Fair Auto-Adaptive Clustering for Hybrid Vehicular Networks

Julian Garbiso

2017-ENST-0061

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité Informatique et Réseaux

présentée et soutenue publiquement par

Julian Pedro GARBISO

le 30 novembre 2017

Fair Auto-Adaptive Clustering for Hybrid Vehicular Networks

Directeurs de thèse: **Marceau COUPECHOUX**, **Ada DIACONESCU**

Jury

M. Christian BECKER, Professeur, Université de Mannheim, Allemagne	Rapporteur
M. Ken CHEN, Professeur, Université Paris 13, France	Rapporteur
M. Bertrand LEROY, Responsable d'équipe de recherche, Institut Vedecom, France	Examinateur
M. Samir TOHMÉ, Professeur, Université de Versailles Saint-Quentin, France	Examinateur
M. Jeremy PITT, Professeur, Imperial College London, Royaume-Uni	Invité
M. Marceau COUPECHOUX, Maître de Conférences, Télécom ParisTech, France	Directeur de thèse
Mlle Ada DIACONESCU, Maître de Conférences, Télécom ParisTech, France	Directrice de thèse

T H È S E

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

PhD ParisTech

A dissertation presented

in fulfillment of the requirements for the degree of Doctor of Philosophy of

TELECOM ParisTech

Speciality Computer Networking and Telecommunications

presented and defended in public by

Julian Pedro GARBISO

On November 30th, 2017

Fair Auto-Adaptive Clustering for Hybrid Vehicular Networks

PhD Directors: **Marceau COUPECHOUX**, **Ada DIACONESCU**

To my mother.
In memory of my father.

Abstract

For the development of innovative Intelligent Transportation Systems applications, connected vehicles will frequently need to upload and download position-based information to and from servers. These vehicles will be equipped with different Radio Access Technologies (RAT), like cellular and vehicle-to-vehicle (V2V) technologies such as LTE and IEEE 802.11p respectively. Cellular networks can provide internet access almost anywhere, with QoS guarantees. However, accessing these networks has an economic cost. In this thesis, a multi-hop clustering algorithm is proposed in the aim of reducing the cellular access costs by aggregating information and off-loading data in the V2V network, using the Cluster Head as a single gateway to the cellular network. For the example application of uploading aggregated Floating Car Data, simulation results show that this approach reduce cellular data consumption by more than 80% by reducing the typical redundancy of position-based data in a vehicular network. There is a threefold contribution: First, an approach that delegates the Cluster Head selection to the cellular base station in order to maximize the cluster size, thus maximizing aggregation. Secondly, a self-adaptation algorithm that dynamically changes the maximum number of hops, addressing the trade-off between cellular access reduction and V2V packet loss. Finally, the incorporation of a theory of distributive justice, for improving fairness over time regarding the distribution of the cost in which Cluster Heads have to incur, thus improving the proposal's social acceptability. The proposed algorithms were tested via simulation, and the results show a significant reduction in cellular network usage, a successful adaptation of the number of hops to changes in the vehicular traffic density, and an improvement in fairness metrics, without affecting network performance.

Résumé

Dans le cadre du développement des innovations dans les Systèmes de Transport Intelligents, les véhicules connectés devront être capables de télécharger des informations basées sur la position sur et depuis des serveurs distants. Ces véhicules seront équipés avec des différentes technologies d'accès radio, telles que les réseaux cellulaires ou les réseaux véhicule-à-véhicule (V2V) comme IEEE 802.11p. Les réseaux cellulaires, avec une couverture presque omniprésente, fournissent un accès à internet avec garanties de qualité de service. Cependant, l'accès à ces réseaux est payant. Dans cette thèse, un algorithme de clustering multi-saut est proposé avec pour objectif de réduire le coût d'accès au réseau cellulaire en agrégeant des données sur le réseau V2V. Pour faire ceci, le leader du cluster (CH, de l'anglais Cluster Head) est utilisé comme passerelle unique vers le réseau cellulaire. Pour le test d'une application d'exemple pour télécharger du Floating Car Data agrégé, les résultats des simulations montrent que cette approche réduit l'utilisation du réseau cellulaire de plus de 80%, en s'attaquant à la redondance typique des données basées sur la position dans les réseaux véhiculaires. Il y a une contribution en trois parties : Premièrement, une approche pour déléguer la sélection du CH à la station de base du réseau cellulaire afin de maximiser la taille des clusters, et par conséquent le taux de compression. Deuxièmement, un algorithme auto-adaptatif qui change dynamiquement le nombre maximum de sauts afin de maintenir un équilibre entre la réduction des coûts d'accès au réseau cellulaire et le taux de perte de paquets dans le réseau V2V. Finalement, l'incorporation d'une théorie de la justice distributive, afin d'améliorer l'équité sur la durée concernant la distribution des coûts auxquels les CH doivent faire face, améliorant ainsi l'acceptabilité sociale de la proposition. Les algorithmes proposés ont été testés via simulation, et les résultats montrent une réduction significative dans l'utilisation du réseau cellulaire, une adaptation réussie du nombre de sauts aux changements de la densité du trafic véhiculaire, et une amélioration dans les métriques d'équité, sans affecter la performance des réseaux.

Acknowledgements

I would like to thank both my supervisors, Ada Diaconescu and Marceau Coupechoux, for having embarked on this rather unusual adventure with me, and for having kindly helped me all along this process. The hand they gave me, and the way they guided me, were fundamental for the completion of this work. I will not forget that their guidance was academically brilliant but also compassionate during difficult times.

I would also like to express my gratitude to *Institut Vedecom* and in particular Bertrand Leroy for their trust and support, and for the opportunity to keep working together.

I am also sincerely grateful to the people I have met in *Telecom ParisTech* during these years, who have made them very enjoyable and have enriched me as a person. I will always remember the time spent together and I hope our paths will cross again one day.

Last, but not least, I would like to thank my friends and family, in Argentina and across the globe, for giving me the strength to accomplish each and every objective with their affection and kind words.

Résumé en français

1 Introduction

Nous assistons peut-être au début de la plus grande évolution de la mobilité humaine en un siècle. Si toutes les transformations prévues aujourd'hui sont confirmées, nos systèmes de transport seront plus intégrés, plus intelligents et optimisés, et nos véhicules seront autonomes, connectés et décarbonnés. L'expérience de conduire changera radicalement, et même l'existence du rôle du *conducteur* humain sera mise en question.

Sans doute, pour que ces *Systèmes de Transport Intelligents* (ITS) puissent exister, les véhicules et l'infrastructure ont besoin d'être connectés. Comme un premier pas, l'Union Européenne a approuvé une législation [2] qui oblige les constructeurs automobiles à équiper[1] toutes les nouvelles unités avec une connexion au réseau cellulaire, pour implémenter le système *eCall*. En cas d'accident, eCall appellera automatiquement les services de secours. Le volume de trafic réseau généré par ce système est négligeable, et donc il a été accordé qu'il sera fourni sans charges. Pourtant, avec l'introduction des véhicules autonomes, les contraintes technologiques vont complètement changer. La conduite autonome demandera une connectivité constante et critique du point de vue de la sécurité. Une grande fiabilité et une grande réduction des latences seront demandées. Avec les véhicules autonomes, toute personne dans le véhicule pourra accéder à du contenu multimédia sans se soucier de la navigation et de la conduite. La demande de bande passante augmentera en parallèle des strictes contraintes de qualité de service pour les communications de sécurité pour la conduite autonome, et tout ceci pour des nombreuses connections simultanées. Ces conditions ne peuvent pas être facilement prises en charge par les technologies disponibles de nos jours de façon individuelle.

Plusieurs Technologies d'Accès Radio (RAT, pour le sigle en anglais) qui existent aujourd'hui peuvent être utilisées pour connecter des véhicules. Tous deux, l'IEEE et l'Institut Européen des Normes de Télécommunications (ETSI) ont développé leurs propres approches pour la communication véhiculaire, les deux s'appuyant principalement sur le protocole d'accès sans fil IEEE 802.11p pour la communication véhicule-à-véhicule (V2V) et véhicule-à-infrastructure (V2I). Plusieurs travaux de recherche ont aussi été menés pour implémenter la connectivité véhiculaire à travers le réseau cellulaire, et dernièrement nous avons vu les premières voitures équipées avec des modules de communication cellulaire arriver sur le marché. Les technologies V2V et cellulaire ont des avantages et limitations différentes et parfois complémentaires.

Pourtant, jusqu'à présent, il n'y a pas une vision consensuelle sur comment intégrer les multiples RATs afin d'utiliser les ressources des réseaux de façon intelligente pour compenser les désavantages d'une technologie avec les capacités complémentaires de l'autre, afin d'avoir une seule approche qui

[1] À partir d'avril 2018

repousse les limites qu'on trouve aujourd'hui. Nous sommes en train de créer le monde du tout-connecté, mais différents types de dispositifs utilisent souvent différentes technologies, une RAT particulière souvent spécifique à un certain type d'application, et dont l'interconnexion avec d'autres RATs n'est pas automatisée. Ceci pourrait commencer à changer avec l'arrivée de la 5G, la cinquième génération de communications mobiles.

La nouveauté des réseaux mobiles 5G est l'intégration intelligente de différentes RATs afin d'avoir une seule approche cohérente pour les communications, qui pourrait effacer les contraintes technologiques actuelles spécifiques à chaque protocole ou standard (couverture, bande passante, latence, etc.), tout en proposant une capacité de réseau supérieure, capable de connecter sans incident tous les dispositifs et objets.

En regardant les technologies actuelles, on se rend compte que les principales alternatives pour les réseaux véhiculaires sont très dissemblables : d'un côté, nous avons le principal protocole V2V, **IEEE 802.11p** —un protocole qui fonctionne dans une bande *sans licence* du spectre radio, et qui est utilisé pour créer des connexions *ad-hoc* entre les véhicules aussi vite que possible. Ce protocole *n'a pas besoin d'un serveur central*, et donc *il n'y a pas de garantie de connexion à internet*. Il a été conçu pour des communications *en local*, et il est affecté considérablement par la *saturation et la perte de paquets*. De l'autre côté, nous avons les réseaux cellulaires, dont la technologie la plus avancée à cette date est la **LTE (4G)**. Cette connexion est *centralisée* et *dépend de la présence de stations de base* à proximité. La couverture est très répandue sur tout le territoire, mais certainement pas à cent pour cent. Néanmoins, quand l'utilisateur est dans la zone de couverture, il a une *connexion permanente à internet*. Ce réseau est très fiable en termes de volume de trafic supporté et bande passante, mais évidemment, *l'accès est payant*.

Heureusement, ces technologies *dissemblables* sont *complémentaires*, et nous pouvons utiliser les deux. Il s'agit du concept d'un *réseau véhiculaire hybride*. Dans ce contexte, la motivation de notre travail est d'étudier la possibilité d'intégrer l'accès aux réseaux V2V et cellulaire dans un réseau véhiculaire hybride. L'objectif principal est de réduire les coûts de communication des véhicules, et en même temps assurer l'accès à l'information en ligne (par exemple les cartes locales, conditions climatiques, état du trafic) et le téléchargement des données.

Nous proposons une solution qui combine du *clustering* multi-saut adaptatif – pour agréger des données et réduire le coût de communication – avec une théorie de la justice distributive – pour distribuer de façon équitable les coûts entre tous les véhicules à long terme.

1.1 Solution proposée et contributions principales

OBJECTIFS :

- Fournir une solution pour l'intégration des réseaux V2V et cellulaire dans un environnement véhiculaire afin de réduire les coûts d'accès au réseau cellulaire, tout en préservant la performance dans les deux réseaux.
- Faire en sorte que la solution proposée soit équitable dans le temps, pour la rendre socialement acceptable.

Pour capitaliser les avantages de chaque technologie, nous proposons de permettre aux véhicules de s'auto-organiser en *clusters* (groupes) à travers la communication V2V. Un cluster est composé d'un *Leader du Cluster* (CH, pour le sigle en anglais) et plusieurs *Membres du Cluster (CM)*. Au sein de chaque cluster, le flux de données entre les véhicules et le CH passe par le réseau V2V, et les données échangés entre le CH et le serveur destinataire sur internet passent par le réseau cellulaire (dans le cas du téléchargement de données dans un serveur, les données envoyées par les véhicules seront agrégées par le CH). Dans un cluster *multi-saut*, les CMs peuvent communiquer avec le CH en passant par d'autres CMs qui fonctionnent comme passerelles vers le CH lorsqu'il est hors de portée.

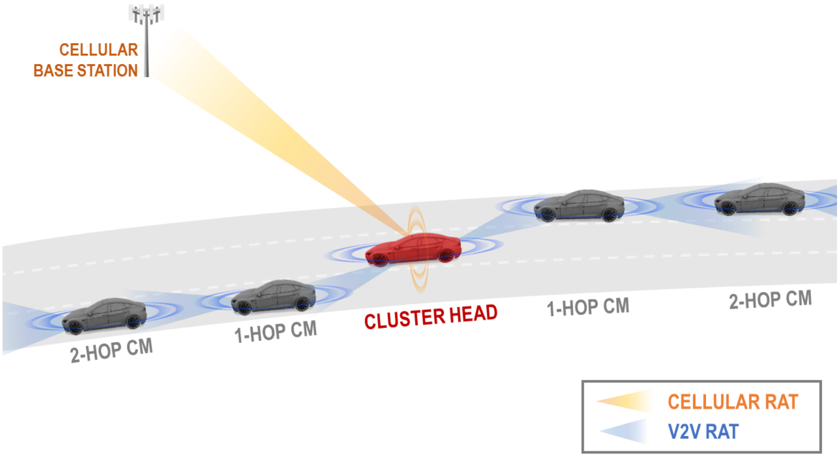


FIGURE 1 – Illustration d'un cluster multi-saut dans un réseau véhiculaire hybride.

Dans notre modèle, on considère un scénario où les véhicules doivent envoyer des données périodiquement à travers le réseau cellulaire (par exemple, du *Floating Car Data*) pour alimenter les services de gestion de trafic. Une réduction du trafic montant dans le réseau cellulaire peut être atteint **en agrégeant et en comprimant cette information dans le CH, qui est le seul nœud à utiliser la connexion cellulaire**. Les CH peuvent aussi diffuser localement l'information reçue dans la voie descendante du réseau cellulaire, sur le réseau V2V (des informations telles que les données demandées à un Système d'Information Géographique).

Cependant, un compromis est nécessaire au moment d'établir la taille du cluster en termes de nombre de sauts. D'un côté, plus grand est le cluster, plus important est le taux de compression atteint par le CH et plus petite est la quantité de ressources à utiliser dans le réseau cellulaire. Mais de l'autre côté, le taux de perte de paquets (PLR) devient plus important pour les grands clusters, à cause du plus grand nombre de collisions dans les canaux V2V. La taille d'un cluster – c'est à dire, le nombre de véhicules dans le cluster – dépend directement de la densité du trafic et du nombre de sauts. Alors, on peut ajuster la taille des clusters en adaptant dynamiquement le nombre de sauts en fonction de la densité du trafic.

Nous proposons un *Algorithme de Clustering Auto-adaptatif* et multi-saut pour réseaux véhiculaires hybrides qui change localement et dynamiquement le nombre maximum de sauts du cluster (et donc augmente ou diminue sa taille) afin de maximiser la compression du volume de données échangées avec l'infrastructure cellulaire, tout en gardant le taux de perte de paquets dans le réseau V2V au dessous d'un seuil de tolérance.

L'algorithme sélectionne le Leader du Cluster et décide l'adaptation du nombre de sauts *au niveau de la station de base du réseau cellulaire*. Pour une distribution géographique améliorée des CH, ce qui permet aux clusters d'augmenter sa quantité de membres, la sélection du CH est déléguée à la station de base. À l'aide d'un *framework* de simulation d'ITS, on montre que l'Algorithme Auto-adaptatif s'adapte correctement aux changements extrêmes de densité du trafic, en reduisant significativement l'usage du réseau cellulaire (ce qui produit d'importantes économies à grande échelle), tout en respectant les contraintes imposées sur le taux de perte de paquets dans le réseau V2V (afin d'être conforme aux paramètres de latence et fiabilité requis par différentes applications et services).

La méthode proposée améliore la performance réseau, mais ouvre un nouveau problème : la distribution des coûts de communications encourus par le CH. Dans les algorithmes de clustering où le CH fonctionne comme une passerelle vers l'internet, être élu CH est une charge significative. Ceci est dû au fait que chaque véhicule est lié à un compte spécifique d'un individu auprès d'un opérateur de téléphonie mobile, avec un quota de trafic limité. Alors, le véhicule qui est élu CH est le seul qui consomme son quota de trafic pour le bénéfice de tous les autres véhicules du cluster. Il devient évident que des améliorations sont nécessaires pour assurer que, même si à un moment donné il n'y aura qu'un CH qui sera le seul à consommer son quota, *il y aura* une consommation *équitable* du quota de tous les véhicules *sur la durée*. La consommation du quota cellulaire est visible pour les conducteurs. S'ils perçoivent que leur quota est consommé de façon inéquitable, ils peuvent, à tout moment, abandonner le système.

Pour éviter ce problème, **nous proposons une approche qui adopte une *théorie de la justice distributive* et l'applique à la sélection du CH dans les algorithmes de clustering pour une distribution *équitable* des rôles et de la consommation du quota cellulaire sur la durée.** En particulier, nous appliquons cette approche sur l'Algorithme de Clustering Auto-adaptatif proposé. En même temps, nous croyons que notre approche est également applicable à d'autres algorithmes de clustering.

2 État de l'art du clustering pour réseaux véhiculaires hybrides

La vaste majorité de la littérature dans ce sujet a été conçue en partant du principe que seulement la communication V2V était disponible. Les hypothèses initiales impliquaient le déploiement de nombreuses Unités de Bord de Route (UBR, ou RSU pour le sigle en anglais), une supposition qui a été laissé de côté dû au coût d'un tel déploiement. Les limitations d'IEEE 802.11p et le besoin d'une connexion à des serveurs distants afin de déployer des services innovants a obligé les constructeurs automobiles à inclure la connectivité cellulaire dans leurs nouveaux travaux de développement. Peu de travaux de recherche sur les techniques de clustering travaillent sur cette hypothèse. Ces travaux sont brièvement discutés dans les paragraphes suivants.

Dans les réseaux véhiculaires hybrides, nous avons au moins deux technologies de communication différentes : normalement, un protocole de communication V2V et une connexion au réseau cellulaire. La plupart des travaux de recherche dans le sujet se concentre sur les algorithmes de sélection de passerelle (une passerelle serait dans ce contexte un nœud qui fonctionne comme une liaison entre deux réseaux). Dans [28], Benslimane et al. proposent une méthode de clustering avec gestion de passerelles entre IEEE 802.11p et UMTS (3G). Ils se focalisent sur la sélection des meilleurs candidats à devenir un nœud-passerelle en fonction de la force du signal reçu (RSS) d'UMTS. Un travail similaire est fait par Zhioua et al. dans [36], où ils utilisent la *fuzzy logic* pour sélectionner le meilleur nœud-passerelle entre un réseau véhiculaire ad-hoc (VANET) clusterisé et LTE, avec la nouveauté de considérer le type de trafic comme un ordre de priorité afin d'incorporer des contraintes de QoS (qualité de service) dans la sélection de la passerelle.

Une autre direction de recherche se concentre sur le problème de formation des clusters. Par exemple, [72] propose une architecture intéressante pour garantir l'arrivée des Messages de Prévention Coopérative (CAM, pour ses sigles en anglais), dans le scénario spécifique des intersections où les véhicules s'approchent tout en étant cachés les un des autres à cause des murs des bâtiments, qui parfois bloquent même les signaux radio. Ils créent une *région de cluster* autour de l'intersection (ces régions sont fixes). Chaque message CAM est transmis via LTE. Une fois qu'un véhicule entre dans une région de cluster, il commence à diffuser des paquets périodiques (*beacons*, spécifiques à l'algorithme de clustering, et pas des messages CAM) à travers l'interface WiFi pour former un cluster (les auteurs ayant choisi IEEE 802.11b au lieu d'IEEE 802.11p à cause du faible coût des dispositifs

WiFi, mais ils estiment que pour les objectifs de l'expérimentation les deux technologies sont interchangeables). Une fois le cluster formé, on lui attribue un canal WiFi spécifique, et les CHs sont la seule entité autorisée à envoyer des messages CAM à travers LTE pour faire en sorte que les véhicules dans les routes adjacentes soient prévenus de la présence d'autres véhicules à proximité de l'intersection devant. Ceci réduit considérablement la quantité de messages CAM transmis dans les hypothèses mentionnées, et semble être une solution correcte pour le problème des intersections. Pourtant, le scénario d'application est très limité dû à la nature fixe des *régions de cluster* dans une surface plutôt petite. Il y a d'autres propositions pour la formation de clusters conçues avec des objectifs différents ou pour scénarios spécifiques comme celui de l'exemple dernier, voir par exemple [67, 54].

Le travail de Rémy et al. dans [64] est la référence la plus proche de notre travail. Les auteurs présentent une architecture appelée LTE4V2X, une approche qui centralise le processus de formation du cluster, et qui le délègue à un eNodeB (la station de base d'un réseau cellulaire LTE), afin de l'accélérer et réduire la congestion dans le spectre radio d'IEEE 802.11p. En outre, ils implémentent un TDMA (accès multiple à répartition dans le temps) interne dans chaque cluster afin d'envoyer l'information de prévention coopérative (position, vitesse et direction), ce qui assure d'éviter les collisions de paquets. Cependant, dans les résultats montrés par les auteurs, on voit que même si la congestion sur IEEE 802.11p est considérablement réduite, le trafic dans le réseau LTE augmente énormément. Nous voyons que LTE est traité comme une ressource abondante, et le facteur économique est ignoré a priori.

À notre connaissance, il n'y a pas eu des travaux de recherche concernant l'analyse du problème d'ajuster la taille des clusters afin de trouver un équilibre entre la perte de paquets dans le réseau IEEE 802.11p et les coûts d'accès cellulaire. Dans le chapitre 5, nous essayons de faire une analyse claire de ce conflit que nous considérons critique parce qu'il compromet, d'un côté, le fonctionnement correct du système, et d'autre côté, sa faisabilité économique.

3 Algorithme de Clustering de Base

3.1 Motivation, objectifs et contribution

Les systèmes de gestion du trafic visent à rendre nos voyages quotidiens plus sûrs et plus efficaces. Ils peuvent le faire en influençant le comportement des conducteurs de différentes manières : régulation des cycles de feux, suggestion de déviations, utilisation de panneaux de messages dynamiques pour les recommandations, etc. Ces systèmes ont besoin d'un maximum d'informations sur l'état du trafic, en temps réel. La collecte de **Floating Car Data** (*données de véhicule flottant*, ou FCD) — fondamentalement la position, la vitesse, la direction et l'heure — de chaque véhicule sur la route est une fonctionnalité très souhaitable pour les systèmes de transport intelligents. Pour ce faire efficacement, nous proposons d'utiliser le **clustering** multi-saut comme moyen pour **agréger des informations**, en profitant des redondances significatives dans les informations locales qui ont tendance à se produire dans les réseaux véhiculaires.

Cependant, il y a un compromis entre l'agrégation de l'information (améliorée en augmentant le nombre de sauts, ce qui élargit les clusters) et le taux de perte de paquets dans le réseau V2V qui doit être traité.

Les algorithmes de clustering proposés sont un outil d'agrégation d'informations (de V2V à cellulaire) et de déchargement (de cellulaire à V2V), utilisant le CH comme passerelle. Afin de tester nos algorithmes de clustering (l'*Algorithme de Clustering de Base* dans le chapitre 5, l'*Algorithme de Clustering Auto-adaptatif* dans le chapitre 6 et l'*Algorithme de Clustering Équitable et Auto-adaptatif* dans le chapitre 7), nous avons toujours utilisé le même exemple d'application : l'agrégation de FCD.

Cette application est destinée à créer une représentation virtuelle et en temps réel du trafic routier dans un serveur centralisé de gestion du trafic. Le fait de disposer de ces données en temps réel permet une gestion optimale du trafic à toutes les échelles et peut fournir une entrée de données parfaite pour les algorithmes d'apprentissage automatique afin de faire des prédictions optimales et ainsi permettre une meilleure organisation. La FCD peut également être utilisée pour obtenir des statistiques détaillées sur la mobilité qui peuvent être très utiles pour l'analyse économique métropolitaine ou régionale, la planification de l'infrastructure et du transport public, et les investissements.

3.2 Formulation du problème

La mise en œuvre de l'application d'agrégation de FCD repose sur l'idée que chaque véhicule doit constamment communiquer son identifiant, sa position, sa vitesse et sa direction aux serveurs de gestion du trafic, à travers le réseau cellulaire. Si chaque véhicule transmet ses données individuellement, cela signifiera un nombre très important de connexions au réseau cellulaire. La création de groupes de véhicules pour agréger cette information réduirait le nombre de connexions et permettrait la compression de la quantité de données transmises.

Dans notre proposition, lorsqu'un véhicule rejoint un cluster, il cesse de transmettre sa FCD individuellement. Nous profitons de l'existence (telle qu'établie dans les normes ETSI) du service de Prévention Coopérative (*Cooperative Awareness*, ou CA). Toutes les données que chaque véhicule est censé envoyer aux serveurs via le réseau cellulaire dans notre exemple d'application, sont présentes dans les messages CAM (*Cooperative Awareness Messages*) qui sont obligatoires et périodiquement envoyés via la connexion V2V. Le CH écoute tous les CAM provenant de ses membres de cluster et regroupe ces informations en prenant par exemple la valeur moyenne (d'autres méthodes peuvent être utilisées en fonction des besoins de chaque application). Lorsqu'un véhicule rejoint un groupe, le CH envoie une notification unique au réseau cellulaire pour informer que ce véhicule particulier est présent dans son groupe à partir de ce moment. Le serveur garde la trace des véhicules présents dans un cluster sans avoir besoin d'être notifié périodiquement. Une seule notification d'arrivée et une seule notification de départ suffisent. Pour le reste du temps, le CH agrège simplement l'information de tous ses véhicules en prenant la valeur moyenne des métriques de vitesse et de position. Ainsi, pour chaque cluster, le serveur de gestion du trafic reçoit en permanence la position moyenne et la vitesse des véhicules du cluster, permettant une réduction du nombre de connexions et une compression de la transmission des données via le réseau cellulaire à un ratio de N à 1, où N est le nombre de membres du cluster.

Nous considérons un réseau véhiculaire composé de véhicules qui circulent sur une section de route de longueur fixe L_s et constituée de L voies. Sur chaque voie, les véhicules arrivent à des instants périodiques, toutes les T secondes, au début de la section. Les véhicules circulent à vitesse constante jusqu'à la fin de la section où ils quittent le réseau. Nous supposons que toute la section d'autoroute est couverte par une seule station de base (BS) cellulaire vers laquelle les véhicules doivent envoyer des informations à un taux de λ paquets/s.

En supposant qu'un algorithme de clustering est implémenté, ce trafic peut être directement transmis à la station de base ou acheminé par un canal de communication. Lorsqu'un véhicule appartient à un cluster de taille 1 (c'est-à-dire qu'il est isolé), il transmet ses informations à la station de base en utilisant des ressources radio cellulaires montantes. Lorsqu'il est inclus dans un cluster c de taille $N_c > 1$, ce trafic est envoyé au CH en utilisant le protocole IEEE 802.11p et le CH agrège les informations provenant des CM et envoie le résultat à la BS.

Le trafic généré par le cluster c pour la BS de destination est alors $\eta(N_c)N_c\lambda$, où $\eta(N_c) \leq 1$ est une fonction de compression effectuée par le CH qui peut être décroissante en fonction de N_c. Sans perte de généralité, on peut supposer que $\eta(1) = 1$. Nous définissons une *partition de cluster* comme un ensemble de clusters sans chevauchement qui inclut tous les véhicules du réseau. Dans la suite, nous considérerons seulement les partitions de cluster avec des clusters ayant un maximum de H bonds entre n'importe quel CM et son CH.

En conséquence, le trafic total généré par le réseau véhiculaire dans la voie montante du réseau cellulaire pour une partition de cluster donnée $\mathscr{C}$ peut être écrit comme :

$$\Lambda(\mathscr{C}) = \sum_{c \in \mathscr{C}} \eta(N_c) N_c \lambda, \tag{1}$$

où $\mathscr{C}$ est l'ensemble de tous les clusters, et N_c est le nombre de véhicules dans le cluster c. On note $N = \sum_c N_c$ le nombre total de véhicules dans le réseau. Nous définissons maintenant le taux global de compression de la partition de cluster $\mathscr{C}$ comme :

$$\alpha(\mathscr{C}) \triangleq 1 - \frac{\Lambda(\mathscr{C})}{N\lambda} = 1 - \frac{\sum_{c \in \mathscr{C}} \eta(N_c) N_c}{N} \tag{2}$$

Notez que α est aussi le taux de compression moyen. Pour cette partition de cluster et pour le modèle de trafic ocnsidéré, nous pouvons calculer le taux de perte de paquets $PLR(\mathscr{C}, \lambda)$, qui est une fonction de la partition de clusters et de la densité du trafic.

Notre problème est, pour une condition de trafic donnée λ, de maximiser le taux de compression moyen sous contrainte de maintenir un taux de perte de paquets acceptable :

$$\max_{\mathscr{C}} \alpha(\mathscr{C}) \tag{3}$$

$$\text{s.t. } PLR(\mathscr{C}, \lambda) \leq PLR_{max}, \tag{4}$$

où PLR_{max} est une contrainte spécifique des applications.

3.3 Algorithmes de sélection des leaders de cluster

En étudiant la littérature, nous avons découvert que les algorithmes de clustering peuvent être très différents les uns des autres. L'une des choses qui les différencie est le mécanisme d'élection du leader du cluster. Le premier mécanisme d'élection de CH que nous avons mis en place à des fins de test était basé sur celui présenté dans [73] pour être représentatif d'une tendance vue dans la littérature. Nous l'appellerons l'**Auto-proclamation du Leader du Cluster**, car il est basé sur l'idée qu'un CH se proclame automatiquement quand, après avoir échangé des informations de position et de vitesse avec ses voisins, il *déduit* qu'il a la vitesse relative minimale par rapport à tous les autres.

Deux méthodes différentes pour la sélection du CH sont implémentées et testées :

1. **Auto-proclamation du CH**, où les CH se sélectionnent eux mêmes en utilisant une métrique de vitesse relative minimale ;

2. **Sélection du CH déléguée à la BS**, notre proposition et contribution, où la sélection du CH est déléguée à la Station de Base du réseau cellulaire.

L'objectif est de comparer les performances de notre proposition avec un exemple représentatif d'une stratégie habituelle pour la sélection du CH dans la littérature sur les clusters multi-saut pour les VANETs.

3.3.1 Auto-proclamation du Leader du Cluster

Dans cette première approche nous avons choisi d'implémenter la méthode de sélection de CH proposée dans [73] pour l'algorithme de clustering VMaSC.

Cette méthode utilise l'information collectée pendant une période initiale fixe : chaque véhicule enregistre localement la vitesse et position de ses voisins, ainsi que la distance en sauts à chacun d'entre eux. Si un véhicule détecte qu'il a la plus petite vitesse relative par rapport à l'ensemble des véhicules qu'il peut voir dans son voisinage, il s'auto-proclame CH et commence à diffuser des messages *CH Advertisement*. Tout véhicule souhaitant joindre le cluster doit envoyer un message *Join Request*, et l'incorporation est seulement effective une fois qu'il a reçu un message *Join Response*, soit du CH (s'il est directement connecté), soit de son nœud parent.

3.3.2 Sélection du CH déléguée à la station de base

Motivation

Dans les premières étapes de notre travail, tous les briques pour la construction d'un algorithme de clustering multi-sauts (par exemple l'architecture de télécommunications, la spécification des messages et une machine d'état) ont été créés.

Au cours des premiers tests, nous avons réalisé que les caractéristiques inhérentes à l'algorithme de sélection du CH affectaient significativement, de manière négative, la possibilité d'augmenter la taille des clusters lors de l'incrémentation du nombre maximum de sauts. En observant en détail plusieurs scénarios de simulation, nous avons confirmé que la quantité de CH auto-proclamés était excessive. La densité de CHs dans le trafic et le principe que les véhicules choisiraient de rejoindre un CH qui est à la distance minimale *en terme de sauts* faisaient en sorte que les véhicules trouveraient toujours un CH à un bond à rejoindre. Ainsi, l'augmentation du nombre maximal de sauts dans l'algorithme aurait des conséquences sur le trafic réseau, sans bénéficier d'une augmentation réelle de la taille du cluster.

Cette première conclusion a inspiré l'idée de l'algorithme de sélection de CH proposé, qui **délègue la sélection du CH à la station de base cellulaire**, le reste du processus de formation de clusters **ayant lieu dans le lien V2V**. Cette décision est basée sur l'hypothèse que *les véhicules connectés du futur seront très certainement en communication constante avec des serveurs distants*, via le réseau cellulaire, pour une variété de services basés sur la position pour l'utilisateur, et pour une gestion du trafic plus efficace. Pouvoir compter sur le réseau cellulaire pour *une centralisation locale (dans la zone de couverture d'une station de base) de la sélection du CH permet de créer des clusters de taille optimale, mieux adaptés à l'agrégation d'information*.

Architecture pour la sélection du CH

Dans notre architecture système, chaque véhicule est équipé de **deux interfaces d'accès réseau** : une pour un lien *V2V* et une pour le réseau *cellulaire*. Dans la *zone de couverture* de la **station de base**, chaque route (dans chaque direction séparément) devient une **région de clustering**. Chaque région est divisée en **secteurs de clustering** de la taille approximative d'un cluster (en fonction du nombre maximum de sauts, qui est *statique* dans cet algorithme de base — un paramètre fixe —, et la portée de communication V2V). La taille d'un secteur est en fait calculée comme le produit de la portée de communication V2V multipliée par le nombre maximal de sauts, de sorte que chaque secteur comprend approximativement un seul cluster (voir Figure 2).

La station de base est responsable de la sélection des leaders des clusters. Elle vérifie à intervalles réguliers s'il y a un CH dans chaque secteur (puisque la BS suit en permanence chaque véhicule, qu'il envoie sa position individuellement ou agrégée par le CH). Si ce n'est pas le cas, elle en choisit un selon l'algorithme présenté plus bas. Le processus de formation du cluster a lieu localement, en suivant les procédures et les messages décrits dans le chapitre 5 de la thèse.

La délégation de l'élection du CH à la station de base cellulaire a été décidée après avoir observé qu'un inconvénient majeur de l'algorithme d'*Auto-proclamation du Leader du Cluster* est qu'il génère trop de CHs. L'élection déléguée s'avère utile pour surmonter ce problème. L'idée principale est de diviser la section d'autoroute en segments, dont la longueur dépend de la portée de communication radio IEEE 802.11p et du nombre maximum de sauts, et de choisir comme CH dans chaque segment

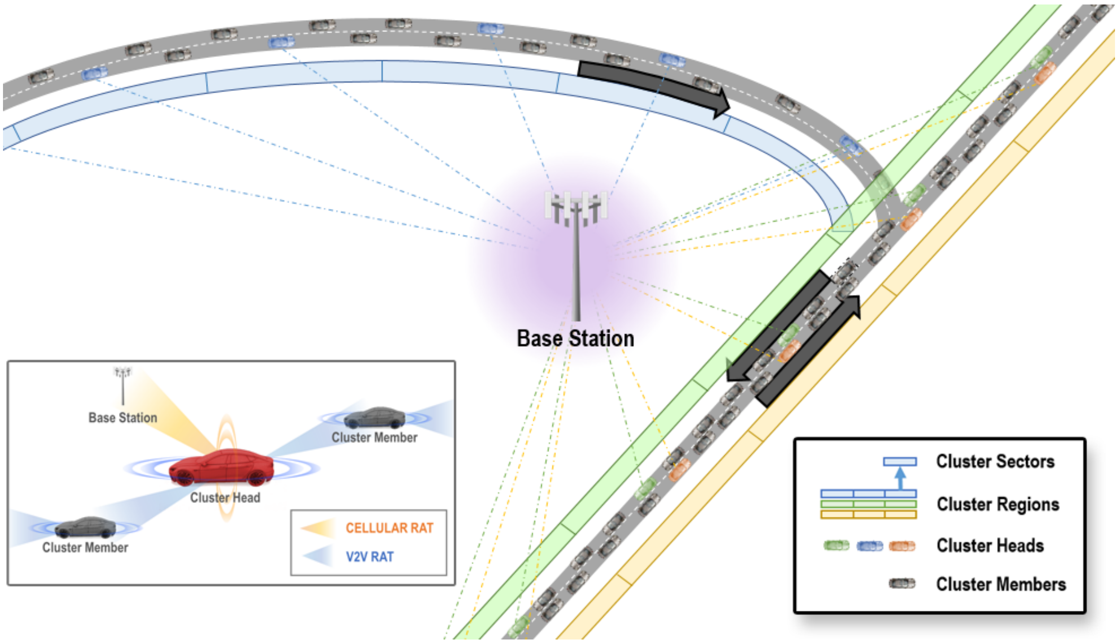


FIGURE 2 – Dans l'architecture proposée, la zone de couverture d'une station de base est divisée en *régions de clustering* (une pour chaque route et direction), et chacune d'entre elles est divisée en *secteurs de clustering*. Notre algorithme de clustering vérifie périodiquement la présence d'un CH dans chaque secteur de clustering, en sélectionnant un nouveau s'il n'y en a pas. La longueur de chaque secteur de clustering est le produit de la portée de communication V2V par le nombre maximum de sauts.

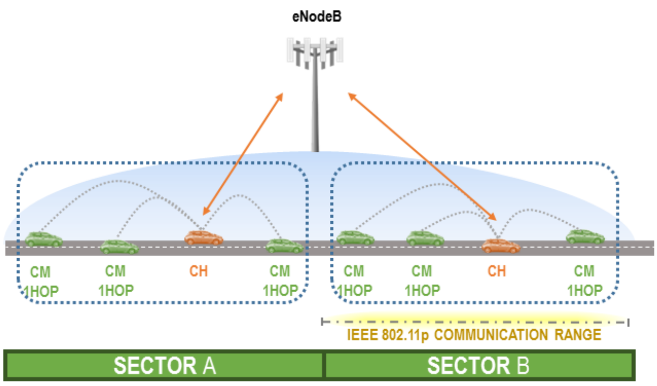


FIGURE 3 – Schéma des clusters, secteurs et station de base.

le véhicule le plus proche du point central du segment. Ce processus est mis à jour tous les T secondes. Cet algorithme peut être étendu à une carte plus complexe en le divisant en sections de route dans la même direction et couvertes par une seule BS, et en appliquant la procédure décrite dans l'algorithme 1.

Cet algorithme ne nécessite pas de couverture cellulaire constante, car en cas de perte de couverture, les CH élus resteront dans cet état, le cluster ne cessera pas de fonctionner, et les véhicules pourront rejoindre, quitter ou changer de cluster. Cependant, il est difficile de prédire comment la

Algorithm 1 Algorithme de sélection du CH déléguée à la BS

Initialisation :
Saisir la période de maintenance T.
Saisir la portée de communication IEEE 802.11p R, le nombre maximum de sauts H et calculer le diamètre de clustering $D = 2R \times H$.
Diviser la section d'autoroute en $S = L_s/D$ segments.
Pour $t = nT$, $n = 1, 2, ...,$ **faire**
    Pour $s = 1, 2, ..., S$, **faire**
        S'il n'y a pas de CH dans s **alors**
        Sélectionner comme CH le véhicule qui est le plus proche
        du centre de s.
        Fin Si
    Fin Pour
Fin Pour

qualité des clusters formées se dégraderait avec le temps après avoir été déconnectée du réseau cellulaire. Une possibilité serait de laisser un cluster passer au mécanisme d'auto-proclamation du CH si la couverture cellulaire est perdue pendant une longue période. Cette question est laissée pour d'autres études.

Le processus de transfert intercellulaire est trivial dans les hypothèses actuelles. L'information de chaque véhicule est disponible à tout moment.

Diagramme d'États du Véhicule

Le diagramme d'états de la figure 4 détaille les états et les transitions d'un véhicule dans notre modèle. Quand un nouveau véhicule arrive, il commence dans l'état de **Découverte** afin d'écouter les meilleures options (avec le plus petit nombre de sauts) pour atteindre un leader de cluster, et commence à construire son *Tableau d'Information du Voisinage* ou **NIT**. Même à la réception d'un message *CH Advertisement*, il n'y a pas de transition directe possible vers l'état de Membre du Cluster.

