# Platform for efficient and secure data collection and exploitation in intelligent vehicular networks

Tarek Bouali

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

# Platform for efficient and secure data collection and exploitation in intelligent vehicular networks

TAREK BOUALI

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

N° | X | X | X

THÈSE présentée par

TAREK BOUALI

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

# Platform for efficient and secure data collection and exploitation in intelligent vehicular networks

Soutenue publiquement le 29 Janvier 2016 devant le Jury composé de :

| | | |
|---|---|---|
| GUY PUJOLLE | Rapporteur | Professeur à l'Université Pierre et Marie Curie, France |
| VÉRONIQUE VÈQUE | Rapporteur | Professeur à l'Université Paris Sud (Paris XI), France |
| MARION BERBINEAU | Examinateur | Directreur de recherche (DR) à l'IFSTTAR, France |
| YACINE GHAMRI-DOUDANE | Examinateur | Professeur à l'Université de La Rochelle, France |
| HOSSAM AFIFI | Examinateur | Professeur à Telecom Sud-Paris, France |
| EL-HASSANE AGLZIM | Coencadrant | Maitre de Conférences à l'Université de Bourgogne, France |
| SIDI-MOHAMMED SENOUCI | Directeur de thèse | Professeur à l'Université de Bourgogne, France |

# DEDICATION

Praise to Allah, the most merciful god, for his guidance and the strength he gave me to come to the end of this thesis.

This Work is dedicated

*To the best mother ever, my mother Mahbouba,*

*To my father EL Tayaa who has sacrificed everything to see me a happy and successful young man,*

*To my uncle Imed, my aunt Samia and her husband Abdallah who have always supported me,*

*To my sister, my lovely sister Salwa,*

*To my brothers, Bassem, Mounir, Ali & Becem,*

*To my fiance, the love of my life Amouna,*

*To my best friends Khalil, Nebras, Ali, Adel, Anis, Abdessami and Oussema who are like brothers that life has given me,*

*To all my friends and people who love me.*

*Tarek*

# ACKNOWLEDGMENT

I owe a great many thanks to a great many people who helped and supported me during the three years of this thesis.

I would like to express my gratitude towards Pr. Sidi-Mohammed SENOUCI, my supervisor, for his kind and continuous support during this thesis, his patience, knowledge, enthusiasm and sympathy. He was like a friend, and for me it was a real pleasure to work with him because he was not only generous with his expertise and precious time during work, but also in the daily life.

I am highly indebted to Dr. El-Hassane AGLZIM for his guidance and supervision without which achieving some aims would not be possible.

Special thanks goes to all DRIVE laboratory team and ISAT employees who greeted me with empathy during the last three years.

Special thanks goes, as well, to the reading committee for their precious time.

# Contents

# 1

# INTRODUCTION

## 1.1/ CONTEXT & MOTIVATION

The great development and evolution of communication technologies together with the variety and potential availability of network access mediums and service providers have led to the appearance of the heterogeneous network concept. This paradigm mainly refers to the seamless and ubiquitous inter-operability between multi-coverage protocols with different access techniques. It offers for mobile nodes featured with Wi-Fi, bluetooth and Long Term Evolution (LTE) the ability to benefit from services and Internet connection anywhere at anytime. Vehicles are not as deprived of this development and car manufacturers are interested in deploying a universal communication module called On-Board Unit (OBU) into vehicles for Intelligent Transportation System (ITS) which is able to support various kind of communication technologies and interface with the embedded sensors. This OBU is, also enhanced with a new technology standardized by the Electrical and Electronics Engineers (IEEE) and dedicated for short range opportunistic communication namely 802.11p/G5 to ease the exchange between high speed vehicles. This development has led to the appearance of the *connected intelligent car* concept with intra- and inter-vehicle communication capabilities. The evolution in the automotive area has also touched vehicles sensing capabilities, thanks to the increasing development and evolution of embedded technologies and integrated sensors and cameras which are being deployed in a massive way internally in new cars engines for diagnostics and externally for potential control and perception of the surroundings. This unprecedented growth in communication techniques together with sensing technologies stimulate the emergence of a large amount of applications and services that aim to enhance driving conditions, increase awareness and bring efficient traffic management (traffic monitoring, congestion detection and control, etc.), road safety (accident detection, collision avoidance, etc.) and infotainment (Gaming, video streaming, messaging, etc.). These applications are mainly based on the massive data collection and exchange between vehicles and deployed infrastructure in the network using specific wireless communication protocols and their analysis using specific techniques to increase reliability and user benefits; two challenging tasks that are difficult to handle, especially, in non reliable environment with specific characteristics (frequent changing topology, vehicles speed and frequent fragmentation) that are highly exposed to security threats. In fact, data generated by embedded sensors and cameras have specific characteristics namely heterogeneity due to the variety of architectures and types of used sensors, redundancy resulting from the notification of the same event from different information sources, incompleteness because of limited visibility of sensors and massive size due to the existence of several data sources. To analyze efficiently these data and extract valuable information that could be exploited by different types of applications and stakeholders (driver, passenger, etc.), it is vital to apply specific mechanisms dedicated to process big data according to application requirements. To exchange these data, used protocols are mainly based on Vehicle-to-

Vehicle (V2V) to establish communication between vehicles and Vehicle-to-Infrastructure (V2I) for an exchange between a vehicle and a fixed infrastructure in the road. However, and due to the big size of the flowing data, it is impossible to exchange them with others at once because V2V and V2I communications are opportunistic, occasional and not long lasting [156]. For this reason, it is always important to reduce the size of exchanged data and choose urgent and representative sets to be exchanged based on predefined techniques. Security, also, remains a weak link in vehicular networks since they are by nature vulnerable to various types of attacks (spoofing, Denial of Service (DoS), etc.). This security problem is due to the lack of infrastructure for authentication and also the fact that all vehicles are equivalent and should play the role of routers in order to exchange data between all parties, which is necessary for a proper functioning of the network. Hence, in the presence of attackers, reliable and trustworthy operations of such networks become impossible without securing data exchange and analysis.

## 1.2/ METHODOLOGY AND CONTRIBUTIONS

In the context presented in the previous section, we aim to benefit from the huge development witnessed by the automotive area and available communication technologies namely their interoperability while targeting to decrease negative effects caused by vehicular network specific characteristics. Therefore, we target to address emerging challenges related to data collection, exploitation and security to bring effective answers to these questions: how to collect and carry efficiently sensed data from vehicles? how to ensure their security? and how to analyze and exploit them via new innovative applications. For this reason, we propose in this thesis to take advantage from the vehicle capabilities to sense and communicate with others to design and develop a platform for secure and efficient data collection & exploitation to provide a reliable and accurate flow of data in a vehicular environment; a flow that could be easily analyzed and exploited at different levels by vehicular applications.

In this thesis, we start our study by taking a closer look at the communication capabilities present in a vehicular network and used to exchange the collected data between vehicles and between a vehicle and infrastructure namely V2V and V2I. However, both V2V and V2I techniques suffer from problems related to connectivity and data security due to the previously described aspects of a vehicular network. Several techniques were proposed in the literature to overcome these problems. Nevertheless, current contributions still require further investigations especially in the security aspect because they assume that all nodes in the network are cooperating, which is not always the case. In addition, some of them neglect the fact of network fragmentation and real traffic information while others rely on an excessive deployment and exploitation of infrastructure, which highly impact their performances. Therefore, we focus in the first part of this thesis on the data collection process in vehicular networks to make it more efficient and secure. We hence propose as a first contribution the use of an optimized number of data harvesters to collect a set of data, that depends on the application, while traveling in a specified area. This centralized solution offers the possibility for a manager to deploy a fleet of vehicles in a rapid manner to get a global overview of a specified region while respecting applications constraint regarding time rigidity. It could be efficiently deployed in urban sensing applications to collect information about free parking lots for example or accidents to decrease passive traffic and congestions. In addition, emergency scenarios, where infrastructure lacks, are among the important use cases of the proposed solution because it helps to collect data about damages, survivors' location, etc. and assess efficient and secure plans for rescue missions. However, data collected by every harvester should be communicated to the manager to be efficiently treated and take actions and counter measures (e.g. rescue missions planification

in a damaged area). For this aim, we propose in our second contribution a secure intersection-based routing protocol (Secure Greedy Traffic Aware Routing Protocol (S-GyTAR)) which takes into account real-time traffic information and controls the cooperativeness degree of vehicles and their behavior in a completely decentralized manner to deliver data from a source to a destination. This protocol relies only on non malicious vehicles to relay collected data to the destination and deploy a position prediction technique to choose the best forwarder among them along a road segment. It also, envisages a store and forward technique to recover from disconnections in sparse networks. But, one security problem still persists, which is the long lasting vulnerability window that may affect running applications and increase losses. Thus, we enhance the protocol with a new Intrusion Detection and Prevention System (IPDS) able to predict attacks and reduce their impact on the network reliability.

In the second part of this thesis, we extend our study to investigate possible ways to exploit, treat and analyze data collected from a vehicular network to provide efficient and reliable applications for users (driver, fleet manager, etc.). Indeed, intelligent transportation applications differ in the way they consider data and require various levels of data analysis and variables Quality of Service (QoS) levels to perform well and enhance driving conditions. Thus, we propose new solutions and services to cope with vehicular network users and application requirements. In our first contribution under the umbrella of data analysis, we develop a new itinerary planning application with the aim of enhancing trip conditions, giving good view about roads topology and preserving money and time. This application offers to its users the most accurate way to plan trips by considering the important factors that may affect their traveling mode. The application is accessible via our secure protocol (S-GyTAR) and performs based on the data collected from vehicles or from the infrastructure. However, it may occur that a user is in a sparse network where no relay could be found and the store and forward technique causes waiting delays. For this reason, we propose our final contribution that consists in a Fuzzy Logic-Based communication medium selection mechanism for QoS preservation in a vehicular network. This spatio-temporal mechanism is based on real-time data collection from the network, application requirements and user preferences, to select the best communication medium among available ones in a seamless and smooth manner to preserve connections continuity of the running applications and protect the QoS applications' requirements.

## 1.3/ ORGANIZATION OF THE THESIS

Except, the introduction and conclusion, this thesis is composed of four major parts which are presented by four chapters in the following manner:

**Chapter 2: State of the art,** reviews the context of our thesis and the related existing solutions. It is composed of three parts. In the first part, we expose different types of data collection protocols in a vehicular network. Second, we present techniques used for data analysis and their exploitation to design useful vehicular applications. Then, we review solutions proposed to enhance the QoS for these applications namely network selection. In the third part, we highlight the different security threats that may be faced in a vehicular network and expose the different techniques proposed in the literature to face them. Finally, we conclude the chapter with a discussion about the different works and motivate our contributions.

**Chapter 3: Optimized deployment of data harvesters for urban sensing,** represents our first contribution in this thesis. In this chapter, we propose a new solution for data gathering in a specified region for urban sensing applications. It proposes to deploy the optimized number of data harvesters that will travel this region and collect a set of data, that depends on the application, and send them to a central third party. Our focus in this contribution is the optimization of data

harvesters deployment, where we, firstly, model the problem mathematically, then solve it.

**Chapter 4: Secure data collection in a vehicular network,** in which we propose, at first, a secure intersection-based routing protocol (S-GyTAR) which considers real-time traffic information in the data exchange process and offers a new mechanism to monitor communicating vehicles, detect possible attacks and evict attackers from the network. In a second step, we propose a new IPDS to monitor vehicles, detect and predict attacks to enhance vehicles security and protect their exchanged data.

**Chapter 5: Efficient data exploitation in a vehicular network,** represents our last area of contributions in this thesis, in which we deal with data exploitation and analysis to firstly, design a new application service to find the most economic itinerary and have a global overview about travel conditions. Afterward, a second contribution is proposed with the aim to enhance applications QoS based on the analysis of the collected data, which consists in a fuzzy logic-based network selection mechanism for quality of service preservation.

# 2

# State of the art

## 2.1/ Introduction

The automotive area is witnessing a tremendous evolution since the creation of the first car in the world with the aim of facilitating the human life and enhancing his comfort. In fact, the number of vehicles all over the world keeps always increasing and has been multiplied thousands of times since the world war reaching more than one billion car traveling in the planet by the end of 2010 [171]. Although, it has handled many problems related to travel duration and possibility to transport big things and led to an economic growth, the potential increase of vehicles number in the world has created new types of problems and challenges which require to be treated. Traffic congestion and jams caused by the concentration of a big number of vehicles in a small area highly impact the behavior of drivers and lead to losses in time and fuel consumption. In addition, the high speed of vehicles together with bad weather conditions may decrease the visibility of drivers and lead to dramatic events such as mortal accidents. Safety reports indicate that worldwide the total number of road traffic deaths remains unacceptably high at 1.24 million per year and between 20 and 50 millions suffer non-fatal injuries [93]. These problems have pushed car manufacturers and research communities to investigate new ways to find solutions and alleviate their impacts. One of the best directions, they went into, is the increase of drivers awareness about their environment to enable them prevent possible problems. For this reason, car manufacturers are till now in a crazy race to instrument their new vehicles with the most recent technologies. Therefore, modern vehicles are being a kind of mobile agents which are able to feel, see and speak thanks to the increasing availability of navigation systems, embedded sensors and newly standardized communication technologies leading to the appearance of a new concept namely Cooperative Intelligent Transport Systems and Services (C-ITS) [99] [84] [100]. In fact, vehicles are being equipped with a set of internal sensors able to collect and inform the driver about their internal functioning such as speed, fuel level, coolant temperature, engine status, etc. and a number of external sensors and cameras which are responsible of the identification of traffic density, weather conditions, etc. Furthermore, they include also a universal communication module called OBU which consists of WiFi, IR, Dedicated Short Range Communication (DSRC), 2G/3G/4G radio, a memory set, a Global Positioning System (GPS) receiver and a processor to communicate with each others and exchange information with infrastructure. Using both V2V and V2I communications, C-ITS will enable cooperation between vehicles and road infrastructure in order to achieve improvements in the areas of safety, mobility, and environment. This cooperation is based on the principle that all parties (vehicles, road side units, etc.) exchange information and make use of them afterward to offer new services (real-time traffic information, improved road safety, etc.). However, the development of this technology creates new challenges and questions: how to collect, treat and exchange these information in such highly dynamic network? and how to secure the communication between all the parties?

In fact, the specific characteristics of the data collected by vehicle's embedded sensors and cameras (heterogeneity, redundancy, incompleteness and massive size) make its analysis to extract valuable information and its exploitation to develop reliable applications very challenging and create new issues that require the application of specific mechanisms able to process such kind of data. Its big size, also, makes its carrying and exchange between all communicating parts via V2V and V2I very hard and sometimes impossible due to the opportunistic, occasional and not long lasting characteristics of these communication capabilities.

Furthermore, the security aspect has been always considered among the hard tasks to deal with especially in wireless networks because opening connections to others expose the data and its holder to various kinds of attacks. Vehicles benefit from an always open connection to ease their exchange of the sensed data which makes them vulnerable by nature to malicious behaviors and easy targets to attackers due to the importance of the data they hold. Hence, it is vital to secure the data exchange and analysis in such non trustworthy networks.

Several projects of high importance were initiated during last years in Europe to deal with problems related to data collection, exploitation and security for Vehicular Ad-hoc Networks (VANETs). Car2Car communication consortium [1] which was created by six European car manufacturers to enhance the road safety and traffic management by exploiting inter-vehicles communications. Its main objectives are: (1) the standardization of V2V communications based on wireless LAN components, (2) the development of road safety application prototypes, and (3) the definition of an exclusive frequency band for Car2Car applications in Europe. NOW (Network-on-Wheels) [2] is a German project joining the efforts of industry and academia to solve key issues related to communication protocols and data security. This project highly cooperates with Car2Car and aims to create communication protocols and security algorithms in vehicular networks. CVIS (Cooperative Vehicle-Infrastructure System) [3] is a European research and development project joining the efforts of sixty partners of manufacturers and academics, and intended to design technologies to enable continuous communications between cars and the car and infrastructure. DRIVE C2X [4] and FOTsis (Field Operational Test on Safe, Intelligent and Sustainable Road Operation) [5] are other co-funded European projects that target to deal with communication, data management and security issues in ITS. Other national projects are also initiated by the french government to enhance and test vehicular communication capabilities such as SCORE@F (Système Coopératif Routier Expérimental @ France) [6] and SCOOP@F (Projet de déploiement pilote de systèmes de transport intelligents coopératifs) [7]. One of the most recent European projects treating problems related to vehicular network communications is CarCoDe [8] (December 2013-December 2015) which aims to create a platform for secure and efficient content delivery between vehicles.

In this chapter, we aim to present current communication techniques and protocols designed for both data collection & analysis and security issues & counter measures to face them in a vehicular network since we aim to deal with these problems in our work and propose new techniques to this aim. Therefore, we firstly introduce data collection concept and review the different architectures and protocols designed in the literature to handle the exchange between the different actors of a vehicular network. Then, we investigate techniques designed for data exploitation and analysis. Finally, we highlight the security issues that threaten vehicular networks and summarize counter

[1] Car2Car communication consortium, https://www.car-2-car.org/index.php?id=5.
[2] NOW (Network on weels), https://dsn.tm.kit.edu/english/projects now-project.php.
[3] http://www.cvisproject.org.
[4] http://ertico.com/projects/drive-c2x.
[5] http://www.fotsis.com.
[6] https://project.inria.fr/scoref.
[7] http://www.developpement-durable.gouv.fr/SCOOP-F-Projet-de-deploiement.html.
[8] ITEA3-CARCODE, https://itea3.org/project/carcode.html.

measures and algorithms designed to face them.

## 2.2/ Data Collection in a vehicular network

A car may be defined, nowadays, as a set of non limited capabilities sensors that are able to detect and analyze a huge variety of data. With the appearance and emergence of on-board units in newly designed cars, collected data are exchanged between vehicles at each encounter while traveling in the road topology using the newly standardized technique IEEE802.11p [161] or cellular capabilities such as Universal Mobile Telecommunications System (UMTS) [16] or LTE [151] offered by the OBU. Several architectures and protocols have been defined to support data collection and exchange between vehicles. However and due to their high mobility and unpredictable frequent change of topology, vehicles are usually delegated to themselves and are self-organized with the help of central third parties. So, a network organization could be either decentralized self-organized without any use of external infrastructure, centralized self-organized where a central third party (e.g Road Side Unit (RSU), eNodeB, Base Station (BS), etc.) is in charge of vehicles' organization management or hybrid where the network management is shared between mobile vehicles and central infrastructures. Data exchange between network members is handled using routing protocols and forwarding strategies based on V2V, V2I and Infrastructure-to-Vehicle (I2V) communications. Various kinds of protocols were proposed in the literature to handle data routing. Many works have surveyed data collection architectures and routing protocols proposed in the VANET field [162, 104, 147, 106]. These protocols are mainly classified by [104] into two categories: (i) topology-based where information about existing links in the network are used to perform packet forwarding and (ii) position-based (or geographic) where neighboring location information are used to forward packets. In the following, we discuss routing techniques belonging to each class.

### 2.2.1/ Topology-based routing

Topology-based routing techniques use global information about the network topology and communication links to handle the data exchange between communicating nodes. In fact, link states and routing tables are exchanged between neighbors to discover and maintain routes between a source and destination. These types of protocols were firstly designed for Mobile Ad-hoc Networks (MANETs) which are characterized by the limited mobility and low speed of their nodes. Vehicular ad hoc networks are a kind of MANETs, but due to their specific characteristics namely the unpredictable frequently changing topology and vehicles speed, protocols developed for MANETs couldn't be directly applied to them. For this reason, various alternatives are slightly modified to fit VANET requirements. Two types of protocols lie under the umbrella of topology-based routing; (i) proactive routing and (ii) reactive routing.

#### 2.2.1.1/ Proactive routing

They are also called table-driven routing techniques. In such kind of routing, routes to destinations are built, maintained and continuously refreshed by periodically distributing routing tables through the network. OLSR (Optimized Link State Routing) was proposed by Toutouh et al. in [146] and is based on the periodic flooding of control information using special nodes acting as Multipoint Relays (MPRs) to maintain fresh routes to destinations. Automatic optimization tools are used to get optimal routes. This protocol is suitable for applications requiring short transmission delays because routes are already established. It is also able to manage multiple addresses of a host and

make it act as a gateway for other nodes in the network. OLSR performs based on three types of messages: (i) HELLO, (ii) Topology Control (TC) and Multiple Interface Declaration (MID). Namboodiri and Gao proposed the Prediction-Based Routing protocol (PBR) [63] where they predict vehicles motion to create new routes before link breakage, preserve vehicles' connections and offer a high packet delivery ratio. This kind of routing requires a high density of deployed gateways and risks disconnection in highways where density is low. DSDV (Destination Sequence Distance Vector) [7] was proposed by Perkins et al. to solve loops problems of routes by the use of sequence numbers. Its data are organized into two tables, namely a routing table and setting time table. The routing table is used to store addresses and sequence numbers of all nodes in the network with the next hop to each one and route metric. The time setting table is used to maintain the time of update advertisements for each destination and the selection of the freshest routes. FSR (Fisheye State Routing) [23] is also a link state protocol which aims at minimizing the routing update in large networks. This is done based on the exchange of link state entries only with immediate neighbors in adapted frequencies based on nodes' distance (entries that are further away are broadcasted in lower frequencies than closer ones). Despite their usefulness in applications requiring short communication delays, proactive routing suffers from routing overheads due to the frequent flooding of the network with link states to update routes. In addition to that, they couldn't be applied to vehicular networks when vehicles are moving at a high speed because the entries in routing tables are valid only for a very short period of time, unless refreshing them at a high rate which is not acceptable for scalability reasons.

#### 2.2.1.2/ Reactive routing

This kind of routing is based on the opening of a route for a node to communicate with another only when needed. It usually contains a route discovery phase where a query packet is broadcast in the network. Therefore, a route is determined hop-by-hop to a destination. Reactive routing is considered to be very efficient regarding packet delivery ratio. In [129], authors developed a reactive content broadcasting protocol in which they deploy FLUTE (File Delivery over Unidirectional Transport) [38] to support multimedia-based information delivery to passing-by vehicles in an urban environment. TORA (Temporally Ordered Routing Algorithm) [15] is based on the building of a direct acyclic graph (DAG) toward the destination. The DAG is built based on a query packet broadcasting where each neighbor replies only if it has a downward link to the destination. RBVT-R (Road Based Vehicular Traffic Reactive protocol) [90] is based on a route discovery mechanism using route request and route reply messages and deploys the store and forward technique to recover from packet loss in case of broken links between nodes. SADV (Static-node assisted Adaptive data Dissemination protocol for Vehicular networks) [57] is based on the use of static nodes deployed into intersections to relay traffic between vehicles and optimize delays. So, they receive packets from moving vehicles in their radio ranges and send them whenever an optimal route to the destination is detected. In this protocol, multipath routing is used to reduce the delay. AODV (Ad hoc On Demand distance Vector) [17] was modified resulting two alternatives namely PRAODV and PRAODVM [37] where the lifetime of an established link is estimated based on the speed and location of the vehicle. PRAODV establishes a new link before the first one breaks down while PRAODVM chooses the shortest path with maximum predicted life time. Several other reactive protocols were also proposed aiming to enhance the delivery ratio and end-to-end communication delay such as AID [111], eMDR [135], MDD [135], HFED [143], NDMR [86] and QoS Aware [134] which combines the use of MPLS and AODV. However and despite its efficiency regarding the delivery ratio of packets, reactive routing is usually affected by the frequently changing topology of the network and high speed of nodes which increase introduced overhead to build and maintain routes.

### 2.2.2/ POSITION-BASED ROUTING

The routing strategy in this kind of protocols is based on information about neighbors' locations instead of link states and routing tables. Therefore, moving vehicles in the network exchange information about their locations with neighbors based on a periodic exchange of short status messages or beacons (also called HELLO messages). Beacons, generally, contain such information as speed, position and vehicle state, from which a cooperative awareness can be constructed. The message format is standardized in the European ITS VANET Protocol (EIVP) [69] Cooperative Awareness Message (CAM) and is defined to be broadcast on a periodic basis. It is of great use in the data forwarding process in position-based routing. Three techniques are identified in such kind of routing:(ii) Dissemination-based routing, (ii) hierarchical routing and (iii) Greedy routing.

#### 2.2.2.1/ DISSEMINATION-BASED ROUTING

It designates data forwarding from a source to a set of destinations or geographic area based on multi-hop communications while minimizing forwarding delays and ensuring good data relaying. Several dissemination protocols were recently proposed for data delivery in vehicular networks [164]. IVG (Inter-vehicle Geocast)[28] was proposed to overcome frequent changing network topologies by introducing dynamic relays to disseminate alerts. It is based on the temporary and periodic definition of a multicast group of vehicles that are in a risk area (accident, obstacle, etc.) to inform them about a detected danger in a highway. MDDV [39] (Mobility-Centric Data Dissemination Algorithm for Vehicular Networks) is a diffusion algorithm which considers vehicles unaware about their neighbors' positions. For this reason, the network is presented by an oriented weighted graph where weights designate distances and traffic density between intersections and the chosen path to relay traffic to destination is the one with minimum sum of weights. In DV-CAST [66], data dissemination is treated in three kinds of VANET densities: dense, regular and sparse. A timer-based technique is proposed to support data forwarding in a dense traffic and a multi-diffusion role-based technique is proposed for data dissemination in a sparse network. A regular network is considered as a mixture of vehicles detecting dense traffic and vehicles detecting sparse traffic and the combination of the two techniques fitting each type are used together. In [40], an opportunistic resource discovery mechanism is proposed where vehicles in the network are exchanging information. The protocol uses a spatio-temporal relevance function to sort resources (parking spaces, traffic information, etc.) and save only relevant ones, thus limiting the distribution of a resource to a bounded area and to the duration for which the resource is of interest. So far, the continuous exchange of information causing overhead could be limited. UMB (Urban Multihop Protocol) [34] benefits from RTB/CTB (same principle as RTS/CTS in CSMA) to choose the farthest node as a packet forwarder. In urban areas, the protocol assumes the existence of special fixed stations called repeaters at every intersection to disseminate traffic in all directions. AMB (Ad hoc Multihop Broadcast protocol) [49] is a distributed version of UMB where the role of repeaters is delegated to vehicles within intersections. In DHVN (Dissemination Protocol for Heterogeneous Cooperative Vehicular Networks) [141], authors exploit vehicles with bigger radio ranges and heights to relay packets to a specific area. The basic idea of this work is that each node receiving a packet arms a timer, inversely proportional to the sum of its height and range, which means that the higher node with bigger radio range will firstly forward the packet. Other vehicles hearing the forwarded packet stop their timers and stop the relaying process. This technique allows a packet to reach the farthest point in the network. A probabilistic dissemination protocol was also proposed in VANET called OAPB (Optimized Adaptive Probabilistic Broadcast) [41] where a packet forwarder is chosen based on a calculated probability in function of distance to the transmitter or neighbor density. Dissemination techniques, in general, enhance the packet delivery ratio and decrease delays, however

they are very challenging regarding the choice of packet relays to avoid network flooding and solve scalability problems despite the ameliorated techniques deployed for this aim.

