Jun Zhang, and Véronique Vèque. A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (vanets). *Vehicular Communications*, 9 :233–241, 2017. <https://doi.org/10.1016/j.vehcom.2016.12.003>.
- [109] Ying Ge, Thomas Kunz, and Louise Lamont. Quality of service routing in ad-hoc networks using olsr. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2003. <https://doi.org/10.1109/hicss.2003.1174847>.
- [110] Hakim Badis and Khaldoun Al Agha. Qolsr multi-path routing for mobile ad hoc networks based on multiple metrics : bandwidth and delay. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 4, pages 2181–2184. IEEE, 2004. <https://doi.org/10.1109/vetecs.2004.1390660>.
- [111] Hakim Badis and Khaldoun Al Agha. Qolsr, qos routing for ad hoc wireless networks using olsr. *Transactions on Emerging Telecommunications Technologies*, 16(5) :427–442, 2005. <https://doi.org/10.1002/ett.1067>.
- [112] Dang-Quan Nguyen and Pascale Minet. Qos support and olsr routing in a mobile ad hoc network. In *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, pages 74–74. IEEE, 2006. <https://doi.org/10.1109/icniconsml.2006.178>.
- [113] Jérémie Leguay, Vania Conan, and Timur Friedman. Qos routing in olsr with several classes of service. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 420–425.

- [114] Patrick Sondi, Dhavy Gantsou, and Sylvain Lecomte. Performance evaluation of multimedia applications over an olsr-based mobile ad hoc network using opnet. In *Computer Modelling and Simulation (UKSim), 2010 12th International Conference on*, pages 567–572. IEEE, 2010. <https://doi.org/10.1109/uksim.2010.109>.
- [115] Omar Abdel Wahab, Hadi Otrok, and Azzam Mourad. Vanet qos-olsr : Qos-based clustering protocol for vehicular ad hoc networks. *Computer Communications*, 36(13) :1422–1435, 2013. <https://doi.org/10.1016/j.comcom.2013.07.003>.
- [116] Tianlei Shi, Neng Wang, and Peng Xiong. An On-Demand Virtual Backbone Broadcast Protocol for VANET. In *Information Science and Engineering (ICISE), 2009 1st International Conference on*, pages 3967–3972. IEEE, 2009. <https://doi.org/10.1109/ICISE.2009.287>.
- [117] Nicklas Beijar. Zone routing protocol (zrp). *Networking Laboratory, Helsinki University of Technology, Finland*, 9 :1–12, 2002. www.netlab.tkk.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf.
- [118] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing : A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70–74. IEEE, 2000. <https://doi.org/10.1109/icc.2000.853066>.
- [119] Kevin C Lee, Jérôme Härri, Uichin Lee, and Mario Gerla. Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios. In *Globecom Workshops, 2007 IEEE*, pages 1–10. IEEE, 2007. <https://doi.org/10.1109/glocomw.2007.4437832>.
- [120] Kashif Naseer Qureshi, Abdul Hanan Abdullah, and Ayman Altameem. Road aware geographical routing protocol coupled with distance, direction and traffic density metrics for urban vehicular ad hoc networks. *Wireless Personal Communications*, 92(3) :1251–1270, 2017. <https://doi.org/10.1007/s11277-016-3604-2>.
- [121] Hosein Bagherlou and Ali Ghaffari. A routing protocol for vehicular ad hoc networks using simulated annealing algorithm and neural networks. *The Journal of Supercomputing*, pages 1–25, 2018. <https://doi.org/10.1007/s11227-018-2283-z>.
- [122] Weihua Sun, Hirozumi Yamaguchi, Koji Yukimasa, and Shinji Kusumoto. Gvgrid : A qos routing protocol for vehicular ad hoc networks. In *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pages 130–139. IEEE, 2006. <https://doi.org/10.1109/iwqos.2006.250460>.
- [123] Carolina García-Costa, Esteban Egea-López, and Joan García-Haro. Evaluation of mac contention techniques for efficient geo-routing in vehicular networks. *Ad Hoc Networks*, 37 :44–62, 2016. <https://doi.org/10.1016/j.adhoc.2015.09.008>.
- [124] Martine Wahl. Les réseaux de terrain embarqués dans les transports guidés. *Synthèse INRETS(Arcueil)*, 2004. ISBN : 2-85782-591-9.
- [125] Martine Wahl. Survey of railway embedded network solutions : Towards the use of industrial ethernet technologies. *Synthèse INRETS. Mayenne, France. INRETS. ISBN 9782857826828*, S61 :104, 2010. ISBN : 978-2-85782-682-8.
- [126] ETSI Technical Committee Intelligent Transport System. Intelligent transport systems (its); vehicular communications; basic set of applications; part 2 : Specification of cooperative awareness basic service. Technical Report ETSI EN 302 367-2 V1.3.2 (2014-11), 2014. http://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.02_60/en_30263702v010302p.pdf.
- [127] ETSI Technical Committee Intelligent Transport System. Intelligent transport systems (its); vehicular communications; basic set of applications; part 3 : Specifications of decentralized environmental notification basic service. Technical Report ETSI EN 302 367-3 V1.2.2 (2014-11), 2014. http://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.02_60/en_30263702v010302p.pdf.
- [128] Intelligent Transport Systems (ITS); vehicular communications; geonetworking; part 2 :scenarios. Technical Report ETSI EN 302 636-2 V1.2.0 (2013-07), 2013. http://www.etsi.org/deliver/etsi_en/302600_302699/30263602/01.02.00_20/en_30263602v010200a.pdf.
- [129] Qin Lin, Changle Li, Xin Wang, and Lina Zhu. A three-dimensional scenario oriented routing protocol in vehicular ad hoc networks. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5. IEEE, 2013. <https://doi.org/10.1109/vtcspring.2013.6691848>.

- [130] Yong Li, Depeng Jin, Zhaocheng Wang, Lieguang Zeng, and Sheng Chen. Exponential and power law distribution of contact duration in urban vehicular ad hoc networks. *IEEE Signal Processing Letters*, 20(1) :110–113, 2013. <https://doi.org/10.1109/lsp.2012.2231412>.
- [131] Jesús Friginal, David de Andrés, Juan-Carlos Ruiz, and Miquel Martínez. A survey of evaluation platforms for ad hoc routing protocols : A resilience perspective. *Computer Networks*, 75 :395–413, 2014. <https://doi.org/10.1016/j.comnet.2014.09.010>.
- [132] Information Sciences Institute. Ns2 documentation. <https://www.isi.edu/nsnam/ns/>. [En ligne], (consulté le 11 juin 2018).
- [133] Mathworks. Matlab documentation. <https://www.mathworks.com/help/matlab/>. [En ligne], (consulté le 11 juin 2018).
- [134] OpenSim. Omnet++ documentation. <https://www.omnetpp.org/>. [En ligne], (consulté le 11 juin 2018).
- [135] Riverbed. Opnet documentation. <https://www.riverbed.com/fr/products/steelcentral/opnet.html>. [En ligne], (consulté le 11 juin 2018).
- [136] EstiNet. Nctuns. <http://nsl.cs.nctu.edu.tw/NSL/nctuns.html>. [En ligne], (consulté le 11 juin 2018).
- [137] Francisco J Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. A survey and comparative study of simulators for vehicular ad hoc networks (vanets). *Wireless Communications and Mobile Computing*, 11(7) :813–828, 2011. <https://doi.org/10.1002/wcm.859>.
- [138] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. Using real-world car traffic dataset in vehicular ad hoc network performance evaluation. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(12) :p390–398, 2016. <https://doi.org/10.14569/ijacsa.2016.071251>.
- [139] Patrick Sondi, Martine Wahl, Lucas Rivoirard, and Ouafae Cohin. Performance evaluation of 802.11 p-based ad hoc vehicle-to-vehicle communications for usual applications under realistic urban mobility. *International Journal of Advanced Computer Science and Applications, IJACSA*, 7(5) :p221–230, 2016. <https://doi.org/10.14569/ijacsa.2016.070532>.
- [140] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. Performance evaluation of aodv, dsr, grp and olsr for vanet with real-world trajectories. In *ITS Telecommunications (ITST), 2017 15th International Conference on*, pages 1–7. IEEE, 2017. <https://doi.org/10.1109/itst.2017.7972224>.
- [141] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. From multipoint relaying to chain-branch-leaf : Improving the clustering in olsr for vehicular ad hoc networks. In *Communications and Vehicular Technology (SCVT), 2017 IEEE Symposium on*, pages 1–5. IEEE, 2017. <https://doi.org/10.1109/scvt.2017.8240311>.
- [142] Lucas RIVOIRARD, Martine Wahl, Patrick SONDI, Marion Berbineau, and Dominique Gruyer. Proposition d’une structuration dynamique d’un réseau de communication intervéhiculaire pour les ITS. In *RDMI 2018, Les rencontres de la mobilité intelligente, ATEC ITS FRANCE*, page 11p, Paris, France, January 2018.
- [143] Lucas RIVOIRARD, Martine Wahl, Patrick SONDI, Marion Berbineau, and Dominique Gruyer. A new clustering structure for vanet. In *7th Transport Research Arena TRA 2018*, page 9p, Vienna, Austria, 2018.
- [144] Lucas RIVOIRARD, Martine Wahl, Patrick SONDI, Dominique Gruyer, and Marion Berbineau. A cooperative vehicle ego-localization application using v2v communications with cbl clustering. In *The 29th IEEE Intelligent Vehicles Symposium*, page 9p, Chang Shu, China, 2018. IEEE.
- [145] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. CBL : A Clustering Scheme for VANETs. In *VEHICULAR 2017-The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications*, pages pp–19, 2017. <https://hal.archives-ouvertes.fr/hal-01573625>.
- [146] Crowdad. <http://www.crowdad.org/>. [En ligne], (consulté le 11 juin 2018).

- [147] Mehdi Rafati Fard, Afshin Shariat Mohaymany, and Matin Shahri. A new methodology for vehicle trajectory reconstruction based on wavelet analysis. *Transportation Research Part C : Emerging Technologies*, 74 :150–167, 2017. <https://doi.org/10.1016/j.trc.2016.11.010>.
- [148] Christine Buisson, Daniel Villegas, and Lucas Rivoirard. Using polar coordinates to filter trajectories data without adding extra physical constraints. In *Transportation Research Board 95th Annual Meeting, Transportation Research Board*. Transportation Research Board, 2016. <https://trid.trb.org/view/1393268>.
- [149] S Hoogendoorn, H Van Zuylen, M Schreuder, B Gorte, and G Vosselman. Microscopic traffic data collection by remote sensing. *Transportation Research Record : Journal of the Transportation Research Board*, (1855) :121–128, 2003. <https://doi.org/10.3141/1855-15>.
- [150] FHWA Office of Operations. Ngsim : the next generation simulation. <http://ngsim-community.org/>. [En ligne], (consulté le 11 juin 2018).
- [151] Michel Ferreira, Hugo Conceição, Ricardo Fernandes, and Ozan K Tonguz. Urban connectivity analysis of vanets through stereoscopic aerial photography. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1–3. IEEE, 2009. <https://doi.org/10.1109/vetecf.2009.5378962>.
- [152] Atulya Mahajan, An-i Andy Wang, and Zhenhai Duan. Urban mobility models for vehicular ad hoc networks. *Florida State University*, 2006. http://purl.flvc.org/fsu/fd/FSU_migr_etd-2794.
- [153] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5) :483–502, 2002. <https://doi.org/10.1002/wcm.72>.
- [154] Neha Garg and Puneet Rani. An improved aodv routing protocol for vanet (vehicular ad-hoc network). *International Journal of Science, Engineering and Technology Research (IJSETR)*, 4(16) :1024, 2015. <https://doi.org/10.1109/ICACTE.2010.5579491>.
- [155] Aditi Roy, Bijan Paul, and Sanjit Kumar Paul. Vanet topology based routing protocols & performance of aodv, dsr routing protocols in random waypoint scenarios. In *Computer and Information Engineering (ICCIE), 2015 1st International Conference on*, pages 50–53. IEEE, 2015. <https://doi.org/10.1109/ccie.2015.7399315>.
- [156] Guoyou He. Destination-sequenced distance vector (dsv) protocol. *Networking Laboratory, Helsinki University of Technology*, pages 1–9, 2002. <https://doi.org/10.1109/icfn.2009.51>.
- [157] Patrick Sondi, Dhavy Gantsou, and Sylvain Lecomte. Design guidelines for quality of service support in optimized link state routing-based mobile ad hoc networks. *Ad Hoc Networks*, 11(1) :298–323, 2013. <https://doi.org/10.1016/j.adhoc.2012.06.001>.
- [158] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. Important : A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM 2003. Twenty-second annual joint conference of the IEEE computer and communications. IEEE societies*, volume 2, pages 825–835. IEEE, 2003. <https://doi.org/10.1109/infcom.2003.1208920>.
- [159] Feliz Kristianto Karnadi, Zhi Hai Mo, and Kun-chan Lan. Rapid generation of realistic mobility models for vanet. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 2506–2511. IEEE, 2007. <https://doi.org/10.1109/wcnc.2007.467>.
- [160] Mitul K Patel. Comparative study of vehicular ad-hoc network mobility models and simulators. *International Journal of Computer Applications*, 47(6) :38–43, 2012. <https://doi.org/10.5120/7196-9967>.
- [161] Jérôme Härri, Fethi Filali, Christian Bonnet, and Marco Fiore. Vanetmobisim : generating realistic mobility patterns for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97. ACM, 2006. <https://doi.org/10.1145/1161064.1161084>.
- [162] Abolhassan Halati, Henry Lieu, and Susan Walker. Corsim-corridor traffic simulation model. In *Traffic Congestion and Traffic Safety in the 21st Century : Challenges, Innovations, and Opportunities* Urban Transportation Division, ASCE ; Highway Division, ASCE ; Federal Highway Administration, USDOT ; and National Highway Traffic Safety Administration, USDOT., 1997.
- [163] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium*