À l'expiration du temporisateur de découverte prédéfini (dans les simulations, il est fixé à 30 secondes), le véhicule passe à l'état **Isolé**. S'il a reçu *CH Advertisements*, il choisira le CH le plus proche en termes de nombre de sauts. Dans le cas où deux CH ou plus sont détectés à la même distance de sauts, la distance physique exacte sera calculée en utilisant les informations de la NIT, et le CH à la plus courte distance sera choisi. Dans le cas extrêmement improbable où deux CH seraient encore à égalité à ce stade, celui avec l'ID le plus bas est choisi.

Si le véhicule n'a reçu aucun *CH Advertisement*, il attendra dans l'état **Isolé** jusqu'à ce qu'il reçoive un *CH Advertisement* ou une notification pour devenir CH depuis la station de base cellulaire. Un véhicule dans l'état **Isolé** qui n'a pas pu rejoindre un cluster va immédiatement essayer de rejoindre le premier CH dont il reçoit un *CH Advertisement*.

Même si la transition d'**Isolé** vers **Membre du Cluster** est simplifiée dans la figure comme la réception d'un *CH Advertisment*, en pratique la transition se produit lors de l'échange de messages *Join Request* et *Join Response* avec le CH correspondant.

La décision de la sélection d'un CH est complètement déléguée à la BS cellulaire. Un véhicule dans l'état **Isolé** ou **Membre du Cluster** peut être nommé CH par la station de base à tout moment. Immédiatement après la réception de cette notification, le véhicule passe à l'état **Leader du Cluster** et commence à diffuser *CH Advertisements*.

Un véhicule peut passer de l'état **Leader du Cluster** à l'état **Isolé** sous deux conditions différentes :

- Il n'a aucun membre du cluster pendant une certaine période de temps ; ou

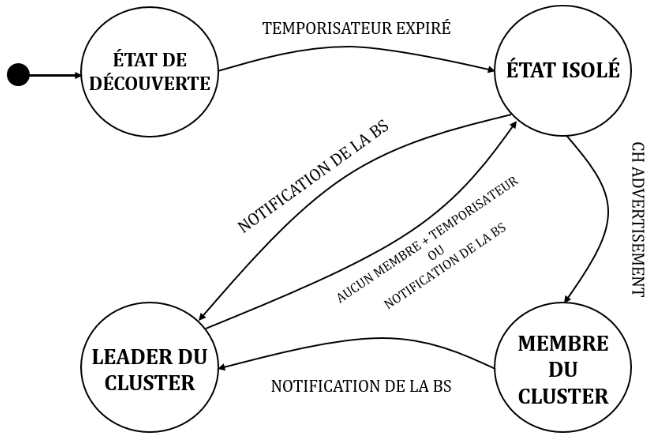


FIGURE 4 – Diagramme d'états d'un véhicule dans notre proposition.

- Il reçoit une notification pour cesser d'être CH depuis la station de base.

La BS peut demander à un véhicule de quitter l'état CH si deux ou plusieurs CH sont présents dans le même secteur de clustering.

3.4 Résultats de la simulation

Le trafic du réseau de base est constitué des CAM obligatoires dont la fréquence à la valeur minimale est de 1 Hz (chaque véhicule émet alors un CAM par seconde). Ces messages sont l'élément de base de la carte dynamique locale (LDM).

Nous supposons un taux d'agrégation $\eta(N_c) = 1/N_c$, où N_c est la taille du cluster. La longueur de la section de la route est de $L_s = 5$ km, et le nombre de voies est de $L = 3$. Nous considérons que, pour que le système fonctionne de manière sûre et fiable, le taux de perte de paquets sur l'interface radio V2V, PLR_{max}, ne peut pas être supérieur à 10%. Un total de $N = 60$ véhicules est simulé dans chaque tour. La vitesse moyenne des véhicules est de 16,6 m/s. Le temps d'inter-arrivée des véhicules varie entre 1 s et 20 s. Cela revient à dire que nous analyserons les variations de nos métriques en fonction de la densité du trafic véhiculaire, car l'augmentation du temps d'inter-arrivée des véhicules implique une réduction de la densité du trafic véhiculaire et inversement. Le lecteur devrait garder à l'esprit cette relation inversement proportionnelle.

Le nombre maximal de sauts est fixé à $H = 1$, 2 et 3 dans des simulations séparées, pour chacun des deux algorithmes de clustering (auto-proclamation du CH et sélection du CH déléguée à la BS). Nous définissons T comme la durée moyenne d'un véhicule pour parcourir le diamètre estimé d'un cluster, D.

3.4.1 Nombre maximum de sauts en fonction du taux de perte de paquets

Dans la figure 5, nous montrons le taux de perte de paquets en fonction du temps d'inter-arrivée pour différents nombres maximum de sauts et pour l'algorithme d'auto-proclamation des CH. Les tendances observées sont similaires pour l'algorithme de sélection du CH déléguée à la BS. Nous pouvons clairement voir que la quantité de paquets perdus augmente radicalement pour des densités élevées de véhicules, et la situation devient bien pire pour chaque saut supplémentaire que nous permettons. La raison réside dans l'effet de tempête de diffusion (*broadcast storm*) qui se produit

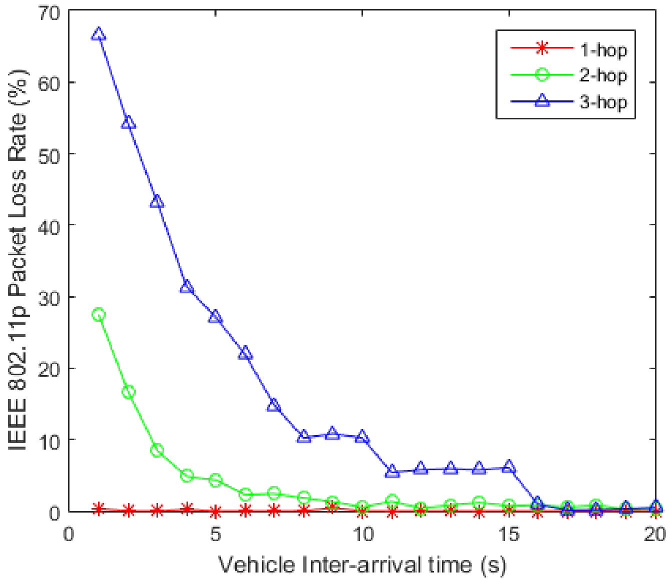


FIGURE 5 – Taux de perte de paquets sur IEEE 802.11p en fonction du temps des inter-arrivées entre les véhicules, pour différents nombres de sauts maximum, et pour l'algorithme d'auto-proclamation des CH.

dans les réseaux à plusieurs bonds très denses lorsque quelques paquets sont diffusés. Cet effet est amplifié lorsque ces paquets sont rediffusés sur un nombre croissant de sauts.

Sur cette figure, nous pouvons voir qu'il existe des densités de véhicules pour lesquelles le clustering à 2 ou 3 sauts est incompatible avec les exigences : au dessous de 10 arrivées par seconde (respectivement 3 arrivées par seconde), l'algorithme d'auto-proclamation du CH à 3 bonds (respectivement 2 bonds) ne respecte pas la contrainte de taux de perte de paquets maximal de 10%.

3.4.2 Nombre maximum de sauts en fonction de la taille du cluster

Les figures 6 et 7 montrent des *box plots* des tailles des clusters obtenus, pour différentes densités de véhicules et nombre de sauts, avec l'algorithme d'auto-proclamation des CH et l'algorithme de sélection du CH déléguée à la BS respectivement.

Comme prévu, on peut voir qu'à des temps d'inter-arrivée intermédiaires et élevés (c'est-à-dire à des densités de véhicule faibles à intermédiaires), lorsque le nombre de sauts augmente, la taille moyenne des clusters augmente également. Lorsque le nombre de sauts est élevé (surtout dans le cas des clusters à 3 bonds), l'augmentation de la densité des véhicules a un effet contradictoire. Une forte densité de véhicules conduit en effet à des pertes de paquets élevées et les CH ne peuvent pas communiquer correctement avec les CM potentiels. Cela implique une diminution de la taille moyenne des clusters. Lorsque le nombre de sauts est faible, l'augmentation de la densité du trafic véhiculaire augmente également la taille moyenne des clusters.

Notre algorithme de sélection des CH déléguée à la BS s'avère plus performant en termes de taille de cluster, montrant un effet direct non ambigu entre l'augmentation du nombre de sauts et l'augmentation de la taille des clusters. Dans le cas de l'algorithme d'auto-proclamation des CHs, étant donné que trop de CHs sont proclamés, l'augmentation du nombre de bonds conduit à peu ou pas de gain en termes de taille de cluster (et donc probablement en termes d'économies de trafic

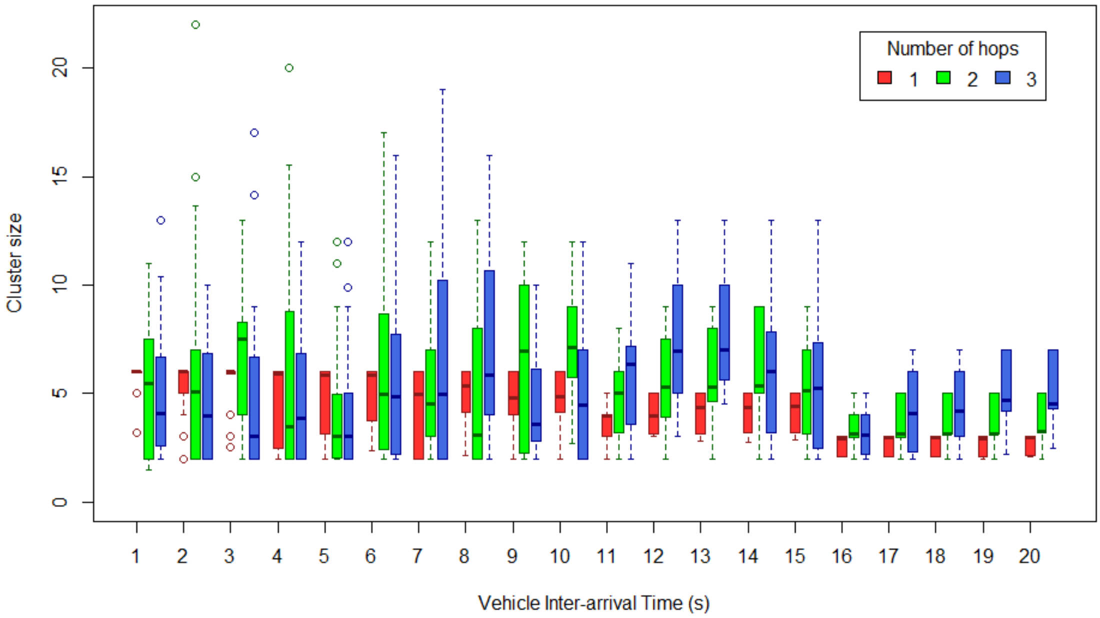


FIGURE 6 – Box plot montrant les tailles ds clusters formés par l'algorithme d'auto-proclamation des CH en fonction du temps d'inter-arrivées des véhicules, pour différents nombres de bonds.

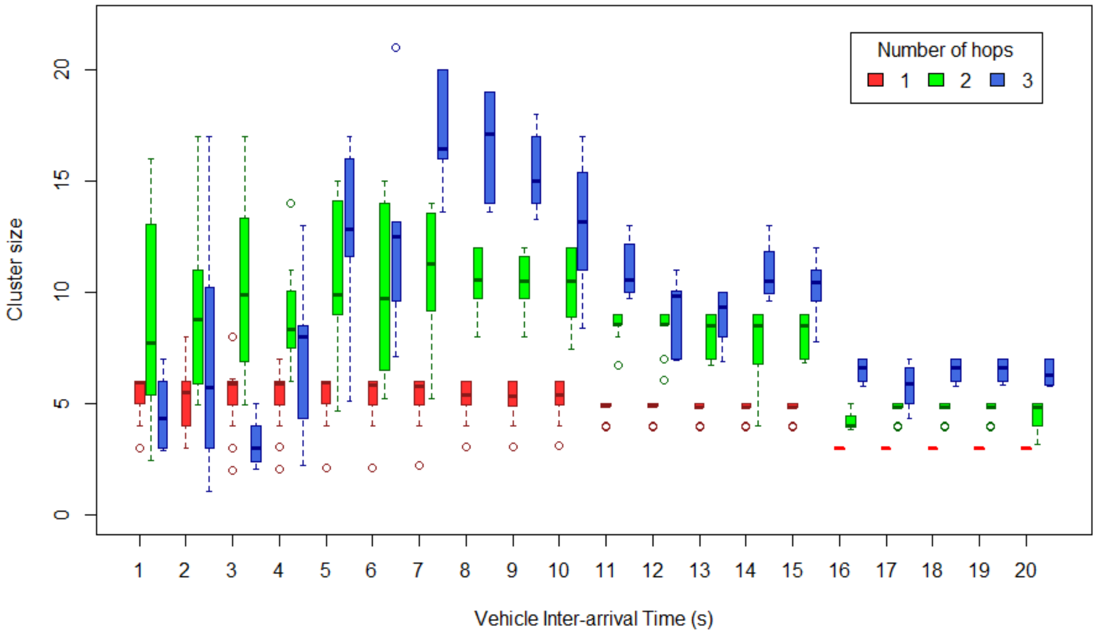


FIGURE 7 – Box plot montrant les tailles ds clusters formés par l'algorithme de sélection des CH déléguée à la BS en fonction du temps d'inter-arrivées des véhicules, pour différents nombres de bonds.

cellulaire non plus).

3.4.3 Taux de compression global

L'évolution de α en fonction du temps des inter-arrivées des véhicules est exprimée, en pourcen-

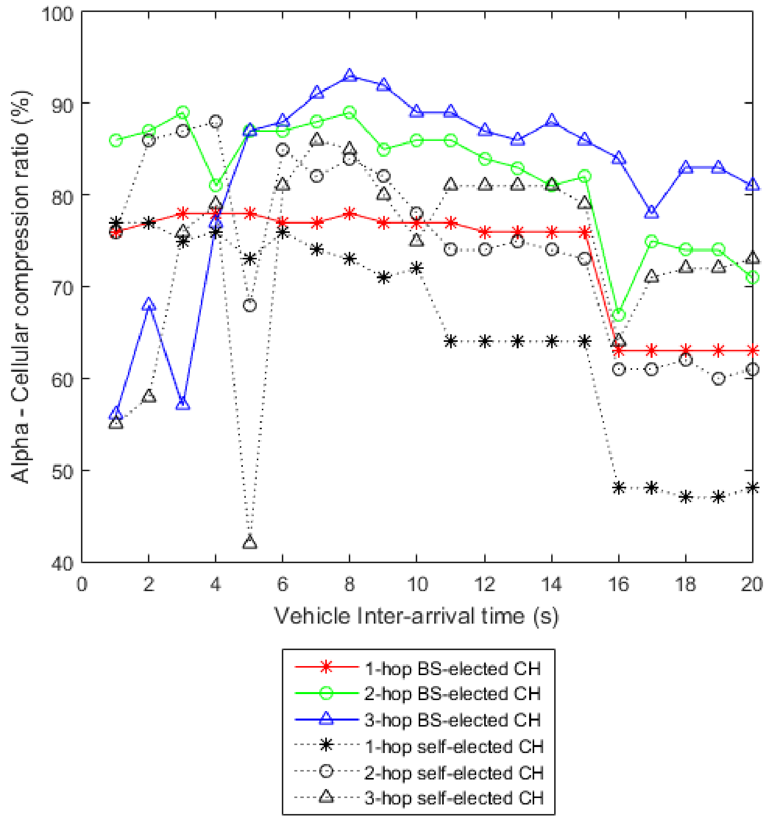


FIGURE 8 – Taux de compression de données global : Comparaison entre les algorithmes d'Auto-proclamation des CH et de Sélection du CH déléguée à la station de base.

tage, sur la figure 8 pour les deux algorithmes et pour $H = 1$, 2 et 3 bonds. D'inter-arrivées intermédiaires à élevées (c'est-à-dire des densités faibles à intermédiaires), les conclusions sont claires : permettre plus de bonds est la meilleure stratégie et notre algorithme de sélection des CHs déléguée à la BS surpasse l'algorithme d'auto-proclamation des CH. Cela est dû au fait que le taux de perte de paquets est maintenu à une valeur acceptable, de sorte que plus de sauts permettent des tailles de cluster plus grandes, ce qui à son tour améliore le taux de compression des données. Comme notre algorithme crée moins CHs, le taux de compression de données est également amélioré. Cependant, à des temps d'inter-arrivée faibles, le taux de perte de paquets devient inacceptable, ce qui empêche la formation de grands clusters. Les performances avec $H = 3$ deviennent donc les pires pour les deux algorithmes. Nous notons cependant que ce phénomène survient à un temps d'inter-arrivée plus faible avec notre algorithme, ce qui illustre la supériorité de l'algorithme de sélection déléguée.

3.5 Conclusions

Nous avons montré et quantifié l'impact du nombre maximum de bonds et de la densité du trafic véhiculaire sur la taille des clusters formées, et par conséquent sur l'efficacité de l'agrégation des informations sur le réseau cellulaire. Nous avons également proposé un nouvel algorithme qui, en déléguant la sélection du leader du cluster à la station de base cellulaire, nous permet de profiter beaucoup plus efficacement des avantages que l'augmentation du nombre de sauts peut apporter en termes de taille moyenne des clusters.

Lors du test de l'algorithme de clustering VMaSC pour la comparaison avec notre proposition, nous avons vu que l'augmentation de la taille des clusters lors de l'augmentation du nombre de sauts, un aspect souhaitable des algorithmes de clustering multi-saut, n'était que marginalement visible. C'est pourquoi l'algorithme proposé a été conçu en partie pour augmenter cet effet afin d'améliorer le taux d'agrégation des données.

Il apparaît que pour les densités élevées de véhicules, le clustering à 1 saut est l'option la plus pratique, car il y a une perte de paquets négligeable (alors que les clusterings à 2 et 3 sauts dépassent largement les niveaux maximales tolérés). Au fur et à mesure que la densité des véhicules diminue, l'option la plus pratique en termes de compression de données consiste à passer d'abord à un clustering à 2 sauts puis à un clustering à 3 sauts lorsque le taux de perte de paquets IEEE 802.11p diminue. C'est pourquoi nous développons dans la partie suivante un algorithme de clustering auto-adaptatif.

4 Algorithme de Clustering Auto-adaptatif

4.1 Motivation et objectif

Nous avons découvert qu'il existe certaines plages de densité de véhicules pour lesquelles, étant donné la contrainte d'un taux de perte de paquets maximum acceptable sur le réseau V2V, nous pouvons déterminer le nombre optimal de sauts pour un algorithme de clustering multi-sauts. Si nous pouvions laisser notre algorithme de clustering s'adapter automatiquement aux changements de la densité véhiculaire, nous assurerions une taille de cluster optimale pour chaque scénario, et nous obtiendrions ainsi une compression maximale des données transmises dans le réseau cellulaire, sans intervention humaine dans le système. Cela nous permettrait d'atteindre l'un de nos objectifs globaux, qui est de réduire les coûts de communication cellulaire, dans tous les scénarios.

Notre objectif à ce stade est donc de proposer un algorithme de clustering multi-saut auto-adaptatif qui s'adapte aux changements de densité de trafic, afin de réduire au maximum la consommation de données dans le réseau cellulaire pour une densité de véhicules donnée sans jamais dépasser le maximum tolérable taux de perte de paquets sur le réseau V2V.

4.2 Algorithme Auto-adaptatif

Notre algorithme consiste de deux parties : un algorithme de sélection de CH (on reprend celui de l'algorithme de base présenté dans la section précédente) et un algorithme d'adaptation du nombre de bonds.

Dans ces algorithmes, l'espace est divisé en **régions de clustering** (voir Figure 9), qui correspondent à la couverture d'une station de base unique (ou eNode-B). Chaque région de clustering est caractérisée par un nombre maximum de sauts H pour les clusters qui y sont formés. Avec la portée de communication V2V, ce nombre maximal de sauts détermine le **diamètre de clustering** (D) dans cette région. Chaque région de cluster de longueur L_r est divisée en **secteurs de clustering** de longueur D.

L'algorithme de sélection des CH est déployé et exécuté dans chaque station de base pour chaque région de clustering. Son rôle est de sélectionner un CH dans chaque secteur de regroupement de cette région et s'assurer qu'il y en ait toujours un. Il prend comme entrée le nombre maximum de sauts H pour sa région, tel que fourni par l'algorithme d'adaptation de saut. Il vérifie s'il y a un CH dans chaque secteur à intervalles réguliers de $T_{election}$ secondes. S'il y a un secteur où aucun CH n'est

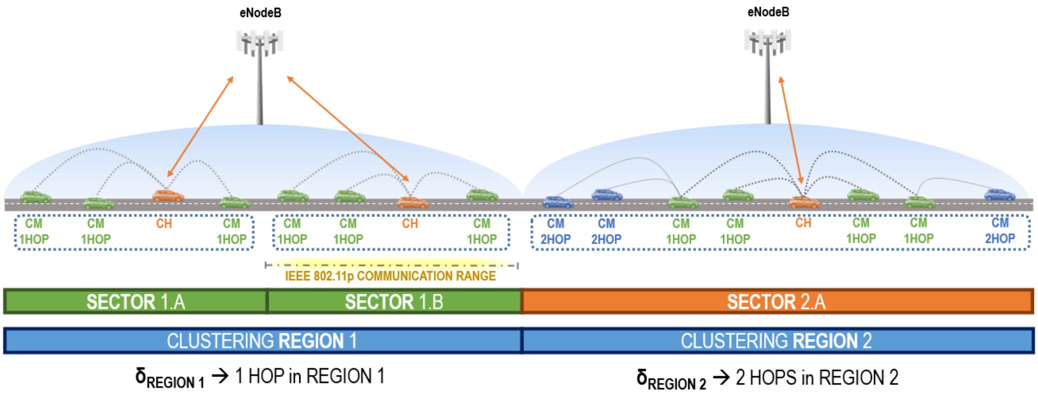


FIGURE 9 – **L'essentiel du modèle auto-adaptatif :** Une région de clustering peut être couverte par une ou plusieurs stations de base cellulaires (eNodeB dans le cas de LTE). Dans chaque région, le nombre maximum de sauts est déterminé dynamiquement par l'algorithme auto-adaptatif (exécuté par la station de base de la région) en fonction de la densité du trafic véhiculaire. Une région de clustering est divisée en plusieurs secteurs de clustering. La longueur du secteur de clustering est égale à la portée de communication IEEE 802.11p, multipliée par le nombre maximal de sauts dans la région. L'algorithme garantit que dans chaque secteur de clustering il y aura un leader de cluster. Seul le CH échange des données avec la station de base.

présent, il choisit comme CH le véhicule le plus proche du centre du secteur.

L'algorithme d'adaptation de saut est également implémenté dans chaque station de base pour chaque région de cluster et est responsable de l'adaptation dynamique du nombre maximum de sauts H en fonction de la densité du trafic véhiculaire. Pour chaque nombre de sauts H, il y a deux seuils $\Delta_{H-hop_{min}}$ et $\Delta_{H-hop_{max}}$. Si la densité observée est inférieure à $\Delta_{H-hop_{min}}$, nous permettons aux clusters d'être plus grandes en définissant $H := H+1$. Si la densité est supérieure à $\Delta_{H-hop_{max}}$, le taux de perte de paquets peut augmenter, donc nous diminuons le nombre de sauts de un. Notez que nous pouvons avoir $\Delta_{H-hop_{min}} \neq \Delta_{(H+1)-hop_{max}}$ pour prendre en compte l'hystérésis. L'adaptation de saut est faite chaque $T_{adaptation}$.

Comme la densité du trafic véhiculaire peut être hétérogène à l'intérieur d'une région de clustering, prendre des décisions basées uniquement sur la densité moyenne peut conduire à de mauvaises performances. Pour résoudre ce problème, nous définissons une **Zone d'Analyse Prédictive** (PAZ) au début de la région de clustering. La formule pour calculer la longueur de la PAZ (L_{PAZ}) est :

$$L_{PAZ} = min(\frac{L_r}{4}, 4 \times R) \quad (5)$$

La densité du trafic véhiculaire est mesurée à la fois dans la PAZ (δ_{PAZ}) et dans le reste de la région ($\delta_{\overline{PAZ}}$). Nous prenons ensuite la décision sur le nombre de sauts basé sur max$\{\delta_{PAZ}, \delta_{\overline{PAZ}}\}$. L'idée est de bénéficier des caractéristiques spécifiques de déplacement observées sur une section autoroutière afin d'éviter les pics de taux de perte de paquets : si la densité est plus élevée dans la PAZ, on anticipe le changement de saut ; si la densité est plus élevée dans le reste de la région, nous considérons le cas le plus défavorable. La complexité computationnelle de nos algorithmes en fonction du nombre de véhicules présents dans chaque secteur peut être caractérisée comme $O(n)$ (où n est le nombre de véhicules dans le secteur). Sachant que la taille d'entrée est contrainte aux éléments d'un secteur de clustering, il ne peut y avoir aucun problème de passage à l'échelle. C'est bien sûr la complexité des algorithmes présentés, mais pas la complexité de la simulation de l'acheminement multi-sauts.

Algorithm 2 Algorithme d'adaptation de sauts (station de base)

Initialisation :
Définir la période de maintenance $T_{adaptation}$.
Définir le nombre maximum de sauts initial $H := H_{default}$.
Définir la Zone d'Analyse Predictive comme les premiers $L_{PAZ} = \mathbf{min}(\frac{L_r}{4}, 4 \times R)$ mètres de la région de clustering.
Définir les seuils de déclenchement $\Delta_{k-hop_{min}}$ et $\Delta_{k-hop_{max}}$ pour $k = 1, 2, 3$.
Routine :
Pour $t = nT_{adaptation}$, $n = 1, 2, ...$, **faire**
    Calculer la densité véhiculaire $\delta = \max\{\delta_{PAZ}, \delta_{\overline{PAZ}}\}$.
    Si $\delta < \Delta_{H-hop_{min}}$ **alors**
        $H := H + 1$
        Notifier tous les véhicules dans la région de clustering et le processus de l'Algorithme de Sélection de CH du changement en H.
    Fin Si
    Si $\delta > \Delta_{H-hop_{max}}$ **alors**
        $H := H - 1$
        Notifier tous les véhicules dans la région de clustering et le processus de l'Algorithme de Sélection de CH du changement en H.
    Fin Si
Fin Pour

4.3 Simulation

4.3.1 Configuration

Les algorithmes présentés ont été implémentés et évalués en utilisant le framework Veins [69], qui synchronise une simulation de trafic qui tourne sur SUMO (Simulation of Urban MObility) [27] et une simulation de réseau full-stack dans OMNeT++.

La carte se compose d'un segment d'autoroute de 10 km de long, divisé en deux *régions de clustering* de longueur égale. En termes de trafic véhiculaire, le scénario testé est constitué de trois flux consécutifs de densités significativement différentes : de 0 s à 2500 s, 100 véhicules entrent dans la section autoroutière à un temps moyen d'inter-arrivée de 25 s. De 2500 à 4500 s, un second flux de 200 véhicules entre dans la section autoroutière à un temps moyen d'inter-arrivée de 10 s. Et enfin, à partir de 4500, un flux de 1600 véhicules entrera avec un temps moyen d'inter-arrivée d'une seconde. La densité de véhicules dans les deux régions de clustering (et dans tout le segment d'autoroute) en fonction du temps peut être vue dans la figure 10.

Nous supposons une fonction de compression égale à $\eta(N_c) = 1/N_c$, où N_c est la taille du cluster. Nous comparons la performance de l'algorithme auto-adaptatif à l'algorithme de clustering de base (en configurant le nombre de sauts à 1, 2 et 3 dans différentes exécutions) en termes de $\alpha(\mathscr{C})$ et PLR.

Nous définissons le taux de perte de paquets dans le réseau V2V comme le rapport entre les paquets perdus (en raison d'un décodage incorrect ou des collisions) et les paquets correctement reçus et décodés. Comme nous avons un média qui utilise la diffusion radio, la définition des messages que nous devrions considérer comme perdus peut ne pas être évidente. Dans notre cas, le simulateur réseau évalue l'affaiblissement de propagation du signal radio et peut générer les erreurs aléatoires associées. Nous considérons qu'un paquet est perdu si la puissance du signal reçu est suffisante pour déclencher le processus de décodage, mais cela conduit à un échec de décodage. Nous considérons également qu'un paquet est perdu en cas de collision dans le canal radio. Le PLR maximum tolérable dépend des contraintes spécifiques de chaque application. Pour un service de prévention coopérative (CA), nous avons fixé notre seuil à 10 pour cent. Dans notre modèle, la quantité de messages V2V *unicast* (surtout pour les messages pour rejoindre et quitter un cluster) est extrêmement faible par rapport à la quantité de messages CA diffusés (*broadcast*).

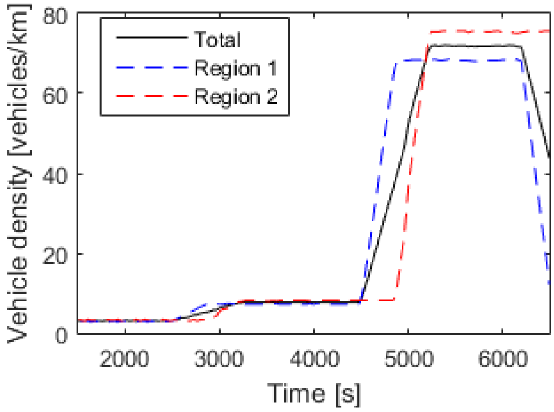


FIGURE 10 – Densité véhiculaire dans le scénario testé en fonction du temps, mésurée dans les régions 1 et 2, et densité dans la totalité du segment d'autoroute qui comprend les deux régions.

4.3.2 Résultats

Les résultats sont présentés dans des figures séparées pour les différents flux de trafic (Figure 10) pour une lecture et une analyse améliorées, mais le lecteur doit garder à l'esprit que ces figures font partie d'une simulation unique et continue.

Pendant la première partie de la simulation, une densité de trafic très légère est introduite. Les véhicules sont trop éloignés les un des autres et, comme nous pouvons le voir sur la figure 18a, l'algorithme à 1 saut est incapable de former des clusters assez grands et est sévèrement pénalisé dans ses performances d'agrégation par rapport aux configurations à deux et trois sauts. La meilleure performance d'agrégation est donc celle du nombre maximum de sauts : l'algorithme à 3 sauts est supérieur à tout moment, et l'algorithme auto-adaptatif suit son comportement. En termes de PLR (voir Figure 18b), tous les algorithmes restent au dessous de 1 pour cent.

Lorsque le deuxième flux de véhicules arrive (à la marque de 2500 s), les courbes de PLR et d'agrégation changent progressivement, et l'algorithme à 3 sauts dépasse le seuil d'acceptabilité du PLR de 10%. L'algorithme auto-adaptatif change alors le nombre de sauts, de 3 à 2, et nous pouvons voir une réduction significative du PLR après le pic que nous obtenons lorsque le nouveau flux commence (voir Figure 18b). La courbe de compression de l'algorithme auto-adaptatif commence à suivre la courbe à 2 sauts.

Enfin, pour la densité la plus élevée (Figures 18b et 18a), les courbes du PLR à 2 et 3 sauts ont monté en flèche, laissant la configuration à 1 saut comme la seule possibilité viable. L'algorithme auto-adaptatif déclenche à nouveau un changement de saut, résolvant un autre pic de PLR, tandis que sa performance d'agrégation suit la courbe de 1 saut.

Nous montrons maintenant que la signalisation induite par l'algorithme auto-adaptatif est négligeable par rapport au nombre de messages "économisés" par le clustering. Nous calculons d'abord le nombre de messages à destination du réseau cellulaire généré par chaque véhicule et le nombre de messages envoyés par les CH. Lors de la simulation de 6500 s, nous observons un gain de 516.184 messages grâce à la compression. Au cours de la même simulation, nous avons également observé

8 notifications de changement de numéro de saut (adaptations) et 98 proclamations de CHs (pour un total de 1900 véhicules dans la simulation). Dans le pire des cas, nous avons ainsi 98 × 8 de notifications de changement et 98 sélections de CH pour un total de 882 messages de signalisation utilisés par notre algorithme. Cela représente seulement 0,17% des économies en termes de nombre de messages. Même si les messages ont des longueurs différentes, cette estimation grossière montre que les surcharges de signalisation associées à notre algorithme sont négligeables.

Le fait que la courbe de l'algorithme auto-adaptatif suit la courbe de l'algorithme avec la meilleure configuration de sauts, dans le contexte de densité de trafic, montre l'utilité de l'approche adaptative.

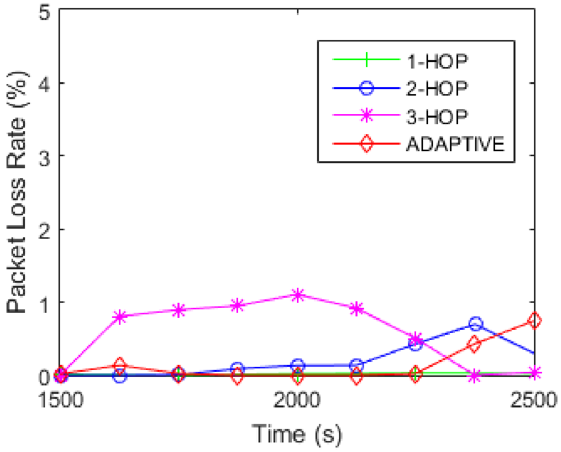
4.4 Conclusions

Nous avons considéré un réseau véhiculaire hybride, où les véhicules sont regroupés en clusters en utilisant IEEE 802.11p (V2V) et ils communiquent via leur CH avec un réseau cellulaire. Nous avons présenté un algorithme de clustering multi-sauts et auto-adaptatif qui modifie dynamiquement la taille des clusters (en adaptant le nombre maximal de sauts en fonction de la densité du trafic) dans le but de réduire l'utilisation de la ressource d'accès au réseau cellulaire tout en gardant le taux de perte de paquets du réseau V2V en dessous d'un seuil maximal spécifié. L'algorithme auto-adaptatif est déployé et exécuté sur chaque station de base du réseau cellulaire. En utilisant une plateforme de simulation d'ITS, nous montrons que l'algorithme auto-adaptatif s'adapte correctement aux changements de densité extrêmes, réduisant au mieux l'utilisation du réseau cellulaire (et réalisant ainsi d'importantes économies monétaires) dans les expériences de simulation et les résultats, tout en respectant des contraintes imposées sur le taux de perte de paquets sur le réseau V2V, garantissant ainsi que les exigences des applications spécifiques soient satisfaites.

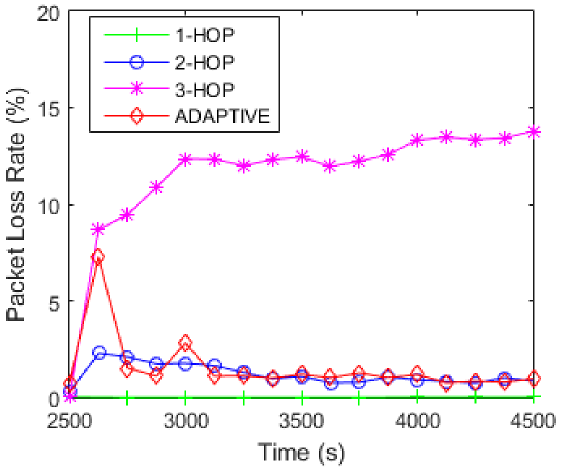
La **surcharge** du mécanisme d'adaptation est petite : il nécessite seulement la diffusion d'un message de la station de base vers les véhicules de la zone. Les véhicules qui sont, par exemple, à 3 bonds de leur CH, et reçoivent une notification indiquant que le nouveau maximum est 2, se dissocieront automatiquement de leur cluster actuel sans autre échange de messages. Leur CH saura qu'ils sont partis et les effacera de sa liste de membres, et les véhicules concernés commenceront immédiatement à écouter les nouveaux *CH Advertisements*.

Le mécanisme d'hystérésis implémenté avec les seuils $\Delta_{H-hop_{min}}$ et $\Delta_{H-hop_{max}}$ empêche la possibilité d'une fréquence excessive de déclenchement d'adaptation, améliorant la **stabilité du système**.

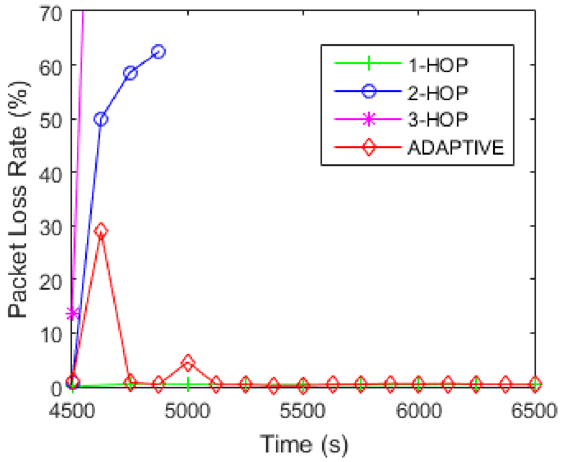
Même si elle réduit l'ensemble des coûts de communication des réseaux véhiculaires, cette méthode soulève le problème de la répartition inégale des coûts pour les leaders de cluster, qui effectuent toutes les communications avec le réseau cellulaire. Par conséquent, assurer l'acceptabilité sociale de l'approche proposée nécessite une meilleure répartition des coûts, ou une distribution plus “équitable”, entre les véhicules participants sur la durée.



(a) Faible densité

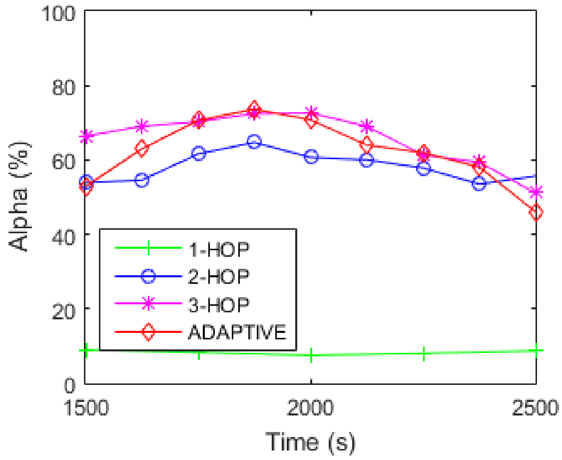


(b) Densité intermédiaire

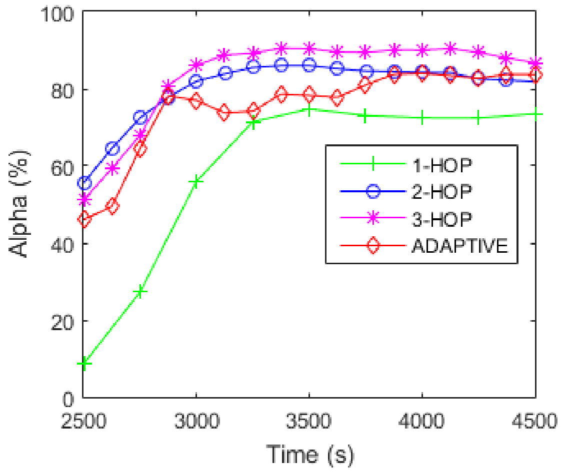


(c) Forte densité

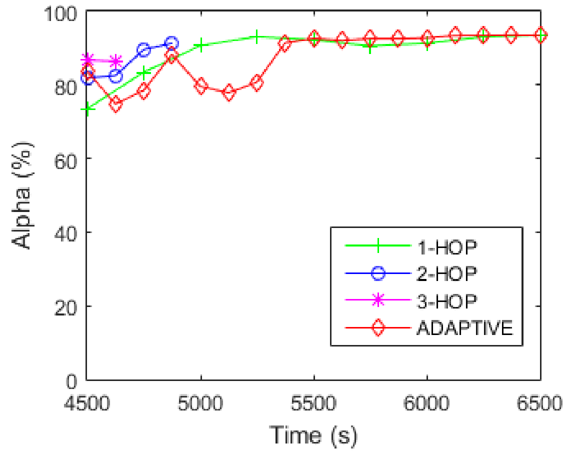
FIGURE 11 – Taux de perte de paquets (PLR) en fonction du temps pour le scénario testé. Comparaisons des configurations à 1, 2 et 3 sauts et l'Algorithme Auto-adaptatif.



