



HAL
open science

Proposition d'une structuration dynamique d'un réseau de communication intervéhiculaire pour les ITS

Lucas Rivoirard, Martine Wahl, Patrick Sonni, Marion Berbineau, Dominique
Gruyer

► **To cite this version:**

Lucas Rivoirard, Martine Wahl, Patrick Sonni, Marion Berbineau, Dominique Gruyer. Proposition d'une structuration dynamique d'un réseau de communication intervéhiculaire pour les ITS. RDMI 2018, Les rencontres de la mobilité intelligente, ATEC ITS FRANCE, Jan 2018, Paris, France. 11p. hal-01777885v2

HAL Id: hal-01777885

<https://hal.science/hal-01777885v2>

Submitted on 15 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proposition d'une structuration dynamique d'un réseau de communication intervéhiculaire pour les ITS

Rivoirard Lucas, Martine Wahl, Patrick SonDi, Marion Berbineau, Dominique Gruyer

1. Enjeux et objectifs

Les systèmes d'aide à la conduite automobile ont pour objectif de réduire les risques routiers. Ils s'appuient notamment sur l'automatisation des véhicules et la mise en œuvre de la communication inter-véhiculaire permettant la coordination et la coopération entre les véhicules.

Les réseaux véhiculaires *ad hoc* (VANET – *Vehicular Ad hoc Network*) permettent d'imaginer une large gamme d'application pour les Systèmes de transports intelligents (STI) en termes de confort, d'optimisation du trafic routier ou de sécurité. De nombreux défis sont à relever tels que le partage des ressources du canal de communication concernant la bande passante, les contraintes de la limitation des durées des connexions entre les véhicules et celles de la densité disparate du trafic routier. Ces systèmes devront en outre faire face aux évolutions technologiques des normes et standards de communication et à l'hétérogénéité des solutions le temps du renouvellement des matériels.

2. État de l'art, descriptif de la situation

Il existe deux architectures principales conçues pour les VANETs. L'une utilise les communications véhicule à infrastructure (V2I, *Vehicle to Infrastructure*), l'autre les communications entre véhicules (V2V, *Vehicle to Vehicle*) comme l'illustre la Figure 1.

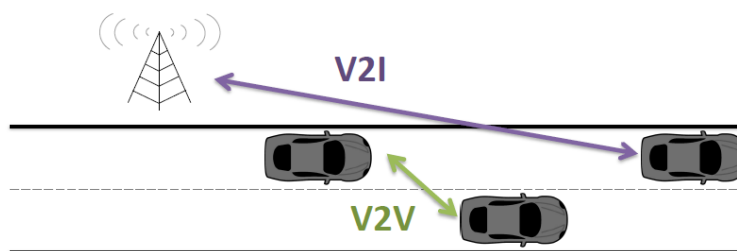


Figure 1. Illustration des échanges dans une architecture de communication V2V et V2I

Dans une architecture V2I, les équipements de communication, appelés unités de bord de route (UBR), sont déployés le long de la chaussée à intervalles réguliers. L'architecture V2I offre une centralisation des messages, ce qui permet une planification et une optimisation par le gestionnaire de voirie du trafic de communication. En contrepartie, le coût du déploiement des UBRs est significatif à l'installation et lors de leur maintenance. De nombreux défis seront à relever comme la gestion de l'obsolescence des équipements ; l'interopérabilité entre les matériels d'anciennes et nouvelles générations ; la compatibilité entre les diverses applications des STIs ; et la continuité (en cas de panne d'UBRs) du service de communication V2I qui supportera des applications d'assistance au conducteur.

Dans une architecture V2V, les véhicules communicants (appelés nœuds du réseau V2V) collaborent de manière décentralisée pour former un réseau *ad hoc*. Le fonctionnement des réseaux V2V est modulaire et flexible, il reste inchangé lors d'une panne d'un nœud du réseau, sous condition toutefois d'un nombre de nœuds suffisants. Cependant, la communication V2V nécessite, pour acheminer les messages, la mise en œuvre d'algorithmes de routage distribués. Ceux-ci sont plus complexes que ceux employés dans une architecture V2I. De plus, la solution V2V, à l'image de celle V2I, devra faire face aux évolutions technologiques et par conséquent à un parc de véhicules communicants technologiquement hétérogènes.

Vraisemblablement, les architectures V2I et V2V devront fonctionner ensemble, se compléter et coopérer afin d'assurer une continuité de service. Dans ce papier, nous nous intéressons à la construction d'une infrastructure virtuelle dynamiquement stable qui repose sur des communications V2V de sorte à offrir une solution V2V qui permettrait de pallier une panne, par exemple des UBRs, panne qui entrainerait une rupture du service de communication V2I. Cette structure aura également pour objectif d'organiser les échanges entre les véhicules en limitant, dans le cas d'une communication point-à-point le nombre de véhicules intermédiaire qui rediffuseront le message initié par un véhicule à un ou des véhicules destinataires.