- on Simulation and Modelling (MESM20002)*, pages 183–187, 2010. https://doi.org/10.1007/978-1-4419-6142-6_7.
- [164] Eiji Teramoto, Miyako Baba, Hiroko Mori, Yoshikazu Asano, and Hiroshi Morita. Netstream : traffic simulator for evaluating traffic information systems. In *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*, pages 484–489. IEEE, 1997. <https://doi.org/10.1109/itsc.1997.660522>.
- [165] Shie-Yuan Wang and Chih-Che Lin. Nctuns 5.0 : A network simulator for ieee 802.11 (p) and 1609 wireless vehicular network researches. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–2. IEEE, 2008. <https://doi.org/10.1109/vetecf.2008.464>.
- [166] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2) :1805, 2000. <https://doi.org/10.1103/physreve.62.1805>.
- [167] Stefan Krauß. *Microscopic modeling of traffic flow : Investigation of collision free vehicle dynamics*. PhD thesis, 1998. e-archive.informatik.uni-koeln.de/319/.
- [168] Dominique Dhoutaut, Anthony Régis, and François Spies. Impact of radio propagation models in vehicular ad hoc networks simulations. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 40–49. ACM, 2006. <https://doi.org/10.1145/1161064.1161072>.
- [169] Francisco J Martinez, Manuel Fogue, Manuel Coll, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Assessing the impact of a realistic radio propagation model on vanet scenarios using real maps. In *Network Computing and Applications (NCA), 2010 9th IEEE International Symposium on*, pages 132–139. IEEE, 2010. <https://doi.org/10.1109/nca.2010.24>.
- [170] Pranav Kumar Singh and Kapang Lego. Comparative study of radio propagation and mobility models in vehicular adhoc network. *International Journal of Computer Applications (0975-8887)*, 16(8), 2011. <https://doi.org/10.5120/2031-2600>.
- [171] Abdallah Rhattoy and Abdelkarim Zatni. The impact of propagation environment and traffic load on the performance of routing protocols in ad hoc networks. *arXiv preprint arXiv :1202.1677*, 2012. <https://doi.org/10.5121/ijdps.2012.3106>.
- [172] Carolina Tripp-Barba, Luis Urquiza-Aguiar, Aníbal Zaldívar-Colado, José Estrada-Jiménez, José Alfonso Aguilar-Calderón, and Mónica Aguilar Igartua. Comparison of propagation and packet error models in vehicular networks performance. *Vehicular Communications*, 12 :1–13, 2018. <https://doi.org/10.1016/j.vehcom.2018.02.001>.
- [173] Wantanee Viriyasitavat, Mate Boban, Hsin-Mu Tsai, and Athanasios Vasilakos. Vehicular communications : Survey and challenges of channel and propagation models. *IEEE Vehicular Technology Magazine*, 10(2) :55–66, 2015. <https://doi.org/10.1109/mvt.2015.2410341>.
- [174] Abduladhim Ashtaiwi, Ali Altayesh, and Khairai Belghet. Ieee 802.11 p performance evaluation at different driving environments. In *Computer Networks and Information Security (WSCNIS), 2015 World Symposium on*, pages 1–8. IEEE, 2015. <https://doi.org/10.1109/wscnis.2015.7368286>.
- [175] Khalid Abdel Hafeez, Lian Zhao, Zaiyi Liao, and Bobby Ngok-Wah Ma. The optimal radio propagation model in vanet. In *Systems and Networks Communications, 2009. ICSNC'09. Fourth International Conference on*, pages 6–11. IEEE, 2009. <https://doi.org/10.1109/icsnc.2009.101>.
- [176] Christoph Sommer and Falko Dressler. Using the right two-ray model? a measurement based evaluation of phy models in vanets. In *Proc. ACM MobiCom*, pages 1–3, 2011. <https://pdfs.semanticscholar.org/a86f/90f1238ccb90181c26335684fd762247408e.pdf>.
- [177] Huaqun Guo, Shen Tat Goh, Nicholas CS Foo, Qian Zhang, and Wai-Choong Wong. Performance evaluation of 802.11 p device for secure vehicular communication. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1170–1175. IEEE, 2011. <https://doi.org/10.1109/iwcmc.2011.5982706>.
- [178] Andreas F Molisch, Fredrik Tufvesson, Johan Karedal, and Christoph F Mecklenbrauker. A survey on vehicle-to-vehicle propagation channels. *IEEE Wireless Communications*, 16(6) :12–22, 2009. <https://doi.org/10.1109/mwc.2009.5361174>.
- [179] Thomas Mangel, Matthias Michl, Oliver Klemp, and Hannes Hartenstein. Real-world measurements of non-line-of-sight reception quality for 5.9 ghz ieee 802.11 p at intersections. In *International*

- Workshop on Communication Technologies for Vehicles*, pages 189–202. Springer, 2011. https://doi.org/10.1007/978-3-642-19786-4_17.
- [180] Rui Meireles, Mate Boban, Peter Steenkiste, Ozan Tonguz, and Joao Barros. Experimental study on the impact of vehicular obstructions in vanets. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 338–345. IEEE, 2010. <https://doi.org/10.1109/vnc.2010.5698233>.
- [181] Mate Boban, Tiago TV Vinhoza, Michel Ferreira, Joao Barros, and Ozan K Tonguz. Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE journal on selected areas in communications*, 29(1) :15–28, 2011. <https://doi.org/10.1109/JSAC.2011.110103>.
- [182] Johan Karedal, Nicolai Czink, Alexander Paier, Fredrik Tufvesson, and Andreas F Molisch. Path loss modeling for vehicle-to-vehicle communications. *IEEE transactions on vehicular technology*, 60(1) :323–328, 2011. <https://doi.org/10.1109/tvt.2010.2094632>.
- [183] Nabeel Akhtar, Sinem Coleri Ergen, and Oznur Ozkasap. Vehicle mobility and communication channel models for realistic and efficient highway vanet simulation. *IEEE Transactions on Vehicular Technology*, 64(1) :248–262, 2015. <https://doi.org/10.1109/TVT.2014.2319107>.
- [184] Christophe GRUET. 5g overview, 2018.
- [185] AM Abdelgader and Wu Lenan. The physical layer of the ieee 802.11 p wave communication standard : the specifications and challenges. In *Proceedings of the world congress on engineering and computer science*, volume 2, page 71, 2014.
- [186] ETSI Technical Committee Intelligent Transport System. Intelligent transport systems (its) ; access layer specification for intelligent transport systems operating in the 5 ghz frequency band. Technical Report ETSI EN 302 663 V1.2.0 (2012-11), 2012. http://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.00_20/en_302663v010200a.pdf.
- [187] Katrin Bilstrup, Elisabeth Uhlemann, and Erik G Ström. Medium access control in vehicular networks based on the upcoming ieee 802.11 p standard. In *15th World Congress on Intelligent Transport Systems (ITS), 15-20 November, 2008, Jacob K. Javits Convention Center, 36th St. & Eleventh Av. Manhattan, New York*, page 12. World Congress on ITS, 2008. <https://trid.trb.org/view.aspx?id=904224>.
- [188] Kashif Dar, Mohamed Bakhouya, Jaafar Gaber, Maxime Wack, and Pascal Lorenz. Wireless communication technologies for its applications [topics in automotive networking]. *IEEE Communications Magazine*, 48(5) :156–162, 2010. <https://doi.org/10.1109/mcom.2010.5458377>.
- [189] Roman Alieiev, Andreas Kwoczek, and Thorsten Hehn. Automotive requirements for future mobile networks. In *Microwaves for Intelligent Mobility (ICMIM), 2015 IEEE MTT-S International Conference on*, pages 1–4. IEEE, 2015. <https://doi.org/10.1109/icmim.2015.7117947>.
- [190] Theodore L Willke, Patcharinee Tientrakool, and Nicholas F Maxemchuk. A survey of inter-vehicle communication protocols and their applications. *IEEE Communications Surveys & Tutorials*, 11(2) :3–20, 2009. <https://doi.org/10.1109/SURV.2009.090202>.
- [191] Alexandre Petrescu, Thierry Ernst, Jing Huang, Rex Buddenberg, and Charles Perkins. Cooperative Adaptive Cruise Control and Platooning at SDOs and Gap Analysis, October 2015. <https://tools.ietf.org/html/draft-petrescu-its-cacc-sdo-04>.
- [192] Rex Chen, Wen-Long Jin, and Amelia Regan. Broadcasting safety information in vehicular networks : issues and approaches. *IEEE network*, 24(1), 2010. <https://doi.org/10.1109/mnet.2010.5395779>.
- [193] David B Johnson. Routing in ad hoc networks of mobile hosts. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 158–163. IEEE, 1994. <https://doi.org/10.1109/mcsa.1994.513476>.
- [194] Elliott Kaplan and Christopher Hegarty. *Understanding GPS : principles and applications*. Artech house, 2005. ISBN : 1580538940.
- [195] Pierre Merdrignac. *Système coopératif de perception et de communication pour la protection des usagers vulnérables*. PhD thesis, Paris, ENMP, 2015. <https://hal.inria.fr/tel-01242774/>.
- [196] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks : A new challenge for localization-based systems. *Computer communications*, 31(12) :2838–2849, 2008. <https://doi.org/10.1016/j.comcom.2007.12.004>.

- [197] Davide Margaria and Emanuela Falletti. A novel local integrity concept for gnss receivers in urban vehicular contexts. In *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, pages 413–425. IEEE, 2014. <https://doi.org/10.1109/plans.2014.6851399>.
- [198] Evangeline Pollard and Denis Gingras. Improved low cost gps localization by using communicative vehicles. In *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, pages 317–322. IEEE, 2012. <https://doi.org/10.1109/icarcv.2012.6485178>.
- [199] Anaïs Martineau, Christophe Macabiau, and Mikaël Mabillean. Gnss raim assumptions for vertically guided approaches. In *GNSS 2009, 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation*, pages pp–2791, 2009. <https://hal-enac.archives-ouvertes.fr/hal-01022160/>.
- [200] ETSI Technical Committee Intelligent Transport System. Intelligent Transport Systems (ITS); vehicular communications; basic set of applications; local dynamic map (ldm). Technical Report ETSI EN 302 895 V1.1.1 (2014-09), 2014. http://www.etsi.org/deliver/etsi_en/5C302800_302899/5C302895/5C01.01.01_60/5Cen_302895v010101p.pdf.
- [201] Hideki Shimada, Akihiro Yamaguchi, Hiroaki Takada, and Kenya Sato. Implementation and evaluation of local dynamic map in safety driving systems. *Journal of Transportation Technologies*, 5(02) :102, 2015. <https://doi.org/10.4236/jtts.2015.52010>.
- [202] Hendrik-jorn Gunther, Oliver Trauer, and Lars Wolf. The potential of collective perception in vehicular ad-hoc networks. In *ITS Telecommunications (ITST), 2015 14th International Conference on*, pages 1–5. IEEE, 2015. <https://doi.org/10.1109/itst.2015.7377190>.
- [203] Anh Tuan Giang. *Capacity of vehicular Ad-hoc NETWORK*. PhD thesis, Université Paris Sud-Paris XI, 2014. <https://tel.archives-ouvertes.fr/tel-00989836/>.
- [204] Hendrik-Jorn Günther, Raphael Riebl, Lars Wolf, and Christian Facchi. Collective perception and decentralized congestion control in vehicular ad-hoc networks. In *Vehicular Networking Conference (VNC), 2016 IEEE*, pages 1–8. IEEE, 2016. <https://doi.org/10.1109/vnc.2016.7835931>.
- [205] Nader Chaabouni, Abdelhakim Senhaji Hafid, Jihene Rezgui, and Soumaya Cherkaoui. Setting up an extended perception in a vehicular network environment : A proof of concept. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–7. IEEE, 2016. <https://doi.org/10.1117/12.2306580.5788820150001>.
- [206] Emna Chebbi, Patrick Sondi, Eric Ramat, Lucas Rivoirard, and Martine Wahl. Simulation of a clustering scheme for vehicular ad hoc networks using a devs-based virtual laboratory environment. *Procedia computer science*, 130 :344–351, 2018.

Annexe A

Algorithmes de la version 1 de CBL

Le contenu de cette annexe en langue anglaise est une reprise du texte des pages 73 à 76 de l'article [8] : Rivoirard L., *et al.*, "Chain-Branch-Leaf : a Clustering Scheme for Vehicular Networks Using Only V2V Communications", Journal, Ad Hoc Networks ,vol. 68, pp. 70-84, January 2018.

A.1 CBL scheme

Each reception of a HELLO message by a node N_i and coming from a neighbor node N_j triggers the following procedure (Fig. A.1). At the reception of an HELLO message from a neighbor node, Algorithm 1 runs the update of the 1-hop neighbor routing table. Then, there are two cases depending on whether the receiver node is a branch or a leaf. If the node is a branch, it executes Algorithm 5. Algorithm 5 turns the branch node into a leaf if the node did not receive for a long time any HELLO messages from any nodes that elected it. Otherwise, if the node is a leaf, it executes the algorithm 2. Algorithm 2 turns the leaf node into a branch if the sender node has elected it as its branch choice. Finally, there are two cases depending on whether the receiver node is now a branch or a leaf. If the node is a branch, it executes Algorithm 4 where the branch node is in charge of the election of the upstream and downstream branch nodes to give form to the chain. Otherwise, if the node is a leaf, it executes the algorithm 3. In Algorithm 3, the leaf node is in charge of the election of a node as its (relay) branch node. By processing algorithm 4, each branch node has its local view of the chain which consists of an upstream branch node and a downstream one. Furthermore, by processing the algorithm 3, a leaf node establishes only a connection to the chain by way of the branch node that it chose.