(a) Faible densité



(b) Densité intermédiaire



(c) Forte densité

FIGURE 12 – Réduction de consommation du quota de données du réseau cellulaire (α) en fonction du temps pour le scénario testé. Comparaisons des configurations à 1, 2 et 3 sauts et l'Algorithme Auto-adaptatif.

5 Algorithme de clustering équitable

5.1 Introduction

Nous avons vu que les algorithmes proposés réussissaient à remplir le premier de nos deux objectifs globaux : *fournir une solution pour intégrer les réseaux V2V et cellulaires, afin de réduire les coûts d'accès cellulaire tout en préservant les performances du réseau*. Dans cette dernière partie de notre travail, nous nous concentrons sur l'extension de notre solution afin d'atteindre le deuxième objectif global de cette thèse : *assurer l'équité sur la durée de la méthode proposée, afin d'améliorer son acceptabilité sociale*.

Nous sommes confrontés au problème de la répartition équitable des coûts de communication cellulaire que les leaders de cluster doivent absorber pour le bénéfice de tous les véhicules du cluster. Même si la répartition des coûts est nécessairement injuste à court terme (puisque tous les coûts sont pris en charge par les CHs), nous pouvons atteindre une répartition des coûts relativement équitable à long terme, en tenant compte des critères d'équité pendant le processus de sélection. Pour cela, nous avons appliqué la *théorie de la justice distributive* présentée par Rescher [65] dans le but d'atteindre *l'équité sur la durée*, suivant les principes de conception pour la gestion des biens communs présentés par Elinor Ostrom [60], qui assurent une gestion durable des ressources à travers des principes explicitement conçus pour l'acceptabilité sociale consensuelle. Ces principes stipulent, par exemple, qu'il est nécessaire que les personnes affectées par les règles de gestion d'une ressource puissent participer à la création de ces règles, et qu'il y ait un suivi cohérent et des sanctions graduelles.

5.1.1 Une théorie de la Justice Distributive

Il est impossible de produire un résultat équitable lors de la désignation d'un leader de cluster, mais nous visons à obtenir un résultat de désignation globalement équitable à long terme.

Rescher [65] introduit dans ses travaux le concept de *justice sociale*, qui consiste à déterminer quelles sont les revendications légitimes d'un individu et à traiter chacune de ces revendications de la même manière. Son travail n'examine pas comment ces revendications peuvent émerger, mais se concentre plutôt sur le concept de *justice distributive*, qui prend en compte la mesure dans laquelle les revendications légitimes de chaque individu devraient être satisfaites, les revendications concurrentes d'autres individus et les ressources limitées.

Rescher a analysé les méthodes les plus habituelles utilisées par d'autres auteurs pour mesurer l'équité et a proposé ce qu'il a appelé les *sept canons de la justice distributive* : traitement comme des égaux, selon les besoins de chacun, selon la contribution productive de chacun, selon les efforts et sacrifices, la valeur sociale des services fournis par l'individu à la société, l'offre et à la demande, ou selon les mérites et réalisations. L'auteur a constaté qu'aucun de ces canons, pris individuellement, ne pouvait accorder une véritable justice distributive. Au lieu de cela, il a proposé d'analyser les revendications légitimes de chaque participant par rapport à chacun de ces aspects, et de se concentrer sur la manière de les équilibrer en cas de conflit.

5.2 Algorithme

Jusqu'ici nous avons proposé un algorithme de clustering auto-adaptatif qui vise à optimiser l'utilisation des ressources du réseau cellulaire. Pour plus de clarté, nous appellerons celui-ci l'**Algorithme Agnostique par rapport à l'Équité** (et nous l'abrégerons comme *Algorithme Agnostique* ou *Fairness-Agnostic*). Le problème que pose cet algorithme est que, puisque le CH est le seul à affronter le coût en termes de consommation de quota cellulaire à un moment donné, cela peut conduire à une distribution inéquitable des coûts, ce qui peut devenir socialement inacceptable. Par conséquent, la contribution présentée vise à fournir un **Algorithme Vigilant de l'Équité** (abrégé comme *Algorithme Vigilant* ou *Fairness-Aware*), qui répond à ce problème.

Nous proposons un *mapping* pratique de notre problème de distribution du coût de l'accès au réseau cellulaire parmi les véhicules du cluster en termes des sept canons de Rescher (puisque le septième canon ne peut pas être traduit en termes de mérites spéciaux ou de réussites particulières dans notre modèle puisqu'il n'y en a pas, nous nous concentrons uniquement sur les six premiers, fusionnant deux d'entre eux qui deviennent équivalents). De cette façon, nous créons cinq critères qui constituent cinq ordres totaux des véhicules dans un secteur de clustering, qui sont ensuite utilisés pour décider qui doit être le leader du cluster. Selon chaque canon, on présente les méthodes pour déterminer l'ordre dans lequel les véhicules devraient être élus comme CH :

- **Le canon de l'égalité** : plus un Membre du Cluster reçoit de données à travers un CH, plus ses chances d'être sélectionné comme chef de cluster sont élevées ;

- **Le canon des besoins** : plus la quantité de quota cellulaire disponible est grande, plus les chances d'être CH sont élevées ;

- **Le canon de la productivité** : plus le volume de données précédemment partagé en tant que CH est grand, plus les chances d'être sélectionné comme CH sont faibles ;

- **Les canons de l'effort / utilité sociale** : plus le nombre de fois qu'un véhicule a servi comme CH est élevé, plus les chances d'être sélectionné comme CH sont faibles ;

- **Le canon de l'offre et de la demande** : plus le nombre d'événements de non-conformité aux règles est élevé (par exemple, en désactivant la connexion de données lorsqu'on occupe le rôle de CH), plus les chances d'être sélectionné sont élevées. Il est important de noter que ce *mapping* entre les canons et les variables de notre problème est fait sous l'hypothèse que nous avons affaire à un problème de **free-riding**.

Ces canons constituent des ordres totaux : ils peuvent fournir une liste ordonnée, où tous les véhicules peuvent être classés en ordre croissant ou décroissant selon ce critère. Les données nécessaires pour calculer les cinq ordres totaux pour un groupe de véhicules sont toujours disponibles pour la station de base cellulaire, qui est, dans notre modèle, celle qui prend la décision finale sur qui sera proclamé CH.

Chaque véhicule a seulement la connaissance de toutes les métriques spécifiques pour lui, et peut ainsi déduire quels critères sont les plus favorables à sa situation. Chaque véhicule a alors la possibilité de demander d'être jugé selon la *revendication légitime* la plus avantageuse pour lui. Par exemple, s'il a été élu CH trop souvent, ou s'il a un quota disponible très faible, il peut exiger que ces critères aient une plus grande importance. Tous les véhicules d'une région de cluster émettent un vote qui affecte les *poids* de chaque canon pour ce groupe de véhicules en particulier. La station de base, en connaissance du résultat de ce vote, établit une élection Borda [33] pondérée : chaque canon agit comme un électeur dans un vote Borda. Par exemple, selon le canon de l'égalité, l'ordre de priorité pour être élu comme CH pourrait être *"1) Véhicule A ; 2) le véhicule B ; 3) Véhicule C"* tandis que le canon des besoins pourrait déterminer *"1) Véhicule B ; 2) le véhicule C ; 3) Véhicule A"*. Si dans cet exemple les poids de ces deux critères étaient les mêmes, le *Véhicule B* serait choisi comme CH.

La décision finale est prise en suivant ces pas :

1. La station de base découvre une région de clustering sans CH et demande aux véhicules dans la zone de faire un vote pour leur canon le plus souhaité ;

2. Chaque véhicule estime le canon qui est le plus favorable pour sa situation et vote localement pour ce canon. Chaque véhicule émet un vote unique, pour un seul canon. D'autres systèmes de vote pourraient être implémentés aussi.

3. Le résultat du vote est l'ensemble des poids pour les différents canon, déterminé par la station de base après réception des votes. Le poids du canon j sera déterminé par la formule suivante :

$$w_j = \frac{\sum_{i=1}^{N_c} \mathbf{v}_\mathrm{i}[j]}{N_c}, \tag{6}$$

où N_c est le nombre de véhicules dans le secteur de clustering et $\mathbf{v}_\mathrm{i}[j]$ est une valeur binaire du vecteur de vote du véhicule i qui indique s'il a voté pour le canon j ou pas.

4. La station de base cellulaire calcule les ordres totaux associés à chaque canon ;

5. La station de base cellulaire exécute l'élection de Borda pondérée, en appliquant le coefficient de poids à chaque vote Borda émis par chaque canon-électeur ;

6. La station de base cellulaire applique une correction finale à l'ordre total obtenu du vote Borda : une normalisation qui rabaisse les points obtenus pour être nommé CH au fur et à mesure que la distance par rapport au centre du secteur de clustering augmente.

7. La notification est envoyée au CH sélectionné.

Il est important de noter que, lors du passage d'une station de base à l'autre, les clusters restent inchangés. Seulement quand il n'y a pas un CH dans un *secteur de clustering* une nouvelle sélection de CH aura lieu. Et à cette occasion, toute station de base aura accès à la même information à jour sur le compte de chaque utilisateur auprès de l'opérateur de téléphonie mobile. Alors, il n'y a pas de possibilité que le passage d'une station de base à l'autre affecte l'équité.

5.3 Simulation

5.3.1 Configuration

Les tests ont été fait en simulant un groupe de 100 véhicules qui traverse un segment d'autoroute de 10 km, environ 100 fois chacun, avec un ordre de rentrée aléatoire. Le protocole V2V simulé est IEEE 802.11p. Le protocole pour les couches réseau et transport est IEEE 1609.3. Dans la couche application nous implémentons le *Cooperative Awareness Messages* (CAM) de la norme européenne ETSI ITS G5. L'affaiblissement de propagation du signal suit un modèle d'interférence à deux rayons [70].

Dans les courbes de résultats, là où le temps est représenté sur l'axe horizontal, il s'agit de l'heure simulée, où les 100 véhicules rentrent au hasard avec un temps d'inter-arrivée déterministe. C'est donc l'équivalent du temps qu'il faudrait à 10 000 véhicules pour parcourir le segment autoroutier de 10 km, avec un temps d'inter-arrivée uniforme.

Dans les boîtes à moustaches (*box plots*) que nous présentons, la marque à l'intérieur de la boîte représente la médiane (deuxième quartile), tandis que la boîte est délimitée par les premier et troisième quartiles. Les moustaches marquent la donnée la plus basse et la plus élevée dans la fourchette interquartile de 1,5 (*Inter-Quartile Range*) des quartiles inférieur et supérieur, respectivement. Les valeurs aberrantes (*outliers*) sont marquées comme des cercles.

5.3.2 Résultats

Véhicules ayant servi comme CH

Le résultat le plus remarquable de l'introduction de la justice distributive à long terme est que le rôle de CH, qui semble être un fardeau temporaire, sera pris par une plus grande proportion des participants si nous analysons des échantillons successifs dans le temps.

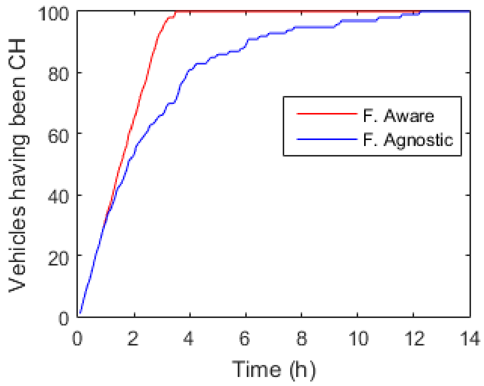


FIGURE 13 – Nombre de véhicules ayant servi comme CH *au moins une fois* en fonction du temps.

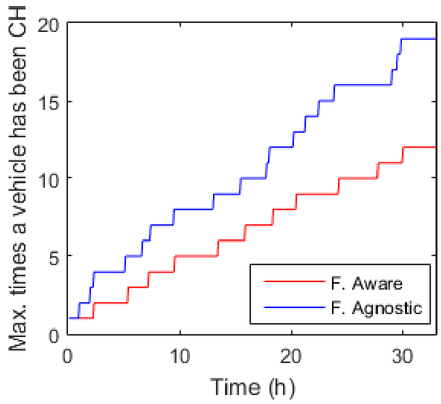


FIGURE 14 – Nombre maximum de fois qu'un véhicule a servi comme CH en fonction du temps.

Dans la figure 13 nous pouvons voir comment la courbe du nombre de véhicules ayant été CH augmente beaucoup plus rapidement en utilisant l'*Algorithme Vigilant* : après **moins de 4 heures simulées**, tous les véhicules ont pris le rôle de CH au moins une fois. En revanche, avec l'*Algorithme Agnostique*, c'est seulement après **plus de 12 heures simulées** que l'on arrive au même résultat.

De plus, nous devons analyser combien de fois le même véhicule a été sélectionné comme CH. Réduire le nombre de fois qu'un utilisateur spécifique prend ce rôle est un moyen important pour améliorer la perception de justice de l'utilisateur. Dans la figure 14 nous pouvons observer l'évolution du nombre maximum de fois qu'un véhicule spécifique a été élu CH. Nous voyons que, pour la valeur maximale, l'*Algorithme Agnostique* est toujours dans une situation de presque doubler son homologue *Vigilant*.

Le graphique de la figure 15 montre la distribution des assignations comme CH parmi les véhicules, au fil du temps. Quel que soit le temps qui passe, la taille des cases de l'*Algorithme Vigilant* reste petite et leur taille ne change pas. Leurs moustaches sont généralement minuscules ou inexistantes. Cela signifie que, même si le nombre moyen de fois que les véhicules en général servent comme CH augmente avec le temps (ce qui est nécessaire), tous les véhicules servent approximativement le même nombre de fois comme tels. D'un autre côté, pour les cases de l'*Algorithme Agnostique*, leur taille augmente, et les moustaches ne cessent de s'agrandir, montrant des disparités extrêmes entre

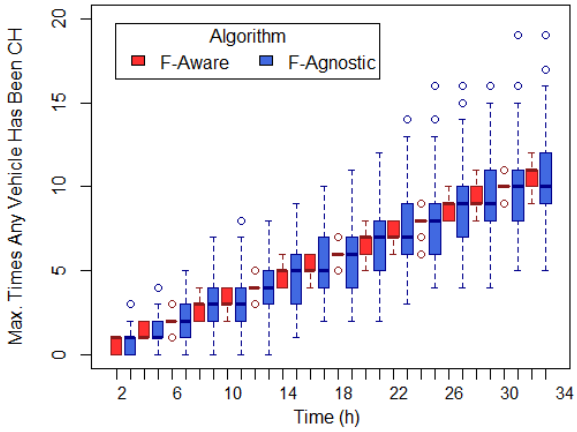


FIGURE 15 – Boîte à moustaches montrant la distribution du nombre de fois que chaque véhicule a servi comme CH pour les deux algorithmes en fonction du temps.

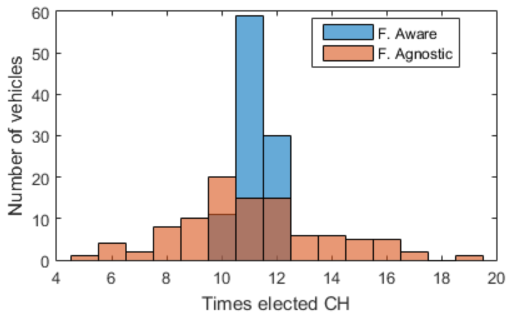


FIGURE 16 – Histogramme du nombre final de fois que chaque véhicule a servi comme CH.

les participants.

L'effet à long terme peut être vu dans l'"*image finale*" de la figure 16 où l'on peut voir l'histogramme du nombre de fois que chaque véhicule a servi comme CH à la fin de la simulation. Nous voyons une dispersion significative entre les valeurs allant de 4 à 20 pour l'*Algorithme Agnostique*, alors que pour son homologue *Vigilant*, près de 60% des véhicules servaient exactement 11 fois comme CH, tandis que les autres 40% l'ont fait soit 10, soit 12 fois.

Consommation du quota cellulaire

Les différences dans les résultats de la sélection de CH dont nous venons de parler au point précédent ont un impact direct sur la consommation de quotas cellulaires. Nous allons maintenant voir quelques exemples de la distribution de l'utilisation du quota cellulaire avec les deux algorithmes, ce qui se traduit presque directement par des coûts économiques différents.

Un meilleur "*pire des cas*"

Nous commençons par analyser le cas de notre utilisateur le plus défavorisé pour les deux algorithmes dans la figure 17. Cette courbe montre le quota individuel le plus bas parmi les véhicules qui ont été CH, avec le temps. Nous pouvons voir une différence marquée dans les pentes. Cela signifie que même

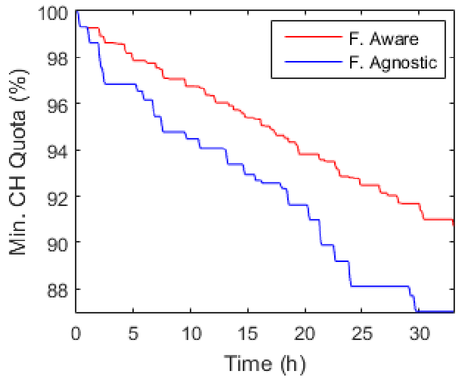


FIGURE 17 – Minimum de quota cellulaire disponible parmi les véhicules ayant servi comme CH pendant la simulation.

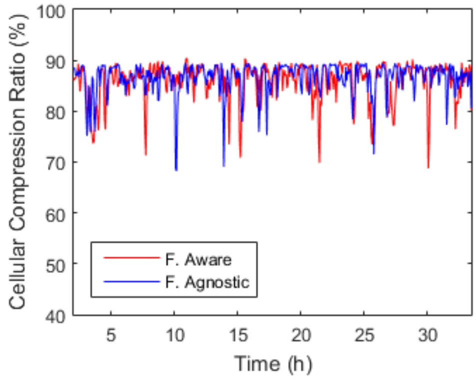
le participant le moins favorisé sera toujours beaucoup mieux, en termes de coût économique, avec l'*Algorithme Vigilant* qu'avec l'Agnostique.

Performance des réseaux

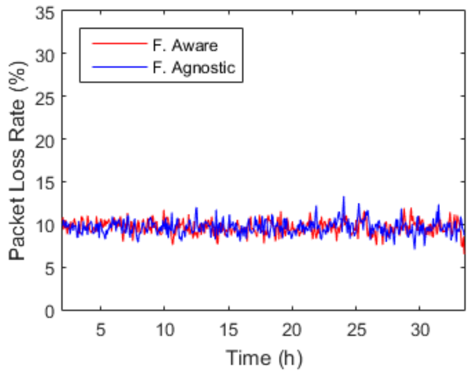
L'*Algorithme Agnostique* visait à auto-adapter la taille des clusters afin de maintenir un équilibre optimal en termes de performance réseau : maximiser la compression des données dans le réseau cellulaire (et donc réduire les coûts économiques) limiter le taux de perte de paquets (PLR) dans le réseau V2V à un seuil maximum acceptable de 10%. Nous pourrions nous attendre à ce que la modification de la géométrie et des critères du processus de sélection de CH affecte les métriques pour lesquelles cet algorithme a été conçu. Les figures 18a et 18b comparent les deux algorithmes en termes de compression de données et de PLR, respectivement. Comme on peut le voir clairement, les améliorations de la justice distributive **n'ont aucun coût** en termes de performance du réseau.

5.4 Conclusions

Nous avons présenté un Algorithme de Sélection de Leader de Cluster Vigilant par rapport à l'Équité basé sur la théorie de Rescher de la justice distributive. Cette solution vise à résoudre le problème de l'acceptabilité sociale des algorithmes de clustering, où les individus sélectionnés comme CH doivent absorber tous les coûts. Nous avons appliqué cette approche à nos travaux précédents sur un Algorithme de Clustering Auto-adaptatif, conçu à son tour pour réduire les coûts d'accès cellulaire en ne communiquant que des données agrégées via le CH, tout en limitant les pertes de paquets sur le réseau V2V. L'algorithme précédent était efficace en termes de compression globale des données et de performance du réseau, mais injuste vis-à-vis des sélections de CH et donc de la répartition des coûts entre les utilisateurs du système (et comme un conducteur peut suivre sa consommation de quota, cela peut devenir une incitation à quitter le système, un phénomène qui peut faire effondrer le modèle). En revanche, l'algorithme d'équité proposé permet aux véhicules d'influencer les sélections de CH en exprimant leurs revendications légitimes via un vote. Les simulations ont montré que cette approche améliore significativement les métriques d'équité pour tous les véhicules, au fil du temps. Les simulations ont également montré que l'algorithme *Vigilant par rapport à l'Équité* préserve toutes les optimisations de performance réseau obtenues par son homologue *agnostique*.



(a)



(b)

FIGURE 18 – Comparaison entre les algorithmes *Vigilant* et *Agnostique*, en ce qui concerne : (a) Le taux de compression des données sur la liaison du réseau cellulaire en fonction du temps (simulé). (b) Taux de perte de paquets (PLR) sur le réseau V2V. Les résultats montrent que l'intégration des améliorations de l'équité n'a pas de coût dans les métriques de performance du réseau pour lesquelles l'algorithme *Agnostique* a été conçu.

6 Conclusion

Dans cette thèse, nous avons abordé le problème de l'intégration de différentes technologies d'accès radio dans les réseaux véhiculaires. Après avoir analysé l'état de l'art, nous avons découvert que les normes de communication de véhicule-à-véhicule telles que IEEE 802.11p et les normes cellulaires telles que *Long Term Evolution* (LTE / 4G) sont très différentes mais leurs caractéristiques sont complémentaires.

Les futures applications des systèmes de transport intelligents (ITS) nécessiteront de grands volumes de données qui devront être téléchargées par les véhicules vers des serveurs distants via le réseau cellulaire. Après avoir exploré différents modèles économiques, nous avons réalisé que le coût économique de l'accès au réseau cellulaire pouvait devenir un problème majeur. Cependant, certaines applications pour les réseaux véhiculaires sont caractérisées par une redondance locale qui peut être exploitée pour l'agrégation et le *off-loading* d'information, ce qui réduit considérablement l'utilisation des ressources cellulaires.

Pour ce faire, nous avons proposé un algorithme de clustering multi-saut pour l'agrégation d'informations et la réduction de l'utilisation du réseau cellulaire. Il a été présenté en trois étapes. L'**Algorithme de Clustering de Base** a servi l'objectif principal d'une agrégation de données efficace, mais à un nombre fixe de sauts. Il nous a permis d'identifier un nouveau problème : pour certaines densités de trafic, lorsque la taille des clusters augmente, le taux de perte de paquets (PLR) dans le réseau véhicule-véhicule (V2V) augmente également considérablement en raison des tempêtes de diffusion (*broadcast storms*).

Dans un deuxième temps, nous avons proposé l'**Algorithme de Clustering Auto-adaptatif**, qui modifiait dynamiquement le nombre de sauts en fonction de la densité du trafic. Les résultats de la simulation montrent que lorsque la densité du trafic change, les performances de cet algorithme en termes d'agrégation de données et de taux de perte de paquets sur le lien V2V atteignent celles de l'*Algorithme de Clustering de Base* dans la configuration de sauts la mieux adaptée au scénario actuel. À ce stade, nous avions un algorithme qui intègre efficacement les deux technologies d'accès radio — *V2V et cellulaire* — dans toutes les situations (du point de vue des variations de trafic).

Un autre problème est apparu lors de l'analyse de cette nouvelle approche : le CH est celui qui absorbe tous les coûts des communications cellulaires pendant la période où il occupe ce poste, tandis que les autres membres du cluster ne touchent pas leurs quotas. Il est nécessaire d'assurer une répartition équitable dans le temps des responsabilités d'être sélectionnée comme CH afin que le système soit socialement acceptable. Nous avons proposé l'*Algorithme de Clustering Vigilant par rapport à l'Équité*, qui incorpore la théorie de Rescher de la justice distributive. Les simulations ont montré que pour chaque métrique considérée (consommation de quota, nombre de fois ayant servi comme CH, volumes de données partagées en tant que CH, etc.), l'*Algorithme Vigilant* a fourni un résultat beaucoup plus équitable, plus rapidement que l'*Algoritme Auto-adaptatif* (ou *Agnostique par rapport à l'Équité*).

Nous avons proposé un algorithme qui intègre efficacement deux technologies d'accès radio différentes dans les réseaux véhiculaires. La connexion des réseaux véhiculaire et cellulaire, ainsi que l'utilisation coordonnée de différentes technologies d'accès radio sont des objectifs clairs du développement des réseaux mobiles de cinquième génération (5G). Le travail présenté dans cette thèse peut donc enrichir la littérature pour de futurs travaux de recherche dans ce domaine.

Les futures orientations de travail pourraient inclure l'intégration d'approches d'apprentissage par renforcement pour le choix (et peut-être le changement dynamique) des seuils de déclenchement de l'adaptation du nombre de sauts, prenant en compte non seulement les effets du trafic mais aussi ceux de la propagation radio. Les seuils actuels ont été fixés pour un modèle de propagation radio donné, mais ils peuvent changer, par exemple, si l'affaiblissement de propagation est plus élevée ou si l'écart-type de l'effet de masque est plus grand. Ces paramètres ont un impact direct sur le PLR.

Une première approche pour prédire les variations de densité du trafic dans les régions de regroupement a été proposée, et elle consiste en une zone spéciale appelée zone d'analyse prédictive (PAZ). Dans les travaux futurs, cette prédiction pourrait être renforcée afin d'éviter un court pic de PLR observé au moment du déclenchement du changement du nombre de bonds. Nous pourrions, par exemple, utiliser des messages inter-cluster qui permettraient d'avoir une idée du trafic entrant.

Nous avons considéré un scénario relativement simple avec un seul segment de route, et le trafic dans une seule direction. Il serait intéressant de voir comment l'algorithme proposé se comporte avec un trafic bidirectionnel, ou en présence de rampes d'entrée et de sortie. En accord avec ces idées, il serait intéressant d'utiliser de vraies traces véhiculaires ou de faire une expérience réelle.

Nous avons considéré le protocole IEEE 802.11p car il s'agit d'un protocole candidat pour ce type d'applications, mais des communications périphérique à périphérique (D2D) partiellement contrôlées par la station de base pourraient être utilisées pour l'échange de messages intra-cluster. Cela pourrait réduire considérablement le PLR au prix d'une utilisation plus importante du réseau cellulaire. C'est aussi une option à étudier. Un autre point important concernant les protocoles est l'aspect sécurité. Afin de mettre en œuvre nos algorithmes, une analyse spécifique des protocoles doit être effectuée afin de déterminer comment transférer les paquets (désencaptulation et analyse suivi de la construction d'un nouveau paquet, ou encapsulation du paquet avec sa signature originale), comment empêcher l'usurpation d'identité et les attaques du type “homme au milieu”, et d'autres questions similaires.

Enfin, en ce qui concerne l'intégration des approches d'équité, des travaux futurs pourraient être réalisés pour comparer avec d'autres propositions, ou différentes méthodes de vote, et notamment faire des expérimentations pour tester le degré d'acceptabilité de la méthode.

Contents

List of Figures

Chapter 1

Introduction

1 Context and motivation

We may be witnessing the beginning of the biggest evolution of human mobility in a century. If all the transformations foreseen at present are confirmed, our transportation systems will be more integrated, smarter and optimized, and our vehicles will be autonomous, connected, and carbon-free. The driving experience will change drastically, with the very existence of the role of human *driver* being put in doubt.

There is no doubt that for these *Intelligent Transportation Systems* (ITS) to even exist, vehicles and infrastructure need to be connected. As a first step, the European Union has passed legislation [2] that requires all new cars[1] to be equipped with a cellular connection, implementing the *eCall* system. In the event of an accident, eCall automatically calls emergency services for help. The volume of network traffic generated by this system is negligible, and so it has been agreed that it will be provided free of charge. However, with the introduction of autonomous vehicles, the technological constraints are set to change completely. Autonomous driving equals constant safety-critical communication. The most robust reliability and the lowest latency times will be mandatory. With autonomous vehicles, every occupant in the vehicle will be able to access media and information without worrying about navigation and driving. The demand for mobile broadband services will significantly increase, coupled with very strict quality-of-service constraints for security information, for great numbers of simultaneous connections. These conditions cannot be met by present-day technologies individually.

Several radio access technologies (RAT) exist nowadays that can serve the purpose of connecting vehicles. Both the IEEE and the European Telecommunications Standardization Institute (ETSI) have developed their own approaches for communications in vehicular networks, both relying mainly on the IEEE 802.11p wireless access protocol for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Several research works were also conducted for implementing vehicular connectivity over the cellular network, and we have lately seen the first cars equipped with cellular communication devices entering the mass market. V2V and cellular technologies have different (and sometimes complementary) advantages and limitations.

However, up to this date, there is not a consensual vision on how to integrate the multiple RATs in order to use network resources in a smart way that can compensate each technology's disadvantages with the complementary features of the others, in order to have a single approach that can push the limits we face today. We are transforming our world so that everything becomes connected,

[1] From April 2018.

but every type of communication relies on a specific technology, often restrained to a small scope of applications. This is about to change with the introduction of 5G, the fifth generation of cellular communications.

Until now, the Internet of Things (IoT) has been treated separately, as independent networks connected with specific technologies like LoRa or IEEE 802.11p in the case of vehicular networks (or by adding specific IoT functionalities to existing technologies, e.g. NB-IoT, a version of LTE for machine-to-machine communications).

Fifth Generation (5G) mobile networks will natively integrate IoT components. It will be composed by a New Radio interface (NR), but also by LTE (Fourth Generation cellular standard), WiFi and other technologies, and everything will be integrated in the same core network. The transition to 5G will be different for each type of IoT application:

- **Very high bandwidth - Mobile Broadband**: the applications that use 3G or 4G today will smoothly transition to 5G when available, getting the benefits of a higher bandwidth and lower latency, paving the way for improvement in their capacities. For instance, we could receive a 4K video signal from a drone with a very responsive control, instead of having to pilot it with a blurry HD image and a perceivable lag, as it happens today with drones controlled over WiFi.

- **Low-bandwidth and low-energy IoT applications**: small sensor networks commonly use technologies like LoRa or Sigfox (low-power wide area networks, or LPWAN), that can cover very long distances while keeping a long battery lifetime. Some mobile operators are also planning to introduce NB-IoT (another LPWAN, developed and standardized by 3GPP) in order to offer yet another solution in this area. NB-IoT will be integrated in 5G since it is part of LTE, which will be managed by 5G networks.

- **Critical communications**: 5G will provide a significant progress in this area, with a 99.999% reliability and a latency in the order of a few milliseconds. This will enable the development of applications whose requirements were not met with the technologies we have nowadays. In the domain of vehicular networks, we talk about safety critical applications like autonomous driving.

All these types of applications are, or will be, present in Intelligent Transportation Systems: low-energy sensors for measuring traffic, high-bandwidth video transmission from infrastructure or on-board cameras for either security purposes or image processing for other services, and safety critical messaging in vehicular networks.

In short, the novelty of 5G mobile networks is the smart integration of diverse Radio Access Technologies in order to have a single and cohesive approach to communications that can erase existing constraints and technological limitations specific to individual protocols and standards (coverage, bandwidth, latency, etc.), while offering a much higher network capacity that can smoothly connect everyone and everything, everywhere, to the extent needed or desired.

Based on today's technologies, we notice that the main alternatives for vehicular networks are very dissimilar: on one hand, we have the main V2V protocol, **IEEE 802.11p** —a protocol which works on a *licence-free* band of the radio spectrum, and is used for creating *ad-hoc* connections between vehicles as fast as possible. It has *no need for a central server*, and thus there is *no guarantee of connection to the internet backbone*. It is intended for communications in a *local area*, and it is prone to *saturation and packet loss*. On the other hand, we have the cellular networks using standards up to **4G / LTE**. This type of connection is *centralized* and it *depends on the presence of base stations* nearby. Coverage is nearly ubiquitous, but certainly not a hundred percent. However, when the user is in the coverage area, there is a permanent *connection to the internet backbone*. The network is very robust in terms of volume of traffic and bandwidth, but of course, it has a *paid access*.

Fortunately, these *dissimilar* technologies are *complementary*, and we can use both. This is the concept of a *hybrid vehicular network*. In this context, the motivation of this work is to investigate the possibility of integrating V2V (*IEEE 802.11p*) and cellular network (*LTE*) access into a hybrid vehicular network. The main purpose is to reduce the vehicles' communication costs, while still ensuring access to on-line information (e.g. local maps, weather conditions, current traffic) and data uploading.

We propose a solution that combines adaptive multi-hop clustering – for aggregating data and reducing costly communication – with the theory of distributive justice – for fairly distributing the remaining costs amongst vehicles, over the long term (Section 3).

2 Problem

There is one important factor when it comes to using the cellular network: as the used spectrum is licensed and managed by operators, the quality of service is expected to be better than in unlicensed technologies like IEEE 802.11p but the cost of using it is also higher. There are several possible economic models for the payment of the cellular access cost, but all of them have an impact for the final consumer. Data volume quotas may also be imposed. Hence, cellular network access becomes a costly and scarce resource.

On the other hand, V2V protocols such as IEEE 802.11p, while free of any monetary cost, do not necessarily (and easily) provide internet access. For this, Road Side Units (RSUs) can be deployed but are often regarded as too costly, while cellular infrastructure is already available. Furthermore, research shows that IEEE 802.11p can easily be congested, rapidly suffering from a high packet loss rate due to collisions.

The deployment of an improved traffic management and the implementation of the *Local Dynamic Map* (LDM) [16] require constant provision of information about the position and speed of every vehicle. Autonomous vehicles will also need to be provided reliable data about their surroundings continuously. Passengers will probably demand to access news and position-based information. We can see that there is an important amount of local redundancy in both uplink and downlink. The lack of an aggregation mechanism would mean an important waste of radio resources that would result in a poorer network performance that would jeopardize the correct functioning of the system, and an individual expenditure on cellular quota consumption much higher than it is actually needed.

Regardless of the incorporation of machine-to-machine (M2M) communication in the development of 5G, it would be hard to imagine that M2M (or, in this case, V2V) communications in a short-range radio access technology can be billed the same way as the cellular communications that we know today. The difference, in economic terms, between both technologies, will not necessarily disappear through integration under the Fifth Generation umbrella.

On the other hand, any proposed mechanism for optimizing communications in hybrid vehicular networks, since it can affect the distribution of the economic cost of cellular access, must be analyzed from the point of view of *social acceptability*. This requires that the integration solution proposed must not only be efficient in terms of communication costs, but also fair, in terms of the distribution of these costs across participating vehicles, over the long term.

3 Proposed solution and main contributions

OBJECTIVES

- Provide a solution for integrating V2V and cellular networks, reducing cellular access costs and preserving network performance.
- Ensure that the proposed solution also provides fairness over time, for social acceptability reasons.

To capitalize on the advantages of each technology, we propose to enable vehicles to self-organise into clusters, via V2V communication. Within each cluster, data between all vehicles and the cluster head is communicated via V2V, and between the cluster head and the targeted Internet server via the cellular network (in aggregate form). To be more specific, a cluster is composed by a group of Cluster Members (CM) and a Cluster Head (CH). In a multi-hop cluster, CMs are able to reach the CH potentially using other CMs acting as relay nodes along several hops.

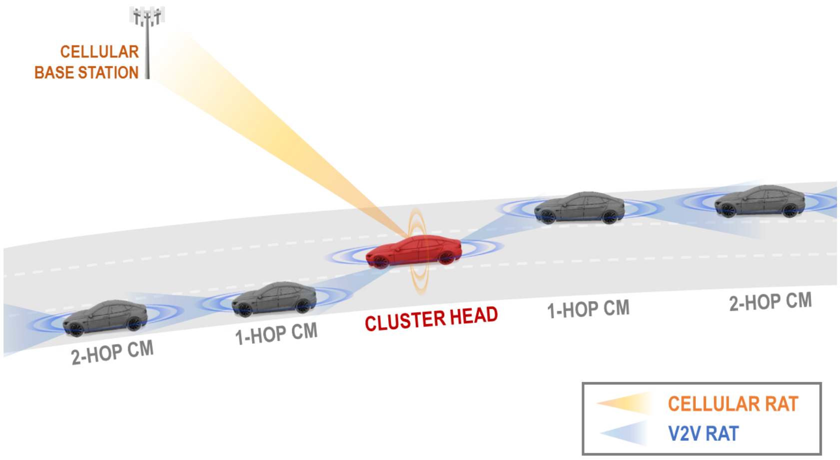


Figure 1.1: Illustration of a multi-hop cluster in a hybrid vehicular network.

In our model, we consider a scenario where vehicles have to periodically send data to the cellular network on the uplink, e.g., Floating Car Data for traffic management purposes. A reduction of the uplink cellular traffic can be achieved **by aggregating and compressing this information at the CH, which is the only node using the cellular connection**. CHs can also locally broadcast information received on the cellular downlink, through the V2V network (such as local data requested to a Geographic Information System). This contribution will be detailed in Chapter 5.

However, a trade-off arises when establishing the cluster size in terms of number of hops. On the one hand, the larger the clusters the higher the data compression ratio at the CH and the smaller the amount of required resource on the cellular link is. On the other hand, packet loss ratio (PLR) becomes higher for larger clusters, due to increasing communication collisions on the V2V channel. The size of a cluster – i.e. the number of vehicles within the cluster – depends directly of the traffic density and of the number of clustering hops. Hence, we can adjust cluster sizes by dynamically adapting the number of hops of self-organising clusters to traffic density.

We propose a self-adaptive multi-hop clustering algorithm for hybrid vehicular networks that locally and dynamically changes the maximum number of cluster hops (thus increasing or

decreasing cluster sizes) in order to maximize compression of the volume of data exchanged with the cellular infrastructure, while keeping the V2V packet loss rate below the maximum tolerable threshold.

The proposed *Self-Adaptive Clustering Algorithm* (detailed in Chapter 6) dynamically changes cluster sizes (by adapting the maximum number of hops in function of the traffic density) with the objective of reducing the usage of the cellular network resource while maintaining the packet loss rate in the V2V network below a certain threshold. The algorithm performs the cluster head selection and hop adaptation in the base stations of the cellular network. For an improved geographical distribution of Cluster Heads, allowing clusters to enlarge their number of members, the Cluster Head selection is delegated to the cellular base station. Using an ITS simulation framework, we show that the Self-Adaptive Algorithm correctly adapts to extreme traffic density changes, significantly reducing cellular network usage (thus making important monetary savings at large scale), while respecting the imposed constraints of packet loss rate on the V2V network (in order to comply with latency or reliability requirements for different applications and services).

While the proposed method improves network performance, it also raises the problem of the distribution of the communication costs borne by the CH. In clustering algorithms where the Cluster Head (CH) acts as a gateway to the internet, being selected as CH (the only vehicle in a cluster which uses the cellular network in this model) is a significant burden. This is because each vehicle is linked to an individual's account with a mobile operator, with a limited traffic quota. Hence, the vehicle that is elected as CH is the only one that consumes its traffic quota for the benefit of all the other vehicles in the cluster. It becomes evident that improvements need to be made in order to ensure that, even though at a certain moment only one vehicle is a CH and will consume its quota, there *will* be a *fair* quota consumption across vehicles *over time*. The cellular quota consumption is visible to the drivers. If they perceive that it is consumed in an unfair way, they can, at any moment, leave the system.

To avoid this shortcoming, **we propose an approach that adopts the *theory of distributive justice* (Chapter 7) and applies it to the selection of the Cluster Head in clustering algorithms for a *fair* distribution of roles and cellular quota consumption over time.** In particular, we apply this approach to the proposed self-adaptive clustering algorithm (Chapter 6). At the same time, we believe that our approach is similarly applicable to other clustering algorithms.

Within the proposed approach for fairness management, we establish a correspondence between the variables in our clustering model and a set of *canons* of justice [65] (e.g. be treated as equals, or in function of their needs, their productivity, etc.). The vehicles that are in the process of forming a cluster cast their votes for their preferred and most convenient canon. The final selector of the Cluster Heads, which in our case is the cellular base station, takes all votes into account, together with the statistics (available quota, previous participations as CH, etc.) of each vehicle, in order to make the best decision for both network performance and fairness criteria. The proposed fairness-aware algorithm is compared, via simulation, with the one presented initially without considering the fairness factor (Chapter 6) with respect to both short-term network performance (for which that algorithm was designed) and long-term fairness metrics. The results show that the algorithm improves fairness criteria without adversely affecting network performance.