3. Méthodes innovantes

3.1. Introduction au protocole CBL

Nous avons développé l'algorithme Chaîne, branche et feuille (CBL –*Chain-Branch-Leaf*) qui offre au réseau V2V les services d'une nouvelle topologie de réseau *ad hoc* [1]. Cette topologie optimise l'utilisation du réseau de communication au moyen de la définition de régions d'intérêt. CBL établit des liens et des regroupements (*clusters*) entre véhicules en fonction de la mobilité des véhicules et de la qualité de leur lien de communication. L'algorithme construit une structure fédératrice qui rend possible aussi bien des communications entre véhicules proches (par exemple pour le partage de variables au moyen de la transmission de messages CAM^{1,2}) que des communications entre véhicules lointains (par exemple pour la transmission de messages d'alertes via des messages DENM^{1,2}). 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).

Dans la suite de notre présentation, nous supposerons que chaque véhicule est équipé d'un système de localisation de type GPS et d'une interface de communication *ad hoc* sans fil de même technologie (par exemple une interface 802.11p). Cette dernière permet la communication entre véhicules dans la limite de la portée de transmission en visibilité directe. D'autre part, pour cet article, l'algorithme CBL est implanté dans le protocole de routage proactif OLSR [2].

¹ Cooperative Awareness Message (CAM) ; Decentralized Environmental Notification Message (DENM)

² L'ETSI a publié deux standards européens pour les ITS qui définissent les ensembles de message nécessaire pour le fonctionnement des applications ITS de sécurité coopératives et critiques : le standard EN 302 637 relatif à la spécification du service « Cooperative Awareness Basic Service » et le standard EN 302 637-3 relatif à la spécification du service « Decentralized Environmental Notification Basic Service ». <http://www.etsi.org/news-events/news/851-2014-12-press-etsi-publishes-european-standards-for-intelligent-transport-systems>

3.2. Définitions

CBL est un algorithme pro-actif distribué pour lequel chaque nœud (véhicule communicant) initie et gère localement son propre processus. L'algorithme crée une hiérarchie entre les nœuds afin de construire des *clusters* à 1 saut de sorte que chaque nœud d'un groupe puisse communiquer directement avec son groupe sans passer par un nœud intermédiaire. CBL définit deux types de nœuds : les nœuds « branche » (*Branch*) et les nœuds « feuille » (*Leaf*). Tout nœud (branche ou feuille) émet et reçoit des messages de routage HELLO, ce qui lui permet de s'annoncer (émission) et de découvrir (réception) les nœuds de son voisinage à 1 saut.

Un **nœud branche** est responsable d'un groupe de nœuds feuille (*cluster*). Il est élu par d'autres nœuds (branche ou feuille). Outre l'émission des messages applicatifs dont il est l'initiateur (la source), il a un rôle de relais pour des messages applicatifs émis entre d'autres véhicules source et destinataire(s) distants de plus d'un saut. Il participe à la construction d'une chaîne de véhicules relais circulant dans une même direction. Il dispose du droit d'émettre les messages de contrôle de topologie (TC) définis dans le protocole OLSR. Les messages TC permettent de partager les tables de routages au sein du réseau. En outre, pour contrôler la propagation d'un message en fonction de l'application spécifiée dans le champ d'en-tête du message, un nœud branche a la capacité de retransmettre les messages à destinations d'un ensemble de nœuds identifiés (voir figure 2) : à ses nœuds feuilles ; au nœud branche situé en amont ; au nœud branche situé en aval ; à tous les nœuds branche (y compris ceux circulant dans la direction opposée).

Un **nœud feuille** est un nœud ordinaire qui établit un lien avec le nœud branche le plus proche. Lorsqu'il ne détecte pas dans son voisinage à 1 saut de nœud branche, il élit comme nœud branche le nœud feuille de son voisinage dont la vitesse de déplacement est la plus faible et qui circule dans la même direction. Un nœud feuille ne transmet que les messages applicatifs dont il est à l'origine par diffusion à destination de son voisinage à un saut ou à son nœud branche.

