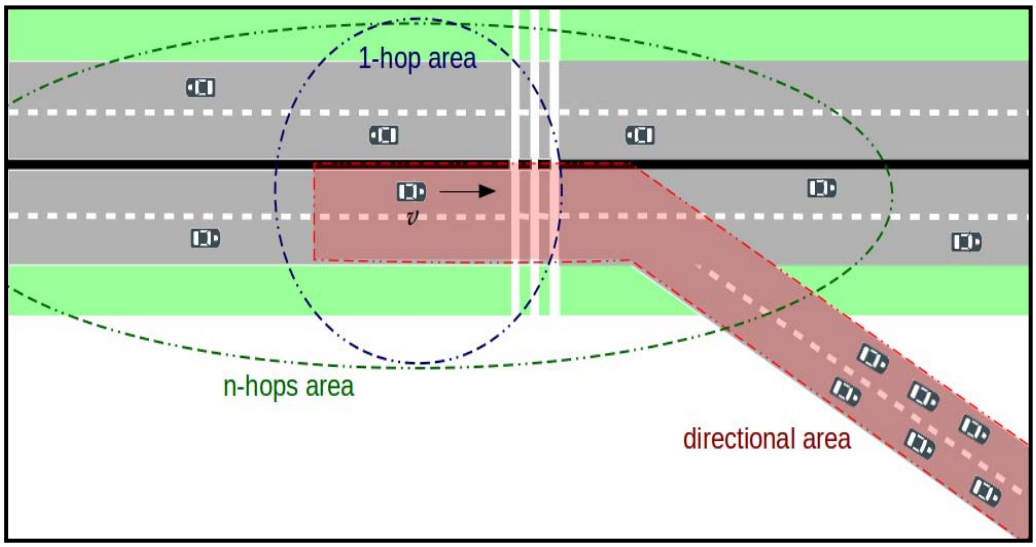
Adaptive solutions for data sharing in vehicular networks

Hermes Pimenta de Moraes Junior

Par **Hermes PIMENTA DE MORAES JUNIOR**

Adaptive solutions for data sharing in vehicular networks

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 4 mai 2018
Spécialité : Technologies de l'Information et des Systèmes :
Unité de recherche Heudyasic (UMR-7253)
D2417

Adaptive solutions for data sharing in vehicular networks

Hermes PIMENTA DE MORAES JUNIOR

Spécialité : Technologies de l'Information et des Systèmes

Proposition de jury:

Rapporteurs:

Hervé RIVANO
Professeur des universités
INSA Lyon

Marcelo DIAS DE AMORIM
Directeur de Recherche
Sorbonne Université - LIP6

Examinateurs:

Fabio D'ANDREAGIOVANNI
First Class Research Scientist (CR1)
Univ. de Tech. de Compiègne

Nathalie MITTON
Directeur de recherche
INRIA Lille

Véronique CHERFAOUI
Professeur des universités
Univ. de Tech. de Compiègne

Directeur de Thèse:
Bertrand DUCOURTHIAL
Professeur des universités
Univ. de Tech. de Compiègne

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

Soutenue le 4 mai 2018

To my lovely family: parents, wife and sons.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Bertrand Ducourthial for the continuous support of my Ph.D study and related research, for his patience and motivation.

Besides my advisor, I would like to thank my thesis committee: Professors Hervé Rivano and Véronique Cherfaoui, Research Directors Marcelo Dias de Amorin and Nathalie Mitton and the CNRS Researcher Fabio D'Andreagiovanni for their insightful comments and encouragement.

My sincere thanks to all Brazilian friends of Compiègne (Marcio and Ana Fuckner, Danilo, Giovanni, Natasha) for the help during my first months in France. Thanks specially to Marcio Costa for the discussions on math topics of my research. Thanks Wendell Diniz for the fast course on MatPlotLib and Pandas. My charts are better than ever ;-).

I thank also my colleagues from the Heudiasyc laboratory: Ahmet Sekercioglu, Mohamed Mkaddem, Youcef Amarouche, Abderrahim Sahli and Mateus Habermann, for the pleasant coffee-breaks.

Résumé

Dans le cadre des systèmes de transport intelligents (STI), les véhicules peuvent avoir beaucoup de capteurs (caméras, lidars, radars, etc.) et d'applications (évitement des collisions, surveillance du trafic, etc.) générant des données. Ils représentent alors une source d'information importante. Les applications locales peuvent augmenter considérablement leur efficacité en partageant une telle information au sein du réseau. La précision des données, la confiance et la pertinence peuvent être vérifiées lors de la réception de données provenant d'autres nœuds. Par conséquent, nous croyons qu'une question importante à répondre dans ce contexte est: "Comment partager efficacement les données dans un tel environnement?"

Le partage de données est une tâche complexe dans les réseaux dynamiques. De nombreuses problèmes telles que les connexions intermittentes, la variation de la densité du réseau et la congestion du médium de communication se posent. Une approche habituelle pour gérer ces problèmes est basée sur des processus périodiques. En effet, un message envoyé plusieurs fois peut atteindre sa destination même avec des connexions intermittentes et des réseaux à faible densité. Néanmoins, dans les réseaux à haute densité, ils peuvent entraîner une congestion du médium de communication.

Dans cette thèse, nous abordons le problème du partage de données dans des réseaux dynamiques en nous appuyant sur des *horizons de pertinence*. Un horizon est défini comme une zone dans laquelle une information devrait être reçue. Nous commençons par nous concentrer sur le partage de données au sein des voisins directs (à 1 saut de distance). Ensuite, nous proposons une solution pour construire une carte des voisins, centrée sur le nœud `ego`, dans un horizon à n sauts. Enfin, nous relâchons la définition de l'horizon pour la définir de façon dynamique, où différents éléments de données peuvent atteindre des distances différentes (sauts).

En ce qui concerne la solution pour les horizons à 1 saut, notre technique adaptative prend en compte la dynamique des nœuds et la charge du réseau. Afin d'assurer une diffusion efficace des données dans différents scénarios, la fréquence d'envoi des messages est définie en fonction des mouvements des véhicules et d'une estimation du taux de perte du réseau.

Après, nous nous concentrons sur la carte des voisins jusqu'à n sauts de distance.

Comme la communication avec des nœuds éloignés apporte des problèmes supplémentaires (actions de transfert, retards plus importants, informations périmées), une évaluation de confiance des nœuds identifiés et une estimation de fiabilité du chemin vers chaque voisin sont ajoutées à la carte.

Au lieu d'exécuter des processus de diffusion séparés, notre troisième contribution porte sur une stratégie de coopération dont l'objectif principal est de diffuser des données tout en satisfaisant la plupart des nœuds. À cette fin, une trame unique est transmise de nœud en nœud. Sa charge utile est mise à jour localement afin qu'elle contienne les éléments de données les plus pertinents en fonction de certains critères (par exemple, urgence, pertinence). Une telle stratégie définit ainsi un horizon centré sur les données.

Nous validons nos propositions au moyen d'émulations de réseaux réalistes. De toutes nos études et des résultats obtenus, nous pouvons affirmer que notre approche apporte des perspectives intéressantes pour le partage de données dans des réseaux dynamiques comme les VANET.

Mots clés: VANETs, découverte de voisin, problème de congestion des réseaux, carte de voisins, évaluation de confiance, perception coopérative, diffusion coopérative de données

Abstract

In the context of Intelligent Transportation Systems - ITS, vehicles may have a lot of sensors (e.g. cameras, lidars, radars) and applications (collision avoidance, traffic monitoring, etc.) generating data. They represent then an important source of information. Local applications can significantly increase their effectiveness by sharing such an information within the network. Data accuracy, confidence and pertinence can be verified when receiving data from other nodes. Therefore, we believe that an important question to answer in this context is: "How to efficiently share data within such an environment?"

Data sharing is a complex task in dynamic networks. Many concerns like intermittent connections, network density variation and communication spectrum congestion arise. A usual approach to handle these problems is based on periodic processes. Indeed, a message sent many times can reach its destination even with intermittent connections and low density networks. Nevertheless, within high density networks, they may lead to communication spectrum scarcity.

In this thesis we address the problem of data sharing in dynamic networks by relying in so-called *horizons of pertinence*. A horizon is defined as an area within which an information is expected to be received. We start focusing on data sharing within direct neighbors (at 1-hop of distance). Then we propose a solution to construct a map of neighbors, centered in the ego-node, within a horizon of n-hops. Finally, we relax the horizon definition to a dynamic defined one where different data items may reach different distances (hops).

Regarding the solution for 1-hop horizons, our adaptive technique takes into account nodes' dynamics and network load. In order to ensure an effective data dissemination in different scenarios, the sending messages frequency is defined according to vehicles movements and an estimation of the network loss rate.

Following, we focus on the map of neighbors up to n-hops of distance. As communication with distant nodes brings additional concerns (forwarding actions, larger delays, out-of-date information), a trust evaluation of identified nodes and a reliability estimation of the multi-hop path to each neighbor is added to the map.

Instead of running separated disseminating processes, our third contribution deals with a cooperative strategy with the main goal of disseminating data while

satisfying most of the nodes. For this purpose a unique frame is forwarded from node to node. Its payload is locally updated so that it contains the most relevant data items according to some criteria (e.g. urgency, relevance). Such a strategy defines thus a data-centered horizon.

We validate our proposals by means of realistic network emulations. From all our studies and achieved results we can state that our approach brings interesting insights for data sharing in dynamic networks like VANETs.

Keywords: VANETs, neighbor discovery, broadcast storm problem, neighborhood map, trust evaluation, cooperative perception, cooperative data dissemination

Contents

List of figures

List of tables

Publications

International conferences

- H. P. de Moraes and B. Ducourthial. “Adaptive inter-messages delay in vehicular networks”. In the IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, NY, 2016, pp. 1-8.
- H. P. de Moraes and B. Ducourthial. “Cooperative Neighborhood Map in VANETs”. In Proceedings of the Symposium on Applied Computing, SAC’18, 2018, New York, NY, USA. ACM.
- H. P. de Moraes and B. Ducourthial. “A strategy for data sharing in VANETs”. In the process of submission.

National communications

- H. P. de Moraes and B. Ducourthial. “Cooperative Neighborhood Map in VANETs”. Submitted to Algotel, Roscoff, France, 2018; answer expected to April 17th, 2018.

Introduction

Contents

1.1 Vehicular ad-hoc Networks

The current technological development has provided the inclusion of new features and technologies by many devices. Vehicles have been incorporating new sensors and applications over several years. For example, it is common to find cars with GPS, cameras, lidars, and radars, offering services like stability control, self-parking, auto-braking and many others.

Although they have large impact on the users, the mentioned technologies do not promote interaction among vehicles. The next improvement, or technological leap, searches - through communication among different vehicles - the ambitious goal of an intelligent transportation system (ITS) [ETSI EN 302 637-3, 2014]. In an ITS, it is expected that vehicles are able to exchange messages in order to enable services such as cooperative traffic monitoring, collision prevention systems and fleet tracking [Cherif et al., 2010]. In addition to the traffic services, provision of Internet access or data sharing may be aimed [Kamakura and Ducourthial, 2014].

Communication systems between vehicles are referred to as vehicular networks, or VANETs. A VANET could be composed of cars, trucks, buses and possibly a fixed infrastructure in the roadside. Although it is a wireless network, a vehicular network has peculiar characteristics that can bring great challenges for a large-scale deployment, such as intermittent communication, very dynamic scenarios, scale and density variation [Karagiannis et al., 2011, Kaiwartya et al., 2016].

1.1.1 VANETs Architectures

The architecture of a network defines how its elements are organized. In vehicular networks there are three well-defined architectures: ad-hoc, infrastructure and hybrid.

Vehicular ad-hoc networks are composed only by mobile nodes, e.g. cars, trucks, buses, etc. which communicate with one another directly. Also called V2V (vehicle-to-vehicle), this network presents challenges concerning connectivity due to high mobility of nodes, and routing when a node needs to forward messages to other distant nodes.

In order to mitigate these problems, vehicle-to-infrastructure (V2I) communication makes use of static devices distributed along the transit route. Such devices, known as RSUs (road side units), are traffic centralizers that can intermediate communication between mobile nodes, or communication with infrastructure-based networks such as Internet. The connectivity expansion in this kind of network depends on the quantity of static nodes deployed, which may lead to prohibitive costs.

The hybrid architecture (V2X or vehicle-to-anything) can be used as an intermediate solution. In hybrid networks, nodes communicate directly (V2V) in areas where there is no achievable RSUs. Once in areas covered by RSUs, vehicles communicate through this centralizing device (V2I). This approach allows an increase on network connectivity without excessively raising the cost of deployment [Giordano and Reggiani, 2014, Katsaros et al., 2013].

1.1.2 Standards for VANETs

In 2004, the IEEE (Institute of Electrical and Electronics Engineers) began a process of standardization for vehicular networks. Initiated as part of the IEEE 802.11 group, the standard was known as IEEE 802.11p WAVE (Wireless Access in the Vehicular Environment). The WAVE architecture is specified in other six documents: IEEE P1609.1, IEEE P1609.2, IEEE P1609.3, IEEE P1609.4, IEEE 802.11 and IEEE 802.11p. The IEEE 802.11p standard provides definitions for lower layers as Physical Layer and MAC sub-layer, based on the IEEE 802.11a standard. The WAVE architecture (Figure 1.1) also defines, through the IEEE 1609 family, standards with focus on the upper layers including security issues, use of multiple channels and an alternative to the IP protocol [802.11p, 2010, Vivek et al., 2017].

The IEEE 1609 family aims to provide a communication and interface standard which can be used for development of V2V, V2I or V2X applications. This pattern is important to keep the interoperability among applications developed by different

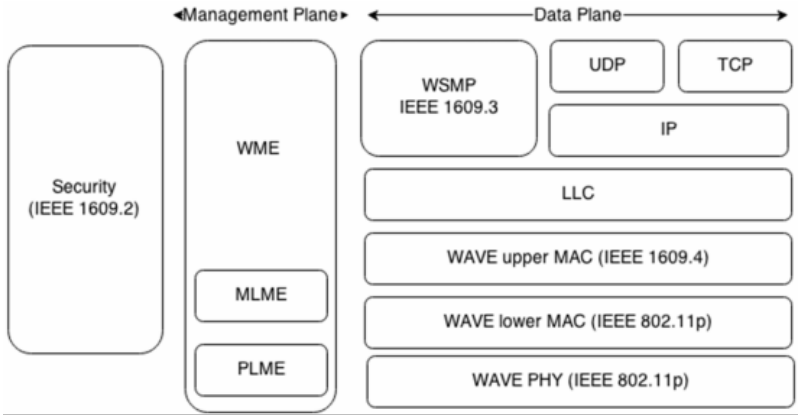


Figure 1.1 – WAVE architecture [Vivek et al., 2017]

car manufacturers or independent developers.

Additional standards have been published by the European Telecommunications Standards Institute (ETSI). Among many specifications produced by ETSI, we can highlight the multi-part definition of a Basic Set of Applications for Intelligent Transportation Systems. The first part of this document specifies functional requirements for the applications supported by vehicular communication systems [ETSI TS 102 637-1, 2010].

The second part of the document contains a "Specification of Cooperative Awareness Basic Service" [ETSI Ts 102 637-2, 2010]. The CAM service, initially developed by the European Car-to-Car Communication Consortium [Car2Car Communication Consortium, 2008], is intended to keep road users and road side infrastructure aware of other's position, dynamics and attributes.

"Specifications of Decentralized Environmental Notification Basic Service" is the subject of the third part of the document. A Decentralized Environmental Notification Message (DENM) is used by ITS applications to alert road users of a detected event [ETSI EN 302 637-3, 2014]. An event can be a road hazard, a driving environment concern or a traffic condition situation.

Standards and specifications from both IEEE and ETSI are mainly based on Dedicated Short-range Communication (DSRC) technologies like 802.11p. Such technologies present, however, relevant constraints. The achievable communication range is quite limited (up to 1000 m in theory but much less in practice [Breu et al., 2014]). Their connectivity suffers severely in non line-of-sight (NLOS) environments, e.g. in the presence of obstacles. They suffer also in high density scenarios where too many packet collisions may take place [Shen et al., 2013].

Long-term-evolution (LTE) technologies like 3G or 4G could help to manage such problems, offering a really wider communication range. Nevertheless, they do not support ad-hoc communication (e.g. vehicle to vehicle) and also present

congestion problems in high-density environments. Additionally, using an infrastructure network adds prohibitive delays when considering traffic safety applications [Gharba et al., 2017].

Long-term-evolution technologies of fifth generation (5G) however, are being developed with both infrastructure and ad-hoc communication (or device-to-device) strategies in mind. Hence, with direct communication (without using the infrastructure) and wide communication range, such a technology has been seriously considered as an alternative to 802.11p protocol [Gharba et al., 2017]. The long-term-evolution-vehicle (LTE-V) is a proposal in this direction. It offers vehicles-to-anything (V2X) communication through two radio interfaces: one for the vehicle-to-infrastructure communications (cellular interface); one for vehicle-to-vehicle communications (based on direct LTE sidelink) [Molina-Masegosa and Gozalvez, 2017].

1.1.3 VANETs Applications

A countless number of applications can be envisioned by means of VANETs. A division into two main categories (traffic safety and infotainment) was adopted in this text to help on describing some examples. Each category presents its own specific goals and requirements.

Traffic Safety Applications

Increasing traffic safety was the first goal of VANETs and the main reason for their development. These applications focus on to diminish the number of accidents and/or the damage caused by them. The information exchanged among nodes (vehicles, RSUs) might be used to warn the driver or by an autonomous system which act in response. Due to its relevance, safety messages have hard requirements of latency and reliability [Javed and Khan, 2014, Ucar et al., 2016].

In this context, applications for collision avoidance, emergency messages dissemination, lane change warning and traffic signal violation, are common. Such applications have been studied by several industry/government consortiums like the Crash Avoidance Metrics Partnership (CAMP) in USA, the Car2Car Communication Consortium in Europe, the Advanced Safety Vehicle (ASV) in Japan [Hartenstein and Laberteaux, 2009].

The projects DAPAD[1] and CoMoSef[2] also have relevant contributions in this context. The Co-operative Mobility Services of the Future project (CoMoSef) aims at increasing traffic safety and efficiency through distributed data collection, fusion

[1] https://dapad.hds.utc.fr/
[2] http://www.comosef.eu/?q=node/2

and retransmission. Safety margin warnings, accurate and reliable road weather information, friction monitoring and forecasting are some examples of applications developed within CoMoSef research.

The DAPAD (Distributed and Augmented vehicle Perception to support Autonomous Driving) project uses a cooperative/distributed perception in order to improve vehicles' comprehension of complex and highly dynamic scenes. The research developed includes: Distributed scene modeling; Multimodal perception of dynamic driving environments; Data consistency and coherency.

Infotainment Applications

With the focus of providing useful information and entertainment for drivers and passengers, these applications can cover: diffusion of available parking places; cross-roads aid systems; point-of-interest localization as service-stations, hotels, supermarkets; Internet access; multimedia content access and sharing; among many others [Lin et al., 2017, Lèbre et al., 2014]. To the contrary of safety applications, here a large bandwidth is usually preferable than small latency and reliability [Sarakis et al., 2016].

A multi-Criteria Ad hoc Real-time Streaming (CARS) service for VANETs is proposed by [Wisitpongphan and Bai, 2013]. CARS is a framework which searches for paths through the network capable of attending minimum QoS requirements. It relies on requirements adequacy and not shorter or faster paths.

A standard for providing information services for safe and secure truck parking places is proposed by [Melo-Castillo et al., 2017]. Such a need arises due to the usually insufficient number of parking facilities in Europe which forces drivers to park in non-secured or unsafe zones. This proposal extends the DATEX II[3] standard for truck parking information dissemination.

1.2 Research Context

This work has been conducted under the SCOP group, one of the three working groups of Heudiasyc[4] (*HEUristique et DIAgnostic des SYstèmes Complexes*) research laboratory. The Heudiasyc laboratory promotes research in Control, Robotics, Decision-making and Computing in order to answer socio-economic challenges in Security, Transports, Environment and Health.

[3]DATEX is a standard for traffic information exchange among traffic centers, service providers and traffic operators in Europe. The interested reader may refer to http://www.datex2.eu/ for additional information.

[4]https://www.hds.utc.fr

Furthermore, this work has been developed in interests of the projects:

- Robotex[5] - a French National Robotics platform
- TOREDY which focuses on algorithms characterization and evaluation regarding their capacity to work within dynamic networks as VANETs or networks of drones
- Arcovin (*Architecture Coopérative pour la Ville Intelligente*), in a cooperation with the city of Compiègne.
- DAPAD and CoMoSef, as described earlier.

The Airplug framework [Ducourthial, 2013] is an important development platform within the SCOP group and the Heudiasyc laboratory. It provides tools for prototyping and experimenting solutions in the context of dynamic networks. By means of Airplug, many applications have been developed by the team during the last few years. Some of them are highlighted here:

HOP implements a conditional transmission service [Ducourthial et al., 2007]. It selects relay and receiver nodes by means of conditions, instead of addresses. Hence, only the nodes that fulfill the conditions will forward or give the message to their application layers. Conditions could be "being in an area x", "those who are behind the sender", or even "those who have the identity x".

GRP constructs and maintains, as long as possible, a group of nodes satisfying a constraint on the diameter. GRP does not rely on a specific node, that may move and leave, but rather searches for stabilizing a view (local knowledge about the group) among all members of the group [Ducourthial et al., 2010].

PTH allows maintaining a path between two mobile nodes going apart after a direct connection. PTH aims at implementing a unicast communication within dynamic networks [El Ali and Ducourthial, 2011].

GTW was built with the objective of providing Internet access to other Airplug applications. GTW searches for available gateways, or access points, to communicate with the infrastructure side of the network. In VANETs nodes, GTW can be used to forward messages from local applications to web servers in the Internet [Kamakura and Ducourthial, 2014].

[5] http://www.femto-st.fr/en/The-institute/Projects/French-National-Robotics-platform-ROBOTEX

COL defines a data collection process controlled by a maximal duration in time and a maximal distance in hops. So, data collection is performed until all n-hops neighbors are reached or the maximum time expires [Dieudonné et al., 2012].

MET is the implementation of a generic framework intended to handle belief functions. The application allows the usage of user-defined frames of discernment to compute direct and distributed confidences on the exchanged data. Thus, MET instances exchange messages, computes the direct and the distributed confidences on the data. Next, MET shares its data along with the computed distributed confidence with neighbors [Ducourthial and Cherfaoui, 2016].

Concerning experiments and evaluation, the Airplug Emulator (EMU) is capable of emulating networks with vehicles running individually and exchanging messages in a realistic way [Buisset et al., 2010]. Since Airplug was proposed to reduce the gap between the simulation and the road testbed, applications in EMU are identical to those used in real tests when it is embedded on On-Board-Units. The only difference is the way messages are exchanged: during emulation, they are not sent over the network device but forwarded to nodes and applications, possibly with losses or delays.

From the research and applications development conducted, the team counts with experimented solutions for data collect, data transmission, data fusion, among others. Nevertheless, there is no clear reasoning about data sharing strategies. This thesis is conducted then in this continuity with an experimental component.

1.3 Challenges and Objectives

The mobility behavior of VANETs turns data exchange potentiality into few opportunities. A node might keep long periods without, or even never have, a reliable connection to the infrastructure network. Some nodes can meet each other for few small periods during the day. A requesting device may never have a complete operating path to the answering one [Boldrini et al., 2010]. Therefore, intermittent connections, neighborhood instability and absence of a topology knowledge represent some challenges to overcome within such environments.

Regardless the specific approach adopted, some general design rules that should be taken into account when addressing these challenges can be highlighted:

1. Topology: With communicating links constantly appearing and disappearing, the network topology becomes quite unstable. Thus, a distributed algorithm should not assume any characteristic regarding the topology.

2. Non-local knowledge: Within dynamic networks, relying on non-local knowledge is inappropriate. For example, keeping the information about a specific node position, or its distance in hops presents a too heavy cost and may be useless. It would require sending messages through multi-hop paths which may fail due to many possibilities: the path is not valid anymore; the receiving node is not there when the message arrives; the sender may have moved before receiving the answer. Moreover, even if the communication has succeeded, the answer might be incorrect due to the movement of the interested node.

3. Addresses: A network address is generally composed of two parts: a unique identification of the node and it's position in the network. Nevertheless, addresses are unstable in dynamic networks. To obtain a node's address, it would be required to send and receive messages through multi-hop paths. Instead, it is preferable to use address with a semantic meaning (e.g. "near the sender", "within a defined area").

Even though the described rules can help on working with dynamic networks, they do not answer all questions and cope with all problems. Additional concerns arise when considering security, network transmission capacity, data format, quality-of-service guarantee and data sharing.

The research developed in this thesis is focused on data sharing within VANETs. In this context, some questions to be answered are:

What to share? In a direct answer, the most relevant data items have to be shared. Nevertheless, data relevance is related to many other characteristics like: reliability, confidence, temporal and geographic dependence. So, before choosing what to share, the available data has to be assessed according to many parameters.

In which frequency? Sending frequency may become a major concern when considering data sharing. Using low frequencies leads to the lack of or outdated information. On the other hand, high frequencies might overload the communication spectrum, leading to message loss. An efficient environment knowledge construction must balance the precision desired with the number of messages sent.

Up to what distance from the source? The shared data should reach nodes to who it is relevant. Within VANETs however, such a classification has to consider time and space. For instance, data about accidents is relevant in specific areas and during a time period. Additionally, large accidents should

be disseminated to wider areas than small ones. An area of interest should be defined for keeping the dissemination within. Finally, it would be useless to share the data beyond the area limits.

The research conducted here search to answer or, at least, to offer insights about such questions. The approach adopted was to address the problem in three steps. First, investigating dissemination in the vicinity with attention to direct neighbors identification and messaging frequency. Second, extrapolating such a dissemination to wider areas, reaching an arbitrary distance of n-hops from the source while evaluating trust and reliability of distant nodes. Finally, searching for a dynamic data-based dissemination area definition where each data item may reach different distances from the source.

1.4 Text Structure

The present chapter introduces the reader to the context, challenges and main problems addressed in the research developed. In each following chapter, a specific topic of the main problem is presented in details, a solution is proposed, evaluated and achieved results are described. Hence, this text continues with the structure:

Chapter 2 presents a state of the art review focused on data sharing in vehicular networks.

Chapter 3 describes a proposal for an Adaptive Neighbor Discovery (AND) algorithm intended to leverage data sharing processes among direct neighbors (1-hop of distance). AND defines the inter-messages-delay of the discovery process with basis on nodes' dynamics and on a network loss-rate estimation. The main objective is to achieve an efficient discovery of direct neighbors while saving the wireless communication spectrum.

Chapter 4 presents the CNM - Cooperative Neighborhood Map - algorithm. CNM is proposed with the purpose of constructing a map of neighbors. It develops a cooperative approach to identify nodes up to n hops of distance while associating to each of them a trust value. Thus, a node is capable of trusting or mistrusting data received from specific neighbors. CNM estimates yet the reliability of the path between the ego-node and each of its neighbors. This information is specially relevant when offering services. In particular, a remote service is usable only if the communication is correct.

Chapter 5 describes an integrated architecture for disseminating relevant information within VANETs. The Relevant Information Frame (RIF) architecture

first searches for data merging/fusion possibilities, to then encourage the dissemination of the resulting items in one unique process. To enable the combined dissemination, RIF selects the most relevant data items (in the scope of a node), put them into a frame and disseminate it in the vicinity. As the process is repeated at every node, some data items will continue to be chosen for further dissemination, reaching distant areas, while others will be kept in the vicinity. Hence, RIF yields a dynamic area of dissemination defined hop-by-hop and by data item.

Chapter 6 gives concluding remarks and perspectives for future works.

Data sharing in VANETs

Contents

2.1 Introduction

In the context of Intelligent Transportation Systems - ITS, it is expected that a vehicle is able to generate a considerable amount of data and share it through vehicular networks. For instance, making cars communicate with one another has a great potential to make traffic safer, more efficient and more pleasant.

Before bringing this potential to reality though, a lot of challenges have to solved. As described in Chapter 1, different applications have different requirements to work properly, presenting thereby different challenges [Lin et al., 2017, Lèbre et al., 2014]. Safety applications (e.g. collision avoidance, blind spot awareness, line change signalization) require high sending frequency and small delays. Infotainment applications (e.g. parking availability, file sharing, gaming) require a large bandwidth but accept larger delays.

Due to inherent characteristics of VANETs, common used approaches for communication between cars involve flooding, store-carry-and-forward and the use of infrastructure networks. Flooding provides good results in sparse networks but easily cause broadcast-storm problems in scenarios with high node density. Store-carry-and-forward approaches are chosen to keep the data dissemination

process operational within sparse scenarios although at the cost of large delays. Infrastructure networks require the deployment of infrastructure devices which may lead to prohibitive costs.

The present chapter brings a state of the art review about data sharing in VANETs. Its structure was defined in order to help answering the three questions proposed in the Introduction (Chapter 1). Hence, Section 2.2 clarifies concepts of data sharing and the relation among them; Section 2.3 describes approaches for data evaluation and selection, addressing the question "What to share?"; the frequency chosen for information dissemination ("In which frequency to share?") is addressed in Section 2.4; Section 2.5 focuses on proposed solutions for disseminating data within a defined area, helping on the reasoning to answer the question "Up to what distance to share?". Finally, Section 2.6 concludes the chapter.

2.2 Data sharing techniques

Data-sharing is the practice of making data available for many devices or people. It is usually performed with the goal of increasing data safety, data availability or data reliability. Data-sharing is indeed a wide concept of data manipulation from where it can be derived techniques as data replication, data dissemination, data collection and data fusion [Saito and Shapiro, 2005]. An overview of these techniques is shown in Figure 2.1.