CONTRIBUTIONS SUMMARY

- A multi-hop clustering algorithm where the CH selection is delegated to the cellular base station for an improved cluster formation efficiency [ITSC'16].
- A multi-hop clustering algorithm with self-adaptation of the number of hops to traffic density [ITSC'17].
- A self-adaptive, multi-hop clustering algorithm which is also fairness-aware [$SASO^{ST}$'17].

The algorithms presented in this thesis represent an important reduction in cellular network traffic (up to 90% for the example application of Floating Car Data aggregation for traffic management) and number of simultaneous connections by attacking the local redundancy that is inherent to several vehicular network applications. This reduction of cellular traffic also means a reduction of economic costs of access. The economic benefits are distributed among users in a fair manner thanks to the implementation of the *theory of distributive justice*, significantly improving the system's *social acceptability*. Our proposal makes an efficient integration of two radio access technologies with very different but complementary characteristics. This type of integration is one of the main objectives for the development of 5G mobile networks. The improved efficiency in the utilization of network resources is crucial for managing the important volume of data and devices that these systems will serve. Our algorithms can contribute to the development of future work in this area.

PUBLICATIONS

ITSC'16 Julian Garbiso, Ada Diaconescu, Marceau Coupechoux and Bertrand Leroy. **Dynamic cluster size optimization in hybrid cellular-vehicular networks**. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 557-563. IEEE, 2016.

ITSC'17 Julian Garbiso, Ada Diaconescu, Marceau Coupechoux and Bertrand Leroy. **Auto-adaptive multi-hop clustering for hybrid cellular-vehicular networks**. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

$SASO^{ST}$'17 Julian Garbiso, Ada Diaconescu, Marceau Coupechoux, Jeremy Pitt and Bertrand Leroy. **Distributive justice for fair auto-adaptive clusters of connected vehicles**. In *Fifth International Workshop on Self-Adaptive and Self-Organising Socio-Technical Systems ($SASO^{ST}$ 2017)*. IEEE, 2017.

4 Overview

This thesis is organized as follows: we introduce the main concepts of vehicular ad-hoc networks and the standards and protocols involved in Chapter 2. We present the fundamentals and state of the art of clustering algorithms in Chapter 3. We describe the simulation framework we used for evaluating our algorithms in Chapter 4. We present the *Base Clustering Algorithm* in Chapter 5, the *Auto-adaptive Clustering Algorithm* in Chapter 6, and its subsequent improvement with the application of the *theory of distributive justice* in Chapter 7. Finally, we draw the conclusions and envisage future research directions in Chapter 8.

Chapter 2

Intelligent Transportation Systems - Vehicular Networks

1 Introduction

The services and applications of Intelligent Transportation Systems (ITS) were since the beginning oriented to reduce energy consumption, mitigate the emissions of greenhouse gases, optimize the usage of the road infrastructure and reduce the number of fatalities due to traffic accidents, among similar goals to improve mobility as we know it today. More recent research works in the domain go far beyond with the introduction of the first autonomous vehicles. This innovation can completely transform the driving experience, and pave the way for a much closer connection between the vehicle and its environment, while allowing drivers to divert their attention to media and other services.

In this context, Vehicular Ad-Hoc Networks (VANETs) have a great challenge: they have to provide a very robust connection, while keeping in constant movement, and integrating different radio access technologies (RAT). The work presented in this thesis constitutes a contribution to improve the joint usage of different RATs in the context of hybrid vehicular networks. In order to proceed to the description of our proposal in the following chapters, we need to present some fundamental concepts about VANETs so that the reader can easily understand the environment, challenges and existing technologies.

This chapter is organized as follows: We describe the elements of a VANET in Section 2, we present the current standards in Section 3, and we describe the main categories of VANET applications in 4 before the conclusion in Section 5.

2 Vehicular Ad-hoc Networks (VANETs)

Vehicular networks are considered as a specialization of Mobile Ad-hoc Networks (MANETs), and their main difference is that nodes' movements are restricted to the limited mobility pattern of vehicles, in roads and lanes. In VANETs, every vehicle is defined as a node in the network and is equipped with a communications unit called an *On-Board Unit* (*OBU*) and an *Application Unit* (*AU*). The main function of the OBU is to exchange information with other vehicles or with access points which are placed by the side of the road. Those stationary access points are called *Road Side Units* (*RSU*). The term *AU* refers to any device that can serve to show information to the user (mainly the vehicle's on-board computer and the driver's smartphone) [51].

We can separate the elements that compose a VANET in three areas (see Figure 2.1)

- **In the vehicle**: The node's OBU and AU exchange information internally, either through wired or wireless connections. The OBU acts as a gateway, connecting this small sub-network to the rest of the VANET;

- **In the air:** The wireless connection between different nodes, or between nodes and the infrastructure, can use different parts of the radio spectrum, and a variety of protocols. The most well-known radio access technologies (RAT) are IEEE 802.11p for ad-hoc networking, and LTE for cellular communications. However, there are other examples that have been considered in the literature, such as WiFi, 3G cellular protocols, or WiMAX;

- **The infrastructure**: Access networks and equipment supporting access to the internet: RSUs and cellular base stations (BS) that connect the VANET to the backbone network.

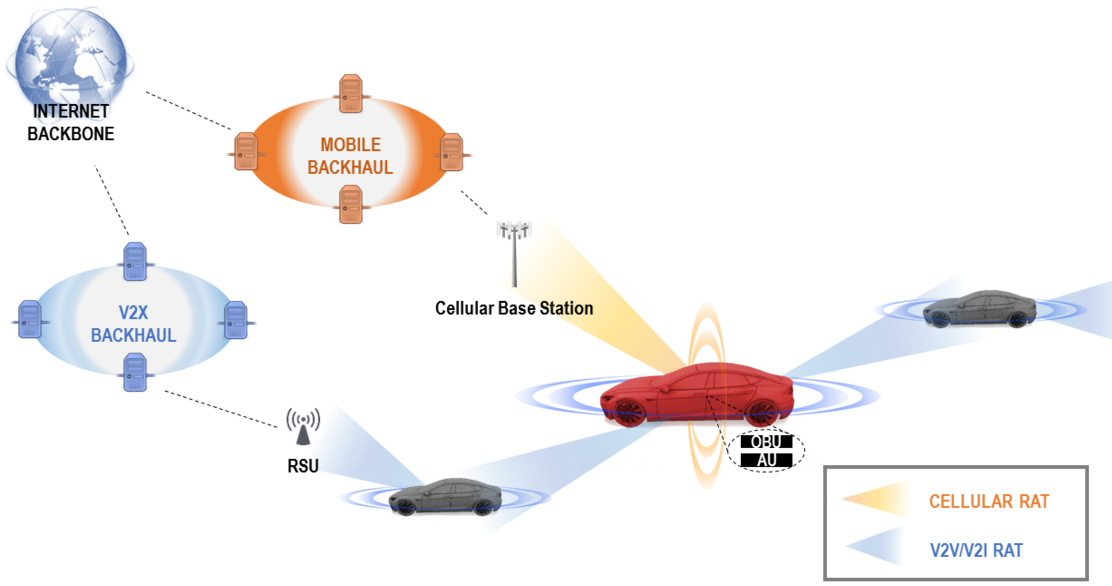


Figure 2.1: Illustration of the elements of a VANET and the links between them.

We can distinguish between two types of communication in VANETs: vehicle-to-vehicle (*V2V*) communication, where vehicles exchange messages directly, and vehicle-to-infrastructure (*V2I*) communications, where the information exchange takes place between the vehicle and the RSUs and access points (although the definition could be extended to cellular base stations too, we will keep the definition of V2I limited to non-cellular access technologies). Put together, V2V and V2I communications are known as *V2X* or *"vehicle-to-anything"* (recent developments also include vehicle-to-pedestrian communications, or V2P for short).

The main characteristics of V2X communications are:

- **Very dynamic topology**: the nature of vehicular mobility implies that nodes are constantly in movement, crossing each other, diverting directions, merging in routes, making V2X links sometimes very short-lived, making it hard to get snapshots of the network topology.

- **Radio signal interference and path loss**: due to the vehicles' speed, the characteristics of weather and environment, and the eventual presence of obstacles for wireless signals such as buildings or trees, communication signals can suffer from significant fading phenomena at higher intensity than other mobile networks.

- **Decentralized channel access**: every node is free to access the medium and transmit and receive packets, without influence or authorization of any central control. OBUs and RSUs handle their communications independently.

- **Unlimited energy supply**: nodes have no restriction in their energy consumption, since compared to a vehicle's battery (especially in the case of electric vehicles), it is negligible by several orders of magnitude for both OBU and AU.

- **High computational capacity**: the OBUs in the nodes have to handle different types of network traffic, like low-priority flows of high volumes of data (e.g. video streaming, if it were to use the V2V network) or high-priority flows of small amounts of data (e.g. road safety messages). Besides, they have to control routing and interfacing with RSUs. In hybrid (multi-RAT) vehicular networks, they also have to manage the coordination of the use of all the present radio access technologies.

3 Standardization

The functioning and operation of vehicular networks is bound to protocols, standards, architectures and technologies that are used for precessing information and satisfying the quality of service (QoS) demands imposed by different applications for this kind of networks. Those technologies are subject to modifications by the diverse standardization organizations worldwide, depending on each region's needs.

At a global scale, two of the most important research and development projects for standardization in VANETs are the American and the European proposals, which share some similarities, even though each of them has its own particular and distinguishable characteristics.

3.1 IEEE Standards - WAVE

The United States have been pioneers in the development of ITS, with many test projects and advancements made to this day. In 1999, the Federal Communications Commission (FCC) assigned a 75-MHz band in the 5,9 GHz frequency range for ITS services [3], named DSRC (for Dedicated Short-Range Communications). In 2002, the Intelligent Transportation Society of America recommended the adoption of a single standard for the physical (PHY) and medium access control (MAC) layers in VANETs.

Subsequently, the IEEE 802.11p Task Group was created by the end of 2004 with the aim of defining a communication architecture for vehicular environments based on the wireless local area networks of the IEEE 802.11 set of specifications [7].

In a second stage, the task group developed a set of specifications for the network, transport and application layers, that were grouped in the IEEE 1609 standards. The duo formed by the IEEE 802.11p and IEEE 1609 [13] standards is called WAVE (acronym for *Wireless Access in Vehicular Environments*). WAVE provides an architecture for V2X communications, designed for the use of road security and traffic efficiency applications. We can see the WAVE protocol stack in Figure 2.2.

	Layer	Name	IPv6 -WAVE		Pure WAVE
Security (IEEE 1609.2)	5	Application	HTTP and others		IEEE 1609.1
	4	Transport	TCP	UDP	WSMP (IEEE 1609.3)
	3	Network	IPv6		
	2	Link	IEEE 802.2 LLC		
			IEEE 1609.4		
	1	Physical	IEEE 802.11p		

Figure 2.2: The WAVE protocol stack.

3.1.1 IEEE 802.11p [7]

The IEEE 802.11p standard defines the characteristics of the physical and MAC layers, necessary for operating in a vehicular environment.

- **Physical layer (PHY)**: Just like the IEEE 802.11a protocol, IEEE 802.11p uses Orthogonal Frequency Division Multiplexing (OFDM). The bit rate varies between 3 and 27 Mbps, with a theoretical communication range under 1000 meters.

- **MAC layer**: The purpose of the MAC layer is to establish methods to access the communication channel, so that a set of stations can efficiently share the wireless medium. The IEEE 802.11p defines the use of *Carrier Sense Mode Access with Collision Avoidance* (CSMA/CA). The MAC layer also considers transmission aspects such as channel access time, congestion control and message prioritization.

3.1.2 IEEE 1609

The IEEE 1609 standard family defines the operational and management aspects of the network, transport and application layers of the WAVE architecture. We will now briefly describe each standard:

- **IEEE 1609.0** [13]: Defines the general WAVE architecture, the communication model, the access methods to the wireless media in a vehicular environment, the general structure of components such as OBUs and RSUs, and the WAVE interfaces.

- **IEEE 1609.4** [19]: Describes the multichannel operation to be implemented in the physical layer, including parameters for message prioritization, timers, channel switching and primitives for multichannel operation.

- **IEEE 1609.3** [6]: Describes the network layer services for vehicular environments and specifies routing and addressing functions, using two alternative stacks, one using TCP and UDP over IPv6, and another one using the Wave Short Message Protocol (WSMP).

- **IEEE 1609.2** [18]: Specifies the security services in WAVE systems, defines message formats and the procedures for processing them.

- **IEEE 1609.1** [4]: Describes the Resource Manager (RM) in WAVE systems, allowing an OBU with limited computational capacity to execute procedures remotely.

3.2 ETSI Standards

Europe has developed its owns approach for ITS, with similarities and differences with respect to its IEEE counterpart. The differences become more evident in the utilization of European protocols for the MAC layer, such as the Decentralized Congestion Control (DCC) [12]. This set of standards is jointly developed by many institutions like the COMeSafety project, the European Committee for Standardization (CEN), and the ETSI (European Telecommunications Standards Institute) TC ITS Working Groups.

3.2.1 Architecture

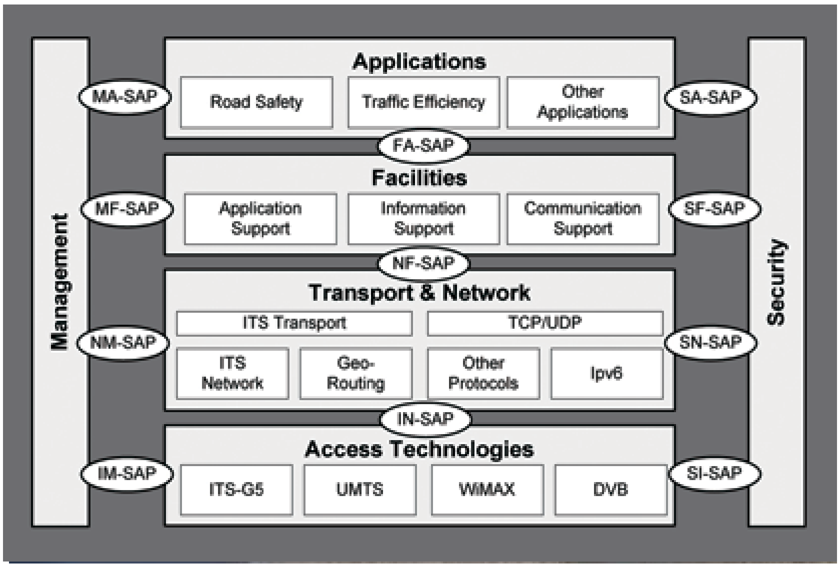


Figure 2.3: The ETSI ITS architecture. Source: [9]

Figure 2.3 shows the ETSI ITS communication architecture [9], which is composed of four horizontal layer, while the security and management layers are transversal.

- **ITS Security layer** [8]: a transversal layer that guarantees privacy protection for ITS applications, controlling authentication, confidentiality and information integrity.
- **ITS Management layer** [17]: a transversal layer that is responsible for the exchange of information between all the other layers of the ETSI ITS architecture in order to provide a functional integration of all the protocols and technologies that operate in a vehicular network. Among its main functionalities, we find the dynamic selection of the appropriate radio access technology for a specific application (the multi-RAT approach differs from the single-RAT proposal of WAVE), the control of the communication interfaces' parameters, management of transmission priorities and implementation of congestion control mechanisms.
- **ITS Applications layer** [5]: The European approach is more specific than its American counterpart with respect to application categorization, and it describes a Basic Set of Applications (BSA) defined in ETSI TR 102 638, classified in four big categories: Co-operative road safety, Co-operative traffic efficiency, Co-operative local services, and Global Internet Services.

- **ITS Facilities layer** [14]: the Facilities layer is tightly bound to the ITS Application layer, given the fact that it provides it with information and communication support. It takes charge of the maintenance and continuous update of all the information of the vehicular environment, including the Cooperative Awareness (CA) service through Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM).
- **ITS Transport and Network layer** [10, 11]: just like in WAVE, we have a dual stack. On one hand, one that provides support for internet connection through TCP/UDP over IPv6, and on the other hand, we find the GeoNetworking protocol, designed by ETSI, which provides single-hop and multi-hop routing.
- **ITS Access Technologies layer**: this layer is subdivided in its physical and MAC sub-layers. The native RAT is ETSI ITS G5, using Decentralized Congestion Control (DCC) in its MAC layer, and IEEE 802.11p for its physical and lower MAC layers. However, other radio access technologies, including cellular networks, are admitted in the Transport and Network layer of the ETSI ITS architecture.

4 Applications

VANETs provide an opportunity for the development of applications that can improve transport and traffic conditions through collaborative systems based on V2X communications. Depending on the functionality, vehicular applications are classified in three categories: road security, traffic efficiency and infotainment.

4.1 Road security

These applications' main function is to continuously monitor and collect information about the road and the vehicles in the environment in order to prevent accidents. They can be grouped in three sub-categories:

- **Collision avoidance:** sometimes relying on vehicle-to-vehicle communication, and sometimes by receiving a warning from a nearby RSU, some of these applications are intersection collision warning, alerts of vehicles driving in the wrong way, lane change warning, or hazardous driving alarm.
- **Road sign notification**: these applications have the functionality of preventing the drivers about the road signs and providing assistance.
- **Incident management**: these applications are used in emergency situations, providing services like emergency vehicle warning, or post-collision warning.

4.2 Traffic management

The purpose of these applications is to improve traffic conditions through monitoring and management of vehicular traffic and road conditions. The aim is to globally improve mobility and energy efficiency. According to their functionality, these applications can be classified in two sub-categories:

- **Traffic management**: these applications process information about the vehicular traffic flow and control infrastructure elements such as traffic lights and toll machines. Some examples of these applications are: intelligent traffic lights control, optimal speed advertisement, enhanced navigation systems and congested route notification.

- **Traffic monitoring**: they monitor vehicles and road conditions and notify drivers and authorities in case of irregularities. Some of these applications are: electronic license plate detection, vehicle tracking agent, and road condition monitoring.

Both traffic monitoring and management applications need a constant feed of information from the vehicles, or about the vehicles but obtained from infrastructure devices like cameras or radars. Connectivity from road to infrastructure is essential for this type of applications. Since these applications often analyze flows of vehicles rather than individual vehicles, aggregating this information can be useful for the network.

4.3 Infotainment

This type of application provides drivers and passengers with entertainment and information services. We understand as *entertainment* the applications that allow the occupants of the vehicle to use the network resources for recreational purposes such as accessing the internet, online gaming, or video streaming. On the other hand, *context information* includes sites of interest and local attractions, as well as other position-based services such as local map download or parking reservation.

5 Conclusion

We have presented the elements that constitute a VANET and the different technologies and standards that can be used for implementing and connecting them, as well as the early ideas for VANET applications that we can see in the literature. Regardless of the eventual choices that will be made during the implementation of VANETs in the real world, we can easily foresee that important volumes of information will come from vehicular network to servers in the internet, creating a very precise virtual image of the roads in real time. The different radio access technologies have, each of them, its own constraints and limitations. For instance, IEEE 802.11p is prone to saturation, while cellular access is paid. In this work we will propose clustering algorithms that can find optimal compromises to make good use of the available resources according to the demands of the future applications for vehicular networks.

Chapter 3

Clustering algorithms – State of the art

The core of this thesis consists of a clustering algorithm (presented in three versions that follow its evolution) for a smart integration of different radio technologies in a vehicular network. In this chapter we will provide the reader with the necessary information to understand clustering algorithms, and the state of the art in the area will be presented, from the historical background, to the specifics of clustering for VANETs.

1 Introduction

In the field of telecommunications, the usage of clustering techniques emerged decades ago as a solution for organizing *wireless ad-hoc networks* (WANET, or MANET for *Mobile* ones, where nodes are free to move). These networks are called *ad-hoc* because they do not rely on a pre-existing infrastructure. All nodes are equal in principle and there is no *a priori* organization. There are no routers or access points. Each node can, however, broadcast information and/or forward packets from origin to destination. The network has to implement self-configuration algorithms in order to serve a purpose, and clustering is one of the available tools.

Some of the first deployment scenarios were military battlefield communications in infrastructure-deprived areas, with the earliest implementations in the 1970s by the United States Defense Advanced Research Projects Agency (DARPA), based on the so-called *packet radios*, even serving as an motivation for the development of the Internet Protocol (IP).

The DARPA launched the Survivable Radio Network (SURAN) project in 1983 in order to develop radio routers for mobile ad-hoc networks. The program aimed to develop portable, low-cost and low-power hardware that could support radio packets that were improved with respect to their 1970s counterparts, while developing algorithms for scaling up to tens of thousands of nodes and being able to resist sophisticated electronic attacks [26].

The bottleneck of ad-hoc network was the complexity of the routing mechanisms for path selection when the number of nodes begins to rise. Since they can freely enter and leave the network (due to loss of signal or voluntarily), the constant update of routing information becomes very costly in computational terms when the size of the network grows.

Here is where clustering becomes an interesting strategy. It consists in creating virtual groups of nodes (**cluster members**, or CMs for short), with one leader known as the **Cluster Head** (CH). From the perspective of routing techniques, these groups become an individual entity, significantly simplifying the routing paths calculation and reducing the dynamics of path recalculations due to arrivals and departures, thus increasing network stability.

In 1987, the DARPA's Low-cost Packet Radio (LPR) program set the goal of the incorporation of dynamic clustering for improving scalability, as well as increasing security by adding spreading codes. The resulting prototype, the VRC-99 radio, was used for experimentation in the 1990s.

Back in the 1990s, the very few implementations in real life scenarios, limited to military operations and some early wireless sensor networks (WSN), along with the technological limitations of the time in terms of miniaturization, development of wireless standards, computational power and energy consumption, led to a loss of relevance of clustering techniques in the research community.

However, the mass market boom of wireless devices, especially since the appearance of the IEEE 802.11-based *WiFi* standards and *Bluetooth*, and the emergence of the paradigm of the *Internet of Things* (IoT), has created, in a very short time, numerous new scenarios where clustering can be used for cooperative communications, with different devices (like smartphones, wearables, or vehicular networks) and for different purposes.

The possible usages of clustering techniques are very diverse. For example, for **energy saving** [79]. This is achieved by combining different wireless protocols (hybrid networks) for short and long range communications. For instance, we could imagine a sensor network where nodes gather and share information by using Bluetooth Low Energy (BLE) and then use WiFi (which has a longer range, but a higher energy demand) for sharing this information with distant clusters. Another use for clustering in hybrid networks can be **coverage extension**, where cluster heads act as a gateway to a backbone network for cluster members that are out of the coverage area [22]. This chapter will provide a broad panoramic view of the different categories of clustering algorithms, and we will gradually converge to the part of the literature that is closest to our work.

The main contribution of this thesis is a **clustering algorithm for vehicular networks**, which integrates several aspects and approaches that we will explore in some areas of the clustering literature in this chapter, while addressing the specific problem of the distribution of the cellular access cost in hybrid vehicular networks. To the best of our knowledge, there is no clustering algorithm for vehicular networks that is designed and tested for this purpose (the most similar approach being LTE4V2X [64], commented on 4.6). Our proposal also uses the theory of distributive justice, as described in Chapter 7, which is also an unique approach for cluster head election.

The aim of this chapter is to provide the reader with a comprehensive understanding of the definitions, parameters and strategies of clustering algorithms, and the state of the art in the different application areas. This chapter is organized as follows: we present the basic concepts of clustering in Section 2; we will present the first clustering algorithms that introduced the main ideas that inspired most of the current state-of-the-art proposals in Section 3, and we make a detailed analysis of metrics, classification, scenarios and examples of clustering algorithms for vehicular networks in Section 4.

2 The basics of clustering

There are no strict definitions in clustering algorithms, and each one can present its own variations in its structure or procedures. However, there are certain trends that can be observed in the vast majority of the existing literature. The main principle is converting an disorganized set of nodes in a clustered network by assigning each of them a specific role. The basic set of roles consists of:

- **Cluster Head (CH)**: one single leader of a cluster, usually performing special functions such as information aggregation or distribution;
- **Cluster Members (CM)**: regular node that belongs to the cluster;

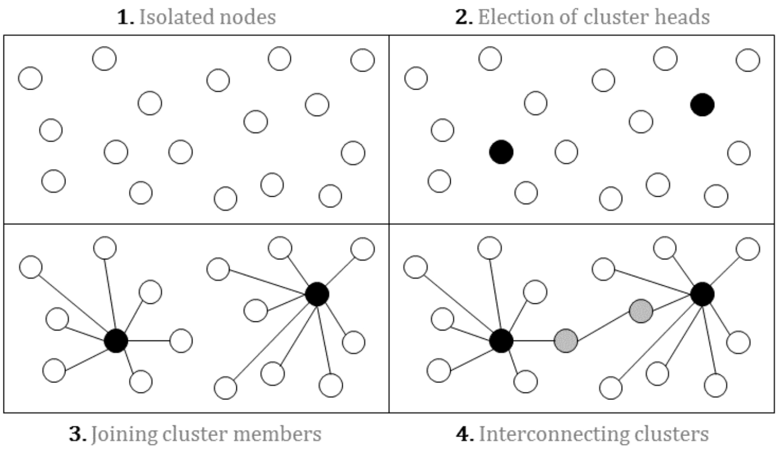


Figure 3.1: Illustration of the clustering formation process, from a flat to an organized network. Black circles: Cluster Heads; White circles: isolated nodes / Cluster Members; Grey circles: gateway nodes.

- **Gateway**: a CM that has the special responsibility of providing inter-cluster communication (the CH can take this role).

The usual cluster formation procedure, as depicted in Figure 3.1, is to elect the Cluster Heads in first place, according to a specific metric which is normally different for mostly every clustering algorithm, and then letting the isolated nodes associate to its nearest CH, becoming its Cluster Members, and eventually choosing a gateway (when it is not the CH) for communicating with other Cluster Heads.

For this process to take place, control information needs to be exchanged between nodes. In every algorithm, nodes transmit specific signalling and metrics. Some of the most usual choices are IDs, neighbour lists, speed and direction (for mobile networks), or battery level. The Cluster Head election will in most cases be a decision taken by the node itself, upon observation of its own metrics and the ones it receives from its neighbours. This step is a fundamental part of each clustering algorithm, differentiating one from the others. Different methods will be presented throughout the rest of this chapter.

The exchange of control information can take place with different degrees of synchronization (either tight synchronization, loose synchronization or no synchronization at all). The usual approach is the regular broadcast of beacons, commonly known as *Hello* packets, which can contain different types of information depending on the algorithm, but they will normally contain at least the node's ID, and its current position if the nodes are mobile.

2.1 Maintenance

Nodes can suddenly enter or leave the network for many reasons. In vehicular networks, simply starting a trip makes a new vehicle appear. The opposite can happen when a trip finishes, or in other networks, because of battery depletion, loss of signal coverage, or other causes. However, the clustered architecture must be resilient. In order to do so, maintenance methods are another important part of clustering algorithms. Different approaches exist:

- **Continuous maintenance**: This is the most common method, most suitable for highly dynamic networks, with frequent arrivals and departures. In this case, every node broadcasts *Hello* packets in a fixed, short period. Clusters can quickly respond and adapt to changes in the network.

- **Temporized maintenance**: Nodes cease to transmit control information after a cluster is formed. After the maintenance timer expires, the whole network re-runs the cluster formation procedure and a completely new clustered topology is created.
- **Event-driven maintenance**: Similar to the previous case, but the timer expiration can be replaced by any other event (or list of events or conditions) that triggers the cluster (re-)formation procedure.

Temporized and event-driven maintenance can significantly reduce overhead and energy consumption, but they are only suitable for stable or static networks. For ad-hoc networking in general, continuous maintenance is the most widely used approach. In some highly mobile networks, especially in vehicular networks, the continuous exchange of information can be mandatory for security reasons. This is why, in our approach, we take advantage of the mandatory *Cooperative Awareness* service (a continuous beaconing of every vehicle's position, speed, direction and other information known as Floating Car Data), for building our clusters.

2.2 Cluster size

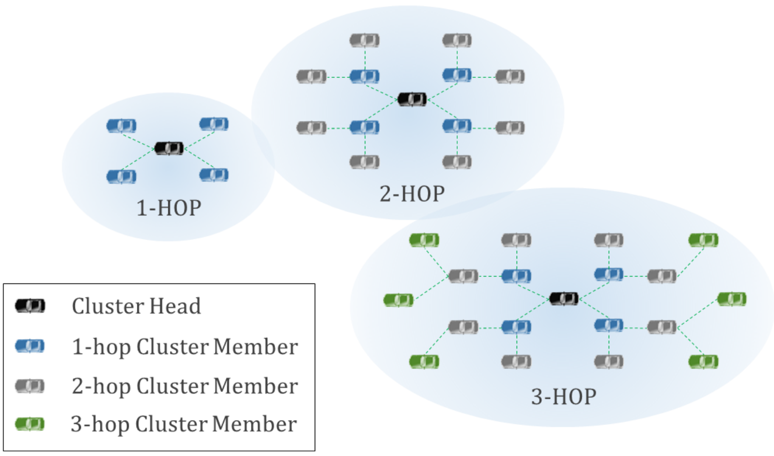


Figure 3.2: **Multi-hop clustering**: Packet forwarding allows for Cluster Members to join a cluster without direct communication with the Cluster Head.

Usually clusters are formed within the range of direct communication with the cluster head (which we call 1-hop communication). However, some algorithms, called **multi-hop clustering algorithms**, allow nodes which are outside the CH's communication range to join the cluster, usually through packet forwarding.

This technique can, if implemented well, increase the size of the clusters. This hypothesis will be analyzed in Chapter 5. From another point of view, this yields structures with a reduced number of clusters. The benefits of these characteristics will be discussed in the cited chapter. In our algorithm, this is used for increasing a cluster's potential for information aggregation and data compression. However, multi-hop clustering can significantly increase information overhead and energy use.

The state of the art of multi-hop clustering algorithms will be presented in Section 4.5.

2.3 Cluster head election

The role of the cluster head takes the responsibility of executing special tasks, including at least the management of intra- and inter-cluster communications, and other specific functions that can be needed in each specific clustering algorithm, like information aggregation (and gateway between different access networks) in the proposal presented on this thesis.

The very existence of the cluster head role creates an imbalance in energy consumption and, generally speaking, resource consumption between nodes. This is especially notorious in static networks, where the topology is fixed and the lifetime of a cluster is limited only by an eventual recreation of the clusters set by a maintenance timer. On the other hand, if the metrics for electing the cluster head depend exclusively on geographic position, the same node will always be chosen as CH. Clearly, mechanisms for rotating the cluster head role are needed. Intra-cluster rotation mechanisms for static networks are proposed in the LEACH (Low-Energy Adaptive Clustering Hierarchy) [47] and HEED (Hybrid Energy-Efficient Distributed clustering) algorithms [80].

It could be possible to think that mobile networks, such as vehicular networks, can take CH rotation almost for granted because the almost random topology variations obtained by combining mobility patterns, shorter cluster lifetime, merging, loss and arrival of new members, etc. However, as we will see in Chapter 7, this is not enough when it comes to cost distribution. In that chapter we will propose a method based in the theory of distributive justice, for a fair Cluster Head election.

2.4 Overlapping and interference

The simplest way to define cluster overlapping is for the case of 1-hop clusters: if a node is in the communication range of at least two cluster heads at the same time, the clusters leaded by these CHs overlap, and that node is in the intersection of those clusters.

We can extend the notion to multi-hop clusters by stating that for a k-hop cluster, a node that is at a k-hop distance or two or more cluster heads is in the intersection of those overlapping clusters.

Even though there are no thorough studies in the literature concerning the effects of cluster overlapping on the network performance, it is an indicator that the cluster heads may not be separated enough, or in other words, that there might be an excess of cluster heads, sometimes leading the network to underperformance. Imagine, for example, that we are implementing packet forwarding up to two hops, but every node is at 1-hop distance from a cluster head. Regardless of the effects on network performance, this would be undesirable since the excessive amount of cluster heads makes them redundant, preventing the clustering algorithm to attain its potential in terms of cluster size, losing the benefits it may yield. This is why most clustering algorithms provide mechanisms for avoiding cluster overlapping.

The clustering algorithm we present as a contribution in this thesis is designed for avoiding cluster overlapping and, moreover, aiming to form clusters that are as far-reaching as possible, maximizing their geographical extension to the limits of the communication range.

2.5 Centralization

So far we have seen that, normally, clustering algorithms perform a local auto-organization by exchanging information in an ad-hoc manner. In other words, they are *a priori* decentralized, and actually that is the case for the most part of the clustering algorithms in the literature.

However, total decentralization is not a mandatory requirement. There are several examples in the literature, like LTE4V2X [64], that rely on a local central controller, usually the cellular network, in order to increase efficiency in cluster formation, or to use specific metrics that could not be obtained in a totally decentralized algorithm, or at least not easily or efficiently.

2.6 Costs and benefits

The benefits of clustering algorithms depend at least partially on the scenario they are applied on, but generally speaking, we could name a few advantages that are tied to the most part of them. For example: they improve scalability of routing techniques and protocols, and improve coordination of the use of the radio spectrum. They create as well an excellent architecture for improving connectivity, linking gateways and distant nodes, and are perfectly suitable for information aggregation and data off-loading for **network traffic reduction and balancing**.

Delegating some degree of control on the cluster heads (CHs) allows us to have a **scalable architecture**, and deciding the appropriate gateways in order to **balance the load** between the different access networks in the case of hybrid networks, and even adapt the message frequency in each cluster according to its needs and possibilities.

Secondly, specifically in the field of Intelligent Transportation Systems where we are positioned, beaconing services and the information that is broadcast in them, just as they are foreseen in the current ITS standards, can lead to an incredible amount of redundant information. A cluster is the ideal structure to manage **information aggregation** [50], alleviating possible congestion on the gateways, and substantially improving the usage of radio resources.

Finally, and still taking case of vehicular networks as an example (although the idea behind is the same in other networks), the source or destination of condensed, relevant information is often remote, and it can only be joined via the cellular network or a road side unit. However, the coverage of the fixed stations (IEEE 802.11p road side units, cellular NodeBs and eNodeBs) is not literally ubiquitous, and it is frequent to find vehicles are out of coverage. Clustering techniques can extend the coverage and improve seamless connectivity to the internet.

Clustering, however, has certain costs. Yu et al. [81] summarize them as follows:

- Mantaining the structure in a dynamic scenario requires an amount of **exchanged messages** proportional to the underlying topology changes. Depending on the scenario and the characteristics of the algorithm, **bandwidth consumption** can cause a lack of resources that could even prevent the upper-layer applications from working properly, or even stop their execution completely.

- There are several examples of clustering algorithms in which a local change in the topology (for instance, the re-election of a cluster head) may lead to a reconstruction of a bigger zone of the network, affecting several clusters. This is called the **ripple effect re-clustering**.

- The **stationary assumption for cluster formation** is rather a defect than an actual cost. It refers to the inconsistencies that can be originated in the cluster formation process. Many clustering schemes are divided in two stages: cluster formation and cluster maintenance. In the former step, a node will be able to decide to become the cluster head after it exchanges a certain amount of metrics with *all* of its neighbours, and there is an implicit assumption that in this stage, the local topology will remain static (no nodes will leave or join the cluster in formation). This assumption, of course, will not always be true, especially for clustering algorithms with a long *convergence time*.

- The algorithms that need this *frozen period* that we discussed in the previous item can require a different number of **computation rounds** in order to complete the cluster formation. The more rounds of message exchanges the algorithm requires, the longer its convergence time is. When this happens, the hypothesis of a *frozen topology* covers a longer period. In other words, the computation rounds are a metric for the stationary assumption issue.

- Finally, the authors enumerate the algorithm's **communication complexity**, which could also be defined as a metric for the first item.

We could think about adding another cost: the algorithm's **computational complexity**, which is not enumerated in the cited article. Computational complexity is often ignored in the VANET (vehicular ad-hoc network) research literature because it is often assumed that vehicles have enough room and energy to host powerful computational equipment. However, embedded systems are normally built with the almost exact capacity to meet technical requirements in order to reduce costs, and this is particularly true in the automobile industry.

3 The timeline of clustering algorithms

In the history of clustering algorithms, there were a few initial proposals which were very influential at the time, and helped shape the theoretical basis of the domain. Even though they have all been outperformed by newer generations of clustering algorithms, with more complex techniques and/or focusing on more specific problems, mostly every new proposal is based on some parts of the set of rules and ideas that compose the legacy of these first algorithms.

We will now present a brief description of these algorithms since, because of their simplicity, they can help the reader understand the fundamentals of clustering in practice, for a better comprehension of the design choices we have made in our own proposal.

The *Lowest-ID algorithm* [37] published in 1987 was the first proposal for clustered networks. The idea behind it is very simple: every node is assigned an unique identifier (ID), which is the only available metric for CH election. The algorithm is executed in periods called *epochs*, and each of them is subdivided in two periods, one for transmitting and one for receiving control information, in a way that resembles *Time Division Multiple Access* (TDMA). A pre-requirement for this to work is that nodes are synchronized, in order for their transmission and reception periods to be the same, and in these periods, every node transmits in an orderly manner in function of its ID. Each node broadcasts its own ID, along with the list of nodes that it can hear.

- If a node only hears nodes with a higher ID than its own, then this node is a Cluster Head (this self-election is then announced);
- The lowest-ID node that one node can hear becomes its cluster head;
- If a node can hear two or more cluster heads becomes a *gateway node*.

The *Highest-Degree algorithm* [62] [42] gets to create larger clusters, thus reducing the number of clusters in the network, by changing the metric for cluster head election. While in Lowest-ID the metric had no correlation with any efficiency metric, analyzing the number of neighbors (*degree*) of a node, and picking those with the highest number of neighbors as Cluster Heads, certainly improves the results in terms of cluster size and robustness of the link between the CH and the other members. In this algorithm, nodes share their IDs, their list of neighbours and their current status during a broadcast period. At the end of this period every node knows who their neighbors are, and it also knows the degree of its neighbors. The rules are:

- A node that hasn't yet elected its "parent" Cluster Head is defined as an *uncovered node*.
- A node that has elected its Custer Head is a *covered node*.
- A node is (self-)elected as Cluster Head if it has the highest degree among its *uncovered* neighbors
- In case of a tie, the lowest ID criteria is adopted
- A *covered node* cannot become Cluster Head.

The *Least Cluster Change (LCC) algorithm* [31] proposed a new perspective for cluster maintenance for improving stability: an *event-driven maintenance*. Instead of periodically and routinely performing a reclustering operation, only two events can trigger a change of Cluster Head:

- When two Cluster Heads are in direct communication (overlapping)
- When a node becomes disconnected (not linked to any cluster)

This algorithm is actually an evolution of the Lowest-ID algorithm. Either *Lowest-ID* or *High Connectivity* are used to create the initial clusters, and then the LCC logic is used for micro-maintenance. This inspired future algorithms that nowadays avoid recreating the whole clustered topology, choosing to perform local micro-maintenance instead.

Last, but not least, the *Max-Min algorithm* [21] is the pioneer of *multi-hop clustering algorithms*, a fundamental technique which is at the heart of our contribution. In the previous algorithms, all cluster members were directly connected to a CH, and no more than two hops away from any other CM. The Max-Min algorithm forms *d-clusters*, where d is a fixed parameter that determines the maximum number of hops. The outcome is a set of clusters where every node either becomes a CH or is at most at a d-hop distance of its CH. It is another evolution of the previous algorithms, tracing back to the Lowest-ID techniques, where there are d rounds of flooding called *FloodMax* and another d rounds called *FloodMin*. In *FloodMax*, every node broadcasts the highest ID they have heard of (called *WINNER*). In *FloodMin*, every node broadcasts the smallest ID they have heard of. Every node keeps a log of its winners after each round. The rules for CH election are as follows

1. If a node receives its own ID during *FloodMin*, it elects itself as CH and does not continue to analyze the following rules.
2. The lowest ID which is present in both the *FloodMin* and *FloodMax* winners log is chosen as the node's cluster head.
3. If previous don't lead to the election of a CH, the node will choose the lowest ID in the *FloodMax* winner list as its cluster head.

After the publication of these pioneer algorithms, the research works in clustering began specializing in different aspects, leading to a wide variety of clustering algorithms that we will classify in the next section.

The algorithms we propose have a different approach for CH election: the partial delegation of the CH election to the local cellular base station results in a completely different scheme. An entity which is external to the cluster (the BS) selects cluster heads for their optimal geographical distribution according to the estimated (and dynamic) size of the clusters that best suits the current traffic density. In our fairness-aware approach, the election criteria also include metrics for social acceptability.

4 Clustering algorithms for vehicular networks

Clustering techniques are applied in multiple kinds of networks, and there are already multiple algorithms specifically designed for vehicular ad-hoc networks (VANETs) [78]. These algorithms usually take into account the specificities of vehicular movement (heading, relative speed, etc.) in order to improve the cluster's stability.