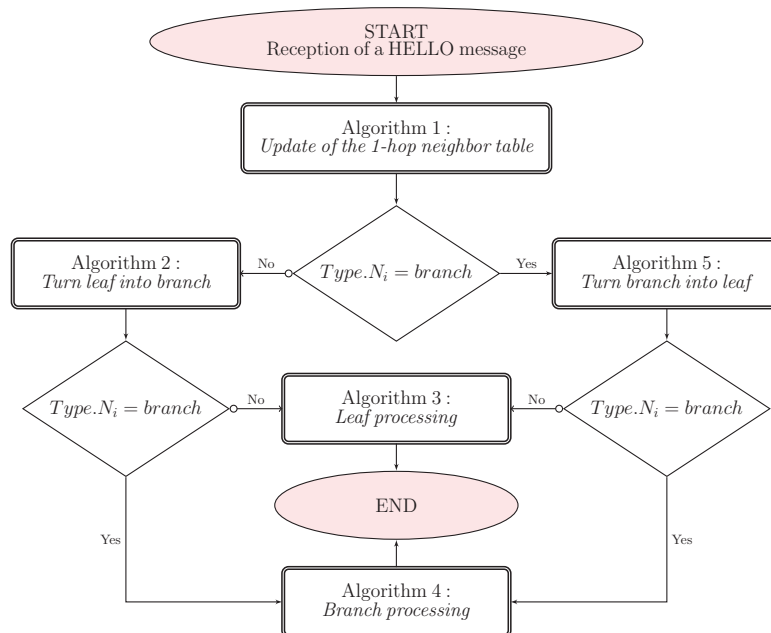


FIGURE A.1 – Algorithm applied when receiving a HELLO message

A.1.1 Algorithm 1. Update the 1-hop neighbor table

CBL scheme uses the same algorithm than the OLSR protocol to update the 1-hop neighbor table and link type. Moreover, it checks for each neighbor in the table if the duration elapsed since the last HELLO message received from it (recorded in timer T1) is higher than Vtime. In that case, the neighbor is removed from the table.

Algorithm 1 Update the 1-hop neighbor table

```

1: procedure UPDATE_NEIGHBOOR_TABLE( $N_i, N_j$ )
2:   if  $LinkType.N_i(N_j) == ASYM$  AND  $N_i \in NeighborTable.N_j$  then
3:      $T2.N_i(N_j) \leftarrow T$ 
4:   end if
5:   if  $N_i \in NeighborTable.N_j$  then
6:      $LinkType \leftarrow SYM$ 
7:   else
8:      $LinkType \leftarrow ASYM$ 
9:   end if
10:  if  $N_j \in NeighborTable.N_i$  then
11:    Update  $NeighborTable.N_i$ 
12:  else
13:    Add  $N_j$  in  $NeighborTable.N_i$ 
14:  end if
15:  if  $\exists N_k \in NeighborTable.N_i$  with  $T1.N_i(N_k) + Vtime > T$  then
16:    Clear  $N_k$  in  $NeighborTable.N_i$ 
17:  end if
18: end procedure

```

SYM : Symmetric, $ASYM$: Asymmetric

A.1.2 Algorithm 2. Turn leaf into branch

When a neighbor N_j has chosen N_i as a branch (the BranchChoice of N_j is set at the address of N_i), the node N_i turns its type into branch. Then, N_i updates its timer T3, initializes its BranchChoice to empty, and writes in its neighbor table that it is elected by N_j . If N_j was a branch, N_i will add automatically the node N_j as its elected upstream or downstream branch nodes to form a chain taking into account its relative position (up- or downstream).

Algorithm 2 Turn leaf into branch

```

1: procedure TURN_LEAF_INTO_BRANCH( $N_i, N_j$ )
2:   if  $BranchChoice.N_j == N_i$  then
3:      $BranchNode.N_i \leftarrow 0$ 
4:     if  $Type.N_j == 1$  then
5:       Add  $N_j$  in  $BranchNode.N_i$ 
6:     end if
7:      $Type.N_i \leftarrow 1$ 
8:      $T3.N_i \leftarrow T$ 
9:      $BranchChoice.N_i \leftarrow 0$ 
10:     $Elected.N_i(N_j) \leftarrow 1$ 
11:   end if
12: end procedure

```

A.1.3 Algorithm 3. Leaf processing

A leaf N_i selects a node as its branch, then it writes the branch address in its BranchChoice variable. When several branch candidates are detected, some optimizations are introduced in the choice process. In this work, N_i first looks for branch nodes driving in the same direction, then it chooses the closer according to the distance. Notice that two nodes drive in the same direction when :

$$|\sigma_{N_i} - \sigma_{N_j}| < \sigma_{max} \quad (\text{A.1})$$

When N_i detects no branch around after a time greater than Vtime with respect to the timer T2, the node sets in its BranchChoice the address of the leaf that is driving in the same direction. Nevertheless, if more than one leaf is a candidate, then the address of the candidate node having the lowest speed is put in its BranchChoice. When N_i has N_j address in its BranchChoice variable, if N_j is still a branch node, N_i updates its time counter T3 with the last reception timestamp of the HELLO message from N_j . However, if N_i received no HELLO message from N_j for a time longer than Vtime, it initializes its BranchChoice to empty in order to join a new cluster.

Algorithm 3 Leaf processing

```

1: procedure LEAF_PROCESSING( $N_i, N_j$ )
2:   if BranchChoice. $N_i$ ! = 0 then
3:     if T3. $N_i$  + Vtime >= T then
4:       if BranchChoice. $N_i$  ==  $N_j$  then
5:         T3. $N_i$   $\leftarrow$  T
6:       end if
7:     end if
8:     if T3. $N_i$  + Vtime < T OR (Type. $N_j$  == 0 AND BranchChoice. $N_i$  ==  $N_j$ ) then
9:       BranchChoice. $N_i$   $\leftarrow$  0
10:      T3. $N_i$   $\leftarrow$  0
11:    end if
12:  end if
13:  if Type. $N_j$  == 1 then
14:    if BranchChoice. $N_i$  == 0 then
15:      BranchChoice. $N_i$   $\leftarrow$   $N_j$ 
16:      T3. $N_i$   $\leftarrow$  T
17:    else
18:      if CompareBranchNode( $N_j$ , BranchChoice. $N_i$ ) == 1 then
19:        BranchChoice. $N_i$   $\leftarrow$   $N_j$ 
20:        T3. $N_i$   $\leftarrow$  T
21:      end if
22:    end if
23:  else ▷ If  $N_j$  is a leaf node
24:    if BranchChoice. $N_i$  == 0 then
25:      if CompareLeafNode( $N_j$ , NeighborTable. $N_i$ ) == 1 then
26:        BranchChoice. $N_i$   $\leftarrow$   $N_j$ 
27:        T3. $N_i$   $\leftarrow$  T
28:      end if
29:    end if
30:  end if
31: end procedure

```

```

1: function COMPARELEAFNODE( $N_i, [N_j, N_k, N_{k+1}, \dots, N_{k+n}]$ ) (returns true if the  $N_j$  node is the most
   appropriate branch node choice with respect to the node list  $[N_k, \dots, N_{k+n}]$  for the node  $N_i$ )
2:   if  $T2.N_i(N_j) + Vtime > T$  then
3:     return False ▷ The node  $N_j$  is not known long enough to choose it.
4:     break
5:   end if
6:   if  $|\sigma_{N_j} - \sigma_{N_i}| \geq \sigma_{max}$  then ▷ Different direction
7:     if  $\exists l \in [k, k+n], |\sigma_{N_l} - \sigma_{N_i}| < \sigma_{max}$  then
8:       return False
9:       break
10:    end if
11:  end if
12:  if  $\forall l \in [k, k+n], V_{N_j} < V_{N_l}$  then
13:    return True
14:    break
15:  else
16:    return False
17:    break
18:  end if
19: end function

```

```

1: function COMPAREBRANCHNODE( $N_i, N_j, N_k$ ) (returns true if the node  $N_j$  is the most appropriate
   branch node choice with respect to the current branch choice  $N_k$  for the  $N_i$  node)
2:   if  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  then ▷ Same direction
3:     if  $|\sigma_{N_k} - \sigma_{N_i}| \geq \sigma_{max}$  then ▷ Different direction
4:       return True
5:       break
6:     else
7:       if  $\|N_i - N_j\| < \|N_i - N_k\|$  then ▷ Euclidean distance
8:         return True
9:         break
10:      end if
11:    end if
12:  else
13:    if  $|\sigma_{N_k} - \sigma_{N_i}| \geq \sigma_{max}$  then ▷ Different direction
14:      if  $\|N_i - N_j\| < \|N_i - N_k\|$  then ▷ Euclidean distance
15:        return True
16:        break
17:      end if
18:    end if
19:  end if
20:  return False
21: end function

```

A.1.4 Algorithm 4. Branch processing

A branch node N_i participates in the creation of a chain. To this purpose, it elects an upstream branch and a downstream one from its current position, taking into account its trajectory direction. The election process selects the branch nodes that, firstly, currently exists and that, secondly, have not yet joined a chain or have chosen N_i as a branch node of their chain.

By keeping the same upstream and downstream branch nodes for N_i , while N_i location is maintained between them (no overtaking) and N_i is still in their transmission range (it receives their HELLO message), the algorithm favors the stability of the chain.

When N_i detects no branch node either upstream or downstream, it selects among its 1-hop neighbors the leaf node driving in the same direction, provided that this latter brings at least one more link, via its own 1-hop neighborhood, to a new node previously unknown from N_i . Next, if several leaf nodes are found, N_i selects, as BranchChoice, the address of the one that has the highest CT value. This election will change this leaf node into a branch node after the reception of the next HELLO message from node N_i . To select a new branch among leaf nodes, we do not consider a fixed distance threshold to avoid restricting the scheme to only few highway contexts. The advantage of using a communication network metric is to decrease the probability of choosing the closest node, and therefore the risk of frequently breaking the chain when vehicles overtake.

N_i updates the “Elected” variable for N_j in its 1-hop neighbor table : if N_j is connected to N_i , the variable is set at true and N_i updates its timer T3, otherwise the variable is set at false.

The branch node N_i checks for the two selected nodes (upstream and downstream) in its chain the duration elapsed since it received the last HELLO message from this latter. Every neighbor having a duration higher than Vtime is removed.

To manage an overtaking between two successive branches of a chain, when the sender N_j is already in the upstream elected node (respectively downstream) in the chain of N_i , this latter checks that the upstream position of N_j (respectively downstream), its type, and its direction have not evolved, and that N_i address appears in the downstream elected nodes (respectively upstream) in the chain of N_j . If these conditions are fulfilled, N_i updates the counter T3. But, if the relative position has changed, N_j position became downstream (respectively upstream), N_i removes the address of N_j in its upstream elected node parameter (respectively downstream), and set it in its downstream elected node parameter.

A.1.5 Algorithm 5. Turn branch into leaf

A branch N_i goes back to leaf type when it received no more HELLO messages from any of the nodes that elected it for a duration above Vtime. This process refers to the value T3.

Algorithm 5 Algorithm 5. Turn branch into leaf

```

1: procedure TURN_LEAF_INTO_BRANCH( $N_i, N_j$ )
2:   if  $T3.N_i + Vtime < T$  then
3:      $Type.N_i \leftarrow 0$ 
4:      $T3.N_i \leftarrow 0$ 
5:      $BranchChoice.N_i \leftarrow 0$ 
6:      $BranchNode.N_i \leftarrow 0$ 
7:   end if
8: end procedure

```

Algorithm 4 Branch processing

```

1: procedure BRANCH_PROCESSING( $N_i, N_j$ )
2:   if  $\exists N_k \in \text{BranchNode}.N_i$  with  $T1.N_i(N_k) + V\text{time} < T$  then
3:      $\text{BranchNode}.N_i(N_k) \leftarrow 0$ 
4:   end if
5:   if  $N_j \in \text{BranchNode}.N_i$  then
6:     if  $\text{BranchChoice}.N_i == N_j$  AND  $\text{Type}.N_j == 1$  then
7:        $\text{BranchChoice}.N_i \leftarrow 0$ 
8:     end if
9:     if  $\text{ChangeChainNode}(N_i, N_j) == 0$  then
10:       $T3.N_i \leftarrow T$ 
11:    else
12:       $[\text{ChoiceU}, \text{ChoiceD}] \leftarrow \text{ChangeChainNode}(N_i, N_j)$ 
13:       $\text{BranchNode}.N_i(U) \leftarrow \text{ChoiceU}$ 
14:       $\text{BranchNode}.N_i(D) \leftarrow \text{ChoiceD}$ 
15:       $T3.N_i \leftarrow T$ 
16:    end if
17:  end if
18:  if  $\text{Elected}.N_i(N_j) == 1$  then
19:    if  $\text{Type}.N_j == 1$  then
20:      if  $N_j \in \text{BranchNode}.N_i$  then
21:         $T3.N_i \leftarrow T$ 
22:      else
23:         $\text{Elected}.N_i(N_j) \leftarrow 0$ 
24:      end if
25:    else
26:      if  $\text{BranchChoice}.N_i == N_j$  then
27:         $T3.N_i \leftarrow T$ 
28:      else
29:         $\text{Elected}.N_i(N_j) \leftarrow 0$ 
30:      end if
31:    end if
32:  end if
33:  if  $\text{Type}.N_j == 1$  then
34:    if  $\text{BranchNode}.N_i(U/D) == 0$  AND  $(\text{BranchNode}.N_j(U/D) == 0$  OR
35:  $\text{BranchNode}.N_j(U/D) == N_i)$  AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  then
36:       $\text{BranchNode}.N_i(U/D) \leftarrow N_j$ 
37:      if  $\text{BranchNode}.N_j(U/D) == N_i$  then
38:         $\text{Elected}.N_i(N_j) \leftarrow 1$ 
39:      end if
40:    end if
41:  else
42:    if  $\text{BranchChoice}.N_j = N_i$  then
43:       $\text{Elected}.N_i(N_j) \leftarrow 1$ 
44:       $T3.N_i \leftarrow T$ 
45:    if  $\text{BranchNode}.N_i(U/D) == 0$  then
46:      if  $\text{CompareChainNode}(N_j, \text{NeighborTable}.N_i) == 1$  then
47:         $\text{BranchNode}(U/D).N_i \leftarrow N_j$ 
48:         $\text{BranchChoice}.N_i \leftarrow N_j$ 
49:         $T3.N_i \leftarrow T$ 
50:      end if
51:    end if
52:  else
53:     $\text{Elected}.N_i(N_j) \leftarrow 0$ 
54:  end if
55: end procedure