Data dissemination is the distribution of information to interested users (Figure 2.1-a). Usually, there is no additional concern than make the information known. Hence, broadcast and flooding are usual strategies adopted to reach a large number of users [Chaqfeh et al., 2014]. Furthermore, push or pull-based approaches might be applied. In a push-based approach, data is disseminated pro-actively to any interested user, while in the pull-based model, data is disseminated on-demand, as answers to requests from different users.

Data replication is a key technique used on distributed systems to improve availability and access performance. Availability is achieved by spreading many copies of the data throughout the system and allowing users to access any of them. In this case, data access is possible even when some of the replicas are unavailable (Figure 2.1-c).

Performance improvement can be offered in two ways: selecting the nearest copy to be accessed by interested users or nodes; or allowing concurrent/parallel access to many copies. In the first way, remote access can be avoided along with large network delays. The second way makes possible the access to many copies, at the same time, allowing different parts of data to be processed or copied simultaneously,

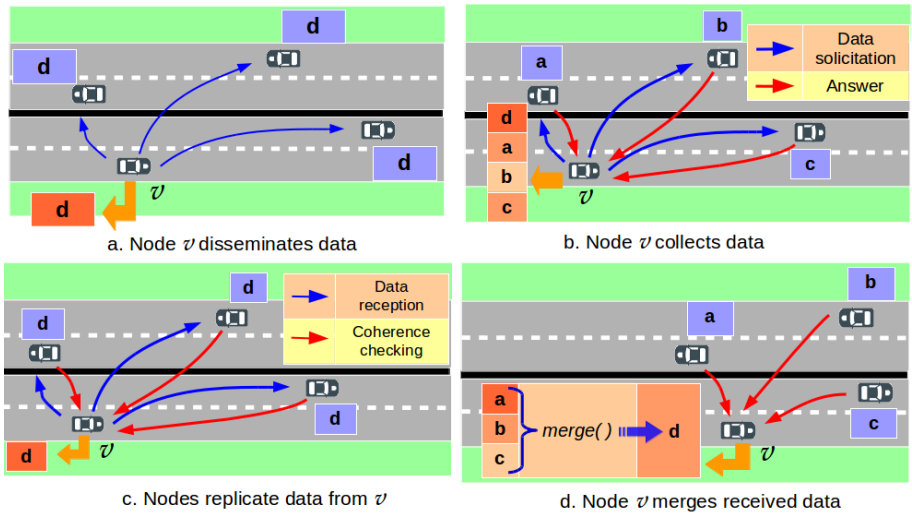


Figure 2.1 – Data sharing techniques

saving time.

Despite the objective and strategy chosen, replication approaches must be thought with a trade-off concerning data consistency. When the system has to provide an illusion of accessing a single copy of data, the access performance and availability are compromised in the name of a hard consistency. Usually, to achieve this goal, access to data is blocked until all replicas are synchronized (having exactly the same value). On the other hand, if the system is able to work with unsynchronized data, allowing small periods of data inconsistency, performance access and availability can be highly improved.

Data Collection is the gathering of information from a variety of sources to construct an accurate knowledge about the interested area or topic. It is usually a demand-fashion data sharing where a user starts collecting data according to its requirements and intended goal, e.g. the user send requests to possible sources of the desired information [Dieudonné et al., 2012]. The process might be conducted randomly, accepting data from any source or with security and credibility concerns when requests are sent only to known and trustworthy sources (Figure 2.1-b).

Once finished the collecting step, the received data has to be processed. In this case, a data fusion approach can be used to combine all received data focusing on errors correction or confidence increasing (Figure 2.1-d). Any action of obtaining data has its own constraints, its own limit of accuracy. Usually, combining different data, from different sources and maybe obtained through different actions, improves data accuracy and confidence. Yet, old saved data might be combined with new readings, making the saved information more recent and reliable [Radak et al., 2016, Čurn et al., 2013].

Although data collect and data fusion are mentioned and considered at some

points, we focus our research on data dissemination. Next sections revise relevant research works chosen due to their impact on the main task of disseminating data in a vehicular network.

2.3 What to share?

Current vehicles are continuously increasing their capacity to produce information. Data gathered through sensors (e.g. radar, lidar, cameras, gps) can be processed to produce a considerable amount of information. Data can also be received from other nodes (vehicles, RSUs) through the network. In this case, data may be related to another geographic area, near or far away of the receiving node.

Despite the source, a node may have a lot of information to spread out. Due to VANETs constraints (see Section 1.3) however, sharing all available data is not possible. Vehicles have to choose what to share, prioritizing data with more quality and relevance. This section describes information issues (Accuracy, Temporal and Geographic dependence, Confidence and Relevance) along with Trust evaluation and Data merging processes, used to assess information quality.

2.3.1 Information issues

A direct and easy way to decide what to share would be to choose only relevant and high quality data. The problem with this approach is that these characteristics are relative and dynamic, assessing them is not easy. Following, several data characteristics are exposed and insights to assess them are presented.

Accuracy

In this text, we consider accuracy as the precision of a measurement. In particular, self-gathered information might have been measured/detected by inaccurate sensors, or the conditions of the detection were not appropriate. As results, we may have obstacles detected with 80% of precision; perceptions of traffic problems with 90% of accuracy; GPS systems yielding positions with 95% of precision and so on.

Accurate information should be favored in dissemination processes. Indeed, disseminating inaccurate information may be useless or even jeopardize an earlier constructed knowledge in the network.

Temporal and geographic dependence

With vehicles moving almost all the time, changing their dynamics and network conditions, the self-gathered information becomes inherently local. Vehicle's sensors have a limited range of action. They can sense data only in their vicinity, capturing data related to a small geographic area around the ego-vehicle. Nevertheless, when receiving data from other nodes, the related area can be largely increased. Information from an area can be iteratively forwarded to distant nodes [Bai and Krishnamachari, 2010].

Dynamic environments bring yet temporal concerns. Every data item presents a time dependence. Neighbors are nearby for a while. Obstacles, accidents, gateways, etc. exist or are available for a finite amount of time.

Before sharing an information, it is imperative to add temporal and geographic references to it. Additionally, the sender has to verify whether it is still valid and whether it is relevant in the current geographic area.

Confidence

The local database of a VANET node can be populated by data from several sources. Despite the source of a data item (embedded sensors, local and distributed algorithms, etc.), it may be described as: "the value is between 10 and 20, for sure"; "the value is probably 15"; "the value is probably between 10 and 20". The first value is imprecise, but is certain. The second one is uncertain, but is precise. The latter is both imprecise and uncertain [Ducourthial et al., 2012]. Imprecision is related to the value, to the measurement accuracy. Uncertainty is related to the confidence, to an indication on the reliability of the information item.

Temporal and geographic dependences also impacts data confidence. Let consider an information about an obstacle with confidence of 90% at the moment of the detection. After a time interval, the confidence should not keep the same value. It should be reduced. The condition might have changed since the first detection.

The problem of confidence becomes worse when considering information received from other nodes. A receiver cannot completely trust a neighbor. Yet, network transmissions increase the time between sensing the data and processing it in the receiver node. Hence, when receiving an information from other node, the confidence should be discounted [Ducourthial and Cherfaoui, 2016].

Relevance

A relevance measure can be defined for the information in reference with the utility (importance) or objective. Considering an information about an accident, if the

receiver is moving in the opposite direction, the information is not relevant to it. The received data is not relevant due to the geographic area. Similarly, data about available parking places is not relevant for nodes not searching for a parking. The received data is not relevant due to objective pursued [Conti et al., 2011].

An evaluation of the relevance is important to avoid disseminating useless data. It can be assessed in reference to time: out-of-date data is rarely relevant; area: data about the vicinity is often more relevant than data from distant areas; objective: different applications require different data (usually, one is not useful for the other).

2.3.2 Trust evaluation

The trust in an entity can be defined as the degree of subjective belief about the behaviors of that particular entity [Cho et al., 2011]. Incorporating trust evaluation in a VANET allows entities to detect dishonest peers and misleading behaviors, improving the evaluation of received data. In particular, a data item received from a trustworthy node should be considered with more confidence.

Many solutions, based on cryptography, search for trust and privacy at the same time. It is the case for [Kim et al., 2014], [Diep and Yeo, 2016] and some solutions from [Zhang et al., 2011]. Nonetheless, within high dynamic scenarios (usually the case of VANETs), cryptography does not perform as well as expected. Excessive delays are introduced [Kerrache et al., 2016].

Kerrache et al. [Kerrache et al., 2016] state yet that trust management should work as a complement of cryptography solutions, contributing to enhance them in delay-sensitive scenarios. Strategies for assessing trust in VANETs are usually divided into three different models: entity-oriented, data-oriented and hybrid. The first one focuses on modeling the *trustworthiness* of entities (nodes) while the second one evaluates the trustworthiness of received data. The combination of these two models enables the hybrid strategy.

From the surveys in [Zhang et al., 2011] and [Kerrache et al., 2016], different strategies for trust management can be highlighted. In the *entity-oriented* models, trust evidence about a node is collected from other nodes. The *experience-based* solutions evaluate the trust based on successive interactions. The *credit values* solutions increase or decrease a credit according to nodes' behaviors. In *data-oriented* models, the *signature-based* technique compare messages with a model of legal messages (a global model is required). Other approaches evaluate the trust through bayesian inference and Dempster-Shafer Theory. In *hybrid* models, every node sends messages along with its own opinion about the trustworthiness of the data.

2.3.3 Data merging

Along with entity's trust evaluation, data merging can be used when assessing information issues: accuracy can be improved by merging different measures of the same data; outdated information can be revalidated, augmenting its confidence, when merging it with new received data.

Two algorithms have been presented by [Ducourthial and Cherfaoui, 2016] for dealing with distributed imprecise and uncertain data in a network. The first one builds the neighborhood confidence of each node based on the inputs of its neighbors. The second one extends this computation to the whole network: each input is taken into account while favoring close information. These algorithms are self-stabilizing, meaning that, they converge in finite time in a legitimate configuration after the topology and the inputs become stable.

Content in opportunistic environments share similarities as well as exhibit temporal/spatial and causal orderings among each other. "Content-fusion" is proposed in answer to opportunistic incident management (e.g. an incident is initially spread in mobile network with many mobile devices sending correlated messages). Moreover, correlating these messages to the spatial environment allows merging or even removing contents that are no longer considered (due to its age or area) [Madhukalya, 2012].

A distributed approach for the construction of a dynamic map is proposed in [Zoghby et al., 2014]. The aim is to exchange maps of dynamic objects among vehicles to increase their environment perception. First, each vehicle uses its local sensors to construct a local map. Then, to avoid disseminating all available data, the local map is combined with received ones by means of belief functions. Only the fusion result is disseminated to other vehicles.

[Li et al., 2013] presents a cooperative multi-vehicle localization method using a split covariance intersection filter. The algorithm maintains an estimate of a decomposed group state in every node. This estimate is shared within the vehicle network. The local estimate is then updated by merging the received estimates with local sensors' data, based on the split covariance intersection filter. Note that the covariance intersection filter is a fusion method which yield consistent estimates even facing unknown degree of inter-estimate correlation. Only direct neighbors are considered.

A cooperative localization strategy is proposed in [Lassoued et al., 2016] with the main goal of diminishing atmospheric and ephemeris errors found when using low cost Global Navigation Satellite Systems (GNSS). In this solution, vehicles cooperate and exchange information such that each vehicle can compute the partner position

with a reliable domain. It is used a set inversion with CSP techniques on intervals. The cooperation is performed between nearby nodes only, and no assess of neighbors identity or confidence.

2.4 In which frequency to share?

As a result of a well performed information evaluation, where every related issue is assessed, vehicles have a set of relevant information to share in the network. The best approach to perform this action however, is not easily defined.

In dynamic networks, the density variation is a crucial characteristic. The number of active nodes has a great impact on network connectivity as well as on the network load [Phe-neau et al., 2014]. In scenarios with high node density, within cities or traffic jams, the network load is the most critical bottleneck for messaging frequency. In low density networks, rural roads or unusual schedules on highways, the connectivity is the major concern. Approaches focused on diminishing the broadcast-storm problem should be used in the former case, whereas store-carry-and-forward strategies are preferable in the latter [Wisitpongphan et al., 2007].

Additionally to the density problem, the information temporal dependency shall be considered for defining the messaging frequency. Dynamic environments demand periodic messages for keeping information up to date. However, sending too many messages in a shared wireless channel causes the broadcast-storm problem. On the other hand, sending too few messages lead to the lack of data or outdated information. An efficient message dissemination process must balance the precision desired with the number of messages sent.

2.4.1 ETSI Cooperative Awareness process

Regarding the development of Intelligent Transportation Systems (ITSs) and traffic safety, the Cooperative Awareness process is a main point [ETSI Ts 102 637-2, 2010]. In this process, ITS stations use a collaborative approach, sharing their dynamics and environment knowledge, to achieve the objective of a common environment awareness. The main rules which guide the process can be summarized as:

- A minimum message generation rate of 1 Hz;
- A maximum message generation rate of 10 Hz.

Additionally, a new message is triggered in any of the following conditions:

- The absolute difference between the current heading of the ITS station and the heading included in the message previously transmitted exceeds 4°;

- The distance between the current position of the ITS station and the position included in the message previously transmitted exceeds 4 meters;
- The absolute difference between the current speed of the ITS station and the speed included in the message previously transmitted exceeds 0,5 m/s.

Even though the common knowledge is constructed through messages exchanges, no information should be forwarded by any node. A clear concern with collaborative processes.

The ITS European specification draws yet definitions about the use of the shared radio channel. This is a big concern taking into account that if many vehicles, in the same area, send messages as defined by the rules, the shared channel may become congested. In a congested channel, a lot of messages collisions take place, leading to reception problems and throughput decreasing [Shen et al., 2013].

2.4.2 Addressing the broadcast-storm problem

[Javed and Khan, 2014] proposed ARC (Adaptive Rate Control), a safety message generation algorithm which combines transmission range and rate generation adaptations. First, every vehicle estimates its safety through the measured headway time. A packet generation rate is defined accordingly. In the second step, the transmission range is adapted based on three metrics estimations: vehicle density; target network load and packet generation rate of each vehicle. The estimated values, transmission range and network load, are shared between nodes in order to achieve a consensus about their value. The main ARC's goal is to improve safety messages spreading while using only the remaining bandwidth of the control channel.

With the main objective of working in a VANET without interfering in other running protocols, [Sommer et al., 2010] proposed the Adaptive Traffic Beacon (ATB) protocol. The key aspect of ATB is to continuously adapt the interval between two beacons to carefully use only the remaining capacity of the wireless channel, without influencing other protocols. In order to achieve its goal, ATB adapts its inter-beacon-interval as a result of the perceived channel quality and the relevance of the message to be sent.

The algorithm LIMERIC was proposed by [Bansal et al., 2013] with two main goals: converging the channel utilization to a desired load level and providing a local fair utilization of this channel. Toward these goals, LIMERIC considers that each node is capable of detecting, or estimating, the aggregate utilization load and that nodes are synchronized, updating their inter-messages delay synchronously.

[Rehman et al., 2014] proposes a distributed alert messaging protocol aiming at to improve relay nodes selection and alert reachability while maintaining

communication delays under recommended thresholds. The solution selects a set of qualified relay nodes based on a proposed bi-directional stable communication (BDSC) protocol. The BDSC protocol takes into consideration the quantitative estimations of link qualities between the broadcaster and the potential relay nodes.

The Best Nodes Approach for Alert Message Dissemination in VANET (BNAMDV) is proposed in [Chelha and Rakrak, 2015]. The main idea consists in classifying nodes in the vicinity according to their moving direction, distance from the sender and response time. Nodes moving toward the alert source, with the larger distance from the sender, and shorter response time are considered better choices.

2.4.3 Addressing the connectivity problem

A Store-Carry-Broadcast (SCB) scheme is proposed by [Sou and Lee, 2012]. SCB nodes keep track of their vicinity to know whether an one-hop neighbor is moving in the same or the opposite direction. Hence, whenever a node cannot communicate with following nodes (vehicles traveling behind in the same direction), it broadcasts its message to vehicles traveling in the opposite direction. The receiver becomes then a SCB forwarder, broadcasting the message in areas not reachable for the first node.

[Li and Jiang, 2011] proposes a routing scheme for intermittently connected mobile networks (ICMNs) based on ferry nodes. Ferries are mobile infrastructural nodes with movements planned for carrying data from source to destination nodes. Geographical and temporal references are used to define paths from RMPs (Receiving Meeting Points) to DMPs (Delivery Meeting Points) in an earlier-constructed graph.

The OSDP (Opportunistic Service Discovery Protocol) uses a store-carry-and-forward approach to spread RSUs' services announcements in mobile networks [Yokoyama et al., 2014]. It works in independent phases of Beaconing (periodic broadcast of services by RSUs), Caching (vehicles save received announcement) and Querying (query broadcast by vehicles searching for a service). A unicast answer R is returned by a vehicle (which has cached the announcement) or a RSU which offers the service.

With similar ideas, caching and store-carry-and-forward strategy, a cooperative service discovery is proposed in [Lakas et al., 2011]. Nodes are classified in Service Providers (SP), which offer services; Service Solicitors (SS), which request and use services; and Proxy Agents (PA), which store and spread services data in the network. Contrarily to OSDP where only RSUs are service providers, [Lakas et al., 2011] have no constraints about this. Any node can play the rule of a SP, SS or PA.

2.4.4 Hybrid solutions for messaging frequency

Several solutions try to manage the broadcast-storm and connectivity problems at the same time by means of hybrid approaches. Adaptive protocols are capable of adjusting their messaging frequency according to the sensed scenario [Limouchi and Mahgoub, 2016].

Pull-based approaches (e.g. Publisher/Subscriber, Information Collect) can also be used to reduce the impact of the mentioned problems. Such techniques usually follow the request-response paradigm requiring less overhead and less latency constraints when compared to push-based approaches [Chaqfeh et al., 2014]. Pull-based techniques often target infotainment data and applications.

Adaptive protocols

The Intelligent Hybrid Adaptive Broadcast (IHAB) protocol takes advantage of two other solutions to offer high level of reachability in sparse networks and to reduce bandwidth consumption in high density scenarios [Limouchi and Mahgoub, 2016]. A potential transmit density metric, represented by the number of neighbors between sender and receiver of the message, is used to determine whether the network is dense or sparse. In the former case, the Bandwidth Efficient Fuzzy Logic-Assisted Broadcast (BEFLAB) is chosen. BEFLAB uses fuzzy logic to obtain a set of candidate forwarding vehicles. Then, based on its distance-to-mean value, a receiver decides whether to forward the message or not. In the latter case, the Fuzzy Logic-Based Broadcast (FLB) is adopted. FLB also uses fuzzy logic to decide if the receiver is qualified to rebroadcast the message but, this time, the reachability is the main parameter to consider.

A speed based adaptive and probabilistic protocol (SAPF) is proposed by [Mylonas et al., 2015]. The protocol is capable of mitigating the broadcast storm and connectivity problems by adapting sending actions to different highway traffic densities. Only the vehicle speed is used to infer the traffic density: low vehicle speed in a highway implies large vehicle densities. In such a situation, low rebroadcast probabilities are enough for ensuring high reachability and low latency of message delivery.

Publish/Subscribe

A hybrid publish/subscribe solution using both ad-hoc and infrastructure networks is presented in [Mishra et al., 2011]. It is assumed a set of info-stations (RSUs) distributed in every major position (e.g. important crossings) of a city scenario. These RSUs are connected to form an overlay network of publications and

subscriptions brokers. Each vehicle can play the role of subscriber, publisher or broker, sending publications/subscriptions to either info-stations or other vehicles.

[Wang et al., 2014] adds a rewarding function to improve the delivery ratio of a publish/subscribe method. The reward is inversely proportional to the total hop count such that nodes are encouraged to choose paths with fewer transmissions. It is assumed that there is a central server which guarantees each node can collect their rewards weekly or monthly. During the warm up, or when it has been idle for a while, a node uses messages to learn potential paths from publishers to subscribers in the network. When two nodes encounter, they exchange messages of interests and estimates potential rewards for forwarding them. According to the reward, they decide to forward the message or not.

A publish/subscribe framework is presented in [Cao and Wang, 2017] with focus on disseminating information about CS (Charging Stations) for EVs (electric vehicles). CSs play the role of publishers whereas EVs play the subscribers. The framework uses yet trusted vehicles (e.g. public buses) as data carriers to enhance opportunistic contacts and increase information dissemination.

Information collect

An anonymous solutions for drivers to query information within a VANET is presented in [Delot et al., 2011]. The algorithm brings as main goal to deliver the query result in a bounded time. A geographic-based routing protocol (GeoVanet) is used to efficiently spread out the request and to take the answer to the requesting node.

A Compressive Sensing based Data Collection (CS-DC) is proposed in [Liu et al., 2013] with the objective of reducing data transmission overhead while ensuring data transmission reliability. CS-DC uses a clustering technique to collect local data and, while considering spatial correlations, to apply data compression methods. A Distance and Mobility based Clustering (DIMOC) algorithm is used to improve clusters stability. Compressive Sensing theory was chosen to efficiently compress and recover in-network data.

The application COL [Dieudonné et al., 2012] offers a collecting data protocol using only vehicle-to-vehicle communication. With COL a collecting process is triggered by a single node (initiator) and finishes when one of the parameters *maxdst*, *maxdur* or *maxstb* is achieved. *maxdst* defines a maximal distance in hops from the initiator; *maxdur* specifies a maximal local duration in time; *maxstb* represents a local maximal duration in case of stable view (when received data does not change the already collected information).

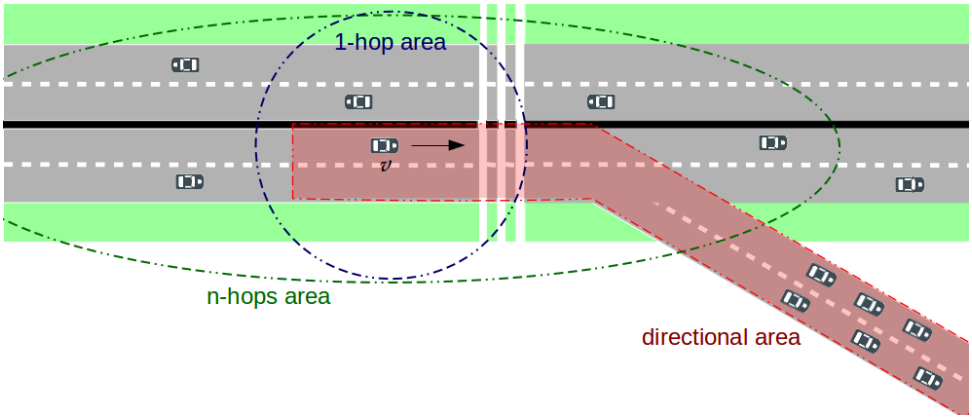


Figure 2.2 – Areas of interest for an ego-vehicle v

2.5 Up to what distance to share?

Dynamic environments require periodic and iterative solutions to keep information up-to-date and reach distant areas. Nevertheless, due to communication constraints and information issues (see earlier sections), it is not possible neither desired to indefinitely broadcast or forward an information. It is preferable to limit the dissemination process to an "area of interest".

An area of interest is an area where the data used, or desired, is relevant, or make sense. Some pertinent questions arise in this context: The information is relevant to whom?; Up to which nodes it should arrive? A node needs an information from which area?. For instance, an information about a traffic jam is relevant to nodes within a large area which encompasses the vicinity of the sender and distant areas with vehicles moving towards the traffic jam. To the contrary, collision avoidance applications often works with small areas, composed only by the vicinity of the ego-node.

Besides the geographic dependency, an area of interest is also affected by the time (see Section 2.3.1). In particular, considering dynamic obstacles in route as animals or small accidents, the area of interest becomes short, due to the ephemeral characteristic of this data. It is not relevant to send an information to distant areas if it will not be valid when the receiving node reaches the original data area.

Additionally, an area of interest can be centered in the ego-node or directional. A centered area is more appropriate for the construction of a general knowledge (e.g. a dynamic neighborhood map). A directional area is useful for specific applications as a traffic jam warning system, for example. In this case, the application requires information from, and disseminates to, specific areas as road junctions, known areas of the city, vehicle's intended path. Figure 2.2 shows different areas for the same vehicle v.

An area of interest can be used then to define how long, and how far, an information should be disseminated. Small areas are usually chosen for information with tight delay constraints or ephemeral data (e.g. lane changing awareness, detection of pedestrians and animals, small accidents, cross-roads systems). In contrast, large areas are often used for long-lasting information which accept larger delays (e.g. traffic jams, icy roads, deviations).

Several solutions are available for defining a dissemination area. Geographic positioning, network partitions and grouping approaches are few examples. In the following sections some of them are described.

2.5.1 Clustering solutions

Grouping approaches can offer different advantages for data dissemination. They can be used to help on defining the process frequency and the reachable area. A data item is often disseminated to all members of the group. Groups, or clusters, might be construct based on geography (vehicles in the same region), data interests (application or data to share), nodes dynamics similarities (same speed, same direction) etc.

GRP constructs and maintains, as long as possible, a group of nodes satisfying a constraint on the diameter. GRP does not rely on a specific node, that may move and leave, but rather searches for stabilizing a view (local knowledge about the group) among all members of the group [Ducourthial et al., 2010].

The TC-MAC algorithm was proposed with the goal of allowing vehicles to exchange non-safety messages without declining safety messages reliability [Almalag et al., 2012]. In order to achieve this objective, TC-MAC constructs groups with collision-free intra-clusters communication where the CH (cluster head) manage the communication through TDMA slots. Moreover, a traffic flow algorithm which takes into account mobility data (location, direction of travel and speed) and the lane where the vehicle resides is used for cluster formation and CH selection.

A mobility driven clustering algorithm for vehicular networks was proposed by [Chiti et al., 2014]. Clusters of 1-hop neighbors are constructed with basis on mobility characteristics and correlation properties among nodes within the same communication range. Relative speed, distance and communication range are used to estimate a connectivity time between nodes. A proper CH is periodically chosen in reference to this estimation. Further, an inter-cluster communication is developed by selecting relay nodes, still based on mobility correlation, in order to provide connectivity between different cluster heads.

The Cluster-Based Location Routing (CBLR) is a reactive and hierarchical

routing algorithm for inter-vehicle communication [Santos et al., 2004]. It employs location information (GPS coordinates) to the routing process and minimizes flooding of Location Request (LREQ) messages. Vehicles in a cluster are classified into cluster head (CH), gateway or member nodes. Member nodes become gateway nodes when they receive messages from more than one cluster head. LREQ messages are sent when a source wants to send data and does not know the location of the destination. A Location Reply (LREP) message returns to the LREQ sender when the location is found. This way, multi-hop communication is achieved in a hierarchical structure.

The Fuzzy-Logic-Based Algorithm (FLBA) organizes vehicles into k-hop clusters according to mobility data (location, direction of travel and speed) and passengers content interests [Tal and Muntean, 2013]. CH eligibility is computed by a Fuzzy Logic Controller (FLC) which takes into account the mobility pattern of the node and a context-aware model represented by a vector of passengers interests. Then, each vehicle periodically broadcasts its ID, vector of interests, location and speed. Clustering process and the election of the CH is performed on these messages.

The VMaSC-LTE hybrid architecture combines multi-hop clustering (802.11p-stations) with the fourth-generation (4G) cellular system [Ucar et al., 2016]. VMaSC-LTE main goal is to achieve a high data packet delivered ratio and low delay while keeping the usage of the cellular architecture at a minimum level. The clustering strategy chosen uses dynamic data, mainly the average relative speed, to construct size- and hop-aware clusters. Still, vehicles give priority to neighboring CHs over neighboring CMs (cluster members) for connection to decrease delay when transmitting to the cluster head. Reactive clustering actions are periodically performed in order to maintain the cluster structure even by re-organizing cluster members or by merging different clusters. Nodes which operates in dual-interface, 802.11p and LTE interfaces, have preference when electing the cluster head.

2.5.2 Geographic and network-partition solutions

[Xue-wen et al., 2010] proposes a Transmission Range Adaptive Broadcast (TRAB) algorithm. TRAB is a greedy algorithm based on the Contention-Based Forwarding (CBF) approach [Füßler et al., 2003]. It focus on improving the broadcast efficiency by selecting the relay node which is farther from the sender and which has the larger transmission range.

The GEographical Data Dissemination for Alert Information and Aware of Network Partition (GEDDAI-NP) is proposed in [Villas et al., 2013]. GEDDAI-NP uses a geographic partition approach where the delay for broadcasting a packet is

defined according to the node's position. If more than one node relies in the same area, that one farther from the source will rebroadcast first, suppressing the other ones' transmissions.

CARRO (Context-Aware Routing pROtocol) is a data dissemination protocol intended to operate on both highway and urban scenarios [Akabane et al., 2016]. Nodes dynamics are used to infer the current scenario. The broadcast storm problem is solved by applying the Zone of Preference mechanism, as in [Villas et al., 2013]. When the algorithm perceives a sparse scenario, the store-and-forward strategy is chosen to keep the data dissemination process operational.

[Ahmadifard et al., 2011] proposes SEFF which aims at increasing efficiency and scalability of multimedia data sharing in Vanets through advantages of BitTorrent protocol and consistent hashing technique. In SEFF, files are distributed without any special device, as gateways. Also, file sharing is possible with all cars in the network, including distant cars. The protocol has 4 phases: Mapping, when every vehicle hash the files it want to share with geographic informations; Establishing the DHTs, when data owners send Notification Messages (NMsg) to the mapped region; Searching and retrieving the file, when the interested node sends a QMsg (query message) to the mapped region requesting an information; and finally Updating the DHTs, performed when a vehicle enters a new region or receives the first new piece of a file and wants to share it.