4.1 Metrics

The algorithms we propose have been carefully designed to reduce the negative effects cited in Section 2.6 as much as possible, and when facing unavoidable costs, an analysis of the possible consequences of the aforementioned effects in the specific case of the implementation of ITS applications and services has been made in order to find a compromise solution that fits our requirements. Finding compromises requires an evaluation and comparison of quantifiable and relevant properties. In [23] we can find an example list of the metrics that can be considered for the specific case of clustering in *vehicular* networks. These metrics are:

- *Vehicle density*: A critical parameter, some algorithms may perform better than others in low or high densities.
- *Vehicle speed*: It refers to the speed range for which the algorithm has been successfully tested.
- *Cluster stability*: The average life-time of a cluster.
- *Cluster dynamics*: Vehicles passing from one cluster to another, or cluster head changes.
- *Clustering convergence*: Refers to the time that the cluster formation phase takes.
- *Cluster connect time*: The time that one vehicle in particular stays connected to the same cluster.
- *Transmission dynamics*: Effectiveness of the algorithm concerning data dissemination inside the cluster.
- *Transmission overhead*: Simple: the lower, the better.

4.2 Theoretical performance limits

As we can see, one of the problems we have to avoid is to increase bandwidth consumption. In other words, to decrease the real throughput. Even in ideal conditions, a wireless network's throughput has scaling problems, often having an upper bound for its performance of $\Theta(\sqrt{n})$ [45]. The authors of [61] claimed that **hierarchical clustering** could improve the scaling performance to $\Theta(n^{h/(h+1)})$, with h levels of hierarchy. However, [43] adjusts this results, showing that there is actually a pre-constant in this scaling factor, making a complete expression that would look like $c(h)n^{h/(h+1)}$, where the pre-constant *c(h)* tends to zero as h goes to infinity. This means that, of course, hierarchical clustering cannot completely remove the scaling problem in wireless networks. Up to a certain number of hierarchy levels, this type of clustering could be considered. However, these articles rely mostly on fixed ad-hoc networks (like *sensor networks*). The combined MIMO and TDMA techniques for interference avoidance and inter-cluster communication that we see in [43] rely strongly on the control over signal fading and connectivity that comes from the fact of the nodes having a fixed position. In the VANET clustering bibliography, when the design of the clustering algorithms is discussed, the **partitioned clustering** (the non-hierarchical clusters we are used to see) is widely adopted [32].

4.3 Classification

Partitioned vs. hierarchical clustering is certainly not the only way to classify clustering algorithms. However, since this sort of algorithms apply to an extremely wide spectrum of scientific fields, and the first surveys of clustering algorithms have been published more than four decades before our work [46], we can deduce that there is no classification method that could be useful for them all.

The more general case we have considered in our literature study is MANET (mobile ad-hoc network) clustering, since its characteristics are very similar to those of vehicular networks. Yu et al.

[81] propose a classification of clustering algorithms for mobile ad-hoc networks that we can see in Figure 3.3. Their classification can provide a quick view of the first emergent branches of clustering when mobility was introduced as part of the problem.

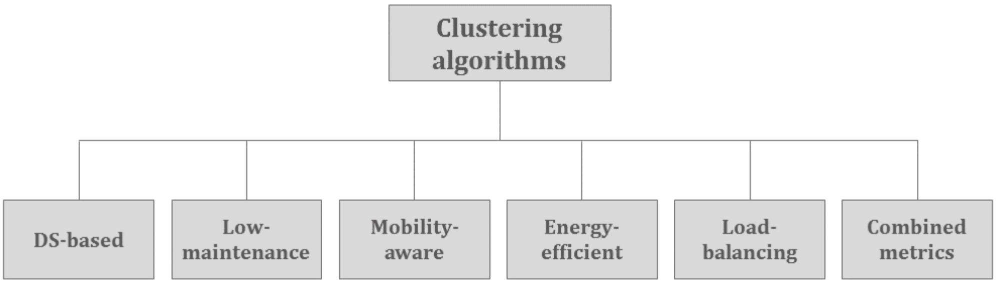


Figure 3.3: Classification of clustering algorithms for mobile ad-hoc networks proposed by Yu et al. [81]

The objective of DS *(dominant set)*-based clustering algorithms was to find connected dominating sets of nodes to simplify routing tables. Low-maintenance algorithms aimed to reduce topology changes and/or maintenance overhead, mainly by using underlying network traffic in order to adapt the topology without supplementary messages. Mobility-aware clustering refers to those algorithms that took into account the speed and direction of movement of nodes in order to group together those who had a similar mobility pattern. Energy-efficient techniques focused on reducing communications to the strict minimum in order to prolong the device's lifetime, a factor which is not as critical in VANETs (this is why we do not find an equivalent in Figure 3.4). Load-balancing algorithms tried to evenly distribute the number of cluster members in a certain region in clusters of similar size for a more efficient workload sharing. Combined-metrics clustering algorithms are, of course, according to this classification, those that mixed more than one of the aforementioned criteria.

The algorithms we have retained for further analysis fall specifically in the category of VANET clustering algorithm, conceived for the specific case of vehicular mobility and their specific quality of service constraints.

Bali et al. [23] propose the following classification scheme for VANET clustering algorithms (these categories are non-exclusive, some algorithms may fall into more than one of them):

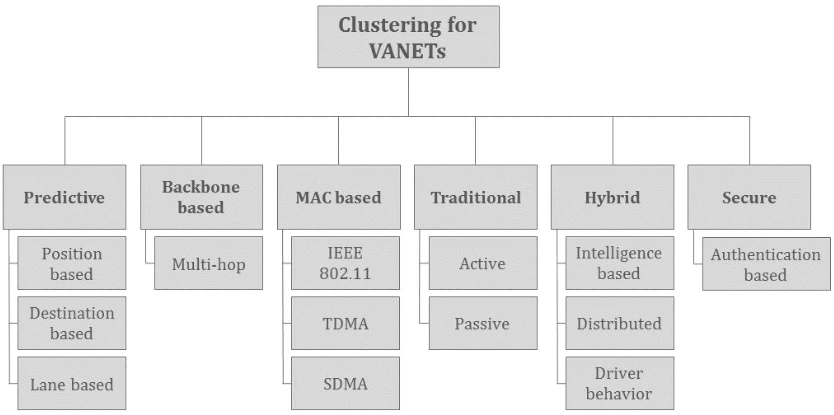


Figure 3.4: Classification of clustering approaches for VANETs according to Bali et al. [23]

- **Predictive clustering**: These algorithms are designed to optimize cluster stability and link lifetime by predicting and comparing the vehicles' movements. They can be sub-classified in:
  - Position-based clustering: The primary metric taken into account for cluster formation is the current geographical position.
  - Destination-based clustering: An improvement over the previous sub-type. These algorithms also take into account destination information in order to further improve the cluster's life-time.
  - Lane-based clustering: These algorithms assume the knowledge of the specific lane that the vehicle is taking (which is technically hard to achieve), getting performance improvements.
- **Back-bone based clustering**: Also called *k-hop clustering algorithms*. The communication distance between the CH and the members of the cluster is limited to a certain number of hops. This allows a cluster to be extended beyond the range of the CH.
- **MAC based clustering**: These algorithms implement clustering management at the media access control layer.
  - *IEEE 802.11 MAC*: Designed specifically for compliance with the 802.11 MAC.
  - *TDMA clustering*: Time slots are assigned to different vehicles in order to prevent collisions.
  - *SDMA clustering*: The road is divided into fixed segments, and each segment is subdivided into blocks. The vehicle(s) in the block have a specific time slot to communicate.
- **Traditional clustering**: Distinguishes *active* and *passive* clustering. The former case represents those algorithms where there are continuous updates of the cluster status information. In passive clustering, the algorithms do not generate overhead at all. They only use the information they can get from the ongoing traffic.
- **Hybrid clustering**: This category is for the clustering algorithms that are developed by using techniques from other domains (artificial intelligence, fuzzy logic, etc.).
- **Secure clustering**: Focused on implementing a Public Key Infrastructure (PKI).

In this classification, the algorithms proposed in this thesis fit mainly in the **multi-hop** subcategory, being also **position-based** for its cluster head election. If the simulation implementation is taken into account, it would also be **IEEE 802.11** compliant.

4.4 Scenarios

After analysing the existing clustering algorithms for VANETs, we see that if we make different assumptions concerning vehicular density and movement patterns, the most suitable algorithm according to the studied metrics will not always be the same, and we could imagine that, *a priori*, choosing one specific algorithm for every scenario could eventually prevent the network from meeting the minimum quality of service (QoS) requirements.

This is why we propose to create a system where the clustering rules change according to the detection of these basic scenarios:

- **Fluid traffic. Very low vehicular density**: The network is not saturated, and the vehicles could be too far from each other in order to form an ad-hoc network. It would be optimal to **deactivate clustering** in order to save the bandwidth consumption of the overhead generated by active clustering control messages. **Passive clustering** techniques could eventually be considered as a method for triggering the transition.

- **Fluid traffic. Regular vehicular density**: Considering the objective of extending connectivity by reaching vehicles that might be out of coverage, and the possibility of distributing the computational cost of data aggregation and reducing redundancy, data traffic and network usage in multiple steps, **k-hop clustering** is clearly the optimal solution for the normal traffic situations.
- **Fluid traffic. High vehicular density**: When vehicle density is too high, we have the highest risk of packet collision. In these extreme cases, the algorithms that perform the best are **TDMA clustering** algorithms, in which each vehicle is assigned a time slot by the cluster head. This does not necessarily mean that multi-hop clustering is not possible, TDMA is better for collision avoidance. Our approach was to use multi-hop clustering (adapting the number of hops dynamically) in order to have a coherent protocol that can swiftly adapt, instead of having to change protocols in function of the scenario.
- **Traffic jam (high density, stopped vehicles):** In the particular case of static vehicles (speed reaching zero), **k-hop clustering** accompanied with a drastic reduction of beaconing and a maximum of data aggregation could become a priority over TDMA.

4.5 Multi-hop vs. Single-hop clustering algorithms

Our interest is then focused on *multi-hop clustering algorithms for VANETs* [73, 53, 35]: by using message forwarding, the clusters can be larger than the communication range of the Cluster Head. Increasing the number of hops makes more intensive use of the radio resources, but can potentially increase the data aggregation ratio towards the cellular network. Besides, it improves connectivity in areas with light traffic, where vehicles are at a certain distance from one another. However, when there are too many vehicles, if the message forwarding is not controlled, the radio interface will be easily saturated.

Two representative multi-hop clustering algorithms for VANETs are the Criticality-Based Clustering Algorithm (CCA) and the Hierarchical Clustering Algorithm (HCA).

4.5.1 Criticality-Based Clustering Algorithm (CCA)

The authors of CCA [53] criticize the usage of Link Expiration Time (LET) as the single metric taken into account for the multi-hop cluster formation in many algorithms like [28], since it is only suitable for 1-hop clusters, and thus it limits the algorithm's performance. They propose to incorporate the usage of a new metric: **Network Criticality**. It measures the robustness of a network graph to changes in the environment (topology modifications, traffic shifts). It is based on the definition of *random-walk betweenness* in graphs. To simplify the idea, we could say that this concept is related to the *"importance"* of a node in the network, according to the amount of paths from a certain source to a certain destination that pass through it.

Computationally, network criticality calculation needs $\Theta(n^2)$ time, and thus, a powerful, centralized infrastructure. The authors proposed a new, *localized* concept of network criticality to implement a distributed algorithm that can provide a similar level of robustness.

CCA improves cluster life-time and shows a more stable multi-hop structure compared to other algorithms of its kind.

4.5.2 Hierarchical Clustering Algorithm (HCA)

The goal of HCA [34] is to create a clustering algorithm with a quick network setup. They have chosen to form 4-hop clusters (any node has at most two hops to traverse until it reaches the CH) because they are "sufficiently local" for safety applications and have a short setup time. It does not require GPS data. HCA creates hierarchical clusters with a poor internal implementation of TDMA, which leads to inter-cluster interference and collisions.

4.5.3 TDMA Clustering Algorithms

It is worth mentioning that there is a specific type of MAC-based clustering algorithms that make use of *Time Division Multiple Access (TDMA)* [20, 59] techniques. These algorithms assign a time slot to each Cluster Member (CM) in order to avoid collisions and hence ensure the arrival of messages. TDMA is technically *not impossible* in multi-hop clustering, but it can be so complex that it would seriously undermine performance and efficiency. We consider TDMA as a possible enhancement for single-hop clustering algorithms. Two of the most relevant examples in the literature are:

- **VeMAC**: Omar et al. [59] created VeMAC, a multichannel TDMA protocol capable of supporting one-hop and multi-hop communications, which is based on the ADHOC MAC protocol [29]. Every node is supposed to be equipped with GPS and two transceivers (one for the control channel, one for service channels). Each node will be able to access the Control Channel (CCH) once per frame. It is a broadcast-based protocol.

- **TC-MAC**: Conceived by Almalag et al. [20], TC-MAC propose a very straightforward approach to implement TDMA while guaranteeing to be compliant with the IEEE 802.11p PHY and MAC layers, which is critical for a clustering algorithm to be deployed in the ITS systems envisioned by the ETSI. The mechanism consists in using k service channels (SCHs), numbered from 0 to *k-1*, and a CCH that is considered to be the channel number k. Clusters are of size N. By using entire division, each node in the cluster is assigned a mini-slot in a specific SCH. The announcements and control information will be found on the CCH, which will be divided into k mini-slots, one for each channel, while each SCH will be divided into $\lfloor \frac{N}{k} \rfloor + 1$ mini-slots.

4.5.4 Multi-hop or Single-hop with TDMA?

In the absence of other control mechanisms, TDMA is a decent choice for scenarios with very dense traffic and a heavy use of the radio resources. Nevertheless, we have decided to focus on multi-hop algorithms because:

- Even if theoretically multi-hop and TDMA approaches are not incompatible, applying time division to a distributed network over radio interface seriously hinders its scalability, and would be especially inefficient when combined with multi-hop forwarding.

- Current ITS standards clearly determine the use of the CSMA/CA channel access control, which would be incompatible with real MAC-level TDMA. Implementing a pseudo-TDMA on superior layers would certainly be less efficient.

- Alternative approaches can be considered in order to reduce packet loss on the V2V link, but multi-hop clustering is strictly necessary for optimal data aggregation and the consequent cellular traffic reduction.

However, exploring the possibility to implement orthogonal multiple access for multi-hop clustering is an interesting question, left for further work.

4.6 Hybrid vehicular networks

The vast majority of the literature in this area has been conceived under the assumption that only V2V communication is available. Initial hypothesis supposed the deployment of numerous IEEE 802.11p Road Side Units, an assumption that is being left behind because of its deployment costs. As we have discussed before, the limitations of IEEE 802.11p and the high desirability of a connection to distant servers in order to deploy innovative services have obliged car manufacturers to include cellular connectivity in their new developments. Little research for improvements in clustering techniques has been done ever since taking into account this assumption. These works are being discussed in the following paragraphs.

In heterogeneous vehicular networks, we have at least two different communication technologies, usually a protocol for V2V communication and a cellular network. Most of the research in this area focuses on gateway selection algorithms (a gateway in this case would be a node that acts as a nexus between two networks). In [28], Benslimane et al. propose a clustering-based gateway management method between IEEE 802.11p and UMTS (3G). They focus on the selection of the best gateway candidates according to the UMTS Received Signal Strength (RSS). A similar work is carried by Zhioua et al. in [36], where they use fuzzy logic to select the best gateway node between a clustered VANET and LTE, with the novelty of considering traffic class as a priority in order to incorporate QoS constraints in the gateway election.

Another research direction focuses on the cluster formation problem. For example, [72] propose an interesting architecture intended to guarantee the arrival of Cooperative Awareness Messages (CAM), in the specific scenario of road intersections where vehicles approach out of the line of sight because of the buildings, sometimes even blocking radio waves. They create a *cluster region* around the intersection (the cluster regions are fixed). Every CAM message is transmitted via LTE. Once a vehicle enters a clustering region, it starts broadcasting beacons (specific to the clustering algorithm, not CAM) through the WiFi interface (the authors have chosen IEEE 802.11b instead of IEEE 802.11p because of its popularity and cost, but specify that they are interchangeable) in order to form a cluster. Once a cluster is formed, it is assigned a specific WiFi channel, and CHs are the only entities authorized to send CAM messages through LTE so that vehicles in different roads will be aware of the presence of vehicles near the intersection ahead. This considerably reduces the amount of CAM messages transmitted under the mentioned hypotheses, and proves to be a correct solution for the intersection problem. Yet, even though the architecture has a clever design, the application scenario is very limited due to the fixed nature of the *cluster regions* in a rather small area. There are other proposals for cluster formation designed for different goals or specific scenarios as the one presented above, see e.g. [67, 54].

The work of Rémy et al. in [64] is the closest reference to our work. The authors present an architecture called LTE4V2X, an approach which centralizes the clustering formation process, delegating it to an eNodeB (the base station of an LTE cellular network), in order to speed it up and save overhead traffic in the IEEE 802.11p spectrum. Furthermore, they implement an internal TDMA in each cluster in order to send the cooperative awareness information (position, velocity and heading). This last measure ensures collision avoidance. However, in the results shown by the authors, we see that even though the overhead traffic in IEEE 802.11p is considerably decreased, the LTE overhead raises dramatically, even compared to the overhead generated in IEEE 802.11p by DCP, a decentralized clustering protocol which does not make use of the cellular network. So we see that LTE is treated as an abundant resource, and the economic factor is ignored a priori.

To the best of our knowledge, no specific research has been done concerning the analysis of the problem of balancing cluster size in order to find equilibrium between IEEE 802.11p packet loss and cellular access costs. In Chapter 5, we try to make a clear analysis of this conflict that we consider to be critical since it compromises, on one hand, the correct functioning of the system, and on the other hand, its economic feasibility.

5 Conclusion

Clustering is a tool that has been studied for a long time, and that can take several forms, being constantly reinvented for being used in the most diverse contexts. It can offer scalability, improved connectivity, and a more efficient consumption of resources. On the other hand, those benefits come at a cost, usually in form of network overhead. We have identified a scenario in which clustering techniques are promising: the efficient integration of different radio access technologies in vehicular networks. We presented the reader the basic definitions of clustering, and a detailed state of the art in the field. We are now ready to introduce our proposal in the following chapters.

Chapter 4

Simulation framework

The clustering algorithms that we will present in Chapters 5, 6 and 7 are evaluated via simulation. In this chapter we are going to present the technical aspects from the simulation framework to the implementation details. In Section 1 we will introduce the Veins framework and their component simulators; in Section 3 we present the implemented protocol stack; Section 4 describes the path loss model for the radio signals; Section 5 contains the information relative to the maps used through our works and the methods of traffic generation and finally in Section 6 we describe the hardware setup used for running the simulations.

1 The Veins simulation framework

Simulating ITS scenarios with connected vehicles requires coupling mobility and networking. Veins [69] is an open source ITS simulation *framework*. This means that it is a tool which can serve as a basis for writing application-specific code on it, taking in charge the abstraction of the synchronization between the traffic and the network simulation.

We are now going to describe the two simulators used by Veins, the way they are connected, and the way we used and modified them.

1.1 SUMO: Urban mobility simulator

SUMO [27] is an open source (licensed under the GNU Public Licence) microscopic road traffic simulation package developed by the Institute of Transportation Systems at the German Aerospace Center. The microscopic simulation concept means that each vehicle is simulated separately, individually, with their own identifier, departure time and route. The simulation is done in time steps in the order of 100 ms, where each vehicle gets its position updated. The movement is thus continuous in space and discrete in time.

Some of the simulator's technical advantages compared to other software in the field are its high portability (with packages available for both Windows and Linux), the possibility to inter-operate with other applications at runtime, the exclusive use of standard C++ libraries, a lightweight and fast OpenGL graphical user interface, the easy interoperability and interpretation of its inputs and outputs which only use XML data, and its globally fast execution speed.

The details of the simulation go as far as multi-lane streets with lane changing, configurable traffic lights, person-based inter-modal trips, vehicle routing, different vehicle types, and the inclusion of several car-following models.

SUMO has the capability to import road network maps in different formats or from different platforms: VISUM, OpenDRIVE, Vissim, RoboCup, Shapefiles, OpenStreetMaps and MATsim, converting in this case the maps to the simulator's native map format. It also includes the NETEDIT tool that allows the user to create its own maps in the XML-based SUMO network format (.net.xml files).

Additionally, SUMO has a vast tool suite, and it is possible to control de simulation via a TCP socket by using the API called TraCI. This is the interface that the Veins framework uses to synchronize with SUMO. The position information of each vehicle is constantly retrieved, every single time step in the simulation, by OMNeT++ through SUMO's TraCI API, in order to update every node's position in the network simulation.

1.2 OMNeT++: Network simulator

The network simulation runs in OMNeT++ [77], with the incorporation of its INET Framework. OMNeT is an open source simulator where every simulated entity is composed by reusable modules coded in C++. The resulting relationship between these modules is declared in a Network Description (NED) file. These files can be created either by code or by using a graphical interface. For instance, when creating a *Connected Car* entity, we will create a NED file describing the network components of this object. In the case of a connected car in a hybrid vehicular network, the components would be the V2V and cellular Network Interface Controllers (**NIC**), the **mobility** type, and the **application layer** logic. Each of the components can be linked to another NED file which in turn can describe a more detailed structure. In other words, NED files can form a *hierarchy*. A NED file is normally associated with a C++ class describing its behaviour.

The simulator's environment is an adapted form of the Eclipse IDE and can be run interactively in a graphical interface or through command line. In our work, the first simulations have been run in the graphical environment in order to visually analyze the clusters' behaviour in order to identify aspects that could be improved. In order to do so, it is enough to include, in the C++ classes associated with the *Network Element* (in the NED file), the code that describes the graphical changes for the GUI in the corresponding syntax (which is specific to OMNeT++).

For all our simulations it was necessary to use the INET Framework, which consists of a set of modules that implement several internet protocols (such as TCP, UDP, IPv4 and ARP) and most importantly, it provides specific modules for the IEEE 802.11 radio transmissions between mobile nodes, which we are going to use for the implementation of the IEEE 802.11p V2V network connectivity.

Every node or element in the OMNeT++ simulation has a specific type of *mobility*, which is yet another element in the NED file, and its behaviour is described by the corresponding mobility C++ class. OMNeT++ includes several options from simple models like fixed, rectilinear or circular mobility, to Random Waypoint and mass-based mobility models. In the case of our simulations, the specific type is **TraCI Mobility**. This class is configured for constantly updating the node's position by retrieving the position of the corresponding car in the SUMO simulation through a socket connected to the TraCI API.

2 Scenario elements

A scenario of our simulation in the Veins framework is an extension (in the sense of code *inheritance*) of the basic Veins *Scenario* NED. We name it **Clustering Scenario**. This NED contains the following elements (as illustrated in Figure 4.1):

- **Connected Car**: They are not part of the NED descriptor, but are inserted in the Scenario in runtime. This module is an incorporation we did to the code in order to represent vehicles with both V2V and cellular access.

- **Cellular Base Station**: Implements the code of the clustering algorithms proposed in our work from the cellular network side.
- **Obstacle Control**: Processes the interference that the obstacles (buildings modelled in the map) produce in the radio signal.
- **Annotation Manager**: Processes the commands sent during runtime for displaying in the graphical interface.
- **Connection Manager**: Calculates the interference distance based on the antenna's power, wavelength, path-loss coefficients and minimal power threshold for correct reception.
- **Base World Utility**: Provides basic information of the simulation space and individual *air frames* (packets transmitted through radio waves).
- **TraCI Scenario Manager**: Provides synchronization with the SUMO simulation through the TraCI API, moving the nodes.

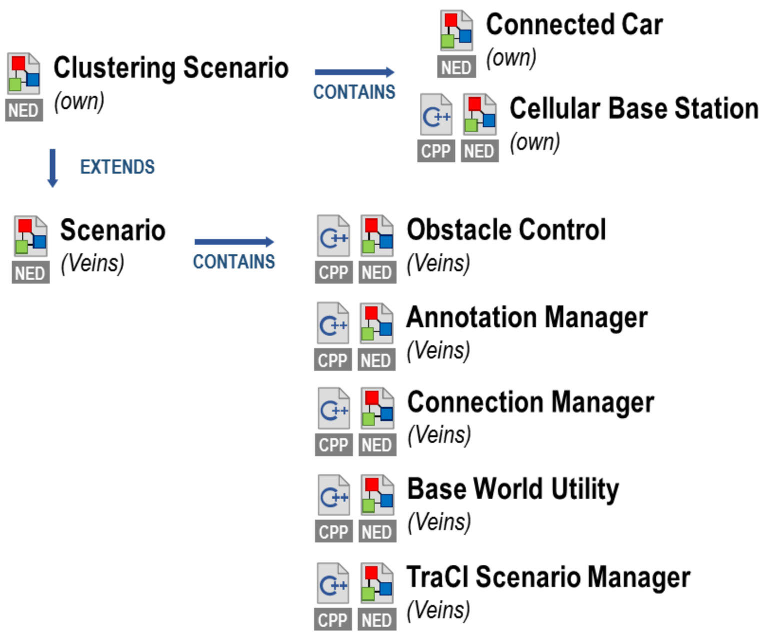


Figure 4.1: Diagram of the relationships between NED files and C++ classes directly linked to the main scenario.

The **Connected Car** NED is in turn composed by the following elements:

- **TraCI Mobility**: Links this vehicle's movement to a specific car in the SUMO simulation through the *TraCI Scenario Manager*.
- **V2V Network Interface Controller (NIC)**: Interfaces to the simulation code of the physical and MAC communication protocols taking place in the V2V network card. The protocols implemented in the simulations run in this work are specified in Section 3.

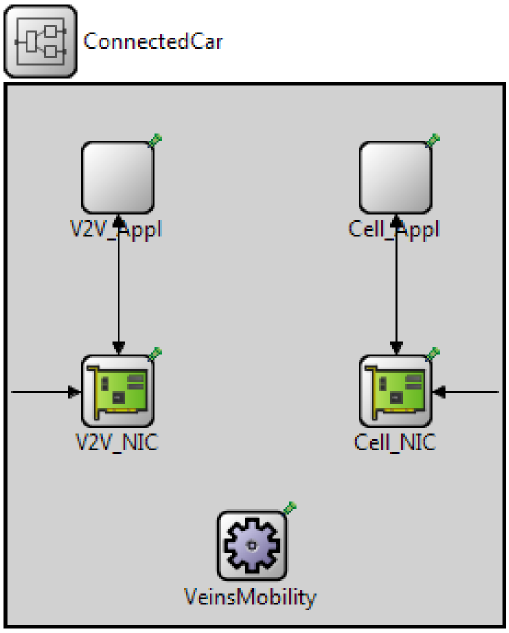


Figure 4.2: Visualization of the components of a Connected Car NED file.

- **V2V Application Layer**: Interfaces to the code of the *Single Hybrid Application Layer* where the code of the proposed clustering algorithms is implemented.
- **Cellular Network Interface Controller (NIC)**: Interfaces to the simulation code of the cellular network access. See implementation details in Section 3.
- **Cellular Application Layer**: Interfaces to the code of the *Single Hybrid Application Layer* where the code of the proposed clustering algorithms is implemented.

The design choice of using interfaces for the PHY and MAC layers of both access networks is useful for eventually changing the underlying protocols, showing that the algorithms we will present through this work are truly protocol-agnostic.

The NED of the **Cellular Base Station** (Figure 4.3) includes a *fixed* mobility type, and the interfaces to the cellular NIC and the application layer, which implements the code of the clustering algorithms from the cellular side.

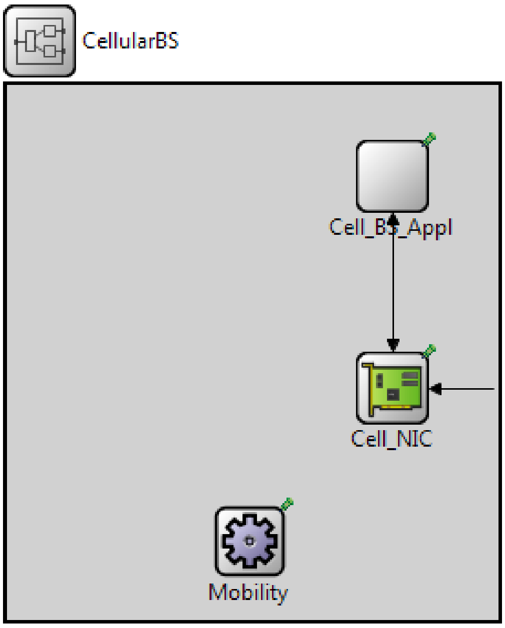


Figure 4.3: Visualization of the components of a Cellular Base Station NED file.

3 Protocol stack

The actual implementation of standards in the simulations that were run during this work was done as follows:

- **V2V NIC**: The physical (PHY) layer implements IEEE 802.11p, the protocol used in both the ETSI ITS standards and the American Wireless Access for Vehicular Environments (WAVE) standards. The MAC layer implements IEEE 1609.4, a standard for multichannel operation.

- **V2V Application Layer**: Together with the clustering algorithms, we implement a simplified version of the Cooperative Awareness service of the ETSI ITS standards, since our algorithms are based on the usage of the CAM messages.

- **Cellular NIC**: After analysing the possibility of making a full-stack simulation of an LTE cellular network by using the framework's extension *VeinsLTE*, we came to the conclusion that it was too costly in computation time while for the number of vehicles and exchanged data, the network performance would not be seriously affected. The model implemented then is perfect coverage with a perfect channel, simply adding a delay corresponding to the average observed delays of LTE[1].

4 Path loss and interference model

We use the simple free space path loss model where the signal attenuation in free space, L (in dB), is calculated in function of the wavelength λ, distance d and path loss exponent α as follows [68]:

[1]LTE latency is roughly estimated at 80 ms.

IEEE 802.11p Network Card Configuration	
Carrier frequency	5.890 GHz
Channel bandwidth	10 MHz
Maximum transmission power	20 mW
Bitrate	18 Mbps
Sensitivity	-89 dBm

Table 4.1: Parameters used for all the V2V network cards

$$L_{freespace}[dB] = 10log\left(\frac{16\pi^2}{\lambda^2}d^\alpha\right)$$

For the obstacle shadowing caused by the buildings, we have also used the method suggested by the authors of the Veins framework [68], who extend the regular path loss calculation by adding the following term:

$$L_{obs}[dB] = \beta n + \gamma d_m$$

This term is applied for the obstacles in the line of sight, and represents the additional attenuation that the signal suffers when it intersects n times the building's border and traverses d_m meters inside it. The two factors, $\beta[dB/wall]$ and $\gamma[dB/m]$ have been obtained by the authors of [68] by calibrating them by contrasting with experimental results obtained with real vehicles equipped with IEEE 802.11p antennas.

The obstacle shadowing model has only been used in the first test simulations presented in Chapter 5 in the Erlangen map. The rest of the test maps in this work do not have buildings.

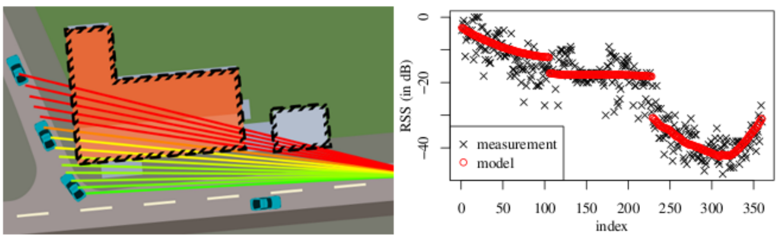


Figure 4.4: Evolution of the received signal strength: results obtained with the chosen path loss and obstacle shadowing models compared to experimental measurements. Source: [68]

The channel model may have an influence on the network performance and in particular on the relationship between packet loss ratio (PLR) and the number of hops. A deeper study of this factor is left for future work.

5 Maps and vehicular flow generation

The early tests described in the beginning of Chapter 5 used the map of the city of Erlangen, Germany, that is included in the Veins framework's example files. For the rest of the simulations, we made use of NETEDIT, a tool in the SUMO suite, that provides a graphical interface for creating SUMO maps. This way we created maps consisting of three-lane, one-way highway segments, of a length of 5 km in Chapter 5 and 10 km in Chapters 6 and 7. In all of these cases, we wanted to study the variations in the system performance in function of vehicular density. This is why we generated flows of vehicles with a fixed inter-arrival time. Subsequent flows can have different inter-arrival times, producing variations in vehicular density.

6 Hardware setup and execution time

The simulations in Chapter 5 were run in two laptop computers, one with an Intel Core i5-4210U CPU and 8 GB of RAM, running Windows 7, and another one with an Intel Core i7-3630QM CPU and 8 GB of RAM, running Windows 10.

The simulations in Chapters 6 and 7 were run in virtual machines in an OpenStack cluster. Each virtual machine has 8 Virtual CPUs and 16 GB of RAM, running Ubuntu 16.04 LTS (*Xenial Xerus*).

A typical round of simulations (composed of several individual runs with density variation and their repetitions for statistical analysis) can take up to slightly more than two days in the virtual machines. A typical simulation would consist of 10 density variations for 1000 vehicles, with 24 repetitions.

A high traffic density (e.g. 70 vehicles/km) increases simulation time, so while a run in very low density (e.g. 5 vehicles/km) can take as little as 10-15 minutes, a run with extremely high density can take several hours. Each run can be executed in only one core. However, many runs can run in different cores in parallel. In the virtual machines, 8 runs were executed in parallel at any given moment.

7 Conclusion

We have presented the framework that we have used for our simulations, and described the implementation details: protocol stack, path loss model, vehicular flow generation, map generation, simulation configuration and hardware setup. This information is necessary for ensuring the reproducibility of our results, and can help the reader understand the level of detail in the simulations that lead to the figures presented in the results sections of the following chapters.

Chapter 5

Base Clustering Algorithm

This chapter presents the *Base Clustering Algorithm*, which was developed in an iterative fashion, based on experimental results obtained from each incremental version.

1 Motivation, objectives and contribution

Traffic management systems aim to make our everyday voyages safer and more efficient. They can do so by influencing drivers' behaviour in different ways: regulation of traffic light cycles, suggesting or imposing deviations, using dynamic message signs for recommendations, etc. These systems need as much information as possible about the status of traffic, in real time. Collecting the **Floating Car Data (FCD)** —fundamentally position, speed, direction and time— of every vehicle on the road is a much desirable functionality for Intelligent Transportation Systems. In order to do so efficiently, we propose to use multi-hop **clustering** as a means for **aggregating information**, taking advantage of the significant redundancies in the local information that tend to occur in vehicular networks.

However, there is a trade-off between information aggregation (improved by increasing the number of hops, which enlarges the clusters) and packet loss ratio in the V2V network that needs to be addressed.

Our **objectives** in this chapter are the following:

1. To design a multi-hop clustering algorithm for hybrid vehicular networks that is suitable for information aggregation, reducing the cellular network usage and its consequent cost. The number of hops is a static parameter.

2. To ensure that allowing a higher number of hops effectively reduces the cellular network usage.

3. To characterize the relationships between the *number of hops* in the cluster formation process and the *size* of the resulting cluster, and between *number of hops* and *vehicle-to-vehicle packet loss ratio* for different vehicular densities.

4. To determine, in light of the results of the previous item, if there are ranges of vehicular densities (*traffic scenarios*) for which the V2V network performance degrades to a point that it would be necessary to limit the number of hops.

In line with the previously stated objectives, the reader will find the following **contributions** in this chapter:

- Detailed simulation results and analysis of the correlation between, on the one hand, the number of hops, and, on the other hand, the *cluster size* and the *Packet Loss Ratio on the V2V link*.
- A proposal for clarification of the implementation of the VMaSC algorithm [73] in order to ensure reproducibility and comparability of the results.
- A multi-hop clustering algorithm (**Base Clustering Algorithm**) for hybrid vehicular networks, with a static but configurable number of hops, which improves the cluster size increment when increasing the number of hops with respect to previously existing clustering algorithms for vehicular networks.

The proposed clustering algorithms are a tool for information aggregation (from V2V to cellular) and off-loading (from cellular to V2V), using the CH as a gateway. In order to test our clustering algorithms (the *Base Clustering Algorithm* in this chapter, the *Auto-adaptive Clustering Algorithm* in Chapter 6 and the *Fair Auto-adaptive Clustering Algorithm* in chapter 6), we have consistently used the same example application: FCD aggregation (see Section 2).

This application is intended to create a virtual, real-time representation of the traffic in a centralized server of traffic management. Having these data in real time allows for an optimal traffic management at every scale, and can provide a perfect data input for machine learning algorithms in order to make optimal predictions and thus allowing for a better organization. FCD can also be used for obtaining detailed mobility statistics that can be very useful for metropolitan or regional economic analysis, infrastructure and public transportation planning, and investments.

This chapter is organized as follows: we formalize our problem in Section 2, we present two different cluster head selection algorithms for comparison in Section 3, we show the simulation results in Section 4 and we draw our conclusions in Section 5.

2 Problem formulation

The implementation of the FCD aggregation application is based on the idea that every vehicle has to constantly communicate its identification, position, speed and direction to the traffic management servers, through the cellular network. If every vehicle transmits its data individually, it will mean a very important number of connections in the cellular network. Creating clusters of vehicles for aggregating it, would reduce the number of connections and enable compression of the amount of data transmitted.

In our proposal, when a vehicle joins a cluster, it ceases transmitting its FCD individually. We take advantage of the existence (as established in the ETSI standards) of the Cooperative Awareness (CA) service. All the data that every vehicle is supposed to send to the servers through the cellular network in our example application, are present in the Cooperative Awareness Messages (CAM) that are mandatory and periodically sent through the V2V link. The Cluster Head listens to all the CAMs coming from its Cluster Members and aggregates this information taking for example the average value (other methods could be used in function of each application's requirements). When a vehicle joins a cluster, the CH sends a one-time notification to the cellular network to inform that this particular vehicle is present in its cluster from then on. The server keeps track of the vehicles present in a cluster without needing to be periodically notified. Only one notification for arrival and one notification of departure is enough. For the rest of the time, the CH simply aggregates the information of all its vehicles taking the average value of the speed and position metrics. This way, for each cluster, the traffic management server continuously receives the average position and speed of the vehicles in the cluster, allowing a reduction of number of connections and a compression of data transmission through the cellular network at a ratio of N to 1, where N is the number of members of the cluster.

We consider a vehicular network consisting of vehicles that circulate on a highway section of fixed length L_s and made of L lanes. On every lane vehicles arrive at periodic instants, every T seconds, at the beginning of the section. Vehicles circulate at constant speed until the end of the section where they leave the network. We assume that the whole highway section is covered by a single cellular BS towards which vehicles have to send information at a rate of λ packets/s.

Assuming a clustering algorithm is implemented, this traffic can be either directly transmitted to the BS or conveyed by a CH. When a vehicle belongs to a cluster of size 1 (i.e., it is isolated), it transmits its information to the BS using uplink cellular radio resources. When included in a cluster c of size $N_c > 1$, this traffic is sent to the CH using IEEE 802.11p protocol and the CH aggregates the information coming from CMs and sends the result to the BS.

The traffic generated by the cluster c for the destination BS is then $\eta(N_c)N_c\lambda$, where $\eta(N_c) \leq 1$ is a compression function performed by the CH that may be a decreasing function of N_c. Without loss of generality, we can assume that $\eta(1) = 1$. We define a *cluster partition* as a set of non-overlapping clusters that includes all the vehicles of the network. In the following, we will consider only cluster partitions with clusters having a maximum of H hops between any CM and its CH.

As a consequence, the total traffic generated by the vehicular network on the uplink of the cellular network for a given cluster partition $\mathscr{C}$ can be written as:

$$\Lambda(\mathscr{C}) = \sum_{c \in \mathscr{C}} \eta(N_c)N_c\lambda, \tag{5.1}$$

where $\mathscr{C}$ is the set of all clusters, and N_c is the number of vehicles in cluster c. We denote $N = \sum_c N_c$ the total number of vehicles in the network. We now define the global compression ratio of the clustering partition $\mathscr{C}$ as:

$$\alpha(\mathscr{C}) \triangleq 1 - \frac{\Lambda(\mathscr{C})}{N\lambda} = 1 - \frac{\sum_{c \in \mathscr{C}} \eta(N_c)N_c}{N} \tag{5.2}$$

Note that α is also the average compression ratio. For this cluster partition and the considered traffic model, we can compute a Packet Loss Rate $PLR(\mathscr{C}, \lambda)$, which is a function of the cluster partition and the amount of traffic.