#### 2.2.2.2/ HIERARCHICAL ROUTING

They are also known by cluster-based routing where vehicles in the network are organized into clusters with a specific node elected to play the role of a cluster head (CH) and manage cluster members. The CH could be either a fixed infrastructure (e.g. eNodeB, RSU, etc.) or a mobile vehicle whereas the size of a cluster is variable (1 to k-hops). The packet exchange between clusters is handled by a vehicle which belongs to two adjacent clusters. Several variants of routing protocols in the literature are cluster-based. LORA_CBR (Location Routing Algorithm with Cluster-Based Flooding)[46] is a cluster-based flooding protocol in which the CH is in charge of destination location searching by broadcasting location requests and replies (LREQ, LREP) like in AODV. So, a vehicle wanting to send data to a destination should request the route from the CH. In [125], a fault tolerant service discovery protocol was proposed where a set of fixed road side routers are clustered to locate a service provider (restaurants, parking lot, hotels, etc.) in a region of interest defined by a service requester (vehicle) in its request. The protocol is mainly based on the grouping of a set of road side routers near service providers into a cluster to enable the diffusion of advertised services and ease their localization and access to moving vehicles. TrafficGather [68] is a completely decentralized mechanism which benefits from the network organization into cluster spaces to gather and enable data exchange between nodes. However, it is not very suitable in a sparse network and engenders an important overhead because it is based on a flooding algorithm. In [112], an efficient clustering algorithm named (Vehicular clustering based on Weighted Clustering Algorithm) VWCA was proposed where the cluster head is chosen based on neighbors number, vehicles direction and trust level to increase the cluster stability. In VWCA, the transmission range of a vehicle is adjusted adaptively based on the number of its neighbors, thus increased when the number is low and decreased when it is high. Therefore, the opportunities to find a packet relay are increased. The protocol was also enhanced with a mechanism for malicious vehicles detection to build and update trust levels. PassCAR (Passive Clustering Aided Routing protocol) [157] organizes vehicles into clusters with one CH and gateways to connect them. In this work, routing is based on route request/reply messages broadcasting where the route selection is a multi-metric based strategy which considers links reliability, stability and sustainability. In COIN (Clustering for Open IVC networks) [29], the cluster head election is based on vehicular dynamics, driver intentions and oscillations of inter-vehicle distances to increase the network stability. MDDC (Multi agent Driven Dynamic Clustering) [138] introduces the use of mobile and static agents to form a moving dynamic cluster between intersections and deliver a rapid response. Vehicle speed, direction, connectivity degree and mobility patterns are considered in this clustering technique. In IPS (Information Propagation Scheme)[44], authors exploit cluster-based message dissemination in an opportunistic manner to route packets between nodes in the network. In [132], [36] [92] [117] and [144], some hybrid architectures were proposed for network organization and data forwarding. They are almost based on the same idea but with different techniques where the network is organized into clusters managed by either an RSU, a base station (BS) or eNodeB and the data is chained from one CH to another in a multi-hop way. Cluster-based routing aims essentially to enhance data delivery to destinations, however it suffers from cluster instability and needs techniques to avoid network inundation. Thus, centralized self-organized architectures require the excessive deployment of central third parties to manage clusters which is not obvious due to their high cost, whereas decentralized organizations are hard to build and maintain because of network mobility.

#### 2.2.2.3/ GREEDY ROUTING

Greedy techniques are based on the principle of choosing the closest vehicle to destination as a packet forwarder. However, in some cases, the packet carrier is the closest one to destination leading to a local maximum problem. For this reason, a store and forward (SNF) (also called carry and forward - CNF) is used to recover from potential disconnections where the packet is carried by a node while moving until finding a neighbor who fits the forwarding conditions. Various protocols were proposed based on greedy forwarding. GPSR (Greedy Perimeter Stateless Routing) [20] was the first greedy protocol developed for VANET which is based on two forwarding methods: greedy forwarding and perimeter forwarding. This protocol exploits locations of 1-hop neighbors to make forwarding decisions. It uses the greedy forwarding to nodes that are always progressively closer to destination until it reaches a local maximum to switch to perimeter technique for recovering. GSR (Geographic Source Routing) [30] was proposed to solve problems related to GPSR and exploits street maps to get the city topology. It uses essentially dijkstra's shortest path algorithm to define the junctions to be traversed by a packet from its source to destination. SAR (Spatially Aware Packet Routing) [31] uses GSR to compute a path to a destination and embeds the given shortest path into the header of all packets. At every hop, the relay node should search the neighbor with shortest path to destination based on GSR in spite of sending packet to the closest node to destination. A-STAR (Anchor-based Street and Traffic Aware Routing) [35] is also based on street maps and waypoints called anchors which are chosen to be inserted into packets' headers. This protocol utilizes statistically related maps to identify city bus routes and dynamically related maps to monitor the latest traffic condition. Both maps are used to find the best anchor paths. GPCR (Greedy Perimeter Coordinator Routing)[45] aims to enhance GPSR performances by exploring a restricted greedy forwarding procedure and a repair strategy. Unlike GPSR, the routing decision in GPCR is made within junctions where a packet is forwarded along a street until reaching a node in a junction to decide about the next one it should be routed through. GVGrid [52] is an on-demand routing protocol designed especially for city topologies. It divides the geographical area into uniform-size squares called grids and constructs routes from a source to destination based on them by assuming that nodes in the same grid or neighbor grids are able to hear from each other. VADD (Vehicle-assisted data delivery) [81] was designed for delay tolerant applications and based on the principle of carry and forward with predictable mobility. The contention-based forwarding algorithm (CBF)[88] utilizes a distributed contention process based on the current positions of all neighbors to select the next hop. To avoid packet duplication, an area-based suppression algorithm is used. DGRP (Directional Greedy Routing Protocol) [73], RIPR (Reliability-Improving Position-based Routing) [121], GPUR (Greedy Perimeter Urban Routing) [120] and GpsrJ+ [61] are among alternatives proposed to enhance the GPSR protocol where the two first take into account directions, velocities and positions of 1-hop neighbors to select a relay node while the third and fourth consider 2-hop neighboring information and greedy perimeter routing to make decisions. In RDGR (Reliable Directional Greedy Routing) [110], the selection of packet forwarders which have progress towards destination is based on the combination of all neighbors information namely: position, speed, direction and link stability. In addition to mobility information, GPGR (Grid-based Predictive Geographical Routing) [130] employs knowledge about the road topology provided by a static street map to better route packets between nodes. In CAR (Connectivity Aware Routing) [64], authors deploy the AODV-based mechanism for route discovery combined with the advanced greedy forwarding technique to relay route replies and data packets. In the route discovery phase, vehicles within intersections define themselves as anchors to relay traffic. Hereafter, the chosen anchors are stacked into packets to make forwarding decisions. Some specific nodes are designated by the protocol as guards used to redirect packets to destination to recover from mobility problems.

Some enhanced techniques for greedy routing have also seen light in the recent years to remedy

from frequent changing traffic concentration problems. Their major contribution is the consideration of real time traffic in the routing process. STAR (Spatial and Traffic Aware Routing) [43] offers this possibility and exploits street topology information given by geographic information systems and information about vehicular traffic in order to perform accurate routing decisions. Traffic information are collected based on beacons exchange between vehicles containing data about neighbors. GyTAR [60] is an intersection-based routing protocol where decisions are made within intersections about the closest one to a destination through which a packet should pass such as in GPCR. However, GyTAR takes into account real-time traffic and a combination between road density and distance to destination to select the best intersection to send packet to. In fact, authors in GyTAR divide the road topology into segments limited by intersections and segments into small fixed subsegments having the size of a vehicle radio range. In these subsegments vehicles are organized into 1-hop clusters where the cluster head computes its neighbors and sends their number to other cluster heads for update when it reaches the end of a segment. This technique allows to have an idea about the distribution of vehicles in a segment and alleviate the network overhead. The way GyTAR is collecting real-time traffic will be the source of our inspiration when proposing our own protocol in chapter 4. RIVER (Reliable Inter-VEhicular Routing) [127] also considers real-time traffic gathering and street topology to make reliable routing decisions.

To recover from problems related to local maximums in the greedy forwarding technique, several mechanisms were proposed by the routing protocols. Among them, we find PBR-DV (Position Based Routing with Distance Vector Recovery) [104] which exploits the technique of flooding requests to neighbors by the packet carrier until reaching a node closer to destination. GRANT (Greedy Routing with Abstract Neighbor Table) [79] proposes the storage of data related to a set of neighbors and selects the best relay node based on a multiplication between the distance of packet carrier and its neighbor and the neighbor and destination. Therefore, the best node to be chosen should have the lowest metric. LOUVRE (Landmark Overlays for Urban Vehicular Routing Environment) [75] introduces the construction of overlays between nodes in the network which are maintained until the network density is less than a threshold. This protocol has a global knowledge about all connected roads density which makes an overlay last for long time. In GeoSVR (Geographic Stateless VANET Routing) [158], authors aim essentially to solve the local maximum problem in sparse networks by proposing two algorithms; optimal forwarding path and restricted forwarding. The first one aims to choose the optimal forwarding path with highest probability to contain more vehicles between a source and a destination while the second targets to choose the next hop. To solve loops problems, LCR [48] proposes to initiate a recursive search on adjacent faces every time a loop is detected while GeoCross [105] uses street maps and records roads and junctions a packet has traveled and unroutable road segments. GeoSpray [167] combines the store-and-forward technique together with geographic location of nodes available via GPS devices to make routing decisions. In ROAMER (ROAdside unit MEssage Routers in VANETs) [142], a new technique to enhance greedy routing is proposed based on RSUs deployment to route packets and increase the forwarding range of vehicles. All the greedy routing protocols proposed for vehicular networks aim to enhance packet forwarding between highly mobile nodes in a frequently changing topology. They propose various mechanisms to recover from potential problems related to local maximum and loops, and thus increase the packet delivery ratio and decrease the end-to-end communication delays. So far, they are being the most investigated techniques in the VANET area especially after the great enhancement of communication, localization and sensing technologies which have highly contributed to their advancement.

### 2.2.3/ SUMMARY AND DISCUSSION

Data collection techniques in a vehicular network with its organization strategies and routing protocols mentioned in section 2.2 relies on a cooperative approach where each node in the network accepts to relay information with the aim of ameliorating security in roads or decreasing travel time and fuel consumption. These protocols aim to limit the number of communicating nodes in order to avoid flooding the network with redundant non useful messages and ease the network monitoring and control. Almost of them rely on the complete building of routes to destinations whereas others only select the next forwarder in an intelligent way.

Various kinds of important and heterogeneous information which are of different priorities and related to different kinds of application are exchanged between vehicles using the reviewed protocols, thus leading to several problems related to quality of service and exploitation complexity in such dynamic network. In the next section, we discuss data exploitation techniques namely data fusion and network selection used for data analysis and quality of service preservation.

## 2.3/ DATA EXPLOITATION IN A VEHICULAR NETWORK

A vehicular environment is a heterogeneous network that encompasses various kinds of access technologies and connected objects. A big amount of data is flowing in such kind of networks due to the increasing number of traveling vehicles, deployed infrastructure and connected sensors. Cooperative Intelligent Transportation System (C-ITS) applications mainly rely on the periodic and continuous collection and analysis of these data from the network. However, their usage differs from application to application regarding the context and time tolerance, leading to complexity in data exploitation techniques. These applications are classified into three types that use data differently [103]; (i) Transportation (traffic) safety, (ii) transportation efficiency, and (iii) user services (infotainment). In fact, data related to traffic safety and efficiency are time-constrained and require a quick delivery, whereas some other kinds of application seems to be more tolerant to delays. However, the delivery delay of data is always highly influenced by the characteristics of the communication technology. For this reason, collected data by embedded sensors in the vehicle should be exploited depending on application requirements before to be stored and exchanged with others using a reliable access medium to meet required QoS and time constraints. Data fusion was among the techniques that appeared to deal with data exploitation and analysis whereas the QoS preservation and application continuity are handled using network selection and handover mechanisms. We provide in the current section details about techniques that are used for data exploitation and QoS management. So, we firstly provide an overview about data analysis techniques namely data fusion. Then, we detail techniques for access management and handover handling.

### 2.3.1/ DATA FUSION IN A VEHICULAR NETWORK

Sensor measurements, images and videos are not of great use if they are treated and analyzed separately, without considering correlation between them and their environment. Resulting information are usually distilled into decisions on actions to be taken on the environment, and historical information regarding these actions can be combined to build links between them and decrease the size of the stored data and messages exchanged over the network. Data fusion appeared to enhance this scenario. It aims to enhance data presentation and understanding, and plan for actions on the environment. In the following, we give an overview about the data fusion concept, then we review some of the fusion techniques proposed in VANET to enhance ITS applications.

#### 2.3.1.1/ DATA FUSION OVERVIEW

The huge deployment of various kinds of sensors and cameras in several domains such as military, automotive, daily human life, smart cities, wildlife habitat monitoring and other fields for surveillance and reconnaissance terms leads to the generation of heterogeneous data coming from different sources, which make the analysis and scene identification tasks more complex for the analyzer. The data fusion concept has been deployed to normalize, combine and exploit heterogeneous data from different sources to get coherent descriptions of scenes and situations that could be reused by automated systems or humans to make enhanced calculations and bring up meaningful decisions. In fact, data measured by electrical or mechanical systems are always exposed to external noise and errors related to the environment, and internal errors introduced by the elementary components of the system (warming, usury, etc.) and imperfect calibrations. Data fusion comes to leverage correlation between different data sources, to minimize noise and solve other problems[154]. Data fusion was firstly introduced by the Joint Directors of Laboratories (JDL) in collaboration with the U.S Department of Defence (DoD) for military aims. They opted for this technique to protect their troops by increasing the accuracy and efficiency of threats and targets identification leveraging multiple deployed resources (cameras, satellite, GPS, etc.) [18]. Their proposal identified the different steps of a fusion process that is still used as a reference model. This model organizes the fusion process into five levels based on the refinement degree of the treated data: a preprocessing phase, then a refinement of objects, situations, threats and processes. It was designed to fit tactical targeting applications and, even when it did not keep the same terminology in other application areas, the way it organizes the data exploitation & analysis process is either the original one or an adaptation of it. The processing degree of the data used in characterization, the type of characterization and the type of characterized entities in the fusion process are the parameters considered to define each fusion level depending on the treated application.

**- Level 0- Preprocessing:** At this level, data collected from the different resources are organized and normalized before their treatment,

**- Level 1- Object Assessment:** At this level, a set of procedures are applied to the data given by the preprocessing phase. These procedures include spatio-temporal alignment, correlation, clustering, false positives removal, identity fusion and combination of features extracted from images. Therefore, the inputs are transformed into consistent data structures,

**- Level 2- Situation Assessment:** It consists on a higher level inference based on the given objects' descriptions resulted from level 1. In fact, objects are combined and relations are identified to determine their significance into a specific environment. So, the main aim of this phase is the identification of activities and events based on the relations built between objects and their environment,

**- Level 3- Impact Assessment:** This level considers the study of potential risks that could be caused by the previously identified activities. Therefore, it is the level of threat evaluation and counter measures planning,

**- Level 4- Process refinement:** It is the resource management process where task scheduling and priority management are being processed to enhance the efficiency of decisions making and reactions.

A more generalized model describing the hierarchy of the data fusion process that could be applied to all application fields was proposed by Dasarthy in [12]. It is a five-level model that was inspired from the JDL model and based its categorization on the presentation of inputs and outputs given at each stage of data processing. It excludes terminologies ambiguity appearing in the first model and gives a clear overview about the fusion degree of gathered data. In the following we describe

the different levels of this model.

**1- Data In-Data Out (DAI-DAO):** It corresponds to the preprocessing phase of the JDL model where incoming data from different sources are treated and refined separately without any combination. At this level, basic functions and algorithms are applied to the gathered data such as signal amplification and redundancy elimination in text data,

**2- Data In-Feature Out (DAI-FEO):** At this level, data with the same type of presentation are analyzed and treated together to extract the feature they could present in the environment. This fusion process is similar to the level 1 of the JDL model,

**3- Feature In-Feature Out (FEI-FEO):** This fusion level consists of the combination and analysis of all features resulted from previous grouped data to give clear and meaningful descriptions of an event or situation in the network,

**4- Feature In-Decision Out (FEI-DEO):** At this level, the given feature descriptions are introduced in a decision making system to clear the situation and plan for possible actions that could be taken,

**5- Decision In-Decision Out (DEI-DEO):** As a final processing for the data, possible decisions given by the previous fusion level are combined and matched together and with other previous stored decisions to result a final action to be taken to face an event or abnormal situation in the environment and react about that.

#### 2.3.1.2/ DATA FUSION IN ITS APPLICATIONS

El Faouzi et al. have demonstrated in [115] the need for data fusion to enhance proposed ITS applications and meet the required efficiency and reliability for passengers safety and reduced transportation time and fuel consumption. In fact, all ITS applications rely on the massive use of sensors and other more or less sophisticated data collection means such as cameras and RMTS (Remote Traffic Microwave Sensor) to manage and control vehicles functions, traffic status and weather conditions. Information given by the different sources should be combined to give an accurate view for application users. In the same work, authors have also identified some kinds of ITS applications that mostly require data fusion processes and studied the applicability challenges of fusion together with the techniques that could be applied. Three types of algorithms that are used to deal with data fusion in the literature, have been identified:

**- Statistical based data fusion techniques:** They are the simplest deployed techniques in the data fusion process but not always suitable especially when data is very complex. Weighted combination and arithmetic means approach are among the most used techniques under this category,

**- Probabilistic based data fusion techniques:** These include the Bayesian approach, maximum likelihood methods, possibility theory and evidential theory,

**- Artificial Intelligence based data fusion techniques:** Neural networks, fuzzy logic, Kalman filter, genetic algorithms and machine learning are widely used in data fusion processing which include some intelligent techniques and rules to exploit and analyze data. They were employed by many works treating safety applications in VANET such as [59] which deploys Support Vector Machine (SVM) for vision-based vehicle detection and classification and [53] that exploits Convolutional Neural Networks for pedestrian detection.

Vehicle diagnostic, traffic engineering, safety and tracking are the most influenced fields by data fusion process where advanced traveler information system (ATIS), automatic incident detection (AID), advanced driver assistance (ADAS), network control, crash analysis and prevention, traffic forecast and monitoring, etc. consider to investigate the data fusion concept in their architectures.

In [56] and [25], the problem of engine diagnostics of a vehicle and fault detection is treated, where the Dempster-Shafer evidence theory is explored to fuse data coming from various sensors that are internally deployed into the vehicle's engine and compared to the estimated normal behavior to detect surging problems in the engine.

Advanced traveler information system is one of the ITS applications that offers users integrated traveler information. Various kinds of information sources and techniques to exchange collected data are used to manage traffic conditions and derive relevant indicators to assist drivers and enhance its awareness about driving conditions. This system always considers the travel time as the metric reflecting routes status, thus detecting potential congestions. A plethora of frameworks dealing with the travel time estimation under ATIS were proposed. ADVANCE [6] [5] [8] is an in-vehicle ATIS providing route guidance in real time that operates in the northwestern portion and northwest suburbs of Chicago. This system considers probe vehicles to generate dynamic travel information about considered routes. It combines data from loop detectors and travel time reports of probe vehicles using a three-stage fusion technique; the first fusion algorithm deals with probe reports sparsity based on regression methods, the second algorithm considers to treat received probes using Bayesian methods, and the final stage is based on inference rules to analyze supersaturated conditions. In [22] and [89] a fusion based mechanism is proposed by El Faouzi et al. to accurately estimate the travel time. The Dempster-Shafer evidence theory is deployed to fuse data incoming from different sources due to its ability to solve problems related to imprecision and uncertainty of data. In the first work, data from conventional induction loop sensors (essentially flow and occupancy measurements) and data from probe vehicles are considered whereas in the second one fused data is gathered from inductive loop road sensors and toll collection stations. In these proposals, the time estimate is considered as a classification problem where times are broken into classes and treated then combined into a one confusion matrix. A recent architecture [137] for vehicles localization to enhance driving and congestion detection is proposed using a dissemination algorithm and data fusion of various information collected from sensors. Information gathered from vehicle's local sensors and received from neighbors (vehicles or RSU) are combined in a probabilistic manner using a Bayesian filtering model implemented based on an extended Kalman filter and particle filter (PF). The fusion algorithms result an estimate about the future position of a vehicle, thus predicting its trajectories to avoid congested roads.

Automatic incident detection is among the most challenging issues encountered in vehicular networks due to its impact on drivers and passengers safety. It has also snatched interest of researchers after the appearance and proliferation of new embedded sensors able to monitor the vehicle's surroundings and report abnormal events. This kind of ITS application has also benefitted from data fusion to accurately ensure safety [55]. Dempster-Shafer evidence, Bayesian inference and voting logic are the most used in the data fusion process. In [10], an automatic incident detection system is proposed based on real time data collection and combination from special equipped probe vehicles and inductive loop detectors deployed in streets. Neural network algorithms were exploited in the data fusion process. Two fusion types were proposed in this work; the first is based on the direct combination of information from all sources to produce a single decision about the presence or absence of incidents on each link, whereas in the second, separate incident-detection algorithms individually preprocess data from each source, reporting outputs that are combined using a neural network. Several variants of neural network representations are studied in [13]. The applicability of Dempster-Shafer inference in traffic management to support incident detection is studied in [21]. This technique is used to eliminate data uncertainty and false alerts about events declared by deployed sources. So, knowledge from different sources are combined and treated using Dempster rules to verify the exactness of an incident. In [11], the automatic incident detection is treated as a multiple attributes decision making (MADM) optimization problem with Bayesian

scores. The authors propose an approach which utilizes the combination of probe travel times, number of probe reports, detector occupancies and volume as attributes of their MADM algorithm. In [82], a probabilistic automatic incident detection technique under sparse and moderate traffic flows is proposed based on Bayesian networks. The proposed technique aims to efficiently exploit vehicle communication capabilities to enhance incident detection in non dense networks while considering lane change as the consequence of an incident. In this work vehicles are assumed to have GPS and local event data recorder used to store records about lane changes.

Data fusion was also deployed for driver assistance, network monitoring and post crash alerting systems with the aim of enhancing ITS safety applications and reduce traveling time. A cooperative architecture based on vehicles information data fusion is proposed in [160]. A multi-level fusion method is used; a low-level fuzzy clustering-based algorithm that combines atomic messages and abstract key attributes (features level fusion), and a high-level modified BPA-based (basic probability assignment) evidence theory used to avoid misjudgment caused by short-term wait events (e.g. traffic lights) and enhance the accuracy of the final decision (decisions level fusion). In the low-level fusion, messages containing basic information such as speed, location, generation time, etc. received from neighbors during a time period are aggregated and clustered based on fuzzy-clustering rules regarding features similarities where an abnormal event is detected. Decisions given by the fuzzy-clustering rules are enhanced based on probabilistic aggregation in the high-level fusion module. The proposed architecture in this work mainly aims to detect congestion and abnormal events in a road to assist drivers. In [152], a high level information fusion architecture through a fuzzy extension to multi-entity Bayesian networks is proposed to enhance drivers awareness about crushes, thus avoiding collisions. The architecture considers semantic data extraction to build causal relationships between the existing entities in a specific context. For this reason, web ontology language is used to model contexts and Bayesian networks with the fuzzy rules are used to manage data uncertainty and ambiguity respectively. The basic idea behind this high level information fusion is the combination of all actors that could influence a decision making and model their relations before making a decision and view it to a driver. In a recent study [123], human factors are taken into consideration while classifying safety messages. Based on the fact that a driver may not be able to receive more than one service in a short period of time and would be unable to deal with a new service until he has finished responding or reacting to the current one, authors propose a multilevel fusion architecture (low and high levels) to not bombard the driver with alerting messages. The low level data fusion aims to check the correctness of incoming data and reduce the congestion in the wireless medium, while the high level is responsible for the application of human factors to choose the messages to view to drivers from the queue. Received warning messages are modeled by utility functions associated with their location, generation time and modality which are later combined to get the global utility function, whereas the familiarity of the driver with a message is used to introduce human factors. The combination of the two models results the most critical message to be firstly viewed to the driver to be treated.

### 2.3.2/ NETWORK SELECTION IN A VEHICULAR NETWORK

Intelligent transportation systems, in general, rely on radio services for communications and use specialized technologies (Wireless, DSRC, Tetrapol, 3G, etc.). However, their supported applications are highly influenced by the quality of service degradation in the used communication technology. The QoS may depend on various conditions mainly related to the network status and vehicles motion. In fact, the speed of vehicles and the vehicular network organization characterized by its fragmentation (dense with congestion or sparse with no relays) and frequent topology changes make preserving high QoS or even access to user applications a very challenging issue in such

kind of networks. In the early years, with the appearance of new standards in the world (European Telecommunications Standard Institute (ETSI)-ITS, ISO/TC204, etc.) for ITS communications, it is now possible to have various inter-operable access technologies in the same communication stack of one node. The Communication Access for Land Mobiles (CALM) standard [96] which was established by the International Organization for Standardization (ISO) in 1993 covering activities in ITS, represents the result of a worldwide standardization effort including Europe, Asia and United states and offers the possibility of co-existence between several variants of communication technologies (WiFi, 3G/4G, DSRC, etc.). However, the inter-switch between these technologies, which is also known in the literature by vertical handover (VHO), still represents an open area for research because it has to be done in a smooth and seamless manner to preserve running applications from disruption, thus protecting user data and ensuring service continuity. CALM proposed an open architecture to manage switching between available networks, however it does not define any specific algorithms or techniques to do the job. In the following, we begin with a short overview about the CALM standard and its architecture to manage available access technologies, then we review some of the proposed techniques in the literature dealing with the vertical handover.

#### 2.3.2.1/ CALM STANDARD OVERVIEW

CALM standard is a general communication system that includes all types of ITS communications and normalized related protocols, which was proposed by ISO/TC204 WG16. The standard mainly aims to enable a software platform that incorporates multiple communication mediums (2G/3G/4G, UMTS, DSRC, etc.) and manages the vertical handover between them to allow connection continuity and benefit from the best available characteristics (use of the best medium). CALM is a layered architecture derived from the OSI model which provides so called cross-layer functionalities to manage inter-layers exchange. It comprises an application layer that contains all ITS applications, the facilities playing the role of OSI layers 5, 6 and 7, a networking and transport layer representing OSI layers 3 and 4 and an access layer which encompasses functionalities of OSI layers 1 and 2 and manages access networks. It is, also characterized by two entities to manage and secure the communication protocol stack namely the management and security entities. The management entity is in charge of the configuration of an ITS station, cross-layer information exchange among the different layers and other beneficial tasks. The security entity provides security and privacy services including secure messages at different layers of the communication stack, identities management and security credentials, and aspect for secure platforms (firewall, security gateway, temper-proof hardware). The global architecture of an ITS node as described by the OSI CALM standard is provided by figure 2.1.