A geographic-based and location-aware service discovery solution is proposed by [Noguchi et al., 2011]. An architecture composed of IPv6 multicast combined with geographical addressing and routing is used to achieved desired results. IPv6 multicast groups are defined according to the type of the offered service [Geo, 2010]. Next, solutions from the GeoNet project are used to combine IPv6 address with GeoNetworking. Hence, service requests can be sent to correct groups and locations.

A "Reliable Neighbor Discovery" layer was proposed in [Cornejo et al., 2014]. The region-based algorithm divides the physical space in regions and control nodes behavior according to their mobility throughout these regions. In this sense, nodes send notification messages whenever it is about to leave or to enter a region. Above the basic algorithm, a reliability classification service is used to set whether a discovered neighbor will stay long enough in the same region to exchange data. In order to achieve its goals, this solution works with strong assumptions: every node must have a priori knowledge about its future movement directions and the existence of an overall agreement about the geographical space division.

2.5.3 Other solutions

The Decentralized Environmental Notification Message (DENM) is used by ITS applications in order to alert road users of a detected event [ETSI EN 302 637-3, 2014]. An event can be a road hazard, a driving environment concern or a traffic condition situation. Ever a node detects an event, it disseminates a DENM to as many nodes as possible within an area of relevance.

The area of relevance is defined by the detecting node of the event and included in the DENM. Afterwards, receiving nodes perform a relevance checking, defined in terms of distance and traffic direction. Conforming to the area of relevance, a DENM can be forwarded, achieving a multi-hop dissemination of the data.

In order to deal with the highly dynamic network, [Ducourthial et al., 2007] proposes a conditional transmission service, called HOP. HOP selects relay and receiver nodes by means of conditions, instead of addresses. Hence, only the nodes that fulfill the conditions will forward or give the message to their application layers. Conditions could be "being in an area x", "those who are behind the sender", or even "those who have the identity x" (a unicast possibility).

In [Li et al., 2016] the D^{2}NFCE is proposed. The algorithm selects as the next-hop relay node that one with the maximum average forward capability. To do so, D^{2}NFCE uses nodes dynamics to compute the effective connection time among sender and all its neighbors. Then, it uses traffic historical characteristics to construct a predicted traffic model of a neural network in order to fit a throughput function of vehicles. Finally, the throughput function is used within the connection time to compute the node forwarding capability. The node with the bigger forwarding capability is chosen as the relay node.

2.6 Concluding remarks

An uncountable number of services and applications is enabled by an efficient data sharing strategy in VANETs. Traffic safety, driver-aid, infotainment and others applications would benefit from such a strategy. Nevertheless, due to inherent characteristics of VANETs, this is not a simple task. A node cannot share all available data neither use the highest available frequency. It is required to choose and define a correct frequency for achieving remarkable results.

When putting all described concepts together, defining data dissemination characteristics becomes more clear. For instance, if the goal is to spread information about dynamic obstacles with small geographic area and short time references (e.g. animals or pedestrians in the route), the area considered should be small,

and the frequency high. Since the message broadcasting is limited to a small area, the frequency can reach higher values without overloading the network. In contrast, information about traffic jams or big accidents (long-lasting and large area references) should use wide dissemination areas and lower frequencies. This way, distant nodes would be warned, allowing them to choose different routes, avoiding the related area. The communication spectrum would not be overloaded due to the small number of messages sent.

Reactive algorithms (e.g. Data Collect, Publish/Subscribe) have the advantage of generating messages only when necessary. In particular, information collect algorithms start the collection process when the information is needed. A node indicates which information is required, other involved nodes may forward or answer the request. The period of collect is generally larger than for a proactive dissemination. Note however, that reactive algorithms are complementary to data dissemination and cannot replace it. These algorithms are interesting for non usual information - relevant for specific vehicles and with larger time constraints. To the contrary, dissemination algorithms are interesting for general purpose information - relevant for the most part of vehicles and usually presenting hard time constraints.

In the search for an efficient data sharing in VANETs, we continue our studies firstly focusing on the vicinity of a node, considering only direct neighbors (1-hop). Hence, in the next chapter, we propose an adaptive neighbor discovery (AND) algorithm. Even though recognizing neighbors is not a requirement to data sharing, such an information can be used to significantly improve data dissemination efficiency.

Neighbor discovery in VANETs

Contents

3.1 Introduction

This chapter describes a proposal of a neighbor discovery algorithm which aims at an efficient discovery process while saving the wireless communication spectrum. The Adaptive Neighbor Discovery (AND) algorithm merges techniques of cooperative awareness and region-based algorithms (see Section 2.5 for details on such solutions). It makes nodes to share their positions and dynamics to achieve its goals. The structure of this chapter is organized as follows: Section 3.1.1 contextualizes the topic and presents the motivation; Section 3.1.2 describes the objectives pursued; in Section 3.2, the AND positioning in relation to the state of the art is presented; Section 3.3 presents an analytical study of neighbors discovery in VANETs; AND algorithm is presented in Section 3.4; Section 3.5 describes experiments and results achieved to validate the proposal. Finally, Section 3.6 concludes the chapter.

3.1.1 Context and Motivation

As described in Chapter 1, Intelligent Transportation Systems (ITS) can really improve traffic safety and pleasure. Applications like drivers assistance, traffic

monitoring and entertainment (access to videos, photos, etc.) are few examples [Cherif et al., 2010].

Such applications, however, require environment awareness and neighbors recognition by means of sensors and/or periodic beacon messages. Focusing on managing such periodic messages and ITS development, the European Telecommunication Standards Institute (ETSI) published many technical specification documents. Regarding environment perception, the ETSI Cooperative Awareness Basic Service [ETSI Ts 102 637-2, 2010] specifies that every ITS station periodically broadcast data as position, motion state and activated systems to other stations, using so-called CAM messages (CAM - *Cooperative Awareness Messages*). The sending process is triggered according to the dynamical data variation of the node. Yet, the ETSI Decentralized Environmental Notification Basic Service [ETSI EN 302 637-3, 2014] defines an event-triggered message broadcasting for the sake of road users alert: DENM (*Decentralized Environmental Notification Messages*) messages. Nevertheless, considering that ETSI let some services characteristics to the developer and that CAM messages might overload the wireless network capacity, ETSI specifications are not a final word about the topic [Lyamin et al., 2015].

Indeed, message exchanging for cooperative awareness has to be carefully taken. VANETs are very dynamic environments, demanding a lot of messages to accomplish with a good environment knowledge construction. Sending too many messages in a wireless network might overload the communication spectrum, leading to message loss. On the other hand, sending few messages leads to lack of data or outdated information. Thereby, an equilibrium is required: sending messages to speed-up neighbors discovery; control messages frequency to avoid network overload.

3.1.2 Objectives

This chapter proposes a high-level adaptive algorithm with the main goal of assuring to any vehicle the recognition in time of neighboring vehicles even in poor networking conditions. Poor networking conditions mean high loss rates.

The proposal presented here is named AND, standing for Adaptive Neighbor Discovery algorithm. Beyond focusing on the node's dynamics (as in the ETSI standard), AND relies on a cooperative packet losses estimation in order to achieve a high neighborhood knowledge accuracy in any condition. In a straight list, the objectives pursued by AND are:

1. Ensure neighbors discovery while preserving a safety time range for drivers reaction.

2. Control messages frequency to avoid communication spectrum scarcity.

3. Find an equilibrium between efficient neighbor discovery and messages frequency.

3.2 AND positioning in relation to the state of the art

The major standard for beaconing in VANETs has been proposed by ETSI [ETSI Ts 102 637-2, 2010]. In the ETSI proposed process, ITS's stations use a collaborative approach, sharing their dynamics and environment knowledge, to achieve the objective of a common environment awareness. The main rules which guide the process can be summarized as:

- A minimum message generation rate of 1 Hz;
- A maximum message generation rate of 10 Hz.

Additionally, a new message is triggered in any of the following conditions:

- The absolute difference between the current heading of the ITS station and the heading included in the message previously transmitted exceeds 4°;
- The distance between the current position of the ITS station and the position included in the message previously transmitted exceeds 4 meters;
- The absolute difference between the current speed of the ITS station and the speed included in the message previously transmitted exceeds 0,5 m/s.

As we will see, the above strategy may lead to message collisions in the channel and, in turn, to a weak neighborhood knowledge. To address this problem, our strategy takes into account both nodes' dynamics and the networking conditions to compute the inter-messages delay.

Besides the ETSI specifications, many other solutions have been proposed for the cooperative awareness and neighbors identification in VANETs. We summarize the main ones somehow related to our proposal. Additional information about them can be found in Section 2.4.

The Adaptive Rate Control (ARC) is a safety message generation algorithm which combines transmission range and rate generation adaptations [Javed and Khan, 2014]. The main drawbacks of the ARC strategy concern the individual estimation of network metrics. Moreover, considering the remaining bandwidth seems not to be always adapted for urgent messages. To the contrary, our AND algorithm relies on nodes cooperation to estimate the packet losses.

An algorithm which calculates beacons transmission probability based on suspected tracking errors on neighboring vehicles was proposed by [Huang et al., 2009]. The ODRC's (On-Demand Rate Control) transmission rate is increased when the measured tracking errors increases. It is decreased when messages collisions increase on the communication channel. As there is no ACK, the suspected tracking errors and collisions are estimated locally. The idea is to send more messages when a node realizes that its knowledge about neighboring nodes' dynamics has not the desired precision. By comparison, our AND algorithm aims also at obtaining an accurate neighbor knowledge. However, it estimates the accuracy and the packet losses by means of cooperation between neighbors.

With the main objective of working in a VANET without interfering in other running protocols, the Adaptive Traffic Beacon (ATB) algorithm adapts its inter-beacon delay as a result of the perceived channel quality and the relevance of the message to be sent [Sommer et al., 2010]. Differently, LIMERIC considers that each node is capable of estimating the aggregated utilization load. Moreover, it supposes that all nodes are synchronized, updating their inter-message delay synchronously.[Bansal et al., 2013]

Both ATB and LIMERIC classify sending messages with priorities. Additionally, they assume very unlikely hypothesis when working with VANETs: a-priori vehicles density knowledge and nodes synchronization. The main idea is to discard messages with lower priorities in case of radio channel congestion. Both algorithms have as main goal to achieve network utilization fairness, not neighbor discovery or traffic safety.

The region-based algorithm proposed in [Cornejo et al., 2014] divides the physical space in regions and control nodes behavior according to their mobility throughout these regions. In this sense, nodes send notification messages whenever they are about to leave or enter a region. Above the basic ND algorithm, a reliability classification service is used to set whether a discovered neighbor will stay in the same region time enough to exchange data. Only nodes with which connections are considered "stable" are kept as real neighbors.

So far, it can be stated that several neighbor discovery protocols were proposed; solutions focused on spreading nodes' dynamics, on saving communication spectrum and others. Despite the strategy adopted, the complex dynamic behavior of VANETs turns the attempts of mimic these networks' characteristics into a great challenge. Regarding loss rate and network load, many solutions use estimations based on lower layers metrics as Channel Busy Ratio - CBR or Signal-to-Noise Ratio - SNR. Although interesting values can be achieved this way, estimations performed at upper layers are usually more relevant. A neighbor identification accuracy, standing

for how well neighboring nodes are identified, is a metric with more significance to CAM services.

3.3 An Analytical study of the Neighbor Discovery problem in VANETs

The current section presents an analytical study of neighbor discovery by means of beacon messages. Initially, insights from ETSI standard are presented and used to show the relevance of the study. Following, the analytical analysis starts considering a totally reliable vehicular network. Latter, this assumption is relaxed to a more realistic environment. Conclusions will be used to design AND algorithm.

3.3.1 Insights from the ETSI standard

From the ETSI definitions, described in Section 3.2, we can extract that a vehicle traveling at 144 km/h will change its position by 4 meters at each 100 ms, reaching the maximum CAM generation rate of 10 Hz. This is a satisfactory speed considering that most European countries use a maximum speed of 130 km/h[1] [RoadSafety, 2018].

Taking cautious values for the transmission rate of a vehicle (6 Mbps) and for the CAM message size (500 B), we reach a CAM message transmission time of 0.66 ms ([802.11p, 2010], [ETSI Ts 102 637-2, 2010]). With this time, it is possible to send about 150 CAM messages in a 100 ms time interval. By considering a speed close to 130 km/h and a safety delay of 2 seconds between vehicles (leaving enough time for the driver reaction [CodeRoute, 2003]), we obtain approximatively 30 vehicles in the interference range of the IEEE 802.11p protocol (1000 m, see [802.11p, 2010]). Hence, with 5 such lanes we reach the maximal channel occupancy (5 lanes $\times$ 30 cars $\times$ 1 msg = 150 messages). Such a situation appears quite often when considering large highways, bridges or highway interchanges.

Moreover, any lane-changing action requires steering more than 4 degrees, triggering more messages. In the same way, accelerate or decelerate actions varying the speed in more than 0.5 m/s (equivalently 1.8 km/h) are also quite frequent.

Thus, in such situations, the network will become congested and many losses will take place. In fact, the communication problem should arise still more often. Figure 3.1 plots the packet loss probability depending on the number of vehicles in the communication range and the size of the contention window [Lyamin et al., 2015],

[1]Exception made for Germany which works without any limit but yet, with a recommended speed of 130 km/h.

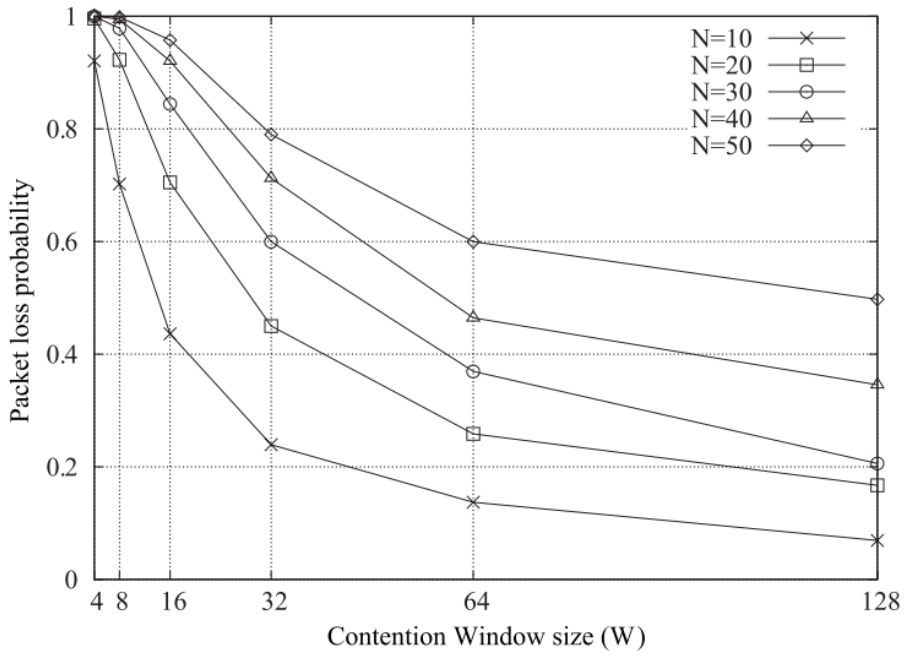


Figure 3.1 – Packet loss probability from [Campolo and Vinel, 2011]

[Campolo and Vinel, 2011]. We can see that, with the 802.11p recommended window size of 16 slots, 20 nodes are enough to overload the network, reaching more than 70% of loss probability.

All such losses will lead to inaccuracies in the neighborhood representation. In this case, despite the triggers, vehicles should decrease their transmission rate in order to avoid the communication spectrum congestion. Hence, a strategy based on neighborhood accuracy and network load appears to be useful.

3.3.2 Scenario and parameters for the analytical study

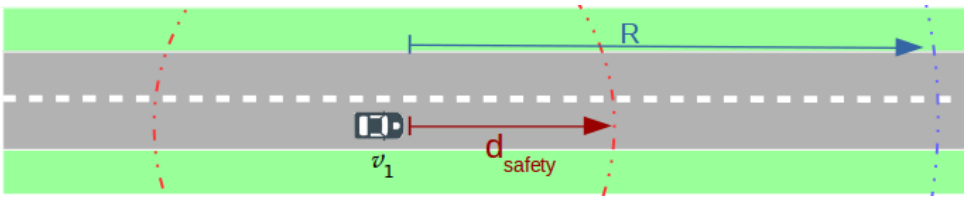


Figure 3.2 – Scenario considered for the analytical study of vehicles identification

Focusing on a wider analytical study of the problem, let consider the scenario depicted in Figure 3.2, where a vehicle travels on a road. The main goal of the study is to ensure that v_1 identifies any other vehicle in its vicinity, R area, with a distance bigger than a safety distance, d_{safety}.

The main parameters adopted are the following:

1. σ **- Time of reaction**. σ holds a time value considered large enough for a driver to react and adapt its driving to changes in the neighborhood.

2. **R - Communication Range**. R holds the radius value of the communication range area of v_1. It means that, in a reliable network, all messages sent within this area are going to be received by v_1.

3. γ **- Minimum inter message delay**. As a message requires some time to be sent, a device cannot send immediately another message. Moreover, a shorter delay may prevent the good reception in case of collision.

4. q **- Reception assurance**. The target reception assurance q represents the probability of receiving a given message which is sent periodically. The parameter q will determine the required number of sending attempts, depending on the loss rate.

Based on the parameters described, the first step is to obtain an equation which gives a maximum inter-messages delay (IMD) value while ensuring the safety reaction time of σ seconds. Accordingly, an upper bound is defined in such a way that keeping IMD values below this upper bound, the main purpose is ensured.

3.3.3 The upper bound

An arriving vehicle v_2 should be detected by v_1 so that v_1 has enough time to react (σ sec). Reciprocally, v_2 should have detected v_1 while it remains σ sec for the driver to react.

Considering s_1 and s_2 as v_1 and v_2 speed values, both measured as vectors, the relative speed Δs between these nodes can be computed by:

$$\Delta s = |s_1 - s_2|$$

So, both v_1 and v_2 have to recognize one another before the distance d_{safety} (in meters) which satisfies:

$$d_{\text{safety}} = \sigma \Delta s$$

Communication in a vehicle wireless network is only possible within a vehicle communication range. Hence, R is always equals to the maximum possible distance of discovery. Additionally, in order to ensure the safety reaction time, the distance of discovery, $d_{\text{discovery}}$, must be larger than the safety discovery distance, d_{safety} (see Figure 3.3). This way, the distance of discovery respects the inequality:

$$d_{\text{safety}} \leq d_{\text{discovery}} \leq R \tag{3.1}$$

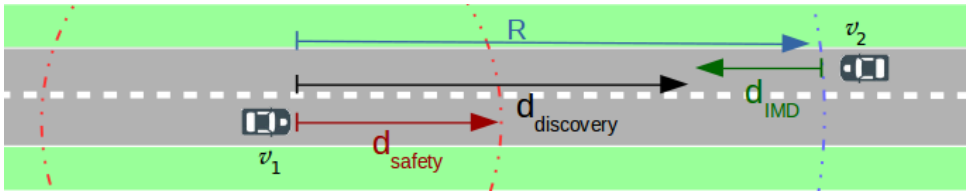


Figure 3.3 – Scenario considered for the analytical study of vehicles identification - Vehicles moving towards one another.

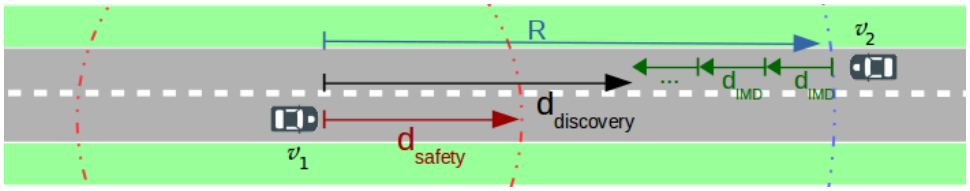


Figure 3.4 – Scenario considered for the analytical study of vehicles identification in an unreliable communication network

Considering that a node discovery can be performed only through a beacon reception, the discovery distance varies in steps of d_{IMD}. d_{IMD} represents the distance traveled by a vehicle between two consecutive beaconing actions. Its value respects:

$$d_{\text{IMD}} = \text{IMD} \times \Delta s \tag{3.2}$$

In fact, in a reliable network, the first node discovery will take place with the first sending beacon within the communication range. For instance, v_1 will discover v_2 when v_2 sends its first beacon within v_1 communication range. It is not possible to know when v_2 enters v_1 communication range, but it is possible to define a range for the distance of discovery with the equation:

$$R - d_{\text{IMD}} \leq d_{\text{discovery}} \leq R \tag{3.3}$$

The worst case here is a node discovery with the smallest distance, i.e. $d_{\text{discovery}} = R - d_{\text{IMD}}$. It is the worst case because it gives the lowest time of reaction to the drivers. By ensuring that the worst discovering case respects the safety time rule, the first goal of this study is accomplished. From this worst case and equations 3.1 and 3.2, a final equation to the inter-messages delay upper bound within reliable networks is given by:

$$d_{\text{safety}} \leq R - d_{\text{IMD}} \implies \text{IMD} \leq \frac{R}{\Delta S} - \sigma \tag{3.4}$$

Unfortunately, reliable wireless networks are not an affordable assumption. When using real networks, the possibility of message loss must be taken into account.

Figure 3.4 shows the same scenario of the Figure 3.3 but now taking unreliable networks into account. Within this new scenario, it might happens that v_2 sends many messages and v_1 doesn't receive any of them. In this case, the distance of discovery is subordinated to the number of attempts required to v_2 to succeed in sending a message to v_1. So, the worst case for the distance of discovery becomes:

$$d_{\text{discovery}} = R - \rho \times d_{\text{IMD}} \tag{3.5}$$

with ρ representing the number of attempts. It can be seen that as more unreliable the network is, more attempts are required, and smaller is the distance of discovery.

From Equations 3.4 and 3.5, a final equation to calculate the IMD upper bound in unreliable networks can be derived:

$$\text{IMD} \leq \frac{1}{\rho}\left(\frac{R}{\Delta S} - \sigma\right) \tag{3.6}$$

Finally, maximum values for the inter-messages delay can be obtained through the Equation 3.6. Nevertheless, one may think that sending beacons as fast as possible (IMD far below the upper bound) is an easy solution. Unfortunately, sending too many messages in a wireless network lead to congestion, message collisions, and subsequently message loss, preventing any success in messages transmission. In order to address this problem, and to achieve a secondary goal: "Save Communication Spectrum", a lower bound definition to IMD values is required.

3.3.4 The lower bound

The rationale to the definition of a lower bound is that message collisions contribute a lot to interferences and the unreliability of wireless networks. The strategy proposed here to avoid such problems is based on neighborhood knowledge and a density coefficient. The neighborhood knowledge represents the number of recognized nodes within the same communication range, while the density coefficient, γ, represents a communication spectrum reservation to each node in that range. The main point here is to keep nodes sending messages while preserving communication spectrum and reducing the number of message collisions. Considering N_{v_1} as the number of nodes within the communication range of node v_1 and γ as the reservation for each node, a lower bound for IMD values can be defined as:

$$\text{IMD} \geq \gamma \times N_{v_1} \tag{3.7}$$

Finally, a general equation to control IMD updating behavior can be obtained

from Equations 3.6 and 3.7. This final equation defines lower and upper bounds to IMD values.

$$\gamma \times N_{v_1} \leq \text{IMD} \leq \frac{1}{\rho}\left(\frac{R}{\Delta s} - \sigma\right) \tag{3.8}$$

3.4 AND - Adaptive Neighbor Discovery

In the earlier section, a theoretical study of neighbor discovery was presented. From the insights and equations defined, an adaptive neighbor discovery algorithm (AND) was proposed and implemented to achieve the desired final goal. Furthermore, AND development was guided by the following predefined rules.

3.4.1 AND Rules

The rules described in this section express strict definitions about the problem of neighbor discovery in vehicular networks and the objectives pursued.

Rule 1 - inter-messages delay upper bound

Every node has to discover all of its neighbors within a defined security time range. Thus, the first (and the most relevant) rule for the inter-messages delay is given by Equation 3.6.

Rule 3 - inter-messages delay lower bound

A node should not overload the network avoiding other nodes to send messages. Thus, the inter-messages delay should fulfill Equation 3.7, aiming at providing fairness and bounding packet loss on the communication channel.

Rule 3 - Inter-messages delay adaptation

Any value between lower and upper bounds is acceptable. A short delay may lead to a more precise neighborhood knowledge. On the other hand, with less messages, less collisions take place and less messages are lost. Hence, the inter-messages delay has to be adapted aiming at diminish the number of messages loss and boosting the neighborhood knowledge construction.

Rule 5 - Inter-messages delay smooth variation

The inter-messages delay could freely float from lower to upper bound but, drastic variations have to be carefully considered. Indeed, a large variation from high to very low values might overload the communication spectrum, leading to messages losses. On the other hand, large updates from low to very high values might lead to late recognition of the neighborhood. Hence, IMD adaptation should be done in a small-step and cumulative strategy.

3.4.2 Parameters Calculation

The direct coding of the rules and equations earlier mentioned is not a complicated task. However, many of the equations make use of parameters really complex to estimate. This section exposes ideas about each one of these parameters. Some of them are not complicated to obtain whereas to other ones, just estimations are possible.

R - Communication Range

The range of communication achieved by current wireless technologies is sensitive to many aspects. Radio-frequency interferences, nodes' speed, physical obstacles and number of transmitting devices are few examples of problems which might affect achievable range. Considering that communication range variability affect mainly the number of lost messages, AND has adopted a fixed value to **R** and an adaptive estimation for the number of lost messages. Although the 802.11p standard defines a maximal R of 1000 m, this value is hardly achieved in practice. Hence, a more cautious value of $R = 500$ m [Breu et al., 2014] was chosen.

Relative Speed

The relative speed between two nodes can be computed based on independent speeds and movement direction. At every beacon reception, the receiving node would be able to compute the relative speed between itself and the sender. However, performing this computation after receiving a first message might be too late for IMD adaptation. So, the relative speed is computed considering vehicle's own speed and a maximal allowed speed[2] for other vehicles, [RoadSafety, 2018].

ρ - Number of Attempts Estimation

Possibly the most complex parameter to obtain, ρ represents an estimation of the network reliability. The idea is to keep counters and sequence numbers for each received message. Based on these counters and numbers it is possible to infer how many messages were lost and then, make an estimation of network reliability. The problem with this approach is that a node might spend some time out of the other node's communication range and, when it comes back, the difference between messages' sequence numbers is going to be very large but, for sure, it is not a number of lost messages. To avoid misinterpretations like this one, all data about neighbors

[2]European countries use a maximum speed of 130 km/h, exception made for Germany which works without any limit but yet, with a recommended speed of 130 km/h.

are kept only while the neighborhood relation exists, e.g. v_1 erases all information it has saved about v_2 when v_2 is not recognized as neighbor anymore. However, results obtained by this approach presented very large variations and, to smooth network reliability estimation, a weighted average was used. Presently, a weight of 70% is being used to last averages and 30% to the actual measurement.

Following, network reliability estimation is a first step to achieve the number of sending messages attempts required to guarantee the reception of, at least, one message. From the probabilities rules, the probability of success of an event **e** in **i** attempts considering its probability as **p** is given by:

$$P(e) = 1 - (1 - p)^i \tag{3.9}$$

Considering **e** as a sending message process and **p**, the network reliability estimated as described above, it is possible to obtain the number of attempts **i** (in this case, ρ) required to ensure a message reception through the Equation 3.10.

$$\rho = \log_{1-p}(1 - q) \tag{3.10}$$

The parameter **q** holds a desired assurance to the process. AND implementation has been worked with a desired assurance of 99%. Thus, ρ represents the number of attempts required to guarantee a message reception with 99% of probability in a network with a reliability of p. A pseudocode used to determine the parameter ρ can be seen in the ComputeReliability() procedure, described in Appendix **??**.

γ - Density Coefficient

The density coefficient represents a communication spectrum reservation to each node in the network. This value has to be chosen in such a way that despite the number of nodes within the same communication range, each node is able to transmit while the number of message collisions remains small. The most part of proposed solutions to the collision problem in VANETs adopt strategies based on lower layers parameters. Several researches were conducted on adjusting nodes' transmission power [Mussa et al., 2014], [Song and Lee, 2013], others were based on carrier-sensing to control a threshold value [Schmidt et al., 2010], [Stanica et al., 2012] and still, researches where the manipulation of the contention-window (CW) were the first idea [Huang et al., 2011], [Reinders et al., 2011].

Even between those proposed solutions based on transmission frequency adaptation, lower level parameters are taken into account. In [Sommer et al., 2011], the adaptive behavior of inter-messages delay is based on radio signal perception and

messages priority (control channel and service channels). Nodes' density is indirectly considered through the number of collisions perceived at MAC level.

AND was thought as a high-level algorithm. Its goal is to provide a solution to neighbor discovery in VANETs within the software layers. Hence, AND estimation to the Density Coefficient was developed considering the 802.11p standard, ETSI definitions and beacon messages characteristics, but not acting in lower layers. The ETSI Cooperative Awareness Basic Service [ETSI Ts 102 637-2, 2010] specifies a minimum delay of 100 ms for its CAM messages. So, this is the value which is going to be adopted as the density coefficient in AND equations.