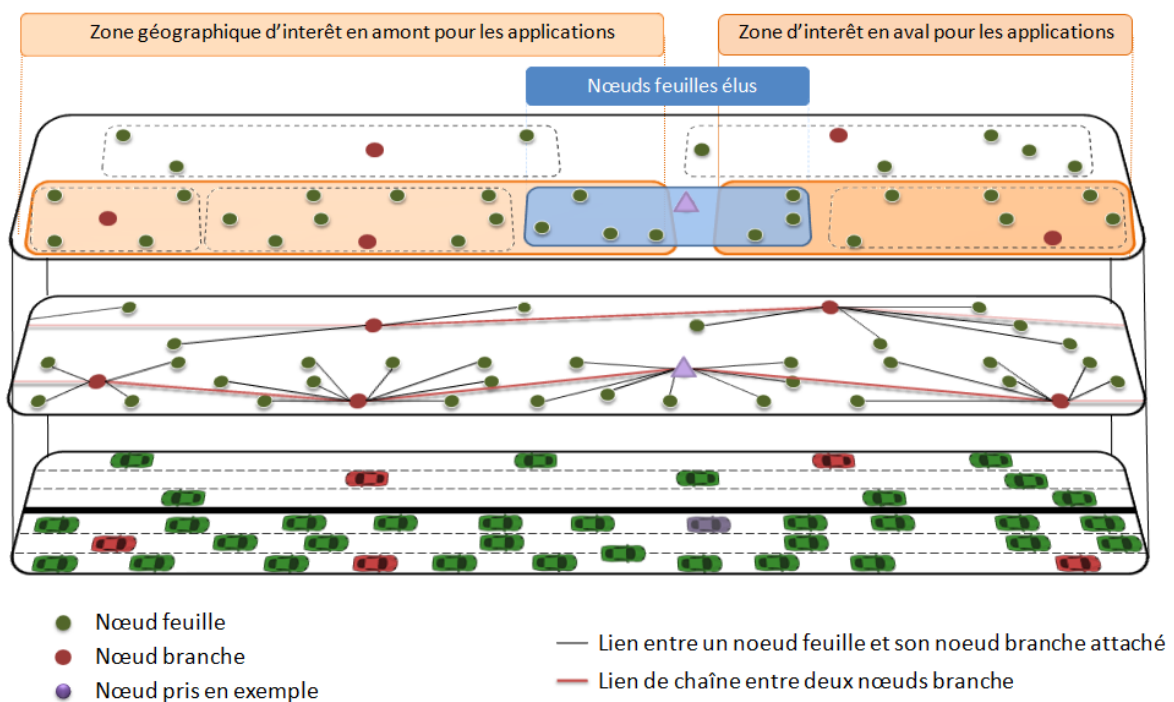


Figure 2. Nœuds et liens fonctionnels de CBL : en bas, illustration d'un trafic routier ; au milieu, nœuds CBL correspondant aux véhicules représentés et liens entre nœuds ; en haut, mise en valeur des zones d'intérêt en amont et aval d'un nœud branche particulier.

Une **chaîne** est un réseau fédérateur (*backbone*) virtuel composé d'une séquence de nœuds branche. Idéalement, une chaîne est créée par direction de trafic routier. Les chaînes offrent ainsi aux nœuds branche du réseau de communication *ad hoc* des chemins pour transmettre les messages applicatifs sur des longues distances. Dans le contexte d'une route, c'est-à-dire d'une infrastructure linéaire, les chaînes se comportent comme un réseau virtuel similaire à celui qui peut être obtenu avec une infrastructure.

Chaque nœud branche est acteur de la constitution de la chaîne dont il fait partie. Il veille à élire et maintenir une liaison valide avec un nœud branche en amont et un en aval de son sens de circulation. Pour le choix de ses nœuds branche, amont et aval, il utilise comme métrique le temps de connexion.

Le **temps de connexion** ou temps de contact (CT) est le temps pendant lequel deux nœuds N_i et N_j pourraient communiquer si leur vitesse de déplacement restait inchangée. La valeur de CT est calculée au moyen de l'équation (1) qui prend en compte la position en coordonnées cartésiennes des deux nœuds ($[X_i, Y_i]$ pour le nœud N_i), leur vitesses et leur direction de déplacement ($[V_i, \sigma_i]$ pour le nœud N_i).

$$(1) \quad CT = \frac{-(ab + cd) + \sqrt{(a^2 + c^2) * P^2 - (ab - bc)^2}}{a^2 + c^2}$$

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

Les illustrations, figure 3, présentent les temps de connexion entre deux véhicules en fonction de la portée de transmission et de leur vitesse relative dans un cas idéal de communication quand les véhicules se déplacent en sens opposé (illustration de gauche) et dans le même sens (illustration de droite). Ainsi, les valeurs maximales du temps de connexion pour deux véhicules circulant en sens opposé lorsque la portée est de 500 m (valeur typique du 802.11p) sont comprises entre 5 et 20 secondes. Cependant, à cette même portée de 500 m, lorsque les véhicules circulent dans le même sens à une vitesse relative faible, inférieure à 10 km/h, le temps de connexion peut dépasser 250 secondes.