The inter-operability and handover between the different available communication interfaces (CI) is managed in the access layer with the consideration of information gathered from other layers via the management entity. This task is performed based on a specific architecture able to collect data from different layers and exploit them to find the best interface. The CALM CI management architecture, described by Fig.2.2, includes six components: (i) user application requirement list defining the specific parameters needed by applications, (ii) Man Machine Interface (MMI) used to specify user preferences, (iii) a set of rules to treat the different inputs and weight parameters, (iv) CI/VCI status list providing real time performance parameters from all available communication interfaces, (v) CI selection module to weight communication interfaces and give a final decision about the best one and (vi) network attributes determined by the designated access network and generic service characteristics.

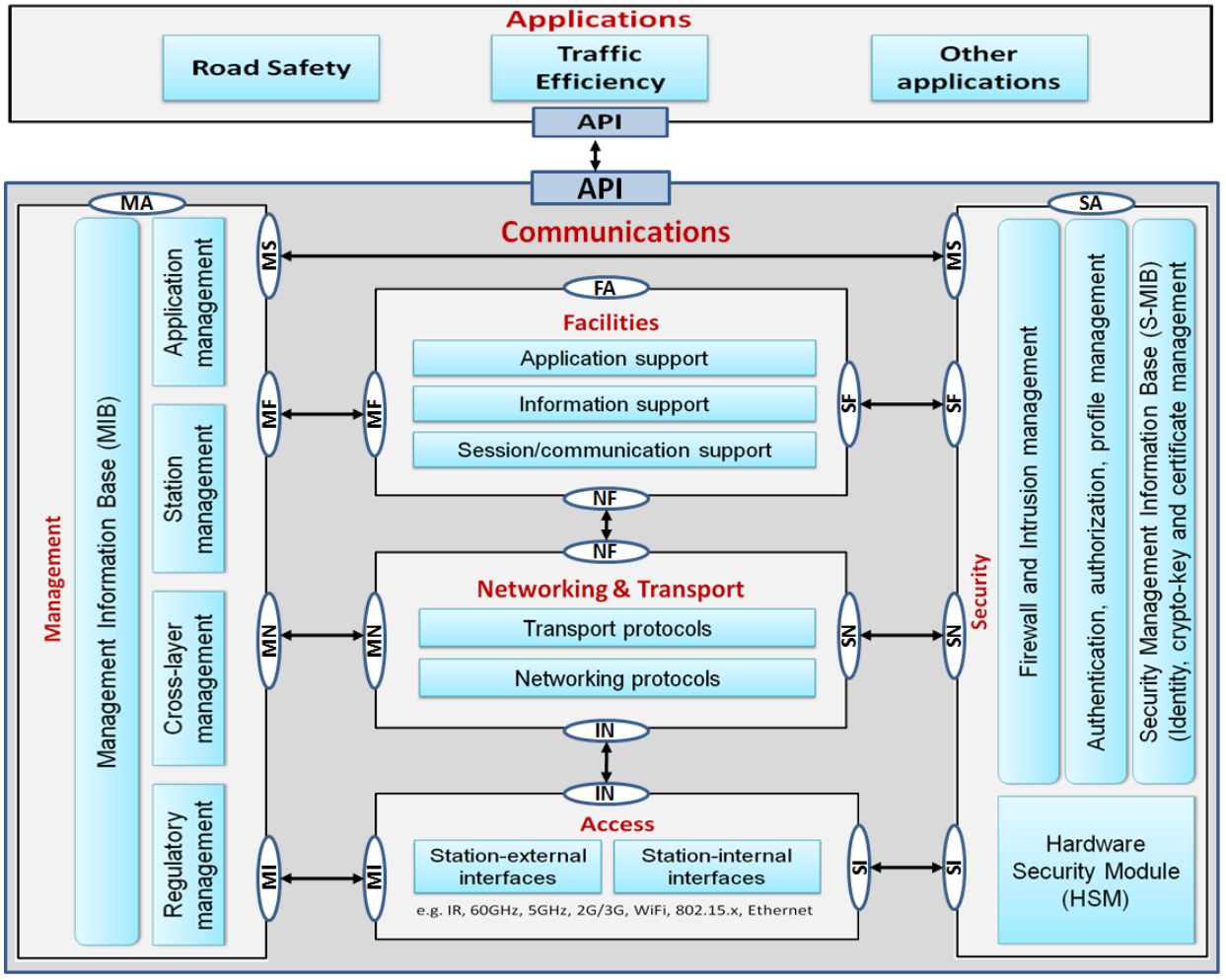



Figure 2.1 – ISO CLAM architecture

#### 2.3.2.2/ NETWORK SELECTION TECHNIQUES

Vertical handover (network selection), unlike horizontal handover, depends on various criteria related to networks, applications and users such as service cost, mobile node speed and quality of service. In the literature, and unlike VANETs, mobile wireless sensor networks have witnessed the development of a plethora of mechanisms dealing with vertical handover. They could be deployed in vehicular environments, however with degradations in performances. The received signal strength intensity (RSSI) at a used medium was considered in [67] to decide about handover initiation for a video streaming application. In their proposal, authors consider to predict the signal strength of a medium by making excessive measurements and decide whether to change to another access network or not based on its variation. A mechanism to prevent from packet losses during handover is considered by adding buffers in the point of attachment (base station, access point) and mobile client. This mechanism only relies on one metric (RSSI), tested on one application type and is unable to support real time applications. More enhanced proposals which consider other potential parameters that may influence the decision accuracy about the necessity of handover are also proposed in MANETs [168] [148] [133] [169]. These techniques mostly rely on a set of advanced algorithms such as fuzzy logic and the modeling into optimization problems with multiple objectives and multiple attributes (Multiple Attribute Decision Making-MADM [94]). In [168] and [148], more

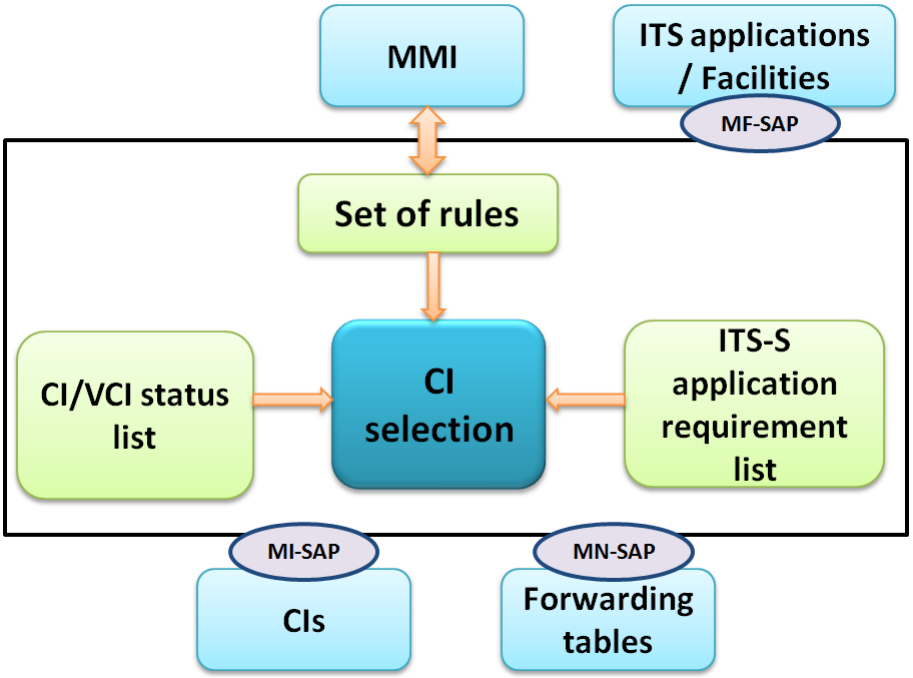



Figure 2.2 – CALM CI management architecture

parameters were considered in the network selection decision process, which are network load and node speed in addition to the received signal strength. These parameters are considered to be fuzzy and presented in a linguistic format. For this reason, a fuzzy logic-based mechanism was proposed to treat them using a set of internal fuzzy rules to give a decision about the best network candidate to be used. The most popular classical methods used to solve a MADM problem are: (i) SAW (Simple Additive Weighting) where the total score of a network candidate is determined by the weighted sum of attribute values, (ii) TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) where the network is chosen based on its closeness to an ideal solution, (iii) AHP (Analytic Hierarchy Process) which decomposes the problem into sub-problems with assigned weights and (iv) GRA (Grey Relational Analysis) ranks candidates and selects the one with the highest rank [72]. In [133], authors propose to use ANP (Analytic Network Process) which is a multi-criteria decision making (MCDM) tool derived from AHP to solve the handover decision problem. The proposed solution is composed of two main components namely the context repository which is responsible for the gathering of information relative to network and terminal characteristics, user preferences and service requirements, and the adaptability manager responsible of handover decision and initiation. ANP is deployed in the adaptability manager to select the best network among existing candidates based on information given by the context repository module. In [169], authors consider to use the SAW algorithm in function of latency, signal-to-noise, power and throughput criteria. The handover scheme is composed of three modules which are:(i) Handoff Need (HN) module used to predict the necessity of handover based on received signal strength (RSSI), (ii) Target Network Selection (TNS) which is done based on the defined criteria such as throughput, signal-to-noise ratio and latency and (iii) Handover Performance Parameter Estimation (HPPE) which is used to estimate and optimize network parameters.

The vehicular network field has also seen the proposition of some works to handle network selection. In [91], authors propose a simple mechanism for vertical handover called VANET Backup Communications (VANBA) in layer 3 of the OSI stack. They consider the V2V medium as the default communication interface and initiate handover to secondary mediums (e.g. 3G, WIMAX, etc.)

only when V2V is inactive (unreachable destination). VANBA collects information registered by applications, monitors the destination IP address and whenever it is not reachable via V2V it triggers the handover to a new medium and returns back to the default one once it becomes reachable again. Although, it is simple to implement and use, VANBA doesn't consider the use of secondary mediums for communication until the QoS of running applications degrades which may cause 'ping-pong' problems. In [140], MACHU (Multi-ACcess network Handover Algorithm for vehicUlar environments) is proposed to select the best point of attachment (e.g base station, access point). The proposed mechanism is based on three main phases: (i) handover information gathering in charge of the network detection and user and node parameters collection, (ii) handover decision to evaluate the necessity of handover based on collected information and decide what the best network is, and (iii) handover execution phase. Authors exploit the Media Independent Handover Function (MIHF) protocol, defined by the IEEE 802.21 standard [70] to exchange information between different layers. The decision algorithm benefits from services offered by IEEE 802.21 standard and GPS to retrieve information about networks and neighborhood to select the worthwhile point of attachment. This work is only based on functionalities of one technology and consider to have very accurate information from GPS which is not always the case. In [98], a handover mechanism based on vehicle's speed is proposed. The speed is used to estimate the throughput and crossing time (time to cross the network coverage area) deployed in the decision process. If the throughput given by the new network exceeds the old one, the crossing time is calculated and handover is executed when its value is higher or equal than a threshold. This work supposes that the vehicle travels in a constant speed which is not the case and does not consider application requirements and network characteristics.

### 2.3.3/ SUMMARY AND DISCUSSION

Data collected from a vehicular network has several use cases and application fields. During subsection 2.3.1, we reviewed some important fields where these data are deployed based on the data fusion concept. Therefore, data fusion seems to be a promising technique able to deal with heterogeneous massive data structures flowing in a vehicular network with the aim of enhancing ITS applications and providing more reliable, efficient and reduced meaningful sets of information to users. Several variants of data fusion algorithms are widely deployed by researchers which are either embedded in the onboard system of vehicles or in fixed management centers targeting to offer more comfortable and safer driving conditions by enhancing driver awareness about its vehicle and environment. However, the use of the exchanged data between vehicles does not stop on these applications, and could be exploited in a different way. Network selection or vertical handover also exploits the collected data to preserve higher quality of service for running applications. It is a necessary task in vehicular environments to compensate vehicles speed and network topology change. However, decision making highly depends on several parameters. The techniques and standard reviewed in subsection 2.3.2 target to offer a smooth and seamless handover without impacting application performances by considering the most important factors that impact decisions. To do this, fuzzy logic and MADM were considered, where fuzzy logic seems to be more lightweight without needs for special processing capacities.

Although, they target to offer more reliable applications and enhance the experienced quality of service by clients in a highly dynamic network such as VANET, the data exploitation techniques mentioned in this section do not interest to security issues that may harm the network and disrupt running applications. In the next section, we review possible security threats in vehicular networks and discuss techniques used to face them and protect applications.

## 2.4/ SECURITY IN A VEHICULAR NETWORK

Collected data in a vehicular network are of high importance and should be communicated efficiently between vehicles or to a central server in the infrastructure. As a carrier of such relevant data, a vehicle is always attracting the attention and may be exposed to potential attacks [65]. These attacks may affect data properties such as consistency, integrity and availability [145], which lead to a perturbation of the normal functioning of applications offered by VANET. VANET attacks could be of various categories and initiated either by vehicles external to the network (non authenticated by a central authority) which are, also, known as external attacks or by vehicles belonging to the network and possessing a legal certificate (internal attacks) using different strategies.

The appearance and standardization of the new communication technology called IEEE802.11p (WAVE) enhances the ability of vehicles to interact and communicate together easily and compensates problems of the high speed and short lasting communication opportunities between them and infrastructure. In fact the standard IEEE802.11p defines a new way to exchange data between vehicles as soon as possible by getting rid of the authentication phase deployed in the majority of other technologies. The security aspect of short range communications between vehicles using 802.11p is managed by another standard namely the IEEE1609.2 which was jointly developed and handles the anonymity, authenticity and confidentiality of exchanged messages and running applications. This standard is mainly based on messages authentication and certificate distribution to secure communications. The exchanged messages are authenticated based on cryptographic mechanisms using digital signature and encryption. However and although it carries a legal certificate, an authenticated node can threaten the network applications security and initiate attacks with respect to used protocol rules. For this reason, supplementary techniques were proposed to enhance the security aspect and plug left holes. They are of tow types; the continuous control of distributed certificates [62] and Intrusion Detection System (IDS) [128].

This section is divided into four main subsections. We devote the first one to review the different attacks that may threaten a vehicular network. In the second subsection, we give a short overview about the standard IEEE1609.2. We describe the security mechanisms proposed to handle malicious threats in vehicular networks in the two last subsections.

### 2.4.1/ SECURITY ATTACKS IN VANET

Vehicular networks are exposed to several kinds of attacks. The objective of these attacks is to create problems for the network users by changing the content of messages and disrupting their applications and communication continuity. They are classified by [122] and [150] into five classes: network, application, timing, social and monitoring attacks. In the following, we describe each class.

#### 2.4.1.1/ NETWORK ATTACK

In this class, attackers directly affect vehicles and infrastructure in the network. Their main objective is to create problems for legitimate users of the network applications. Attackers may be misbehaving and faulty nodes or malicious users. Some types of attacks residing under the umbrella of the network attack class are described as follows:

**- Denial of Service attack (DoS):** This type is one of the most serious levels of attacks in vehicular networks because it threatens the network availability. It consists in jamming the main communication medium to prevent authentic users from accessing network services,

**- Distributed Denial of Service attack (DDoS):** This attack consists of initiating a DoS from different locations to down the network (V2V and V2I communications). Attackers may initiate their attacks in different time slots and jam the network with their messages,

**- Sybil attack:** A Sybil attacker sends messages to vehicles in the network pretending the existence of other vehicles on the road by fabricating some fake identities,

**- Node impersonation attack:** Each vehicle has a unique identifier to verify a message whenever an accident happens. However, in this kind of attack a vehicle may change its identity to decline its implication in the accident to police agents.

#### 2.4.1.2/ APPLICATION ATTACK

Safety and non safety are two types of potential applications in vehicular networks. Application attacks are very dangerous to these types of application, especially, safety ones because they aim to alter the message content or send a false one in order to use it for malicious aims. They consist in packet alteration and false alerts sending to force other vehicles changing their directions and cause congestion or accidents in some predefined roads.

#### 2.4.1.3/ TIMING ATTACK

The timing attack consists in adding time slots and creating delay in original messages. So, attackers do not change the whole content of the message but they will be received by other nodes late or after being deprecated with no benefits. Therefore time-sensitive applications such as safety ones will loose their main objectives when a delay occurs, thus causing perturbation in the network organization and traffic conditions and threatening the safety of drivers and passengers.

#### 2.4.1.4/ SOCIAL ATTACK

This kind of attack aims to provoke legitimate users in the network and make them change their behavior to anger, which may affect the driving behaviors causing some accidents. It consists in sending unmoral messages (e.g. *Hello, You are Idiot*) by a node to its neighbor to cause conflicts between drivers. The impact of this attack on the network is indirect.

#### 2.4.1.5/ MONITORING ATTACK

Monitoring and vehicles tracking attacks reside under this class. In a monitoring attack, attackers monitor the whole network and listen to communications between vehicles and between them and infrastructure (V2V and V2I) and whenever they detect any important information they send them to the concerned person. We can give the example of police agents planning to perform some operations against a criminal and communicate with each other the location of the operation. An attacker spies their communications and communicates the information to the criminal. In the tracking attack, an attacker uses the unique identifier of a vehicle to track its locations and enable external third parties to send viruses to its neighbors and get their collected data.

### 2.4.2/ IEEE1609.2 STANDARD OVERVIEW

The Institute of Electrical and Electronics Engineers 1609.2 standard was created to manage and specify a set of security services for supporting vehicular communications. It defines a subsystem architecture for security management between communicating entities which offers various security services that could be efficiently used by different consumers (WAVE services' provider and WAVE management entities). These security services mainly rely on Elliptic Curve Cryptography (ECC), public key certificates and the Public Key Infrastructure (PKI). So, each entity, such as a vehicle or a Certificate Authority (CA) server should have IEEE1609.2 certificates and a certificate management entity (CME) to manage these certificates. In fact, a CME is composed of a set of functions for: (i) requesting certificates and Certificate Revocation Lists (CRLs) from the CA (further details about CRL are in subsection 2.4.3), (ii) processing received certificates, CRLs and other security management messages, and (iii) managing security-related information. In addition to CME, IEEE1609.2 standard provides cryptographic services such as creating digital signatures, verifying signatures, and decrypting messages. Therefore a security consumer has to request a certificate before to be able to send signed and encrypted messages. Once it obtains a certificate, the consumer can prepare signed and encrypted messages based on security services of the IEEE1609.2 standard. Figure 2.3 highlights the functional architecture of the security management system of IEEE1609.2. This latter defines an hierarchy that organizes roles between the different stakeholders of a vehicular network and classifies entities providing or using IEEE1609.2 security services into two categories; certificate authority (CA) and end entities. The CA is responsible of the certificates and CRLs generation whereas end entities are the actors that use them and they include vehicles, roadside units (RSUs) and application servers.

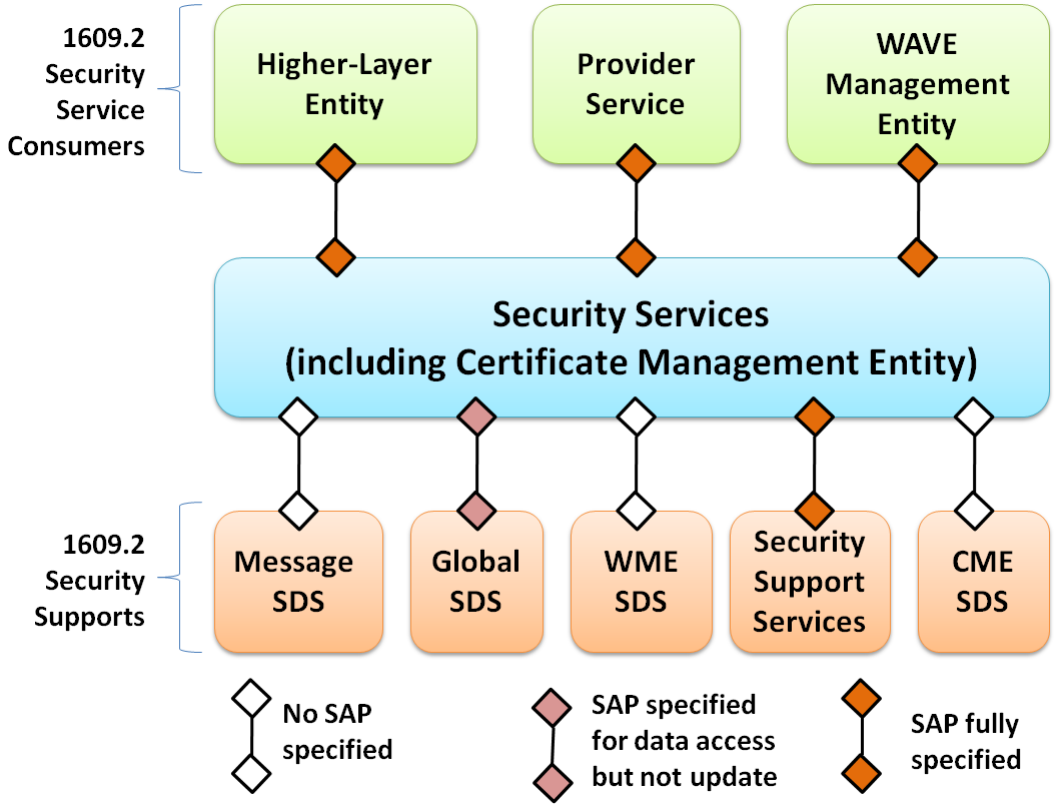



Figure 2.3 – The functional architecture of the standard 1609.2 [153]

The communication between end entities and a CA requires, always, a mutual authentication. This mutual authentication is achieved using two types of security management certificates: (i) a certificate signing request (CSR) used by the end entity to authenticate to the CA and (ii) a CA certificate used by the CA to authenticate to the end entity. A certificate usually contains at least one public key and a set of permissions associated with that public key. The CA keeps controlling its generated certificates and whenever it detects a malicious behavior from their owners they are revoked and a so called CRL is used to inform network members about that. The CRL structure and informing techniques are further detailed in the next subsection.

### 2.4.3/ CERTIFICATE REVOCATION

Beside the cryptography techniques and security structures such as public key, signature and certificate, IEEE1609.2 offers another data structure that enhances communication security namely Certificate Revocation List (CRL). A CRL is a kind of authenticated message distributed by the certificate authority (CA) containing a list of revoked certificates which are previously issued and are no more valid. Revoked certificates contained in a CRL are verified using the certificate management entity (CME) presented in figure 2.3. So, whenever a CRL is received by a CA or a vehicle it should be verified to confirm if it is validly signed. Then, certificates in the CME Security Data Store (SDS) should be checked to see whether they are still trustworthy or not. However, despite their ability to notify legitimate users in the network about compromised and deprecated certificates, the distribution of CRLs in a vehicular network at time still represent a very hard task due to nodes speed and frequently changing topology. Therefore, to protect vehicles from misbehaving nodes the certificates of compromised attackers should be distributed to users as soon as possible. However, when a vehicular network is concerned, this task is not obvious and requires special techniques and architectures to deal with vehicles speed and density. Some techniques were proposed in vehicular networks to detect compromised certificates and distribute CRLs in an efficient way to alert legitimate users and preserve communications from illegal behaviors and attacks. These techniques are either based on the excessive use of RSUs and V2I communications or the exploitation of V2V capabilities.

The use of road side units deployed in the vehicular network topology was among the first investigated techniques to distribute CRLs, where vehicle to infrastructure communication (V2IC) capabilities were exploited. [77] was among the first mechanisms developed based on this method considering the dissemination of a CRL across the entire region governed by a CA without degrading vehicular communications for other more critical operations such as safety. Based on the fact that the size of a CRL is proportional to the number of authenticated vehicles in the network and keeps increasing with their density, authors consider to only maintain a CRL that corresponds to a specific regional CA to keep CRLs small in size. This was motivated by the hierarchical architecture of the certificate authority (CA). To fit the aims of their proposed CRL distribution mechanism, authors make the following assumptions: (i) the deployment of RSUs is by nature done by the same organizations that instantiate CAs, which offers the possibility to a CA to reliably leverage on them for CRLs distribution, and (ii) the communication between RSUs is not required. The CRL distribution technique is then developed based on three main building blocks, where each one deals with a specific problem: (i) collaboration between CAs to manage vehicles traveling across geographical boundaries of one CA, (ii) segmentation of CRLs into small pieces, and (iii) the use of a low rate broadcast of CRL pieces by RSUs. However, despite their ability to reduce the distribution time and preserve time-critical applications in an urban area, such kind of mechanisms would fail to reach all nodes in a large network. They only rely on V2I capabilities to distribute CRLs in a very large network, so a sparse infrastructure distribution like in rural areas and highways highly decreases

the CRL distribution capability. Therefore, vehicles may rarely encounter an RSU, and wouldn't be able to collect all required pieces to reconstruct the CRL, in this case they will be harmed by compromised nodes because they are not up to date regarding revoked certificates. Nevertheless, if we consider to deploy a large number of RSUs in all roads topology, this would lead to high costs and problems of cross authentication between CAs.

Some other works considered to jointly exploit V2V and V2I communications to distribute CRLs to vehicles in the network. In [74], Laberteaux et al. proposed an epidemic distribution of CRL updates based on their exchange between vehicles at each encounter. In their proposal, RSUs are considered as first relays for CRL updates, then vehicles rebroadcast them to their neighbors. In [85] and [114], Haas et al. proposed a new mechanism for efficient exchange of a reduced CRL size together with an efficient technique for determining if a certificate is in a CRL or not. The proposal also relies on V2V and V2I communications to distribute CRLs in the network. It aims to optimize the organization, storage and exchange of CRL information based on Bloom filter and the broadcast of a limited size of CRLs. These techniques are able to achieve a high rate of vehicles reachability compared to RSU-based techniques in case of sparse networks without infrastructure, but they still suffer from large vulnerability windows and high generated overhead. In [109][108], Nowatkowski et al. target to limit overhead problems caused by the two previous techniques using a so called Most Pieces Broadcast (MPB) technique by limiting the number of broadcasting vehicles to only the ones who have the biggest number of CRL pieces. Two possible communication scenarios may be seen in MPB; the first is an RSU-based CRL distribution where only the RSU will be chosen to disseminate because it has the complete pieces of a CRL, this appears always when vehicles are moving within an RSU radio range, the second case appears when no RSU is available where the vehicle with the biggest number of CRL pieces will be chosen to disseminate in a V2V manner. To choose the broadcaster, authors rely on DSRC (Dedicated Short Range Communication) architecture and its channels definition. In fact, DSRC defines two kind of channels; (i) CCH (control channel), which is a single channel reserved for short, high-priority application and system control messages, and (ii) SCH (service channel), which has six different 10MHz channels that support a wider range of applications and data transfer. Nodes in the network exchange beacon information during CCH interval, a message that MPB authors reshape to introduce more information about CRL pieces count that the sender has locally. After the exchange of the pieces count during CCH interval, the node with the highest number of pieces is chosen to broadcast its pieces during the SCH interval.

All these techniques are able to detect attacks initiated by non authenticated vehicles and may succeed to detect authenticated attackers, a very difficult task in the absence of an omni-present monitoring facility. However, they take much time (vulnerability window) to inform legitimate users about compromised certificates which increase attack duration and losses [76]. They, also, require the existence of a minimum number of deployed infrastructure to perform well. For this reason, other alternatives were investigated to enhance security in vehicular networks. Among these, intrusion detection systems (IDS) are the most investigated techniques and will be the focus of the next subsection.