3.4.3 Cooperative network reliability estimation

In order to estimate the network reliability, each AND instance keeps counters and sequence numbers for every received message. They are used to compute how many messages were received (msg_rcvd) and lost (msg_lost). Such values allow the computation of a local loss rate:

$$\text{local\_rate} = \frac{\text{msg\_rcvd}}{\text{msg\_rcvd} + \text{msg\_lost}} \tag{3.11}$$

This local estimation is sent within beacon messages enabling every node v to compute an estimation for the overall neighborhood. The overall estimation is obtained through a sum of nodes' local estimations (stored in the array local_rate), weighted by the number of neighbors of each node (stored in the array neigh_numb):

$$\text{neigh\_rate} = \frac{\sum_{i=1}^{\text{nodes}} \frac{\text{local\_rate}[i]}{\text{neigh\_numb}[i]}}{\sum_{i=1}^{\text{nodes}} \frac{1}{\text{neigh\_numb}[i]}} \tag{3.12}$$

Nevertheless, results obtained by this approach might present very large variations. In order to smooth the obtained values, a weighted average with the previously computed (old_neigh_rate) and the current loss rate is used. The final loss rate is then given by the following expression:

$$\text{loss\_rate} = \alpha \times \text{old\_neigh\_rate} + (1 - \alpha) \times \text{neigh\_rate} \tag{3.13}$$

Finally, using this loss rate estimation, the number of sending attempts ρ can be computed according to Equation 3.10.

3.4.4 AND algorithm description

From the analytical study presented in Section 3.3 and rules specified in Section 3.4.1, an algorithm for neighbor discovery in VANETs was constructed (Algorithm 1). A detailed description of the implemented algorithm is presented in Appendix A. The algorithm computes the inter-messages delay (denoted IMD). It is an event-driven algorithm, reacting on message arrival and timer expiration.

At each message arrival, the receiver node saves sender's network reliability estimation, neighboring nodes and dynamic data (position, speed, heading) (Line 6). These parameters are used to compute the relative speed and the distance between the sender and the receiver nodes. These two last values are used to obtain the sender's neighborhood lifetime.

The algorithm relies on a periodic behavior guided by a constant, named `aTimer`. This timer represents the lowest time interval allowed (100 milliseconds) in the algorithm. At each `aTimer` expiration, AND deletes the data related to old neighbors (neighbors that did not send messages recently) according to their lifetime (Line 8). This way, the node keeps an up-to-date neighborhood. Next, if any message has arrived since the last timer expiration, the network parameters are updated (line 10). AND uses the high level cooperative approach described in Section 3.4.3 to cope with this task. It permits to estimate the loss rate and in turns the number of attempts ρ. Then the `timeToSend` value is decreased by one `aTimer` and a beacon is sent if `timeToSend` reaches zero (Line 14). Finally, the IMD is updated (Line 16) and, in case of changing, `timeToSend` is adapted accordingly.

It is an easy perception that with more messages, a more precise neighborhood knowledge can be constructed. So, AND is always searching for the lowest IMD value allowed, i.e., the lower bound. AND keeps this behavior until it detects packet losses. In this case, the beacon sending frequency is updated according to an Additive Increase Multiplicative Decrease (AIMD) approach. Each time the network conditions appear to be better, the sending frequency increases using an additive factor. Each time the network conditions appear to be worst, the sending frequency decreases using a multiplicative term. Such a strategy ensures rapid convergence. More precisely, if the loss rate increases, the inter-messages delay (IMD) is increased using a multiplicative factor (Line 20). To the contrary, when the loss rate decreases, the IMD is decreased using an additive term (Line 22).

Finally, besides the AIMD update, the final IMD value is bounded using the upper and lower bounds, obtained with equations 3.6 and 3.7 respectively.

Algorithm 1: **AND algorithm for any node** v

```
1  Initialization:
2     aTimer ← 100ms
3     IMD ← 1000ms
4     timeToSend ← IMD
5  Upon message reception:
6     Save received data related to the sender
7  Upon aTimer expiration:
8     Delete data related to too old neighbors
9     if new messages have been received then
10       Update the network parameters (cooperative estimation)
11    end if
12    timeToSend ← timeToSend - aTimer
13    if timeToSend ≤ 0 then
14       Send a message with node's data
15    end if
16    UpdateIMD()
17    Update timeToSend according to the new IMD value
18 procedure UpdateIMD():
19    if there was message loss then
20       IMD ← IMD × (1 / loss_rate)
21    else
22       IMD ← IMD - aTimer
23    end if

      ▷ Update lower and upper bounds
24    lower_bound ← γ × N_{v_1}
25    upper_bound ← (1/ρ) (R/ΔS − σ)

      ▷ Keep IMD between lower and upper bounds
26    IMD ← min(IMD, upper bound)
27    IMD ← max(IMD, lower bound)
```

3.5 Simulated experiments and results

The present section describes the simulated experiments' goals, the protocols and metrics chosen for comparison. Following, the scenario, tools and parameters adopted to conduct all experiments are detailed. Latter, obtained results are analyzed.

3.5.1 Experiments' goals

AND has been proposed with the main idea of adapting message sending rate based on nodes' dynamics and network usage. This approach provides better results than adapting the sending rate based on nodes' dynamics only. So, experiments

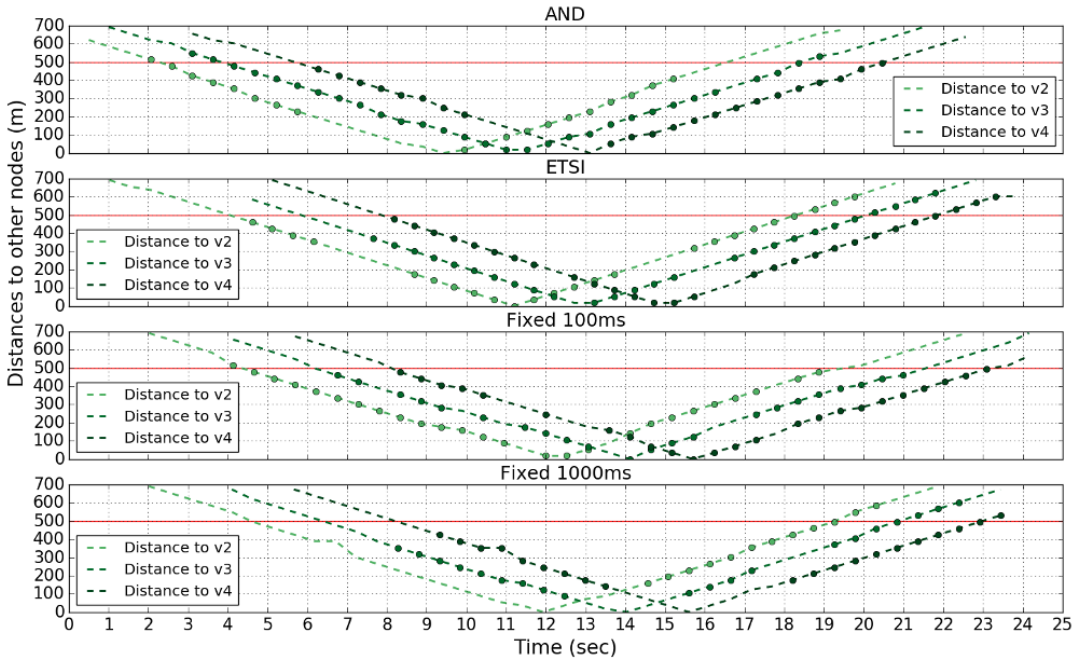


Figure 3.5 – Distances of discovery obtained by each considered algorithm and a packet loss rate of 70%. $v1$ is the ego-node.

aiming at evaluating this proposal and comparing it with other known solutions were conducted. The protocols chosen for comparison were:

1. Fixed: Beacons messages are always sent in a fixed time interval. Two values were chosen, 100 ms and 1000 ms (denoted "Fixed 100" and "Fixed 1000").

2. ETSI: Solution described in Section 3.2 which adapts the delay based on vehicles' dynamics.

3. AND: The proposed neighbor discovery solution.

These three solutions have been compared through the following metrics:

Distance of Discovery The distance of discovery is a crucial value regarding traffic safety. This value must always be larger than the safety distance (Section 3.3). If not, the safety reaction time is not ensured.

Accuracy The Accuracy represents how well an algorithm constructs a perception of the network (*discovered network*) compared to the real network. The *real network* is known by analyzing the GPS trace of each scenario and, based on geographic data and communication range, identifying neighbors of each node. *Discovered network* represents the knowledge constructed by a node through a specific neighbor discovery algorithm, e.g. Fixed, ETSI or AND.

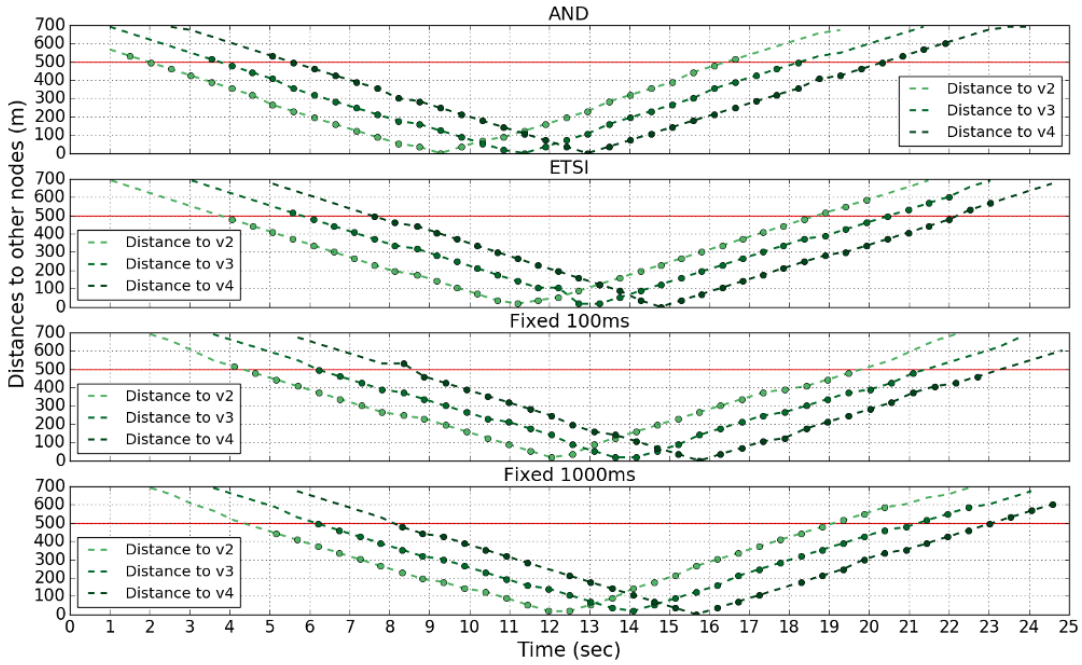


Figure 3.6 – Distances of discovery obtained by each considered algorithm and a variable packet loss rate. $v1$ is the ego-node.

Number of Messages Sent Usually, as more messages are sent, as better is the accuracy achieved. However, sometimes this behavior leads to network congestion, messages losses and poor accuracy marks. Usually, it is preferred sending less messages for the sake of better adaptation to high network densities and to avoid bandwidth wasting.

Putting all metrics together, it can be briefly stated that an ideal solution would be that one which achieves better accuracy with less messages sent and maximum distance of discovery.

3.5.2 Scenario

In order to develop AND and to implement the other two solutions, the Airplug software distribution was adopted. Airplug provides tools for prototyping and experimenting solutions in the context of dynamic networks [Ducourthial, 2013]. The Airplug Emulator, EMU, is capable of emulating networks with vehicles running individually and exchanging messages in a realistic way [Buisset et al., 2010]. Additionally, aiming at improving its network reliability emulation, EMU was extended with the broadcast efficiency metric as described in [Campolo and Vinel, 2011] (see Appendix B).

Regarding the scenario, our choice has been to consider 4 vehicles having to discover each other while encountering different density conditions. We used a

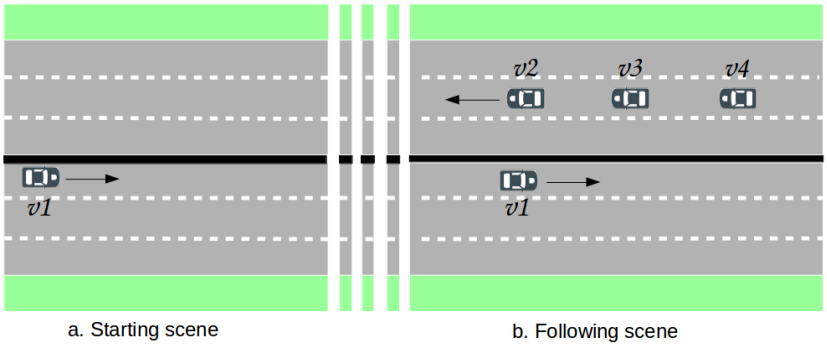


Figure 3.7 – Scenario considered for the emulated experiments

highway scenario constructed by real GPS data, captured in a stretch of a French highway (Figure 3.7). Four nodes traveling in speeds varying from 110 to 130 km/h were used. The scene starts with the node v1 traveling alone in direction of three other nodes, v2 to v4. Later, v1 passes by the platoon, entering and then exiting nodes' communication range. Other scenarios have been studied (eg. cars in a single lane) but the presented one is more challenging.

First, experiments considering only the four described nodes, without congested areas, were conducted. In this case, thanks to the new EMU network reliability emulation, the loss rate varied according to the number of vehicles in the vicinity (See Figure 3.1). Following, the scenario was simulated in such a way that v1 passes by the platoon in the vicinity of a congested interchange. During the meeting of all nodes, an artificial packet loss rate of 70% was created in the network in order to simulate the heavy congested scenario. These scenarios were named "Variable" and "70%" respectively. In both experiments, all actions took place in a time interval of 25 seconds.

Among several configuration parameters offered by Airplug Emulator, the main ones considered during the experiments are shown in Table 3.1.

Table 3.1 – Main parameters for AND experimentation

Parameter	**Value**
Comm. Range	500 meters
Net. Reliability	70%
Number of Nodes	4
Lower Bound	100 ms
Upper Bound	1000 ms

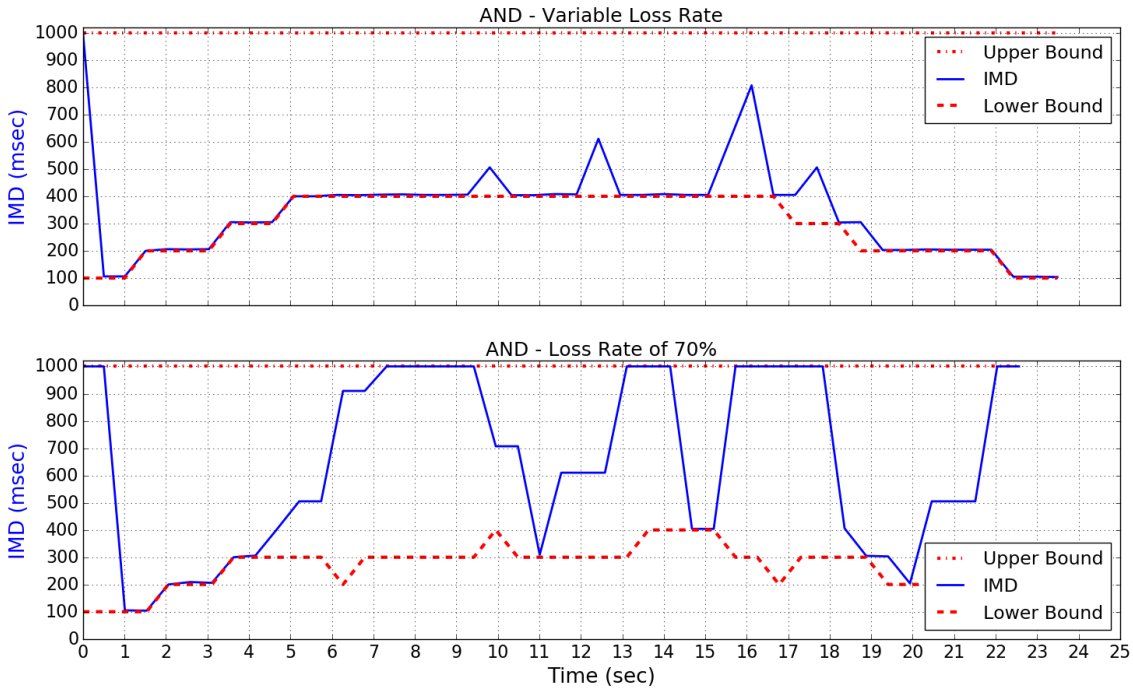


Figure 3.8 – Inter-Message-Delay (IMD) variation during the experiments with AND algorithm

3.5.3 Results

We chose to plot results focusing on vehicle v1 by reason of its larger variation regarding neighboring nodes.

First, the inter-messages delay variation is plotted in order to show how AND adapts to each scenario. Figure 3.8 shows IMD variation for both loss rate values: Variable and 70%. IMD starts at 1000 ms with AND quickly adapting this value to send messages as fast as possible (IMD equals the lower bound). When neighbors start to be recognized, the lower bound is updated (Equation 3.7), pushing IMD up. IMD holds the lower bound value until messages losses start to be detected. At this moment, IMD increases according to the estimated network reliability (lines 19 to 23 in Algorithm 1).

Network unreliability makes IMD and its lower bound vary larger and faster. The lower bound in the second plot of Figure 3.8 hardly reaches the value of 400 ms. This is the right value when v1 recognizes all other 3 vehicles. However, due to the high loss rate, neighbors recognition loose efficiency and v1 hardly recognizes all its neighbors at the same time. On the other hand, with a lower loss rate, the first plot shows a smooth behavior for both IMD and lower bound.

The line charts, illustrated in figures 3.5 and 3.6, were plotted using the time of experiments as X axis, discretized in intervals of one second. In particular, from 9 to 16 seconds, v1 is passing by the platoon of 3 nodes. The constant red line

represents the communication range of 500 meters. Each green line represents the distance between v1 and another vehicle. A dot is drawn over each line whenever v1 has recognized the related vehicle as neighbor.

Figure 3.5 shows distances of discovery achieved in the congested scenario. It can be seen that v1 recognized the other nodes as neighbors with larger distance when using AND or Fixed 100 ms solutions. The Fixed 1000 ms took more time to recognize neighbors, achieving smaller distances. Particularly, v1 recognized v2 only after they have passed by each other. This result clearly shows that an IMD of one second is too long for this scenario. Some holes, lack of dots, can be seen in all plots. Loosing neighborhood identification in some moments was expected due to the high loss rate applied. Yet, AND showed less holes than ETSI and Fixed 1000 ms. Only Fixed 100 ms achieved a better result on this metric but at the cost of too many messages.

The presence of dots above the red line is justified by Airplug emulation of the communication range and by the computed neighbor lifetime. To mimic the reality, Airplug simulates small variations on the communication range (obstacles and interferences). Neighbor lifetime means how long a node is considered as neighbor. AND presented less markers at the end of distance lines due to its adaptive neighbor lifetime estimation. A node is considered as neighbor only during the correct time. For the other solutions, a neighbor lifetime of twice the IMD value was chosen.

Figure 3.6 shows distances of discovery achieved considering a variable loss rate ([Campolo and Vinel, 2011]). A maximum value of 18% of loss was achieved during the time when all nodes are within the same communication range. With few losses, neighbor recognition becomes easier, with all solutions achieving good results.

Finally, figures 3.9 and 3.10 present overall results for the experiments. For the congested scenario, it can be stated that AND achieved a better result with less messages and better accuracy. Too few messages were sent by Fixed 1000 ms but at the cost of very poor accuracy marks. On the other hand, Fixed 100 ms achieved the best accuracy mark at the cost of too many messages. In fact, the best accuracy achieved is less than 2% higher than the accuracy achieved by AND while the number of messages sent was 460% higher. Yet, in all cases AND has achieved better results than the ETSI solution.

AND, thanks to its adaptive behavior aware to message loss, is capable of reducing its sending rate every time losses were detected, and to increment it in losses absence. This behavior leads AND to achieve a small number of lost messages while keeping very good neighbor recognition.

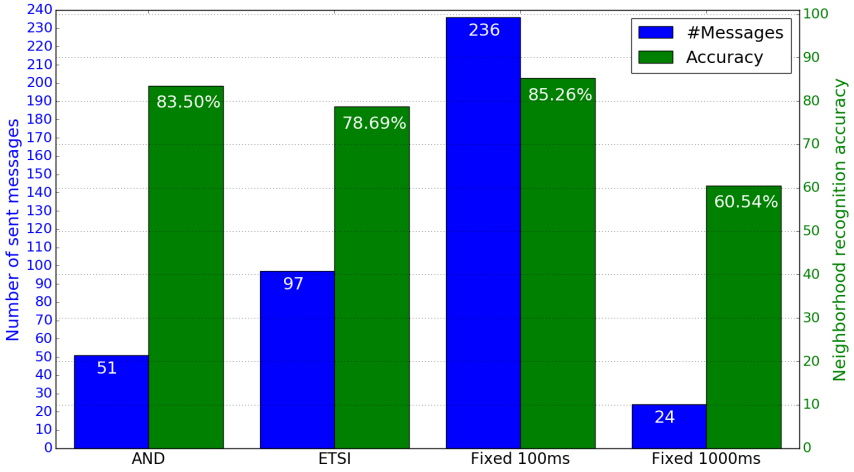


Figure 3.9 – Overall experiment results obtained by each considered algorithm in a network with 70% of loss rate

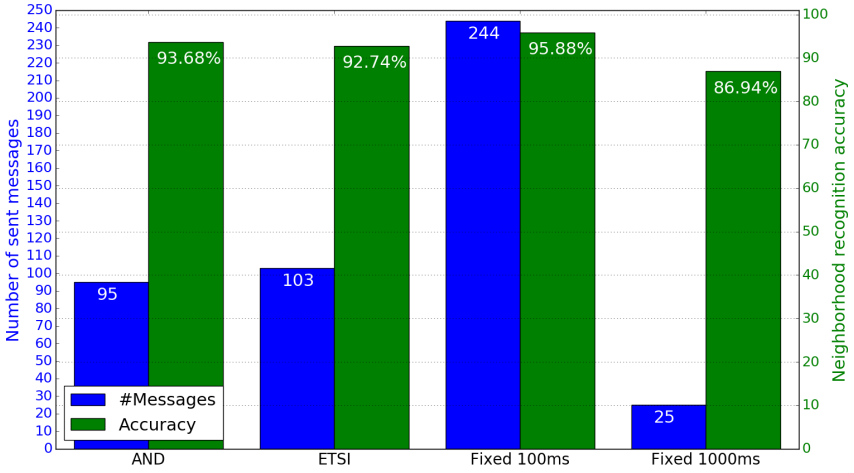


Figure 3.10 – Overall experiment results obtained by each considered algorithm in a network with a variable loss rate

3.6 Conclusions

This chapter provided an extensive study of the neighbor discovery problem in VANETs. It was provided an analytical study of the problem along with a critical view of many known solutions from the literature. From the studies results, a new solution named AND was proposed in order to explore the key points identified. To the best of our knowledge, AND algorithm is the first one to put message density, adaptive behavior and cooperative concerns together. Simulated experiments were conducted for comparative evaluation purpose. The results achieved showed that AND is able to adapt to the dynamical characteristics of the nodes as well as to the wireless network loss rate while preserving very good accuracy in neighbors discovery. This was not the case of other tested solutions and, in particular, the ETSI standard.

Cooperative Neighborhood Map

Contents

4.1 Introduction

In this chapter, an algorithm for the construction of a map of neighbors is proposed. A cooperative approach is used to identify nodes up to n hops of distance while associating to each of them a trust value. The so-called Cooperative Neighborhood Map (CNM) algorithm estimates also the reliability of the path between the ego-node and each of its neighbors. The structure of this chapter is organized as follows: Section 4.1.1 contextualizes the topic and presents the motivation; Section 4.1.2 describes the objectives pursued; in Section 4.2, the CNM positioning in relation to the state of the art is presented; CNM algorithm is detailed in Section 4.4; Section 4.5 describes experiments and results achieved to validate the proposal. Finally, Section 4.6 concludes the chapter.

4.1.1 Context and Motivation

The identification of 1-hop neighbors is a starting point to many other services and applications in mobile environments like VANETs [Moraes and Ducourthial, 2016]. Once a node has its direct neighbors identified, it can cooperate with them. It enables services like lane changing signalization, blind spot awareness, collision

avoidance, etc. It is also possible to diminish communication spectrum congestion by transmitting in an organized fashion.

Nevertheless, limiting nodes identification only to 1-hop neighbors excludes the achievement of wider knowledge and services. By communicating with distant nodes, it becomes possible to enlarge the environment knowledge. Remote obstacles, accidents, deviations, etc. can be identified. Infotainment services (e.g. parking places and hotels availability, chatting nodes, access to the Internet [Kamakura and Ducourthial, 2014]) become wider in the same way with the dissemination of distant service providers.

The recognition of distant nodes can be performed by means of many different strategies, with different goals. For instance, when focusing on data sharing or service offer/discovery, a neighborhood knowledge centered in the ego-vehicle is preferable. Additionally, identifying nodes at many hops of distance requires cooperative approaches. Within such strategies, a wide map of neighbors can be constructed and passed from one node to another, hop-by-hop. Trusting the participants is required, though. When a vehicle adds data received from a neighbor to its own knowledge, it is accepting the data as correct and trustworthy. Therefore, additional security actions are required to guarantee the results correctness [Cho et al., 2011].

Due to the dynamic behavior of VANETs, information related to nodes usually becomes rapidly obsolete. As more an information is forwarded, as more the delay and the distance from the sources increase, augmenting the obsolescence. Also, when it is forwarded by unknown and possibly dishonest vehicles, trust in the information should decrease. Hence, trust should decrease both in time and distance. On the contrary, collaboration may help to reinforce the trust into a received information. If several neighbors agree to an information, trust in it should be increased.

Additionally to the idea of trust, it is required to assess nodes' communication reliability. This metric is specially relevant when using/offering services in the network. Even though a node is considered trustworthy, it cannot appropriately offer services through an unreliable communication: services would become intermittent, presenting many failures.

In this sense, a system where nodes announce their position and available services should contain the following steps:

1. Filtering improbable data. At the reception, each node should discard any improbable information [Ruj et al., 2011]. For instance, the received information could be incoherent regarding previous knowledge (impossible movements; improbable positions, etc.). It may have discovered that a neighbor announces a service that does not work properly, and so on.

2. Reinforcing data accuracy. The received node could increase the accuracy about the information by merging it with other sources [Zoghby et al., 2014, Li et al., 2013].

3. Reinforcing the trust in received data. Besides the accuracy improvement, trust can also be reinforced when the information is corroborated by several neighbors. In this chapter, accuracy and trust are considered as two characteristics of an information. The former is related to the precision of the measure, while the latter is related to the belief in result's veracity. One may assign a poor trust to an accurate disposal, for example, because it suffers damage or because its results are unstable. The reader may refer to [Cho et al., 2011, Kerrache et al., 2016] and Section 4.2 for more details about these terms and concepts.

4. Forwarding data. The receiver node should contribute to data propagation by sharing its information. For this purpose it will send messages to its neighbors, containing its own data (position and services) as well as the received data in order to obtain a n-hop map. As the sent data passed through the previous steps, it includes the computed trust.

This chapter does not deal with filtering nor accuracy in order to focus on trust management within the produced map. The next section describes the main goals of its proposal.

4.1.2 Objectives

The main objective here is to design a Cooperative Neighborhood Map (CNM) algorithm. With CNM, a node shall be capable of constructing a map of neighbors up to n hops of distance while evaluating their trust. Also, CNM performs an estimation of the communication reliability between the ego-node and the identified neighbor.

In a straight list, the objectives pursued here are:

1. To identify neighbors up to n hops of distance. The position and offered services are kept for every neighbor.

2. To evaluate the trust on each identified neighbor (based on time, distance and multiple sources reinforcement).

3. To estimate the quality of the path towards the remote node. This last information is relevant because a remote service is usable only if the communication is correct.

4.2 CNM positioning in relation to the state of the art

Identifying nodes at n hops of distance can be achieved through many solutions and with different goals. In fact, since the advent of vehicular networks, many different solutions have been proposed such as clustering, local dynamic maps, cooperative perception and service discovery. Trust evaluation and management in dynamic networks were also widely studied with several solutions being proposed. Some of these contributions are reviewed in Chapter 2. Here, we discuss their differences in relation to CNM.

CNM algorithm is not focused on the clustering process. It is rather a neighborhood mapping solution. Instead of searching for nodes to keep together in a cluster, CNM identifies all nodes in the neighborhood, up to n hops, and evaluate their trust and communication reliability. Despite some similarities with clustering solutions, CNM is intended to work pro-actively (without requests), based on high level identifications (without relying on specific IP address) and centered on the ego-vehicle.

The cooperative localization solutions presented are closer to CNM when compared to clusters algorithms, but still with different approaches. The solution from [Zoghby et al., 2014] focus on received data classification. Belief functions are used for data fusion. Similarly, [Li et al., 2013] combines local sensor's data with received data to construct a map of neighbors. This time, a split covariance intersection filter is used. Still, both solutions use cautious operators to cope with data incest and have no concerns about nodes mistrust estimation. Differently, CNM address the data incest problem with an approach based on graphs and its Trust evaluation is performed on nodes, not on data.

Regarding the trust evaluation, CNM implements an entity-oriented strategy by means of a distributed and cooperative algorithm. In this strategy, neighbors trust decreases in time and distance from the data source and increases in case of multiple sources confirm an information.

4.3 CNM Approach reasoning

The main objective pursued by this work is to construct a map of neighbors identified up to n hops of distance. Moreover, the resulting map should have an indication of the quality of each neighbor, i.e., a receiver must be able to know if it can trust the data sent or use the services offered by a neighbor. Hence, by means of such a map,

vehicles would be able to identify other vehicles in advance and cooperate with them without additional concerns.

In this section, the approach adopted by CNM for the trust evaluation and path reliability estimations are presented in details.

4.3.1 Distributed trust evaluation

In this text, *trust* is the degree of subjective belief about nodes information; it is represented as a variable between 0 and 1. For the sake of simplicity in result analysis, we admit that each node completely trusts itself (trust = 1), though our algorithm is able to work with other inputs.