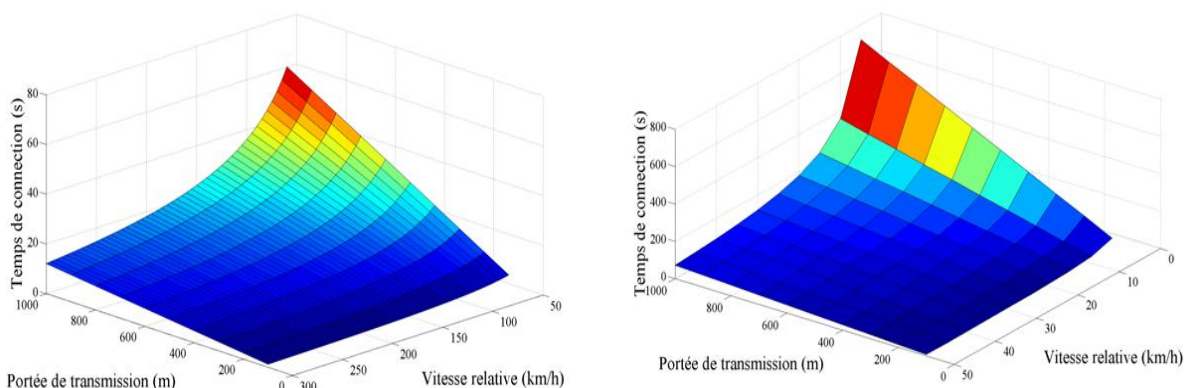


Figure 3. Temps de connexion en fonction de la vitesse relative et de la portée de transmission dans le cas : à gauche, de deux véhicules en sens opposé de circulation ; à droite, de deux véhicules dans le même sens de circulation.

3.3. Fonctionnement de l'algorithme

À l'initialisation du réseau, aucun groupe n'est formé, tous les nœuds sont définis comme des nœuds feuilles ordinaires. Les messages HELLO échangés par les nœuds enclenchent le mécanisme de création de la topologie. En plus des informations habituellement transmises dans un message HELLO d'OLSR, un message HELLO de CBL comporte les informations de position et de type (branche ou feuille) du nœud qui l'émet.

À chaque fois qu'un nœud N_i reçoit un message HELLO en provenance d'un nœud voisin N_j , il déclenche la procédure illustrée en figure 5. 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 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 si N_i n'a plus reçu de messages HELLO d'aucun des nœuds voisins qui l'avaient élu branche depuis une certaine période. L'algorithme 2 a pour fonction de transformer le nœud feuille N_i en un nœud branche dans le cas où le message HELLO du nœud expéditeur N_j l'informe que N_j l'a élu comme nœud branche, c'est-à-dire en tant que chef du groupe qu'il rejoint.

À l'issue des algorithmes 5 ou 2, selon la nature résultante du nœud récepteur N_i , branche ou feuille, l'algorithme 4 (si N_i est un nœud branche) ou l'algorithme 3 (si N_i est un nœud feuille) est exécuté. 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. L'algorithme 3 permet au nœud feuille N_i de réaliser, vérifier ou maintenir l'élection de son nœud branche associé. Ce nœud branche élu, et associé au nœud N_i , sera le relais vers la chaîne des messages de N_i qui doivent être acheminés sur une longue distance, en amont ou en aval du trafic routier.