### 2.4.4/ INTRUSION DETECTION SYSTEMS

The vehicular network, as previously discussed, is a very active area which is by nature exposed to malfunctions due to hardware errors and also vulnerable to attacks caused by malicious nodes. We have presented till now techniques based on certificate revocation list and demonstrated their major inconvenient in the previous subsection. Now, we are going to review new techniques that enhance vehicular network security based on intrusion detection systems. They were among the

most investigated methodologies dealing with the security aspect in the literature where dozens of works were proposed. In such kind of techniques, vehicles are delegated to themselves to manage their proper security and inform each others about potential attacks based on mutual trust relationships built between them either using past interaction experiences or information collected from the environment about data validity in received messages. So, vehicles traveling on the network analyze neighbors behavior and decide about their trustworthiness by themselves and build so-called trust levels to be associated with each one. Three variants of trust models, through which intrusion detection systems are built, could be identified: entity-oriented, data-oriented and hybrid modeling [124]. In the following, we review some of the works proposed under each kind of trust model.

#### 2.4.4.1/ ENTITY-ORIENTED TRUST MODEL

Entity-oriented trust modeling is mainly based on the evaluation of the whole trustworthiness of a node based on its behavior. Two typical trust models are proposed; (i) a sociological model proposed by Gerlach [58], and a multi-faceted model proposed by Minhas et al. [107]. The **sociological trust management model** is based on the principle of trust and confidence tagging. The trust is derived based on four different input parameters: (i) a situational trust, which only depends on the situation treated, (ii) dispositional trust, which is based on the vehicle's own belief to trust others, (iii) system trust, which depends on the system and assurances it provides, and (iv) belief formation process which is the evaluation of data based on previous factors. Based on these trust models, Gerlach propose in his work a whole architecture to secure vehicular communication and a privacy model to preserve vehicles' locations. Trust models are after that related to each component of the proposed architecture to be maintained and updated. However, a combined method that groups all models together lacks in the work. The **multi-faceted trust management model** is developed based on three kinds of trust which are: (i) role-based trust, (ii) experience-based trust, and (iii) majority-based trust. These trust metrics are integrated to determine the trustworthiness of vehicles in a logical framework. The role-based trust exploits some predefined roles that are enabled through the identification of vehicles, the ones owned by the government or authorities deployed for law enforcing are given higher trust compared to other normal vehicles. The experience-based trust is built and updated based on direct interactions between vehicles and the satisfaction degree on their communicated information. The two previous trusts are combined together to result a so-called priority-based trust used to restrict the number of reports received by a vehicle from other neighbors after requesting information about an event in the network. In fact, a vehicle is able to request reports and advices from its neighbors based on the priority trust model. Finally, the majority-based trust is based on the evaluation of received reports and data related to vehicles such as locations and reporting time to decide about the most reliable advice to be followed. The main problem with this technique is that robustness was not extensively studied.

#### 2.4.4.2/ DATA-ORIENTED TRUST MODEL

Unlike entity-oriented trust modeling, which relies on vehicles trustworthiness, data-oriented models consider to evaluate data messages exchanged between vehicles to decide about their validity. It was demonstrated by Raya in [78] that this kind of trust modeling is more appropriate in ephemeral ad hoc networks such as VANET when they proposed a data-centric oriented trust establishment using Bayesian inference and Dempster-Shafer evidence to evaluate data related to a particular signaled event. Several intrusion detection systems to detect and evict malicious vehicles from the network are proposed while deploying this kind of trust modeling to evaluate vehicles [50].

Some of the proposed works target to build a security mechanism for protecting communications

and routing protocols without identifying specific kinds of attacks [50] [136]. In [50], Leinm et al. aim to secure a geographic routing protocol based on vehicles' trust level verification. To do that, authors propose multiple algorithms to control and verify the maximum accepted range of nodes and their mobility limit. The functioning of the proposed algorithms is based on beacons and information exchanged between vehicles to decide whether to consider one node cheating its position or not. Observations made by algorithms are combined to make a global decision about the real behavior of a vehicle and update its trust level. The proposed work suffers from some weaknesses such as non cooperation between vehicles, which limits the knowledge of the decision maker to its local one, and thresholds setting and manipulation which are hard to do. In [136], Gazdar et al. introduce a Markov model to manage the trustworthiness of nodes where each vehicle monitors its neighbors (monitored vehicles) and assigns to each one a local trust metric, which can change depending on the reliability of the neighbor and its cooperation ratio. A finite state machine with $n$ states models the trust and each state represents a trust value. The study of the switching process between trust states is performed based on the probability of the expected reactions of a vehicle. In their study, authors also treat the problem of camouflage attack where a node reacts positively to benefit from its trust metric and attacks by considering a transition from any state to the null trust, although they settle for an analytical study without any application to a specific scenario. However, developing a mechanism which encompasses all rules to deal with the different types of attack seems to be a very hard task which requires high soft and hardware capabilities and keeps always some open issues. So and based on these assumptions, the major part of studies target specific kinds of attacks (e.g. DoS, false alert, Sybil, etc.).

Two recent studies dealing with intrusion detection and eviction from the network were proposed by Sedjelmaci and al. in [165] et [166]. In the first work, authors introduce a new reputation based technique to face some types of attacks such as black-hole, warm-hole and resource exhaustion (known by DoS attacks) based on the cooperation between different network members. This IDS assumes that all vehicles are monitoring each others behavior in a promiscuous mode. In fact, vehicles are grouped into one hop clusters based on their speed where a cluster head (CH) is elected to monitor its neighbors reputation values and make a decision about the real behavior of each one. The CH also, collects reputations built by other normal vehicles before making its decision to evict a node. Whenever malicious vehicles are detected, their identifiers are sent to nearby RSUs, then to a central authority to be treated where a decision is made and an information is diffused in the network to eject attackers. Attacks are detected based on some rules dealing, essentially with the control of packet drop ratio and message duplication ratio. The major shortcoming of this IDS is its generated overhead due to the diffusion of malicious vehicles list in the whole network, also the time taken by CHs to reach an RSU and contact a CA may delay the reaction against attackers which increases the vulnerability window of a legitimate node. The second work follows the same idea, but extends rules to detect more types of attacks such as Sybil and decreases the generated overhead by designating some special nodes in the cluster to monitor neighbors and send reputations to the CH. In DCMD (on Data-Centric Misbehavior Detection) [119], a new mechanism was proposed to deal with false alert attacks by an excessive verification of the alerts diffused by vehicles. In fact, the identification of false alerts is based on a set of rules preloaded into each vehicle. For this reason, authors identify a list of actions and conditions to be verified by the alert sender, which are relative to each type of alert. Whenever the condition relative to the diffused attack is not respected, the vehicle should be considered malicious. Vehicles are working on cooperation with RSUs and CA to detect a malicious vehicle that changes its pseudonym.

Several studies have focussed on the Sybil attack detection in a vehicular network and proposed various techniques to face that kind of misbehaving which almost rely on the radio resource testing, identity registration and position verification. [27], [65] propose a resource testing technique to

detect Sybil nodes based on the assumption that each physical entity has limited resources. In [27], authors use computational puzzles [3] to test the computational resources of nodes. However, in [65] they demonstrate that an attacker may have more resources than a honest node and propose an amelioration using radio resources testing instead. In [83], [33] and [54], authors propose a new approach to detect Sybil attacks based on the Received Signal Strength (RSS) of exchanged messages. This approach is based on the verification of a message (beacon) content by eventually collecting measurements about its generator to verify if it is in the claimed position or not. So, the node should be considered as a Sybil attacker if its claimed position is far from the evaluated one. In [113], authors assume that two nodes should not have the same neighbor for a period of time that they call $\mu$ and proceed for the exchange of neighbors' lists between vehicles to verify common neighbors after that period. So, if a vehicle remains in common between two nodes more than $\mu$, it should be considered as a fake identity initiated by a Sybil attacker. Based on the assumption that an attacker and its fake identities share the same physical mean, they can identify the attacker.

#### 2.4.4.3/ HYBRID TRUST MODEL

Hybrid trust models are built based on a combination of the two previous ones where the data validation is considered together with the entity trust to build and update a trust level related to each vehicle. In [42], Dotzer et al. propose a distributed reputation model that exploits a notion called *piggybacking* where each vehicle that forwards a message about an event appends its own opinion about the trustworthiness of data in the message. An algorithm to generate an opinion about the validity of an event is proposed while considering five types of trust; direct trust, which is based on the node's own experience with the sender, indirect trust, which consists in the transitive second-hand reputation provided by nodes with an already known reputation information, node closeness to the area of an event, node familiarity with the declared event, and geo-situation oriented reputation level which is deduced from the two previous trusts. A decision about the validity of a message is made using an aggregation of opinions received from other vehicles. The major weakness of the proposed mechanism is its recursive use of opinions which always favors influence of the earliest received opinions than later ones. In [51], an approach for vehicles reputation building based on data validation is proposed. In this approach, nodes are of two types; a set of pre-authenticated nodes called anchors that are assumed to be trustworthy and their data are trusted, and normal nodes. So, data is validated directly when received from anchors or by an agreement between neighbors when communications between normal vehicles are considered. Malicious nodes are identified if the data they present is invalidated by the validation algorithm. The validation is based on experiences between vehicles and could be either negative or positive based on satisfactory completion of transactions, fulfillment of expectations, or some other forms of verifiable fiduciary action. For the exchange of reputations and agreements between vehicles, the authors propose to use an epidemic algorithm to enhance data availability and reliability. This mechanism does not consider reputations of normal vehicles to build the majority consensus and only relies on the reception of a sufficient number of reports despite the passive wait for these reports. In [97] and [159], authors propose a trust modeling technique to control the message relaying and decide about its validity. They target to detect false alerts diffused by malicious vehicles. Cooperation between vehicles is established to gather opinions about generated messages for the decision-making. The collection of opinions from network members is made based on a clustered architecture where each member introduces its opinion about a reported event in a message before forwarding. The final decision about the accuracy of a generated alert is delegated to the cluster-head based on two types of information: the first represents its own experience with the monitored node and the second is the combination of aggregated opinions in the received message. Therefore, the behavior of the message generator could be distinguished and the message continues to be diffused if it is

real and stopped in the opposite case. The idea of this trust modeling technique solves some problems left by the two previous ones related to decisions robustness and reliability and seems to be scalable thanks to its cluster-based opinions collection. In [163], authors propose a new framework for continuous vehicles' tracking while preserving their privacy (A-VIP: Anonymous Verification and Inference of Positions) based on a centralized architecture. A Local Authority (LA) is introduced to collect, manage and check the consistency of vehicles' broadcasted reports. They assume that vehicles own cryptographic material and are able to establish a secure channel to the LA through either an RSU or a cellular infrastructure. They employ a cooperative reporting mechanism to speed up reports delivery to the LA and enhance the position identification of a beaconer using car-to-car communication. Reports from neighboring vehicles are used to identify the beaconer tile. Two approaches are studied: (i) the Q-aware approach where the LA knows the quality of signal with which the reporter has received the beacon, and (ii) the Q-unaware approach describing the opposite case. In this work, the proposed mechanism is enhanced to calculate reporters trustworthiness and detect fake identities based on received information in the reports (reports combination). The trust mechanism is able to face other attacks such as transmit power, false location and replay attack. In [131], the use of connected RSUs managed by a trust authority (TA) is proposed to verify vehicles trajectories and detect sybil attacks with preservation of vehicles position privacy. Vehicles in the network communicate their trajectories based on authorized messages signed and given by an RSU. The set of trajectories given by one node is used for the Sybil attack verification based on a social relationship identification between generated trajectories using a similarity definition. In VWCA [112], a distributed monitoring technique was proposed to detect and classify vehicles based on their behaviors and exchanged information. In fact, vehicles with the highest trust level are chosen to monitor their neighbors and build a distrust level to be thereafter exchanged with a decision maker (cluster head), which makes a classification of vehicles into: normal, abnormal and malicious. Whenever a cluster head detects a malicious node, it sends an encrypted message to the central authority including its identity to be blacklisted and diffused to other vehicles in the network.

### 2.4.5/ SUMMARY AND DISCUSSION

Securing communications in a vehicular network and protecting its components from misbehavior and soft & hardware faults was till nowadays a challenging issue. Certificate revocation lists were integrated to the IEEE1609.2 standard to increase its capability to detect malicious vehicles and evict them from the network. However, their distribution to legitimate users still suffers from various weaknesses which keeps open doors for attackers to disrupt the network applications. For this reason, research directions were interested to intrusion detection systems (IDS), which are based on mutual evaluations between vehicles to increase the trustworthiness of the network. They mainly rely on vehicles trust building by evaluating nodes reactions, their issued data or combine the two techniques. This kind of protection is developed and used jointly with legal certificates acquired by nodes at their entrance to the network. Unlike mechanisms based on certificate revocation list, which always rely on centralized control and decision making based on certificate authorities, intrusion detection systems mainly rely on vehicles opinions to control their behavior. They enhance the cooperation between entities in the network (V2V and V2I) and push them to goodly behave because they are controlled by their neighbors. Intrusion detection systems overcome problems related to the time consumed to communicate the list of malicious vehicles to legitimate users encountered by CRL-based techniques and increase the cooperation between nodes based on a continuous and distributed control of their activities and interactions with their neighbors. For this reason, we aim in our work to investigate these techniques and base our proposal on them to

enhance routing and applications' security.

## 2.5/ CONCLUSION

During this chapter, we presented works related to the most important components in intelligent transportation systems related to this thesis, which are data collection and exploitation. A secure and efficient collection and exploitation of data flowing in the vehicular network open possibilities to develop new applications and enhance existing ones. In the first section of this chapter, we reviewed protocols developed for data collection which leads to an identification of weaknesses related to real-time traffic consideration that are still not convincing. In the second section, we discussed techniques for data exploitation and analysis to enhance ITS applications efficiency and preserve higher quality of service for the clients, a study that confirmed the existence of some issues that are not well investigated related to the non exploitation of vehicles cooperation and non consideration of their capabilities and parameters affecting the quality of service in a network. In the third section, we highlighted security issues that are encountered in a vehicular environment and reviewed mechanisms proposed in the field with the aim of protecting data and applications. Some reactive mechanisms detect attackers after they harm the network and cause losses and were not able to predict an attacker before it attacks. To overcome problems kept open by projects and works described during this chapter, our contributions described in the following chapters consist in the proposition of a secure routing solution and efficient and accurate techniques to exploit data exchanged between vehicles to protect the network from potential attackers, increase the quality of service of ITS applications and offer a new beneficial service able to preserve travelers time and money. Therefore, we propose in the next chapter, our first solution to deploy an optimized number of data harvesters for data collection from urban areas.

# 3

# Optimized deployment of data harvesters for urban sensing

## 3.1/ Introduction & problem statement

Cooperative Intelligent Transportation Systems require the establishment of a high cooperation between all vehicular network stakeholder (vehicles, infrastructure, etc.) to collect efficient and accurate data to offer reliable applications that will enhance security, driving conditions and travelers comfort. Building urban sensing applications seems to be more simple and sophisticated using vehicles' capabilities. Vehicles, while traveling through an urban area, collect many kinds of local information like pollution level, videos, temperature, chemical toxicity, or free parking lots which gives a global vision about what is really happening in that area, reduces passive traffic and help to locate survivors and plans for rescue missions in case of disasters. Collected data by each vehicle should be delivered to a third-party to be used for continuous monitoring and sensing applications refresh. Data collection protocols reviewed in section 2.2 rely on the cooperation between vehicles to achieve a high packet delivery ratio and enhance information availability to users. They are mainly based on two kinds of network organization which can be either decentralized, where vehicles are autonomously organized, or centralized where a static fixed third-party is responsible of the network organization. The most investigated type of data collection architectures is decentralized, but in some cases such as emergency scenarios where connectivity and communication between vehicles is not guaranteed, these solutions have proven inefficiency. This limitation could not be ignored especially in our work because we are dealing with the problem of emergency cars management in case of search and rescue missions in disaster urban areas. This is also the use case treated by the European project CarCoDe [1]. Therefore, we choose to consider a centralized solution where a central unit plans and continuously monitors a set of vehicles namely harvesters to get a global overview of the affected area. A harvester is traveling through an area defined by the central unit to collect data from vehicles and send them back to be exploited by urban sensing applications. However, our proposed solution brings a new challenge and two questions should be answered: How many harvester we have to deploy in a geographic area? and which area to be monitored by one harvester while considering time constraints of the application? For this reason, optimizing the number of deployed harvesters in an urban area will be the first focus of our study while proposing our techniques to handle problems related to data collection and exploitation in vehicular networks.

Optimizing harvesters in an urban area is a nested problem that contains two main fields to be treated which are the optimization of each harvester movement circuit and the optimal number of

[1] https://itea3.org/project/carcode.html.

these harvesters. Typically, many works have dealt with optimization problems but none of them has treated the problem of data harvesters deployment. Facilities location, covering problem, Vehicle Routing Problem (VRP) and Chinese postman problem (CPP) were among the most investigated problems in the literature.

In facilities location problem [32], each facility should be optimally placed in order to distribute goods to clients at a minimum cost. In [155], Mehar et al. treat the problem of locating charging stations to serve electricity demands generated by electric vehicles traveling in the network. A Genetic algorithm was used to minimize the investment and transportation costs needed to place the charging stations depending on clients demands which increase during rush hours. The covering location problem is how to find optimal positions to cover all clients demands. In [126], the problem of optimizing Road Side Units placement was investigated. The main objective in this problem was the minimization of the delivery time of an abnormal activity detected by a vehicle in a road segment to the nearest RSU. In this work, authors consider intersections as potential candidates where to place RSUs and try to choose the best ones among these locations to cover the maximum area and minimize the delivery time. They compare two heuristics: Balloon Expansion Heuristic (BEH) and Binary Integer Programming (BIP) to solve the problem. In VRP, the problem is how to manage a set of vehicles to optimally distribute goods to a number of dispersed clients while respecting capacity constraints of vehicles, the priority of demands and time intervals of delivering. In [102], authors treat the problem of emergency vehicles planning to serve emergency calls. They consider a number of stations where each one is delegated to manage a set of vehicles and satisfy calls in a predefined area. In this work, an Ant-Tabu algorithm was used to minimize a number of functions (e.g. time to a station, time to a call, etc.). In [71], authors compare exact algorithms, heuristics and meta-heuristics to solve the VRP with time window constraints. One of the used heuristics is the construction and amelioration where they used a greedy Best Insertion (BI) algorithm to build an initial solution and a Variable Neighborhood Search (VNS) heuristic [24] for optimization. In the VNS heuristic, a combination of six heuristics were defined as neighborhood relations. Three mono-tour heuristics: 2-Opt, Or-Opt, 2-Exchange and three multi-tours: String Relocation, String Exchange and String Cross. The Chinese postman problem is a graph problem, which consists in finding the best circuit to be traversed by the postman while minimizing the cost and covering at least once all arcs of the graph. It becomes a NP-Hard problem when a mixed graph is considered. Several heuristics were developed to solve this problem. Among them, we cite GRASP (Greedy Randomized Adaptive Search Procedure) which was deployed by Corberan et al. [19] and is based on two phases; randomized construction and an amelioration phase which are executed successively several times to result the optimal solution.

We notice that all of the treated problems are NP-hard and in most cases heuristics and meta-heuristics are deployed to solve them. Compared to them, our data harvesters optimization problem is similar to the Chinese postman and could be treated as a covering or vehicle routing problem, however with a different objective which is the covering of each road segment and not only some points. This treated objective introduces more difficulties to our optimization problem compared to others. In addition, applications time constraint constitutes a rigid parameter to be taken into account. In the remainder of this chapter, we describe how we model and solve the data harvesters optimization problem.

## 3.2/ Optimization of Data Harvesters Deployment in Urban Area

We aim to optimize the number of harvesters to be deployed by a central unit to monitor an urban area and cover all its roads to collect useful data for urban sensing applications. For this reason,

we target to maximize the distance covered by every harvester in the monitored area with respect of the time constraint imposed by the application limiting the delay between two successive visits to the same region. Therefore, the global problem of harvesters optimization is divided into a set of tours optimization sub-problems. In this section, we first give a mathematic model to formulate the problem, then we detail the proposed solution.

### 3.2.1/ PROBLEM MODELING

In this work, we target to give answers to the questions given in the introduction by defining the number of harvesters and their areas to explore. The idea is to divide the roads topology into segments separated by intersections. So that the urban area is modeled by an oriented graph where intersections are vertices and road segments are edges and a harvester should travel a portion of the graph and visit a number of edges. Typically, we should define an objective function to be optimized and a number of constraints to be respected, but before that lets introduce the set of parameters used in our model which are presented in Table 3.1:

| Parameter | Definition |
|---|---|
| $n$ | Number of vertices/intersections in the graph |
| $V$ | Set of all vertices/intersections: $\{i$=0,..$n$ where $i \in V\}$ |
| $E$ | Set of edges/road segments: $\{i \in V$ and $j \in V$ where $(i,j) \in E\}$ |
| $z_{ij}$ | Binary variable: =1 if segment $(i,j)$ exists and =0 else |
| $l_{ij}$ | Length of a segment $(i,j)$ |
| $t_{ij}$ | Time needed to travel a segment $(i,j)$ |
| $v_{ij}$ | Maximum speed of a harvester in a segment $(i,j)$ |
| $y_{ij}^k$ | Decision variable: =1 if the segment $(i,j)$ is visited by a harvester k and =0 else |
| $T_{MAX}$ | The maximum time between two successive visits to a point in the segment |
| $m$ | Number of resulted harvesters |

Table 3.1 – Mathematic model notations

To optimize the number of harvesters being deployed in an urban area, each harvester should cover a maximum portion of the oriented graph which means maximizing the traveled distance in the area before to return to the starting point while respecting the time constraint $T_{MAX}$. For this reason, we divide the problem into $m$ sub-problems where the objective of each one is the maximization of the distance traveled by a harvester $k$. So, let $F$ be the distance function to be maximized by a harvester $k$, such as described by equation 3.1.

$$F = Max \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} l_{ij} y_{ij}^k \tag{3.1}$$

The feasibility of the solution depends on different constraints represented by the following equations: equation 3.2 guarantees that the maximum time between two successive visits to a point in a segment should not exceed a threshold $T_{MAX}$ defined by the application, equation 3.3 ensures that every road segment should be visited at least once by a harvester where $m$ is the number of harvesters, equation 3.4 guarantees to have a circuit with the same start and end point, and equation 3.5 is the integrity constraint.

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} t_{ij} y_{ij}^{k} \leq T_{max} \tag{3.2}$$

$$\sum_{k=0}^{m-1} y_{ij}^{k} \geq 1 \tag{3.3}$$

$$\sum_{i=1}^{n-1} y_{i0}^{k} + \sum_{j=1}^{n-1} y_{0j}^{k} \geq 2 \tag{3.4}$$

$$y_{ij} \in \{0,1\} \tag{3.5}$$

In the next subsection, we explain how to solve the problem we have defined to optimize the number of data harvesters to collect data from an urban area.

### 3.2.2/ Optimization solution

Urban sensing applications, especially emergency ones in case of disasters, require data to be received at predefined bounded delays. So, time is an important rigid constraint to be respected. Hence, the algorithm to be used to solve the harvesters' deployment problem should give results in a reasonable time. Furthermore, harvesting a big area makes the optimization problem we are studying NP-hard, where its resolution with exact solution is not possible and may take huge time to deal with all possibilities and constraints. For this reason, we opt for the use of meta-heuristics to handle complexity. It is clear that they don't give always the optimum solution but they approximately hit optimality in a reasonable time.

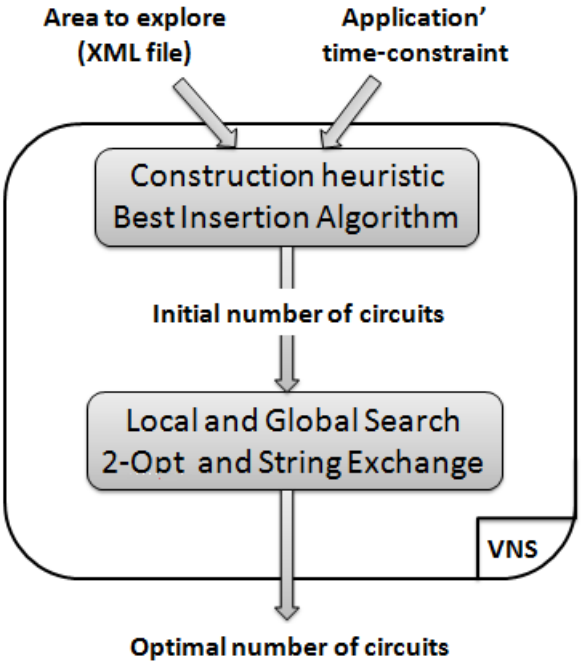



Figure 3.1 – VNS Operation

To deal with the complexity of the problem and solve it, we modified a famous meta-heuristic which is the Variable Neighborhood Search (VNS) [24]. VNS is a meta-heuristic which always starts with an initial solution obtained by construction heuristics, then it defines a number of operators $N_k$ (k=1..$k_{max}$) to be applied to this state (current solution) resulting a set of successor states. The set of obtained states by applying an operator is called a neighborhood and it is optimized by local search techniques. For the construction of initial solutions, we reshape the greedy Best Insertion (BI) heuristic to respect our specification and constraints. However for neighborhoods, we have defined an heuristic which is the combination of two well known heuristics: 2-Opt used for TSP [14] which is a Local Search Heuristic and String Exchange [116] which aims to globally optimize the solution by exchanging k strings between two circuits (tours). Figure 3.1 introduces the global functioning and transitions in VNS. More details about the solution steps namely construction and neighborhood are presented in the following subsections.

#### 3.2.2.1/ CONSTRUCTION HEURISTIC

We detail, here, the Best Insertion heuristic modifications we have made to adapt it to our problem. Traditionally, BI is used for minimization problems and the construction deals always with a set of points (graph vertices) which are mostly cities in case of TSP or clients in case of VRP. However in our case, the objective is maximization and we have to deal with road segments (graph edges). So, we modify the BI to accept edges while building a cycle and consider the maximum length as the addition criteria without forgetting the time constraint of the application. The algorithm is detailed below 1:

**Algorithm 1** Pseudo-code of the modified BI algorithm

1: **Input:** A valuated graph with ($l_{ij}$,$t_{ij}$) distance and time needed to travel over segment $(i, j)$
2: **Output:** A set of circuits delimiting the areas covered by each harvester
3: Choose a random segment as first cycle;
4: Browse all segments and add the one with maximum length;
5: **while** $\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} t_{ij}y_{ij}^{k} < T_{max}$ **do**
6: Repeat action 4;
7: **end while**
8: **if** $\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} t_{ij}y_{ij}^{k} \geq T_{max}$ **then**
9: Choose the last segment as the first cycle of the new circuit;
10: **end if**
11: Repeat actions 6 and 8;
12: Stop the algorithm when there is no more segments to add;

At the end of the modified greedy best insertion algorithm process, we get a set of non optimal circuits affected to each harvester. These circuits are not the optimal ones. For this reason, they should be introduced as inputs for the amelioration algorithm (neighborhood heuristic) to be improved to hit the optimal solution that gives the minimum number of harvesters needed for deployment in all the studied area.