Trust decreasing

Discovering distant neighbors requires information forwarding from node to node. As many issues may arise, trust decreases from hop to hop. Indeed, forwarded messages inform about a remote environment that may have changed; it could be a false message; the receiver may not trust sender's sensors, etc. Moreover, relay nodes may be unreliable or not trustworthy vehicles. There is then a relative lack of trust in 1-hop neighbors (trust <1). Since a node identifies its 2-hops neighbors through its 1-hop neighbors, the trust in these nodes is even smaller (trust << 1), and so on.

Besides this initial trust evaluation, data in VANETs are usually related to time, e.g. road conditions, traffic jams, available services. In order to make receiving nodes aware about data validity, a discount by time should be performed in the trust metric. Basically, as older an information is, as more discounted its trust is.

Trust increasing

Receiving the same information from different sources, however, increases the trust in it. With multiple sources, it is possible to check the received information: false data is not confirmed by other sources; inaccuracies in sensors' readings can be reduced; etc.

When several neighbors confirm an information, trust is reinforced. For instance, let consider Figure 4.1, assuming a single path from Node a to Node d. Then Node a recognizes Nodes b_1, c and d with decreasing trust, as explained above. Following, as new neighbors appear (b_2 to b_n), adding new paths from a to c (dotted lines in Figure 4.1), the trust of Node a in Node c should increase because more sources confirm the information from c.

It is important to note that, by only considering multiple sources at one hop, there is no possibility of data incest. Here, it is the trust in c which is increased on a. Trust in d will only increase because it is relayed by a node (c) inheriting a larger trust. Such a scenario is very common in vehicular networks (see Multiple lanes use-case in Section 4.5.1).

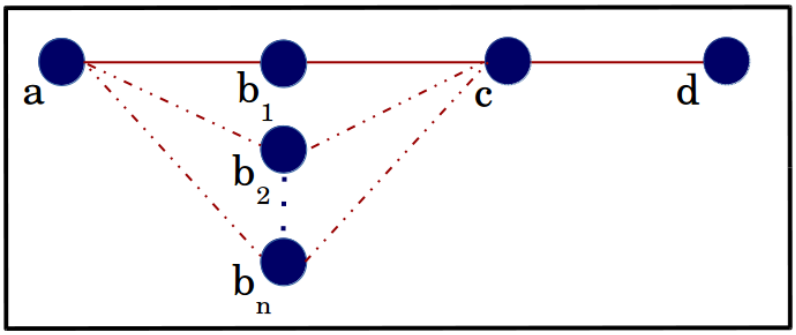


Figure 4.1 – Multiple sources informing about the same node

Hence, our strategy controls both trust decreasing and increasing depending on the topology. Nonetheless, such a topology is unstable in vehicular networks. This may lead to large and rapid variations in the trust metric, given its difficult to exploit by embedding applications willing to take decisions about n-hops neighbors. A smoothing technique is then required at the final stage of trust computation.

4.3.2 Path reliability estimation

In these studies, a path is considered as a "possibility" of communication between two different nodes. It does not represent the way, or track, with all possible intermediate nodes, but only the extremities. In this sense, retaking Figure 4.1 as reference, there is a path $a \rightarrow d$ with several possible ways (passing through nodes b_1 to b_n).

Following, the reliability of a path is defined here as the probability of success of a message exchange between the nodes representing the path. Such a metric can be estimated by keeping up the sequence numbers of received messages and verifying the missing numbers. As all instances of the application works in the same frequency, a reliability discount in time can be applied: an expected message was not received before the timeout.

The reasoning of multiple sources reinforcement applies also to this metric. If a node a recognizes a 2-hops neighbor c through multiple sources, it might happen that some messages were lost by one source but not by the other one. In this case, a may receive all messages sent by c, some messages from one source, other messages from another. As a result, a increases the reliability between itself and c.

As described earlier, VANETs present unstable topology, influencing the trust metric. In fact, instability leads to communication failures, i.e. messages losses. Since the path's reliability is an estimation of messages losses, it should be used in the smoothing strategy for the trust metric.

4.3.3 Rules

From the reasoning explained in the previous section, the following rules were derived:

- **Rule 1 - Oldness decreases trust**: As older an information is, as higher is the probability of it be wrong. Trust should be decreased as the data get older.
- **Rule 2 - Distance decreases trust**: Forwarding messages certainly cause delays and may cause errors. So, trust should be decreased at every hop performed by the information.
- **Rule 3 - Multiple sources increase trust**: When several 1-hop neighbors inform about the same data, the trust in it should be increased.
- **Rule 4 - Trust smooth variation**: Trust should vary smoothly in order to be usable.

4.4 CNM - Cooperative Neighborhood Map Algorithm

From the ideas discussed and rules defined in earlier sections, Algorithm 2 was proposed with the main goal of constructing a map of neighbors up to n-hops. The map is constructed with neighbors' ID, GPS coordinates, available services, trust and path reliability.

The main actions performed by CNM are defined by the `merge` function and by the calculus of trust and path reliability metrics. All of them are explained in details before exposing the complete algorithm.

4.4.1 The `merge` function

CNM algorithm relies on local *views* introduced in [Ducourthial et al., 2010]. A *n-view* of a node is the list of its neighbors up to n hops, ordered by distance. This

is then a list of sets of neighbors. For instance, by taking Figure 4.2 as reference, a *3-view* of w_2 is $\{w_2\}, \{w_1, w_3\}, \{v_1\}, \{v_2, v_4\}$.

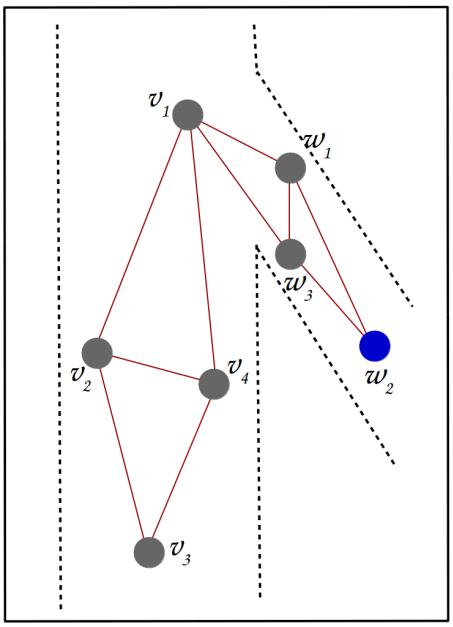


Figure 4.2 – Connection graph for the intermediate connectivity of the highway scenario (Section 4.5.2)

Such views can be computed on the fly as follows:

- Each node keeps its *0-view* composed by itself: $\{w_2\}$ for node w_2.
- on the reception of neighbors' views, they are shifted to the right (because the distance has increased by one more hop).
- Then, they are merged together and with the local *0-view*. For instance, if w_2 receives the *1-views* $(\{w_1\}, \{v_1, w_2, w_3\})$ and $(\{w_3\}, \{v_1, w_1, w_2\})$, from w_1 and w_3 respectively, the merging process would be:

$$\begin{array}{lll} \{w_2\} & & \\ \{\emptyset\}, & \{w_1\}, & \{v_1, w_2, w_3\} \\ \{\emptyset\}, & \{w_3\}, & \{v_1, w_1, w_2\} \\ \hline \{w_2\}, & \{w_1, w_3\}, & \{v_1, w_1, w_2, w_3\} \end{array}$$

- The resulting list is simplified by deleting nodes appearing more than once, keeping the closest one (i.e. the leftmost). The process gives then:

$$\{w_2\}, \{w_1, w_3\}, \{v_1\}$$

- The resulting list is a new, and more complete view (here a *2-view*) for the node w_2. It will be sent to the neighbors.

This merging algorithm is then based on periodic sending of local views. Since only bounded views are considered (limited at n hops), this merging algorithm converges rapidly despite transient failures. Still, it reacts well in case of changes in the network [Ducourthial et al., 2010, Dieudonné et al., 2012].

4.4.2 Trust Computation

Every CNM node v maintains an array $T_v[\ \ ]$ of trusts it puts in nodes it has learnt about. Yet, this array is forwarded to other neighbors along with the views.

In order to implement the rules defined in Section 4.3.3, a time discount denoted by α is applied when the timer expires and no new message has been received for the referenced node. The distance discount denoted by β is applied at every message reception. Finally, the trust metric is reinforced in case of several sources confirming the data.

Let consider Figure 4.1 again. When b_1 receives a message from c, it computes its trust in c by applying the discount β on the received trust (which is 1 because c trusts itself). It then stores $T_{b_1}[c] = \beta \times 1 = \beta$. Similarly, a will store $T_a[b_1] = \beta$ after receiving a message from its 1-hop neighbor b_1. Nonetheless, if the message sent by b_1 also contains b_1's trust in c, a will discount it by β and by its own trust in the sender b_1. Node a then stores

$$T_a[c] = \beta \times (T_a[b_1] \times \beta) = \beta^3 \tag{4.1}$$

Now, suppose that several nodes (b_1, ..., b_n) inform a about c (Fig. 4.1). The trust reinforcement for c is estimated by combining the complement of the trust in each sender ($1 - T_a[b_1]$ to $1 - T_a[b_n]$) similarly to probabilities, giving

$$1 - \prod_{i=1}^{n} (1 - T_a[b_i])$$

In such a situation, many messages for a given 2-hops neighbor are received from several 1-hop neighbors. An strategy is then required to deal with these messages. They may contain copies or different versions of the same information. In order to focus on the trust computation, CNM simply selects the most recent information based on the sequence numbers. Nevertheless, more complex strategies (e.g. data fusion, kalman filter [Rohani et al., 2013]) could be used here.

The selection of the most recent information is performed by the function $F()$.

Hence, the following equation is obtained:

$$T_a[c] = \beta \times \left(1 - \prod_{i=1}^{n}(1 - T_a[b_i])\right) \times F(T_1, T_2, \ldots, T_n) \tag{4.2}$$

where T_i is the trust received from the sender node b_i.

Equation 4.2 combines the distance discount β (first term), the multiple sources reinforcement (second term), and the sender's trust in the 2-hops neighbor (third term). Note that this equation is similar to Eq. 4.1 when there is a single neighbor, $F(T_1) = \beta$. Indeed, in this case, F() returns β, the trust of b regarding c ($T_b[c]$). It is important to note that our multiple source reinforcement technique prevent any data incest [Čurn et al., 2013]. As can be seen in Section 4.5, it gives interesting results.

4.4.3 Trust smoothing

Equation 4.2 gives a punctual result for the node's trust. Nonetheless, this value may present large variations, specially in unreliable scenarios. Hence, for the sake of a smoothed metric, the punctual result is inserted in a variable sliding window of trust measures. The final trust value is then the average of values within the window.

Knowing that messages losses are the main responsible for the variation, trust window's size is defined in relation to the loss rate. As more messages are lost (generating larger variations), as wider is the trust window (more values are used to obtain the average).

Considering p the probability of a unique message loss. The aggregate probability of receiving at least one message in m messages sent is given by $1 - p^m$. In order to ensure at least one message in the window with a probability q, it is obtained: $1 - p^m = q$. Hence, the trust window's size m, in number of messages, can be defined according to the estimated loss rate p and a fixed probability of insurance q, by the Equation 4.3.

$$m = \left\lceil \frac{\ln(1-q)}{\ln p} \right\rceil \tag{4.3}$$

4.4.4 Reliability estimation

Similarly to the trust, each node v maintains an array $R_v[\ ]$ of reliabilities it estimates for nodes it has learnt about. To perform this estimation, v inserts the sequence numbers of messages from u, whatever was the path they used to reach v, into a fix sliding window.

At every timeout, the window is shifted (dropping the last sequence number) and the reliability is computed by dividing the length of the window (number of messages received) by the number of total messages sent (estimated with basis on the maximum sequence number received). This value gives the communication reliability between v and u. It is also used for smoothing the trust computation (parameter p).

4.4.5 CNM Algorithm

CNM (Algorithm 2) is detailed hereafter. For the sake of simplicity, it does not present the collected information such as GPS position and available services.

CNM is a distributed algorithm running on each involved node. It executes its actions periodically, guided by a timer. At every expiration, CNM recomputes the local view by merging it with the received ones, then updates recognized nodes data and sends a message with the local knowledge. It relies on parameters α, β and the sliding window size.

Exchanged messages contain local views represented as lists (denoted by list_v for Node v). Each list is composed by nodes IDs and all related information (offered services, GPS position, trust and path reliability). At reception, the message is stored as a list in an array tabLst indexed by the sender ID - the first element of the list (Lines 47-49). When scanning such received lists, if the data related to a given node is denoted by U then U.id denotes the node's id; U.trust the node's trust; etc.

At every timer expiration, the initial local view is updated (Lines 9-11) and then, a new local view is obtained by merging it with all received ones, lines 12-15. After applying `merge` to the received views, the algorithm computes or updates nodes' data (Lines 16-44). If the saved data about a node is already the most recent, only the time discounting is applied (Lines 33 and 34). However, with new information for a node, all its data has to be updated. The algorithm implements then the trust and reliability computation as explained in Section 4.4.4 and Section 4.4.2 respectively.

Algorithm 2: **CNM algorithm for any node** v

```
1  Starting_action:
     ▷ Data of known nodes, indexed by nodes id
2    tabSeq[ ] ← ∅                          ▷ Sequence numbers of messages
3    tabTrust[ ] ← ∅                        ▷ Trust for each neighbor
4    tabRel[ ] ← ∅                          ▷ Reliability for each neighbor
     ▷ Dynamic slidding windows for smoothing
5    winRel[ ] ← ∅                          ▷ Reliability window
6    winTrust[ ] ← ∅                        ▷ Trust window
     ▷ tabList contains received lists of nodes along with their data
7    tabList[ ] ← ∅
```

8 Upon timer expiration:
9 V.id $\leftarrow v$
10 V.nseq $\leftarrow$ nseq++
11 V.trust $\leftarrow 1$
▷ *Computing the local view*
12 list$_v$ $\leftarrow V$
13 **for each** list in tabList **do**
▷ *Merging received views with the local one*
14 list$_v$ $\leftarrow$ merge(list$_v$, shiftRight(list))
15 **end for**
▷ *Save/Update nodes' data*
16 **for each** U in list$_v$ **do**
17 Create entries in local arrays if they do not exist
18 **if** tabSeq[U.id] $<$ U.nseq **then**
19 Update U's nseq ▷ *New message informing U*
▷ *Insert the received nseq in Reliability window*
20 winRel[U.id] $\leftarrow$ winRel[U.id] $\cup$ tabSeq[U.id]
▷ *Compute nodes' punctual trust*
21 senders $\leftarrow \emptyset$ ▷ *List of different senders of U*
22 nSend $\leftarrow 0$ ▷ *Number of different senders*
▷ *If U is a 2-hop neighbor, search for different 1-hop senders*
23 **if** $U \in$ lindex(list$_v$, 2) **then**
24 **for each** $W \in$ lindex(list$_v$, 1) **do**
25 **if** $U \in$ tabList[W.id] **then**
▷ *If U was sent by W, take it into account*
26 senders.append(W)
27 nSend++
28 **end if**
29 **end for**
30 **end if**
▷ *Computing node's trust*
31 tmpTrust $\leftarrow T_a[c] = \beta \times \left(1 - \prod_{i=1}^{n}(1 - T_a[b_i])\right) \times F(T_1, T_2, \ldots, T_n)$
32 **else** ▷ *No new message informing U*
33 Drop the oldest nseq in winRel[U.id]
▷ *Discount the last value in trust window*
34 tmpTrust $\leftarrow$ winTrust[U.id].last $\times \alpha$
35 **end if**
▷ *Compute node's Reliability using the fix sliding window*
36 Drop x in winRel[U.id] if $x \leq$ max(winRel[u]) - size(winRel)
37 tabRel[U.id] $\leftarrow \dfrac{\text{length}(\text{winRel}_{[U.\text{id}]})}{\min(\max(\text{winRel}[U.\text{id}]), \text{size}(\text{winRel}))}$
▷ *Smooth nodes' trust using the variable sliding window*
38 winTrust[U.id] $\leftarrow$ winTrust[U.id] $\cup$ tmpTrust
39 Update winTrust[U.id] size with $p =$ tabRel[U.id] and $q = 90\%$
40 tabTrust[U.id] $\leftarrow \dfrac{\sum_{i=1}^{\text{length}(\text{winTrust}_{[U.\text{id}]})} \text{winTrust}_{[U.\text{id}].i}}{\text{length}(\text{winTrust}_{[U.\text{id}]})}$
41 **if** tabTrust[U.id] is too small **then**
42 Delete U and all related data in local arrays
43 **end if**
44 **end for**
▷ *Send message with the new view and nodes' data*
45 Send(list$_v$)

```
46 Upon message reception:
47    receive( list )
      ▷ Sender's data is in the first position of the list
48    S ← lindex(list, 0)                          ▷ Get sender's data
49    tabList[S.id] ← list
```

Note that, when searching for multiple sources for a node, it is enough to consider only nodes at 2-hops of distance, as explained before. Multiple sources are searched among 1-hop neighbors (Lines 21 to 30). Final values for trust and reliability are then computed in lines 36 to 40. In case the trust is too small, the node (and all related data) is discarded.

By using CNM characteristics presented so far, it is possible to accomplish with the task of constructing a dynamic map of neighbors while classifying them according to their trust (see Figures 4.7 and 4.10). Notwithstanding, such a map might be used with many different goals in mind: traffic monitoring and safety, services or applications availability, etc.

4.5 CNM validation

The validation of the CNM algorithm was performed by means of a set of experiments in different scenarios. First, a static scenario illustrating the interest of the adopted approach and trust computation on a multiple lane road with a stable traffic was performed. Following, a dynamic scenario where two flows of vehicles merge in a highway junction was used to conduct experiments without packet losses and then with losses. The main goal was to show the ability of CNM to construct a map of neighbors while evaluating the trust of each identified node.

4.5.1 Multiple lane use-case (static scenario)

It is considered a scenario where the traffic is stable on multiple lane roads to show the interest of trust discounting and reinforcement. Figure 4.3 displays the communication graph for a single-, two- and three-lane roads.

Figures 4.4 and 4.5 plot the trust obtained by Node a for the vehicles in front of it at one, two, three... hops. Computations have been done with $\beta = 0.6$ and $\beta = 0.8$. The trust metric decreases as more hops are performed by the message.

The impact of trust reinforcement can be observed by comparing results for different types of roads. Trust decreases less when considering an additional lane, though it is no more significant after 4 lanes (roads with more than 4 lanes are rare). For instance, with two lanes the trust is multiplied by 1.2 at 2 hops and by

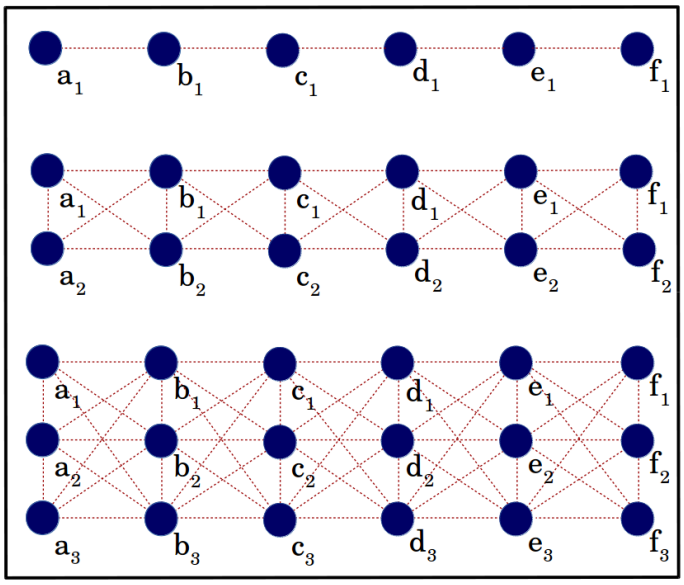


Figure 4.3 – Connection graphs for one, two and three-lane roads

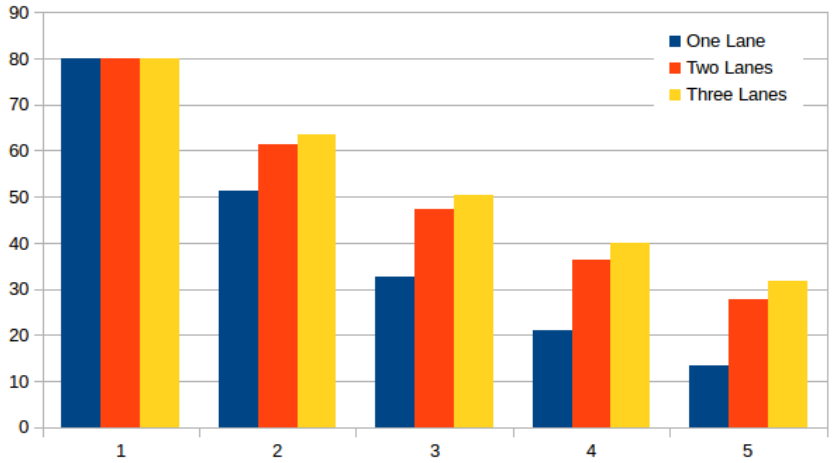


Figure 4.4 – Trust computed on Node a for nodes at 1 to 5 hops. Using $\beta = 0.6$.

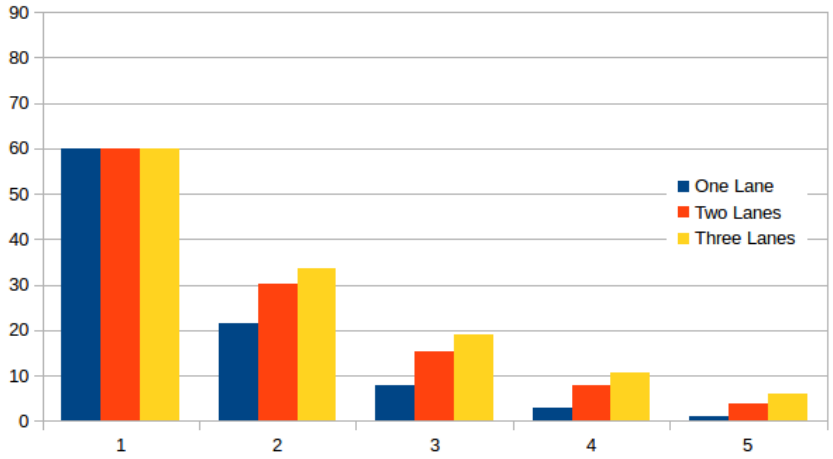


Figure 4.5 – Trust computed on Node a for nodes at 1 to 5 hops. Using $\beta = 0.8$

2 at 5 hops (when $\beta = 0.8$). Results are even more evident when $\beta = 0.6$: trust is multiplied by 2 at 3 hops.

4.5.2 Highway with a junction use-case (dynamic scenario)

Interchange and junction areas present many hazards to drivers with cars entering high traffic highways without good vision and perception of the area.

In the scenario illustrated in Figure 4.6, two flows of vehicles with a different knowledge of the network meet and merge. Observing the trust evolution is then interesting not only for available services discovery but also for road safety. Indeed, a n-hop map of neighbors would permit to assist vehicles in both highway and junction to be aware of one another.

As seen in the previous scenario, the density of the network impacts the results. Hence, this time three phases with different network densities are considered:

1. **Minimal connectivity**: Three vehicles ($v1$ to $v3$) are traveling in the main road while other two ($w1$ and $w2$) are getting into the road through a junction. The minimal connectivity is characterized by the existence of only one path to spread out the information.

2. **Intermediate connectivity**: Two vehicles ($v4$ and $w3$) are added to the scene. One is added to the main road and the other one to the junction. With an intermediate connectivity, several paths exist to spread out the information.

3. **High connectivity**: Other two vehicles ($v5$ and $w4$) are added to the scene. Again, one is added to the main road and the other one to the junction. In this scenario, a node has many 1-hop neighbors. Information can be spread out through many different paths.

4.5.3 Experiments characteristics and tools

Before starting the experiments described here, several tunning analysis were performed on CNM parameters. Table 4.1 shows the values chosen to guide the behavior of CNM algorithm.

The Airplug framework [Ducourthial, 2013] was used to implement the CNM algorithm and to carry on the experiments on dynamic scenarios. Among several configuration parameters offered by the emulator, the main ones considered here are described in Table 4.2.

Another Airplug application, called MAP, was used to help in results displaying. MAP is able to plot a point on an OpenStreet map according to its GPS coordinates

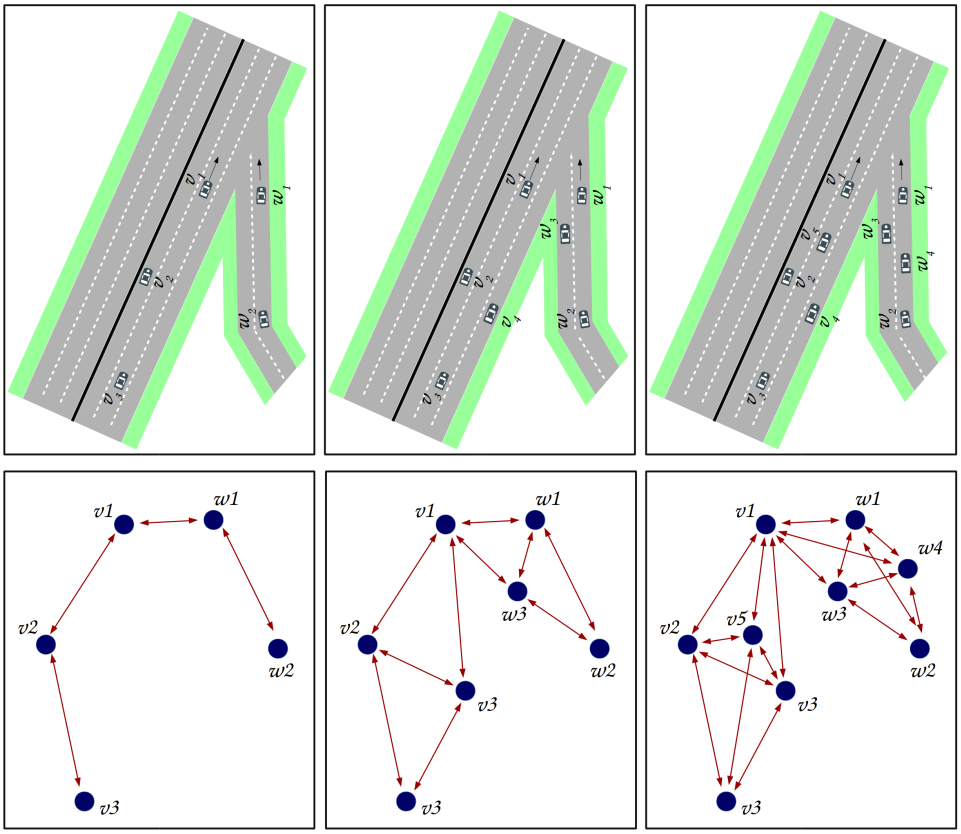


Figure 4.6 – A highway junction and related communication graphs

Table 4.1 – CNM parameters used in experiments

Parameter	Value
timer	1 sec
β - Distance discounting	20%
α - Time discounting	10%

and to change the point appearance with respect to many parameters. MAP was used to show identified nodes along with their related trust. For instance, Figures 4.7 and 4.10 show typical outputs of MAP, captured in a specific moment of the experiments execution.

The strategy used for maps construction was to show nodes as circles which change their color according to the trust value (see Trust Palette in figures). Along with the maps, a table was used to show the achieved results.

In all maps and table, the "ego-vehicle" (owner of results) was the vehicle $w2$. It was chosen due to the fact of being in the edge of the communicating group. Hence, it is relevant to know how well a distant node can map the highway before reaching it. $w2$ is shown in blue in the center of every map.

Table 4.2 – Network emulation parameters for CNM experimentation

Parameter	Value
Experiment time	60 sec
Communication Range	300 meters
Network Reliability	100% / 60%
Number of Nodes	9

4.5.4 Results for the dynamic scenario without packet losses

Figure 4.7 gives results for low and large densities plotted with the MAP application. Note that these figures do not represent an average of the obtained results but a snapshot during the dynamic scenario execution.

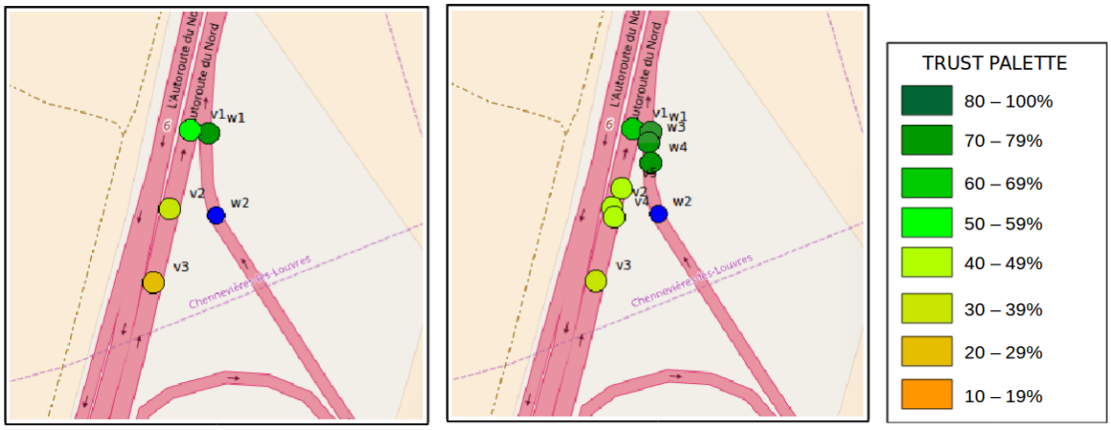


Figure 4.7 – Trusts of Node w_2 in other nodes during the experiment - no loss.