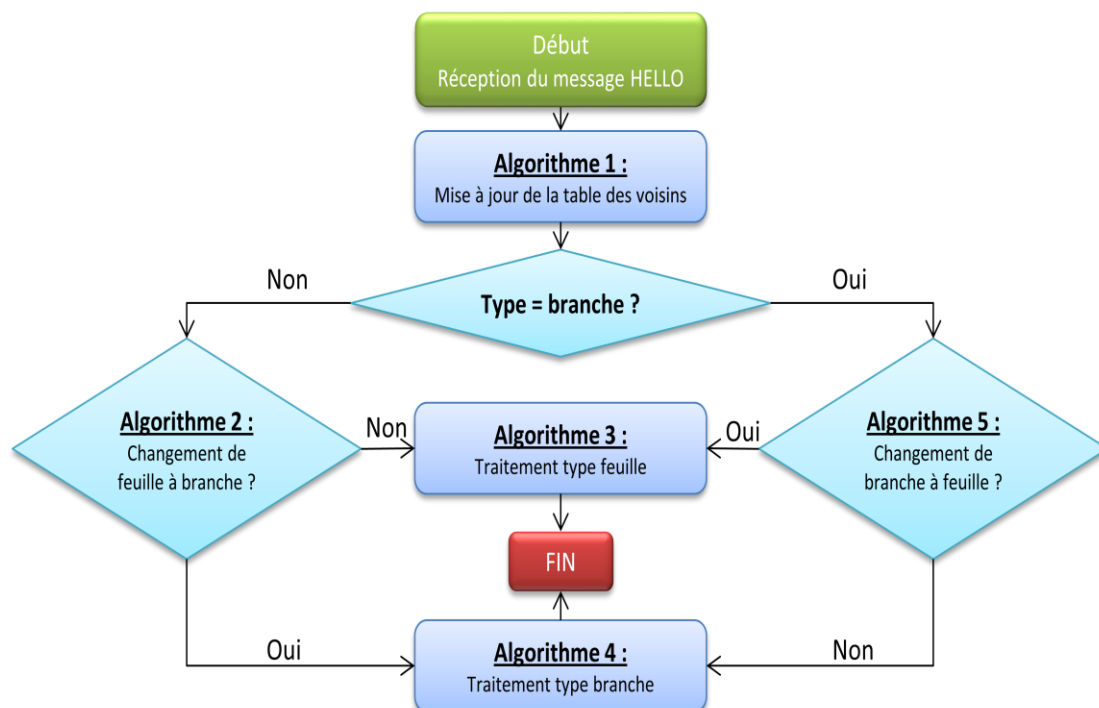


Figure 4. Algorithme exécuté à de la réception d'un message HELLO

4. Évaluation de performances et interprétations

L'algorithme CBL a été implanté sous le logiciel de simulations à événements discrets de protocoles de communication OPNET Riverbed Modeler. Les performances de CBL sont présentées comparativement à celles obtenues avec le protocole pro-actif OLSR [2] en prenant en compte les mêmes conditions de trafic routier. Nous présentons d'abord, ci-dessous, le trafic routier modélisé, puis le « système réseau » modélisé. Enfin, les résultats obtenus sont analysés.

4.1. Route et trafic routier simulés

Afin d'avoir un trafic routier réaliste pour l'évaluation des performances de l'algorithme de groupement pro-actif CBL pour les communications *ad hoc* V2V, nous avons modélisé les trajectoires de véhicules circulant sur une section de 10 km de l'autoroute A27 qui relie la ville française Lille à la Belgique (figure 5), et celles de véhicules sur la départemental D90. L'autoroute A27 est une autoroute à deux fois deux voies. La départemental D90 est une route à deux voies traversant perpendiculairement la section A27 étudiée.

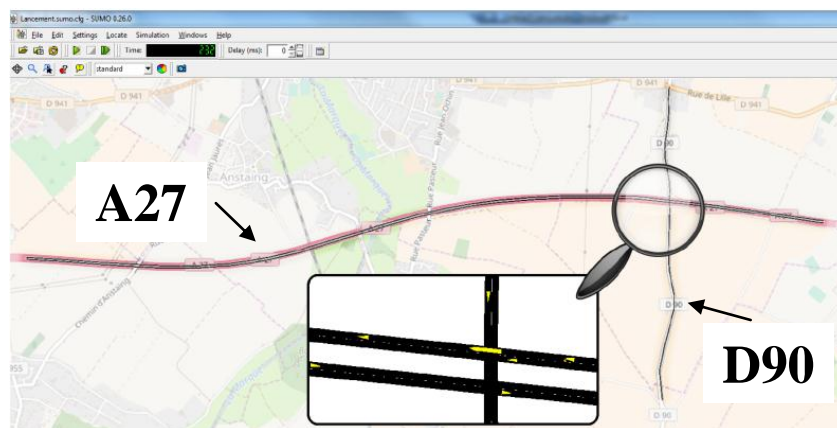


Figure 5. Modélisation SUMO du réseau routier constitué de l'autoroute A27 à deux fois deux voies (route horizontale) et de la route à deux voies à sens unique D90 (route verticale).

La modélisation des trajectoires des véhicules a bénéficiée des données réelles du trafic routier de l'A27, enregistré le 6 avril 2017 à partir de 12h35 par la DIRN (Direction interdépartementale des routes du Nord). Ces données, représentées en figure 6, ont été mesurées par les boucles électromagnétiques placées à l'intérieur de la chaussée.

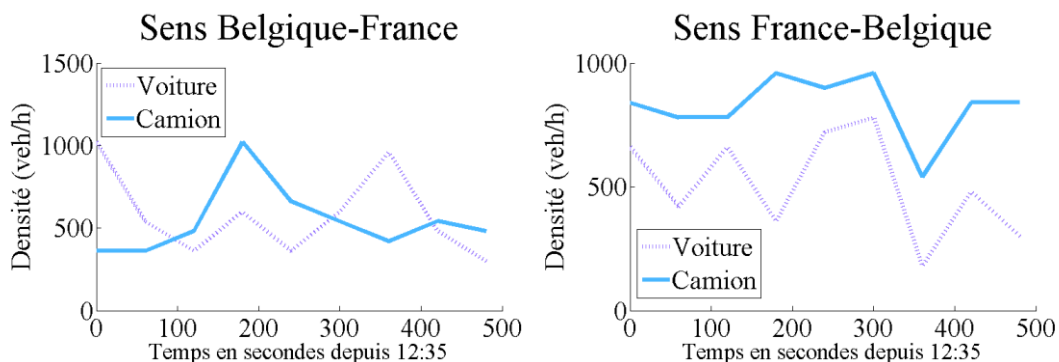


Figure 6. Densité de véhicules sur l'autoroute A27 le 6 avril 2017 à compter de 12 :35, obtenue à partir de boucles électromagnétiques placées sous la chaussée en fonction du type de véhicule

Pour transformer ces données sur l'A27 en trajectoire de véhicules, nous avons utilisé le logiciel SUMO [3]. Nous avons d'autre part pris comme hypothèse pour générer les trajectoires sur la D90 celle d'un trafic de 500 voitures par heure.