Our problem is for a given traffic condition λ to maximize the average compression ratio under the constraint of an acceptable packet loss rate:

$$\max_{\mathscr{C}} \alpha(\mathscr{C}) \tag{5.3}$$

$$\text{s.t. } PLR(\mathscr{C}, \lambda) \leq PLR_{max}, \tag{5.4}$$

where PLR_{max} is an application specific constraint.

3 Cluster Head Selection Algorithms

Studying the literature, we found out that there are some ways in which clustering algorithms can be significantly different from each other. One of them is the Cluster Head selection mechanism. The first CH selection mechanism implemented for test purposes was based on the one presented in [73] for being representative of a trend seen in the literature. We will call it the **Cluster Head Self-Appointment**, because it is based on the idea of one CH auto-proclaiming itself when, after exchanging position and speed information with its neighbours, it *deduces* that it has the minimum relative speed with respect to all others.

Two different methods for Cluster Head selection are implemented and tested:

1. **Cluster Head Self-Appointment**, where CHs are self-selected by using a minimum relative speed metric;

2. **BS-based Cluster Head Selection**, our proposal and contribution, where the CH selection is delegated to the cellular BS.

The aim is to compare the performance of our proposal with a representative example of a common strategy for CH selection in the multi-hop clustering literature for VANETs.

3.1 Cluster Head Self-Appointment

In this first approach we have chosen to implement the Cluster Head selection method proposed in [73] for the VMaSC clustering algorithm.

This method makes use of the information collected during a fixed initial period: every vehicle locally stores speed and position of its neighbours, as well as the hop distance to them. If a vehicle detects that it has the lowest relative speed with respect to the set of vehicles it can see in its vicinity, it selects itself as CH and starts broadcasting *CH Advertisement* messages. Any vehicle wanting to join the cluster has to send a *Join Request* message, and the incorporation is only effective once it received a *Join Response* from either the CH (if directly connected) or its parent node.

The inputs of VMaSC's cluster head selection can be easily mapped to elements of our proposal. The position and speed information for neighbouring vehicles can be obtained from the *Neighbour Information Table* (*NIT*). The discovery messages beaconed by each vehicle can be replaced by the enhanced CAMs from our proposal, thus avoiding excessive overhead. *Join* messages and *CH Advertisements* work in the same way.

3.1.1 VMaSC State Diagram

The implementation of VMaSC also required the coding of its own states and transitions between them, different to the one in our proposal. It can be seen on Figure 5.1.

A new vehicle in the simulation starts in an **Initial** state where it will only receive beacons from other vehicles, gathering the necessary information for the calculation of the *CH Condition*, and broadcasting its own beacons. It will remain in this state during the time period specified as the *IN TIMER*.

Once that timer is elapsed, the vehicle passes to a **State Election** state. If the *CH Condition* is fulfilled, it will instantly switch to **CH** state and start broacasting its *CH Advertisement* beacons. Otherwise, if a vehicle in **State Election** state receives a *CH Advertisement*, it will switch to **CM** state. Preference will be given to the smallest possible number of connection hops until reaching the CH.

A Cluster Member will verify, following the *CM TIMER*, if it has received messages from its CH during that period. If it is not the case, it will jump back to **State Election** state.

A Cluster Head will verify, following the *CH TIMER*, if it has no CMs attached to it. If that is the case, it will jump back to **State Election** state.

3.1.2 CH Self-selection improvements

In order to ensure reproducibility of the results, we need to detail a few implementation decisions we took in order to cover some aspects that cannot be found in the referenced articles [73] and [74].

- There is a gap in the description of the VMaSC algorithm when the following situation happens: Vehicle A is a Cluster Head, and broadcasts a *CH Advertisement* message. Vehicle B is in **Initial** state and receives this packet, but does not request to join Vehicle A's cluster since it is not yet in **State Election** state. However, according to the generalities of the algorithm, it should forward

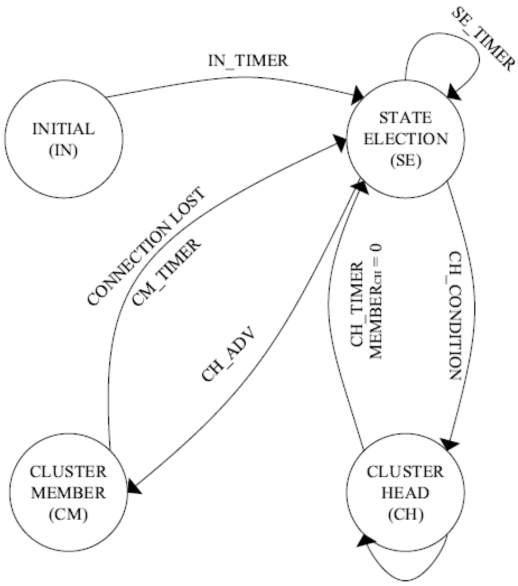


Figure 5.1: The VMaSC state machine. Source: [74]

the *CH Advertisement* message anyway. If Vehicle C is in **State Election** state and does receive the forwarded *CH Advertisement* from B, but it does not receive it from Vehicle A, then it will send a *Join Request* message to B, which is not ready to take it as a *first-hop child*. This would only slow down the whole process until Vehicle C realizes its *Join Request* message has not been answered (see next item). This is why we have decided to **stop forwarding *CH Advertisement* while the vehicle is in the Initial state**.

- Another detail that was not specified in the algorithm description was whether the *Join Request* message was exclusive or not (whether the vehicle should or should not wait for *Join Response* after sending a *Join Request* to one specific vehicle instead of being able to send multiple requests at once). We have decided to make it exclusive, since for every *Join Request* received (and accepted by sending a *Join Response*), the children counter is increased, and the children/neighbour purging algorithm would not even succeed to correct that. Further acknowledgement mechanisms would be necessary. Thus, after sending a *Join Request*, we set a timer (5 seconds in the first simulations) after which, in absence of *Join Response*, the vehicle is allowed to send a new *Join Request*.

3.2 BS-based Selection

3.2.1 Motivation

In the first stages of our work, all the building blocks of a multi-hop clustering algorithm (e.g. the telecommunications architecture, messages specification, and a state machine) were created.

During the first tests we came to the realization that the inherent characteristics of the CH selection algorithm significantly affected, in a negative way, the possibility of increasing cluster size when incrementing the maximum number of hops. By observing the details of several simulation scenarios, we confirmed that the amount of CHs that were being auto-proclaimed was excessive. The density of CHs in the traffic and the principle that vehicles would choose to join a CH which is at a minimum *hop distance* made that vehicles would always find a CH at one hop to join. Thus, increasing the maximum number of hops in the algorithm would have its consequences on network traffic, without getting any benefit in terms of cluster size.

This first conclusion inspired the idea for the proposed CH selection algorithm, which **delegates the CH appointment to the cellular base station**, with the rest of the **cluster formation process taking place in the V2V link**. This decision is based on the assumption that *future connected vehicles will most certainly be in constant communication with distant servers*, through the cellular network, for a variety of position-based services for the user, and for a more efficient traffic management. Being able to rely on the cellular network for *a local (in the coverage area of one base station) centralization of the Cluster Head selection allows for the creation of clusters of optimum size, which are better suited for information aggregation*.

3.2.2 Architecture for Cluster Head Selection

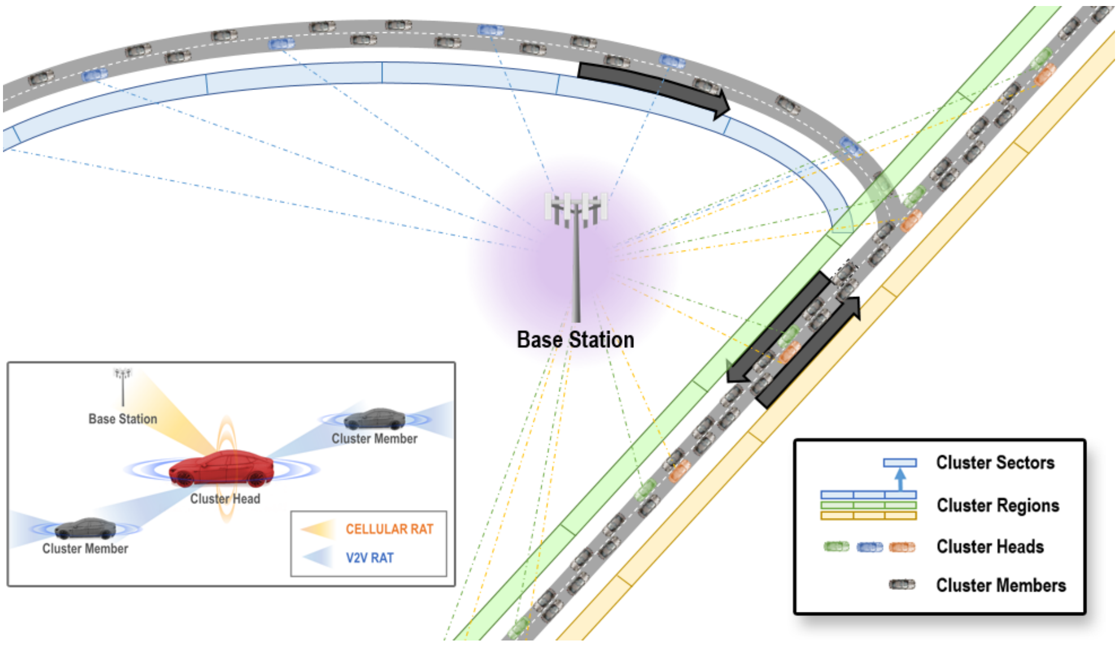


Figure 5.2: In the proposed architecture, the coverage area of a Base Station is divided in *clustering regions* (one for each road and direction), and each of them is divided in *clustering sectors*. Our clustering algorithm periodically verifies the presence of a CH in every clustering sector, selecting a new one if there is none. Each clustering sector's length is the product of the V2V communication range times the maximum number of hops.

In our system architecture, every vehicle is equipped with **two access network interfaces**: one for a *V2V* link, and one for the *cellular* network. In a **base station**'s *coverage area*, every road (in each direction separately) becomes a **clustering region**. Each region is divided in **clustering sectors** of the approximate size of a cluster (according to the maximum number of hops, which is *static* in this base algorithm —a fixed parameter—, and the V2V communication range). The size of a sector is actually calculated as the product of the V2V communication range times the maximum number of hops so that every sector approximately includes a single cluster (see Figure 5.2).

The base station is responsible for selecting the Cluster Heads. It verifies at regular intervals if there is a CH in every sector (since the BS continuously tracks every vehicle, whether it sends its position individually, or aggregated through the CH). If it is not the case, it selects one according to the algorithm presented in Section 3.2.2. The cluster formation process takes place locally, following the procedures and messages described in Section 3.2.4.

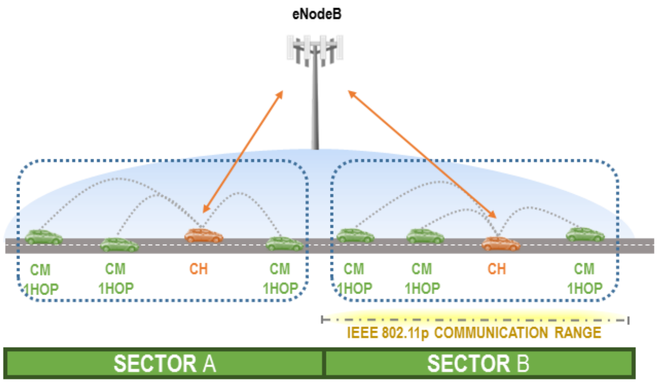


Figure 5.3: Schema of clusters, sectors and base station.

The delegation of the CH selection to the cellular base station was decided after observing that a major drawback of the *Cluster Head Self-Appointment* algorithm (see Section 3.1) is that it generates too many CHs. The delegated selection (see Algorithm 4) proves useful to overcome this issue. The main idea is to divide the highway section into segments, whose length depends on the IEEE 802.11p radio communication range and the maximum number of hops and select as CH in every segment the vehicle that is the closest to the center point of the segment. This process is updated every T seconds. This algorithm can be extended to a more complex map by dividing it into road sections in the same direction and covered by a single BS and applying the procedure described in Algorithm 4.

Algorithm 3 BS-based CH Selection Algorithm

Initialisation:
Set maintenance period T.
Set IEEE 802.11p radio range R, maximum number of hops H and compute the clustering diameter $D = 2R \times H$.
Divide the highway section into $S = L_s/D$ segments.
For $t = nT$, $n = 1, 2, ...,$ **do**
    For $s = 1, 2, ..., S$, **do**
        If there is no CH in s **then**
        Select as CH the vehicle that is the closest
        to the center of s.
        Endif
    Endfor
Endfor

This algorithm does not require constant cellular coverage, since in case of losing coverage selected CHs will remain in that state, and the cluster would not break, and vehicles would be able to join, leave or change clusters. However, it is difficult to predict how the quality of formed clusters would degrade with time after being disconnected from the cellular network. One possibility would be to let a cluster switch to CH self-appointment if cellular coverage is lost for a long time. This question is left for further studies.

Cluster handover process is trivial under current assumptions. The information of every vehicle is available at all times.

3.2.3 Vehicle State diagram

The state diagram in Figure 5.4 details the states and transitions of a vehicle in our model. When a new vehicle arrives, it will start in the **Discovery** state in order to listen for the best options (shortest number of hops) to reach a Cluster Head, and start building its *Neighbour Information Table* or **NIT** (See 3.2.4). Even upon reception of a *CH Advertisement* message, there is no possible direct transition to Cluster Member.

Upon expiration of the pre-set discovery timer (in the simulations it is set at 30 seconds), the vehicle passes to the **Isolated** state. If it has received *CH Advertisements*, it will choose the nearest CH in terms of number of hops. In the case where two or more CHs are detected at the same hop distance, the exact physical distance will be calculated using the information from the NIT, and the one at the smallest distance will be chosen. In the extremely unlikely case where two CHs would still be tied at this stage, the one with the lowest ID is chosen.

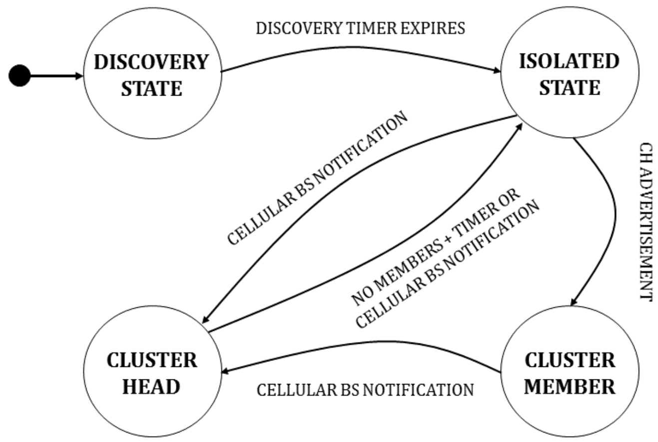


Figure 5.4: State diagram of a vehicle in our proposal.

If the vehicle has not received any *CH Advertisement*, it will wait in the **Isolated** state until it receives either a *CH Advertisement* or a notification to become CH from the cellular Base Station. A vehicle in the **Isolated** state which has not been able to join any cluster, will immediately try to join the first CH from which it receives a *CH Advertisement*.

Even if the transition from **Isolated** to **Cluster Member** is simplified in the figure as the reception of a *CH Advertisment*, in practice the transition happens upon exchange of *Join Request* and *Join Response* messages with the corresponding CH.

The decision of the appointment of a CH is completely delegated to the cellular BS. A vehicle in either the **Isolated** or **Cluster Member** state can be appointed as CH by the base station at any moment. Immediately upon reception of this notification, the vehicle will pass to the **Cluster Head** state, and start broadcasting *CH Advertisements*.

A vehicle can transition back from **Cluster Head** to **Isolated** state under two different conditions:

1. It has no cluster members during a certain time period; or

2. It receives a notification to cease being CH from the Base Station.

The BS can demand a vehicle to leave the CH state if two or more CHs are present in the same clustering sector.

3.2.4 Procedures and messages

We are now going to introduce the messages used by our clustering algorithm, explaining in detail the context in which they are generated, the procedures that take place upon their reception, their packet format and the data structures that are used in order to store and use the information they contain.

GeoNet packet structure

LLC / MAC	GeoNetworking	Security	GeoNetworking	Application	Security
headers	basic header	header	Common + Extended headers	payload	trailer

Figure 5.5: ETSI ITS secured GeoNet packet structure. CAMs and every other application messages presented in this section are located in the **Application payload** [10].

These messages are meant to be encapsulated in the V2V network protocol's packets. Taking as an example the ETSI ITS standards, in Figure 5.5 we show the structure of a GeoNet packet, highlighting the space for application payload, where the messages of our algorithm would be included.

It is important to take into account that the clustering algorithm we propose is completely agnostic of the V2V network protocol being used. As of 2017, different V2V standards such as ETSI ITS G5 and its American counterpart, WAVE, have similar performances and shortcomings.

Enhanced Cooperative Awareness Messages (CAMs)

Clustering algorithms, especially during the phase of cluster formation, require the exchange of information between the participants. This is usually the main source of control information overhead. A fundamental idea of our clustering algorithm is to avoid redundancy of control messages by taking advantage of the information that will be made available by services like the Cooperative Awareness (CA) service and the Local Dynamic Map (LDM) in the European ETSI ITS standards.

Specifically, in our simulated implementation, we use the Cooperative Awareness Messages (CAM) of the European ETSI ITS-G5 standard as a replacement for the otherwise needed regular beacons. Most of the needed information is indeed already transmitted periodically on CAMs:

- Sender ID
- Sender position
- Timestamp
- Heading (angle)
- Speed

We propose to add a few extra fields in the CAM messages in order to include the necessary information for the implementation of multi-hop forwarding:

- **Original sender state**: the state of the vehicle that actually emitted this message (Cluster Head, Cluster Member, Isolated).

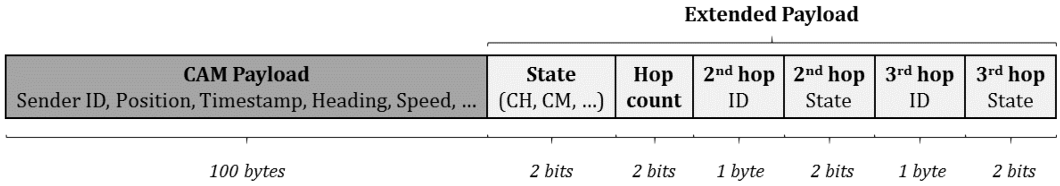


Figure 5.6: Enhanced CAM packet including extended payload for clustering information [15].

- **Hop count**: te number of hops that this message has passed by. The original sender sets it to 1. As long as this field is less than the maximum allowed, the vehicles getting this message will have to forward it, incrementing this counter by 1 and filling the corresponding fields:

- **Second hop ID**: a vehicle that forwards this message for the first time (i.e., the second hop), registers its ID in this field which is otherwise set to a value of -1;

- **Second hop state**: same as before, but for the state.

- **Third hop ID**: similar to the previous case, but for the third hop's ID.

- **Third hop state**: similar to the previous case, but for the third hop's state.

With this minimum incorporation to the CAM format (it represents roughly 3% of the regular CAM payload) we can implement the multi-hop forwarding of Cooperative Awareness messages, which we need in our example application for aggregating and uploading every vehicle's Floating Car Data, as well as for the implementation of the clustering formation of the algorithm we use for comparison in Section 3.1 (see **Note**). The only messages specific to the clustering process are occasional *CH Advertisement*, *Join Request* and *Join Response* messages.

Note

For the tests concerning the VMaSC algorithm (described in Section 3.1), these extra fields needed to be added in the Enhanced CAMs:

1. the individual calculation **by the original sender** of the minimum relative speed in **its** observable *k-hop* vicinity;

2. idem, for the second hop, calculated for **its own** *k-hop* vicinity (optional);

3. idem, for the third hop, calculated for **its own** *k-hop* vicinity (optional).

For this algorithm, the multi-hop forwarding of all this information in regular beacons is strictly necessary for the cluster formation process. Otherwise, there would be a lack of information for the CH selection.

Cluster Head discovery: CH Advertisement

When a vehicle is appointed as Cluster Head, it will start broadcasting these messages regularly. Only the CH selection needs the intervention of the cellular base station. The rest of the cluster formation process takes place in a completely decentralized manner, through *CH Advertisement* and *Join* messages.

CH ID	Hop count	2^{nd} hop ID	3^{rd} hop ID
1 byte	*1 byte*	*1 byte*	*1 byte*

Figure 5.7: Structure of the *CH Advertisement* message

In order to build multi-hop clusters, these messages have to be forwarded. This is why, in its basic structure (see Figure 5.7), besides the CH ID we also include the hop count, and the IDs of the second and third hop vehicles.

The CH will emit these messages with a hop count of 1 and it will assign a value of -1 to the second and third hop ID fields. When the message is rebroadcast, the hop count will be incremented an the second and third hop ID fields will be filled accordingly by each forwarding node.

The Neighbour Information Table (NIT)

With the information received from the multi-hop CAMs and *CH Advertisements*, every vehicle builds its own *Neighbour Information Table* (**NIT**), which is a proto-implementation of the notion of *Local Dynamic Map*. It is implemented in the C++ code as a **`map`** from the `<map>` library, associating every vehicle's ID with a **`NIT Element`**. Each of these elements contain the following information, which is updated constantly with every new beacon that arrives:

- **ID**: The neighbour's ID;
- **Parent CH ID**: The ID of the CH that this vehicle is attached to. It is equal to the **ID** field if the vehicle is a CH, and it has a value of -1 if the neighbour is not a Cluster Member or Cluster Head;
- **Message time**: The timestamp of the last received message (CH Advertisement or Enhanced CAM) received from this neighbour;
- **Direction**: The angle of movement of the vehicle (received in the last Enhanced CAM);
- **Speed**: The speed of the vehicle (received in the last Enhanced CAM);
- **State**: The state of the neighbour (Discovery, Isolated, Cluster Member or Cluster Head);
- **Hop counter**: The number of hops through which we are connected to this neighbour:
- **Second hop ID**: The ID of the second hop if the *hop count* is greater than 1 (otherwise, the value is -1);
- **Third hop ID**: The ID of the second hop if the *hop count* is greater than 2 (otherwise, the value is -1);

Please note that whether we receive the *NIT Element* directly from the source or through multiple hops, all the fields will remain exactly as they were emitted at the **source**, and the only changes when being forwarded are the modification of the hop counter, and the completion of the last two fields.

> **Note**
>
> **Implementation detail:** When receiving a CAM from a certain ID, if the neighbour's ID was already stored in the NIT, we will **ignore** every CAM that arrives from this ID with a **higher hop**

count than what is registered. If, for instance, *Vehicle A* was stored as a 1-hop neighbour but it goes further from us, and now the only way to communicate with it is at two hops, its CAMs will be ignored during the *freshness threshold*, after which it will be removed from the NIT and re-registered as a 2-hop neighbour upon reception of the next CAM.

When implementing the algorithm used for comparison in Section 3.1, the NIT and *NIT Elements* also include the *observed average relative speed* in a *k-hop* vicinity for: the source, the second hop vehicle, and the third hop vehicle.

Cluster Formation: the Join messages

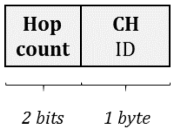


Figure 5.8: Structure of the *Join* messages

When a vehicle wants to join a cluster, a handshake takes place using **Join** messages, containing the concerned CH's ID and the hop count (as seen in Figure 5.8). Since these messages are intended to be *unicast*, the sender and recipient's IDs are not included in the figure because they are part of the packet header.

Cluster maintenance and lifetime: the freshness threshold

There is a regular procedure of verification of the *NIT*. A **freshness threshold** (measured in seconds) is set for the execution of the algorithm. At regular intervals, equal to the freshness threshold, a procedure verifies the timestamp of the last message received from every vehicle present registered in the NIT. If the difference between the current time and the timestamp exceeds the freshness threshold, the corresponding *NIT Element* is removed from the *NIT*.

If the removed element happens to be the vehicle's CH or its second or third hop vehicle for connecting to it (when the vehicle performing the freshness verification procedure is a Cluster Member), the vehicle goes back to the **Isolated** state.

4 Simulation results

4.1 Simulation settings

The base network traffic consists of the mandatory CAMs setting the frequency at the minimum value of 1 Hz (every vehicle emits, then, one CAM per second). These messages are the basic element of the Local Dynamic Map (LDM).

We assume an aggregation ratio $\eta(N_c) = 1/N_c$, where N_c is the cluster size. The length of the highway section is $L_s = 5$ km, and the number of lanes is $L = 3$. We will consider that, for the system to work in a secure and reliable manner, the packet loss rate on the V2V radio interface, PLR_{max}, cannot be higher than 10%. A total of $N = 60$ vehicles is simulated in each round. The average vehicle speed is 16.6 m/s. The vehicle inter-arrival distance varies between 1 s and 20 s. This is equivalent to say that we will analyze the variations of our metrics in function of the vehicle density,

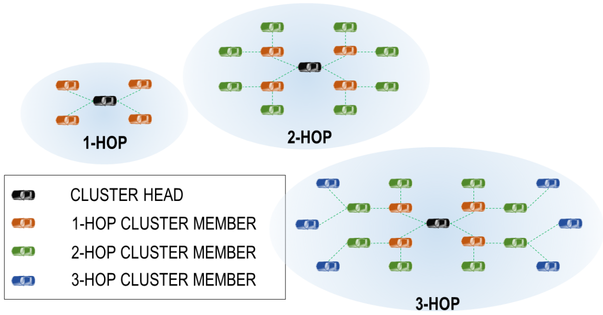


Figure 5.9: Illustration of the three examples of multi-hop clusters studied in our simulations, with the maximum number of hops varying from 1 to 3.

since increasing vehicle inter-arrival time implies reducing vehicle density and vice-versa. The reader should keep in mind this inversely proportional relationship.

The maximum number of hops is set to $H = 1$, 2 and 3 in separate simulation runs (See Figure 5.9), for each of both clustering algorithms (CH self-appointment and BS-based CH selection). We set T as the average duration for a vehicle to traverse the estimated cluster diameter D.

4.2 CH Self-selection improvements

The first step for evaluating the behaviour and performance of multi-hop clustering algorithms, was to take the example of an algorithm of the literature, in this case, VMaSC, to analyse its behaviour.

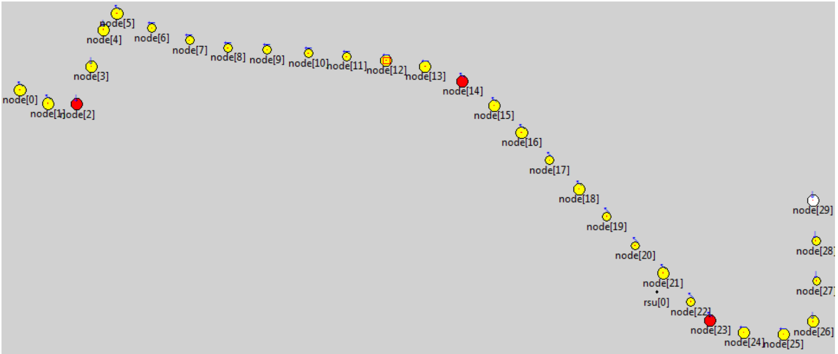
Those tests were run inserting flows of vehicles of different densities (different inter-arrival times) in the default map of the Veins framework (the Erlangen map). Some of the phenomena we will now detail, repeatedly observed in different runs, inspired the creation of the BS-based CH Selection of our proposal. We have set the maximum number of hops to three.

In Figure 5.10 we show three examples captured from the OMNeT++ Graphical User Interface. The color code (and icon size) differentiates the different states of the VMaSC state machine, and the number of hops that Cluster Members use for connecting to their chosen CH (see details in the figure caption).

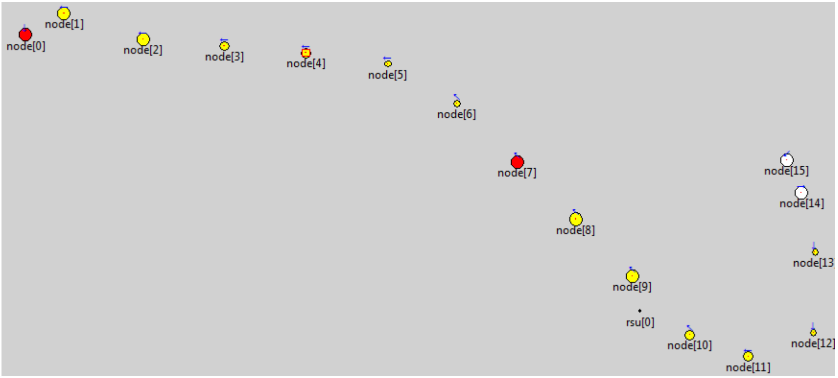
Figure 5.10a shows a case in which the algorithm behaves in a desirable way: there is enough space between Cluster Heads, and the number of hops used for connection smoothly follows the distance to the nearest Cluster Head. However, this behaviour was only observed for some combinations of densities and calibrations of the algorithm's multiple timers (mainly the *IN* and *SE* timers, which are the ones closely involved in the initial cluster formation process).

In Figure 5.10b we see one of the undesirable behaviours of VMaSC: the tendency to form row-like structures. In this case, the first vehicle auto-proclaims itself as CH, and the following vehicles join it. When a new CH is selected, there is a discontinuity: the vehicles that will come *after* it, will choose it as their CH. However, the vehicle that came *right before*, will be connected by two hops to the first Cluster Head. The result is that instead of having sequences like "*2-hop CMs, 1-hop CMs, CH, 1-hop CMs, 2-hop CMs*", the clusters are half this size, in a pattern of "*CH, 1-hop CMs, 2-hop CMs*".

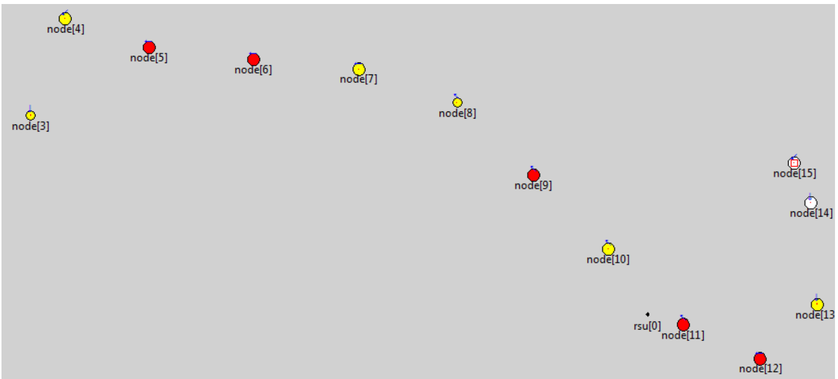
The cause of the regular selection of the first vehicle as a CH is a consequence of the influence of traffic dynamics in the relative speeds used for the calculation of the *CH Condition*. The discontinuity in the number of hops of Cluster Members can be understood if we consider that a new CH can appear at any moment, and if it happens when the concerned vehicle is still even in the 3-hop range of the previous CH, it will necessarily create this discontinuity since it will be, at best, in the far outreach



(a) **Normal behaviour:** Regularly spaced CHs and gradually changing number of hops.



(b) **Problematic behaviour example 1:** Tendency to form discontinuous rows. Cluster Heads are almost always in one extreme of the cluster, and there is a cut when the next CH is selected. As a result, one vehicle next to the second CH is connected using three hops to the first CH, when it could be directly connected to the second CH.



(c) **Problematic behaviour example 2:** Excess of Cluster Heads. A critical problem. It undermines the possibility of actually increasing cluster size when incrementing the number of hops.

Figure 5.10: **Cluster Head Self-Appointment**: Captures from the Graphical Interface in the preliminary tests, showing examples of particular behaviours observed. **Color and size reference:** *Red circle* for a CH, *Yellow circle* for CMs (regular size for 1 hop, small for 2 hops, tiny for 3 hops), *White circle* for vehicles in **Initial** state, *Green circle* for vehicles in **State Election** state.

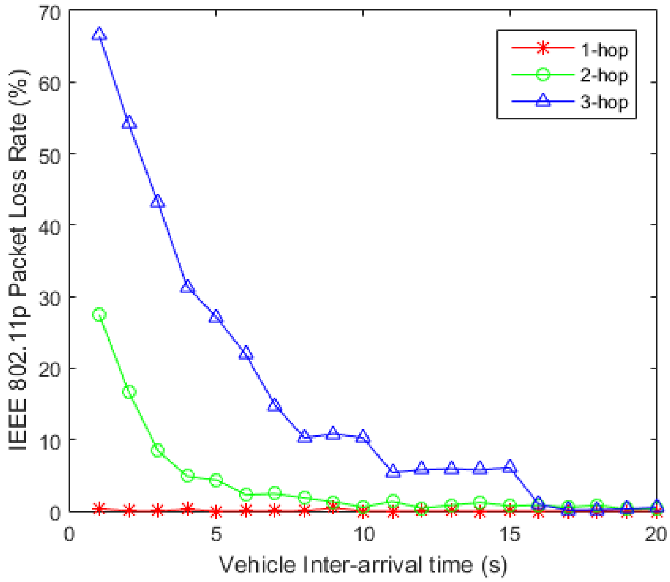


Figure 5.11: IEEE 802.11p packet loss rate as a function of the vehicular inter-arrival time, for different numbers of hops and the CH self-appointment algorithm.

of the previous CH, and on the other hand, the following vehicles will join the new CH because of its proximity. The combination of all of these issues, which are caused by the type of leader selection taking place, produce the *"half-clusters"* behaviour.

Finally, in Figure 5.10c we can observe the biggest issue found with the VMaSC approach. When analysing the initial results of increase in V2V packet loss and in cluster size when incrementing the maximum number of hops, we observed that cluster size only marginally increased, and sometimes it did not increase at all. The observation we have done with this ad-hoc modification of the GUI to trace the VMaSC algorithm states allowed us to identify cases like this one, in which an excessive number of Cluster Heads are (self-)appointed. Since the VMaSC algorithm also demands to join the nearest CH, increasing the number of hops activates the corresponding packet forwarding, thus increasing V2V packet losses significantly, but fails to deliver an increment in cluster size, which would allow for better cellular access cost reductions.

The phenomenon of the excess of Cluster Heads goes against the idea that allowing a higher number of hops increases the amount of vehicles in a cluster, which is the basis for building a clustering algorithm that efficiently reduces cellular traffic. The other observed issues can only worsen its outcome. After seeing the results of these simulations, we decided to propose as a solution to this problem one of the main characteristics of our approach: the delegation of the CH selection to the cellular base station, and more specifically, the notion of clustering sectors. These sectors' dimensions are determined by the V2V communication range and the number of hops, meaning that the clustering sector is as large as a cluster can be, at the corresponding maximum number of hops. Besides, the algorithm periodically verifies and makes sure that for each sector, there can be no less and no more than one Cluster Head. This way, we aim to have an adequate number of cluster heads: there must always be one in the multi-hop communication range, but there cannot be an excess of them.

4.3 Maximum number of hops vs. packet loss rate

In Figure 5.11, we show the packet loss rate as a function of the inter-arrival time for different maximum number of hops and for the CH self-appointment algorithm. Observed trends are similar for the BS-based CH selection algorithm. We can clearly see that the amount of lost packets grows drastically for high vehicular densities, and the situation gets much worse for every supplementary hop we allow. The reason lies in the broadcast storm effect arising in highly dense multi-hop networks when some packets are broadcast. This effect is amplified when these packets are rebroadcast over an increasing number of hops.

In this figure, we can see that there are some vehicular densities for which 2 or 3-hop clustering is incompatible with the requirements: below 10 arrivals per second (resp. 3 arrivals per second), the CH self-appointment algorithm with maximum 3 hops (resp. 2 hops) does not meet the packet loss rate constraint of 10%.

4.4 Maximum number of hops vs. cluster size

Figures 5.12 and 5.13 show box plots of the cluster sizes obtained, for different vehicular densities and number of hops, with the CH self-appointment algorithm and the BS-based CH selection algorithms respectively.

As expected, we can see that at intermediate to high inter-arrival times (i.e., at low to intermediate vehicle densities), as the the number hops increases the average cluster size increases as well. When the number of hops is high (especially in the case of 3-hop clusters) however, increasing the vehicle density has a contradictory effect. High vehicle density indeed leads to high packet losses and CHs cannot properly communicate with the potential CMs. This implies a decrease of the average cluster sizes. When the number of hops is small however, increasing the vehicle density also increases the average cluster size.

Our BS-based CH selection algorithm proves to perform much better in terms of cluster size, showing an unambiguous direct effect between increasing the number of hops, and increasing the cluster size. In the case of the CH self-appointment algorithm, since too many CHs are proclaimed, increasing the number of hops leads to little or no gain in terms of cluster size (and thus, presumably, in cellular traffic savings).

4.5 Global compression ratio

The evolution of α as a function of the vehicle inter-arrival time is expressed, in terms of percentage, in Figure 5.14 for both algorithms and for $H = 1$, 2 and 3 hops. From intermediate to high inter-arrival times (i.e., low to intermediate densities). Conclusions are clear: allowing more hops is the best strategy and our BS-based CH selection algorithm outperforms the CH self-appointment algorithm. This is due to the fact that the packet loss rate is maintained at an acceptable value, so that more hops allows for larger cluster sizes, which in turn improve the data compression ratio. As our algorithm creates less CHs, the data compression ratio is also improved. At low inter-arrival times however, the packet loss rate becomes unacceptable, so that large clusters fail to form. Performance with $H = 3$ thus become the worst for both algorithms. We however note that this phenomenon arises at a lower inter-arrival time with our algorithm, which illustrates the superiority of the BS-based CH selection algorithm.

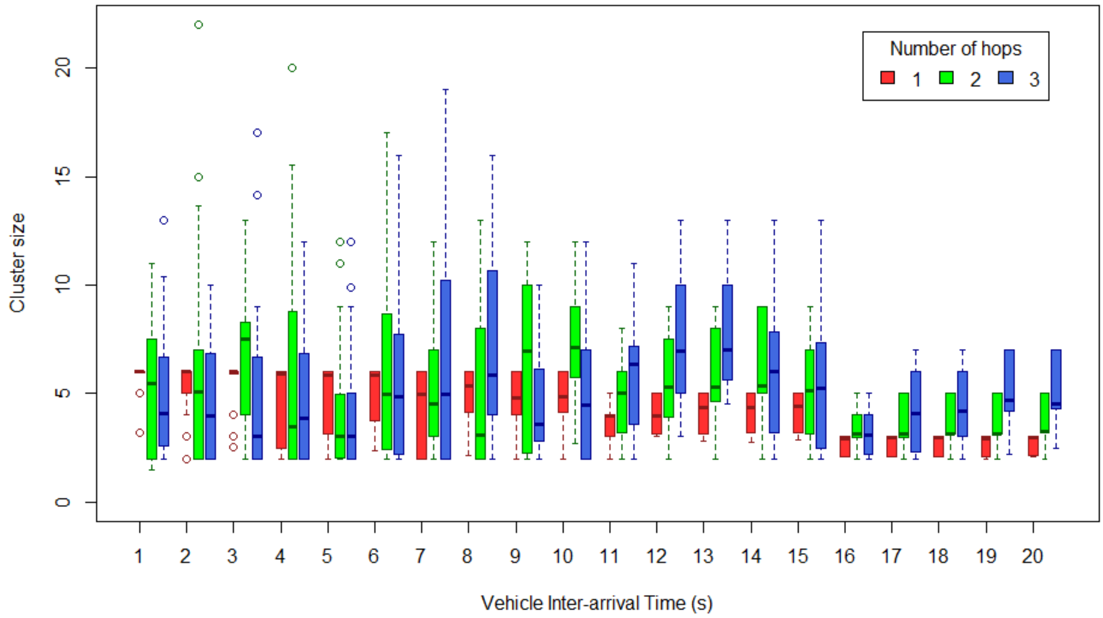


Figure 5.12: Box plot showing the sizes of the clusters formed by the CH self-appointment algorithm as a function of the vehicle inter-arrival time, for different maximum numbers of hops.