```

$U = \text{Upstream}, D = \text{Downstream}$

```

1: function [CHOICEU, CHOICED]= CHANGECHAINNODE( $N_i, N_j$ )(Returns true if the node  $N_j$  that is
   a branch node selected by  $N_i$  is to be deleted or replaced by another node in the TableCahin of  $N_i$ )
2:   ChoiceD  $\leftarrow$  0 ChoiceU  $\leftarrow$  0 ▷ Initialization
3:   if  $Type.N_j == 1$  then
4:     if ChangePosition_U/D( $N_i, N_j$ ) $==1$  then ▷ Return true if  $N_i$  or  $N_j$  overtakes
5:       if Position_U/D( $N_i, N_j$ ) $==1$  then ▷ Return true if  $N_i$  is upstream to  $N_j$ 
6:         return True
7:         ChoiceD  $\leftarrow$   $N_j$ 
8:         if  $BrNode.N_j(U) \in NTable.N_i$  AND  $BrNode.N_j(U) \neq N_i$  then
9:           ChoiceU  $\leftarrow$   $BranchNode.N_j(U)$ 
10:        else
11:          ChoiceU  $\leftarrow$  0
12:        end if
13:      else
14:        return True
15:        ChoiceU  $\leftarrow$   $N_j$ 
16:        if  $BrNode.N_j(D) \in NTable.N_i$  AND  $BrNode.N_j(D) \neq N_i$  then
17:          ChoiceD  $\leftarrow$   $BranchNode.N_j(D)$ 
18:        else
19:          ChoiceD  $\leftarrow$  0
20:        end if
21:      end if
22:    end if
23:  end if
24:  if  $Type.N_j == 1$  AND  $|\sigma_{N_j} - \sigma_{N_i}| < \sigma_{max}$  AND  $N_i \in BranchNode.N_j$  then
25:    return False
26:  else
27:    return True
28:  end if
29: end function

```

U=Upstream, D=Downstream

```

1: function COMPARECHAINNODE( $N_i, [N_j, N_k, N_{k+1}, \dots, N_{k+n}]$ )(returns true if the node  $N_j$  is the most
   appropriate as a branch node choice for the node  $N_i$  compared to the neighboring nodes ( $N_k, N_{k+1},$ 
    $\dots, N_{k+n}$ ))
2:   if  $Size(NeighborTable.N_j) \leq Size(NeighborTable.N_j \cap NeighborTable.N_i)$  then
3:     return False
4:     break
5:   else
6:     if  $|\sigma_{N_i} - \sigma_{N_j}| \geq \sigma_{max}$  then ▷ Different direction
7:       return False
8:       break
9:     else
10:       $LinkDurationRef \leftarrow$   $CalculLinkDuration(N_i, N_j)$ 
11:      if  $\forall l \in [k, k+n], CalculLinkDuration(N_i, N_l) < LinkDurationRef$  then
12:        return True // break
13:      else
14:        return False // break
15:      end if
16:    end if
17:  end if
18: end function

```

Annexe B

Proposition d'un codage de la variable "position" dans les messages HELLO de CBL

Le codage de l'information de position de longitude et de latitude sur 14 bits est le suivant :
Soit la position d'un nœud N_i : (par exemple Longitude : 51° 42' 38.9 et Latitude 44° 50' 42.9)

$$\begin{aligned} \text{Longitude} &: Y_1 Y_2^\circ \quad Y_3 Y_4' \quad Y_5 \cdot Y_6 \\ \text{Latitude} &: X_1 X_2^\circ \quad X_3 X_4' \quad X_5 \cdot X_6 \end{aligned}$$

$$\text{Avec} \begin{cases} \{Y_1, X_1\} \in \llbracket 0; 360 \rrbracket \\ \{Y_2, Y_4, Y_6, X_2, X_4, X_6\} \in \llbracket 0; 9 \rrbracket \\ \{Y_5, X_5\} \in \llbracket 0; 60 \rrbracket \\ \{Y_3, X_3\} \in \llbracket 0; 6 \rrbracket \end{cases}$$

On souhaite uniquement transmettre, au sein des messages HELLO, les trois derniers chiffres significatifs des informations de longitude et de latitude respectivement $Y_4'Y_5.Y_6$ et $X_4'X_5.X_6$.

Or $\{Y_4, X_4\} \in \llbracket 0; 9 \rrbracket$ et $\{Y_5, X_5\} \in \llbracket 0; 60 \rrbracket$. On code X_4 et X_5 avec Z tel que $Z = X_4 * 60 + X_5$, ainsi on a $X_4 = Z/60$ et $X_5 = Z\%60$. On a alors $Z \in \llbracket 0; 540 \rrbracket$. De même pour Y_4 et Y_5 avec $W = Y_4 * 60 + Y_5$. On a $W \in \llbracket 0; 540 \rrbracket$. Ainsi (Seconde Longitude) et (Seconde Latitude) sont codés sur 10 bits correspondant respectivement à W et Z . (dsec Long) et (dsec Lat) sont codés sur 4 bits correspondant respectivement à X_6 et Y_6 .

La procédure de reconstruction de la position d'un nœud voisin est la suivante :

Ce nœud, reçoit une information d'un de ces nœuds voisins N_j sous forme (Seconde Longitude), (Seconde Latitude), (dsec Long) et (dsec Lat) respectivement pour A, B, C_6 et D_6 . Il reforme, grâce aux formules $C_4 = A/60$ et $C_5 = A\%60$ et $D_4 = B/60$ et $D_5 = B\%60$, une partie de la position du nœud N_j :

$$\begin{aligned} \text{Longitude} &: \text{---}^\circ \quad \text{---} C_4' \quad C_5 \cdot C_6 \\ \text{Latitude} &: \text{---}^\circ \quad \text{---} D_4' \quad D_5 \cdot D_6 \end{aligned}$$

Pour reformer l'entièreté de la longitude du nœud N_j :

- Si le nœud N_i reçoit une position tel que $|Y_4 - C_4| \leq 5$ alors il recalcule la longitude du nœud N_j par :

$$\text{Longitude : } Y_1 Y_2^\circ \quad Y_3 C_4' \quad C_5 . C_6$$

- Si un nœud N_i reçoit une position tel que $|Y_4 - C_4| > 5$ il ajoute ou retranche une dizaine de minutes d'angle pour calculer la longitude du nœud :

- Si $Y_4 > 5$ et $C_4 < 5$, la longitude du nœud N_j est :

- Si $Y_3 < 5$:

$$\text{Longitude : } Y_1 Y_2^\circ \quad (Y_3 + 1) C_4' \quad C_5 . C_6$$

- si $Y_3 = 5$ et $Y_2 < 9$:

$$\text{Longitude : } Y_1 (Y_2 + 1)^\circ \quad 0 C_4' \quad C_5 . C_6$$

- si $Y_3 = 5$ et $Y_2 = 9$ et $Y_1 < 359$:

$$\text{Longitude : } (Y_1 + 1) 9^\circ \quad 0 C_4' \quad C_5 . C_6$$

- si $Y_3 = 5$ et $Y_2 = 9$ et $Y_1 = 359$:

$$\text{Longitude : } 0^\circ \quad 0 C_4' \quad C_5 . C_6$$

La même mécanique s'applique pour reformer l'entièreté de la latitude du nœud N_j .

Annexe C

Petit guide d'utilisation d'OPNET

OPNET Riverbed Modeler est un simulateur à événement discret dont l'approche de modélisation est orientée objet. Le logiciel fournit de nombreux modèles et outils pour la modélisation, la simulation et l'analyse des performances des réseaux. Nous présentons ici uniquement les modules utilisés dans le cadre de nos travaux de thèse.

C.1 Fonctionnement d'OPNET Riverbed Modeler

Trois domaines de travail (figure C.1), donnent accès à différents niveaux de modélisation d'un réseau de communication. Le domaine du réseau (*Network model*) est le plan de description de la topologie du réseau. Le domaine des nœuds (*Node model*) permet d'intervenir sur le diagramme des processus (MAC, IP, TCP, UDP ...). Le diagramme de processus respecte le modèle de référence en couche OSI de l'ISO. Il permet de modéliser différentes couches protocolaires d'un réseau de communication en fonction des objectifs que l'on se donne. Le domaine du processus (*Process model*) contient le diagramme d'état du processus. Chaque état du processus est décrit en un code C/C++ qui est supporté par une bibliothèque de fonctions préprogrammées (proto-C).

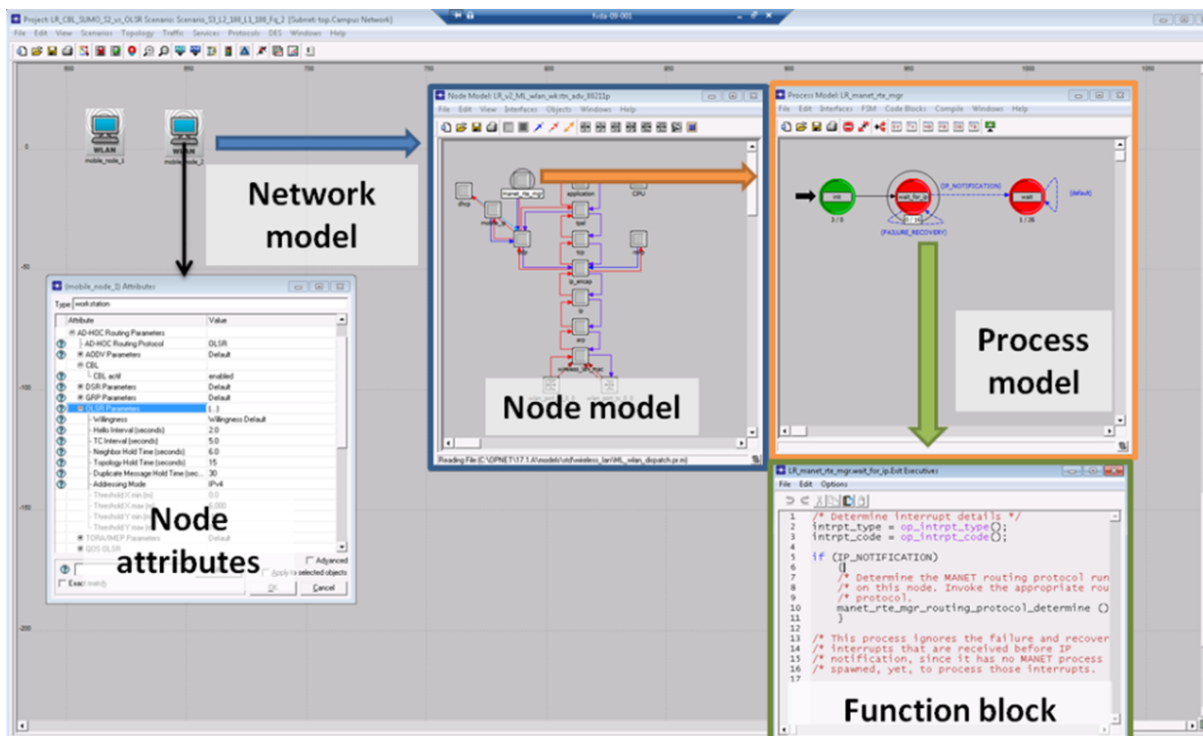


FIGURE C.1 – Fonctionnement du logiciel OPNET Riverbed Modeler

C.2 Lancement d'une simulation

La procédure à suivre pour lancer une simulation est la suivante. Tout d'abord il est nécessaire de créer un modèle du réseau avec différents nœuds communicants. Un assistant de déploiement de réseau est disponible permettant de définir facilement l'espace de déploiement, le nombre de nœuds du réseau, la technologie utilisée ou encore le protocole de routage choisi. Les nœuds du réseau peuvent par la suite être paramétrés différemment (figure C.2).

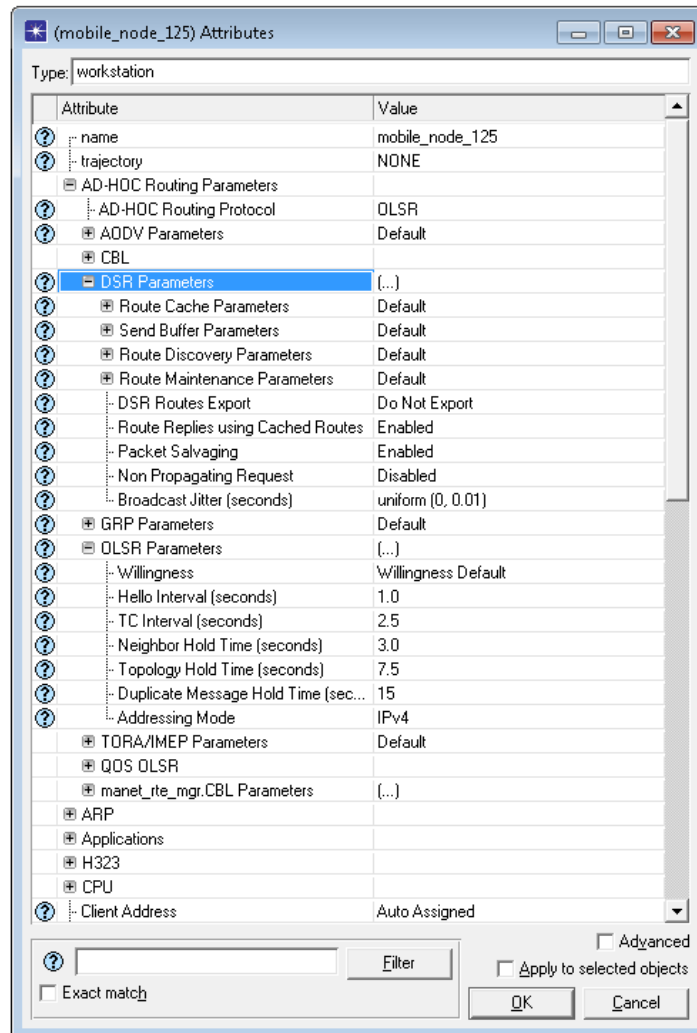


FIGURE C.2 – Paramétrage des nœuds communicants sous OPNET

Une fois le réseau déployé, il est nécessaire de déployer une application. Le logiciel permet de déployer des applications prédéfinies ou alors d'en personnaliser d'autres. Le profil de l'application détermine alors la manière dont l'application est exécutée et sa durée dans le cadre de la simulation (figure C.3).