Le modèle de poursuite par défaut inclus dans le simulateur SUMO est une variante du modèle Krauß : chaque véhicule circule jusqu'à sa vitesse désirée, tout en maintenant une distance de sécurité parfaite avec le véhicule leader. La limite de vitesse est fixée à 130 km/h, ce qui correspond à la limite légale de vitesse sur les autoroutes en France. Nous avons défini deux types de véhicules : les voitures et les camions. Ces véhicules sont modélisés par un ensemble de paramètres dont les valeurs sont résumées dans le tableau 1 ci-dessous.

Tableau 1. Paramètres de configuration des types de véhicules pour le logiciel SUMO

Symbole	Description	Unité	Valeur pour les voitures	Valeur pour les camions
Acc	Capacité d'accélération des véhicules	m/s ²	2	1
Dcc	Capacité de décélération des véhicules	m/s ²	3	2
L	Longueur du véhicule	m	5	15
σ	Imperfection du pilote	[0,1]	0.5	0.5
τ	Temps minimum désiré par le conducteur	s	1	1
MinGap	Interdistance entre deux véhicules à l'arrêt	m	2.5	5
MaxSpeed	Vitesse maximale du véhicule	Km/h	150	130
SpeedFactor	Coefficient multiplicateur de limite de vitesse	-	1	0.84
SpeedDev	Écart-type du speedFactor	-	0.1	0.1

4.2. Système réseau simulé

Le système simulé est un réseau *ad hoc* de véhicules équipés de cartes IEEE 802.11p configurées avec un débit de 12 Mbps. La puissance de transmission est réglée sur 0,005 W et la sensibilité du récepteur sur -95 dBm afin d'obtenir une portée de communication de 500 m pour correspondre à une communication de type IEEE 802.11p.

Nous avons utilisé pour nos simulations les modules d'interfaces 802.11p et le protocole OLSR modélisés par OPNET Riverbed Modeler. Le protocole de *clustering* CBL a été implanté par nos soins sous OPNET en modifiant une copie du protocole OLSR. D'autre part, à chaque nœud mobile du réseau, nous avons affecté une des trajectoires simulées sous SUMO qui ont été converties au préalable dans un format utilisable par le simulateur de protocoles.

Le paramétrage du trafic de routage émis par les protocoles de routage pro-actif OLSR et de *clustering* CBL est le même. Afin de prendre en compte l'aspect dynamique des réseaux véhiculaires, nous avons considéré deux cas. Un cas A adopte les valeurs des paramètres définies par défaut dans le document RFC 3626 de l'IETF (*Internet Engineering Task Force*). Un cas B prend pour valeur des paramètres la moitié de celles définies pour le cas A. Ainsi, dans le cas A, chaque nœud émet un message HELLO de découverte de son voisinage toutes les deux secondes. Dans le cas B, leur périodicité d'émission est d'une seconde.

4.3. Résultats et analyse

Les nœuds relais sont en charge de la diffusion périodique des messages TC de contrôle de topologie et de la rediffusion de messages applicatifs en *broadcast*. La limitation de leur nombre dans un réseau sans fil est importante pour optimiser l'usage de la bande passante, ressource restreinte du réseau. Plus les nœuds relais sont nombreux, plus le nombre de rediffusion de paquets augmente, et plus la probabilité de collisions sera forte, jusqu'au risque de saturation de la bande passante. Étant donné l'importance de ces nœuds dans le trafic de routage émis, nous nous intéressons donc ci-dessous au nombre de nœuds relais générés par les protocoles OLSR et CBL dans le contexte d'un réseau routier comportant 140 véhicules. Nous étudions aussi la quantité de trafic de routage émis.

La figure 7 présente le nombre de nœuds « relais » comptabilisé en fonction du temps pour les protocoles OLSR (nombre de nœuds MPR) et CBL (nombre de nœuds « branche ») pour les cas de paramétrage A et B. Une fois le réseau simulé stable (après environ les 150 premières secondes de simulation), OLSR compte près de cent nœuds MPR, soit près de 71% des nœuds du réseau, contre une cinquantaine de nœuds branche pour CBL, soit environ 36%. Ainsi, CBL génère moins de nœuds relais qu'OLSR. Ce résultat s'explique par le fait que CBL force chaque nœud feuille, respectivement branche, à ne sélectionner qu'un seul nœud relais, respectivement deux, alors qu'OLSR n'impose aucune contrainte lors du choix des MPRs.