#### 3.2.2.2/ NEIGHBORHOOD HEURISTIC

At this stage, we take the initial solution (set of circuits) given by the Best Insertion heuristic and we make amelioration at each iteration to finally give an optimal solution. For this reason, we define a new heuristic combining 2-Opt and String Exchange. This heuristic defines the VNS neighborhood used for local and global optimum search. In this heuristic, we take the set of circuits produced by the construction algorithm, we browse all of them and choose at each iteration two adjacent circuits. We first, browse local segments of each circuit and try to exchange two segments if this action improves the time consumption. Then, we compare the two circuits and see if we can exchange segments between them. The exchange is made if it minimizes the number of travels for one segment or minimizes the time of the circuit. A simplified process about the local and global optimizations in the neighborhood heuristic is given by figure 3.2 and its different instructions are detailed in algorithm 2 with $N$ the number of circuits produced by the modified BI algorithm, $C_i$ the circuit $i$, $K^i$ the number of segments in $C_i$, $S^i_j$ is the segment $j$ in the circuit $i$.

**Algorithm 2** Pseudo-code of the amelioration algorithm

```
1: Input: Set of harvesters circuits produced by the BI algorithm, the distance of each circuit and
   time to travel each Circuit.
2: Output: A set of optimal routes (circuits) delimiting the areas covered by each harvester.
3: for i = 1..N do
4:     Choose C_i and C_{i+1};
5:     for j = 1..K^i − 1 do
6:         for k = j + 1..K^i do
7:             Make a virtual exchange between the two segments S^i_j and S^i_k;
8:             Recalculate the new configuration time and compare it with the initial time;
9:             if New time < Initial time then
10:                Validate exchange;
11:            end if
12:            for h = 1..K^{i+1} − 1 do
13:                for x = h + 1..K^{i+1} do
14:                    Do 7, 8 and 9 with S^{i+1}_h and S^{i+1}_x;
15:                    Make a virtual exchange between S^i_j and S^{i+1}_h;
16:                    Recalculate the two times of C_i and C_{i+1};
17:                    if Time is ameliorated then
18:                        Validate the exchange;
19:                    end if
20:                end for
21:            end for
22:        end for
23:    end for
24: end for
```

The heuristic we have defined uses a local search approach given by the 2-Opt heuristic and a global optimization by comparing adjacent circuits to make improvements which gives a global optimum for the problem. The algorithm gives a number of optimal circuits as a final result. In our case, and as said above, a harvester is delegated to explore/monitor an area and travel a defined number of road segments (circuit) which means that the number of harvesters will be the same as the circuits number.

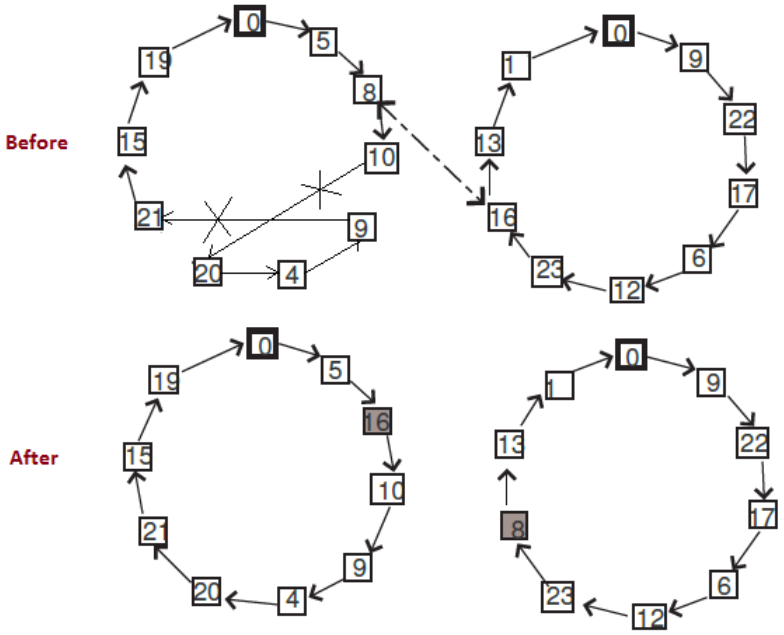



Figure 3.2 – Neighborhood heuristic illustration

## 3.3/ PERFORMANCE EVALUATION

In this section, we evaluate our proposed solution using the Manhattan road topology and we expose results to discuss the impact of our heuristic on time and CPU calculation costs.

### 3.3.1/ CASE STUDY

Manhattan is one of the most dense borough of the New York city, it is of $87.46Km^2$ of size with approximately $28Km^2$ the size of water. Most of the Manhattan roads are unidirectional which leads sometimes to a bigger time consumption to reach a near point and can affect the final results of the proposed algorithm (see figure 3.3). So, to test our proposed algorithms, we extract the manhattan topology in an XML format using OpenStreetMap (OSM)[2], then we treat it to eliminate redundancy and keep only roads to be able to use them for circuits construction in the first heuristic.

### 3.3.2/ RESULTS

We firstly evaluate the number of harvesters given by the initial solution (Best Insertion algorithm) and the final one given by the neighborhood heuristic. Then, we calculate the CPU time consumed by each of them to give the solution. Initially, we set $T_{MAX}$ to one hour and we start the execution. Results are summarized in table 3.2 where each value represents an average of twenty runs. We observe that the number of harvesters given by the initial best insertion algorithm is approximately optimal and the time consumed by this latter is higher than the Neighborhood time. This is due to the higher value of $T_{MAX}$ which gives the possibility to the BI to loop several times and increase

[2]http://openstreetmap.fr.

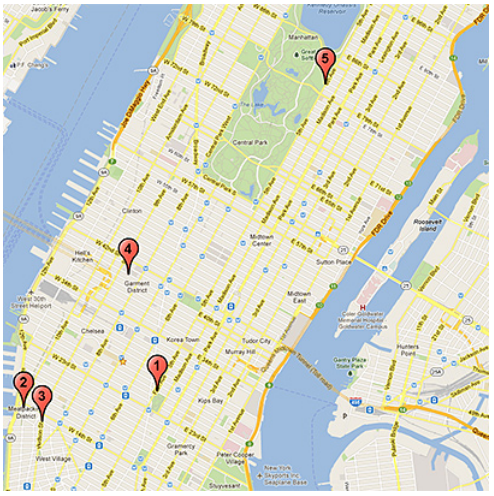



Figure 3.3 – Manhattan roads topology

the circuit length as the application time-constraint is respected. So, in this case, we can limit our optimization to the BI and we do not need to consume more time in the neighborhood heuristic.

| Algorithm | **Harvesters' Number** | **CPU Time (ms)** |
|---|---|---|
| **Best Insertion** | 24 | 47 |
| **Neighborhood** | 21 | 30 |

Table 3.2 – Best Insertion vs Neighborhood heuristics

In figure 3.4, we depict the evolution of the harvesters number depending on the application time-constraint. The x axis represents $T_{MAX}$ value increasing from 15min to 2hours and y axis represents the number of deployed harvesters for each value of $T_{MAX}$. The figure demonstrates the impact of the time-constraint rigidity on the final solution of our heuristics where the number of harvesters rapidly decreases when $T_{MAX}$ increases. We need a high number of harvesters for $T_{MAX}$=15min (73 harvesters) where we need only 5 harvesters for $T_{MAX}$=2hours. Hence, in the first case, we should collect information and take a global overview of a topology of $87Km^2$ in 15 min.

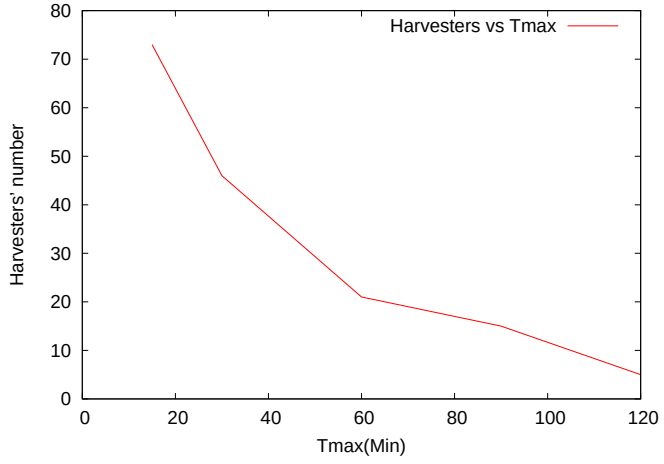
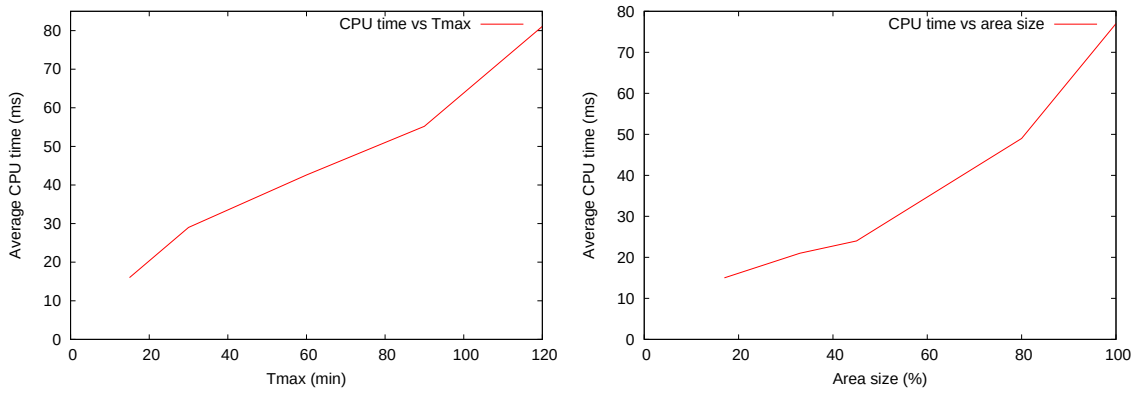



Figure 3.4 – The number of harvesters vs Tmax




(a) (b)

Figure 3.5 – Average CPU calculation time vs (a) Tmax and (b) topology size

Figure 3.5 (a) highlights the average CPU calculation time when the time-constraint increases. We run the algorithms 10 times for each value of $T_{MAX}$ and we calculate the average CPU calculation time. We notice that the calculation time of the proposed heuristics increases from 15ms for $T_{MAX}$=15min to 81ms for $T_{MAX}$=2hours. The time increase is due to the number of loops executed by each heuristic because the time-constraint is being higher which means that it won't be exceeded rapidly and the search is larger. However, it stays always reasonable and doesn't evolve exponentially.

As far as we know, in case of a chemical attack or fire, the size of the affected area rapidly increases to affect the surroundings. So, the given heuristics should continuously give faster solutions regarding the size of the area. To evaluate the ability of our algorithms to give an optimal solution at a minimum calculation time cost, we apply them to different sizes of Manhattan topology. We consider different percentages of the total topology as affected areas increasing from 15% to 100%. Results, in figure 3.5 (b), show that the CPU time of our proposed algorithms increases linearly when the size of the affected area increases. This means that even for a very large topology, the time to calculate the optimal solution stays reasonable and doesn't jump rapidly to an exponential value.

Regarding the given results in table 3.2, figures 3.4and 3.5, we notice that our heuristics are able to deliver an optimal solution at a very reasonable time even if the size of the topology to be monitored is very large or the time-constraint increases. These results prove the higher performance of the given algorithms. In table 3.2, we also conclude that the Best Insertion heuristic is able to give an approximative optimal solution especially for a high value of $T_{MAX}$ which means that we can minimize the CPU time by limiting calculation to the first construction solution.

## 3.4/ CONCLUSION

In this chapter, we propose an optimized deployment of data harvesters to monitor urban areas and be able to gather information for building sensing applications. We first model the problem to identify objective functions, then we propose a Variable Neighborhood Search (VNS) heuristic to solve it. For this reason, we modify the greedy Best Insertion heuristic to give an initial solution for the problem. The given solution is used as an input for the neighborhood heuristic defined by the

combination of the 2-Opt and String Exchange heuristics to get a final global solution. The adoption of heuristics and not exact resolutions is imposed by the urban sensing applications' requirements and the complexity of the problem when the monitored area is big. Experimental results given by the application of the algorithms to a Manhattan topology prove the efficiency of our proposals regarding the processing time and the number of resulting harvesters.

The optimized number of deployed harvesters in an area are traveling along specified circuits to gather information from vehicles and capture their own data based on embedded sensors. The collected data should be communicated to the central unit in a secure and efficient manner to be exploited to get a global overview about the network and increase applications reliability. Thereby, we dedicate the next chapter to propose a secure geographic routing protocol to deliver securely data to a final(s) destination(s).

# 4

# SECURE DATA COLLECTION IN A VEHICULAR NETWORK

## 4.1/ INTRODUCTION & PROBLEM STATEMENT

As stated above, vehicles capabilities are increasingly deployed by various kinds of applications to enhance the daily human life and satisfy its requirements. Sensing capabilities such as embedded sensors and cameras are used to collect several environmental information and internal diagnostics, whereas available communication technologies are used to exchange data with each others and enhance their knowledge. Figure 4.1 illustrates a simple data collection process which includes vehicles and infrastructure.

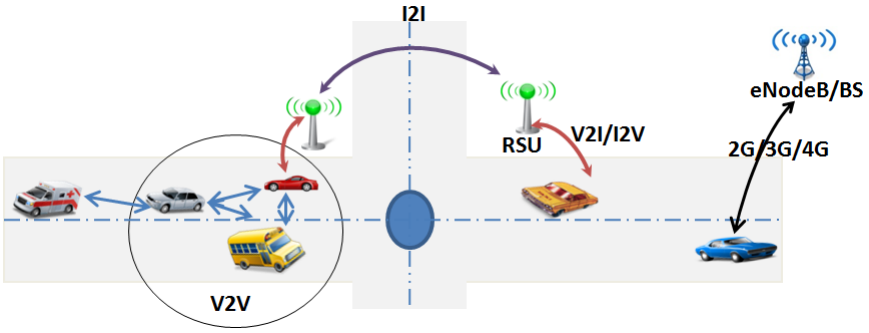



Figure 4.1 – Data collection in a vehicular network

Data collection protocols reviewed in section 2.2 propose several techniques to exchange and relay data between communicating nodes (vehicles and infrastructure) while exploiting existing communication capabilities. They rely on the cooperation between all network stakeholder and assume that all nodes will agree to communicate their collected data and forward received information from neighbors. However, in some cases, some malicious nodes may benefit from this confidence to threaten the network and affect running applications by acting on the data they receive or falsifying their own sent data. Several kinds of attacks that may be used by nodes with malicious intentions to harm the network which are summarized in subsection 2.4.1. Network and application attacks are considered the most dangerous classes due to their impact on network capabilities since they aim to slow down communications and disrupt running applications. For this reason, we focus in this chapter on the study of attacks which fall under these classes namely denial of service (DoS), Sybil, false alert and packet alteration attacks to firstly design a secure geographical routing protocol namely S-GyTAR able to protect exchanged data, then develop a prediction technique that contin-

uously monitors the network members and prevents from these attacks namely Intrusion Detection and Prevention System (IPDS).

Regarding the dynamicity of a vehicular environment, its unique characteristics via the frequently changing topology and lack of deployed infrastructure, it is highly recommended to imagine the worst case where interactions and information exchange between vehicles and fixed third parties are not always guaranteed. For this reason, we base our proposals on a completely distributed architecture for vehicles monitoring and real-time information collection. In fact, the proposed architecture organizes the network members into one-hop clusters leaded by a carefully chosen node called cluster head (CH). The use of a clustered architecture in our case is motivated by three reasons. Firstly, due to speed limitations, stop signs and traffic lights in an urban area, vehicles are often moving into groups which means that a formed cluster will hold along a road segment until a CH changes its way guaranteeing thus the architecture stability and low communication overhead. Secondly, it has been proven that cluster-based algorithms are more appropriate for a better delivery ratio and reduce broadcast storms [144] [149] [165]. Finally and as stated above, we cannot rely on infrastructure deployment especially in an emergency case which is the scenario of the European project CarCoDe.

In our designed secure routing protocol, we aim to consider a completely decentralized and real-time evaluation of the traffic to identify malicious nodes and exclude them from the candidates' list of data forwarders to guarantee that only normal (trusted) nodes participate to the data collection process, thus secure the data being exchanged. In fact, we based our work on the well known position-based protocol, GyTAR [87], previously designed within our team work. We benefit from its consideration to the real-time traffic and reshape it to continuously monitor vehicles to secure the routing. We consider, in this protocol, the trustworthiness of a node to define its sociability and eligibility to forward a packet. For this reason, the network is organized into a clustered architecture where a cluster head (CH) is responsible of its cluster members control to detect selfish and misbehaving ones to evict them from the network and inform other members about that.

As a next contribution under the umbrella of secure and efficient data collection, we have designed a new intrusion detection and prevention system (IPDS) which is also based on the same idea of network organization and monitoring to detect and prevent from potential attacks that may occur in a vehicular network. So, based on a continuous mutual monitoring in a promiscuous mode between vehicles in the same cluster, a prediction technique using the well known Kalman filter [1] and a classification method, the IPDS aims to deal with various attacks such as DoS, Sybil and false alerts, detect and evict attackers before they harm the network performances. It is also able to face oscillating attackers and camouflage attack where a node behaves normally to gain the confidence of its neighbors before to attack. So, IPDS targets to build a secure network and prevent running applications from potential disruptions and losses by detecting malicious nodes before they persist in their attacks, thus decreases the vulnerability window.

This chapter is divided into three main sections. In the first section, we give a short overview about GyTAR. The second section is delegated to detail S-GyTAR, our first contribution in this area, where we highlight the main building blocks of the routing technique, the network organization and nodes monitoring to collect real-time traffic information and detect attackers. We devote the third section to study IPDS, where we detail the monitoring architecture, nodes classification and their eviction from the network.

## 4.2/ GyTAR overview

GyTAR [60] is a geographic routing protocol based on vehicles positions exchanged periodically between vehicles using CAM messages (Beacons). Routing decisions in this protocol are made intersection by intersection based on a weighting method to choose the next intersection where a packet should be sent such as described in figure 4.2. The weights of intersections are calculated based on two factors: (i) the distance of the intersection from the final destination of the packet, and (ii) the density of the segment leading to that intersection. The distance from a packet destination is given based on the geometric shape of roads, whereas the density of the segment is gathered based on clustered organization of vehicles moving through that segment. In fact, the segment is divided into equal small cells with the size of vehicle's radio range, where a cluster head is chosen based on its closeness to the center of the cell to calculate its neighborhood and send the number back to the previous intersection. The density of the cell is included into a Cell Density Packet (CDP) and sent whenever the CH reaches the next intersection. During its passage by every CH, the CDP packet is augmented by the density of the cell leaded by that CH until reaching the previous intersection. When a packet holder reaches an intersection, it receives CDPs from all underlying intersections, it weights them, then forwards the packet through vehicles in the segment leading to the intersection with the highest weight.

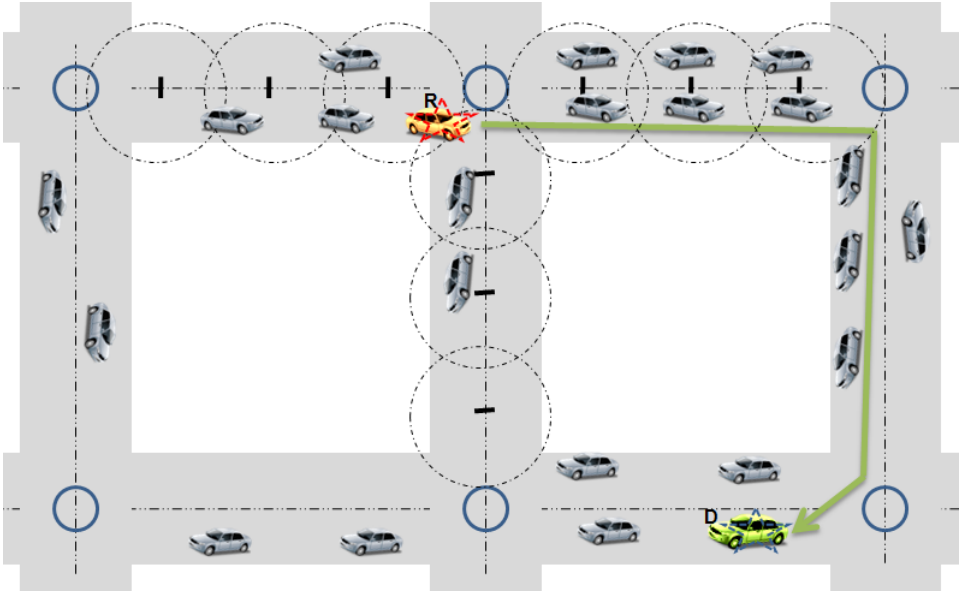



Figure 4.2 – GyTAR routing strategy

## 4.3/ Secure intersection-based routing protocol for data collection in urban vehicular networks

The main concern of this routing protocol is to ensure a secure data exchange between moving nodes in a completely mobile network that lacks infrastructure such as RSU or eNodeB while considering real-time information about the traffic, something that may lack in the majority of the reviewed protocols in section 2.2. The denial-of-service attack is the major security concern faced by the actual routing protocol. Therefore, a new distributed technique able to monitor the network members periodically is proposed. It is based on a cluster hierarchy where a cluster head evalu-

ates its neighbors behavior based on a reputation and trust model, identifies attackers and alerts other nodes in the network. In the following we detail the trustworthiness management technique, network organization and the routing mechanism, then, we highlight the performance results of the proposed protocol.

### 4.3.1/ PROTOCOL DESCRIPTION

We devote this section to explain in details the different components of the proposed routing protocol including the reputation mechanism used to evaluate vehicles behavior, the way vehicles are classified and evicted from the routing process and how data are routed to a destination.

#### 4.3.1.1/ REPUTATION & TRUST MODEL

Nodes in the network are judged regarding their behavior and interactions. This behavior is translated to a quantitative criterion to enable the evaluation. A metric named trust level is introduced to model the trustworthiness degree of each node that can vary according to its reaction in the road. We highlight in this subsection the trust value calculation of each node in the network.

A cluster-head in the network maintains a trust value for each one-hop neighbor. It is always listening to its neighbors generated traffic to update their associated trust values. At the first time a new vehicle enters to its range, it is considered trustworthy and an initial trust value equal to 1 is associated to it. After that, this value is updated according to its behavior. Let's denote $T_{ij}$ the trust value built by a cluster-head $i$ for a node $j$. The CH captures all the traffic generated by each neighbor in its radio range and periodically evaluates its trust value. After each evaluation, we could guess that one node either keeps its normal behavior or tries to attack. For this reason, we introduce a new metric that reflects the reputation of the node after each evaluation and used later on to get the trust value. The reputation of a cluster head $i$ toward a node $j$ at the $n$-th evaluation is denoted $R_{ij}^n$ and is calculated according to equation.4.1.

$$R_{ij}^n = \left\{ \begin{array}{rl} \lambda * R_{ij}^{n-1} + (1-\lambda) * r_{i,j}^n & \text{if} \quad n > 1 \\ r_{i,j}^n & \text{if} \quad n = 1 \end{array} \right\} \tag{4.1}$$

$R_{ij}^{n-1}$ represents the reputation of node $j$ calculated after $n-1$ evaluations and $\lambda$ is a weighting factor between the latest evaluation and the previous ones, which is used to moderate and control the impact of the last evaluation compared to the existing value ($\lambda$ may increase or decrease depending on the variation of the latest behavior of the node in question). While $r_{i,j}^n$ is a note assigned by the cluster head $i$ to the node $j$ at the $n$-th evaluation and could be equal to -1 if the node behaves maliciously and equal to 1 if it acts normally.

Because we aim to face the denial-of-service (DoS) attack and more precisely the resource exhaustion in our routing protocol as previously stated, the main criterion we are considering to evaluate the nodes is the number of generated/forwarded packets by each host. This metric is chosen to limit the network jamming with non useful messages and protect the available bandwidth from potential overloads, if we control the number of packets generated/forwarded by each node and ensure a fair partition between all of them regarding the bandwidth utilization. Therefore, we define a threshold $P_{th}$ to specify the highest number of packets a normal node could send during the monitoring period without being considered malicious. Firstly, this threshold is chosen to be equal to the number of packets sent by the cluster head as it is considered to be the most trustworthy node, then it should take the value given by the average number of packets sent by all non malicious vehicles. In fact,

a CH maintains for each node an association between its identifier and its generated packets to keep a track of their number and compute them. For this reason, the time space is divided into different intervals controlled by a timer and counted using a local counter. At the expiration of a timer, the CH evaluates all nodes in its association table where the number of packets is compared to the defined threshold. At this stage of node evaluation and packets comparison, two possible cases may appear, where $Nb^{ij}_{packet}$ is the number of captured packets by CH $i$ and generated by neighbor $j$ (includes all packets for all destination and either locally generated or forwarded from other sources):

**- Case 1:** If $Nb^{ij}_{packet} \geq P_{th}$, the node has excessively generated or forwarded a high number of packets compared to the fixed threshold, which means that it could be suspected as a potential attacker aiming to jam the network, so it is punished by the CH and $r^n_{i,j} = -1$.

**- Case 2:** If $Nb^{ij}_{packet} < P_{th}$, the node has generated or forwarded a normal number of packets and didn't try to harm the network by non useful forwarding, so it shall be considered by the CH as a normal node and its reputation value will be positive by $r^n_{i,j} = 1$.

When the reputation related to a node $j$ is computed at the end of every time interval, a trust value ($T_{ij}$) varying between 0 and 1 should be attributed to $j$ and registered by the CH based on equation 4.2 where $n$ is the number of evaluations ($n$-th time interval). This trust level value is later on used to classify vehicles and identify malicious ones to evict them from the routing process. It is kept updated at each time interval based on vehicles reputations.

$$T_{ij} = Max\{R_{ij}/n, 0\} \tag{4.2}$$

#### 4.3.1.2/ PERIODIC GATHERING OF TRAFFIC DATA & ROUTING

Data, in the network, is routed using a geographic routing protocol which deploys the greedy forwarding technique. In fact and as stated above, the actual work is based on a previous routing mechanism namely GyTAR where the packet is forwarded intersection by intersection until it reaches its final destination. The selection of the next road segment where the packet should be sent is made based on a real time collection of information about roads densities. Data about traffic is gathered based on a cluster organization proposed in GyTAR. In this work, we propose to secure this mechanism as follows. Each segment is divided into small cells and the trustworthiest and nearest node to the center of a cell is chosen as a cluster head to collect information about traffic. The trustworthy node is identified using a pre-processing phase at the beginning of the network organization where all nodes in a cluster are mutually monitoring for a period of time to get an initial idea about the trust level of each vehicle. This pre-processing phase leads, therefore, to the choice of the node with the highest trust level to monitor nodes within the cell. After choosing the first CH, the maintenance of the architecture and the swapping of CHs would be simple as this one knows about all the trusts of its neighbors, it will directly choose the trustworthiest node to be a CH before getting out from the cell. The information in a cell (one-hop cluster) encompass the density of nodes in such a cluster and the trust level of each one. Therefore, each cluster head monitors its one-hop neighbors, identifies their reactions and behaviors regarding their packet generation rate and classifies them into two categories: normal and malicious vehicles. This classification is done based on the trust level value of each node calculated and updated based on the rules described in the previous subsection.