Lorsque les applications sont configurées, il est nécessaire de choisir les statistiques que l'on souhaite collecter pour analyser les performances. Ces statistiques sont soit globales au réseau soit locales à chaque nœud. Les statistiques fournies par OPNET concernent aussi bien les applications choisies que les protocoles impliqués à tous les niveaux du modèle OSI. Le choix des statistiques est important car seuls les résultats relatifs aux statistiques sélectionnées seront disponibles à la fin de la simulation (figure C.4).

Au terme de ce processus, il est possible d'exécuter une simulation pour obtenir les performances dans les conditions définies à travers toutes les étapes précédentes (figure C.5).

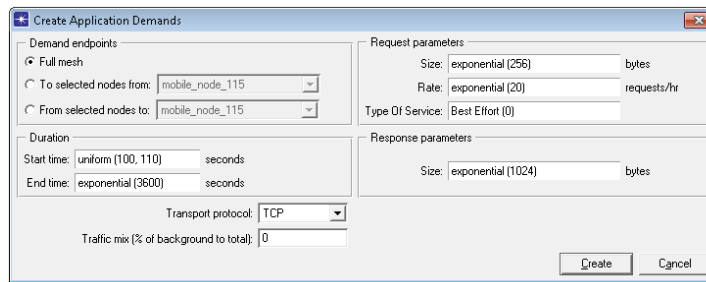


FIGURE C.3 – Déploiement d’une application sous OPNET

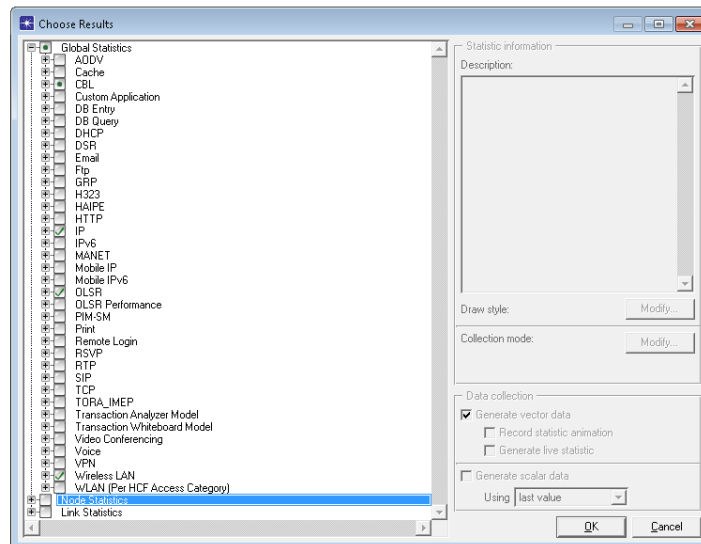


FIGURE C.4 – Sélection des statistiques sous OPNET

C.3 Utilisation de l’affectation automatique de trajectoire

L’option d’affectation automatique d’une trajectoire à chaque nœud du réseau ajoutée au modèle du nœud 802.11p nécessite l’utilisation de fichiers externes de format TRJ. Ces fichiers contiennent la trajectoire de chaque nœud du réseau. Leur en-tête définit les différentes unités utilisées, le nombre de fois que la trajectoire doit être jouée par le nœud, si elle doit être parcourue en aller-retour, le temps d’attente avant d’exécuter la trajectoire en aller-retour et le nombre de coordonnées composant la trajectoire. Le fichier C.1 montre un exemple de fichier TRJ.

C.4 Implantation de CBL au sein du protocole OLSR

Le logiciel OPNET Riverbed Modeler propose des modèles pour divers types de nœuds communicants. Chaque modèle peut être paramétré différemment pour simuler plusieurs situations (par exemple pour faire varier la fréquence d’envoi des messages HELLO).

Il est également possible d’accéder au code de chaque modèle et d’en modifier le fonctionnement à des fins de recherche, le modèle obtenu peut être sauvegardé sous un nom différent.

Le processus “*Routing Manager*” (`manet_rte_mgr`) est responsable de l’initialisation du protocole de routage. Par défaut ce processus offre le choix entre les protocoles de routage DSR, AODV, OLSR, TORA et GRP (figure C.6).

Un protocole de routage peut donc très simplement être rajouté à cette liste en créant un nouveau processus sans entraîner de modification sur les autres modèles du nœud communicant.

L’implémentation du protocole OLSR proposé par OPNET est conforme au RFC 362 [31]. Il se compose d’un modèle de processus représenté par des états et des transitions présenté en figure C.7.

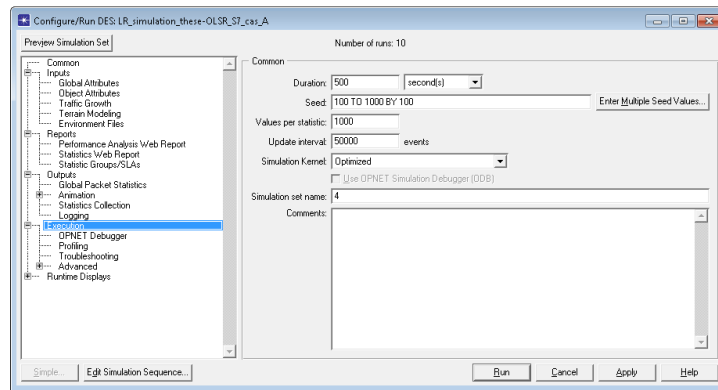


FIGURE C.5 – Lancement d’une simulation sous OPNET

Fichier C.1 – Exemple de fichier TRJ utilisable par OPNET

```

Version: 5
Position_Unit: Meters
Altitude_Unit: Meters
Coordinate_Method: fixed
Altitude_Method: absolute
locale: C
Calendar_Start: unused
Execute_Count: 1
Reverse_Trajectory: no
Reverse_Wait_Time: 0.000000
Coordinate_Count: 3
# X Position ,Y Position ,Altitude ,Traverse Time ,Wait Time ,Pitch ,Yaw ,Roll
0.000000 ,0.000000 ,0 ,0h0m0.1s ,0h0m0.0s ,Autocomputed ,Autocomputed ,Unspecified
0.330000 ,0.000000 ,0 ,0h0m0.1s ,0h0m0.0s ,Autocomputed ,Autocomputed ,Unspecified

```

Le premier état procède à l’initialisation du protocole et le deuxième état réagit aux événements (réception de messages HELLO et TC, ainsi qu’aux délais d’expiration associés au calcul de routes).

Afin de préserver la compatibilité avec les autres composants, CBL a été implanté dans une copie du modèle OLSR en conservant la même structure qu’OLSR et son invocation est réalisée par le biais du “*Routing Manager*”. Aucun mécanisme de cross-layer n’a été mis en œuvre avec les autres couches. Par conséquent, les performances obtenues sont également imputables aux seuls mécanismes mis en œuvre pour le routage.

Les modifications introduites dans le modèle du protocole OLSR natif, pour implanter CBL et créer CBL-OLSR, peuvent être résumées ainsi :

- La fonction exécutée à chaque réception des messages HELLO est modifiée pour exécuter l’algorithme CBL ;
- Un nouvel état (Initialisation) (figure C.8) et les fonctions associées sont ajoutés permettant de créer une zone d’attente éloignée du réseau à simuler lors de l’initialisation des noeuds ;
- De nouvelles variables globales incluant les seuils temporels de CBL sont ajoutées ;
- Une nouvelle transition et les fonctions associées sont ajoutées pour réinitialiser les noeuds à la fin de leur traversée de la section du réseau ;
- Le fichier d’en-tête contenant les définitions de la structures de données associées aux messages HELLO est modifié afin d’introduire les informations supplémentaires nécessaires au fonctionnement de CBL ;
- Les structures de données représentant les tables des voisins sont modifiées afin de conserver les informations supplémentaires nécessaires au fonctionnement de CBL
- L’heuristique de sélection des MPRs est remplacé par le choix de noeuds branche ;
- De nouvelles statiques et leur calculs associés sont ajoutés.

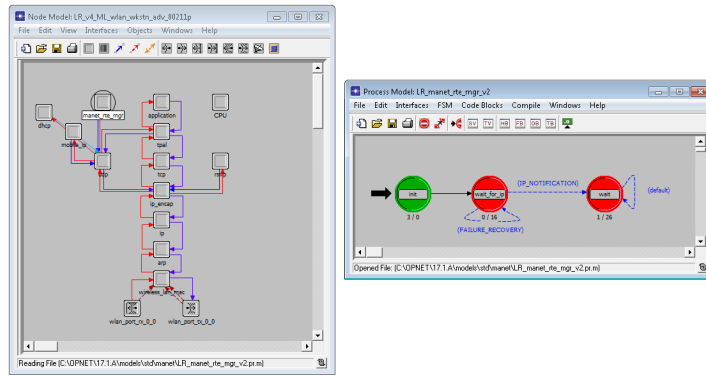


FIGURE C.6 – Routing Manager d’OPNET Riverbed Modeler

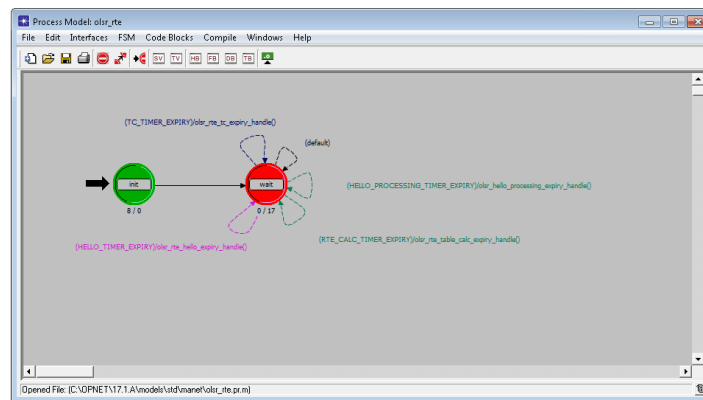


FIGURE C.7 – Modèle de processus de OLSR sous OPNET Riverbed Modeler

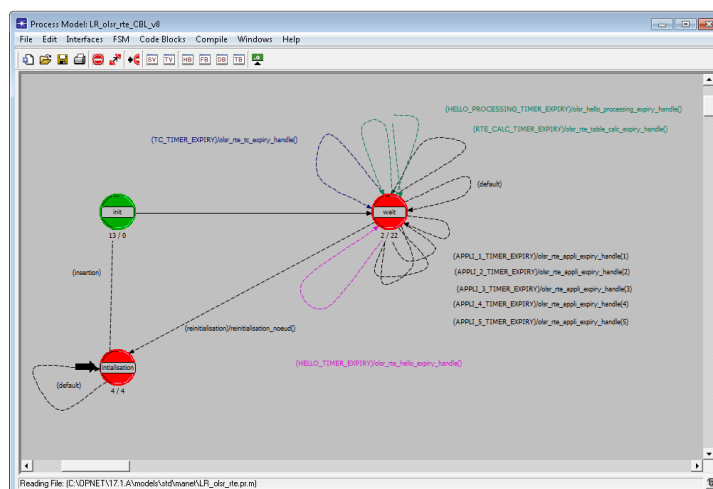


FIGURE C.8 – Modèle de processus de CBL sous OPNET Riverbed Modeler

Annexe D

Petit guide d'utilisation de SUMO

Nous avons souhaité détailler ici l'utilisation du logiciel SUMO.

D.1 Création du réseau

La première solution consiste à définir manuellement l'ensemble du réseau routier. Pour cela, il faut définir l'ensemble des points du réseau avec leurs coordonnées géographiques et leur identifiant unique comme l'illustre le fichier D.1 :

Fichier D.1 – Fichier de définition des points du réseau routier (.nod.xml)

```
<nodes>
<node id="1" x="0" y="0" />
<node id="2" x="5000" y="0" />
</nodes>
```

Il faut ensuite définir les liaisons entre ces nœuds du réseau routier. Pour cela, il est possible de spécifier le nombre de voies et la vitesse maximale autorisée sur chaque liaison comme l'illustre le fichier D.2. Il est également possible de définir une priorité au tronçon routier, une largeur ou encore une autorisation de circulation à certains véhicules spécifiquement.

Fichier D.2 – Fichier de définition des connections entre les points du réseau routier (.edg.xml)

```
<edges>
<edge from="1" id="1to2" to="2" numLanes="3" speed="36.1"/>
<edge from="2" id="2to1" to="1" numLanes="3" speed="36.1"/>
</edges>
```

D.2 Génération du réseau

A l'aide de ces deux fichiers, l'outil NETCONVERT permet de créer un fichier du réseau routier utilisable par SUMO. L'outil NETEDIT permet de visualiser le réseau ainsi créé (voir figure D.1). Il est possible d'importer automatiquement un réseau réel grâce à un export issu d'Open Street Map. Dans ce cas, l'utilisation du logiciel JOSM permet d'afficher et de nettoyer l'export OSM pour l'ouvrir directement avec NETEDIT.

D.3 Demande en véhicules

Il convient alors de définir la demande en véhicules grâce au fichier de routes. Ce fichier définit l'ensemble des types de véhicules grâce à leurs caractéristiques comme leur taille, leur accélération et décélération maximale ou encore leur vitesse maximale. Il définit également la demande en véhicules de chaque type sur les itinéraires constitués des liaisons entre chaque nœud du réseau routier.