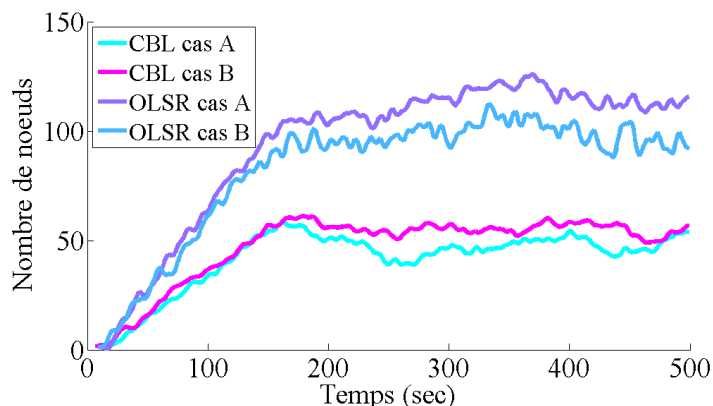


Figure 5. Nombre de nœuds « relais » d'OLSR (nombre de MPRs) et de CBL (nombre de « branche »)

La figure 8 montre le débit de transmission des messages de contrôle de topologie TC envoyés et rediffusés, courbes (a) à gauche, et de recherche de voisinage HELLO, courbes (b) de droite.

La figure 8.(a) met en évidence qu'OLSR (qui compte environ 71% de nœuds MPR) génère de 3 à 17 fois plus de trafic TC que CBL (qui ne compte qu'environ 36% de nœuds branche). En effet le nombre de messages TC est lié au nombre de nœuds relais.

La Figure 6.(b) montre que le débit de messages HELLO est quasiment le même pour les protocoles OLSR et CBL dans le cas A, respectivement dans le cas B. Ceci est en cohérence avec le fait que le nombre de messages HELLO est directement lié au nombre de nœuds sur le réseau VANET (ici 140 nœuds). Il est cependant très légèrement supérieur pour CBL en raison de la longueur légèrement supérieure des messages HELLO de CBL : CBL 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.

La Figure 6.(b) montre également que le débit est deux fois plus élevé dans le cas B que dans le cas A, ce qui est conforme à la périodicité d'émission des messages HELLO par chaque nœud pour chacun de ces cas : les fréquences d'envoi des messages HELLO sont doublés dans le cas B par rapport au cas A (§4.2).

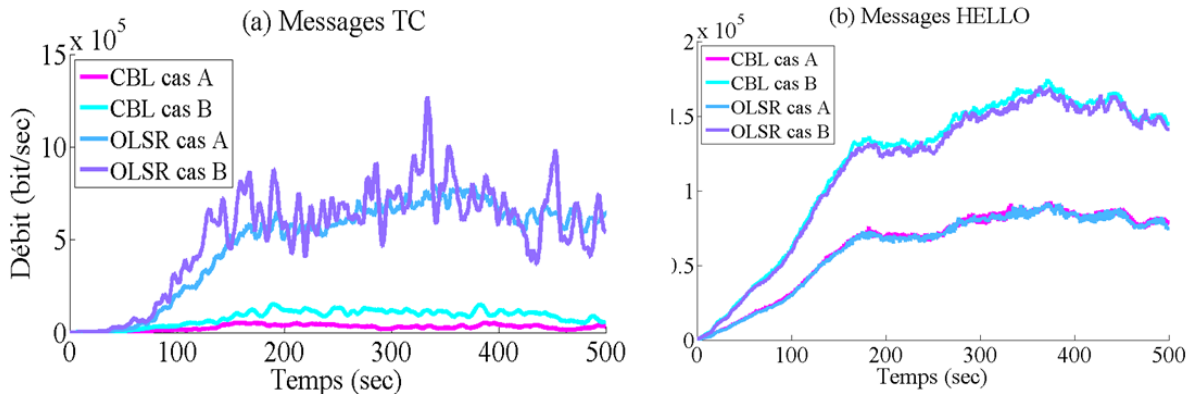


Figure 6. Débit de transmission des messages de routage en bits par secondes : (a) les messages TC; (b) les messages HELLO. Un facteur dix distingue les échelles de l'axe des ordonnées des deux figures.

Si l'on considère maintenant le nombre total de bits transmis pour les messages de trafic de routage (TC et HELLO) dans l'intervalle de temps [150 s, 500 s], Figure 7, nous observons que le trafic de routage global généré est bien inférieur pour CBL, dans les deux cas de simulation A et B. Le trafic de routage de l'algorithme CBL occupe donc moins de bande passante qu'OLSR, bande passante libérée qui pourra donc être récupérée au profit du trafic généré par les applications ITS.