In the first map of Figure 4.7, constructed with a minimal connectivity, there exists a single path $w_2 \leftrightarrow w_1 \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow v_3$. Node $w2$ recognizes then $w1$, $v1$, $v2$ and $v3$ with decreasing trust. This is the expected behavior considering the increasing distance from $w2$.

When the density is larger (second map), the connectivity is more important. The connection graph (bottom sub-figures on Fig. 4.6) admits more edges both in the highway lane (vehicles v_i) and in the highway junction (vehicles w_i). Hence, the trust reinforcement applies, reducing the influence of the number of hops. As a consequence, vehicle w_2 obtains a larger trust for nodes in front of him as well as for nodes in the highway.

Average trust values for each scenario are shown in Table 4.3. It can be seen that $w2$ sets a trust of 80% to $w1$, $w3$ and $w4$, the maximum value achievable. This value is reached only by 1-hop neighbors. $v1$ is at 2 hops of distance, which should provide

Table 4.3 – Average trust of Node w_2 on the other nodes of the experiment - Scenarios with and without losses

Vehicles	No Loss			40% Loss		
	Connectivities			Connectivities		
	Minimal	Intermediate	High	Minimal	Intermediate	High
v1	51%	61%	63%	41%	58%	63%
v2	33%	39%	41%	23%	31%	40%
v3	21%	30%	32%	15%	25%	29%
v4	-	39%	41%	-	29%	38%
v5	-	-	41%	-	-	40%
w1	80%	80%	80%	71%	77%	78%
w3	-	80%	80%	-	78%	80%
w4	-	-	80%	-	-	80%

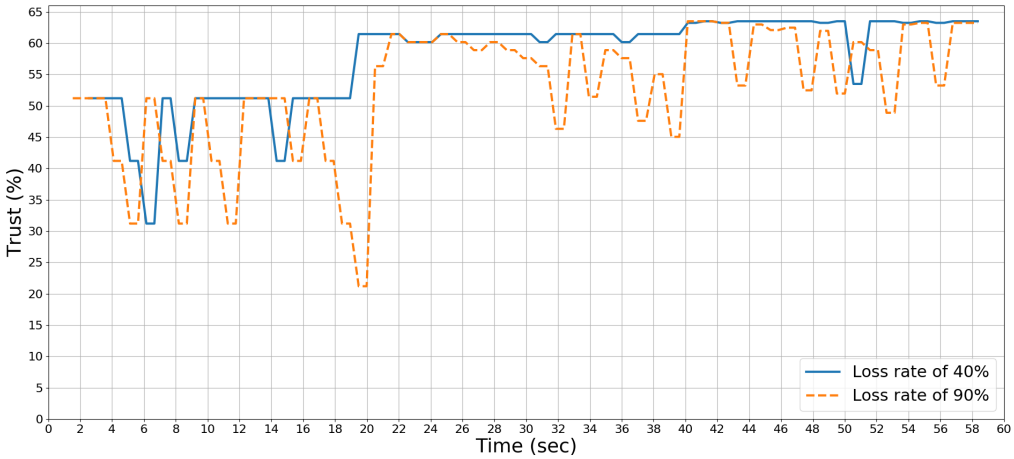


Figure 4.8 – Trust variation of Node w_2 in other nodes when not applying the smoothing mechanism

a trust of 51% (value achieved in the first scenario). Following, as more nodes inform $w2$ about $v1$, the trust given increases. Hence, the final value for $v1$'s trust is 63%. Applying the same process to node $v3$, the farther one, the values achieved were 21%, 30% and 32%, for the first, second and third phases respectively. For all nodes, the trust value achieved is in accordance with the rules.

4.5.5 Results for the dynamic scenario with packet losses

The previous results have been obtained in a reliable network without packet losses. Obviously, this is not the case in a dynamic vehicular network. In this section, the impact of 40% of packet loss in the highway junction scenario is studied.

Packet losses increase the variations into the trust metric. Before showing the results, it is worth to show the relevance of the trust window used to smooth the metric (Section 4.4.3). Figures 4.8 and 4.9 show the trust values given by Node w_2

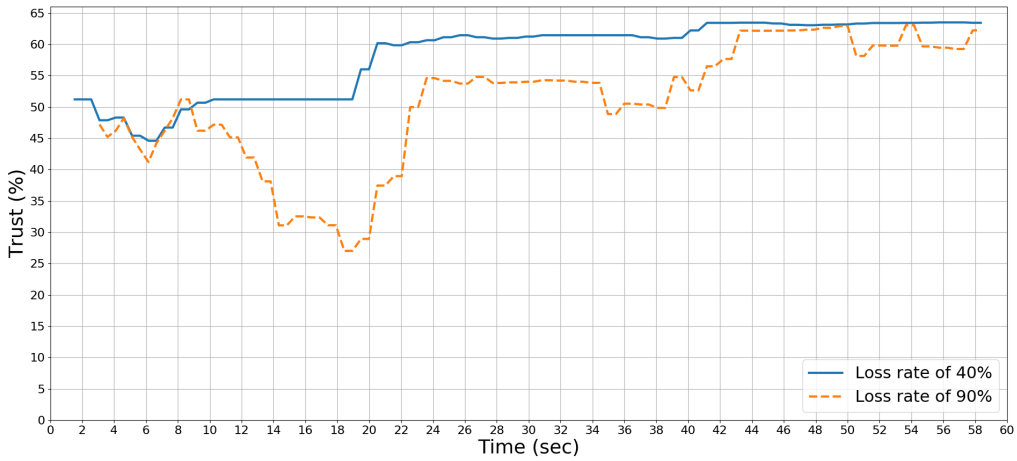


Figure 4.9 – Trust variation of Node w_2 in other nodes when applying the smoothing mechanism

to Node v_1. It can be seen that as the loss rate increases (40 and 90%), the trust curve smoothing is more evident. While for the first chart (Fig. 4.8) large and fast changes are observed, specially for high loss rates, the smoothed curve, in the second chart (Fig. 4.9) shows a really more fluid behavior. Still, bordering values are hardly reached by the smoothed curve.

Regarding experiment achieved results, Figure 4.10 illustrates maps obtained when running CNM in a network with a loss rate of 40%. Due to the impact of message loss at the trust computation, several different colors can be seen in the maps. Nevertheless, it can be seen that the impact of losses on the trust computation is not high.

Detailed statistics are shown in "40% Loss" column of Table 4.3. In general, lower trust values were obtained for this scenario when compared to the scenario without loss. Note however that, as the connection density increases (from minimal to high) the trust values achieved are closer to the ones obtained in the first scenario. This behavior is explained by the fact that a sent message may be lost by one receiver but not by all of them. Hence, as the number of potential receivers increases (high connectivity), the loss rate attenuation increases in the same way.

4.6 Conclusions

Neighbor identification and classification is a complex task when considering the aspects of dynamic networks. Such an operation is not only useful for road safety but also for service discovering, a necessary step for ITS cooperative applications.

There are many proposed solutions in literature, including clustering, service discovery and data fusion strategies. Nevertheless, the results achieved by such

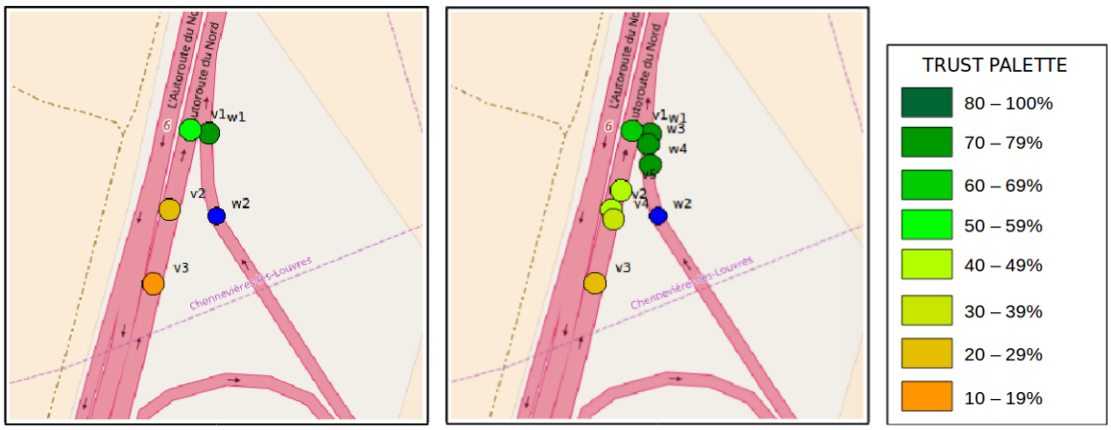


Figure 4.10 – Trusts of Node w_2 in other nodes during the execution - 40% of Loss

solutions are often influenced by the application intended and not centralized on a specific node, e.g. “ego-vehicle”.

CNM was proposed with the main goal of constructing a general dynamic map of neighbors up to n hops of distance. The map includes node position, available services and an evaluation of neighbors’ trust. A distributed and cooperative strategy where neighbors’ trust decreases in time and distance from the data source and increases in case of multiple sources reinforcement was developed. Also, the strategy avoids data incest with an approach based on graphs. CNM offers yet an evaluation of the multi-hop path quality towards each identified node. Such an evaluation is specially relevant for qualifying the usability of a remote service. It is also used to smooth the trust using a dynamic sliding window.

This approach has been validated by means of extensive study using a dynamic network emulator, showing very interesting properties. We believe that CNM is a promising solution to build a cooperative neighborhood map, displaying nodes and available services in the vicinity of the vehicles up to n hops. It is expected that the resulting map can be used by latter applications in order to choose nodes to communicate with.

An integrated architecture for data sharing in VANETs

Contents

5.1 Introduction

In this chapter, we address the problem of data sharing in dynamic networks with a cooperative strategy capable of selecting data and disseminating it within a dynamically defined horizon. In our Relevant Information Frame (RIF) architecture, we select the most relevant data items (in the scope of a node), put them into a frame and disseminate it in the vicinity. As the process is repeated at every node, some data items will continue to be chosen for further dissemination while others will be kept local. The structure of this chapter is organized as follows: Section 5.1.1 contextualizes the topic and present our motivations; Section 5.1.2 describes the main goals pursued; The reasoning leading to our RIF architecture is developed in Section 5.2; Section 5.3 exposes and explains the proposed architecture. A validation of our proposal is performed by means of a comparative study in Section 5.4. Finally, Section 5.5 concludes the chapter.

5.1.1 Context and Motivation

In the context of Intelligent Transportation Systems - ITS, a vehicle may have a lot of sensors and applications generating data. For instance, a VANET node may

have cameras, lidars, radars, among other sensors. It may also use applications for obstacle/pedestrian detection, collision avoidance, parking assistance, etc.

Vehicles represent important sources of information worth to share. Local applications can be largely extended by sharing the gathered data in the network. The detection of distant obstacles, traffic jams, and the offer of distributed services like route management, chatting, etc. become possible. Furthermore, data accuracy, confidence and relevance can be verified when receiving data from multiple nodes.

Several data sharing solutions are available nowadays. Classic solutions like the Network File System (NFS), mutual exclusion and consensus protocols [Saito and Shapiro, 2005, Lamport, 1998] achieve good results on infrastructure networks but, their application to dynamic networks is doubtful. The possibility of leaving and arriving participants presents hard challenges for such solutions. Consensus protocols are more flexible then the other ones. They can manage leaving and arriving nodes, but they suppose that a leaving node will be back in the future. Such a constraint cannot be guaranteed within VANETs.

Regarding approaches developed for MANETs and VANETs, the most part of them is focused on solving the broadcast-storm or sparse networks problems. Indeed, vehicles may face periods with a lot of neighbors (traffic jams or busy roads - when the broadcast-storm problem usually arises), and periods without any neighbor (sparse networks - when keeping the communication is the main question). Common strategies to cope with these problems are based on geographic positions, network partitions, context-aware and relative position to the sender (the reader may refer to Chapter 2 for more details).

Despite individual results achieved, current solutions work individually, trying to share all data they have gathered. Nonetheless, within a node with several applications, such behavior will certainly conduct to network problems.

5.1.2 Objectives

The main objective pursued here is to provide an efficient data sharing within the VANET environment. Our approach is based on two main points: I) a cooperative data sharing strategy; II) a dynamically defined horizon of data dissemination. In particular, data sharing is a wide concept of data manipulation from where it can be derived techniques as data replication, data dissemination, data collection and data fusion. In this chapter, we investigate data sharing in VANETs by means of data fusion and data dissemination techniques.

In our Relevant Information Frame (RIF) architecture, we first search for data merging/fusion possibilities, to then encourage the dissemination of the resulting

items in one unique process. Hence, different data from several applications can be combined and disseminated together.

To enable the combined dissemination, we select the most relevant data items (in the scope of a node), put them into a frame and disseminate it in the vicinity. As the process is repeated at every node, some data items will continue to be chosen for further dissemination while others will be kept local, leading to a dynamic horizon defined hop-by-hop and by data item. A network load estimation is used to complement the choosing strategy. For instance, within congested networks, only really relevant items (urgency alerts, accidents) are chosen for dissemination.

In a straight list, the RIF architecture perform the following steps:

1. Merging local information.
2. Assessing information relevance and priority.
3. Constructing a frame with the most relevant information.
4. Disseminating the frame.

5.2 Rationale

There are many solutions proposed for data sharing in VANETs. Some of them focused on traffic events (safety and efficiency), [ETSI Ts 102 637-2, 2010], [ETSI EN 302 637-3, 2014], [Madhukalya, 2012]; other ones developed with multimedia and entertainment data in mind [Ahmadifard et al., 2011], and even solutions intended for generic data [Ducourthial et al., 2007], [Li et al., 2016], [Akabane et al., 2016] (see Chapter 2 for details about these solutions and explanations about additional ones). Even though good results can be achieved with these strategies, a more efficient data sharing process can be offered when adopting a cooperative dissemination strategy which works within a dynamic defined horizon.

In this section, we develop the reasoning leading to our Relevant Information Frame architecture.

5.2.1 Advocacy for a cooperative dissemination

Since vehicles embed more and more sensors and calculators, they are important sources of information for ITS applications. They also require information from other vehicles to increase the accuracy, confidence and pertinence of their embedded processes. Such processes are intended for cooperation, environment perception,

danger mitigation, automatic driving and so on. Hence, the question of sharing local information with others is a core issue.

Data sharing algorithms generally adopted in usual networks cannot be applied in the context of vehicular networks. Such networks are dynamic, with connections breaking regularly. Moreover, a node could have a given neighbor during a single and very short time period, leading to failures of most classical data sharing algorithms. Nonetheless, such data sharing algorithms would require an almost stable group of nodes, reporting the difficulty to the group membership algorithms, which is not a straightforward task in VANETs [Ducourthial et al., 2010].

Another approach consists in data dissemination algorithms [Villas et al., 2013, Akabane et al., 2016, Chelha and Rakrak, 2015]. However, with the hypothesis of each vehicle hosting different applications with something to share with others, the problem consists in, from a theoretical point of view, a *gossip* [Hedetniemi et al., 1988]. Such algorithms bring the drawbacks of message complexity, bandwidth consummation and also the so-called broadcast storm.

A different strategy is offered by reactive algorithms [Mishra et al., 2011, Dieudonné et al., 2012, Liu et al., 2013]. For instance, in data collect algorithms, a node starts the collection process by indicating which information is required. Other involved nodes may forward or answer the request. The advantage of such a scheme is that messages are generated only when a given information is requested but, at the cost of longer periods of communication.

Nevertheless, data collection is complementary to data dissemination and cannot replace it. Collection algorithms are interesting for non usual information - relevant for specific vehicles and with larger time constraints. To the contrary, dissemination algorithms are interesting for general purpose information - relevant for the most part of vehicles and usually presenting hard time constraints.

Despite the strategy adopted (proactive or reactive), with different applications focused on their own subject and scope, a lot of individual communication processes will take place. Such a behavior has the disadvantages of requiring as many processes as applications and not performing a global evaluation of the available data neither of the communication spectrum.

An interesting approach to minimize these problems would be to encourage the cooperation among different applications. First, because local information may be combined, reducing the total amount of data available and potentially increasing information accuracy and relevance [Zoghby et al., 2014]. Second, a unique and appropriate dissemination frequency may be defined to accommodate the requirements of individual local applications. For instance, the same traffic event could have been detected by different applications. It should be enough to have just

one dissemination of this information.

The Local Dynamic Map - LDM - is a proposal in this direction. It is a conceptual data store, located in the ego-node, containing information on objects influencing or being part of traffic [ETSI TR 102 863, 2011]. Data can be received from several sources like local applications, neighbors, RSUs, traffic centers and on-board sensors. LDM mechanisms are provided to applications in order to grant safe and secure data access. Therefore, several applications can benefit from the available data. For instance, a traffic monitoring application could use data from collision avoidance or weather forecast applications to improve its results.

5.2.2 Advocacy for a dynamic horizon

To mitigate bandwidth consummation and broadcast storm problems, we can also limit the dissemination process in time and space, defining the concept of *horizon*.

Definition 1. Horizon *is the physical area achieved by a data item disseminated in the network.*

Time limitation consists basically in determining the ending date of the dissemination. Limiting the dissemination space usually consists in defining a maximum distance in hops from the source node.

Restricting data exchange to the vicinity (1-hop neighbors) usually relieves the broadcast problem, allowing higher sending frequencies. It is often used by safety applications where tight delays are the rule.

Extending this action to n hops from the source allows the identification of distant hazards and services opportunities. However, since n-hop communication requires re-forwarding actions, with probably multiple nodes sending the message in the same area, the mentioned problems have to be addressed with more attention.

Fixed destination addresses, geographic and network partition solutions do not present limitations in number of hops but rather based on the physical positions of nodes [Akabane et al., 2016, Chelha and Rakrak, 2015]. Such strategies often require a common map shared among all users (all nodes have to know the division adopted), along with path discovery and maintaining algorithms [De Medeiros et al., 2017].

Due to ephemeral connections and network density variations, such necessities become quite complex to meet. Managing shared maps, network partitions and areas identification is costly for the network. Multi-hop paths are still worst. Too many messages are required to keep this information up to date. Additionally, sending data to distant areas might be useless. The dynamic behavior of VANETs imposes

local references to data items - what is relevant in one area might have no value in another place. In particular, available parking places in the vicinity are often more relevant than an information about a distant deviation.

Instead of fixing a destination, regardless the strategy, an iterative approach focusing on one hop at a time would be more reasonable and effective. With nodes spreading information only in its vicinity, it is not required to manage multiple-hop paths neither areas identifications. Reaching distant areas would still be possible thanks to data forwarding. In such situations, a node assesses all data locally available in order to set a relevance ordering. Next, it transmits data items starting from the most important one. The amount of data transmitted will be defined by the network condition: few items in a congested network; a lot of items in free networks.

For instance, let consider the data items: accident, available parking places and deviation. Usually, data about accidents has priority over data about available parking places; available parking places in the vicinity are often more relevant than an information about a distant deviation. A node s disseminates these three data items in its vicinity. A receiver r will add these data to its local data base and perform its own local evaluation. Node r may have other data items classified as more relevant (e.g. presence of near obstacles). It may also find that the deviation data is not relevant in its area. So, items disseminated by r will not be the same of that disseminated by s.

The horizon achieved by a data item is then dynamically defined, in a hop-by-hop strategy. Hence, it is not possible to know, a priori, which will be the resulting horizon. It is evaluated by each node, at each hop, in a cooperative approach.

5.3 The Relevant Information Frame Architecture

The Relevant Information Frame (RIF) Architecture brings as main goal the efficient information sharing within VANETs. It was thought as a cooperative data sharing service which uses a dynamic defined dissemination area (horizon). RIF works below other applications, selecting their data, inserting it into a frame of relevant information and disseminating it in the vicinity of the ego-node.

5.3.1 Proposed Architecture

Figure 5.1 shows a general view of the proposed RIF architecture embedded in a VANET node. The local database is composed by data from different sources (sensors and applications), including remote RIF instances. A RIF instance accesses the database to periodically perform the following steps:

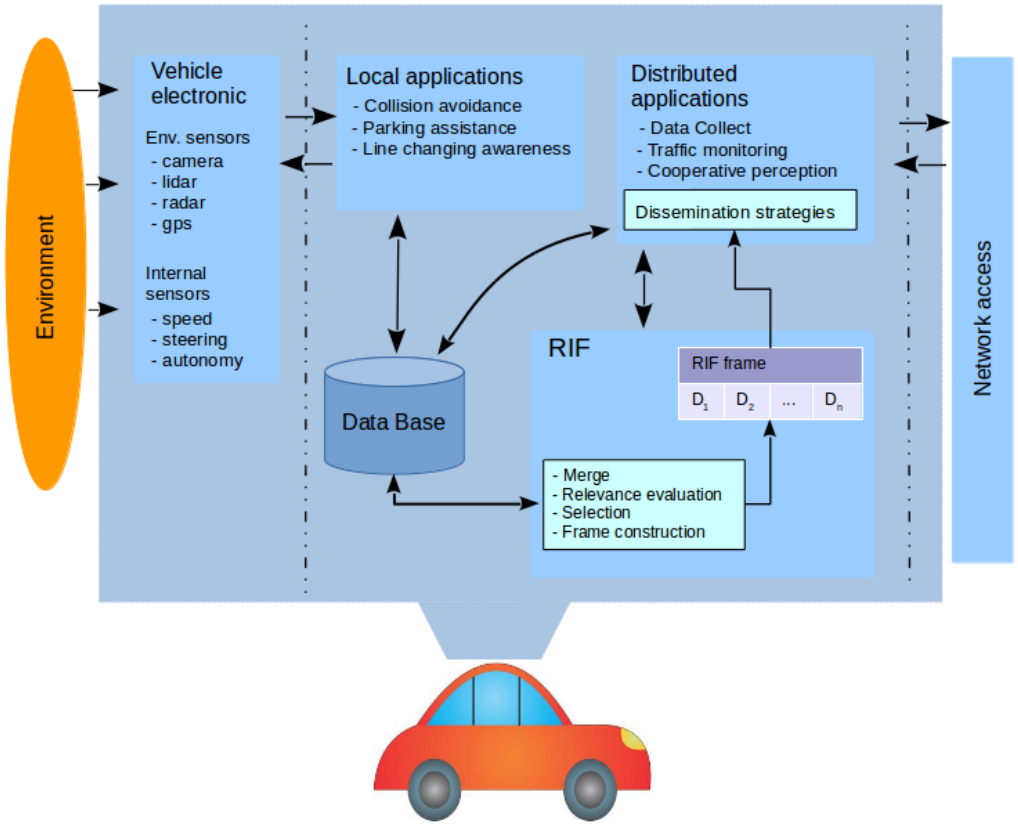


Figure 5.1 – High level view of RIF architecture for a VANET node

1. Merging local information
2. Assessing information relevance
3. Constructing a frame with the most relevant information
4. Disseminating the frame.

The following sections explain each one of these main steps.

5.3.2 Merging local information

Every node has a database (DB) containing data from either local and distributed applications (see Fig. 5.1). Each one of these applications may perform its own data evaluation respecting information issues as Reliability, Temporal and Geographic persistence, Accuracy and Confidence. Thus, the local DB is populated with the information itself and possibly a related quality evaluation. For instance, applications detecting obstacles may use different sensors to capture data while performing an average computation of the captured values in time. Only the averaged value is submitted to the database.

RIF's strategy here is to go through the local database searching for correlated data items in order to merge them. Different merging strategies may be used here. The merge process typically produces information with more quality: accuracy may be improved by merging different measures of the same data, as well as outdated information may be revalidated, augmenting its confidence, when merging it with new received data.

In particular, an application which has an outdated information about a deviation and which receives new messages about it, can increase its confidence on the data and renew its validity in time; An inaccurate information about a traffic jam can be corroborated or contradicted when combining it with data about the average speed of nodes in the same area. Despite the result, doubts can be clarified.

5.3.3 Assessing information characteristics

RIF evaluates the information relevance with a heuristic based on data Temporal and Geographic dependence, Confidence and Priority.

Temporal and geographic dependence With vehicles moving almost all the time, changing their dynamics and network conditions, the self-gathered information becomes inherently local. Vehicle's sensors have a limited range of action. They can sense data only in their vicinity, capturing data related to a small geographic area around the ego-vehicle. To the contrary, when receiving data from other nodes, the related area can be largely increased. Information from an area can be iteratively forwarded to distant nodes. In both cases, a reference in space exists.

Dynamic environments bring yet temporal concerns. Data usually presents a validity in time. Neighbors are nearby for a while. Obstacles, accidents, gateways, etc. exist or are available for a finite amount of time.

Confidence The local database of a VANET node can be populated by data from several sources. Despite the source of a data item (embedded sensors, local and distributed algorithms, etc.), it may be described as: "the value is between 10 and 20"; "the value is probably 15"; "the value is probably between 10 and 20". The first value is imprecise, whereas the second one is uncertain and the latter is both imprecise and uncertain. Confidence is related to the data certainty [Ducourthial et al., 2012, Ducourthial and Cherfaoui, 2016].

Priority Within a VANET environment, data items present different priorities. An ordered classification of them could be: I) Critical II) Important III) Informational. "Critical" represents the highest level and "Informational", the lowest one. In

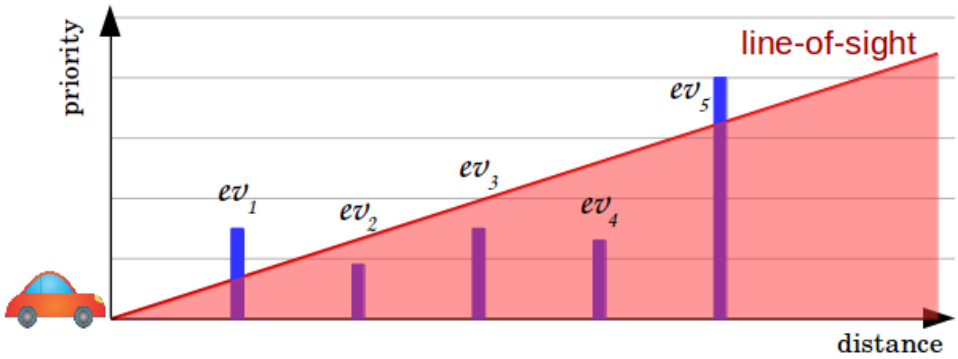


Figure 5.2 – Line-of-sight for an ego-vehicle

particular, an accident is usually classified as Critical, whereas service-stations availability is often classified as Informational.

5.3.4 Constructing a frame with the most relevant information

After merging and assessing data items according to the former characteristics, RIF counts with a set of data considered to share. Often, there is much more information to share than a frame could contain. RIF has then to select the most relevant items among the available ones. Such a selection is performed hop by hop, with every node choosing what to send by means of a heuristic. This behavior leads to a dynamic horizon of dissemination, as explained earlier.

Heuristic for the relevance computation Let consider the ego vehicle on a road and represent traffic events ahead as vertical segments. Each segment is positioned according to its distance from the ego vehicle, and its height is related to the event priority (Fig. 5.2). A simple heuristic consists in plotting a line-of-sight of the vehicle (the red line in Fig. 5.2) and checking events' positions with respect to this line. All events surpassing the line shall be considered as relevant: a near event will then be considered relevant with a medium priority (e.g. ev_1), while a far event will be considered only if its priority is high (e.g. ev_5).

In order to apply such heuristic, we need to sort the data items according to the distance and the priority. We then have to translate the time, distance, confidence and priority attributes to only two attributes, namely distance and priority.

Therefore, RIF first eliminates data with very low Confidence. Next, temporal and geographic references are merged into only one reference. Considering that RIF works with a defined dissemination frequency, it is possible to associate the elapsed time, since the data was first sent, with the distance reached in terms of hops. In

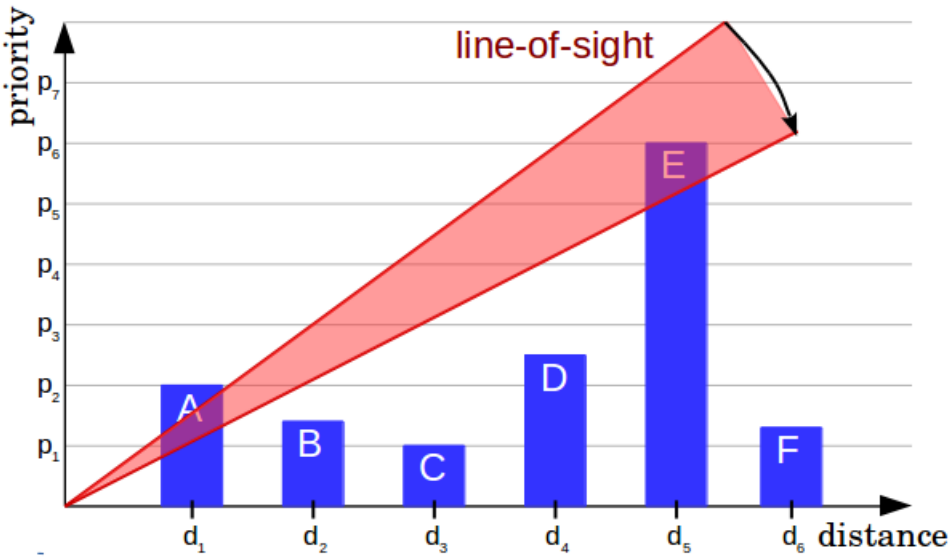


Figure 5.3 – Line-of-sight for relevant items selection

particular, when a node receives data items with an elapsed time of ten seconds, and RIF is working in the frequency of one hertz, it is clear that the maximum distance achievable is ten hops.