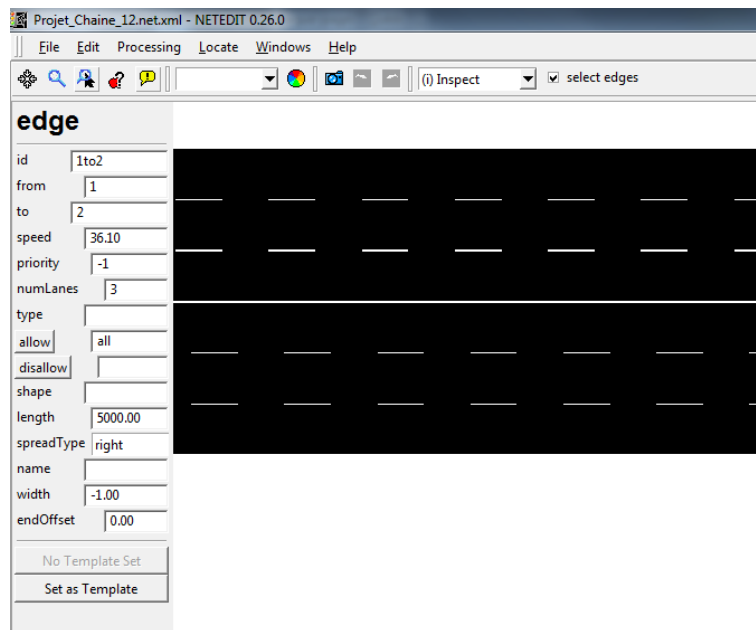


FIGURE D.1 – Visualisation du réseau créé par NETCONVERT

Fichier D.3 – Fichier de définition de la demande en véhicules (.rou.xml)

```

<routes>
  <vType accel="2.0" decel="3.0" id="Car" length="5.0" minGap="2.5" maxSpeed="42" sigma="0.5" speedFactor="1"
    speedDev="0.1" />
  <vType accel="1.0" decel="2.0" id="Truck" length="15.0" minGap="5" maxSpeed="36" sigma="0.5" speedFactor="
    0.84" speedDev="0.1" guiShape="truck" />

  <flow id="type1" begin="1" end="600" vehsPerHour="500" type="Car" departSpeed="max" departLane="random">
  <route edges="1to2"/>
  </flow>
  <flow id="type2" begin="1" end="600" vehsPerHour="500" type="Car" departSpeed="max" departLane="random">
  <route edges="2to1"/>
  </flow>
  <flow id="type3" begin="1" end="600" vehsPerHour="100" type="Truck" departSpeed="max" departLane="0">
  <route edges="1to2"/>
  </flow>
  <flow id="type4" begin="1" end="600" vehsPerHour="100" type="Truck" departSpeed="max" departLane="0">
  <route edges="2to1"/>
  </flow>
</routes>

```

D.4 Simulation

Le lancement de la simulation permet d'obtenir les traces de véhicules comme l'illustre le fichier D.4. L'outil de visualisation du SUMO est représenté sur la figure D.2.

Fichier D.4 – Début du fichier de traces issues de la simulation avec SUMO

```

<timestep time="0.00"/>
  <timestep time="1.00">
    <vehicle id="type1.0" x="5.10" y="-1.65" angle="90.00" type="Car" speed="36.10" pos="5.10" lane="1to2_2"
      slope="0.00"/>
    <vehicle id="type2.0" x="4994.90" y="4.95" angle="270.00" type="Car" speed="36.10" pos="5.10" lane="2to1_1"
      slope="0.00"/>
    <vehicle id="type3.0" x="15.10" y="-8.25" angle="90.00" type="Truck" speed="30.00" pos="15.10" lane="1
      to2_0" slope="0.00"/>
    <vehicle id="type4.0" x="4984.90" y="8.25" angle="270.00" type="Truck" speed="25.20" pos="15.10" lane="2
      to1_0" slope="0.00"/>
  </timestep>
  <timestep time="2.00">
    <vehicle id="type1.0" x="40.45" y="-1.65" angle="90.00" type="Car" speed="35.35" pos="40.45" lane="1to2_2"
      slope="0.00"/>
    <vehicle id="type1.1" x="5.10" y="-4.95" angle="90.00" type="Car" speed="33.87" pos="5.10" lane="1to2_1"
      slope="0.00"/>
    <vehicle id="type2.0" x="4959.05" y="4.95" angle="270.00" type="Car" speed="35.85" pos="40.95" lane="2
      to1_1" slope="0.00"/>
    <vehicle id="type2.1" x="4994.90" y="4.95" angle="270.00" type="Car" speed="35.23" pos="5.10" lane="2to1_1"
      slope="0.00"/>
    <vehicle id="type3.0" x="45.10" y="-8.25" angle="90.00" type="Truck" speed="30.00" pos="45.10" lane="1
      to2_0" slope="0.00"/>
    <vehicle id="type4.0" x="4959.97" y="8.25" angle="270.00" type="Truck" speed="24.93" pos="40.03" lane="2
      to1_0" slope="0.00"/>
  </timestep>

```

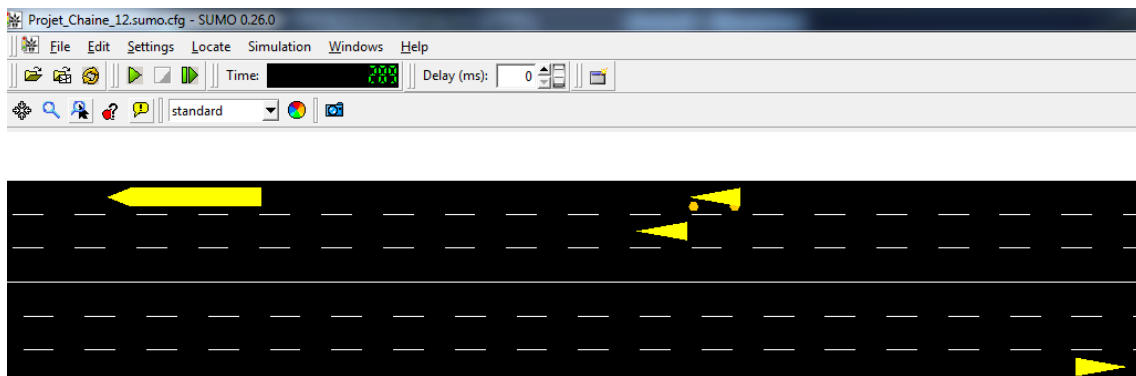


FIGURE D.2 – Visualisation de la simulation générée par SUMO

Annexe E

Résultats complémentaires à l'étude de sensibilité des coefficients pondérateurs

Les figures E.1 à E.4 présentent les résultats de l'étude de sensibilité des coefficients pondérateurs aux métriques Ms2, Ms5, Ms6 et Ms7.

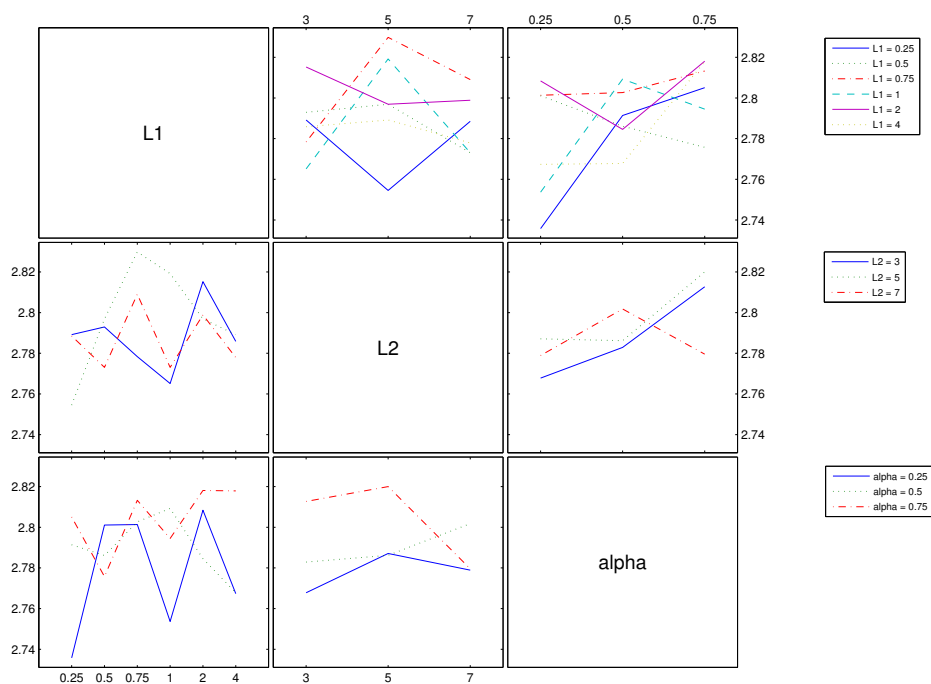


FIGURE E.1 – Impact observé via Ms2 entre t=150-300 s des coefficients pondérateurs L1, L2 et alpha sur la structure CBL – Scénario S5 cas A

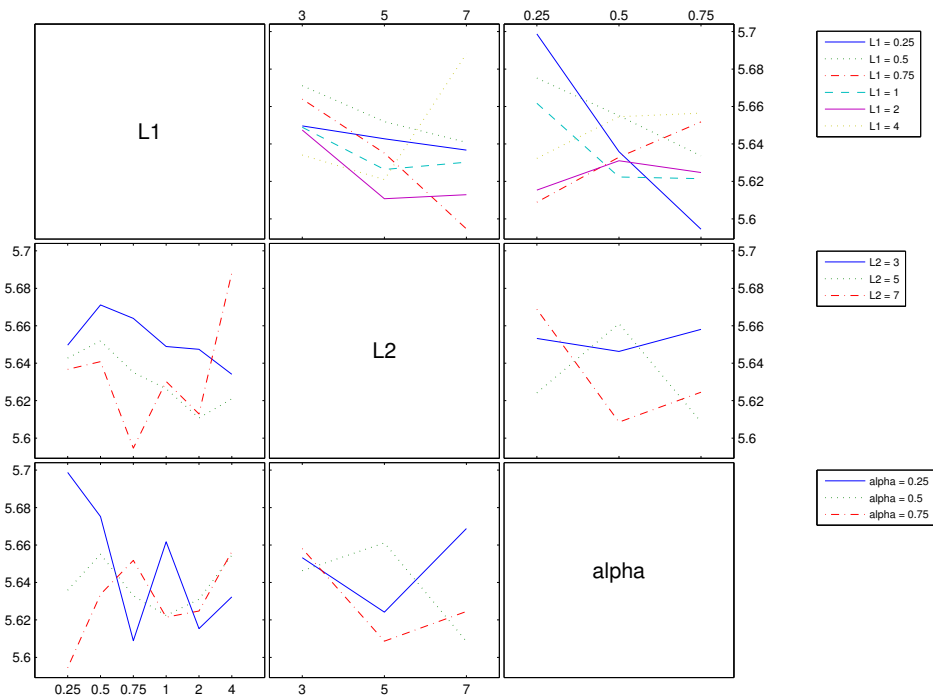


FIGURE E.2 – Impact observé via Ms5 entre t=150-300 s des coefficients pondérateurs L1, L2 et alpha sur la structure CBL – Scénario S5 cas A

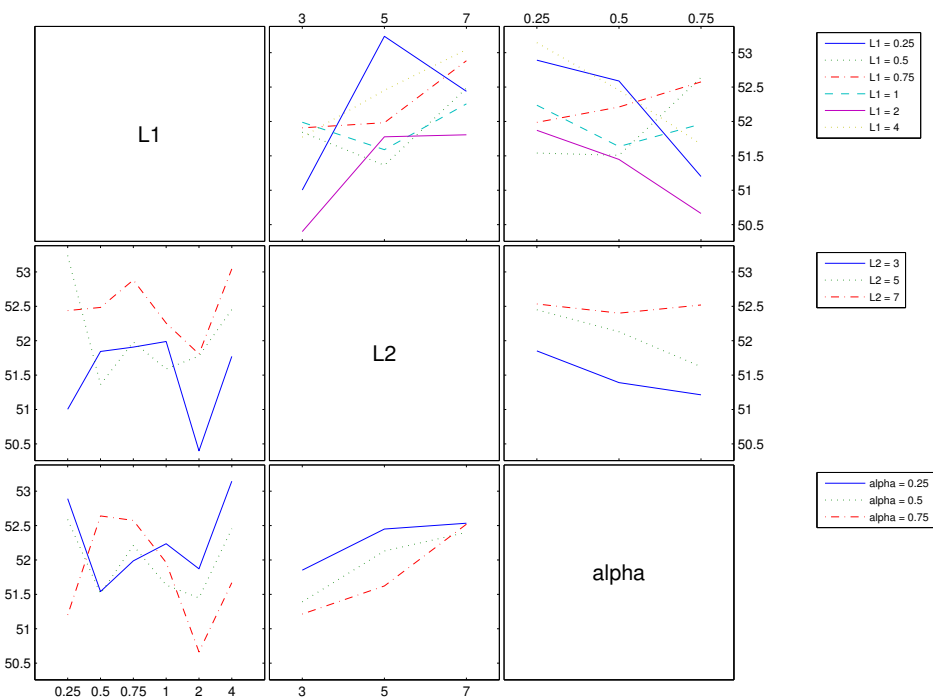


FIGURE E.3 – Impact observé via Ms6 entre t=150-300 s des coefficients pondérateurs L1, L2 et alpha sur la structure CBL – Scénario S5 cas A

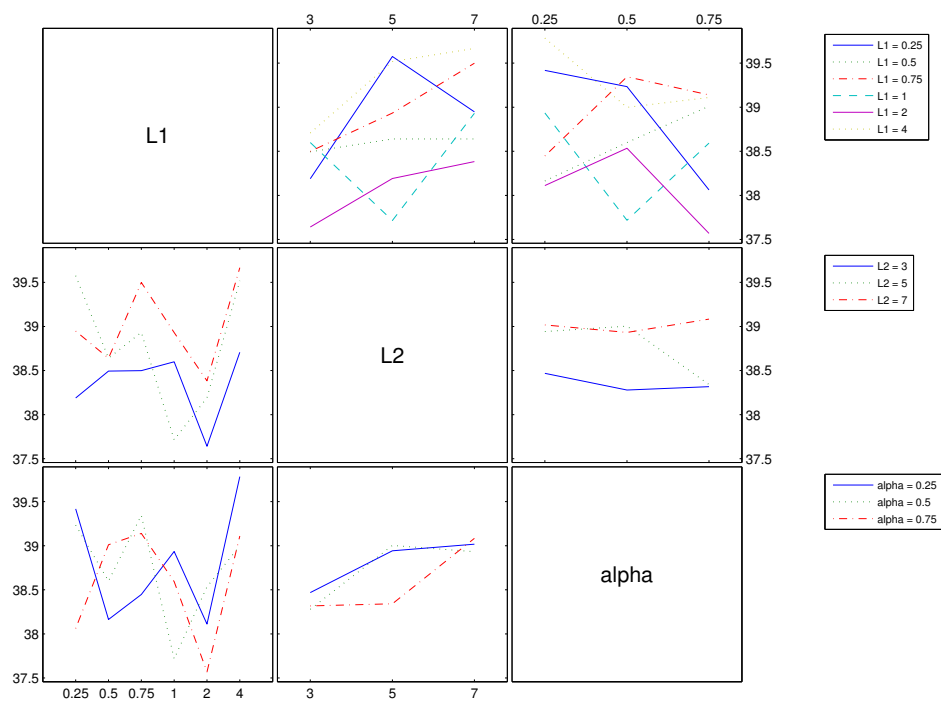


FIGURE E.4 – Impact observé via Ms7 entre $t=150-300$ s des coefficients pondérateurs L1, L2 et alpha sur la structure CBL – Scénario S5 cas A

Annexe F

Résultats de l'étude de sensibilité de la mobilité

Les tableaux F.1 et F.2 présentent les valeurs moyennes des résultats du trafic de routage respectivement pour les protocoles CBL-OLSR et OLSR.