A cluster head in a cell maintains a list of its neighbors and associates to each one a trust value. It creates two types of lists to classify and differentiate nodes' behaviors: a black and a white list. The white list is used to store the neighbors' identifiers that are normally behaving and considered to be

trustworthy whereas the black list contains nodes' identifiers that behave maliciously and considered as attackers that require ejection from the network. So and as the CH is always listening to the network, it evaluates the trustworthiness of every neighbor continuously in a periodic manner as stated above. After each evaluation and trust level update, a node $j$ is either considered trustworthy or malicious based on the following rules where $T_{ij}$ is the trust value calculated by the cluster head $i$ for the node $j$ and $d$ is a maliciousness threshold indicating the tolerated limit value of trust to consider a node trustworthy, which may vary depending on the application security requirements.

**- First case:** If $T_{ij} \in [d,1]$ the node $j$ is considered trustworthy and its identifier is stored in the white list. This means that it can be used by any other node as a potential forwarder for its packets in a secure manner,

**- Second case:** If $T_{ij} \in [0,d[$ the node $j$ is classified malicious and stored in the black list to be later ejected from the network. It can not be used at any packet forwarding process and its packets won't be treated by other nodes neither forwarded.

After the collection of information about its neighbors, their evaluation and classification, the CH calculates the density of its cluster by only counting the number of nodes contained in the white list. This strategy of calculation favors the roads with the minimum number of malicious nodes when selecting the next intersection through which we have to route a packet during the routing phase. Figure 4.3 highlights the complete process of monitoring in a cluster and the information exchange between CHs to build a global overview about traffic in a road segment. Therefore, the CH should react in two ways: informs its one-hop neighbors about malicious nodes and communicates the information to the other clusters. For this reason, we rely on the so-called CDP (Cell Density Packet) packet defined in GyTAR and we reshape it to add information about monitored neighbors in each cell.

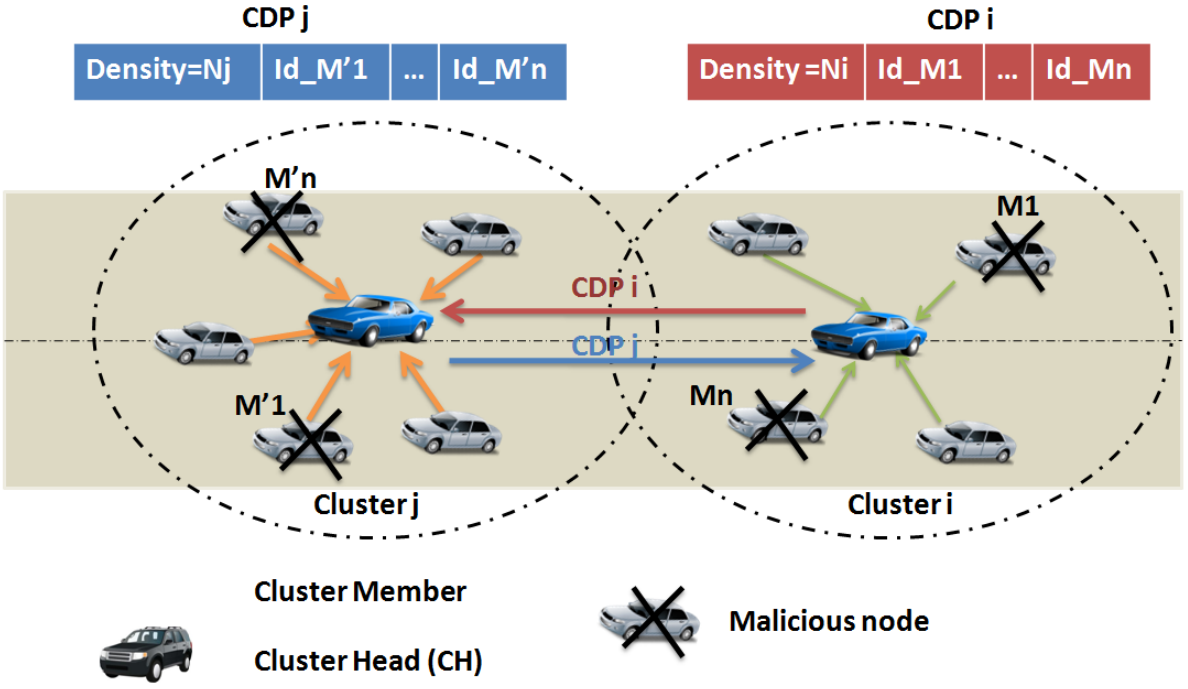



Figure 4.3 – Cluster monitoring & information exchange in S-GyTAR

So, the density of the cluster and identifiers of nodes stored in the black list are fulfilled in the CDP packet, which is later on diffused in the one-hop cluster and relayed by the farthest node in

the radio range to the next cluster head. It is relayed hop by hop from one CH to another until reaching the intersection. When a member in the cluster receives a CDP, it stores the contained identifiers of malicious nodes and ejects them from its routing table. However, when the packet reaches a CH, it is treated in a different way. In fact, the CH firstly extracts the list of black nodes and stores them into its own black list, then it updates the packet by adding its cluster density and black list and finally it forwards it to the next CH and informs its neighbors about the new black list it has. When a CDP reaches an intersection, the forwarding and update mechanisms are stopped for two reasons: the first one is to limit the flooding of all the network by a big number of messages that could not be useful at a certain time because of the frequent change in the network topology and density, and the second reason is that the information a CDP contains are really needed at the end of a segment because the routing decisions are made at each intersection. So, the node carrying a packet calculates the density of the segment from which it has received the CDP and stores black listed nodes. After the reception of the different CDPs from all underlying segments, a weight is calculated for each one by combining the white nodes density included in the packet and the distance of its next intersection from the destination (equation 4.3). Finally, the road with the highest weight value is chosen to forward the packet through it until reaching its end to repeat the same process of intersections weighting.

$$Weight(j) = \alpha * f(D_k) + (1-\alpha) * g(T_k) \tag{4.3}$$

$f(D_k)$ symbolizes the distance function of an intersection $k$ from the destination of a packet, $g(T_k)$ is the density function of the segment calculated based on information received in a CDP packet that has traversed the segment and $\alpha$ is a weighting factor to balance the impact of each criteria on the routing decision. For the distance that separates the candidate intersection from the destination node, various methods can be deployed to calculate it which are either based on the geometric shape of roads using the Dijkstra algorithm which consumes more time or based on the direct distance between coordinates ($D_{k,d} = \sqrt{(x_d - x_k)^2 + (y_d - y_k)^2}$, where $(x_d, y_d)$ and $(x_k, y_k)$ are respectively the destination and intersection coordinates). In our protocol, we consider to use the second method for distance calculation to decrease the time consumed in the decision process. Finally $f(D_k)$ and $g(T_k)$ are calculated in the same manner as in GyTAR based on equations 4.4 and 4.5. $D_p$ is the fraction between the distance of the candidate intersection to the destination and the actual one which represents the closeness of a point to a destination ($D_p = \frac{D_{k,d}}{D_{h,d}}$), $\sigma = \sqrt{\frac{1}{N_c} * \sum_{i=1}^{N_c} (N_i - N_{avg})^2}$ is the standard deviation of clusters density, $N_{avg} = \frac{1}{N_c} * \sum_{i=1}^{N_c} N_i$ is the average number of trustworthy nodes per cluster ($N_c$ is the number of clusters per segment and $N_i$ represents the number of trustworthy nodes per cluster), and $N_{con}$ represents the ideal connectivity degree in a cluster and will be set to 12.

$$f(D_k) = 1 - D_p \tag{4.4}$$

$$g(T_k) = min(\frac{1}{\sigma + 1} * \frac{N_{avg}}{N_{con}}, 1) \tag{4.5}$$

In the selected segment, a packet is routed based on a greedy forwarding technique where the selection of a forwarder is based on the prediction of the next position of neighbors using their speeds, headings and directions. So, the closest node to the next intersection of the segment is chosen based on this predicted position. A recovery strategy using the technique of store and forward (SNF) is also provided to face intermittent connections and local optimums. The pseudo-code of the greedy routing technique along a segment together with the position prediction algorithm are highlighted successively in algorithms 3 and 4.

**Algorithm 3** Pseudo-code of the greedy routing along a road segment

**Input:** Set of $N$ neighbors with their positions $p_i$, speed $v_i$, heading $h_i$ and time $t_i$, empty set $PP$ to store predicted positions, information of the packet carrier $c$ $(p_c, v_c, h_c, t_c)$ and a forwarder $N_f = 0$.
**Output:** A chosen neighbor $N_f$ as a next packet forwarder.
**for** $i = 1..N$ **do**
    Predict neighbor position ($p_i$, $v_i$, $h_i$, $t_i$);
    Store predicted position in $PP(i)$;
**end for**
Predict carrier position ($p_c$, $v_c$, $h_c$, $t_c$);
Get distance to next intersection $d_{PPc}$ of $PP(c)$;
**for** $j = 1..N$ **do**
    Get distance to next intersection $d_{PPj}$ of $PP(j)$;
    **if** $d_{PPc} > d_{PPj}$ **then**
        $N_f = j$;
    **end if**
**end for**
**if** $N_f == 0$ **then**
    $N_f = c$ (Store and Forward mode)
**else**
    Forward packet to $N_f$;
**end if**

**Algorithm 4** Pseudo-code of the position prediction algorithm

**Input:** A neighbor $i$ with information about its position, speed, heading and time $(p_i, v_i, h_i, t_i)$
**Output:** A predicted position of neighbor $i$ $PP(i)$.
$\delta t = Now - t$ (time in seconds);
**if** $v_i == 0$ **then**
    $PP(i) = p_i$;
**else**
    **if** $h_i \geq 0$ && $h_i \leq 360$ **then**
        **if** $\cos(h_i * \Pi/180) \geq 0$ **then**
            $PP(i).x = p_i.x + v_i.x * \delta t$ ;
        **else**
            $PP(i).x = p_i.x - v_i.x * \delta t$;
        **end if**
        **if** $\sin(h_i * \Pi/180) \geq 0$ **then**
            $PP(i).y = p_i.y + v_i.y * \delta t$;
        **else**
            $PP(i).y = p_i.y - v_i.y * \delta t$;
        **end if**
    **end if**
**end if**

### 4.3.2/ PERFORMANCE EVALUATION

In this section, we present the performance study conducted on our secure intersection-based routing protocol. In the first subsection, we introduce the simulation environment and metrics deployed

for validation. Then, we devote the second subsection to highlight and discuss the different results in hand.

#### 4.3.2.1/ SIMULATION ENVIRONMENT

We implement our approach using NS3.17[1] simulator and we conduct simulations in a Manhattan grid area of size 3000*3000$m^2$, composed of 9 intersections with a segment length of $1000m$. The different simulation parameters are summarized in the table 4.1. For the other parameters related to the calculation of trust level, we have chosen to fix: $\lambda = 0.5$ to make equitable the impact of previous and actual evaluations and d=0.5 while considering applications that did not require a high security level. We firstly, analyze the capability of our proposed mechanism to detect malicious nodes where we vary the number of malicious nodes in the network and see its impact on the detection ratio of the protocol, then, we highlight its impact on the end-to-end communication delay in a malicious environment and its generated overhead for the monitoring and informing processes.

| **Parameter** | **Value** |
|---|---|
| Simulation area | 3000*3000$m^2$ |
| Simulation time | 400s |
| Road length | 1000m |
| Number of vehicles | 100 - 400 |
| Speed | 30 - 50 Km/h |
| Radio Range | 250m |
| Monitoring period | 5s |
| Pre-processing period | 20s |
| Propagation loss model | Two-Ray Ground |
| Propagation delay model | Constant Speed |
| Malicious vehicles ratio | 10% - 40% |

Table 4.1 – S-GyTAR Simulation parameters

#### 4.3.2.2/ SIMULATION RESULTS & ANALYSIS

In this section we present the simulation results and analyze them in order to prove the performances of the routing protocol. We vary the number of malicious nodes between $10\%$ and $40\%$ for various densities (number of nodes between 100 and 400) and we analyze its impact on the capability of the monitoring technique to detect malicious vehicles. Figure 4.4 highlights the detection rate variation of our proposed mechanism when the number of vehicles increases. Results show that our designed intrusion detection system (IDS) is able to detect all the malicious nodes when they are a little minority in the network (10% to 20%). However, its performance decreases slightly when this number increases (30% and 40%), but it still stay capable of detecting more than 92% of the malicious nodes which is a very reasonable and promising result when the number of vehicles in the network reaches 400. This decreasing in the detection rate is due to the tendency of malicious nodes to build false information and condemn normal nodes when their number becomes very important in the network which makes difficult for CHs to differentiate and get the right

[1] http://www.nsnam.org

decisions because the majority of information will come from malicious vehicles. But, in general, our proposed monitoring and malicious nodes detection technique shows a very good accuracy in detecting DoS attacks because when the number of malicious vehicles exceeds 40% the network should be subsequently quarantined and no more considered trustworthy for potential use by deployed applications.

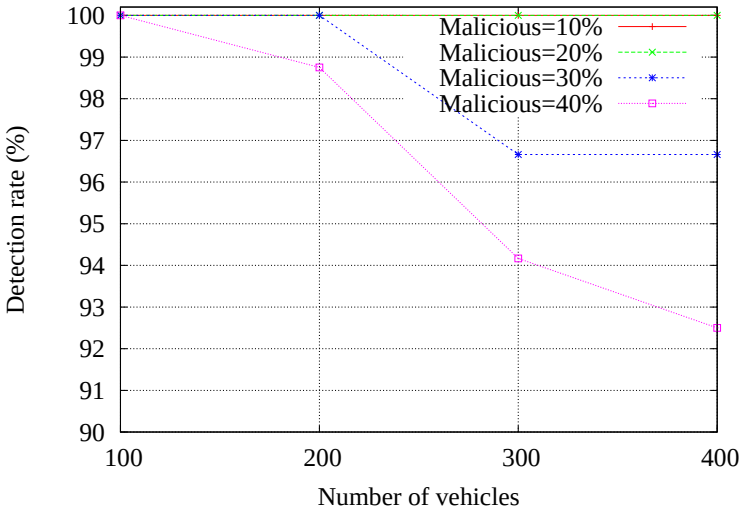



Figure 4.4 – The Detection rate

In figure 4.5, we analyze the overhead generated by the proposed monitoring technique compared to the basic routing protocol. In fact, we do not introduce any new packets to monitor and inform nodes about black lists. The monitoring is done in a promiscuous mode where the CH is continuously listening to its neighbors without any need to exchange more messages with them to count their reputations and update their locally built trust levels. Therefore, the only generated overhead is due to the introduction of black lists in the CDP packet and their update by each CH. So this overhead is relative to the number of detected malicious nodes. For this reason, we consider, in the plot, the worst case when the number of malicious nodes is very high (40% of the network) where the size of the black list would be the biggest one, which leads to the highest overhead. The given results show that our proposed mechanism only adds a little overhead compared to the basic protocol. However, the increase of this overhead with the increase of vehicles density is explained by the fact that the rate of generated control packets is proportional to the number of vehicles in the network.

We, also, study the ability of our monitoring and routing technique to limit the inundation of the network with packets coming from malicious vehicles as a malicious node tries to overload the bandwidth in a large region by flooding its packets in $k$-hops to increase the end-to-end communication delay. So, this could impact the delivery of some packets as they are dropped at different layers if they are not received before a certain time limit. Figure 4.6 shows the difference between end-to-end communication delays variations using a basic GyTAR (B-GyTAR) and our secure GyTAR (S-GyTAR) with the presence of 10% of attackers over the network. The given results prove the effectiveness of our proposed mechanism regarding the limitation of the end-to-end delay. Therefore, the mechanism stops the forwarding of packets issued from a detected malicious node which minimizes the overload of the bandwidth from $k$-hops to one hop and by consequence decreases the end-to-end communication delay.

We can state that the proposed intersection-based routing protocol, which is enabled with a mon-

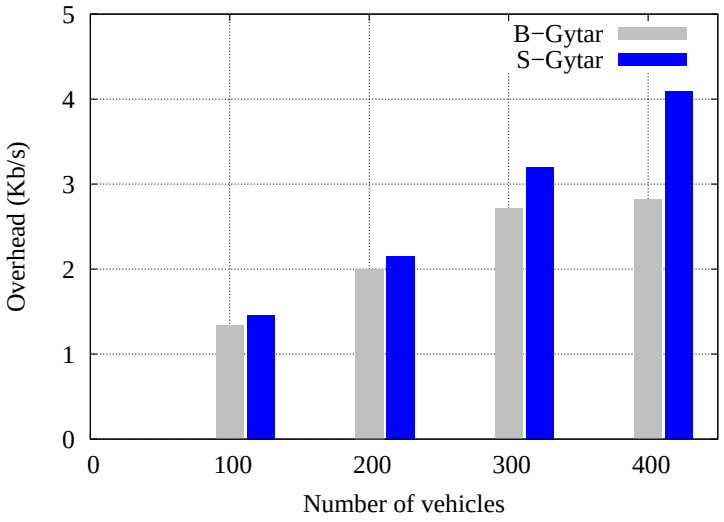



Figure 4.5 – The generated overhead

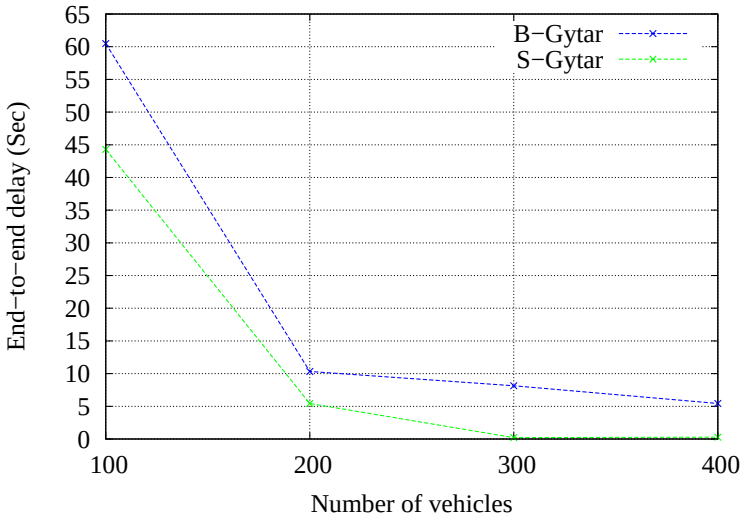



Figure 4.6 – The end-to-end communication delay

itoring mechanism able to continuously control vehicles behavior and detect malicious attacks, increases the security of the data collection process and protects applications from possible malfunctions or disconnections (decrease the end-to-end communication delay). It is completely distributed without relying on deployed infrastructure based on a mutual monitoring and judgment between mobile nodes, which ensure its processing in sparse networks and encourage selfish vehicles to cooperate to have their data relayed.

The major weakness of this protocol together with its intrusion detection system resides in the fact that it is reactive and detects attacks after they happens which may increase the vulnerability window, and its specialization on DoS attacks which keeps doors open for other kinds of attacks. For this reason, we propose in the next section our new intrusion detection and prevention mechanism able to face various kinds of attacks and stop them before threats persist. Indeed, this prediction mechanism is based on a distributed architecture similar to S-GyTAR to monitor nodes, which makes it easy to integrate to the routing protocol to replace its basic intrusion detection mechanism

in order to offer counter measures to face a larger set of attacks with an enhanced prediction capability.

## 4.4/ A DISTRIBUTED DETECTION AND PREVENTION SCHEME FROM MALICIOUS NODES IN VEHICULAR NETWORKS (IPDS)

Despite the huge availability of intrusion detection systems proposed in the vehicular area and variety of the deployed trust models, which were reviewed in section 2.4.1, all of them are similar in the fact that they are reactive mechanisms able to detect an attack after it happens and none of them has the ability to prevent from malicious attacks. This weakness could not be ignored especially in case of applications that require a high security level. For this reason, we propose a new security mechanism namely IPDS to fulfil some vulnerabilities based on the following techniques: (i) the enhancement of the cooperation between all members in the network based on a cluster-based architecture, (ii) the use of the Kalman filtering technique to make accurate predictions about vehicles behavior and (iii) the introduction of a new classification technique to deal with vehicles oscillating their behavior and handle them differently from nodes which attack continuously.

In fact, some proactive mechanisms based on trust modeling are proposed in the literature and are mainly based on either a Hidden Markov Model (HMM) or Kalman filter. In [139], authors consider to treat the problem of trust between agents based on the context of information exchanged between them while proposing a HMM for trust prediction. In fact, a HMM has the ability to find the optimal state for a Markov process given past states/observations. Therefore, authors based their approach on that property to predict the next state of the trust model. For the definition of the considered context, they extract a set of features depending on the application type, calculate their entropy and gain, then combine them using multiple discriminant analysis. This results in an observation probability based on the outcomes of past transactions and associated transformed feature sets. In [47], authors introduce the use of Kalman filtering technique in pervasive systems to autonomously predict trustworthiness of a service provider by a client. The idea is that each client stores the services proposed by a provider with their values of quality, then, based on its previous experience it compares the difference between what is promised and what is really provided and assigns a trust value for each provider to be used afterward for the prediction of its behavior in the next transaction. In our IPDS, we aim to explore the idea of using Kalman filter because it is a lightweight mechanism which does not require big processing resources and offers the possibility of parameters' regulation based on the given outputs [47]. For this reason, we give a short overview about the Kalman filter in the first subsection before detailing the IPDS architecture and its different components in the second one.

### 4.4.1/ KALMAN FILTER OVERVIEW

Kalman filter is mainly based on a set of recursive mathematical functions able to provide an optimal way to estimate the current state of a dynamic system starting from observations that may contain some errors due to the lack of accuracy in the measures provided by connected sensors[47]. Kalman filter is the best linear estimator especially in the case that the introduced noise is Gaussian because it minimizes the mean square error of estimated parameters. To simplify the understanding of Kalman filter, we consider a mono-dimensional system with a state $x \in IR^n (n = 1)$ and governed by equation 4.6, where $x_{t+1}$ is the state of the system at time t+1 given by adding a random Gaus-

sian noise $V_t$ to the previously calculated state $x_t$ at time t.

$$x_{t+1} = x_t + V_t, t = 1,2,3... \tag{4.6}$$

To calculate the state of the system at time t+1, we need to introduce a kind of observations $y_t$ that will be periodically made at each time t. But these observations are also subject to a Gaussian noise and depends on the actual state of the system (Equation 4.7).

$$y_t = x_t + W_t, t = 1,2,3... \tag{4.7}$$

To determine the best estimate of the next system state, Kalman filter combines the actual known state with the noisy measured observations under the assumption that noises are Gaussian with covariances $Q_t$ and $G_t$ consecutively to result these equations (4.8 and 4.9):

$$x_{t+1} = x_t + \omega_t/(\omega_t + G_t) * (y_t - x_t) \tag{4.8}$$

$$\omega_{t+1} = \omega_t + Q_t - \omega_t^2/(\omega_t + G_t) \tag{4.9}$$

By looking at equation 4.8, we can notice that a prediction for the state of the system at t+1 is given by the previously predicted value $x_t$ at t augmented by a term proportional to the difference between the prediction and its relative observation given that $\omega_0 = E[(y_0 - x_0)^2]$. We can observe from equation 4.8 and equation 4.9 that the impact of noise on the weight of each term in the estimated value is crucial. If the signal of the noise imposed to the observation ($G_t$) is high, the impact of this latter is lower than previous estimations and its impact increases when the noise decreases. The same impact could be seen for the noise imposed to the estimate which either increases the estimation impact on the prediction (low $Q_t$ ) or decreases it (high $Q_t$).

### 4.4.2/ NETWORK MONITORING & BEHAVIOR PREDICTION IN IPDS

As mentioned previously (section 2.4.1), network and application attacks, mainly, target the network communication's capabilities and running applications resulting losses in both client and service provider sides. However, malicious vehicles may attack in a continuous manner or only for a limited period then return to the normal behavior due to some internal disfunctions in hard or softwares processing or communication systems. In the reviewed related works, all of them are treated in the same way and classified malicious which is not very reasonable because a node will be condemned forever due to a short period of attack or unintentional misbehavior and automatically evicted from the network. For this reason, vehicles which attack only for short periods should be given a chance before being evicted especially when the traffic density is low and there is a need for vehicles to help in the routing process for example. Therefore, a new distributed technique able to periodically monitor the network members and predict their behaviors is proposed while treating the two kinds of attackers differently. We assume, in our proposition, that all vehicles in the network are equipped with the necessary cryptographic material (such as elliptic curve cryptography in the standard IEEE1609.2) to ensure messages authentication and signing. So, the proposed mechanism uses existing communication and security standards and enhances them with new capabilities. It is based on a clustered hierarchy where a cluster head observes its neighbors behavior based on a Kalman filtering prediction, identifies future attackers and alerts other nodes in the network. However, to make a Kalman filter-based mechanism give good predictions at the output, we need two kinds of information to be introduced at the input. The first entry is a kind of observations made periodically by the system to verify the estimated value of the Kalman filter and which will be the

trust level calculated by monitoring agents. The trust value calculated after an estimation serves in one hand as a proof for the prediction validity and improve the future results (at time t+1) of the Kalman filter by adjusting its parameters and introducing the impact of real behaviors in the other hand. The second entry is the previously predicted information. In the following, we detail the monitoring architecture with the proposed trust models and highlight the used technique for behavior prediction each in a subsection.

#### 4.4.2.1/ MONITORING ARCHITECTURE

The proposed intrusion detection and prevention system (IPDS) is based on a completely distributed architecture to monitor the network periodically and continuously, predict the vehicles' behavior and detect malicious ones. The IPDS organizes vehicles in the network into one-hop clusters and classifies them into three categories: (i) Cluster head (CH) which is the decision maker, (ii) Recommenders which are also referred as monitor agents, specific nodes chosen by the CH to collect trusts about nodes in their vicinity and, finally, (iii) Normal vehicles which are the cluster members or monitored nodes. The cluster head has two major roles: it monitors its one-hop neighbors, keeps tracks of their behaviors and trustworthiness to evaluate their future reactions using a Kalman filter and, also, chooses recommenders to be its monitor agents in the cluster under its responsibility.

The organization of our monitoring architecture passes through two different steps: the first involves the CH election and maintenance which are described in the following and the second one is the recommenders designation which will be explained later on in subsection 4.4.2.4. In fact, we employ in our IPDS a new more flexible clustering method different from the one used in S-GyTAR because clusters in S-GyTAR are static which limits their lifetime and engenders a frequent clustering and CH election processes, thus causing more overhead.

**(i) Bootstrapping phase:** At the beginning of the network organization and vehicles categorization, we assume that nodes are authenticated and certificates are distributed. Nevertheless, the possession of a certificate does not guarantee that its holder will not misbehave as indicated before. Therefore, each node in the network has to keep listening to the traffic of its neighbors and gathers information to get initial knowledge about them and evaluates their trust levels. So, at the end of this phase, a trust value is assigned to each node by its neighbors. The bootstrapping phase is necessary to make nodes know about their neighbors, build trust values and identify potential use of fake identity in the election process of the CH. This phase should last for a period of time which depends on various parameters related to the network namely its density, vehicles' communication capabilities and authentication time.