Figure 5.3 illustrates an example with six (A to F) data items plotted as explained. RIF applies the heuristic by taking a line-of-sight near of the vertical axis (priority), and then rotating it in direction of the horizontal axis (distance). During the rotation, every time a data item "touches" the line-of-sight, this item is selected to be in the frame. The process continues until the frame is filled up or the horizontal axis is achieved. The process is similar to taking data items according to the result of the equation

$$arctan\left(\frac{d_i}{p_i}\right)$$

In order to minimize fragmentation, and to respect the maximum size of the physical layer PDU, RIF shall work with a parameter to the maximum frame size.

5.3.5 Disseminating the frame

Once the information frame is ready, it has to be disseminated in the vicinity of the ego-node. As described in Section 2.4, the periodicity, or frequency, of sending actions is not easily defined. High frequencies may overload the network while low frequencies may loose relevant data. The application AND (Chapter 3) presents a good reasoning about message frequency within VANETs, it is though intended for 1-hop communication with high inter-message-delay constraints.

The information evaluation performed earlier can help in this step. If the goal

is to spread information about dynamic obstacles as animals or pedestrians in the route (data with low persistence and high priority), the sending frequency shall be high. In contrast, a traffic jam or a big accident (high persistent data) shall use a lower frequency.

Nevertheless, both types of information might be chosen to be sent in the same frame. In order to accommodate so different requirements, RIF was thought to use the highest sending frequency possible before overloading the communication spectrum. In this sense, RIF shall use a network condition evaluation similar to that used by the application AND, increasing or decreasing its dissemination frequency accordingly.

Although RIF architecture presents a proposal for the message dissemination, it may use other applications, optimized for this task (see Fig. 5.1). For instance, the conditional transmission approach offered by the application HOP [Ducourthial et al., 2007] can be used. The main idea in HOP is to replace addresses by conditions in order to enable sending actions to destinations like "those who are behind the sender", "those who are in a given geographical area", "those who can offer a specific service", etc.

Regardless the strategy, RIF proposes that a Push approach should be the natural behavior, with nodes broadcasting their frames in the vicinity, in a proactive algorithm.

5.3.6 RIF algorithm

In this section, a high-level algorithm for RIF is presented.

Algorithm 3: **RIF algorithm for any node** v

```
1  Upon aTimer expiration:
     ▷ Heuristic for the relevance computation
2    Discard data with very low Confidence
3    Convert Time to Distance
4    Sort data items according to arctan(d_i / p_i)
     ▷ Fill the RIF frame with the most relevant data
5    while n do  ot filled
6      select the most relevant item
7      insert it into the frame
8    end while
     ▷ Broadcast the RIF frame in the vicinity
9    send( RIF frame )  to neighbors
10   Restart aTimer                    ▷ Applying AND strategy for timer computation

11 Upon RIF reception:
12   receive( RIF frame )
13   Extract data from the frame
```

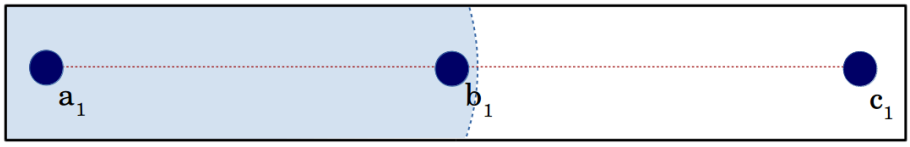


Figure 5.4 – Low density scenario represented by a single lane road

▷ *Update local database*
14 Merge local information with the received one
15 Assess information characteristics ▷ *Accuracy, Temp. and Geog. references, Priority*

5.4 Analytical study of RIF

In this section, we provide an analytical and comparative study of the RIF performance.

5.4.1 Scenarios

In a general scenario, vehicles host local applications which periodically inform other applications (in other vehicles) about their results.

We consider two communication strategies in this study:

- One dissemination per application: each application periodically disseminates its own local information;
- RIF: a single frame is propagated with several data items. Its content is determined at each hop.

The propagation of messages in a network is heavily impacted by nodes density. A very connected network can be assimilated as a complete network (each node is neighbor of all the others) while at the other extremity, we have a disconnected graph (a node has no neighbor). Between these extreme scenarios, we chose three intermediate topologies: I) a low density network on a country side road (see Fig. 5.4); II) a medium density network on a two-lanes road (Fig. 5.5); III) a high density network on a congested four lane highway (Fig. 5.6).

For the sake of simplicity and to avoid any border effect in the comparisons, we consider a portion of such networks (supposed to be larger) composed by n nodes. Moreover, we suppose that every node has pertinent data for others (whatever is the mean used to propagate them).

We compared the two strategies on these three networks. The metrics considered for the comparison were the following:

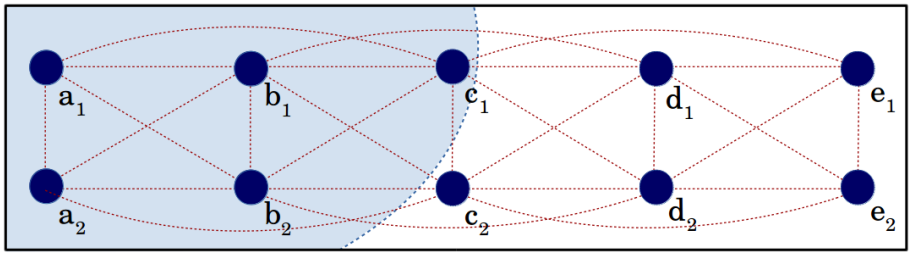


Figure 5.5 – Medium density scenario represented by a two lanes road

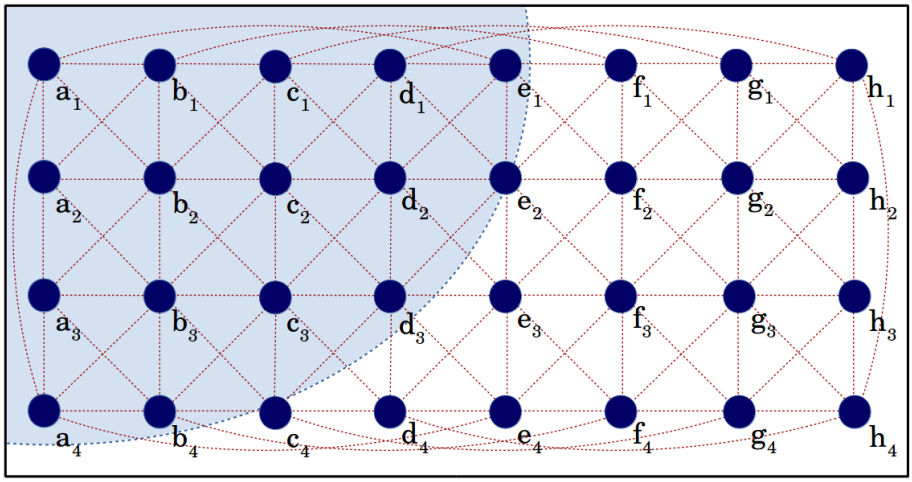


Figure 5.6 – High density scenario represented by a four lanes road with traffic jam

- Number of different messages generated (M_{diff}): It represents the sum of messages firstly created and sent, per second, by each node in the network.

- Total number of messages exchanged in the area if reference (M_{total}): Every message in M_{diff} might be either forwarded or not by a receiver node. Adding all forwarded messages to M_{diff} yields the total number of messages exchanged in the network.

- The total amount of time (Delay) required to disseminate all local relevant data.

5.4.2 Number of different messages generated

The number of different messages generated in a network is directly related to the number of nodes, the number of hosted applications and the sending frequency of such applications. Let $V = \{v_1, v_2, ..., v_n\}$ be the set of nodes in the network, $A = \{a_1, a_2, ..., a_m\}$ the set of applications running in every node of V and $F = \{f_1, f_2, ..., f_m\}$ the sending frequency of each application in A. Considering a data dissemination strategy, the total number of different messages, generated per second,

is given by the equation:

$$M_{\text{diff}}^{D} = n \sum_{i=1}^{m} f_i \tag{5.1}$$

RIF, however, is capable of reducing the number of messages sent thanks to its strategy of cooperative dissemination. With data from different applications being disseminated within a unique RIF frame, it shall be necessary only one message per node, not one per application. Equation 5.1 considering RIF becomes then:

$$M_{\text{diff}}^{R} = nf \tag{5.2}$$

With RIF's sending frequency defined as described in Section 5.3.5.

5.4.3 Total number of messages exchanged in the network

When considering direct communication (1-hop), the total number of messages exchanged in the network equals the number of messages generated by the sender (M_{diff}). Nevertheless, forwarding actions are necessary to reach distant areas, increasing the number of required messages to accomplish with the dissemination. The number of forwarding actions is directly related to the diameter of the network.

5.4.3.1 Diameter evaluation

A network diameter is the maximum number of hops between any pair of nodes. Taking Fig. 5.4 as reference, a message from vehicle a_1 reaches vehicle c_1 in 2 hops ($D = 2$). In fact, for low density networks with n nodes (e.g. Fig. 5.4), the diameter is given by the equation

$$D = n - 1$$

In another way, within more dense networks, a node has more connections, reaching more neighbors and saving forwarding actions. For instance, a_1 in the network of Figure 5.5 reaches c_1, the second node in lane, directly. Hence, the diameter in this case is divided by 2, due to the two lanes road, and by 2 again, due to the longer communication reachability. We have then:

$$D = \frac{\frac{n}{2}}{2} \implies \left\lceil \frac{n}{4} \right\rceil \tag{5.3}$$

With the same reasoning, we have for the high density network depicted in Figure 5.6:

$$D = \frac{\frac{n}{4}}{4} \implies \left\lceil \frac{n}{16} \right\rceil \tag{5.4}$$

5.4.3.2 Number of messages

From a given network diameter and message dissemination strategy, it is possible to estimate the total number of messages per second, required for message dissemination with Equation 5.5.

$$M^D_{\text{total}} = nD \sum_{i=1}^{m} f_i \tag{5.5}$$

With RIF, we have

$$M^R_{\text{total}} = nDf \tag{5.6}$$

In fact, the number of messages per second required by RIF strategy is defined by the amount of relevant data present in each node. Only one message is necessary in cases where all local data fits to a unique RIF frame, whereas m messages shall be necessary in the other extreme case (one data item per RIF frame). Therefore, an ordinary dissemination strategy equals RIF in number of messages in cases where every relevant data item is large enough to completely fill the RIF frame, or in cases where there is only one item to be sent.

5.4.4 Delay

Given the number of messages exchanged in the network, the corresponding amount of data transfered can be estimated with basis on the message size. A message is composed by headers (h) and payload (p). With estimations for the message size and the total number of messages exchanged in the network, we can obtain the total amount of data (T_{data}) transfered with the Equation 5.7.

$$T_{\text{data}} = M_{\text{total}} \times (h + p) \tag{5.7}$$

The delay to transfer a message from one node to its neighbor can be calculated by dividing the message size by the network bandwidth (Delay $= M_{\text{size}}/B_w$). Further, when multiple applications are hosted by the same node, the sending process occurs sequentially, leading the delay for sending m messages to $m \times M_{\text{size}}/B_w$. For instance, considering previous calculus and equations, we can define the delay to disseminate all relevant data in the network as:

$$\text{Delay}^D = M^D_{\text{total}} \times \frac{M_{\text{size}}}{B_w} = nD \sum_{i=1}^{m} F_i \times \frac{M_{\text{size}}}{B_w} \tag{5.8}$$

As RIF produces less messages in the network (see Equation 5.6), it also achieves

lower delays. For instance, in case of many applications in the same node, it wont be necessary to sequentially send multiple messages. RIF can construct a unique frame and send only one message, significantly reducing the delay. Equation 5.8 for RIF becomes

$$\text{Delay}^R = M^R_{\text{total}} \times \frac{M_{\text{size}}}{B_w} = nDF \times \frac{M_{\text{size}}}{B_w} \tag{5.9}$$

5.4.5 Summarizing

Table 5.1 summarizes the performed analysis by comparing the chosen strategies (Dissemination, RIF) according to the metrics: Number of different Messages (M_{diff}) and Total number of messages (M_{total}) per second, Total amount of time to disseminate the data (Delay). It can be seen that RIF has a great potential to improve such metrics.

Table 5.1 – RIF theoretical comparison summary

	M_{diff}	M_{total}	Delay
Dissemination	$n\sum_{i=1}^{m} F_i$	$nD\sum_{i=1}^{m} F_i$	$nD\sum_{i=1}^{m} F_i \times \frac{M_{\text{size}}}{B_w}$
RIF	nF	nDF	$nDF \times \frac{M_{\text{size}}}{B_w}$

5.4.6 Practical example

Consider a vehicle traveling in a highway. It detects, by means of sensors and local applications, heavy rain and some small accidents. For the sake of a safer traffic, this information is disseminated by weather forecast and accident detection applications - different applications working independently. Another vehicle, receiving such an information before reaching the related area, can use a third application to search for alternative paths, avoiding the area. We have then, three independent applications.

RIF however, can bring some improvements to the described scenario. The first vehicle keeps its two separate applications but, broadcasting the data is performed by RIF. A unique, or few RIF frames with weather and accidents information are spread out. In the same way, the second vehicle is capable of disseminating all three information items (heavy rain, small accidents, alternative paths) within few RIF frames.

From the metrics presented and considering some hypotheses, we can rewrite the Table 5.1 with focus on this example. To compose Table 5.2 we have considered: headers of sizes: UDP (8 bytes); IP (20 bytes); 802.11p (32 bytes); Physical layer

(24 bytes); composing a total of 84 bytes per message ($h = 84$); a medium density network ($D = n/4$); twenty nodes ($n = 20$); three applications ($m = 3$); average payload size of 400 Bytes ($p = 400$); average frequency $F = 1\,\text{Hz}$; bandwidth $B_w =$ 6Mbps.

Table 5.2 – Practical example for RIF comparison

	M_{diff}	M_{total}	Delay (sec)
Dissemination	60	300	0.024
RIF	20	100	0.008

Since RIF will send its messages in the highest allowed frequency - before messages losses - (see Section 5.3.5), we can expect that messages will be delivered as soon as possible.

5.5 Conclusions

In the context of ITS, vehicles represent important sources of data. Sharing this data can improve individual environment knowledge and significantly increase the effectiveness of such systems. Nevertheless, environment knowledge like cooperative perception, neighbors mapping, etc. requires a lot of messages. When considering different applications working at the same time, the problem becomes even worst.

The Relevant Information Frame - RIF - architecture proposed here decreases the number of messages sent by selecting the most relevant data items in a node and disseminating them in the fewer possible number of frames. Moreover, RIF avoids area identification or multi-path algorithms by adopting a dynamic hop-by-hop horizon strategy. The relevance criteria of each information is evaluated at the receiver side, limiting the control and avoiding to spread out a message on the basis of the initiator node. This collaborative strategy avoids trying to define the horizon of an information before sending it. Our validation and comparison study (Section 5.4) shows that the gain becomes rapidly important.

Future works encompass an implementation of RIF with focus on practical evaluation. Experiments in simulated scenarios shall be used for tunning RIF parameters and for comparing the architecture to other data sharing solutions. Additionally, a formal definition for data priority classification would leverage RIF results and standardization. To best of our knowledge, there is no standard definition for data classification widely accepted in ITS systems.

Concluding Remarks

Contents

6.1 Conclusions

In this thesis we have addressed the subject of data sharing in vehicular ad-hoc networks (VANETs). Playing the rule of a building block for ITS (Intelligent Transportation Systems) development, VANETs really present a lot of possibilities and challenges. Challenges arise mainly due to the fact that VANETs compose very dynamic environments with communicating entities moving all the time, changing network topology, density and scale (Chapter 1).

When focusing on data sharing in VANETs, the challenges faced can be represented in a high level by the questions of Chapter 2: "What to share?", "In which frequency?" and "Up to what distance from the source?". In the search for answering such questions, we adopted a progressive study, in terms of area of interest.

We start with studies on data sharing in the vicinity of a node, considering only direct neighbors (1-hop). In this stage, an Adaptive Neighbor Discovery (AND) algorithm was proposed. AND adapts its messaging frequency according to nodes' dynamics and the communication spectrum load. An important implementation work has been done in order to validate our proposal. Appendix A describes in details this implementation. Moreover, the emulation tool of the team (Airplug EMU [Buisset et al., 2010]) has been extended with a more realistic packet loss model (Appendix B). With the new EMU version, emulation experiments were conducted. Obtained results showed that AND is able to adapt to the dynamical characteristics of the nodes and to the wireless network loss rate while preserving very good accuracy in neighbors discovery [Moraes and Ducourthial, 2016].

A second stage was started by extending the area of interest to n hops. The main goal of this step was to identify nodes with which it would be more appropriate to share data and services. In this sense, the Cooperative Neighborhood Map (CNM) algorithm constructs a map of neighbors, up to n hops of distance, while evaluating them. The CNM evaluation is performed in terms of trust and path reliability. Trust represents the degree of subjective belief on nodes information. Path reliability is specially relevant for offering/requesting services in the network, e.g. an offered service cannot be used if too many messages are lost during the communication. CNM has been validated by means of extensive study using Airplug. Very interesting properties have been shown. In particular, the Airplug application MAP (see Chapter 4) was improved with new graphic properties to yield better visual results. It is expected that the resulting map can be used by latter applications in order to choose nodes to communicate with [Moraes and Ducourthial, 2018].

The third phase was conducted with a relaxed definition of the area of interest. There was no static definition of the desired area, but rather a dynamic evaluation of data items to be shared. Thus, some items may achieve distant areas while others keep limited to the vicinity or to few hops. In this context, the Relevant Information Frame (RIF) architecture was proposed. By means of RIF, a node is capable of selecting the most relevant information among all available one and to periodically share it. RIF proposes yet a dynamic horizon (area of dissemination) evaluated at every hop according to the data being disseminated. Hence, each receiver evaluates if the data item should be forwarded or not. A comparative study was performed with RIF and ordinary data dissemination strategies. Results have shown that RIF has a great potential to improve data sharing within VANETs.

From the obtained results, we can conclude that our three algorithms present interesting answers to the proposed questions. Moreover, the progressive study adopted to address the data sharing problem proved to be a wise choice.

6.2 Future works

Concerning AND and CNM, two algorithms already tested in simulated scenarios, we envisage tests in real scenarios by means of small robot cars and then with real vehicles. Such tests should bring insights for tunning the parameters used in both algorithms. In particular, with CNM we would like to deeply evaluate the trust updating steps. The aim would be to improve the interpretation of probabilities and certainty of the shared data.

Our Relevant Information Frame - RIF - architecture was evaluated by means of analytical and comparative studies. The next step is to perform a practical

implementation of RIF by using the AirPlug framework with focus on practical experimentation and comparison of RIF. The heuristic defined needs to be challenged in more realistic scenarios.

We believe that data sharing techniques in general need to be evaluated in a large real testbed to be proven in terms of scalability and applications' requirements. Such experiments can give precise answers for questions like: Will these techniques work in a real VANET with dozens, hundreds, of nodes? Are they capable of meeting specific requirements from different applications? Are they capable of meeting users' needs? What is the impact of real scenarios characteristics and constraints on these solutions?

From a theoretical point of view, we investigate the impact of dynamics on different algorithms. This work has began with the approach of [Ducourthial and Wade, 2016]. Regarding a technological bias, additional issues arise when considering new technologies intended to the vehicular environment. The most part of the available data sharing techniques for VANETs suppose the use the IEEE 802.11p WAVE standard. Nevertheless, the advent of new technologies (e.g. 5G networks) brings new possibilities and it would be interesting to consider them.

Finally, studies and research works are visualized in the context of smart cities and Internet of Things (IoT). For instance, applications proposed to act in such scenarios can be adapted to take advantages from the CNM map. It is expected that specific applications improve their results by using the CNM map.

Detailed AND algorithm description

Contents

A.1 AND algorithm description

AND is an event-driven algorithm, reacting on message arrivals and timer expirations. Hence, at every message reception, the receiver node saves sender's data (sequence numbers, network reliability estimation, neighbors), computes the distance and relative speed between them (Lines 54 to 58) and checks for problems in the neighborhood. The relative speed and distance values are used to obtain the sender's neighborhood lifetime (Line 64). Problems in the neighborhood are related to supposed unknown neighbors (Lines 65 to 74). Considering node u as the sender and node v as the receiver, v verifies the list of neighbors of u. If it exists any node within the core range ($R/2$) of u which is not neighbor of v, messages are being lost (v should have recognized these nodes). The obtained value is used later to complement the network reliability estimation (Line 75).

The constant `aTimer` represents the lowest time interval allowed (100 milliseconds) in the algorithm. At every `aTimer` expiration, AND firstly checks whether is it time to send a message or not, then it deletes the data related to old neighbors (neighbors that did not send messages recently) according to their lifetime (Line 34). This way, the node keeps an up to date neighborhood. Following, if any message has arrived since the last timer expiration, the network parameters are updated (Lines 37 to 39), and the network reliability estimation is recomputed. Finally, IMD and `timeToSend` are updated accordingly.

In order to estimate the network reliability (ComputeReliability() procedure), AND uses a cooperative approach where each instance keeps counters and sequence

numbers for every received message. They are used to compute how many messages were received (msg_rcvd) and lost (msg_lost). The msg_lost counts also the number of unrecognized neighbors. Different counters and sequence numbers are kept for each identified neighbor during its neighborhood lifetime. From these values, a local reception rate is estimated. This local estimation is sent within beacon messages enabling every node v to compute an estimation for the overall neighborhood. The overall estimation is obtained through a sum of nodes' local estimations (stored in the array local_rate), weighted by the number of neighbors of each node (stored in the array neigh_numb). See Line 82.

Nevertheless, results obtained by this approach might present very large variations. In order to smooth the obtained values, a weighted average with the previously computed (old_neigh_rate) and the current reception rate is used. The final rate is then given by the equation in Line 83. Finally, using this reception rate estimation, the number of sending attempts ρ can be computed according to Equation 3.10 (Line 84).

After updating the number of nodes and network reliability, AND is capable of renewing the inter-messages delay (UpdateIMD() procedure). It is an easy perception that without packet losses, sending more messages leads to a more precise neighborhood knowledge. So, AND is always searching for the lowest IMD value allowed, i.e., the lower bound (Line 97). AND keeps this behavior until it detects packet losses. In this case, the IMD is decreased according to an Additive Increase Multiplicative Decrease (AIMD) approach (Line 95). Hence, every time the network conditions appear to be better, the sending frequency increases using an additive factor. Every time the network conditions appear to be worst, the sending frequency decreases using a multiplicative term. Such a strategy ensures rapid convergence.

A.2 AND algorithm

Algorithm 4: **AND algorithm for any node** v

```
1  Starting_action:
     ▷ Parameters
2    R ← 500 meters                          ▷ Communication range fixed value
3    aTimer ← 100                            ▷ Lowest timer value allowed in milliseconds
4    lower_bound ← aTimer                    ▷ IMD lower bound
5    upper_bound ← 1000                      ▷ IMD upper bound in milliseconds
6    s_max ← 130/3.6                         ▷ French maximum speed in m/s
7    q ← 0.99                                ▷ Assurance of 99% for a message reception
```

▷ *Timers to control the algorithm behavior*
8 IMD ← upper_bound ▷ *Time value between consecutive sent messages*
9 timeToSend ← IMD ▷ *Timer to send the next message*
▷ *Global variables initialization*
▷ *GPS_{node} is a record with the fields: latitude, longitude, altitude, speed, heading*
10 GPS_v ← GetGPSData()
11 msg_rcvd ← 0 ▷ *Number of received messages during aTimer*
12 $\rho \leftarrow 1$ ▷ *Number of attempts required to ensure a message reception*
▷ *Arrays indexed by nodes' IDs*
13 tab_neigh[ ] ← ∅ ▷ *Known neighbors' IDs*
14 tab_seq[v] ← 0 ▷ *Sequence numbers of received messages*
15 tab_rel[v] ← 1 ▷ *Estimated network reliability*
16 tab_dis[ ] ← ∅ ▷ *Distance values between nodes*
17 tab_speed[ ] ← ∅ ▷ *Relative speed values between nodes*
18 tab_life[ ] ← ∅ ▷ *Neighborhood lifetime values*

19 Upon aTimer expiration:
20 timeToSend ← timeToSend - aTimer ▷ *Counting down timer to send a message*
21 **if** timeToSend ≤ 0 **then**
▷ *snd_ IMD timer expired. Send a message.*
22 tab_seq[v] += 1
23 GPS_v ← GetGPSData() ▷ *Update GPS data*
24 **for each** u in tab_neigh[v] **do** ▷ *Select neighbors to send*
25 **if** tab_dis[u] ≤ (R/2) **then** ▷ *Only nodes within the core range of v*
26 neigh_List ← neigh_List ∪ u
27 **end if**
28 **end for**
29 **send(** tab_seq[v], v, tab_rel[v], GPS_v, tab_neigh[v] **)**
30 snd_IMD ← IMD ▷ *Restart the timer to send a message*
31 **end if**

▷ *Check recognized neighbors*
32 **for each** u in tab_neigh[v] **do**
33 **if** tab_life[u] ≤ current_time **then**
34 Delete u and all related data ▷ *Neighbor lifetime is ended. Drop node u*
35 **end if**
36 **end for**

▷ *If there was a message reception, update related values*
37 **if** msg_rcvd > 0 **then**
▷ *Smoothed number of received messages during aTimer*
38 msg_rcvd_smt ← (msg_rcvd × weight) + (msg_rcvd_old × (1 − weight))
▷ *Smoothed number of lost messages during aTimer*
39 msg_lost_smt ← (msg_lost × weight) + (msg_lost_old × (1 − weight))
40 ComputeReliability() ▷ *Network reliability estimation*
41 **end if**

▷ *After intermediate values updating, compute the new IMD*
42 UpdateIMD()

▷ *Check whether is it required or not to update snd_ IMD*
43 **if** IMD ≤ timeToSend **then**
44 timeToSend ← IMD
45 **end if**

▷ *Update variables to the next iteration*
msg_lost_old ← msg_lost
msg_rcvd_old ← msg_rcvd
msg_lost ← 0
msg_rcvd ← 0

Upon message arrival:
▷ *Considering node u as the sender and node v as the receiver*
receive(seq_number$_u$, u, rel$_u$, GPS$_u$, neighbors$_u$ **)**
msg_rcvd += 1
msg_lost ← seq_number$_u$ - tab_seq[u]
▷ *Save data received from u*
tab_seq[u] ← seq_number$_u$
tab_rel[u] ← rel$_u$
tab_neigh[u] ← neighbors$_u$
tab_dis[u] ← ComputeDistance(GPS$_v$, GPS$_u$)
tab_speed[u] ← ComputeRelSpeed(GPS$_v$, GPS$_u$)

▷ *Neighborhood space estimation*
if tab_speed[u] ≥ 0 **then** ▷ *Nodes are getting closer each other*
  neigh_spa ← tab_dis[u] + R
else ▷ *Nodes are getting away each other*
  neigh_spa ← tab_dis[u] - R
end if
▷ *Compute neighbor lifetime for node u*
tab_life[u] ← $\frac{\text{neigh\_spa}}{\text{tab\_speed}[u]}$

▷ *Check for problems in the neighborhood. Checking only neighbors within R/2*
unknown_neighbors ← 0
for each $u \in$ tab_neigh[v] **do**
  if tab_dis[u] ≤ (R / 2) **then** ▷ *for every u within the core range of v*
    for each *node* ∈ tab_neigh[u] **do** ▷ *for every neighbor of u*
      if *node* ∉ tab_neigh[v] **then**
        ▷ *u has a neighbor which is not neighbor of v*
        unknown_neighbors += 1
      end if
    end for
  end if
end for
▷ *If there are unknown neighbors within core range, messages are being lost*
msg_lost ← msg_lost + unknown_neighbors

procedure ComputeReliability():
▷ *Node's estimation for network reliability is based on the numbers of received and lost messages*
old_rel ← tab_rel[v]
tab_rel[v] ← $\frac{\text{msg\_rcvd\_smt}}{\text{msg\_rcvd\_smt+msg\_lost\_smt}}$
▷ *Total network reliability estimation is a sum of estimations of every recognized node weighted by the number of nodes' neighbors*

79 **for each** u in tab_neigh[v] **do**
80 neigh_numb[u] ← length(tab_neigh[u])
81 **end for**
82 curr_rel ← $\dfrac{\sum_{i=1}^{nodes} \frac{\text{tab\_rel[i]}}{\text{neigh\_numb}[i]}}{\sum_{i=1}^{nodes} \frac{1}{\text{neigh\_numb}[i]}}$
▷ *The resulting Network Reliability is a weighted average between the current and last measures*
83 tab_rel[v] ← (old_rel × weight) + (curr_rel × (1 − weight))
▷ *Number of attempts required for a successful message reception given an assurance q and the just calculated network reliability*
84 $\rho \leftarrow \log_{(1 \text{ - tab\_rel}[v])} (1 - q)$

85 **procedure** UpdateIMD():

▷ *Before update IMD, update Upper and Lower Bounds*
86 numb_of_neigh ← length(tab_neigh)
87 lower_bound ← aTimer × numb_of_neigh ▷ *Lower Bound equation*
88 tmp_ub ← $\frac{1}{\rho}\left(\frac{R}{\text{speed+s\_max}} - \sigma\right)$ ▷ *Temporary Upper Bound equation*

▷ *Updating the upper bound with a smoothing approach in case of increasing values*
89 **if** tmp_ub > upper_bound **then**
90 upper_bound ← (upper_bound × weight) + (tmp_ub × (1 − weight))
91 **else**
92 upper_bound ← tmp_ub
93 **end if**

▷ *If there was message loss, IMD must increase*
94 **if** msg_lost_smt > 0 **then**
95 tmp_`IMD` ← `IMD` × (2 - tab_rel[v]) ▷ *IMD increases based on the network reliability estimation*
96 **else**
97 tmp_`IMD` ← `IMD` - aTimer ▷ *Without message loss, IMD decreases by an aTimer step*
98 **end if**

▷ *Check whether IMD is between the bounds*
99 **if** tmp_`IMD` > upper_bound **then**
100 `IMD` ← upper_bound
101 **else if** tmpIMD < lower_bound
102 `IMD` ← lower_bound
103 **else**
104 `IMD` ← tmp_`IMD`
105 **end if**

Extension of the Airplug Emulator

Contents

B.1 Airplug

B.1.1 Airplug framework

The Airplug framework was developed in the intention of helping in application development for dynamic ad hoc networks. It has been proposed to reduce the gap between the simulation and the road testbed, in terms of development time and protocol implementation [Ducourthial, 2013].