TABLEAU F.1 – Valeurs moyennes des métriques de performances Mr1 à Mr7 calculées dans l'intervalle de temps 150 à 500 s – Scénarios S10 à S13, CBL-OLSR

Réseaux		R4		R5		
Scénarios		10	11	12	13	
Densité		+	+	++	-	
Nombre de nœuds		128	116	137	67	
Métriques	Mr1 : Charge_routage (kbit/s)	cas A	235	219	249	85,3
		cas B	115	106	126	43,6
	Mr2 : HELLO_total (kbit/s)	cas A	149	134	174	52,1
		cas B	78,6	70,1	92,3	27,8
	Mr3 : TC_total (kbit/s)	cas A	86,5	85,3	74,9	33,2
		cas B	36,3	36,1	34,2	15,9
	Mr4 : HELLO/Charge_routage (%)	cas A	64,7	62,6	70,7	61,9
		cas B	69,7	67,4	73,7	64,4
	Mr5 : TC_total/Charge_routage (%)	cas A	35,3	37,4	29,3	38,1
		cas B	30,3	32,6	26,3	35,6
	Mr6 : TC_relayé/TC_envoyé (%)	cas A	91,6	92,6	91,6	88,1
		cas B	89,3	90,6	90,2	86,8
	Mr7 : Délai WLAN (ms)	cas A	0,135	0,132	0,144	0,124
		cas B	0,137	0,135	0,147	0,125

Définition des métriques de routage en section 4.2.2.1.

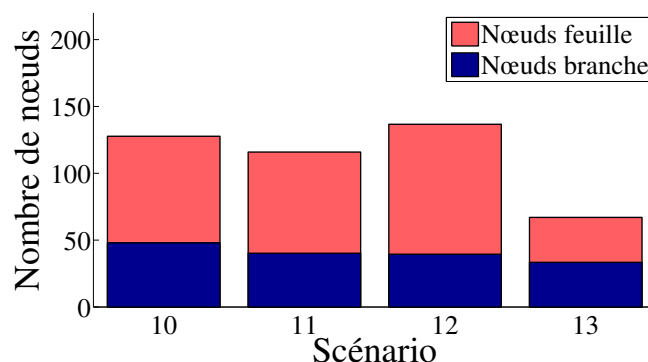


FIGURE F.1 – Nombre et type des nœuds – Scénarios S10 à S13, cas A, CBL-OLSR

La charge de routage (métrique Mr_1) augmente lorsque le nombre de nœuds du réseau augmente. Elle varie pour CBL-OLSR et le cas A entre 85,3 kbit/s pour le scénario S13 et 249 kbit/s pour le scénario S12. Comparativement, pour OLSR elle varie entre 320 kbit/s et 1520 kbit/s. La charge du trafic de routage entre les scénarios S11 et S12 qui compte uniquement 21 nœuds supplémentaires est plus que doublée pour OLSR alors qu'elle n'augmente que de 14% pour CBL-OLSR. Le passage à l'échelle de CBL-OLSR grâce au *clustering* est encore une fois démontré.

Le trafic des messages HELLO est lié également au nombre de nœuds du réseau comme nous l'avons vu section 4.2.2.4. Il est légèrement supérieur pour CBL-OLSR à cause des informations supplémentaires incluses dans les champs d'en-tête.

Le trafic des messages TC varie de 33,2 kbit/s à 56,5 kbit/s pour le cas A de CBL-OLSR tandis qu'il varie de 270 kbit/s à 1350 kbit/s pour OLSR. La charge des messages TC est très supérieure pour OLSR dont le pourcentage d'occupation dépasse 78,5% pour tous les scénarios tandis qu'elle se limite au maximum à 38,1% pour CBL-OLSR.

Les délais WLAN (métrique Ms7) sont stables pour tous les scénarios avec en moyenne 0,13 ms que ce soit pour les protocoles CBL-OLSR ou OLSR.

TABLEAU F.2 – Valeurs moyennes des métriques de performances Mr1 à Mr7 calculées dans l'intervalle de temps de 150 à 500 s – Scénarios S10 à S13, OLSR, comparaison avec CBL-OLSR

Réseaux		R4		R5		
Scénarios		10	11	12	13	
Densité		+	+	++	-	
Nombre de nœuds		128	116	137	67	
Métriques	Mr1 : Charge_routage (kbit/s)	cas A	838	637	1520	320
		<i>Diff</i>	603	+418	+1271	+234,7
		cas B	706	594	864	174
		<i>Diff</i>	591	+488	+738	+130,4
	Mr2 : HELLO_total (kbit/s)	cas A	145	131	170	50,2
		<i>Diff</i>	-4	-3	-4	-1,9
		cas B	77,7	69,5	91,1	27
		<i>Diff</i>	-0,9	-0,6	-1,2	-0,8
	Mr3 : TC_total (kbit/s)	cas A	693	507	1350	270
		<i>Diff</i>	606,5	+421,7	+1275,1	+236,8
		cas B	629	525	773	147
		<i>Diff</i>	592,7	+488,9	+738,8	+131,1
	Mr4 : HELLO/Charge_routage (%)	cas A	18,1	21,5	13,4	17,8
		<i>Diff</i>	-46,6	-41,1	-57,3	-44,1
		cas B	11,8	13	12	18,1
		<i>Diff</i>	-57,9	-54,4	-61,7	-46,3
	Mr5 : TC_total/Charge_routage (%)	cas A	81,9	78,5	86,6	82,2
		<i>Diff</i>	46,6	+41,1	+57,3	+44,1
		cas B	88,2	87	88	81,9
		<i>Diff</i>	57,9	+54,4	+61,7	+46,3
Mr6 : TC_relayé/TC_envoyé (%)	cas A	97,9	97,5	98,6	97	
	<i>Diff</i>	6,3	+4,9	+7	+8,9	
	cas B	98,6	98,5	98,7	96,9	
	<i>Diff</i>	9,3	+7,9	+8,5	+10,1	
Mr7 : Délai WLAN (ms)	cas A	0,144	0,136	0,17	0,122	
	<i>Diff</i>	0,009	+0,004	+0,026	-0,002	
	cas B	0,137	0,133	0,142	0,118	
	<i>Diff</i>	0	-0,002	-0,005	-0,007	

Où *Diff* est la différence entre les valeurs de OLSR moins celles de CBL-OLSR.

Définition des métriques de routage en section 4.2.2.1

Annexe G

Tables des valeurs pour les scénarios Sa1, Sa2 et Sa3

TABLEAU G.1 – Valeurs des métriques de performances pour le scénario Sa1 avec 10 noeuds sources

Métriques	Situation	Type	AODV	DSR	GRP	OLSR	QOLSR	CBL
Ma1 : Charge (bit/s)	Toute la simulation	1er quantile	0	0	2.6e+04	2.2e+05	4.2e+05	8.7e+04
		Moyenne	1.9e+06	6.8e+06	2.2e+06	1.2e+06	3.3e+06	1.6e+06
		3er quantile	4.6e+06	1.1e+07	5.5e+06	2.3e+06	9.1e+06	3.8e+06
		Maximum	1.7e+07	2.4e+07	1.6e+07	6.7e+06	1.4e+07	1.2e+07
	Au cours de l'échange	1er quantile	4.8e+06	1.1e+07	5.6e+06	2.6e+06	9.3e+06	4.2e+06
		Moyenne	5.7e+06	1.3e+07	6.2e+06	3.3e+06	1e+07	5.3e+06
		3er quantile	6.4e+06	1.5e+07	6.7e+06	3.9e+06	1.1e+07	6.1e+06
		Maximum	1.7e+07	2.4e+07	1.1e+07	6.7e+06	1.4e+07	1.2e+07
Ma2 : Débit (bit/s)	Toute la simulation	1er quantile	0	0	3.1e+05	2.7e+06	5.4e+06	1.1e+06
		Moyenne	3.1e+06	2.6e+06	2.6e+06	3.7e+06	8e+06	2.7e+06
		3er quantile	7.7e+06	3.8e+06	4.4e+06	4.5e+06	1.2e+07	4.8e+06
		Maximum	1.8e+07	2.8e+07	2.3e+07	1.1e+07	1.8e+07	1.4e+07
	Au cours de l'échange	1er quantile	7.8e+06	3.1e+06	3.7e+06	3.9e+06	1.2e+07	5.2e+06
		Moyenne	9.4e+06	3.7e+06	4.5e+06	4.6e+06	1.3e+07	6.2e+06
		3er quantile	1.1e+07	4e+06	5.1e+06	5.1e+06	1.4e+07	7e+06
		Maximum	1.8e+07	2e+07	1.2e+07	9.4e+06	1.8e+07	1.4e+07
Ma3 : Trafic de routage envoyé (bit/s)	Toute la simulation	1er quantile	0	0	2.2e+04	1e+05	1.9e+05	5.3e+04
		Moyenne	3.7e+05	2.1e+06	1.9e+05	1.4e+05	2.4e+05	7e+04
		3er quantile	8.2e+05	3.4e+06	1.9e+05	1.7e+05	2.9e+05	8.5e+04
		Maximum	2.8e+06	6.5e+06	3.7e+06	5.7e+05	7.1e+05	1.7e+05
	Au cours de l'échange	1er quantile	8.5e+05	1.8e+06	2.6e+04	7e+04	1.5e+05	5.5e+04
		Moyenne	1.1e+06	2.2e+06	1.1e+05	1.2e+05	1.9e+05	7e+04
		3er quantile	1.4e+06	2.8e+06	9.2e+04	1.5e+05	2.2e+05	8.4e+04
		Maximum	2.8e+06	4.2e+06	1.5e+06	4.1e+05	4.6e+05	1.5e+05
Ma4 : Trafic de routage reçu (bit/s)	Toute la simulation	1er quantile	0	0	2.4e+05	1.2e+06	2.3e+06	6.5e+05
		Moyenne	2.1e+06	1.8e+06	1.3e+06	1.7e+06	3e+06	8.6e+05
		3er quantile	4.5e+06	2.9e+06	1.3e+06	2.1e+06	3.8e+06	1.1e+06
		Maximum	1.6e+07	2.7e+07	2.2e+07	6.5e+06	8.2e+06	2.3e+06
	Au cours de l'échange	1er quantile	4.6e+06	9.9e+05	2.5e+05	6.6e+05	1.7e+06	6.3e+05
		Moyenne	6.4e+06	1.5e+06	6.8e+05	1.2e+06	2.1e+06	8.3e+05
		3er quantile	8.1e+06	1.8e+06	6.4e+05	1.6e+06	2.5e+06	1e+06
		Maximum	1.6e+07	1.8e+07	8.1e+06	4.6e+06	4.8e+06	1.7e+06
Ma5 : Taux de retransmission	Toute la simulation	1er quantile	1.8	1.5	1.6	1.7	0.52	0.45
		Moyenne	1.8	1.7	1.3	2.1	0.78	0.64
		3er quantile	2.2	1.8	2.7	2.4	0.62	0.61
		Maximum	10	10	10	10	10	6
	Au cours de l'échange	1er quantile	1.9	1.7	2.2	1.7	0.51	0.45
		Moyenne	2	1.8	2.5	2.1	0.57	0.54
		3er quantile	2.2	2	2.8	2.4	0.62	0.61
		Maximum	3	2.8	3.9	3.6	0.89	1.3
Ma6 : Nombre de saut	Toute la simulation	1er quantile	2.3	3.3	1	2.2	4.2	3
		Moyenne	4.3	4.6	2.8	2.5	4.6	3.4
		3er quantile	3.5	5	1.6	2.7	4.7	3.6
		Maximum	15	23	1e+02	10	12	8.8
	Au cours de l'échange	1er quantile	2.3	2.9	1.5	2.2	4.2	3
		Moyenne	2.8	3.7	1.8	2.5	4.4	3.3
		3er quantile	3.3	4.3	2	2.7	4.7	3.6
		Maximum	6.9	8.7	8.3	4.7	8	8.8
Ma7 : Délai WLAN (s)	Toute la simulation	1er quantile	0.062	3.8	0.0001	0.00012	0.00012	0.00013
		Moyenne	0.16	4.6	4.4	0.025	0.02	0.0046
		3er quantile	0.26	5.9	0.24	0.0028	0.0066	0.003
		Maximum	10	10	1e+02	10	10	2
	Au cours de l'échange	1er quantile	0.14	1.7	0.18	0.0038	0.0077	0.0033
		Moyenne	0.22	3	0.27	0.05	0.029	0.0077
		3er quantile	0.28	4.3	0.35	0.062	0.036	0.0062
		Maximum	1.3	6.6	0.88	0.59	0.73	0.18
Ma8 : Délai d'accès au médium (s)	Toute la simulation	1er quantile	0.11	3.8	0	6.7e-07	7e-06	0
		Moyenne	0.19	4.6	0.14	0.035	0.022	0.0061
		3er quantile	0.32	5.8	0.34	0.0071	0.012	0.0043
		Maximum	10	10	10	10	10	2
	Au cours de l'échange	1er quantile	0.22	1.8	0.33	0.0087	0.014	0.0046
		Moyenne	0.27	3	0.39	0.082	0.035	0.012
		3er quantile	0.33	4.3	0.45	0.11	0.046	0.011
		Maximum	0.78	6.1	0.76	0.59	0.47	0.18

Voir la section 4.2.2.1 pour la définition des métriques.