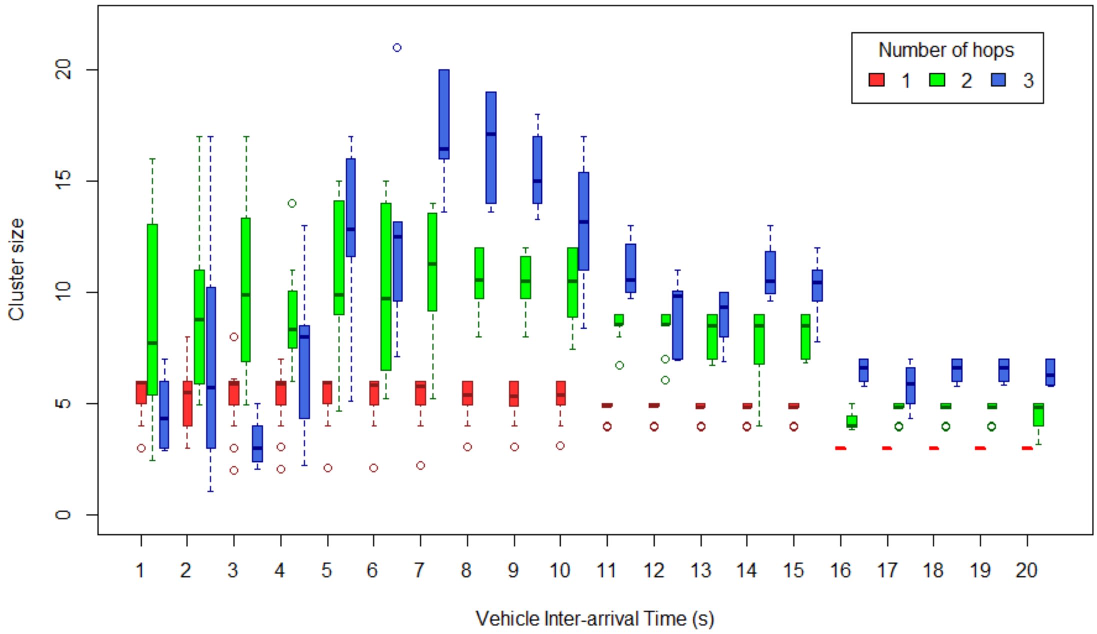


Figure 5.13: Box plot showing the sizes of the clusters formed by the BS-based CH selection algorithm as a function of the vehicle inter-arrival time, for different maximum numbers of hops.

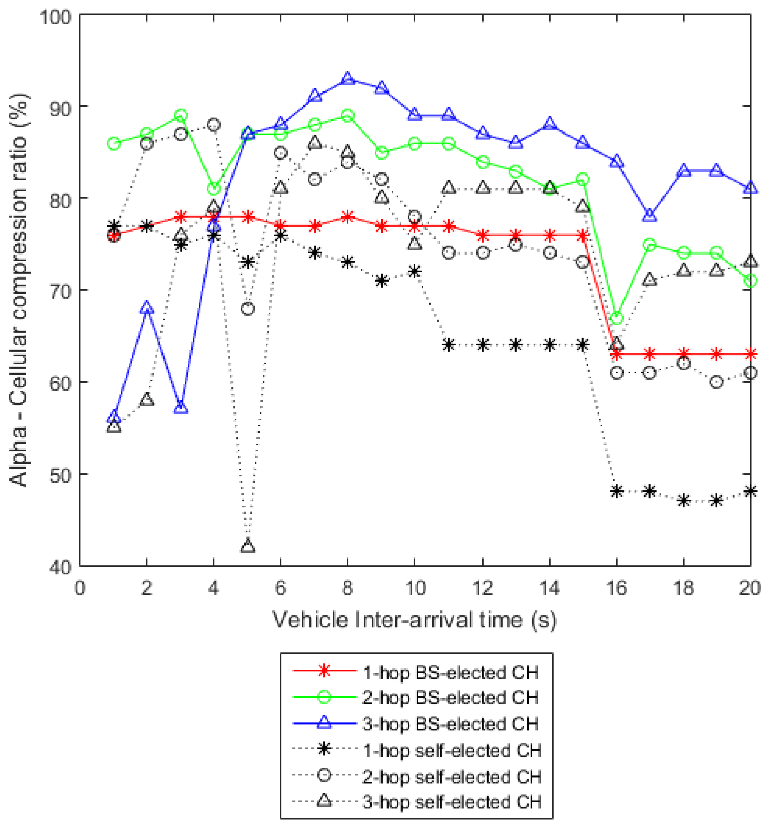


Figure 5.14: Global data compression ratio: Comparison between CH self-appointment and BS-based CH selection algorithms, for different maximum numbers of hops.

5 Conclusions

In this chapter, we showed and quantified the impact of the maximum number of hops and vehicular density on the size of the formed clusters, and consequently on cellular information aggregation efficiency. We also proposed a new algorithm which, by delegating the Cluster Head selection to the cellular Base Station, lets us profit much more efficiently of the advantages that increasing the number of hops can provide in terms of average cluster size.

When testing the VMaSC clustering algorithm for comparison with our proposal, we saw that the augmentation of the cluster size when increasing the number of hops, which is a desirable aspect of multi-hop clustering algorithms, was only marginally observable, if visible at all. This is why the proposed algorithm was designed in part for increasing this effect in order to improve the data aggregation ratio.

It comes to evidence that in high vehicular densities, 1-hop clustering is the most convenient option, since it has negligible packet loss (while 2-hop and 3-hop clustering largely surpass the maximum tolerated levels). As vehicular density decreases, the most convenient option in terms of data compression is to switch first to 2-hop clustering then to 3-hop clustering as the IEEE 802.11p packet loss rate decreases. This is why the will introduce an Auto-Adaptive Clustering Algorithm in the following chapter.

Chapter 6

Self-Adaptive Clustering Algorithm

1 Motivation and objective

In Chapter 5 we discovered that there are certain ranges of vehicular densities for which, given a constraint of a maximum acceptable packet loss ratio on the V2V network, we can determine the optimum number of hops for a multi-hop clustering algorithm. If we were able to let our clustering algorithm auto-adapt to changes in vehicular density, we would be ensuring a continuously optimum cluster size for every scenario, and thus achieving maximum data compression in the cellular network, without the need of human intervention in the system. This would allow us to achieve one of our global objectives, which is reducing the cellular communication costs, in every scenario.

Our objective in this chapter is to propose a self-adaptive multi-hop clustering algorithm that adapts to changes in traffic density, so that it can deliver maximum data consumption reduction in the cellular network for any given vehicular density, while never surpassing the maximum tolerable packet loss ratio on the V2V network.

In the following we introduce the **Self-Adaptive Clustering Algorithm**, an extension of the **Base Clustering Algorithm** presented in Chapter 5. We position ourselves in a scenario where there is an obligation, for every vehicle, to transmit its position, speed, and all of the data contained in its Cooperative Awareness Messages (CAM) to a distant server through the cellular network. This data is used for an improved traffic management system and other potential services that could need individual or aggregated data of the vehicles on the road. An isolated vehicle is obliged to send, through the cellular network, the information contained in every CAM that it broadcasts via the V2V network. If the vehicle is in a cluster, then it will only broadcast the CAMs over the V2V network, and the cluster's CH will upload the aggregated information via the cellular network.

The result of the information uploaded by CHs and isolated vehicles is that at any given moment the cellular network knows the position and general data of every vehicle on the road. This is how our *Auto-Adaptive Clustering Algorithm* gets to know and even to *predict* the vehicular density in each of the *clustering regions*, and thus it is able to dynamically adapt the maximum number of hops, by triggering a change (increment or decrement) of the maximum number of hops when reaching certain density thresholds, with hysteresis, empirically deduced from the results of the previous chapter.

In this chapter, we introduce the Auto-Adaptive Clustering Algorithm in Section 2. Subsequently, we present the configuration and results of the simulations in Sections 3.1 and 3. Finally, we draw our conclusions in Section 4.

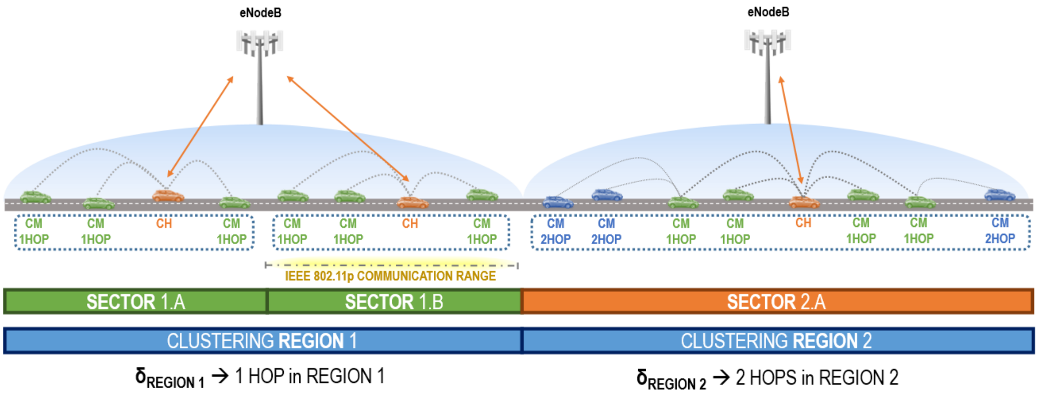


Figure 6.1: **Self-adaptive model basics:** A clustering region can be served by one or more cellular base stations (eNodeB in the case of LTE). In each region, the maximum number of hops is dynamically determined by the Self-Adaptive Algorithm (executed by the region's base station) in function of vehicular density. A clustering region is divided into multiple clustering sectors. The length of the clustering sector is equal to the IEEE 802.11p communication range, multiplied by the maximum number of hops in the region. The algorithm ensures that in every clustering sector there will be a Cluster Head. Only the CH exchanges data with the Base Station.

2 The Self-Adaptive Clustering Algorithm

2.1 Network Model

We consider a highway section where vehicles move in one direction[1]. We assume a scenario that corresponds to the multi-RAT environments that we expect to see in the upcoming 5G wireless systems: every vehicle is equipped with one transceiver for direct vehicle-to-vehicle communication, and another transceiver for cellular network access.

In this chapter, we continue our study of the example application of *Floating Car Data aggregation* seen in Section 2 of Chapter 5. Every vehicle has to send its identification and position to a distant Traffic Management Server at a rate of λ packets/s. If the vehicle does not belong to a cluster (i.e., it belongs to a cluster of size 1), it is obliged to send this information through its own cellular connection. If the vehicle is in a cluster c of size $N_c > 1$, it sends this information to the CH, which aggregates the data collected from all the vehicles in the cluster and send it to the remote server. This way, the cellular network is constantly aware of the traffic density in every area, and is able to identify and locate every vehicle that is not part of a cluster.

The model of network traffic and compression remains the same as in Section 2 of Chapter 5: any cluster c generates for the cellular network a traffic equal to $\eta(N_c)N_c\lambda$, where $\eta(N_c) \leq 1$ is a compression function performed by the CH. $\eta(N_c)$ may be a decreasing function of N_c. For simplification purposes, and without losing generality, we assume $\eta(1) = 1$.

The total cellular traffic of the set of all clusters ($\mathscr{C}$) is, then:

$$\Lambda(\mathscr{C}) = \sum_{c \in \mathscr{C}} \eta(N_c)N_c\lambda, \tag{6.1}$$

[1]For the case of vehicles moving in opposite direction, there are multiple possible solutions. For the cluster formation process, the easiest way is to enhance the speed information in the Cooperative Awareness Messages by adding the angle (vectorial speed). If this information is not available, it can be deduced from two consecutive messages including position. The messages coming from a vehicle moving in an opposite direction can then be filtered from the cluster formation algorithm. The CH selection algorithm runs then separately for each direction. This is left for further study.

where N_c is the number of vehicles in cluster c and $N = \sum_c N_c$ is the total number of vehicles.

We can now define the global compression ratio, α, as:

$$\alpha(\mathscr{C}) \triangleq 1 - \frac{\Lambda(\mathscr{C})}{N\lambda} = 1 - \frac{\sum_{c\in\mathscr{C}} \eta(N_c)N_c}{N} \tag{6.2}$$

Our objective is to maximize $\alpha(\mathscr{C})$ while respecting the constraint of keeping the Packet Loss Rate (PLR) below the acceptability threshold:

$$\max_{\mathscr{C}} \alpha(\mathscr{C}) \tag{6.3}$$

$$\text{s.t. } PLR(\mathscr{C}, \lambda) \leq PLR_{max}, \tag{6.4}$$

where PLR_{max} is an application-specific constraint.

2.2 Self-Adaptive Algorithm

Our algorithm consists of two parts: a CH selection algorithm (Algorithm 4) and a Hop adaptation algorithm (Algorithm 5).

For the purpose of these algorithms, the space is divided into **clustering regions** (see Figure 6.1), which correspond to the coverage of a single base station (or eNode-B). Every clustering region is characterized by a maximum number of hops H for the clusters that are formed there. Together with the V2V communication range, this maximum number of hops determines the **clustering diameter** (D) within this region. Every clustering region of length L_r is divided into **clustering sectors** of length D.

The CH selection algorithm (Algorithm 4) is deployed and executed within every base station for a single clustering region. Its role is to select one CH in every clustering sector of that region. It takes as an input the maximum number of hops H for its region, as provided by the hop adaptation algorithm. At regular intervals of $T_{selection}$ seconds, Algorithm 4 verifies if there is a CH in every sector. If there is a sector where no CH is present, it selects as CH the vehicle that is the closest to the sector center.

Algorithm 4 Cluster Head Selection Algorithm (Base Station)

```
Initialisation:
Set maintenance period T_selection.
Set the maximum number of hops H, an output of the hop adaptation algorithm
Set IEEE 802.11p radio range R and compute the clustering diameter D = 2 × R × H.
Divide the clustering region into S = L_r/D sectors.
Routine:
For t = nT_selection, n = 1,2,..., do
    For s = 1,2,...,S, do
        If there is no CH in s then
        Select as CH the vehicle that is the closest
        to the center of s.
        Endif
    Endfor
Endfor
```

Table 6.1: Model variables and their simulation values.

Name	Description	Value
H	Number of hops	*(dynamic)*
$H_{default}$	Default number of hops	3
L_r	Length of the clustering region	5 km
R	IEEE 802.11p communication range	800 m
D	Clustering diameter (length of a clustering sector)	2.R.H
$T_{selection}$	Timer for CH selection control	10 s
$T_{adaptation}$	Timer for hop number adaptation control	40 s
L_{PAZ}	Length of the Predictive Analysis Zone	*(formula 6.5)*
δ_{PAZ}	Vehicle density in the region's PAZ	*(dynamic)*
$\Delta_{1-hop_{min}}$	Density threshold for adaptation from 1 hop to 2 hops	17.5 vehicles/km
$\Delta_{2-hop_{min}}$	Density threshold for adaptation from 2 hops to 3 hops	5.5 vehicles/km
$\Delta_{2-hop_{max}}$	Density threshold for adaptation from 2 hops to 1 hop	22.0 vehicles/km
$\Delta_{3-hop_{max}}$	Density threshold for adaptation from 3 hops to 2 hops	7.0 vehicles/km

The hop adaptation algorithm (Algorithm 5) is also implemented in every base station for a single clustering region and is responsible for dynamically adapting the maximum number of hops H as a function of the vehicle density. For every number of hops H, there are two thresholds $\Delta_{H-hop_{min}}$ and $\Delta_{H-hop_{max}}$. If the observed density is less than $\Delta_{H-hop_{min}}$, we allow clusters to be larger and set $H := H+1$. If the density is higher than $\Delta_{H-hop_{max}}$, packet loss rate may increase, so we decrease the number of hop by one. Note that we may have $\Delta_{H-hop_{min}} \neq \Delta_{(H+1)-hop_{max}}$ to account for hysteresis. The hop adaptation is done every $T_{adaptation}$.

As the vehicle density may be heterogeneous inside a clustering region, taking decisions based only on average density may lead to poor performance. To tackle this issue, we define a **Predictive Analysis Zone** (PAZ) at the beginning of the clustering region. The formula for calculating the length of the PAZ (L_{PAZ}) is:

$$L_{PAZ} = min(\frac{L_r}{4}, 4 \times R) \tag{6.5}$$

The vehicle density is measured in both the PAZ (δ_{PAZ}) and in the rest of the region ($\delta_{\overline{PAZ}}$). We then take the decision about the number of hops based on $\max\{\delta_{PAZ}, \delta_{\overline{PAZ}}\}$. The idea is to benefit from the specific movement pattern observed on a highway section in order to avoid peaks in packet loss rates: if the density is higher in the PAZ, we anticipate the hop change; if the density is higher in the rest of the region, we account for the worst case. The computational complexity of our algorithms in function of the number of vehicles present in each sector can be characterized as $O(n)$ (where n is the number of vehicles in the sector). Knowing that the input size is constrained to the elements of a clustering sector, there can be no scalability problems. This is of course the complexity of the presented algorithms, but not the complexity of the simulation of multi-hop forwarding.

Algorithm 5 Hop adaptation algorithm (Base Station)

Initialisation:
Set maintenance period $T_{adaptation}$.
Set maximum number of hops $H := H_{default}$.
Set the Predictive Analysis Zone as the first $L_{PAZ} = \mathbf{min}(\frac{L_r}{4}, 4 \times R)$ meters of the clustering region.
Set triggering thresholds $\Delta_{k-hop_{min}}$ and $\Delta_{k-hop_{max}}$ for $k = 1, 2, 3$.
Routine:
For $t = nT_{adaptation}$, $n = 1, 2, ...,$ **do**
  Compute vehicular density $\delta = \max\{\delta_{PAZ}, \delta_{\overline{PAZ}}\}$.
  If $\delta < \Delta_{H-hop_{min}}$ **then**
    $H := H + 1$
    Notify all vehicles in the clustering region and Algorithm 4 of the change in H.
  Endif
  If $\delta > \Delta_{H-hop_{max}}$ **then**
    $H := H - 1$
    Notify all vehicles in the clustering region and Algorithm 4 of the change in H.
  Endif
Endfor

Figure 6.2 shows the message exchange between the base station (eNode-B) and the CHs or isolated vehicles. Table 6.1 shows the different variable definitions used in the description of the algorithms.

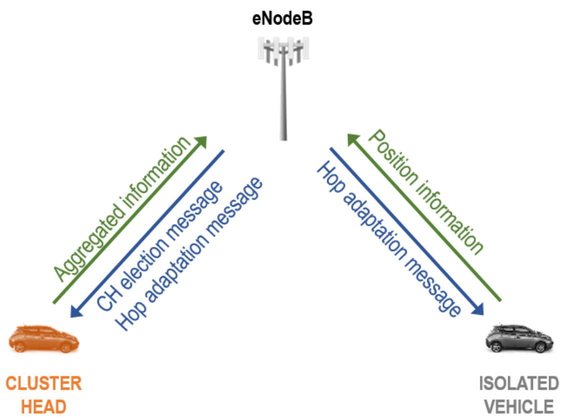


Figure 6.2: **Message exchange over the cellular network:** In the simple simulated model, only Cluster Heads and eventually isolated vehicles (technically, CHs of a cluster of size one) are the only ones that access the cellular network, exchanging the information seen in this figure.

3 Simulation

3.1 Simulation settings

The algorithms presented in Section 2 of this chapter were implemented and evaluated using the Veins [69] framework, that synchronizes a traffic simulation running in SUMO (Simulation of Urban MObility) [27] and a full-stack network simulation in OMNeT++ (see Chapter 4).

The map consists of a 10 km long highway segment, divided into two *clustering regions* of equal length. In terms of vehicular traffic, the tested scenario consists of three consecutive flows of significantly different densities: from 0 s to 2500 s, 100 vehicles enter the highway section at an average inter-arrival time of 25 s. From 2500 s to 4500 s, a second flow of 200 vehicles enter the highway section at an average inter-arrival time of 10 s. And finally, starting from 4500s, a flow of 1600 vehicles will enter at an average inter-arrival time of 1 second. The vehicular density in the two clustering regions (and in the entire highway segment) in function of time can be seen in Figure 6.3.

Table 6.1 lists the parameter values for the self-adaptive algorithm. Density thresholds are taken from the results in Chapter 5. We assume a compression function equal to $\eta(N_c) = 1/N_c$, where N_c is the cluster size. We compare the peformance of the Self-Adaptive Algorithm to the Base Clustering Algorithm (setting the number of hops to 1, 2 and 3 in different runs) in terms of $\alpha(\mathscr{C})$ and PLR.

We define the Packet Loss Rate in the V2V network as the ratio between lost packets (due to incorrect decoding or collisions) and correctly received and decoded packets. Since we have a medium that uses radio broadcast, defining which messages we should consider as lost may not be evident. In our case, the network simulator evaluates the path-loss of the radio signal, and can generate the associated random errors. We consider a packet as lost if the received signal power is enough to trigger the decoding process, but it leads to a decoding failure. We also count a packet as lost in case of a collision in the radio channel. The maximum tolerable PLR depends on the specific constraints of each application. For a Cooperative Awareness (CA) service, we have set our threshold at 10%. In our model, the amount unicast V2V messages (mostly for joining and leaving clusters) is extremely small compared to the amount of broadcast CA messages.

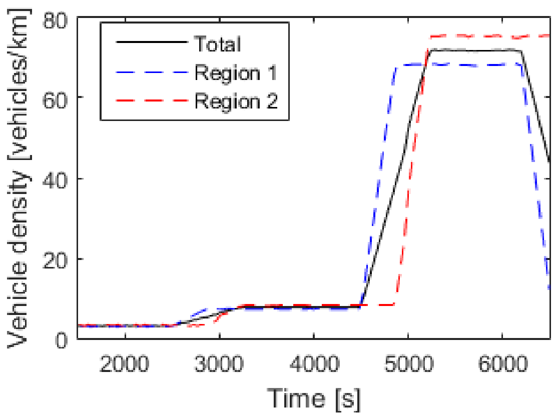


Figure 6.3: Vehicular density in the tested scenario in function of time, measured in regions 1 and 2, and the density in the entire highway segment comprising both regions.

3.2 Simulation results

The results are presented in separate figures for the different traffic flows (Figure 6.3) for an improved reading and analysis, but the reader should keep in mind that they are part of a single, continuous simulation.

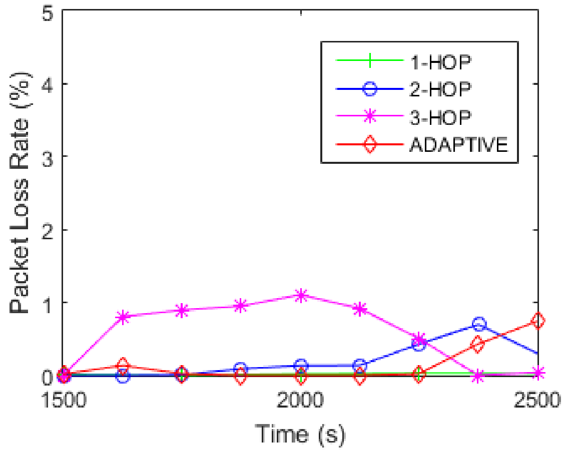
During the first part of the simulation, a very light traffic density is introduced. Vehicles are too far away from each other and, as we can see in Figure 7.10a, the 1-hop algorithm is unable to form big enough clusters, and is severely penalized in its aggregation performance when compared to the others (2- and 3-hop). The best aggregation performance goes, then, for the maximum number of hops: the 3-hop algorithm leads at all times, and the Auto-Adaptive Algorithm follows its behaviour. In terms of PLR (see Figure 7.10b), all the algorithms remain below 1%.

When the second flow of vehicles arrives at the mark of 2500 s, PLR and aggregation curves gradually change, and the 3-hop algorithm goes beyond the PLR acceptability threshold of 10%. The Auto-Adaptive Algorithm then changes the number of hops, from 3 to 2, and we can see a significant reduction of the PLR after the peak we get when the new flow starts (see Figure 7.10b). The Auto-Adaptive Algorithm's compression curve starts following the 2-hop curve.

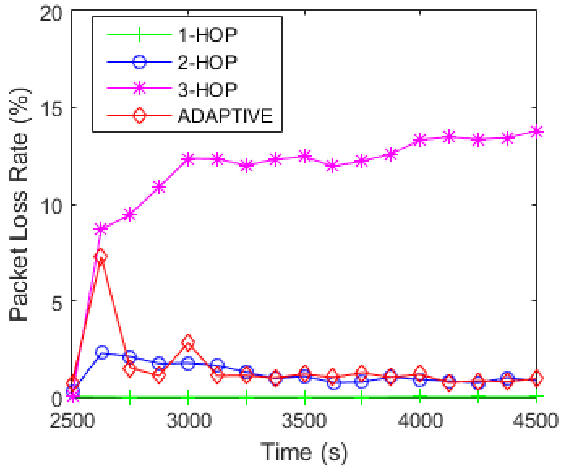
Finally, for the highest density (Figures 7.10b and 7.10a), the PLR curves of 2- and 3-hop skyrocketed, leaving 1-hop as the only viable possibility. The Auto-Adaptive Algorithm triggers a hop change again, resolving another PLR peak, while its aggregation performance follows the curve of 1-hop.

We now show that the signalling induced by the Auto-Adaptive Algorithm is negligible compared to the number of messages saved by clustering. We first compute the number of messages with destination the cellular network generated by every vehicle and the number of messages sent by the CHs. During the simulation of 6500 s, we observe a gain of 516, 184 messages thanks to compression. During the same simulation, we also observed 8 hop number change notifications (adaptations), and 98 cluster head proclamations (for a total of 1900 vehicles in the simulation). In the worst case, we thus have 98×8 change notifications and 98 CH selections for a total of 882 signalling messages used by our algorithm. This represents only 0.17% of the savings in terms of number of messages. Even if messages have different lengths, this rough estimation shows that the signalling overheads associated to our algorithm is negligible.

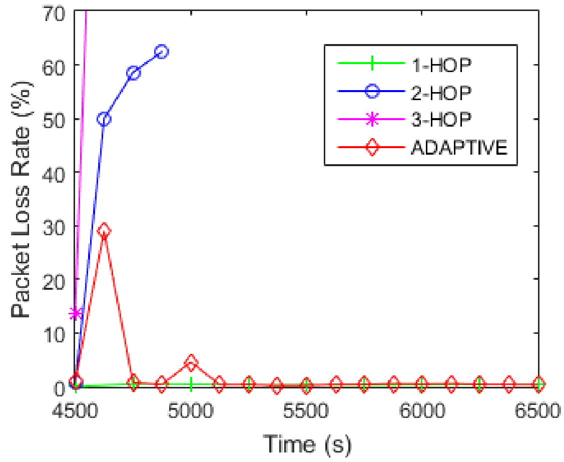
The fact that the the curve of the self-adaptive algorithm follows the curve of the algorithm with the best hop configuration, in the context of traffic density, shows the utility of the adaptive approach.



(a) Low density

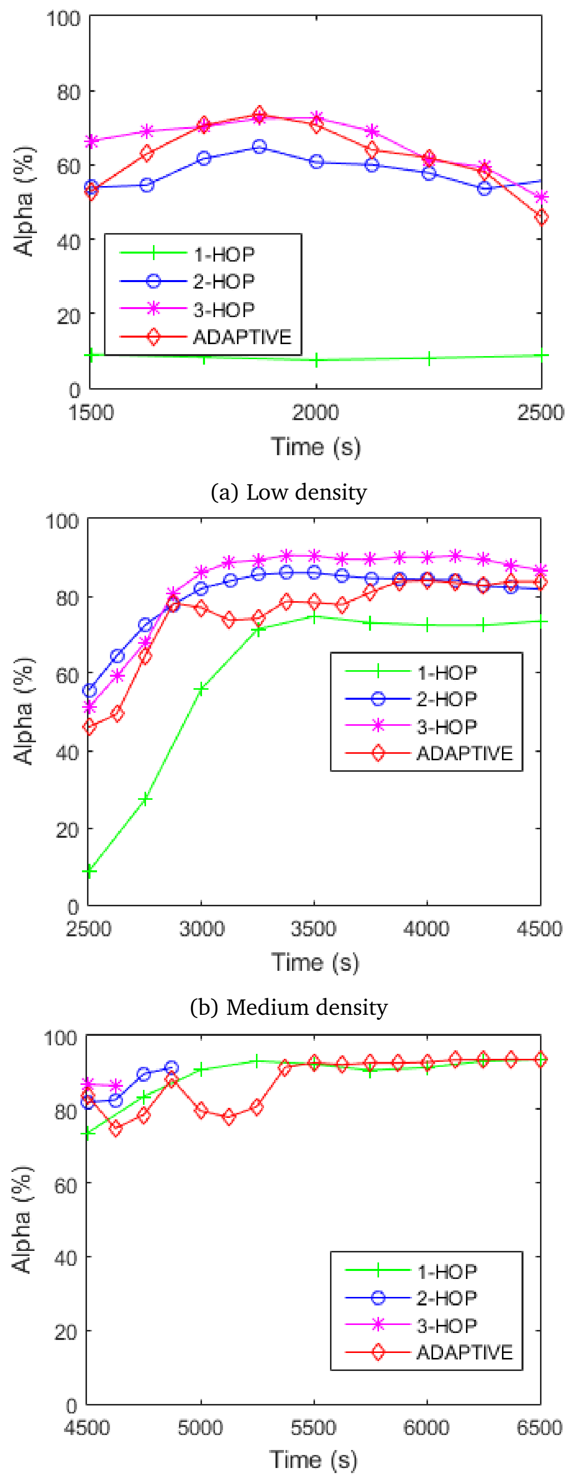


(b) Medium density



(c) High density

Figure 6.4: Packet Loss Rate (PLR) in function of time for the tested scenario. Comparison between 1-,2- and 3-hop configurations vs. Auto-Adaptive Algorithm.



(a) Low density

(b) Medium density

(c) High density

Figure 6.5: Cellular data consumption reduction (Alpha) in function of time for the tested scenario. Comparison between 1-,2- and 3-hop configurations vs. Auto-Adaptive Algorithm.

4 Conclusions

In this chapter, we considered a hybrid vehicular network, where vehicles are clustered using IEEE 802.11p (V2V) and communicate via their cluster head to a cellular network. We presented a Self-Adaptive multi-hop clustering algorithm that dynamically changes cluster sizes (by adapting the maximum number of hops in function of the traffic density) with the objective of reducing the usage of the cellular network resource while maintaining the packet loss rate in the V2V network below a certain threshold. The self-adaptive algorithm is deployed and executed onto each base station of the cellular network. Using an ITS simulation framework, we show that the Self-Adaptive Algorithm correctly adapts to extreme density changes, making the best possible reduction in cellular network usage (thus making important monetary savings at large scale) in the simulation experiments and results, while respecting the imposed constraints of packet loss rate on the V2V network, guaranteeing that specific applications' requirements can be met.

The **overhead** of the adaptation mechanism is small: it only requires one broadcast message from the base station to the vehicles in the area. Those vehicles that were, for example, at 3 hops from their CH, and receive a notification telling that the new maximum is 2, will automatically dissociate from their current cluster without any further message exchange. Their CH will know they have left and will erase them from its list of *children*, and the concerned vehicles will immediately start listening for new *CH Advertisements*.

The hysteresis mechanism implemented with the thresholds $\Delta_{H-hop_{min}}$ and $\Delta_{H-hop_{max}}$ prevents the possibility of excessive frequency of adaptation triggering, improving the **system's stability**.

While reducing the communication costs of the vehicular networks overall, this method raises the problem of the unequal distribution of costs for the Cluster Heads, which carry-out all communication with the cellular network. Hence, ensuring social acceptability of the proposed approach requires a better, or more "fair" distribution of costs across participating vehicles, over time. This is the object of study in Chapter 7.

Chapter 7

Distributive Justice: Fairness-Aware Clustering Algorithm

1 Introduction

In the previous chapters (Chapters 5 and 6) we saw that the proposed algorithms succeed in fulfilling the first of our two global objectives: *provide a solution for integrating V2V and cellular networks, in order to reduce cellular access costs while preserving network performance*. In this last part of our work, we focus on further extending our solution in order to achieve the second global objective of this thesis: to *ensure fairness over time of the proposed method, in order to improve its social acceptability*.

We face the problem of the fair distribution of the cellular communication cost that Cluster Heads have to absorb for the benefit of all the vehicles in the cluster. Even if the cost distribution is necessarily unfair over the short term (because all the costs are supported by the CHs) we can achieve a relatively fair cost distribution over the long term, by taking fairness into account during the CH selection process. For this, we applied the *Theory of Distributive Justice* as presented by Rescher [65] with the aim of achieving *fairness over time*, following the design principles for managing commons presented by Elinor Ostrom [60], which ensure a durable management of resources through principles explicitly conceived for consensual social acceptability. These principles state, for instance, that it is necessary that those affected by the rules of how a resource is managed can participate in the creation of these rules, and that there has to be a consistent monitoring, and gradual sanctions.

This chapter is organized as follows: we present the different possibilities for the billing of cellular access in Section 2; we comment on Ostrom's approach for Common-Pool Resources (CPR) problems in Section 3.1; we introduce the theory of distributive justice in Section 3.2; the *Fairness-aware Clustering Algorithm* is presented in Section 3.3; the simulation configuration and results are presented in Sections 4.1 and 4.2 respectively, and finally conclusions are drawn in Section 5.

2 Distribution of access costs to the cellular network

The cost of cellular network access is a key factor in the users' analysis of the convenience of participating in a cluster for information aggregation. There are different possibilities when it comes to billing, data consumption and access restrictions (see Figure 7.1). We analyze some of these in the following.

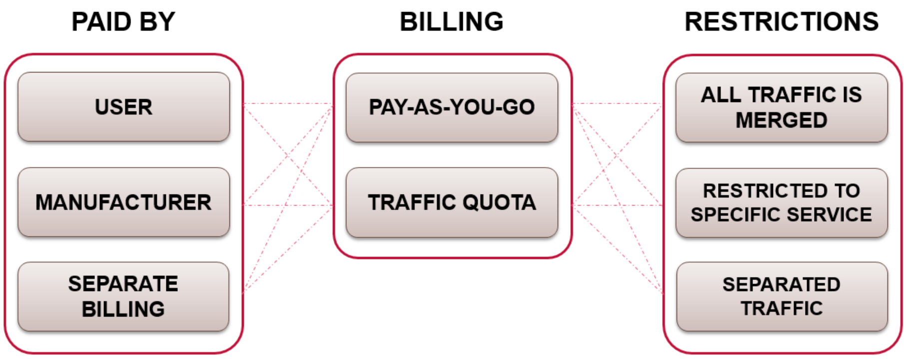


Figure 7.1: Possible economic models for cellular network access in the vehicular network.

2.1 The payer

Different business models attribute the costs for cellular network access to different system actors. In one model, communication costs are billed directly to **the driver**. Here, the driver uses his or her personal mobile subscription for both the personal data consumption and the demands of the cluster as a whole. in this case it becomes critical to find ways to motivate, compensate and balance the users' participation.

In another model, it is the **car manufacturers** that assume the communication costs of the vehicle's SIM card. For instance, Tesla offered unlimited connectivity to every person buying a Tesla Model S when it was released to the market back in 2002. However, the company has later changed its policy and is now offering it for free for a 4-year period only. European manufacturers considered absorbing the cellular cost only for covering the implementation of an automatic eCall, an initiative of the European Commission for the implementation of immediate automatic emergency calls in crash situations across the whole European territory. We could imagine that the manufacturer could extend this to some other safety features or messages but something further than that seems unlikely. Nevertheless, the same approach could be used if a single entity (for instance, an enterprise or public entity in charge of multiple services in the vehicular network) designed a business model where absorbing the cost of the vehicles' cellular network access could be compensated by the benefits of the generated information, or some other indirect income, like drivers accepting advertisements in exchange for valuable aggregated information mined from the vehicular network as a whole. In all of these scenarios, however, the usage of the cellular network with that SIM card would obviously be restricted to the specific traffic that the manufacturer or enterprise decides.

Fnally, cars can be equipped with double SIM cards, one for personal data and individual multimedia access, and another one for shared data (for which clustering can be applied), and the **billing can be split** between these cards. It is possible by using dual SIM slots, by using one SIM with split billing on the operator side (uncommon for the moment), or by enabling routing of specific traffic (generated by selected applications) through a different Access Point Name (APN).

The option of separate billing can even distribute the cost between the user and more than one enterprise (the car manufacturer, the provider of a specific service or application, etc.). However, there is no mean, at least by directly using these approaches, of splitting the bill between the members of the cluster.

2.2 The billing method

- **Pay-as-you-go:** The total bill is calculated as a function of the data consumption, usually measured in MBytes. This function can be either linear or stair-like.

- **Traffic quota:** Each vehicle, or driver, has a subscription for data consumption that cannot go beyond a certain quota (e.g. 1, 3, 5, 20 GB per month). Once this limit is reached, the operator can apply certain restrictions, such as reduced data rate, less access priority, or even total connection block.

2.3 The access restrictions

- **Merged traffic:** The user's personal data and the floating car data that is sent through the cluster are transmitted through a single account, using the same SIM card, with no differentiation. Since normally no third party would pay for the user's data consumption, we assume that it is most likely that in this case, the information transmitted between the cluster and the cellular network is billed to the personal account of the user. In this scenario the users are expected to be very reluctant to participate as Cluster Head.

- **Restricted traffic:** The on-board SIM card's associated account in the operator's network is only allowed to exchange specific types of traffic. This is the most likely case when the floating car data transmission is paid by a third party (manufacturer, traffic management entity, another enterprise).

- **Separated traffic:** The cellular operator manages a separate billing for a single SIM card. This could be a method for billing the FCD transmission to a third party, while not restricting access to other types of content for the user (but billing him accordingly).

2.4 Retained model

Having examined current market trends [58][44][57], it seems unlikely that car manufacturers will take charge of the cost of cellular network communications. In any case, the owner or regular driver of the car will always be the ultimate payer, directly or indirectly. Therefore, in our model, cellular accounts and quotas are strictly individual. Every transmission to the cellular network, as an isolated vehicle or as a Cluster Head, is charged to the account of that specific person.

3 Fairness-Aware Clustering Algorithm

3.1 Common-Pool Resource (CPR) Problems

Our application domain features different types of shared resources – e.g. aggregated information from clusters of vehicles; subscription-based access to cellular networks, at a cost; and unlicensed V2V access, prone to saturation if unregulated. For all of these resources, we have different degrees of ability for regulation (for preventing someone from using them). The usage that someone makes of each of these resources affects in different degrees how much others can yield from it. This corresponds to the criteria of *excludability* and *subtractability* used by Elinor Ostrom [60] to classify goods in four categories (i.e. public goods, toll goods, private goods and common-pool resources).

After studying several cases of resource exploitation in different communities, and for very diverse types of goods, Ostrom presented eight design principles for governing commons, ensuring the endurance of the resources. These design principles are considered important: from the case studies, communities that managed to preserve their shared resources successfully featured these principles; and the communities that failed had excluded at least one of the design principles from their self-governance organization. In our work, we have kept these design principles in mind: group boundaries are clearly defined, and every subject can participate in the determination of the rules for selecting a Cluster Head through a vote. One of the proposed metrics for the cluster head selection translates to a gradual sanction for non-compliance with the rules. And since the cellular network, which is the ultimate decider for the Cluster Head selection, has all the data consumption statistics and participation history for everyone, a constant monitoring is ensured.

3.2 A Theory of Distributive Justice

Ostrom's design principles for managing commons may increase the chances that a resource *endures*, but they do not imply that the resource distribution outcome is actually fair. It is however not trivial to define fairness. In [63], the authors summarizes several desirable properties of a *fair* result:

- *Proportional*: Each agent receives a resource allocation proportional to its participation in the system;
- *Envy free*: The allocation of a given agent should not be "more desirable" to another agent, compared to its own allocation;
- *Equitable*: No agent derives more utility from its received allocation than any other agent in the system derives from its allocation;
- *Efficient*: Deliver the greatest good for the greatest number;
- *Cost-effective*: The computational cost of the distribution is not excessive, or in other words, it is economically feasible and reasonable for the served purpose;
- *Timely*: The computation of the distribution can be completed fast enough to be useful.

Nonetheless, evaluating fairness based on all of these properties at the same time raises two main problems. On the one hand, these properties do not always come together and there can even be a conflict between them. For instance, a cost-effective distribution method that does not produce an equitable outcome. On the other hand, and most importantly, these properties are usually applied to static distribution methods (there is not a series of allocations, but only a one-time distribution). The proposal in [63] focuses on *fairness over time*, through a series of individually unfair allocations that eventually lead to a fair result on the long term, by following Rescher's idea of *legitimate claims*. In this respect, our interest matches their motivation: it is impossible to produce a fair outcome when designating a cluster head, but we aim to achieve a globally fair designation outcome over the long run.