Airplug is based on **message oriented** communication between local and remote processes. It is self-contained, to avoid relying on libraries that may change; and modular, with a core program and many small applications running in the **user space** of the operational system. Interprocess communications are done by means of standard input and output of each process. The standard error output (`stderr`) is used for printing information when necessary (information, warning, errors). An implementation of the framework is in charge of routing messages from sending to receiving processes, either locally or remotely.

B.1.2 Airplug emulator

The emulation mode of Airplug, named EMU, also takes advantage of shell facilities to artificially manipulate network lower layers (wireless communication). At a first glance, any network topology can be constructed in order to perform tests. Simulated scenarios are defined through XML files where the possibilities include: number of

```
<map width="250" height="250">
    <node id="vehicle1" lossrate="LossRate.csv">
        <app name="AND" exe="./and.tk --whatwhowhere --mode=emu+ ..."/>
        <move type="gpsfile" path="log-a1-compiegne-ouest.gps" ... />
    </node>
    <node id="vehicle2" lossrate="60">
        <app name="AND" exe="./and.tk --whatwhowhere --mode=emu+ ... "/>
        <move type="gpsfile" path="log-a1-compiegne-ouest.gps" ... />
    </node>
</map>
```

Figure B.1 – Example of XML configuration file for the Airplug Emulator

nodes; moving characteristics (by using GPS traces); network reliability and communication range; among others (see Figure B.1). Dynamic topologies can be achieved through creating and deleting pipes between processes [Buisset et al., 2010].

The loss rate, or network reliability, is defined in EMU by a fixed value in XML files. This value is used to randomly discard messages in a rate up to the passed value. Despite efficient and simple to simulate some scenarios, a fixed message loss rate does not mimics the reality. Actually, messages losses in dynamic networks are quite more complex [Han et al., 2012, Campolo et al., 2011].

B.2 A new loss rate emulation strategy

B.2.1 Principle

In order to improve the emulation capabilities of EMU, the broadcast efficiency metric from [Campolo and Vinel, 2011] was incorporated to the emulator. Hence, EMU became capable of dynamically adapting its loss rate according to the Equation B.1.

$$P(l,n,w,k) = \left(1 - \frac{l-1}{w}\right)^n \binom{n}{k} \left(\frac{1}{w-l+1}\right)^k \left(1 - \frac{1}{w-l+1}\right)^{n-k} \tag{B.1}$$

In Equation B.1, the number of nodes in the interference range of the sender (n), and the size of the contention window (w) are used to calculate the probability of success of a transmission. The equation considers yet that $(l-1)$ empty slots pass before the first transmission attempt, and k vehicles transmit in the l^{th} slot.

By means of Equation B.1, we calculated a table of Success Transmission Probabilities. For this purpose, a C++ code was developed and put to run in the UTC servers. The resulting table is indexed by the number of nodes and the

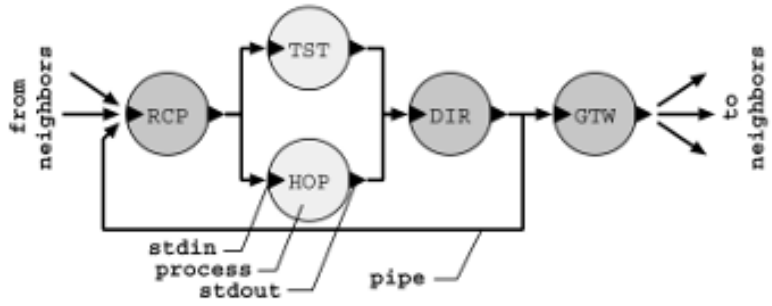


Figure B.2 – Communicating architecture of EMU

contention window size. Table B.1 illustrates an example of the resulting table.

Table B.1 – Success Transmission Probabilities

		Contention Window sizes			
		4	8	16	32
Number of nodes	1	1.0	1.0	1.0	1.0
	2	0.75	0.875	0.937	0.969
	3	0.563	0.766	0.879	0.938
	4	0.422	0.670	0.824	0.909
	5	0.316	0.586	0.772	0.881
	6	0.237	0.513	0.724	0.853
	7	0.178	0.449	0.679	0.827
	8	0.133	0.393	0.637	0.801
	9	0.100	0.344	0.597	0.776
	10	0.075	0.301	0.559	0.751

B.2.2 The new loss rate implementation

EMU uses pipes to exchange messages among process during simulations. Figure B.2 illustrates these pipes connections. RCP, DIR and GTW are internal EMU processes, while TST and HOP are external applications. Basically, RCP receives messages either from external applications and internal processes but, it sends messages only to external applications. DIR receives messages from external applications and sends to internal processes.

Each message received by the RCP module is processed in many ways: it can be directly forwarded to other processes; it can be forwarded after receiving a time delay; it can be discarded, according to the loss rate defined. One of these behaviors is chosen, on the fly, based on experiments' parameters. Most parameters

definitions for an EMU experiment is registered in an XML file, which is managed in a XML processing module (`emu-xml.tk`). So, the main coding amendments for EMU improvement were performed on `emu-xml.tk` and `rcp.tcl` files.

- `emu-xml.tk`
  With the main goal of working as an XML parser, the `emu-xml.tk` module reads parameters' values from an XML file and sets starting values. In order to trigger the new strategy (loss rate variation), an XML syntax must be respected. When defining nodes' characteristics, the loss rate may be defined in one of two ways. For the EMU starting strategy of a fixed loss rate, only the fixed value is required.

  If the new variable loss rate is desired, a file with loss rate values has to be provided. The strategy of using a file with values was adopted as consequence of the time required to compute loss rate values. It is a time-consuming computation process, impossible to be tackled on the fly.

- `rcp.tcl`
  As explained earlier, the `rcp.tcl` is in charge of communication among active processes in the emulation. It is in this module that EMU decides to forward a message directly, forward after applying a delay or discarding it. When using the new loss rate strategy, every time a message is received, EMU uses the number of nodes in the vicinity, along with a fixed contention window size, to search the correct value in the "Success Transmission Probabilities" table. A random number is then drawn and compared to the reference value in the table. If it is smaller than the reference number, the message is sent normally. On the contrary, if it is bigger than the reference value, the message is discarded.

Figure B.1 shows a section of a typical XML file used for setting EMU parameters. In the figure, node "vehicle2" uses a fixed loss rate definition. It means that, during the experiment, EMU will discard 60% of the messages which would be received by "vehicle2". Differently, "vehicle1" uses the new variable loss rate. In this case, a file ("LossRate.csv") is passed with the possible loss rate values to be chosen. The "LossRate.csv" contains the values of Table B.1 in a comma-separated-values (csv) format.

Before starting an EMU experiment, the entire table is read to system memory and then, accessed whenever a loss rate adaptation is required. A starting loss rate indexed by Number-of-Nodes equal to 1 and Contention-Window equal to 16 is used. One node because, in the starting phase, all nodes are alone. EMU has to start other

processes before recognizing nodes within a unique area. A contention window of 16 slots is a suggestion from the 802.11p standard.

Bibliography

[Geo, 2010] (2010). Geonet project, d1.2 final geonet architecture design.

[802.11p, 2010] 802.11p (2010). *IEEE Standard for information technology – Amendment 6: wireless access in vehicular environments.*

[Ahmadifard et al., 2011] Ahmadifard, N., Nabizadeh, H., and Abbaspour, M. (2011). SEFF: A Scalable and Efficient Distributed File Sharing Technique in Vehicular Adhoc Networks. In *Communications (APCC), 2011 17th Asia-Pacific Conference on*, number October, pages 456–460.

[Akabane et al., 2016] Akabane, A. T., Pazzi, R. W., Madeira, E. R., and Villas, L. A. (2016). CARRO: A context-Awareness protocol for data dissemination in urban and highway scenarios. *2016 8th IEEE Latin-American Conference on Communications, LATINCOM 2016*, pages 0–5.

[Almalag et al., 2012] Almalag, M. S., Olariu, S., and Weigle, M. C. (2012). TDMA cluster-based MAC for VANETs (TC-MAC). *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2012 - Digital Proceedings.*

[Bai and Krishnamachari, 2010] Bai, F. and Krishnamachari, B. (2010). Exploiting the wisdom of the crowd: Localized, distributed information-centric VANETs. *IEEE Communications Magazine*, 48(5):138–146.

[Bansal et al., 2013] Bansal, G., Kenney, J. B., and Rohrs, C. E. (2013). LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control. *IEEE Transactions on Vehicular Technology*, 62(9):4182–4197.

[Boldrini et al., 2010] Boldrini, C., Conti, M., Delmastro, F., and Passarella, A. (2010). Context- and social-aware middleware for opportunistic networks. *Journal of Network and Computer Applications*, 33(5):525–541.

[Breu et al., 2014] Breu, J., Brakemeier, A., and Menth, M. (2014). A quantitative study of Cooperative Awareness Messages in production VANETs. *EURASIP Journal on Wireless Communications and Networking*, 98:1–18.

[Buisset et al., 2010] Buisset, A., Ducourthial, B., El Ali, F., and Khalfallah, S. (2010). Vehicular networks emulation. In *International Conference on Computer Communication Networks - ICCCN 2010*, pages –, Switzerland.

[Campolo et al., 2011] Campolo, C., Koucheryavy, Y., Molinaro, A., and Vinel, A. (2011). Characterizing broadcast packet losses in IEEE 802.11p/WAVE vehicular networks. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, pages 735–739.

[Campolo and Vinel, 2011] Campolo, C. and Vinel, a. (2011). Modeling broadcasting in IEEE 802.11 p/WAVE vehicular networks. *... Letters, IEEE*, 15(2):199–201.

[Cao and Wang, 2017] Cao, Y. and Wang, N. (2017). Toward efficient electric-vehicle charging using VANET-based information dissemination. *IEEE Transactions on Vehicular Technology*, 66(4):2886–2901.

[Car2Car Communication Consortium, 2008] Car2Car Communication Consortium (2008). Car2Car Communication Consortium Manifesto - Version 1.1. `http://www.car-to-car.org/`. [Online; Accessed in March, 2018.].

[Chaqfeh et al., 2014] Chaqfeh, M., Lakas, A., and Jawhar, I. (2014). A survey on data dissemination in vehicular ad hoc networks. *Vehicular Communications*, 1(4):214–225.

[Chelha and Rakrak, 2015] Chelha, I. O. and Rakrak, S. (2015). Best nodes approach for alert message dissemination in VANET (BNAMDV). *Proceedings - 2015 3rd International Workshop on RFID and Adaptive Wireless Sensor Networks, RAWSN 2015 - In conjunction with the International Conference on NETworked sYStems, NETYS 2015*, pages 82–85.

[Cherif et al., 2010] Cherif, M. O., Secouci, S. M., and Ducourthial, B. (2010). How to disseminate vehicular data efficiently in both highway and urban environments? *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'2010*, pages 165–171.

[Chiti et al., 2014] Chiti, F., Fantacci, R., and Rigazzi, G. (2014). A mobility driven joint clustering and relay selection for IEEE 802.11p/WAVE vehicular networks. *2014 IEEE International Conference on Communications, ICC 2014*, pages 348–353.

[Cho et al., 2011] Cho, J. H., Swami, A., and Chen, I. R. (2011). A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys Tutorials*, 13(4):562–583.

[CodeRoute, 2003] CodeRoute (2003). Code de la route - Article R412-12. `https://www.legifrance.gouv.fr`. [Online; Accessed in May, 2016.].

[Conti et al., 2011] Conti, M., Mordacchini, M., and Passarella, A. (2011). Data dissemination in opportunistic networks using cognitive heuristics. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6.

[Cornejo et al., 2014] Cornejo, A., Viqar, S., and Welch, J. L. (2014). Reliable neighbor discovery for mobile ad hoc networks. *Ad Hoc Networks*, 12(1):259–277.

[Čurn et al., 2013] Čurn, J., Marinescu, D., O'Hara, N., and Cahill, V. (2013). Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. In *Proceedings of the 16th International Conference on Information Fusion*, pages 68–76.

[De Medeiros et al., 2017] De Medeiros, D. S., Campista, M. E. M., Mitton, N., De Amorim, M. D., and Pujolle, G. (2017). The Power of Quasi-Shortest Paths: ρ-Geodesic Betweenness Centrality. *IEEE Transactions on Network Science and Engineering*, 4(3):187–200.

[Delot et al., 2011] Delot, T., Mitton, N., Ilarri, S., and Hien, T. (2011). Decentralized pull-based information gathering in vehicular networks using GeoVanet. *Proceedings - IEEE International Conference on Mobile Data Management*, 1:174–183.

[Diep and Yeo, 2016] Diep, P. T. N. and Yeo, C. K. (2016). A trust-privacy framework in vehicular ad hoc networks (vanet). In *2016 Wireless Telecommunications Symposium (WTS)*, pages 1–7.

[Dieudonné et al., 2012] Dieudonné, Y., Ducourthial, B., and Senouci, S. M. (2012). COL: A data collection protocol for VANET. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 711–716.

[Ducourthial, 2013] Ducourthial, B. (2013). Designing applications in dynamic networks: The Airplug Software Distribution. In ROY, M., editor, *SAFECOMP 2013 - Workshop ASCoMS (Architecting Safety in Collaborative Mobile Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*, Toulouse, France.

[Ducourthial and Cherfaoui, 2016] Ducourthial, B. and Cherfaoui, V. (2016). Experiments with self-stabilizing distributed data fusion. In *IEEE 35th Symposium on Reliable Distributed Systems (SRDS 2016)*, pages 289–296, Budapest, Hungary.

[Ducourthial et al., 2012] Ducourthial, B., Cherfaoui, V., and Denoeux, T. (2012). Self-stabilizing Distributed Data Fusion. In Richa, A. W. and Scheideler, C., editors, *Stabilization, Safety, and Security of Distributed Systems*, pages 148–162. Springer Berlin Heidelberg.

[Ducourthial et al., 2007] Ducourthial, B., Khaled, Y., and Shawky, M. (2007). Conditional transmissions: Performance study of a new communication strategy in VANET. *IEEE Trans. Veh. Technol.*, 56(6):3348–3357.

[Ducourthial et al., 2010] Ducourthial, B., Khalfallah, S., and Petit, F. (2010). Best-effort group service in dynamic networks. In *Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures*, pages 233–242. ACM.

[Ducourthial and Wade, 2016] Ducourthial, B. and Wade, A. M. (2016). Dynamic p-graphs for capturing the dynamics of distributed systems. *Ad Hoc Networks*, 50:13–22.

[El Ali and Ducourthial, 2011] El Ali, F. and Ducourthial, B. (2011). A distributed algorithm for path maintaining in dynamic networks. In *DYNAM 2011: 1st International Workshop on Dynamicity.*

[ETSI EN 302 637-3, 2014] ETSI EN 302 637-3 (2014). ETSI EN 302 637-3 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of

Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. *Etsi*, 1:1–73.

[ETSI TR 102 863, 2011] ETSI TR 102 863 (2011). ETSI TR 102 863 - Local Dynamic Map (LDM). 1(Ldm):1–40.

[ETSI TS 102 637-1, 2010] ETSI TS 102 637-1 (2010). Vehicular Communications ; Basic Set of Applications ; Part 1 : Functional Requirements. *Intelligent Transport Systems*, 1:1–60.

[ETSI Ts 102 637-2, 2010] ETSI Ts 102 637-2 (2010). Intelligent Transport Systems (ITS) - Vehicular Communications - Basic Set of Applications - Part 2 : Specification of Cooperative Awareness Basic Service. *History*, 1:1–22.

[Füßler et al., 2003] Füßler, H., Widmer, J., KÃďsemann, M., Mauve, M., and Hartenstein, H. (2003). Contention-based forwarding for mobile ad hoc networks. *Ad Hoc Networks*, 1(4):351 – 369.

[Gharba et al., 2017] Gharba, M., Cao, H., Gangakhedkar, S., Eichinger, J., Ali, A. R., Ganesan, K., Jain, V., Lapoehn, S., Frankiewicz, T., Hesse, T., Zou, Y., Tang, C., and Gu, L. (2017). 5G enabled cooperative collision avoidance: System design and field test. *18th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM 2017 - Conference*.

[Giordano and Reggiani, 2014] Giordano, L. G. and Reggiani, L. (2014). *Vehicular Technologies: Deployment and Applications*. InTech.

[Han et al., 2012] Han, C., Dianati, M., Tafazolli, R., Kernchen, R., and Shen, X. (2012). Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):873–886.

[Hartenstein and Laberteaux, 2009] Hartenstein, H. and Laberteaux, K. (2009). *VANET vehicular applications and inter-networking technologies*, volume 1. John Wiley & Sons.

[Hedetniemi et al., 1988] Hedetniemi, S. M., Hedetniemi, S. T., and Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349.

[Huang et al., 2009] Huang, C.-L., Fallah, Y. P., Sengupta, R., and Krishnan, H. (2009). Information dissemination control for cooperative active safety applications in vehicular ad-hoc networks. *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–6.

[Huang et al., 2011] Huang, C. L., Fallah, Y. P., Sengupta, R., and Krishnan, H. (2011). Intervehicle Transmission Rate Control for Cooperative Active Safety System. *Ieee Transactions on Intelligent Transportation Systems*, 12(3):645–658.

[Javed and Khan, 2014] Javed, M. a. and Khan, J. Y. (2014). Performance analysis of an adaptive rate-range control algorithm for VANET safety applications. *2014 International Conference on Computing, Networking and Communications (ICNC)*, pages 418–423.

[Kaiwartya et al., 2016] Kaiwartya, O., Abdullah, A. H., Cao, Y., Altameem, A., Prasad, M., Lin, C.-T., and Liu, X. (2016). Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges and Future Aspects. *IEEE Access*, 4:1–1.

[Kamakura and Ducourthial, 2014] Kamakura, K. and Ducourthial, B. (2014). Experimental validation of cooperative approach near road side units. In *IWCMC 2014 - 10th International Wireless Communications and Mobile Computing Conference*, pages 1010–1015.

[Karagiannis et al., 2011] Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., and Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys Tutorials*, 13(4):584–616.

[Katsaros et al., 2013] Katsaros, K., Dianati, M., and Le, L. (2013). Effective implementation of location services for vanets in hybrid network infrastructures. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 521–525.

[Kerrache et al., 2016] Kerrache, C. A., Calafate, C. T., Cano, J. C., Lagraa, N., and Manzoni, P. (2016). Trust management for vehicular networks: An adversary-oriented overview. *IEEE Access*, 4:9293–9307.

[Kim et al., 2014] Kim, Y., Kim, I., and Shim, C. Y. (2014). Towards a trust management for vanets. In *The International Conference on Information Networking 2014 (ICOIN2014)*, pages 583–587.

[Lakas et al., 2011] Lakas, A., Serhani, M. A., and Boulmalf, M. (2011). A hybrid cooperative service discovery scheme for mobile services in VANET. *International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 25–31.

[Lamport, 1998] Lamport, L. (1998). The Part-Time Parliament. *ACM Trans. Comput. Syst.*, 16(2):133–169.

[Lassoued et al., 2016] Lassoued, K., Bonnifait, P., and Fantoni, I. (2016). Cooperative localization of vehicles sharing GNSS pseudoranges corrections with no base station using set inversion. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2016-Augus(Iv):496–501.

[Lèbre et al., 2014] Lèbre, M.-A., Mouël, F. L., Ménard, E., Dillschneider, J., and Denis, R. (2014). Vanet applications: Hot use cases. *arXiv preprint arXiv:1407.4088*.

[Li et al., 2013] Li, H., Nashashibi, F., and Yang, M. (2013). Split covariance intersection filter: Theory and its application to vehicle localization. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1860–1871.

[Li and Jiang, 2011] Li, Z. J. and Jiang, S. X. (2011). Planning the mobility of routing ferries for intermittently connected mobile networks. In *Proceedings of the 2011 6th International ICST Conference on Communications and Networking in China, CHINACOM 2011*, number 20102302110036, pages 816–821.

[Li et al., 2016] Li, Z. Y., Wu, P. P., Song, Y., and Bi, J. L. (2016). Intermittent data dissemination using node forwarding capability estimation in vehicle delay tolerant networks. *Proceedings - 2016 IEEE Symposium on Service-Oriented System Engineering, SOSE 2016*, (1):147–153.

[Limouchi and Mahgoub, 2016] Limouchi, E. and Mahgoub, I. (2016). Intelligent hybrid adaptive broadcast for VANET. *2016 IEEE 7th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2016*.

[Lin et al., 2017] Lin, T., Rivano, H., and Le Mouel, F. (2017). A Survey of Smart Parking Solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253.

[Liu et al., 2013] Liu, C., Chigan, C., and Gao, C. (2013). Compressive sensing based data collection in vanets. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1756–1761.

[Lyamin et al., 2015] Lyamin, N., Vinel, A., and Jonsson, M. (2015). Does ETSI beaconing frequency control provide cooperative awareness? *2015 IEEE International Conference on Communication Workshop (ICCW)*, (Dvc):2393–2398.

[Madhukalya, 2012] Madhukalya, M. (2012). Event based content fusion in Opportunistic Environments. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, number March, pages 550–551.

[Melo-Castillo et al., 2017] Melo-Castillo, A., Bures, P., Herrera-Quintero, L. F., and Banse, K. (2017). Design and implementation of DATEX II profiles for truck parking systems. In *Proceedings of 2017 15th International Conference on ITS Telecommunications, ITST 2017*, pages 1–7.

[Mishra et al., 2011] Mishra, T., Garg, D., and Gore, M. M. (2011). A publish/subscribe communication infrastructure for VANET applications. *Proceedings - 25th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2011*, pages 442–446.

[Molina-Masegosa and Gozalvez, 2017] Molina-Masegosa, R. and Gozalvez, J. (2017). LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39.

[Moraes and Ducourthial, 2016] Moraes, H. P. and Ducourthial, B. (2016). Adaptive inter-messages delay in vehicular networks. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8.

[Moraes and Ducourthial, 2018] Moraes, H. P. and Ducourthial, B. (2018). Cooperative Neighborhood Map in VANETs. In *Proceedings of the Symposium on Applied Computing*, SAC '18, New York, NY, USA. ACM.

[Mussa et al., 2014] Mussa, S. A. B., Manaf, M., and Ghafoor, K. Z. (2014). Beaconing and transmission range adaptation approaches in vehicular ad hoc networks: Trends & research challenges. *2014 International Conference on Computational Science and Technology, ICCST 2014*.

[Mylonas et al., 2015] Mylonas, Y., Lestas, M., Pitsillides, A., Ioannou, P., and Papadopoulou, V. (2015). Speed adaptive probabilistic flooding for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 64(5):1973–1990.

[Noguchi et al., 2011] Noguchi, S., Tsukada, M., Ernst, T., Inomata, A., and Fujikawa, K. (2011). Location-aware service discovery on IPv6 GeoNetworking for VANET. *2011 11th International Conference on ITS Telecommunications, ITST 2011*, pages 224–229.

[Phe-neau et al., 2014] Phe-neau, T., Amorim, M. D. D., and Conan, V. (2014). Uncovering vicinity properties in disruption-tolerant networks. *COMPUTER NETWORKS*, 73:210–223.

[Radak et al., 2016] Radak, J., Ducourthial, B., Cherfaoui, V., and Bonnet, S. (2016). Detecting road events using distributed data fusion: Experimental evaluation for the icy roads case. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):184–194.

[Rehman et al., 2014] Rehman, O. M., Bourdoucen, H., and Ould-Khaoua, M. (2014). Improving reachability of multi-hop alert messages dissemination in VANETs. *International Conference on ICT Convergence*, pages 510–515.

[Reinders et al., 2011] Reinders, R., Van Eenennaam, M., Karagiannis, G., and Heijenk, G. (2011). Contention window analysis for beaconing in VANETs. *IWCMC 2011 - 7th International Wireless Communications and Mobile Computing Conference*, pages 1481–1487.

[RoadSafety, 2018] RoadSafety (2018). Directorate General for Mobility and Transport. `http://ec.europa.eu/transport/road_safety/index_en.htm`. [Online; Accessed in January, 2018.].

[Rohani et al., 2013] Rohani, M., Gingras, D., Vigneron, V., and Gruyer, D. (2013). A new decentralized bayesian approach for cooperative vehicle localization based on fusion of gps and inter-vehicle distance measurements. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 473–479.

[Ruj et al., 2011] Ruj, S., Cavenaghi, M. A., Huang, Z., Nayak, A., and Stojmenovic, I. (2011). On data-centric misbehavior detection in vanets. In *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5.

[Saito and Shapiro, 2005] Saito, Y. and Shapiro, M. (2005). Optimistic replication. *ACM Computing Surveys (CSUR)*, 37(1):42–81.

[Santos et al., 2004] Santos, R., Edwards, R., and Edwards, a. (2004). Cluster-based location routing algorithm for inter-vehicle communication. *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, 2:914–918.

[Sarakis et al., 2016] Sarakis, L., Orphanoudakis, T., Leligou, H. C., Voliotis, S., and Voulkidis, A. (2016). Providing entertainment app lications in vanet environments. *IEEE Wireless Communications*, 23(1):30–37.

[Schmidt et al., 2010] Schmidt, R. K., Leinmüller, T., Böddeker, B., and Schäfer, G. (2010). Adapting the Wireless Carrier Sensing for VANETs. *In Proceedings of the 7th International Workshop on Intelligent Transportation (WIT 2010)*.

[Shen et al., 2013] Shen, X., Cheng, X., Zhang, R., Jiao, B., and Yang, Y. (2013). Distributed congestion control approaches for the ieee 802.11p vehicular networks. *IEEE Intelligent Transportation Systems Magazine*, 5(4):50–61.

[Sommer et al., 2010] Sommer, C., Tonguz, O. K., and Dressler, F. (2010). Adaptive beaconing for delay-sensitive and congestion-aware traffic information systems. *2010 IEEE Vehicular Networking Conference, VNC 2010*, pages 1–8.

[Sommer et al., 2011] Sommer, C., Tonguz, O. K., and Dressler, F. (2011). Traffic information systems: Efficient message dissemination via adaptive beaconing. *IEEE Communications Magazine*, 49(5):173–179.

[Song and Lee, 2013] Song, H. and Lee, H. (2013). A survey on how to solve a decentralized congestion control problem for periodic beacon broadcast in vehicular safety communications. *... Communication Technology (ICACT), 2013 ...*, pages 649–654.

[Sou and Lee, 2012] Sou, S. I. and Lee, Y. (2012). SCB: Store-carry-broadcast scheme for message dissemination in sparse VANET. *IEEE Vehicular Technology Conference*.

[Stanica et al., 2012] Stanica, R., Chaput, E., and Beylot, A. L. (2012). Congestion control in CSMA-based vehicular networks: Do not forget the carrier sensing. *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks workshops*, 1:650–658.

[Tal and Muntean, 2013] Tal, I. and Muntean, G. M. (2013). User-oriented fuzzy logic-based clustering scheme for vehicular ad-hoc networks. *IEEE Vehicular Technology Conference*, pages 2–6.

[Ucar et al., 2016] Ucar, S., Ergen, S. C., and Ozkasap, O. (2016). Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination. *IEEE Transactions on Vehicular Technology*, 65(4):2621–2636.

[Villas et al., 2013] Villas, L. A., Boukerche, A., Araujo, R. B., Loureiro, A. A. F., and Ueyama, J. (2013). Network partition-aware geographical data dissemination. In *2013 IEEE International Conference on Communications (ICC)*, pages 1439–1443.

[Vivek et al., 2017] Vivek, N., Sowjanya, P., Sunny, B., and Srikanth, S. V. (2017). Implementation of ieee 1609 wave/dsrc stack in linux. In *2017 IEEE Region 10 Symposium (TENSYMP)*, pages 1–5.

[Wang et al., 2014] Wang, Y., Chuah, M.-c., and Chen, Y. (2014). Incentive based data sharing in delay tolerant mobile networks. *Wireless Communications, IEEE Transactions on*, 13(1):370–381.

[Wisitpongphan and Bai, 2013] Wisitpongphan, N. and Bai, F. (2013). Microscopic experimental evaluation of multi-hop video streaming protocol in vehicular networks. In *2013 IEEE Vehicular Networking Conference*, pages 87–94.

[Wisitpongphan et al., 2007] Wisitpongphan, N., Bai, F. B. F., Mudalige, P., Sadekar, V., and Tonguz, O. (2007). Routing in Sparse Vehicular Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1538–1556.

[Xue-wen et al., 2010] Xue-wen, W., Wei, Y., Shi-ming, S., and Hui-bin, W. (2010). A Transmission Range Adaptive Broadcast Algorithm for Vehicular Ad Hoc

Networks. *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, 1:28–32.

[Yokoyama et al., 2014] Yokoyama, R. S., Kimura, B. Y. L., Jaimes, L. M. S., and Moreira, E. D. S. (2014). A beaconing-based opportunistic service discovery protocol for vehicular networks. In *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pages 498–503.

[Zhang et al., 2011] Zhang, X., Wang, X., Liu, A., Zhang, Q., and Tang, C. (2011). Cooperation enforcement scheme based on reputation for delay tolerant networks. In *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, pages 2372–2376.

[Zoghby et al., 2014] Zoghby, N. E., Cherfaoui, V., and Denoeux, T. (2014). Evidential distributed dynamic map for cooperative perception in vanets. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1421–1426.