TABLEAU G.2 – Valeurs des métriques de performances pour le scénario Sa2 avec 15 noeuds sources

Métriques	Situation	Type	AODV	DSR	GRP	OLSR	QOLSR	CBL
Ma1 : Charge (bit/s)	Toute la simulation	1er quantile	0	0	2.6e+04	2.2e+05	4.3e+05	8.8e+04
		Moyenne	2.7e+06	8.8e+06	3.3e+06	1.7e+06	4.9e+06	2.5e+06
		3er quantile	6.3e+06	1.6e+07	9.1e+06	3.9e+06	1.4e+07	5.8e+06
		Maximum	1.8e+07	2.9e+07	1.6e+07	1e+07	2e+07	1.5e+07
	Au cours de l'échange	1er quantile	6.6e+06	1.7e+07	9.3e+06	4.1e+06	1.4e+07	6.3e+06
		Moyenne	8e+06	1.8e+07	9.9e+06	4.8e+06	1.5e+07	8.2e+06
		3er quantile	9e+06	2e+07	1.1e+07	5.1e+06	1.5e+07	9.5e+06
		Maximum	1.8e+07	2.9e+07	1.6e+07	1e+07	2e+07	1.5e+07
Ma2 : Débit (bit/s)	Toute la simulation	1er quantile	0	0	2.9e+05	2.7e+06	5.5e+06	1.1e+06
		Moyenne	4e+06	2.9e+06	2.5e+06	3.8e+06	8.8e+06	3.5e+06
		3er quantile	9.4e+06	4.4e+06	4.6e+06	4.6e+06	1.4e+07	6.7e+06
		Maximum	2.3e+07	2.7e+07	2.3e+07	1.3e+07	2e+07	1.4e+07
	Au cours de l'échange	1er quantile	9.2e+06	3.9e+06	4.1e+06	4.1e+06	1.4e+07	7.1e+06
		Moyenne	1e+07	4.5e+06	4.9e+06	4.6e+06	1.5e+07	8.8e+06
		3er quantile	1.2e+07	4.8e+06	5.6e+06	5e+06	1.5e+07	1e+07
		Maximum	2.3e+07	2e+07	1.2e+07	8.3e+06	2e+07	1.4e+07
Ma3 : Trafic de routage envoyé (bit/s)	Toute la simulation	1er quantile	0	0	2.2e+04	9.5e+04	1.8e+05	5.1e+04
		Moyenne	5.6e+05	2.2e+06	1.7e+05	1.4e+05	2.5e+05	6.7e+04
		3er quantile	1.3e+06	3.6e+06	1.7e+05	1.6e+05	3e+05	8.2e+04
		Maximum	4.5e+06	8.4e+06	3.7e+06	6.1e+05	8.4e+05	1.7e+05
	Au cours de l'échange	1er quantile	1.3e+06	1.8e+06	3.1e+04	6.6e+04	1.2e+05	4.8e+04
		Moyenne	1.6e+06	2.4e+06	1.1e+05	9e+04	1.6e+05	6.2e+04
		3er quantile	1.8e+06	3.1e+06	1.4e+05	1.1e+05	2e+05	7.6e+04
		Maximum	4.5e+06	5.2e+06	1.2e+06	2.3e+05	3.9e+05	1.4e+05
Ma4 : Trafic de routage reçu (bit/s)	Toute la simulation	1er quantile	0	0	2.5e+05	9.3e+05	1.8e+06	5.9e+05
		Moyenne	3e+06	2e+06	1.1e+06	1.6e+06	3e+06	8.1e+05
		3er quantile	7.1e+06	3.2e+06	1e+06	2e+06	3.9e+06	1e+06
		Maximum	2e+07	2.6e+07	2.3e+07	7.2e+06	1.1e+07	2.1e+06
	Au cours de l'échange	1er quantile	6.6e+06	1.1e+06	2.6e+05	5.6e+05	1.2e+06	5e+05
		Moyenne	7.9e+06	2e+06	6.2e+05	7.7e+05	1.6e+06	6.8e+05
		3er quantile	9.3e+06	2.6e+06	7e+05	9.5e+05	1.9e+06	8.4e+05
		Maximum	2e+07	1.9e+07	6.7e+06	2.3e+06	3.3e+06	1.7e+06
Ma5 : Taux de retransmission	Toute la simulation	1er quantile	1.7	1.4	1.9	1.9	0.65	0.52
		Moyenne	1.7	1.6	16	2.2	0.93	0.63
		3er quantile	2.1	1.7	2.8	2.4	0.74	0.63
		Maximum	15	15	97	15	15	5.1
	Au cours de l'échange	1er quantile	1.8	1.6	2.1	1.9	0.65	0.53
		Moyenne	2	1.8	2.4	2.1	0.69	0.58
		3er quantile	2.2	1.9	2.7	2.4	0.74	0.63
		Maximum	3	2.8	4.1	3.7	0.92	0.93
Ma6 : Nombre de saut	Toute la simulation	1er quantile	1.6	2.8	1	1.7	2.7	2.1
		Moyenne	3.8	4	1.3	1.9	3.2	2.4
		3er quantile	2.5	4.6	1.4	1.9	3.3	2.5
		Maximum	15	15	21	15	15	5.9
	Au cours de l'échange	1er quantile	1.5	2.2	1.4	1.7	2.7	2.1
		Moyenne	1.8	2.9	1.7	1.8	3	2.3
		3er quantile	2.1	3.4	1.8	1.9	3.3	2.5
		Maximum	4.6	8.5	6.3	5.6	6.2	4.8
Ma7 : Délai WLAN (s)	Toute la simulation	1er quantile	0.15	3.5	0.0001	0.00012	0.00012	0.00013
		Moyenne	0.5	4.2	0.56	0.086	0.089	0.015
		3er quantile	0.82	5.3	0.41	0.089	0.15	0.0097
		Maximum	15	15	26	15	15	2
	Au cours de l'échange	1er quantile	0.52	2	0.36	0.11	0.15	0.011
		Moyenne	0.68	2.9	0.49	0.24	0.23	0.04
		3er quantile	0.87	4	0.61	0.32	0.27	0.042
		Maximum	2	5.8	1.6	1.1	1.8	0.6
Ma8 : Délai d'accès au médium (s)	Toute la simulation	1er quantile	0.24	3.3	0	8e-07	7.2e-06	0
		Moyenne	0.54	4.1	0.69	0.12	0.095	0.018
		3er quantile	0.89	5.1	0.6	0.16	0.19	0.012
		Maximum	15	15	26	15	15	2
	Au cours de l'échange	1er quantile	0.62	2.1	0.55	0.19	0.19	0.013
		Moyenne	0.76	2.8	0.64	0.35	0.24	0.048
		3er quantile	0.93	3.8	0.75	0.45	0.27	0.053
		Maximum	1.7	5.4	1.3	1.3	0.66	0.53

Voir la section 4.2.2.1 pour la définition des métriques.

TABLEAU G.3 – Valeurs des métriques de performances pour le scénario Sa3 avec 20 noeuds sources

Métriques	Situation	Type	AODV	DSR	GRP	OLSR	QOLSR	CBL
Ma1 : Charge (bit/s)	Toute la simulation	1er quantile	0	0	2.6e+04	2.2e+05	4.2e+05	4.9e+04
		Moyenne	4.3e+06	1.1e+07	4.8e+06	2.3e+06	7e+06	2.9e+06
		3er quantile	9.6e+06	2.2e+07	1.4e+07	5.5e+06	2.1e+07	1.7e+05
		Maximum	2.6e+07	3.8e+07	2e+07	1.4e+07	2.6e+07	2.3e+07
	Au cours de l'échange	1er quantile	9.9e+06	2.3e+07	1.4e+07	5.7e+06	2.1e+07	8.3e+06
		Moyenne	1.2e+07	2.5e+07	1.5e+07	6.9e+06	2.2e+07	9.3e+06
		3er quantile	1.3e+07	2.7e+07	1.6e+07	6.9e+06	2.3e+07	1.2e+07
		Maximum	2.6e+07	3.8e+07	2e+07	1.4e+07	2.6e+07	2.3e+07
Ma2 : Débit (bit/s)	Toute la simulation	1er quantile	0	0	3.3e+05	2.8e+06	5.4e+06	8.6e+05
		Moyenne	5e+06	3.6e+06	2.6e+06	4e+06	9.8e+06	4e+06
		3er quantile	1.1e+07	5.5e+06	4.5e+06	4.8e+06	1.7e+07	8.2e+06
		Maximum	2.2e+07	3e+07	2.2e+07	1.3e+07	2.1e+07	1.8e+07
	Au cours de l'échange	1er quantile	1.1e+07	5.2e+06	3.8e+06	4.3e+06	1.7e+07	9.5e+06
		Moyenne	1.2e+07	5.9e+06	4.6e+06	4.7e+06	1.8e+07	1.1e+07
		3er quantile	1.3e+07	6.3e+06	5.2e+06	5.1e+06	1.9e+07	1.2e+07
		Maximum	2.2e+07	2.4e+07	9.5e+06	7.3e+06	2.1e+07	1.8e+07
Ma3 : Trafic de routage envoyé (bit/s)	Toute la simulation	1er quantile	0	0	2.2e+04	9.1e+04	1.6e+05	5e+04
		Moyenne	7.1e+05	2.4e+06	1.9e+05	1.4e+05	2.4e+05	1.5e+05
		3er quantile	1.6e+06	3.8e+06	2.1e+05	1.6e+05	3e+05	8.8e+04
		Maximum	5e+06	1.1e+07	3.5e+06	6.2e+05	8.9e+05	1.9e+06
	Au cours de l'échange	1er quantile	1.6e+06	2.3e+06	2.6e+04	6.2e+04	1.1e+05	4.9e+04
		Moyenne	1.9e+06	3e+06	1e+05	8.4e+04	1.4e+05	2e+05
		3er quantile	2.1e+06	3.6e+06	1.5e+05	1e+05	1.7e+05	9.1e+04
		Maximum	5e+06	6.7e+06	9.7e+05	2.1e+05	3.7e+05	1.5e+06
Ma4 : Trafic de routage reçu (bit/s)	Toute la simulation	1er quantile	0	0	2.5e+05	8.3e+05	1.5e+06	5e+05
		Moyenne	3.6e+06	2.4e+06	1.1e+06	1.7e+06	2.9e+06	7.1e+05
		3er quantile	7.8e+06	3.8e+06	1e+06	2.1e+06	3.9e+06	9.4e+05
		Maximum	1.9e+07	2.9e+07	2.2e+07	7.4e+06	1.2e+07	1.9e+06
	Au cours de l'échange	1er quantile	7.3e+06	1.7e+06	2.8e+05	4.8e+05	9.3e+05	4.5e+05
		Moyenne	8.4e+06	2.9e+06	5.9e+05	6.6e+05	1.2e+06	6.2e+05
		3er quantile	9.7e+06	3.6e+06	7.2e+05	8.2e+05	1.5e+06	7.8e+05
		Maximum	1.9e+07	2.3e+07	4e+06	1.5e+06	3.1e+06	1.5e+06
Ma5 : Taux de retransmission	Toute la simulation	1er quantile	1.6	1.3	2	2	0.66	0.00015
		Moyenne	1.6	1.5	1.8	2.3	0.9	0.34
		3er quantile	2.1	1.6	3.1	2.5	0.74	0.61
		Maximum	20	20	20	20	20	2.1
	Au cours de l'échange	1er quantile	1.8	1.5	2.6	2	0.66	0.48
		Moyenne	2	1.7	2.9	2.3	0.7	0.46
		3er quantile	2.2	1.8	3.2	2.5	0.73	0.62
		Maximum	3	3.1	4.2	3.6	0.89	0.8
Ma6 : Nombre de saut	Toute la simulation	1er quantile	1.8	2.9	1	1.6	2.7	1
		Moyenne	4.7	4	1.2	1.8	3.2	1.3
		3er quantile	4	4.5	1.4	1.9	3.3	2.1
		Maximum	20	20	22	20	20	3.4
	Au cours de l'échange	1er quantile	1.7	2.5	1.3	1.6	2.7	1.7
		Moyenne	2	3	1.5	1.8	3	1.8
		3er quantile	2.3	3.5	1.6	1.9	3.2	2.2
		Maximum	5.8	6	2.9	6.2	5.5	3.4
Ma7 : Délai WLAN (s)	Toute la simulation	1er quantile	0.047	3.3	0.0001	0.00012	0.00012	0.00013
		Moyenne	0.6	3.8	0.26	0.16	0.13	0.045
		3er quantile	1.1	4.6	0.58	0.3	0.24	0.057
		Maximum	20	20	20	20	20	2
	Au cours de l'échange	1er quantile	0.7	2.4	0.54	0.32	0.25	0.045
		Moyenne	0.88	2.8	0.71	0.46	0.33	0.11
		3er quantile	1.1	3.6	0.88	0.56	0.38	0.17
		Maximum	2.1	4.9	1.9	1.5	1.7	0.6
Ma8 : Délai d'accès au médium (s)	Toute la simulation	1er quantile	0.06	3.1	0	1.5e-06	7.2e-06	0
		Moyenne	0.59	3.8	0.31	0.21	0.13	0.053
		3er quantile	1	4.6	0.8	0.45	0.27	0.074
		Maximum	20	20	20	20	20	2
	Au cours de l'échange	1er quantile	0.74	2.3	0.81	0.47	0.27	0.097
		Moyenne	0.87	2.7	0.89	0.61	0.31	0.15
		3er quantile	1.1	3.4	1	0.72	0.35	0.21
		Maximum	1.8	4.3	1.4	1.5	0.72	0.49

Voir la section 4.2.2.1 pour la définition des métriques.