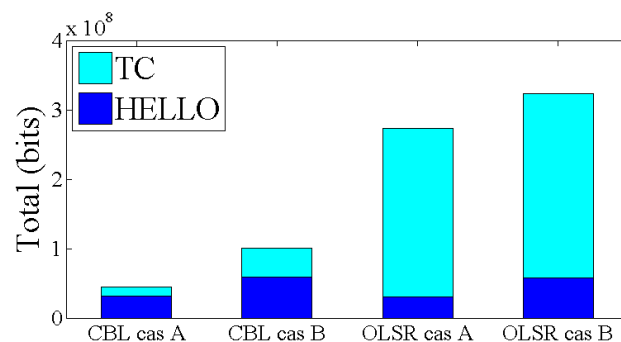


Figure 7. Nombre total de bits transmis pour les messages de trafic de routage (TC et HELLO) dans l'intervalle de temps [150 s, 500 s]

5. Conclusion

Nous avons présenté l'algorithme CBL. CBL est un algorithme de groupement (ou *clustering*) pro-actif distribué qui organise les communications V2V autour d'une infrastructure virtuelle qu'il construit et en définissant des régions d'intérêt. Via la définition de nœuds au rôle distinct, « branche » et « feuille », l'algorithme offrira des services de communication entre véhicules proches et entre véhicules lointains aux applications.

Nous avons choisi de mettre en œuvre CBL dans un protocole pro-actif (ici OLSR). Le choix d'un protocole pro-actif, plutôt que réactif (comme par exemple AODV), permet de s'affranchir de l'émission apériodique de messages de recherche de route préalablement à la transmission de messages applicatifs en mode point-à-point (unicast).

Nous avons proposé une évaluation du protocole CBL en comparaison avec OLSR dans un cas réel de trafic routier composé d'une section de l'autoroute A27 et de la départementale perpendiculaire D90. Les résultats de simulations sous OPNET montrent l'intérêt de la proposition de *clustering* entre véhicules. En limitant le trafic de routage, il libèrera de la bande passante au profit du trafic généré par les applications ITS. Le trafic de routage a été globalement divisé entre 3 et 6 en fonction des cas étudiés. Les travaux futurs se concentreront sur l'évaluation d'applications ITS utilisant CBL dans les VANETs.

Références

- [1] Lucas Rivoirard, Martine Wahl, Patrick Sondi, Marion Berbineau, Dominique Gruyer (2018). « Chain-Branch-Leaf: a Clustering Scheme for Vehicular Networks Using Only V2V Communications », *Ad Hoc Networks*, vol. 68, pp. 70–84, Janvier 2018, disponible depuis le 10 octobre 2017 : <https://doi.org/10.1016/j.adhoc.2017.10.007>.
- [2] T. Clausen et al. (2003). « Optimized link state routing protocol (OLSR) », IETF, no. RFC 3626, 2003.
- [3] S. R. Santana, J. J. Sanchez-Medina, and E. Rubio-Royo (2015). « How to Simulate Traffic with SUMO », in *Computer Aided Systems Theory*, EUROCAST'2015. Springer International Publishing, 2015, vol. 9520, pp. 773–778.

Brève biographie des auteurs :



Lucas Rivoirard a reçu le diplôme d'ingénieur en génie des transports et de la mobilité de l'Ecole Nationale des Travaux Publics de l'Etat et un Master à l'INSA de Lyon. Il a rejoint IFSTTAR en 2015 où il prépare sa thèse de doctorat. Ses domaines de recherche comprennent l'ingénierie des protocoles et la qualité du service dans les communications sans fil pour les systèmes d'assistance à la conduite et les véhicules coopératifs.



Dr. Martine Wahl a reçu le diplôme d'ingénieur en électronique de Polytech Paris-Sud (anciennement FIUPSO) en 1991 et le doctorat de l'Institut polytechnique de Grenoble en 1997. Elle a rejoint l'IFSTTAR en tant que chercheur en 1998. Son domaine de recherche concerne les protocoles de communications de systèmes de communication filaires (de type réseaux de terrains ou embarqués) et sans fil, en particulier pour les transports terrestres.



Dr Patrick Sondi a reçu son doctorat en informatique en 2010 à l'Université de Valenciennes. Il a rejoint l'Université de Littoral et Côte d'Opale en tant que Professeur Associé en 2013. Ses domaines de recherche comprennent l'ingénierie des protocoles, la qualité du service, la sécurité et les simulations de réseaux sans fil, en particulier leur application dans les systèmes de transport.



Dr. Marion Berbineau a reçu le diplôme d'ingénieur en Électronique, automatique et métrologie de Polytech'Lille (France) et le doctorat en électronique de l'Université de Lille respectivement en 1986 et 1989. Elle a rejoint l'IFSTTAR en tant que Chercheur en télécommunications en 1989. Son domaine d'expertise concerne les systèmes de communication sans fil pour les STI et pour les applications ferroviaires y compris le contrôle-commande.



Le docteur Dominique Gruyer a reçu les diplômes de master et de doctorat respectivement en 1995 et 1999 à l'Université de Technologie de Compiègne. Depuis 2001, il est chercheur IFSTTAR, dans l'équipe de perception du laboratoire LIVIC (Département COSYS) et il travaille sur l'étude et le développement de la fusion de données multi-capteurs.