**(ii) CH election phase:** After the bootstrapping phase, each vehicle knows about the trust levels of others in its vicinity. Therefore, a CH election process is initiated and each node builds a message (CHEAD_MESSAGE (IP Address, Trust Value)) in which it introduces the address of the neighbor who has the highest trust value with its trust level and diffuses it in its radio range. At the reception of a CHEAD_MESSAGE, one node compares the received trust value with the value it has and changes its previous address and trust value of CH to the new received ones if the newly received trust is greater than the local one and ignores the message if it is not the case. Figure 4.7(a) highlights the cluster-head election process.

**(iii) CH maintenance:** The cluster head maintenance process is managed by the CH before getting out from the cluster. In fact, the CH periodically calculates the distances separating it from its cluster members, whenever these distances are higher than a threshold (variable parameter which could be the radio range) it has to designate a new CH. The CH, thereafter, sends the address and trust value of the trustworthiest vehicle it has monitored along its leading period to every cluster member.

Cluster members update and store the new cluster-head address with its trust level. Figure 4.7(b) gives an overview about the CH update.

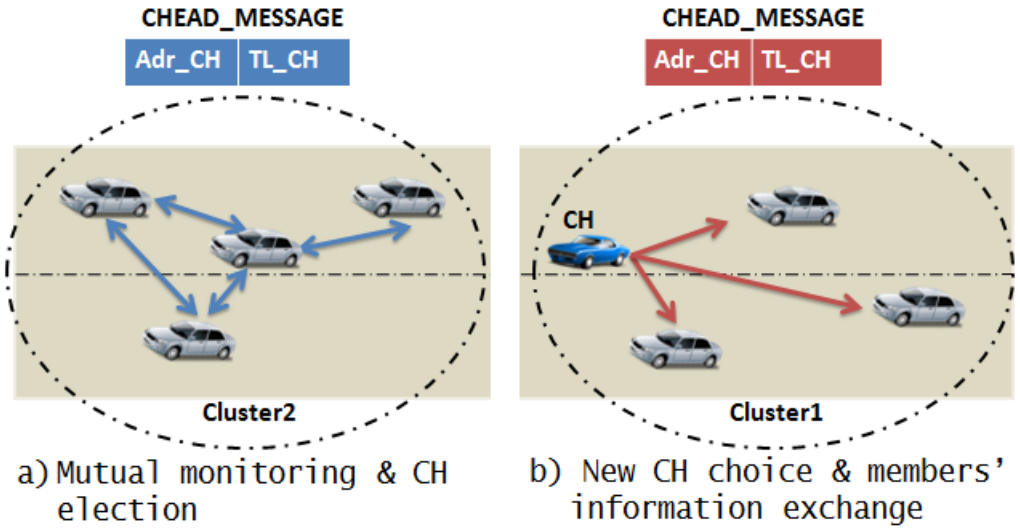



Figure 4.7 – Cluster-head election and maintenance in IPDS

As defined above, a Kalman filter is based on previous estimations and periodic observations made by the system. In our case, we are interested to predict the trust level of a neighbor vehicle. So, the CH should collect observations from its environment about all trust levels of vehicles in its vicinity based on direct and indirect interactions with them. These observations could be of two types: (i) the first is the experience-based trust, which is the only type collected by a recommender based on direct interactions with monitored nodes, and (ii) the second represents recommendations received by the CH from its monitor agents. Unlike recommenders, a CH has to combine these two types of trusts to make decisions. So, the total trust $T_{ij}$ defined to be used as an observation built by a cluster-head $i$ towards a member $j$ is given by equation 4.10 where $T_{ij}^{R}$ is the mean value of trusts received from relative recommenders, $T_{ij}^{P}$ represents the CH's experience-based trust and $\alpha$ is a weighting factor used to balance the impact of received recommendations on the local knowledge of the CH.

$$T_{ij} = \alpha * T_{ij}^{P} + (1 - \alpha) * T_{ij}^{R} \tag{4.10}$$

The value of $\alpha$ depends on recommenders accuracy and shouldn't be fixed to one value all the time. Therefore, its value has to be calculated and updated after each evaluation. So, $\alpha$ increases when the average value of recommenders' trusts increases to boost their impact on the decision and decreases when their average value decreases to favor the experience of the CH. The experience-based trust calculation is detailed in the next subsection followed by an investigation of the technique to calculate recommendation-based trusts in subsection 4.4.2.3.

#### 4.4.2.2/ EXPERIENCE-BASED TRUST

While moving in the network, a vehicle has the possibility to interact and exchange information with every neighbor it has. Therefore, it is able to build a local knowledge about its one-hop neighbors by monitoring their links in a promiscuous mode. This latter allows a node to hear continuously all generated traffic by each neighbor. In our actual work, we are interested to the most dangerous

attacks that may engender multiple damages in materials or human lives. Among these, we focus on Denial of Service, Sybil attack, false alert and packet alteration to develop a technique to verify all of them based on direct interactions between nodes and build an experience-based trust. In the following we describe the policy to build and update the trust relative to each attack type. Therefore, in the verification process, we verify successively attack by attack before to update the trust value. So, whenever a malicious vehicle initiates a combined attack (e.g. Sybil and false alert), the proposed IPDS is able to detect this. In fact, when an attack is detected, it continues to verify the other attacks before updating the trust level of the vehicle in question. This successive verification of all the attacks guarantees a high efficiency level and eliminates the possibility of cheating the system.

**1- DoS Attack:** The DoS covers different types of attacks such as resource exhaustion where the attacker jams the communication medium in the network, black hole and selective forwarding where the malicious node decides to not forward all received packets or a part of them. In such kinds of attacks, we should concentrate on the generated traffic and packets passing by the CH and its monitor agents. In fact, the CH/Recommender analyzes every communication in its vicinity, captures and keeps tracks of all received and sent packets to see if the monitored nodes are not maliciously behaving (cooperating or not abusing of bandwidth use). The packet delivery ratio (PDR) is the base of our calculation and decision process to update the reputation values of neighbors. Therefore, the time axis is divided into equal time slots during each slot, the CH/Recommender calculates the PDR of all neighbors it has. At the end of each time slot, an average PDR (Equation 4.12) is calculated and all the PDRs of neighbors are compared to that value to see if a vehicle is normal or not. In fact, to determine the degree of trustworthiness of such a node as well as its efficiency, a reputation value is set and updated after each time slot based on equation 4.11 .

$$R_{ij}^{n} = \left\{ \begin{array}{rl} \lambda * R_{ij}^{n-1} + (1-\lambda) * r_{i,j}^{n} & \text{if} \quad n > 1 \\ r_{i,j}^{n} & \text{if} \quad n = 1 \end{array} \right\} \tag{4.11}$$

Where $R_{ij}^{n}$ is a reputation value associated to node $j$ by $i$ after n evaluations, $r_{i,j}^{n}$ that could take one of the two values:1 if the evaluated node is cooperating and -1 in the opposite case as seen from the $i^{th}$ point of view following equation 4.13, and $\lambda$ is a weighting factor.

$$pdr_{avg} = \sum_{j=1}^{m} pdr_j / m \tag{4.12}$$

$$r_{i,j}^{n} = \left\{ \begin{array}{rll} 1 & \text{if} & |pdr_{avg} - pdr_j| < pdr_{th} \\ -1 & \text{else} & \end{array} \right\} \tag{4.13}$$

The $pdr_{avg}$ is the average value of the packet delivery ratio, $pdr_j$ is the ratio of the $j$-th neighbor of the $i$-th CH/Recommender and $pdr_{th}$ is a threshold that allows to bound the nodes' delivery ratio between two critical values (Min and Max). The use of a $pdr_{th}$ is necessary to prevent vehicles from overloading the network by an excessive forwarding and also motivate them to forward the received packets to their destinations. Because its value depends on network parameters such as the average packet delivery ratio and number of neighbors accessing the channel ($pdr_{th}$ calculation is detailed in subsection 4.4.2.5), it allows to detect the selective forwarding and black hole attacks as the packet delivery ratio of each vehicle should respect the indicated limits.

The reputation $R_{ij}^{n}$ is used to get the trust value relative to each node $j$ as seen by $i$. The trust denoted $T_{ij}^{p}$, which reflects the experience-based trustworthiness of node $j$, is given by equation 4.14 and varies between 0 and 1 where $n$ is the $n$-th time slot (or evaluation).

$$T_{ij}^{p} = Max\{R_{ij}^{n}/n, 0\} \tag{4.14}$$

Trusts, calculated by each recommender, are periodically sent to the cluster-head where a decision about the behaviors of vehicles is made.

**2- False alert and packet alteration:** These kinds of attacks are classified among the most dangerous ones especially when they are related to safety applications. To detect the malicious node in these two cases, the CH and recommenders have to verify every disseminated alert to confirm the trustworthiness of the sender. The verification is done based on the coordinates $(x_a, y_a)$, time of alert $t_a$ and node velocity at the alert time $Vel_a$ indicated by the message. Usually, when a vehicle detects an accident or any other abnormal activity in the road, it should reduce its velocity and/or change its moving line before sending an alert to others to inform them. For this reason, a comparison between the real velocity of the vehicle and its diffused one in the alert is crucial to verify the exactness of an alert. Therefore, after receiving an alert from a direct neighbor, the monitor agent waits for the next hello message sent by the same neighbor containing the new coordinates $(x_h, y_h)$, time $t_h$ and $Vel_h$ to compute the real velocity based on equation 4.15 and compare it with the previously indicated one in the alert, where $d = \sqrt{(x_h - x_a)^2 + (y_h - y_a)^2}$ is the traveled distance between the alert and hello time.

$$Vel_v = d/(t_h - t_a) \tag{4.15}$$

So, if the condition described by 4.16 is not verified the node should be considered as a suspected attacker and its trust level is decreased.

$$RealAlert \Leftrightarrow Vel_v \leq Vel_a \& Vel_h \leq Vel_a \tag{4.16}$$

The reputation of the node as well as its trust level are, thereafter, updated based on previously described equations 4.11 and 4.14 with an $r_{i,j}^{n}$ that takes 1 if condition 4.16 is verified and -1 in the opposite case.

**3- Sybil attack:** A Sybil attacker as defined in section 2.4.1 sends multiple beacons pretending the existence of a fake vehicle in different places. To face this kind of attack, each node has to verify the real existence of every neighbor it has. Several works were reviewed in section 2.4.1 to face this attack due to its impact on the network consistency. Among proposed techniques, we cite the use of received signal strength (RSS), radio channel controlling and neighbor information. Although they are widely used to face a Sybil attack, some studies [83] demonstrate that the RSS-based Sybil attack detection has some limitations and the use of neighbor information technique is more promising in the detection process. Therefore, we aim to deploy a neighboring information technique to assess nodes' trustworthiness and build their trust levels. For this reason, we extend the periodically exchanged beacons to support neighbors identities of the sender. Basically in a beacon, vehicles introduce their positions, speeds and generation time, but in our case each node has to identify all the neighbors in its radio range and introduce their identities in the message before to send it. Neighbors identities in the beacon message are important for the monitor agent in the Sybil attack detection process. So, a CH/Recommender has to store the received identities from vehicles in its vicinity, then compares their neighbors to identify fake identities created by a potential Sybil attacker. In fact, the main idea is that two nodes should not have the same neighbor for a well defined period of time.

To ease understanding the trust level building mechanism, we present a simple scenario that includes an agent vehicle $M_k$ and two of its neighbors from which it receives beacons $V_i$ and $V_j$. We

assume that $V_i$ declares $h$ neighbors $N_i = (n_i^1, n_i^2, ..., n_i^h)$ in the beacon and $V_j$ declares $l$ neighbors $N_j = (n_j^1, n_j^2, ..., n_j^l)$. After comparing identities of the two neighbors' sets, $M_k$ notices that they have a common neighbor ($n_i^1 = n_j^2$). The identity of the common node $n_i^1$ should be stored in a local table and a backoff timer $\mu_i$ relative to that node should be calculated and triggered. After the expiry of the backoff, the monitor agent $M_k$ verifies if $n_i^1$ is always a common neighbor for the two vehicles to update its trust value. The value of $\mu_i$ is variable and, generally, depends on neighbors' velocities, the width of the common area between their radio ranges and moving directions. Its value is get by equation 4.17 when neighbors are moving in the same direction, while equation 4.18 is used to get its value in the case of an opposite moving direction.

$$\mu_i = d/(vel_j - vel_i) \tag{4.17}$$

$$\mu_i = d/(vel_j + vel_i) \tag{4.18}$$

The $d$ parameter represents the height of the common radio range between neighbors which varies depending on the moving directions. We assume that the velocity $vel_j$ of the vehicle $V_j$ is greater than the velocity $vel_i$ of vehicle $V_i$. To get the value of $d$, $M_k$ has to distinguish between two cases: (i) when two nodes move in the same direction, and (ii) when they are moving in opposite directions. Each defined case is split into two cases in turn where the first describes two nodes moving one in front of or perpendicularly (in intersection where nodes take different directions) to the other and the second concerns the case of a parallel moving (when overtaking). In figure 4.8, we explain the calculation of the $d$ value when neighbors are moving either in parallel or one in front of the other.

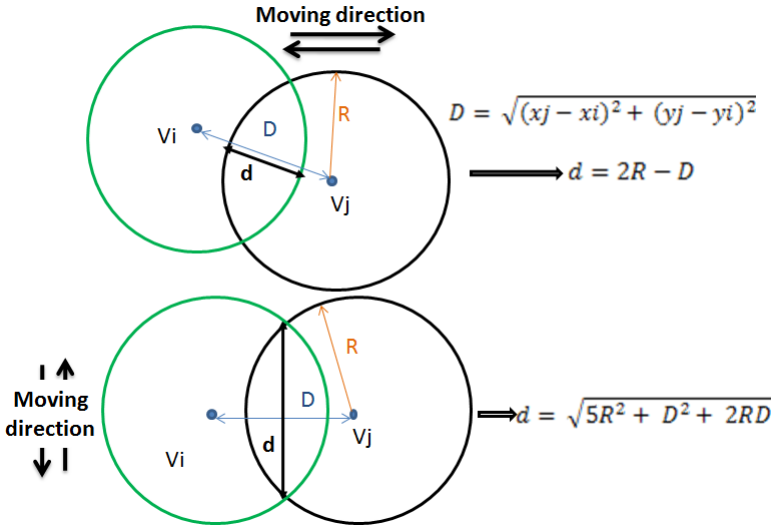



Figure 4.8 – Distance $d$ calculation

After the expiry of the backoff time $\mu_i$, $M_k$ verifies if the neighbors have always $n_1^i$ as a common neighbor. So, if $n_1^i$ is in neighbors' lists of $V_i$ and $V_j$, it should be considered as a Sybil node and its reputation value and trust level are reduced. In the opposite case where the node is no longer a common neighbor, it should be considered trustworthy and its trust value will be increased. The update of the reputation and trust level is made based on equations 4.11 and 4.14 with an $r_{i,j}^n$ taking either 1 in case of a positive evaluation or -1 if negative evaluation.

#### 4.4.2.3/ RECOMMENDATION-BASED TRUST

To make a wider observation and increase the accuracy of vehicles behavior estimation, a CH should collect information about its neighbors and calculate their trust based on the experience-based model if it is directly interacting with them and gather their related trusts from recommenders. The CH calculates, afterward, the recommendations average ($T_{ik}^{R}$) based on equation 4.19 to be combined with the local observed trust before triggering the prediction and classification process.

$$T_{ik}^{R} = \frac{1}{m} * \sum_{j=1}^{m} T_{ij} * T_{jk}^{R} \tag{4.19}$$

Where $T_{jk}^{R}$ is the recommended trust for node $k$ from recommender $j$, $T_{ij}$ is the total trust assigned by the $i$-th CH to the $j$-th recommender and m is the number of recommenders.

#### 4.4.2.4/ RECOMMENDERS' CHOICE POLICY

A malicious node can initiate various types of attacks depending on its goals. It can launch a Denial-of-Service, Sybil, false alert, packet alteration or impersonation attack to disrupt the network functioning, cause roads congestion or even mortal accidents. A CH used alone in the detection process could be easily cheated by attackers especially if there are many of them. So to detect these attacks in an accurate way, a cooperation between the CH and other cluster members is highly recommended. Therefore, we opt for a mutual monitoring concept where a number of nodes in the network called $recommenders$ monitor their one-hop neighbors in a promiscuous mode based on previously described rules and periodically communicate their experience-based trust levels, which are identified as $recommendations$, to the CH (decision node) to be used in the decision and classification process. In fact, there exist two different architectures that could be deployed to designate recommenders in a vehicular network. The easiest way is to enable the monitoring in all vehicles and make all of them recommenders. However, this strategy engenders a big overhead which may lead to scalability problems. Figure 4.9 highlights the overhead evolution depending on the number of recommenders in the network. This latter increases with the number of recommenders, which leads to a traffic congestion and bandwidth overload with control packets. For this reason, we have to maintain a trade-off between the detection rate of our prediction technique and its generated overhead. The second technique that could be deployed is able to decrease the control packet generation by minimizing the number of vehicles sending recommendations. It is based on a specific choice of a reduced number of recommenders in the network. So, to release a lightweight prediction mechanism which alleviate the exchange of information and decrease overhead introduced by control packets, we have chosen to deploy the second architecture and to not enable the monitoring in all cluster members. However, the number of recommenders and their localizations should be carefully chosen to cover all one-hop neighbors in a cluster which needs a specific strategy.

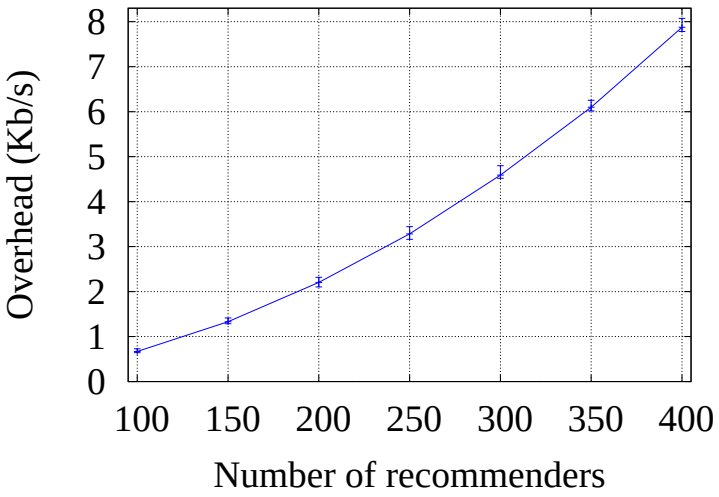
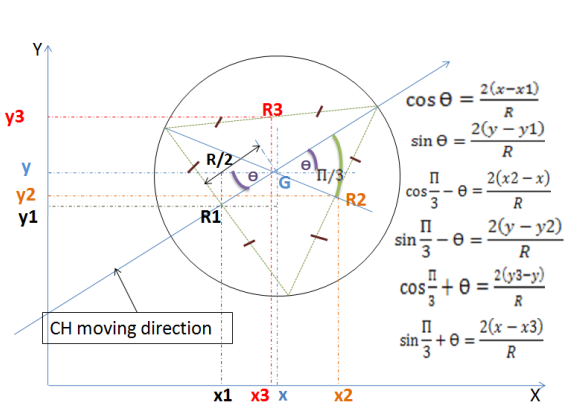
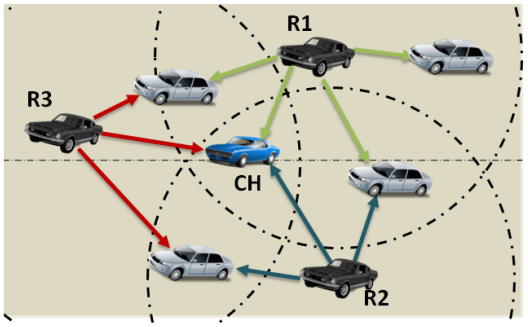



Figure 4.9 – Recommendations overhead evolution

In fact, recommenders' choice is delegated to cluster heads where each one is in charge of the designation of three vehicles from its radio range which cover all its one-hop neighbors. To achieve this job, a CH has to divide its range perimeter into three equal regions, locate the center of each zone, group its neighbors relative to each region and, thereafter, choose the most trustworthy ones close to each center to designate them as recommenders. Hereafter, we describe the choice policies. To simplify things, we assume that a CH is moving on a planetary topology according to the X and Y-axes with a stationary Z-coordinate. We, also, suppose that the recommenders' designation process is always happening when vehicles are moving in a linear portion of the road and not in intersections. Let's denote G(x,y), R1(x1,y1), R2(x2,y2) and R3(x3,y3) coordinates of the CH and the centers of the three regions respectively. The CH is moving in a linear line with a known heading θ according to the X-axis and a radio range equal to R (see figure 4.10). It starts calculating R1 coordinates, then R2 and finally R3.




The black vehicles are chosen recommenders (R1, R2 and R3) in the different regions because they are the closest ones to each region center. They monitor their neighbors including the CH

Figure 4.10 – Recommenders choice strategy

According to figure 4.10, we get all coordinates of the regions' centers which will be used as references to recommenders. In the following, we highlight the relative equations used to determine the

different coordinates in function of CH's coordinates and heading.

$$x1 = x - (R/2) * cos\theta \tag{4.20}$$

$$y1 = y - (R/2) * sin\theta \tag{4.21}$$

$$x2 = x + (R/2) * cos(\Pi/3 - \theta) \tag{4.22}$$

$$y2 = y - (R/2) * sin(\Pi/3 - \theta) \tag{4.23}$$

$$x3 = x - (R/2) * sin(\Pi/3 + \theta) \tag{4.24}$$

$$y3 = y + (R/2) * cos(\Pi/3 + \theta) \tag{4.25}$$

After getting regions' centers coordinates, a CH extracts the neighbors coordinates from their relative beacons and compares their distances from each center $d_{vr} = \sqrt{(x_v - x_r)^2 + (y_v - y_r)^2}$ and their relative trust levels previously computed to finally identify the closest one to each center with the highest trust value and designate it to be its recommender. $(x_v, y_v)$ and $(x_r, y_r)$ are respectively the coordinates of any cluster member and its region center. After locating its relative recommenders, the CH has to inform them about their new job by sending a unicast message for each one. The message is called REC(CH_ADR, REC_ADR1, REC_ADR2, TRUE) that contains the relative CH address, addresses of the other chosen recommenders and a boolean value indicating TRUE which means that the recommender is activated. Selected nodes have to monitor their neighbors in each region and send periodically their trusts to their related CH. The CH is also monitored by recommenders to decrease the possibility of behaving maliciously. Figure 4.10 highlights recommenders positions, their activities and how their recommendations are being sent to a CH.

A CH or even a recommender may act normally during a period of time to gain the trust of its neighbors and switch later to a malicious mode which we call a camouflage attack. For this reason we get to the mutual monitoring strategy between the CH and its Recommenders as a solution to face that kind of attackers. So, the CH is always evaluating its related recommenders and if one of them changes its behavior, it will be punished, stored in the black list and replaced by another normal node more trustworthy. Recommenders are also keeping an eye on their CH and evaluating its behavior and whenever they remark that it is suspected, they are able to replace it by another more trustworthy vehicle based on a voting mechanism as described in the following (see diagrams figure 4.11).

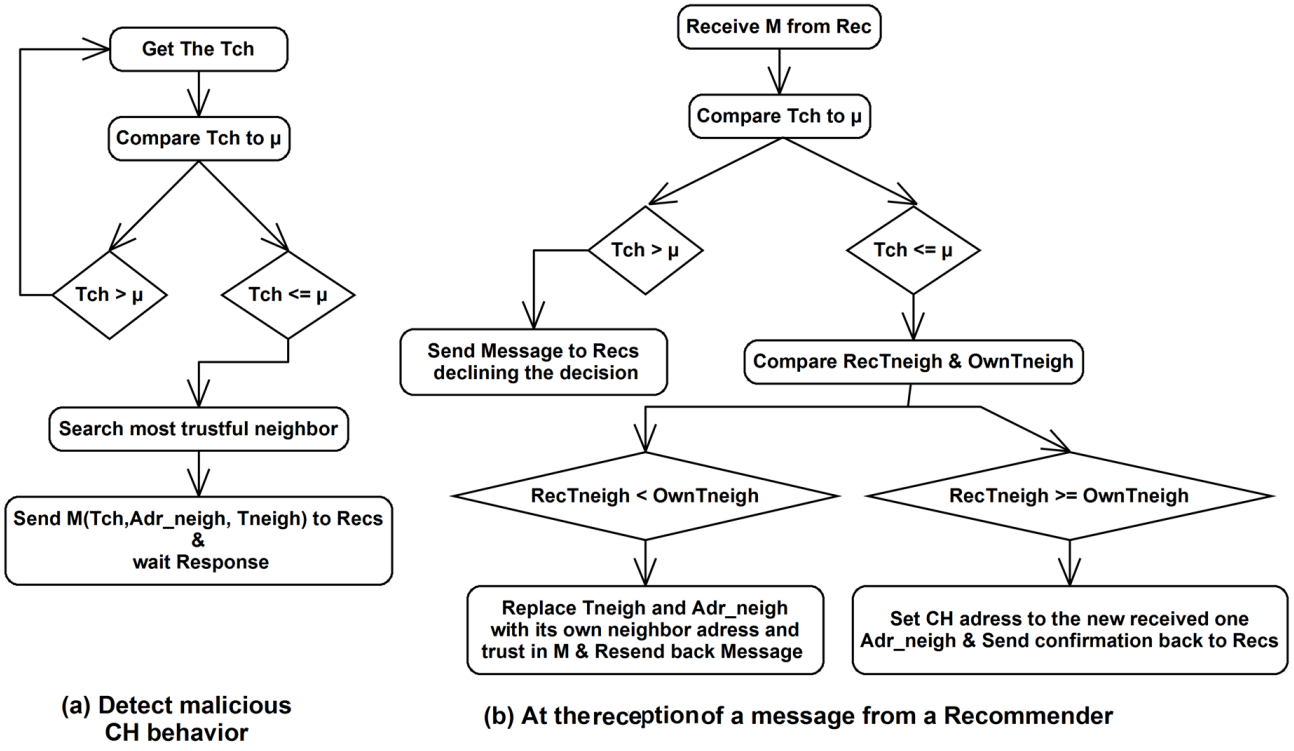



Figure 4.11 – CH Malicious behavior detection and replacement

A recommender can either detect a malicious behavior of its CH by itself based on process described in figure 4.11 (a) or receive an information from other recommenders about that. In fact, it continuously compares the trust of the CH ($T_{ch}$) to a $\mu$ threshold (described in subsection 4.4.2.5) and whenever it is below, it has to send a message $M$ to other recommenders (Recs) in a unicast way indicating the $T_{ch}$ it has, the address (Adr_neigh) and trust value of the most trustworthy node it monitors (Tneigh). The recommenders, after receiving that message, may confirm the CH state or decline it based on process in figure 4.11 (b). In the first case, each one searches its most trustworthy monitored node and compares its trust value (OwnTneigh) to the one sent by other recommenders (RecTneigh): if it is higher, it will be introduced in the message with the address of its owner instead of received ones and the message $M$ is sent back, in the opposite case the recommender stores the received address and wait for a confirmation from the last recommender to make it the new CH. In the second case, recommenders send a message indicating that the CH is normal giving their built trust values. If the recommenders choose a new CH, one of them, eventually the first to detect the malicious behavior of the last one, sends a message indicating the identity of the new elected CH to monitor the network.