Rescher [65] introduces the concept of *social justice*, which consists in determining what legitimate claims an individual has, and treating each of these claims equally. His work does not investigate how those claims can emerge, but rather focuses on the concept of *distributive justice*, which deals with how far the legitimate claims of each individual should be met, given competing claims from other individuals, and limited resources.

Rescher analyzed the most usual ways that other authors had used to determine fairness and proposed what he called the *seven canons of distributive justice*: treatment as equals, according to each one's needs, according to each one's productive contribution, to efforts and sacrifices, to the social value of the services provided by the individual to the society, to supply and demand, or to merits and achievements. Rescher found that none of these canons, taken individually, could grant true distributive justice. Instead, he proposed to analyze every participant's legitimate claims following each of these aspects, and focus on how to balance them in case of conflict.

3.3 The Algorithm

In the previous chapters (Chapters 5 and 6), we proposed a self-adaptive clustering algorithm that aims to optimise the usage of the cellular network resources. For clarity, we will call this one the **Fairness Agnostic Algorithm**. The problem that this algorithm has is that, since the CH is the only one to bear the cost in terms of cellular quota consumption at a given moment, this can lead to unfair cost distribution, which may become socially unacceptable. Therefore, the presented contribution aims to provide a **Fairness-Aware Algorithm**, which addresses this issue.

We propose a practical mapping of our problem of distributing the cost of the cellular network access among the clustered vehicles in terms of Rescher's seven canons (since the seventh canon is not translatable to any term of merit or achievement in our case since there are no such concepts in our problem, we are focusing only on the first six, merging two of them that become equivalent). This way, we create five criteria that constitute five total orders of the vehicles in a cluster region, that are subsequently used for deciding who has to be the Cluster Head. According to each canon, these are the methods to determine the order in which vehicles should be selected as CH:

- **The canon of equality**: the more data a cluster member received previously, the higher the chances of being selected as cluster head;
- **The canon of needs**: the bigger the amount of available cellular quota, the higher the chances of being cluster head;
- **The canon of productivity**: the bigger the volume of data previously shared as Cluster Head, the lower the chances of being cluster head;
- **The canons of effort / social utility**: the higher the number of times having previously served as Cluster Head, the lower the chances of being cluster head;
- **The canon of supply and demand**: the higher the number of non-compliance events (for instance, switching off the data connection while occupying the Cluster Head role), the higher the chances of being selected cluster head. It is important to notice that this canon mapping takes place under the hypothesis that we are dealing with a **free-riding** problem.

These canons constitute total orders: they can provide an ordered list, where all the vehicles can be sorted from higher to lower values for that criterion. The data needed to calculate all five total orders for a group of vehicles is always available for the cellular base station, which is, in our model, the one that has the ultimate decision on who will be proclaimed as CH.

However, each vehicle has the knowledge of all the specific metrics for itself only, and can thus deduce which criteria are more favorable to its situation. Each vehicle then has the opportunity to demand to be judged by the most advantageous claim it has. For example, if it has been selected CH too many times, or if it has a very low available quota, it can demand these criteria to have a higher importance. All the vehicles in a cluster region cast a vote that results in the *weights* of each canon for this group of vehicles in particular. The base station, aware of this result, establishes a weighed Borda [33] election: each canon acts as one Borda voter. For instance, according to the canon of Equality, the order of priority for being selected as CH could be "*1) Vehicle A; 2) Vehicle B; 3) Vehicle C*" while the canon of needs could determine "*1) Vehicle B; 2) Vehicle C; 3) Vehicle A*". If equal weights are assumed here, *Vehicle B* would be selected as CH.

The final decision is then made as follows:

1. The base station discovers a clustering region without Cluster Head and requests the vehicles in the area to cast a vote;

Algorithm 6 Distributed Criteria Vote (Vehicle)

Precondition:
Elapsed time of the current billing period is not zero.
Procedure:
Retrieve from local database:
    n_{CH}: Times having been selected CH
    d_{CH}: Data shared as CH
    d_{CM}: Data saved by being CM
    Q_a: Available quota
    n_{NC}: Number of non-compliance events
    t_e: Elapsed time of the current billing period
    t_b: Billing period total time
    $M_{t_{CH}}$: Historic maximum value of n_{CH} per time unit
    $M_{d_{CH}}$: Historic maximum value of d_{CH} per time unit
    $M_{d_{CM}}$: Historic maximum value of d_{CM} per time unit
    Q_t: Data consumption quota per billing period
    $M_{n_{NC}}$: Historic maximum value of n_{NC} per time unit
Estimate the normalized preference values for each criterion:
    $E_{n_{CH}} = \frac{n_{CH}}{t_e/t_b.M_{t_{CH}}}$
    $E_{d_{CH}} = \frac{d_{CH}}{t_e/t_b.M_{d_{CH}}}$
    $E_{d_{CM}} = 1 - \frac{d_{CM}}{t_e/t_b.M_{d_{CM}}}$
    $E_{Q_a} = 1 - \frac{Q_a}{Q_t}$
    $E_{n_{NC}} = 1 - \frac{n_{NC}}{t_e/t_b.M_{n_{NC}}}$
If any of these estimations is bigger than 1 (new historic maximum)
    then set it to 1
If any of these estimations is smaller than 0 (new historic maximum)
    then set it to 0
Create the vote vector **v** (5 positions corresponding to the 5 criteria)
Initialize $(\mathbf{v}[1], \ldots, \mathbf{v}[5]) = (0, \ldots, 0)$
Let i be the number associated to the criterion corresponding to the maximum normalized preference value.
Set $\mathbf{v}[i] = 1$
Submit the vote vector **v**.

2. Each vehicle estimates the canon that is most favorable to it and votes for this canon locally. Each vehicle can cast a single vote, for only one canon. Since we have five criteria, the vote of vehicle i will be a vector $\mathbf{v_i}$ of five elements, with one element equal to 1, and the rest equal to zero. Different voting systems could also be implemented. Another option could be assigning all integral values from 1 to 5 in each of the vector's position, resulting in a nested Borda vote. This option could be evaluated in future work;

3. The outcome of the vote is the set of weights for each canon, determined by the cellular base station after receiving the votes. The weight of the canon j will be determined by the following formula:

$$w_j = \frac{\sum_{i=1}^{N_c} \mathbf{v_i}[j]}{N_c}, \tag{7.1}$$

   where N_c is the number of vehicles in the clustering sector;

4. The cellular base station calculates the total orders associated to each canon;

5. The cellular base station runs the weighed Borda election, by applying the weight coefficient to each Borda vote emitted by each canon;

6. The cellular base station applies a final correction to the total order obtained from the Borda election: a normalization lowering the score for being selected as CH as the vehicle's distance from the center of the clustering region increases.

7. The notification is sent to the designated Cluster Head.

It is important to note that, when passing from one base station to another, the clusters remain formed as they are. Only when there is no cluster head in a *clustering sector*, a new CH selection takes place. And when this happens, any base station has access to the same up-to-date information about every user's account information in the mobile operator's system. Thus, there is no way that passing from one base station to another could affect fairness.

Algorithm 7 Fairness-aware CH Selection Algorithm (Base Station)

Parameters:
β: a parameter that can be based in social input, to balance fairness and performance.
Triggering condition:
No CH in clustering sector s.
Procedure:
Request all vehicles in sector s to run Algorithm 6.
Calculate the weight of each criterion: $w_j = \dfrac{\sum_{i=1}^{N_c} \mathbf{v_i}[j]}{N_c}$
For every vehicle i in sector s:
    Retrieve n_{CH_i}, d_{CH_i}, d_{CM_i}, Q_{a_i}, N_{NC_i}.
End for.
Create $L_1, \ldots, L_5$: ordered lists for each criterion.
    Define list element: (value, vehicle ID)
    Order by: value
    L1: (n_{CH_i}, i), increasing order
    L2: (d_{CH_i}, i), increasing order
    L3: (d_{CM_i}, i), decreasing order
    L4: (Q_{a_i}, i), decreasing order
    L5: (N_{NC_i}, i), decreasing order
Calculate the weighted Borda score for every vehicle in s:
    Let N_s be the number of vehicles in sector s
    Let $Pos_{i,j}$ be the position of vehicle i in L_j $(1 \leq Pos_{i,j} \leq Ns)$
    Initialize $FairnessScore_i = 0$
    $FairnessScore_i = \sum_j (N_s + 1 - Pos_{i,j}).w_j$
Calculate the centrality score for every vehicle in s:
    Let d_i be the distance between vehicle i and the central point of s
    Let d_{max} be the maximum value of d_i for $i = 1, \ldots, N_s$
    Let d_{min} be the minimum value of d_i for $i = 1, \ldots, N_s$
    Calculate normalized distance $D_i = \dfrac{d_i - d_{min}}{d_{max} - d_{min}}$
    $CentralityScore_i = N_s.D_i$
Calculate final scores:
    $FinalScore_i = \beta.FairnessScore_i + (1 - \beta).CentralityScore_i$
Select the vehicle with the maximum $FinalScore$ as CH
    Notify the new CH.

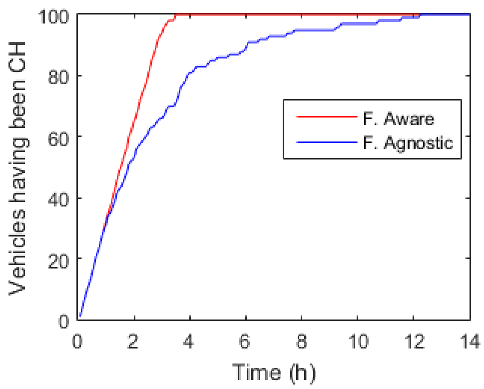


Figure 7.2: Number of vehicles having served as CH *at least once*, in function of time.

4 Simulation

4.1 Simulation configuration

The experiment that we ran for testing the proposed algorithm consists of a group of 100 vehicles that pass through a 10 km highway segment, for approximately 100 times each, with randomized order of re-entrance. The simulated access protocol in the V2V network is IEEE 802.11p. The protocol for the network and transport layers is IEEE 1609.3. In the application layer we implement the Cooperative Awareness Messages (CAM) from the European ETSI ITS G5 standard. The path loss follows a Two-Ray Interference model [70].

In the results curves, where time is represented in the horizontal axis, it refers to the simulated time, where the 100 vehicles randomly re-enter with a precise inter-arrival time. This is thus the equivalent of the time that it would take for 10,000 vehicles to traverse the 10 km highway segment, with an uniform inter-arrival time.

4.2 Simulation results

4.2.1 Vehicles having served as CH

The most remarkable outcome of introducing distributive justice in the long term is that the role of Cluster Head, which seems like a temporary burden, will be taken by a greater proportion of the participants if we analyze successive samples over time.

In Figure 7.2 we can see how the curve of the number of vehicles having been Cluster Head rises much faster when using the *Fairness Aware Algorithm*: after **less than 4 simulated hours**, all of the vehicles have taken the CH role at least once. In contrast, with the *Fairness Agnostic Algorithm*, it is only after **more than 12 simulated hours** that we reach the same result.

Additionally, we have to analyze how many times the same vehicle has been selected as CH. Reducing the amount of times that a specific user takes this role is an important way for improving the user's perception of justice. In Figure 7.3 we can observe the evolution of the maximum number of times that a specific vehicle has been selected CH. We see that, for the maximum value, the *Fairness Agnostic Algorithm* is always in a situation of almost doubling its counterpart.

The box-plot in Figure 7.4 shows the distribution of CH assignments across vehicles, over time. Regardless of the time passing, the size of the boxes of the *Fairness Aware Algorithm* remain small and

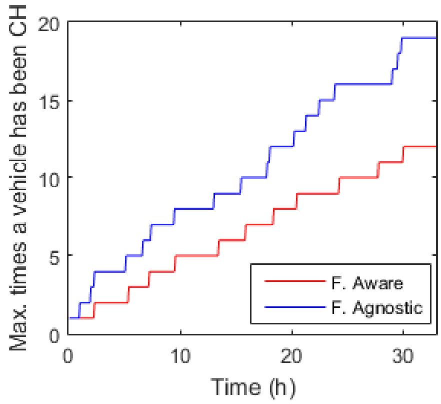


Figure 7.3: Maximum number of times that any one vehicle has served as CH in function of time.

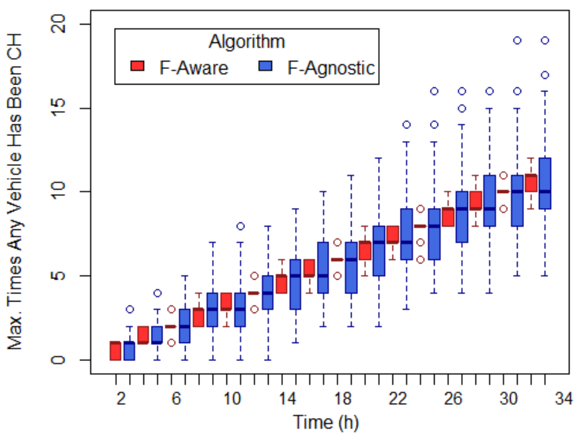


Figure 7.4: Box plot showing the distribution of the number of times that every vehicle has served as CH, for both algorithms, over time.

don't change their size. Their whiskers are usually tiny or non-existent. This means that, even if the average number of times that vehicles in general serve as CH increases with time (which is necessary), all of the vehicles serve approximately the same amount of times as CH. On the other hand, for the *Fairness Agnostic* boxes, their size increases, and the whiskers keep getting bigger, showing extreme disparities between the participants.

In the box plots we present, the solid band represents the median (second quartile), while the box is delimited by the first and third quartiles. The whiskers mark the lowest and highest datum still within 1.5 Inter-quartile Range (IQR) of the lower and upper quartiles, respectively. The outliers are marked as circles.

The effect in the long term can be seen in the *"final picture"* of Figure 7.5 where we can see the histogram of the number of times that every vehicle has served as CH when the end of the simulation is reached. We see an significant dispersion between values ranging from 4 to 20 for the *Fairness Agnostic Algorithm*, while for its *Aware* counterpart, almost 60% of the vehicles served exactly 11 times as CH, while the other 40% did so either 10 or 12 times.

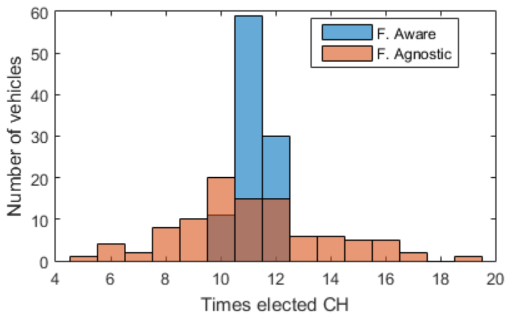


Figure 7.5: Histogram of the final number of times that any one vehicle has served as CH.

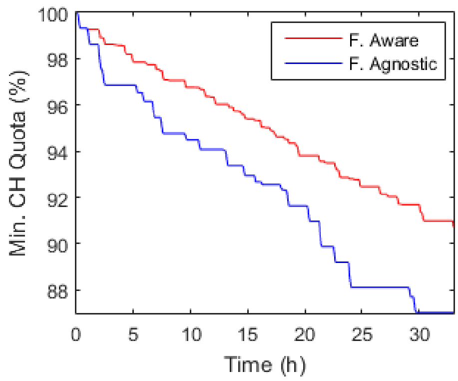


Figure 7.6: Minimum available cellular quota among the vehicles having served as CH during the simulation.

4.2.2 Cellular quota consumption

The differences in the outcome of the Cluster Head selection that we have just discussed in the previous point, has a direct impact on cellular quota consumption. We will now see some examples of the different distribution of the cellular quota usage with both algorithms, which translates almost directly into different economic costs.

4.2.3 A better worst case

We start by analyzing the case of our most unfavoured user for both algorithms in Figure 7.6. This curve shows the lowest individual available quota among the vehicles who have been CH, over time. We can see a pronounced difference in the slopes. This means that even the most unfavoured participant will always be much better off, in terms of economic cost, with the *Fairness Aware Algorithm* than with the agnostic one.

4.2.4 A fair distribution

Let us now see the impact on the available quotas of all vehicles (including those that have not been CH at a given moment). Figure 7.7 shows the box plot of the available quotas of all vehicles over time. Once again, the *Fairness Aware Algorithm*'s boxes and whiskers remain always small, tightly following the straight line of the global average data consumption. On the other hand, the distribution of the

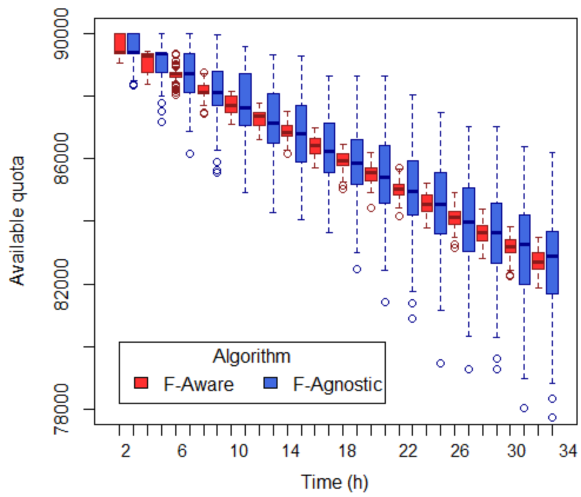


Figure 7.7: Box plot showing the distribution of the available cellular network quota for every vehicle, for both algorithms, over time.

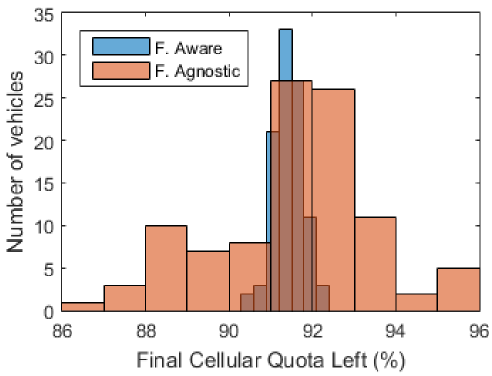


Figure 7.8: Histogram of the final available cellular quotas among the vehicles having served as CH.

quota consumption (and hence the economic cost) in the *Fairness Agnostic Algorithm* becomes more unfair with time, with major deviations from the median.

The clear final picture of the situation can be seen in the histogram of Figure 7.8, where we can see that for the *Fairness Aware Algorithm*, all 100 vehicles fit in a very tight range of final quotas, while for its *Agnostic* counterpart, they are distributed in a much broader range of possible final quotas, with just a few vehicles in each bin.

4.2.5 Coincident selections

The *Fairness Agnostic Algorithm* has only one possible outcome for every CH selection: the vehicle which is closer to the geometric center of the clustering sector. For the *Fairness Aware Algorithm*, this criterion is merely one amongst several choices (though it has a special weight, being considered in a second phase of the selection process). Figure 7.9 shows the proportion of selections that share the same result in both algorithms: those where the "central" vehicle is selected. The solid line represents the number of CH selections taking place over time, while the dashed red line shows the

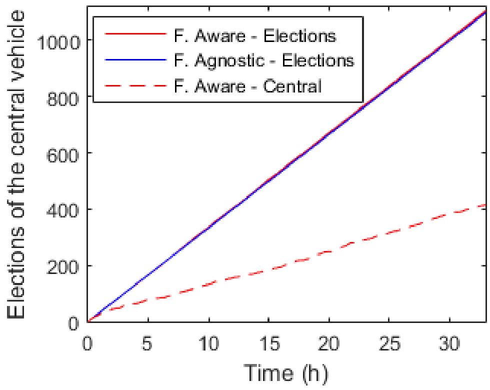
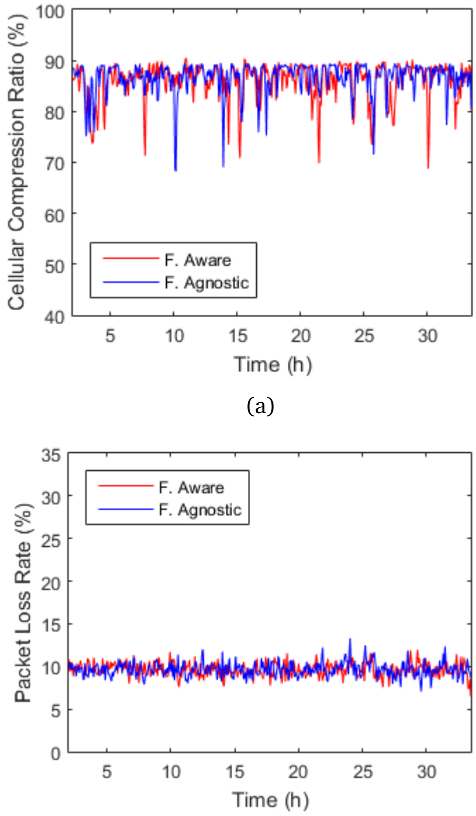


Figure 7.9: Total number of CH selections in both algorithms, compared to the number of selections in which the *Fairness Aware Algorithm* and the *Fairness Agnostic Algorithm* make the same choice: the vehicle in the geometric center of the clustering sector.

number of selections in the *Fairness Aware Algorithm* that result in the central vehicle being selected. It is interesting to see that in both algorithms, the number of selections over time remains the same, and that the slight deviation from the original algorithm's geometry implied by not always selecting the central vehicle does not have an impact on the number of selections. On the other hand, the proportion of selections having the outcome of selecting the central vehicle for the *Fairness Agnostic Algorithm* is, trivially, equal to the number of selections.

4.2.6 Network performance

The original (*Fairness Agnostic*) clustering algorithm aimed to self-adapt the size of the clusters in order to keep an optimal balance in terms of network performance: maximizing data compression in the cellular network (and thus reducing economic cost) while limiting the Packet Loss Rate (PLR) in the V2V network, to a maximum acceptable threshold of 10%. We could expect that modifying the geometry and criteria of the selection process could affect the metrics that this algorithm was designed for. Figures 7.10a and 7.10b compare both algorithms in terms of data compression and PLR, respectively. As we can clearly see, the improvements in distributive justice come at **no cost** in terms of network performance.



(a)

(b)

Figure 7.10: Comparison between the *Fairness Aware* and *Fairness Agnostic* algorithms, regarding: (a) Data compression ratio on the cellular network link over (simulated) time. (b) Packet loss ratio (PLR) on the V2V network. The results show that incorporating fairness improvements does not have a cost in the network performance metrics that the *Fairness Agnostic* algorithm was designed for.

5 Conclusions

We have presented a Fairness-aware CH Selection Algorithm based on Rescher's theory of distributive justice. This solution aims to address the problem of social acceptability of clustering algorithms, where the individuals selected as Cluster Heads bear all the costs. We have applied this approach to our previous work on a specific self-adaptive clustering algorithm, conceived for reducing cellular access costs by only communicating aggregated data via the CH, while limiting overheads onto the V2V network. The previous algorithm was effective in terms of global data compression and network performance, but unfair with respect to CH selections and hence to the cost distribution across system users (and since a driver can keep track of its quota consumption, it could act as a motivation to leave the system, potentially leading to a collapse). In contrast, the proposed fairness-aware algorithm enables vehicles to influence CH selections by expressing their legitimate claims via voting (based on a Borda-type vote). Numerical simulations showed that this approach significantly improves fairness across all vehicles, over time (analysis based on several metrics that a regular user would most likely analyze when elaborating its own perception of justice). Meanwhile, simulations also showed that the fairness-aware algorithm preserves all network performance optimisations achieved by its fairness-agnostic counterpart.

We aim to design a system for cluster-based vehicular networks that follows Elinor Ostrom's [60] design principles for managing commons. These principles ensure that a resource endures, being governed equitably and sustainably. Our algorithm currently follows some of these principles. The vote of the participant vehicles (P1) for determining the weight of each legitimate claim in the CH selection is a way for them to be involved in the rule-making process. The groups have clearly limited boundaries (P1). Even one of our criteria (taking into account non-compliance events) gets close to the gradual sanctioning required in Ostrom's principles (P6). In future work, it would be interesting to consolidate the monitoring and sanctioning required by Ostrom, and explore possible comparisons of the Fairness Aware Algorithm with other existing methods that may be adapted to this problem.

Chapter 8

Conclusion

In this thesis we have addressed the problem of the integration of different radio access technologies in vehicular networks. Having analyzed the state-of-the-art technology, we discovered that vehicle-to-vehicle communication standards such as IEEE 802.11p and cellular standards such as *Long-Term Evolution* (LTE/4G) have very dissimilar but complementary characteristics.

Future Intelligent Transportation Systems (ITS) applications will require big volumes of data that will have to be uploaded from the vehicles to distant servers or devices through the cellular network. Having explored different economic models, we realized that the economic cost of the cellular network access can become a major problem. However, some applications for vehicular networks are characterized by a local redundancy that can be exploited for information aggregation and offloading, significantly reducing the usage of cellular resources.

In order to do so, we proposed a multi-hop clustering algorithm for information aggregation and reduction of the cellular network usage. It was presented in three stages. The **Basic Clustering Algorithm** served the main objective of efficient data aggregation, but at a fixed number of hops, it helped us identify a new problem: for some traffic densities, as the cluster size increases, aggregation is higher but packet loss ratio (PLR) in the vehicle-to-vehicle (V2V) network also increases significantly due to broadcast storms.

In a second stage, we proposed the **Self-Adaptive Clustering Algorithm**, which dynamically changed the number of hops as a function of the traffic density. Simulation results showed that when the traffic density changes, the performance of this algorithm in terms of data aggregation and packet loss ratio on the V2V link quickly matches that of the *Base Clustering Algorithm* when configured at the static number of hops that best suits the current scenario. At this stage, we had an algorithm that efficiently integrated both radio access technologies —*V2V and cellular*— in every situation (from the point of view of traffic variations).

Another problem emerged when considering this new approach: the Cluster Head was the one bearing all the cost of cellular communications during the time it was occupying that role, while other cluster members would not touch their quotas. It was necessary to ensure a fair distribution over time of the responsibilities of being elected Cluster Head in order for the system to be socially acceptable. We proposed the *Fairness-Aware Clustering Algorithm*, which incorporated the Theory of Distributive Justice, and followed Elinor Ostrom's design principles for managing resources (see Section 5). Simulations showed that for every considered metric (quota consumption, number of times being elected CH, volumes of data shared as CH, etc.), the *Fairness-Aware Clustering Algorithm* provided a much more evenly distributed outcome, much faster than the *Self-Adaptive* (or *Fairness-Agnostic Algorithm*).

We have proposed an algorithm that efficiently integrates two different radio access technologies in vehicular networks. The linking of vehicular and cellular networks, as well as the coordinated usage of different radio access technologies are clear objectives of the development of Fifth Generation Mobile Networks (5G). The work presented in this thesis can therefore enrich the literature for future research works in this area.

Future work directions could include the incorporation of reinforcement learning approaches for the choice (and perhaps dynamic change) of the hop adaptation triggering thresholds, taking into account not only the effects of traffic but also those of radio propagation. The current thresholds have been fixed for a given radio propagation model but they can change, for example, if the path loss is higher, or if the standard deviation of shadowing is bigger. These parameters have a direct impact on PLR.

A first approach for predicting traffic density variations in clustering regions has been proposed, and it consists of a special zone called the Predictive Analysis Zone (PAZ). In future work, this prediction could be reinforced in order to avoid a short peak of PLR that is observed at the moment of triggering the change of the number of hops. We could, for instance, use inter-cluster messages that allow to have an idea of the incoming traffic.

We have considered a relatively simple scenario with only one highway segment, with traffic in only one direction. It would be interesting to see how the proposed algorithm behaves with two-way traffic, or entrance and exit ramps. In line with these ideas, it would be interesting to use real vehicular traces or making an actual experiment.

We have considered IEEE 802.11p because it is a candidate protocol for this type of applications, but device-to-device (D2D) communications partly controlled by the base station could be used for the intra-cluster message exchange. This could significantly reduce the PLR at the cost of a more important usage of the cellular network. It is also an option to study. Another important point regarding protocols is the security aspect. In order to implement our algorithms, a protocol-specific analysis should be done in order to determine how to forward packets (whether a packet should be de-encapsulated and analyzed, and then building a new packet, or if the forwarded packet should be encapsulated with its original signature), how to prevent spoofing and man-in-the-middle attacks, and other similar questions.

Finally, concerning the incorporation of fairness approaches, future work could be done for comparing with other proposals, or different voting methods, and especially running experimentations in order to test the method's acceptability degree.

Bibliography

[1] itetris project.

[2] Regulation (eu) 2015/758 of the european parliament and of the council of 29 april 2015 concerning type-approval requirements for the deployment of the ecall in-vehicle system based on the 112 service and amending directive 2007/46/ec.

[3] Title 47 of the code of federal regulations, parts 90 and 95. Federal Communications Commission (FCC), 2003.

[4] 1609.1-2006 - trial-use standard for wireless access in vehicular environments (wave) - resource manager. IEEE, 2006.

[5] Tr 102 638 - intelligent transport systems (its); vehicular communications; basic set of applications; definitions. ETSI, 2009.

[6] 1609.3-2010 - ieee standard for wireless access in vehicular environments (wave) - networking services. IEEE, 2010.

[7] 802.11p-2010 - ieee standard for information technology– 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 Standard, 2010.

[8] Ts 102 731 - intelligent transport systems (its); security; security services and architecture. ETSI, 2010.

[9] En 302 665 - intelligent transport systems (its); communications architecture. ETSI, 2011.

[10] Ts 102 636-5-1 - intelligent transport systems (its) - vehicular communications - geonetworking - part 5: Transport protocols - sub-part 1: Basic transport protocol. ETSI, 2011.

[11] Ts 102 636-6-1 - intelligent transport systems (its); vehicular communications; geonetworking; part 6: Internet integration; sub-part 1: Transmission of ipv6 packets over geonetworking protocols. ETSI, 2011.

[12] Ts 102 687 - intelligent transport systems (its) - decentralized congestion control mechanisms for intelligent transport systems operating in the 5 ghz range - access layer part. ETSI, 2011.

[13] 1609.0-2013 - ieee guide for wireless access in vehicular environments (wave) - architecture. IEEE, 2013.

[14] Ts 102 894-1 - intelligent transport systems (its); users and applications requirements; part 1: Facility layer structure, functional requirements and specifications. ETSI, 2013.

[15] En 302 637-2 - intelligent transport systems (its) - vehicular communications - basic set of applications - part 2: Specification of cooperative awareness basic service. ETSI, 2014.

[16] En 302 895 - intelligent transport systems (its); vehicular communications; basic set of applications; local dynamic map (ldm). ETSI, 2014.

[17] Ts 102 723-1 - intelligent transport systems (its); osi cross-layer topics; part 1: Architecture and addressing schemes. ETSI, 2014.

[18] 1609.2-2016 - ieee standard for wireless access in vehicular environments–security services for applications and management messages. IEEE, 2016.

[19] 1609.4-2016 - ieee standard for wireless access in vehicular environments (wave) – multichannel operation. IEEE, 2016.

[20] Mohammad S Almalag, Stephan Olariu, and Michele C Weigle. Tdma cluster-based mac for vanets (tc-mac). In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–6. IEEE, 2012.

[21] Alan D Amis, Ravi Prakash, Thai HP Vuong, and Dung T Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 32–41. IEEE, 2000.

[22] Adrian Andronache, Matthias R Brust, and Steffen Rothkugel. Multimedia content distribution in hybrid wireless networks using weighted clustering. In *Proceedings of the 2nd ACM international workshop on Wireless multimedia networking and performance modeling*, pages 1–10. ACM, 2006.

[23] Rasmeet S Bali, Neeraj Kumar, and Joel JPC Rodrigues. Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions. *Vehicular communications*, 1(3):134–152, 2014.

[24] Soheila V Bana and Pravin Varaiya. Space division multiple access (sdma) for robust ad hoc vehicle communication networks. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 962–967. IEEE, 2001.

[25] S. Barker. Consumer connected cars: Telematics, insurance and in-vehicle applications 2015-2020. Juniper Research, December 2015.

[26] Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic. *Mobile ad hoc networking*. John Wiley & Sons, 2004.

[27] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.

[28] Abderrahim Benslimane, Tarik Taleb, and Rajarajan Sivaraj. Dynamic clustering-based adaptive mobile gateway management in integrated vanet—3g heterogeneous wireless networks. *Selected Areas in Communications, IEEE Journal on*, 29(3):559–570, 2011.

[29] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. Adhoc mac: new mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. *Wireless Networks*, 10(4):359–366, 2004.

[30] Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang. Forwarding group multicast protocol (fgmp) for multihop, mobile wireless networks. *Cluster Computing*, 1(2):187–196, 1998.

[31] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *proceedings of IEEE SICON*, volume 97, pages 197–211, 1997.

[32] Francesco Chiti, Romano Fantacci, Enrico Dei, and Zhu Han. Context aware clustering in vanets: A game theoretic perspective. In *Communications (ICC), 2015 IEEE International Conference on*, pages 6584–6588. IEEE, 2015.

[33] Jean-Charles de Borda. Mémoire sur les élections au scrutin, histoire de l'académie royale des sciences. *Paris, France*, 1781.

[34] Efi Dror, Chen Avin, and Zvi Lotker. Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 1–8. IEEE, 2011.

[35] Efi Dror, Chen Avin, and Zvi Lotker. Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks. *Ad Hoc Networks*, 11(7):2002–2015, 2013.

[36] Ghayet El Mouna Zhioua, Nabil Tabbane, Houda Labiod, and Sami Tabbane. A fuzzy multi-metric qos-balancing gateway selection algorithm in a clustered vanet to lte advanced hybrid cellular network. *Vehicular Technology, IEEE Transactions on*, 64(2):804–817, 2015.

[37] Anthony Ephremides, Jeffrey E Wieselthier, and Dennis J Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73, 1987.

[38] TR ETSI. 102 863 intelligent transport systems (its). *Local Dynamic Map (LDM)*, 2012.

[39] IBM Institute for Business Value. Advancing mobility: The new frontier of smarter transportation. Executive Report, November 2010.

[40] Julian Garbiso, Ada Diaconescu, Marceau Coupechoux, and Bertrand Leroy. Dynamic cluster size optimization in hybrid cellular-vehicular networks. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 557–563. IEEE, 2016.

[41] Julian Garbiso, Ada Diaconescu, Marceau Coupechoux, and Bertrand Leroy. Auto-adaptive multi-hop clustering for hybrid cellular-vehicular networks. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*. IEEE, 2017.

[42] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.

[43] Javad Ghaderi, Liang-Liang Xie, and Xuemin Shen. Hierarchical cooperation in ad hoc networks: Optimal clustering and achievable throughput. *Information Theory, IEEE Transactions on*, 55(8):3425–3436, 2009.

[44] BMW Group. Bmw connecteddrive store - https://www.bmw-connecteddrive.co.uk/app/gb/index.html#/portal/store (last retrieved on october 15th, 2017).

[45] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000.

[46] John A Hartigan. Clustering algorithms. 1975.

[47] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, pages 10–pp. IEEE, 2000.

[48] Michael A Heller. The tragedy of the anticommons: property in the transition from marx to markets. *Harvard law review*, pages 621–688, 1998.

[49] Rafia Inam, Nicolas Schrammar, Keven(Qi) Wang, Athanasios Karapantelakis, Leonid Mokrushin, Aneta Vulgarakis Feljan, and Elena Fersman. Feasibility assessment to realise vehicle teleoperation using cellular networks. In *Proceedings of the 19th IEEE Intelligent Transportation Systems Conference (ITSC) 2016*.

[50] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)*, 23(3):219–252, 2005.

[51] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE communications surveys & tutorials*, 13(4):584–616, 2011.

[52] Bernhard Kloiber, Thomas Strang, Matthias Röckl, De Ponte-Müller, et al. Performance of cam based safety applications using its-g5a mac in high dense scenarios. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 654–660. IEEE, 2011.

[53] Weiwei Li, Ali Tizghadam, and Alberto Leon-Garcia. Robust clustering for connected vehicles using local network criticality. In *Communications (ICC), 2012 IEEE International Conference on*, pages 7157–7161. IEEE, 2012.

[54] Sung-Han Lin, Junn-Yen Hu, Cheng-Fu Chou, Ing-Chau Chang, and Chien-Chun Hung. A novel social cluster-based p2p framework for integrating vanets with the internet. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6. IEEE, 2009.

[55] Changyue Liu, Supeng Leng, Kun Yang, Liang Tong, and Ke Zhang. A cooperative-pricing-based access selection mechanism for vehicular heterogeneous networks. In *Communications and Networking in China (CHINACOM), 2014 9th International Conference on*, pages 181–186. IEEE, 2014.

[56] Ning Lu, Ning Zhang, Nan Cheng, Xuemin Shen, Jon W Mark, and Fan Bai. Vehicles meet infrastructure: Toward capacity–cost tradeoffs for vehicular access networks. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1266–1277, 2013.

[57] Mercedes-Benz. Mecedes-benz mbrace service pricing - https://www.mbusa.com/mercedes/mbrace (last retrieved on october 15th, 2017).

[58] General Motors. Onstar service pricing - https://www.onstar.com/us/en/plans-pricing.html (last retrieved on october 15th, 2017).

[59] Hassan Omar, Weihua Zhuang, and Li Li. Vemac: A tdma-based mac protocol for reliable broadcast in vanets. *Mobile Computing, IEEE Transactions on*, 12(9):1724–1736, 2013.

[60] Elinor Ostrom, Roy Gardner, and James Walker. *Rules, games, and common-pool resources*. University of Michigan Press, 1994.

[61] Ayfer Özgür, Olivier Lévêque, and David NC Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *Information Theory, IEEE Transactions on*, 53(10):3549–3572, 2007.

[62] Abhay K Parekh. Selecting routers in ad-hoc wireless networks. In *Proceedings of the SBT/IEEE International Telecommunications Symposium*, volume 204, 1994.

[63] Jeremy Pitt, Dídac Busquets, and Sam Macbeth. Distributive justice for self-organised common-pool resource management. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 9(3):14, 2014.

[64] Guillaume Rémy, Sidi-Mohammed Senouci, François Jan, and Yvon Gourhant. Lte4v2x: Lte for a centralized vanet organization. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[65] Nicholas Rescher. *Fairness: Theory and practice of distributive justice*. Transaction Publishers, 2002.

[66] Ismail Salhi, Mohamed Oussama Cherif, and Sidi-Mohammed Senouci. A new architecture for data collection in vehicular networks. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, 2009.

[67] Rajarajan Sivaraj, Aravind Kota Gopalakrishna, M Girish Chandra, and P Balamuralidhar. Qos-enabled group communication in integrated vanet-lte heterogeneous wireless networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 17–24. IEEE, 2011.

[68] Christoph Sommer, David Eckhoff, Reinhard German, and Falko Dressler. A computationally inexpensive empirical model of ieee 802.11 p radio shadowing in urban environments. In *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, pages 84–90. IEEE, 2011.

[69] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, January 2011.

[70] Christoph Sommer, Stefan Joerer, and Falko Dressler. On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation. In *4th IEEE Vehicular Networking Conference (VNC 2012)*, pages 64–69, Seoul, Korea, November 2012. IEEE.

[71] Lingyang Song, Dusit Niyato, Zhu Han, and Ekram Hossain. Game-theoretic resource allocation methods for device-to-device communication. *Wireless Communications, IEEE*, 21(3):136–144, 2014.

[72] Lung-Chih Tung, Jorge Mena, Mario Gerla, and Christoph Sommer. A cluster based architecture for intersection collision avoidance using heterogeneous networks. In *Ad Hoc Networking Workshop (MED-HOC-NET), 2013 12th Annual Mediterranean*, pages 82–88. IEEE, 2013.

[73] 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.

[74] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. Vesca: Vehicular stable cluster-based data aggregation. In *Connected Vehicles and Expo (ICCVE), 2014 International Conference on*, pages 1080–1085. IEEE, 2014.

[75] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. Multihop-cluster-based ieee 802.11 p and lte hybrid architecture for vanet safety message dissemination. *IEEE Transactions on Vehicular Technology*, 65(4):2621–2636, 2016.

[76] András Varga. Discrete event simulation system. In *Proc. of the European Simulation Multiconference (ESM'2001)*, 2001.

[77] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 60. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[78] Samo Vodopivec, Janez Bešter, and Andrej Kos. A survey on clustering algorithms for vehicular ad-hoc networks. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, pages 52–56. IEEE, 2012.

[79] Jong-Woon Yoo and Kyu Ho Park. A cooperative clustering protocol for energy saving of mobile devices with wlan and bluetooth interfaces. *IEEE Transactions on Mobile Computing*, 10(4):491–504, 2011.

[80] Ossama Younis and Sonia Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, 3(4):366–379, 2004.

[81] Jane Yang Yu and Peter Han Joo Chong. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 7(1):32–48, 2005.