#### 4.4.2.5/ TRUST LEVEL PREDICTION & CLASSIFICATION

A cluster-head is responsible of the estimation of future behavior of its neighbors and information collection from other nodes in the network about suspected attacks. The prediction process is based on the Kalman filter described in subsection 4.4.1. So, to adapt our problem to the basic Kalman filtering technique, let's consider $P_{ij}$ the prediction of the trust level of node $j$ made by $CH_i$, and for an observation we introduce the trust level value $T_{ij}$ gathered by the same $CH_i$ using the experience and recommendation-based trust models. So, if we introduce our defined parameters for the trust level prediction in equation 4.8, we get equation 4.26 that defines the whole problem.

$$P_{ij}(t+1) = P_{ij}(t) + (\omega(t)/(\omega(t) + G(t))) * (T_{ij}(t) - P_{ij}(t)) \quad (4.26)$$

In some cases, a malicious node uses to behave normally for a long period of time to gain the confidence of its neighbors then starts attacking. For this reason, we have targeted this problem based on parameters' manipulation $G(t)$ and $Q(t)$ to affect the decision of the CH. In fact, a comparison between the real behavior of a vehicle and the prediction made by the CH is periodically done and based on that parameters' values are updated. Therefore, the impact of the last observation is increased by decreasing $G(t)$ value, if the observation made is more reliable or decreased in the opposite case (prediction is more reliable) by decreasing the value of $Q(t)$. A representative schema of the complete Kalman filtering technique is given by figure 4.12.

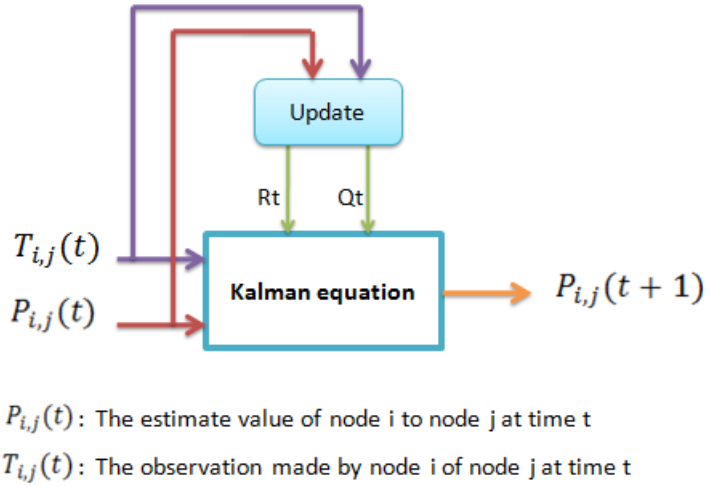



Figure 4.12 – Kalman filter based prediction mechanism

At the end of the prediction process, a CH proceeds to a vehicles classification according to their trust values. So, a behavior is associated to each trust value and vehicles are being classified into three different classes using three predefined lists and two thresholds $\mu$ and $\delta$. These thresholds, like the previously used one for DoS detection $pdr_{th}$, should be carefully chosen to make a good classification and keep a trustworthy environment for VANET applications. For this reason, we enhance their choice using a learning algorithm to be dependent from the network characteristics. There exist various kinds of learning algorithms that are able to give good results such as Support Vector Machine (SVM) [26] and Artificial Neural Network (ANN) [9]. The SVM, based on a training and classification processes takes a list of features as input vector to calculate a set of so called support vectors that allows data classification, while the ANN, inspired from biological nervous systems and based on a large number of interconnected processing elements (neurons) working in unison, is able to solve specific problems and give optimal results. Unlike ANN which gives outputs depending on the treated problem, SVM always gives a binary result classifying data. Therefore, we choose to use an ANN algorithm to get the optimal thresholds to be used in the classification process of the Kalman filtering. In fact, we install in each CH/Recommander a learning algorithm to update its relative parameters. We assume that one node does not need to have a global knowledge about the network characteristics to get optimal results and only local knowledge about its one to k-hop neighbors are sufficient to do the job. Therefore, the monitor vehicle has to know about surrounding malicious vehicles, the packet delivery ratio (PDR) of its neighbors and their trust levels. It computes the average value of the PDR and trust value, then introduces them

with the number of malicious nodes in its black list as inputs to the ANN algorithm. Thereafter, the learning algorithm trains entries to give optimal thresholds a vehicle has to use (outputs of the algorithm). So, the ANN algorithm takes, in fact, three inputs (average PDR, trust value and number of malicious nodes) and generates a set of three outputs ($pdr_{th}$, $\mu$ and $\delta$). The data given within inputs is divided into three sets: training ($70\%$), cross validation ($15\%$) and testing ($15\%$). As shown in [4], one hidden layer in the network can approximate any continuous function and to reduce the processing time and resource consumption, we use only one hidden layer in the network architecture and deploy sigmoidal transfer functions between neurons in hidden and output layers. The training stops when the error (given by the cross validation) drops towards zero.

Vehicles classification done by the CH is based on rules described in the following with thresholds that take values given by the learning ANN algorithm.

**- White list:** If a vehicle has $P_{ij} \in ]\mu, 1]$, it is considered highly trusted and could be safely used to support a VANET application. White vehicles are always prioritized by running applications because they don't change their behavior in the future and continue to cooperate,

**- Gray list:** If the vehicle has $P_{ij} \in ]\delta, \mu]$, it is considered weekly trusted and used in the data exchange required by applications only in case of empty white lists. This list is used to store vehicles that are oscillating their behavior to give them a chance to stop their attacks before being definitely evicted from the network. The gray vehicles are usually controlled and their future behavior is predicted and are thus evicted from the network if they persist in their attacks or return to the white category if they regain their normal behavior,

**- Black list:** This list contains all the vehicles with $P_{ij} \in [0, \delta]$, which means that they are considered malicious and shouldn't be used in application scenarios. Blacklisted vehicles should be definitely evicted from the network.

After the categorization of its neighbors, a CH has to inform other nodes in the network about the blacklisted vehicles using the same informing technique introduced in S-GyTAR. So it builds a so called CDP (Cells Density Packet) packet [87] in which it introduces the black and gray lists and configures the size of the region to inform. A CDP is sent back to other CHs to be forwarded cluster by cluster until reaching the end of the configured region. The size of a region could be defined either by the number of intersections a message has to traverse before to stop or a distance. We introduce this kind of information to decrease the communication overhead and avoid the waste of bandwidth in the network as the size of a CDP keeps growing at each passage by a CH. Each time a vehicle receives a CDP, it updates its black and gray lists. If this message arrives to a CH, it should also update the CDP by adding its black and gray nodes, forwards it to the next CH and informs its neighbors. Figure 4.13 illustrates an example of two adjacent clusters $i$ and $j$. In each cluster, cluster members are monitored by a CH. We can see the information exchange about black and gray lists where each CH collects recommendations from its recommenders and inserts the identities of its gray and black nodes (here $Mk$ & $Gk$ for cluster $i$ and $M'l$ & $G'l$ for cluster $j$) in the CDP before to send it to the other CH.

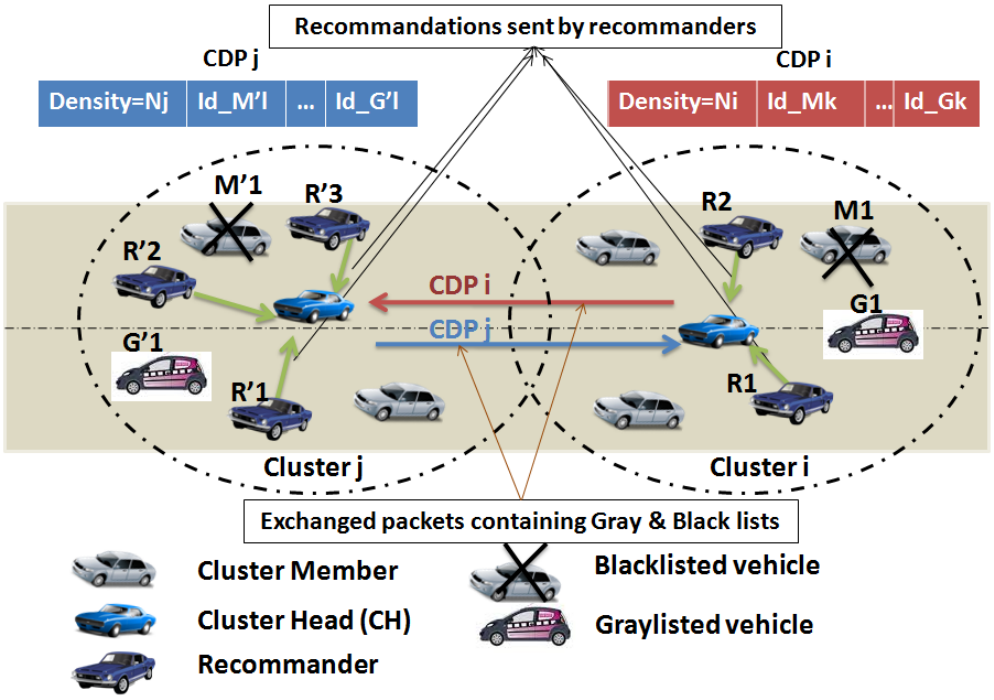



Figure 4.13 – The IPDS monitoring architecture

### 4.4.3/ PERFORMANCE EVALUATION

This section is devoted to, first, study the security analysis of our proposed IPDS and its reaction against some abnormal scenarios, then we highlight the simulation environment and finally, discuss given results.

#### 4.4.3.1/ SECURITY ANALYSIS

This subsection is dedicated to study the security analysis against some abnormal scenarios that may affect the normal operation of IPDS and countermeasures taken by our mechanism to face them.

**- Attacks from external vehicles:** To face such kind of attack we assume, as mentioned above, that vehicles are equipped with security material and are able to communicate securely based on authentications and messages signing. So, an external vehicle is not able to communicate with vehicles in the network without being authenticated and thus unable to decrypt messages without appropriate keys. However, if it succeeds to get a key and initiate a DoS attack, change the packets information or claim to be in a different place it will be identified based on the techniques developed for each attack.

**- Misuse of the CDP packet:** It may happen that a malicious vehicle tries to give false information about the network organization claiming to be the CH by sending a CDP packet in which it introduces fake data about malicious and normal vehicles. However, such kind of unauthorized use of packet is directly detected by recommenders using identity verification of the packet originator because they already know their CH or even by the CH itself. So, the attacker is detected, classified as malicious and evicted from the network.

**- Malicious CH attempting to select malicious recommenders:** A vehicle may act normally during the bootstrapping period and gain trust of its neighbors to be CH. Despite its success to become a CH, the malicious vehicle is not able to designate malicious recommenders to comply with it because its choice is controlled by cluster members. In fact, the malicious CH will try to choose vehicles with lower trust levels to be recommenders based on the idea that they are probably malicious. However, this behavior is considered malicious by cluster members. So, they will decline the choice of the CH and repeat the election process without considering this latter as a candidate.

**- The use of fake identities to announce a high trust value by a malicious vehicle at the CH election process:** This kind of attacks is not allowed in our mechanism because vehicles are mutually monitoring and know about their neighbors which means that all the messages received from a non monitored neighbor are not accepted in the election process.

**- Malicious behavior of a CH or recommender:** This kind of attack is controlled by the continuous mutual monitoring between the CH and its recommenders as designed by our IPDS.

#### 4.4.3.2/ SIMULATION ENVIRONMENT

We implement our approach using the same simulator, NS3.17[2] over our secure routing protocol (S-GyTAR) while replacing its reactive security technique (IDS) by the new IPDS and conduct simulations (10 times for each value) in a Manhattan grid area of size 3000*3000$m^2$ generated using Simulation of Urban Mobility (SUMO) simulator [3]. The main simulations' parameters are summarized in Table 4.2. The monitoring period represents the duration separating two successive evaluations done by a CH which is managed by the user based on its preferences and confidence on the network. In our simulations, we consider two kinds of attackers: the first is a malicious vehicle which continues to attack all the time after changing its behavior and the second one is a node who oscillates its behavior, which means that it will not continue attacking forever but it attacks for only a short period then regains its normal behavior for the rest of the time (represents 20% of the attackers). The kind of attack per node is chosen randomly. The number of attackers in the network varies from 10% to 40%. As in our previous work, we consider here that the maximum number of attackers should not exceed 40% because above that value the network is not considered to be reliable for running applications and should be quarantined [166].

| Parameter | Value |
|---|---|
| Simulation area | 3000*3000$m^2$ |
| Simulation time | 400s |
| Road length | 1000m |
| Number of vehicles | 100 - 400 |
| Vehicles speed | 30 - 50 Km/h |
| Radio Range | 250m |
| Monitoring period | 5s |
| Pre-processing period | 20s |
| Propagation loss model | Two-Ray Ground |
| Propagation delay model | Constant Speed |
| Malicious vehicles ratio | 10% - 40% |

Table 4.2 – IPDS Simulation parameters

[2] http://www.nsnam.org
[3] http://sumo-sim.org

#### 4.4.3.3/ EXPERIMENTAL RESULTS & ANALYSIS

We first, analyze the capability of our proposed intrusion detection and prediction system to detect malicious nodes, then, we highlight its impact on the end-to-end communications delay, delivery ratio and its generated overhead in a malicious environment. We highlight, in figures 4.14 and 4.15, the ability of our proposed mechanism to detect malicious vehicles. We first examine its ability to detect each kind of attack apart by activating only one attack's type in all malicious vehicles (see figure 4.14). Then, we compare its performances to VWCA [112] and DCMD [119] in a highly malicious environment (Figure 4.15) with 40% of malicious vehicles that initiate various kinds of attacks (DoS, Sybil, False alert and Packet alteration). We assume having 10% of malicious vehicles per attack kind. Given results show that the prediction mechanism outperforms VWCA and DCMD in terms of attack detection and exhibits a high detection rate (over 92%). The comparison between our scheme without prediction (IDS) and the one with that technique demonstrates that prediction partially increases its detection capability. However, the aim of this comparison is not to demonstrate the improvement in the detection rate because all of them are conceived to detect efficiently the malicious vehicles, but to prove that IPDS does not loose its detection capabilities despite the prediction errors and the frequently change of vehicles' behavior. These results are achieved thanks to the cooperative messages exchange between monitoring vehicles. Our mechanism encourages cooperative behavior by punishing selfish nodes in the network even cluster heads and recommenders.

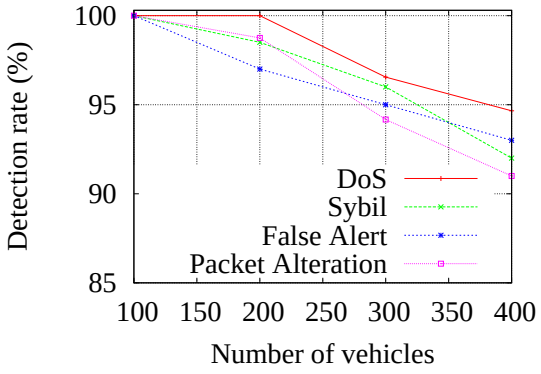



Figure 4.14 – Detection rate vs. Attack type

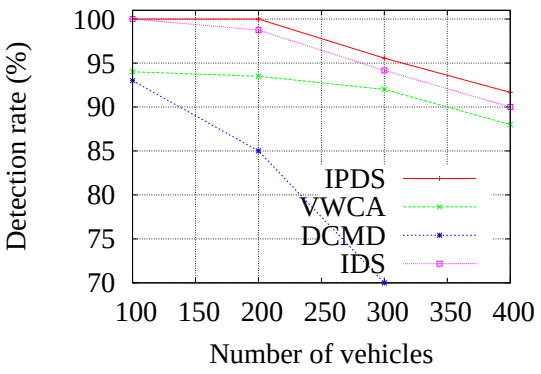



Figure 4.15 – Detection rate

We also make a study to see the impact of our prediction mechanism on data routing and results show that it has a good impact on the packet delivery ratio and end-to-end communication delay. Figures 4.16 and 4.17 demonstrate the difference between the basic routing protocol and the one enhanced with the prediction mechanism in the presence of malicious vehicles. As we can see, the basic protocol suffers from a higher delay and low delivery ratio in a malicious environment as vehicles do not cooperate to forward data and behave selfishly to only forward their own needed information. However, the proposed prediction technique is able to enhance the performances of the protocol while increasing the delivery ratio and decreasing the end-to-end communication delay. Therefore, the proposed schema targets to firstly evict malicious vehicles (Blacklisted vehicles) from routing tables of data carriers which limits their probabilities to be used as packet forwarders and by consequence minimizes the dropped packets. Furthermore, the proposed architecture limits the forwarding of packets issued from black listed nodes which avoid the bandwidth overload, ease the access to channels and decrease the end-to-end delay. However, the higher delay we can see in figure 4.16 when the number of nodes is low ($\leq 200$) is due to the fact that the network is not dense

and distances between vehicles are very high which may reduce the probability of finding a packet forwarder quickly and increase the carrying delays before reaching the destination.

In figures 4.16 and 4.17, IPDS impact on the delivery ratio and end-to-end delay is also compared to the basic IDS (without prediction). Results show that IPDS highly increases the delivery ratio and decreases the delay of the routing protocol compared to IDS. This can be explained by the fact that the prediction mechanism introduces the use of a new list: the gray list. As explained above, this list is used to store vehicles with low trust values which are till now not considered malicious. So, if we look at our attack model, the 20% of vehicles which attack for a small period will be firstly stored in the gray list and then return to the white list when they stop attacking before having a trust value less than $\delta$. These nodes could be used in the routing process when no white vehicle exists in the routing table (case of sparse network). Therefore, the delivery ratio is increased and delay is decreased. Unlike IPDS, the basic IDS is based on a binary categorization to classify vehicles into two classes (malicious and normal), store the two kinds of attackers in the black list and directly evict them from routing tables which dramatically decreases the number of packet forwarders in a highly malicious environment and limits the impact of IDS on the delivery ratio and end-to-end delay. So, the use of the gray list allows to give more chance to attacking vehicles to change their behavior before being evicted forever from the network which is more beneficial for network applications, especially in non-dense networks. Figure 4.18 further proves our reasoning by producing a comparison between the number of forwarders in basic IDS and IPDS. We can see that the number of forwarders (white + gray vehicles) in IPDS is higher than IDS (white vehicles) along the simulation time due to the integration of gray vehicles in the routing process.

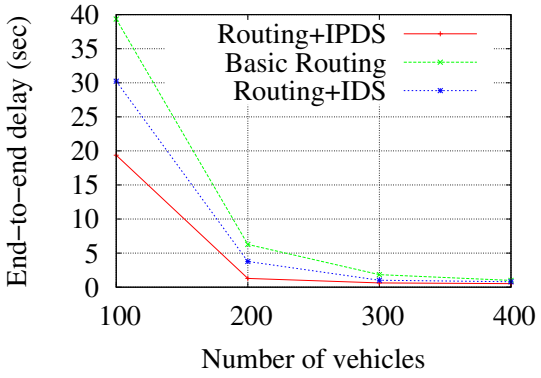



Figure 4.16 – End-to-end communication delay

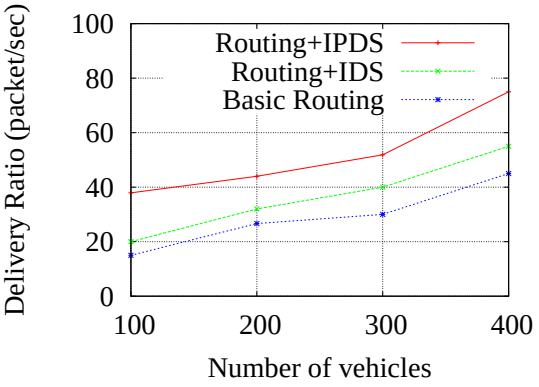



Figure 4.17 – Packet delivery ratio (PDR)

The communication overhead generated by the prediction technique is studied to confirm its higher performance and scalability. Figure 4.19 highlights the difference of overhead between basic routing protocol and secured one. It is clear from the plots that the prediction mechanism we have integrated the protocol does not add a huge amount overhead compared to the basic routing (overhead approximately equal to S-GyTAR). As the monitoring is done in a promiscuous mode, the only overhead sources are control packets and informing process. However, they do not add big overhead to the position-based routing protocol we have used in our tests. Therefore, we conclude that the developed prediction mechanism resists to the scalability problem.

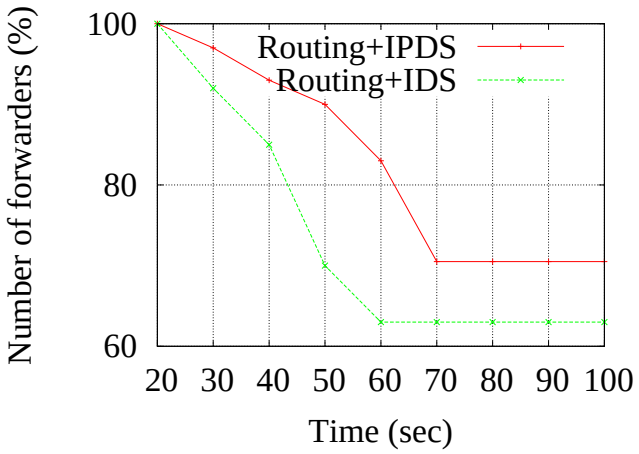



Figure 4.18 – Evolution of the number of forwarders over time with 40% of attackers among 400 vehicles in the network

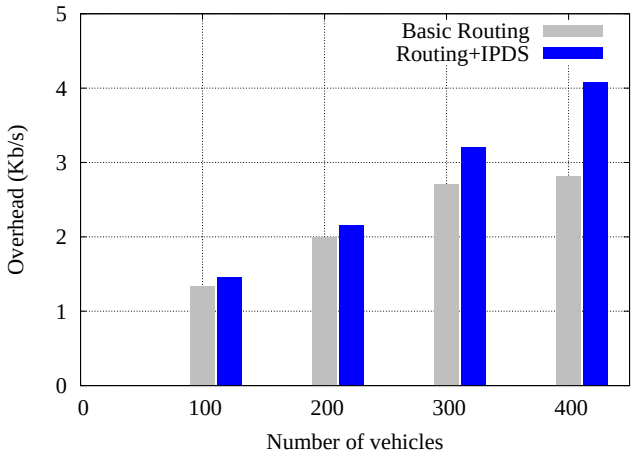



Figure 4.19 – Generated communication overhead

## 4.5/ CONCLUSION

Ensuring a secure and efficient data collection in a vehicular environment is a very challenging field since vehicles are being always connected which make them vulnerable and usually exposed to various attacks. For this reason, we proposed in this chapter two techniques to secure data flowing between vehicles and preserve their integrity to protect running applications. We, first, proposed a secure intersection-based routing protocol which takes traffic information into account, then, we enhanced its capability by designing a new intrusion detection and prevention mechanism which is based on attacks prediction and faces several kinds of them namely network and application attacks. The two proposed security mechanisms are completely distributed based on a hierarchical architecture to continuously monitor vehicles in the network and identify malicious behaviors. In the first proposal, named S-GyTAR, roads are divided into small cells where clusters are organized. The trustworthiest and nearest node to the center of a cell is chosen as a cluster head. Its role is to monitor its neighbors and build a reputation model to calculate their trust levels, classify them into malicious and normal nodes and inform about malicious behaviors. The second proposal, named IPDS, is able to survey the network, detect malicious nodes before they attack based on their trust

levels prediction and deal with oscillating attackers by introducing a new kind of classification using an intermediate list (gray list) to hold vehicles changing their behavior. In this behavior prediction mechanism, we deployed a Kalman filter technique together with two trust models to estimate vehicles behavior and classify them into three classes to evict malicious ones (black list) and keep monitoring others in the gray and white lists. The new classification offers the possibility to use vehicles oscillating their behavior when they are still normal (in the gray list) in the routing process in case of white vehicles' scarcity in the network (case of sparse networks). S-GyTAR and IPDS were developed and tested in a simulation environment and given results prove that they exhibit a high detection rate, low end-to-end delay and high delivery ratio. These results are achieved due to the cooperation enhancement between vehicles and the encouragement of selfish ones to cooperate to protect data. Therefore, data is being highly protected and efficiently delivered to destinations while respecting applications requirements using the two contributions described in this chapter. This data can hence be exploited and analyzed for various aims namely the development of new ITS applications and enhancement of their required quality of service. The next chapter is devoted to develop techniques to exploit and analyze the collected data in a vehicular network.

# 5

# EFFICIENT DATA EXPLOITATION IN A VEHICULAR NETWORK

## 5.1/ INTRODUCTION & PROBLEM STATEMENT

Collected data in a vehicular network are of various use and, mainly, aim to enhance the human daily life and protect the environment based on efficient data exploitation and analysis techniques. In fact, data produced by the embedded sensors and cameras integrated in a vehicle or infrastructure have heterogeneous representations and do not allow the straightforward interpretation of an object or scene for the driver, passenger or decision maker. Furthermore, applications do not have the same requirements and their needs depend on the treated field. Non-delay tolerant applications for example such as ITS-safety and ITS-efficiency require fresh data to be analyzed and presented to drivers in bounded delays, or they get obsolete. For this reason, in some application areas data should be quickly gathered and analyzed to be useful. On the other hand, in some kinds of applications, data are useful even after some time has passed if they are stored in central databases or in in-vehicle recorders (for example location and video recorders). For example, data may be reused by police agents to track outlaw vehicles and in court for crime scenes reconstruction. So, depending on the application to be developed and its requirements data should be treated in a different manner either internally in the vehicle or in a central fixed trustworthy server where a set of services are enabled for potential clients. Figure 5.1 illustrates a set of various application fields that exploit and analyze data to fit their offered applications.

ITS-applications related to vehicle trips, fleet management and diagnostics are of potential necessity due to their economic and environmental impact. In fact, vehicles are being among the first sources of greenhouse gases emission because of the increase of their number all over the world. In addition, a driver traveling without any knowledge about the traffic conditions in the road may be exposed to excessive losses due to time and fuel consumed during a passage by a traffic bottle neck caused by an accident, working zone or even bad weather conditions. For this reason, it is highly recommended to offer services that efficiently exploit and analyze the gathered data from vehicles environment and internal diagnostics to optimize the number of trips, their length and duration. However, offering such reliable services always requires to deal with flowing data characteristics (heterogeneity, incompleteness, redundancy and massive size) at different levels. Data analysis techniques based on data fusion reviewed in subsection 2.3.1 represents a promising technique to efficiently exploit collected data because it splits its processing into different levels (5 levels) depending on its complexity degree and offers several methods to securely fuse it (statistical, probabilistic and artificial intelligence methods) and extract meaningful information of high importance to ITS-applications. So, during this chapter we target to benefit from these techniques to

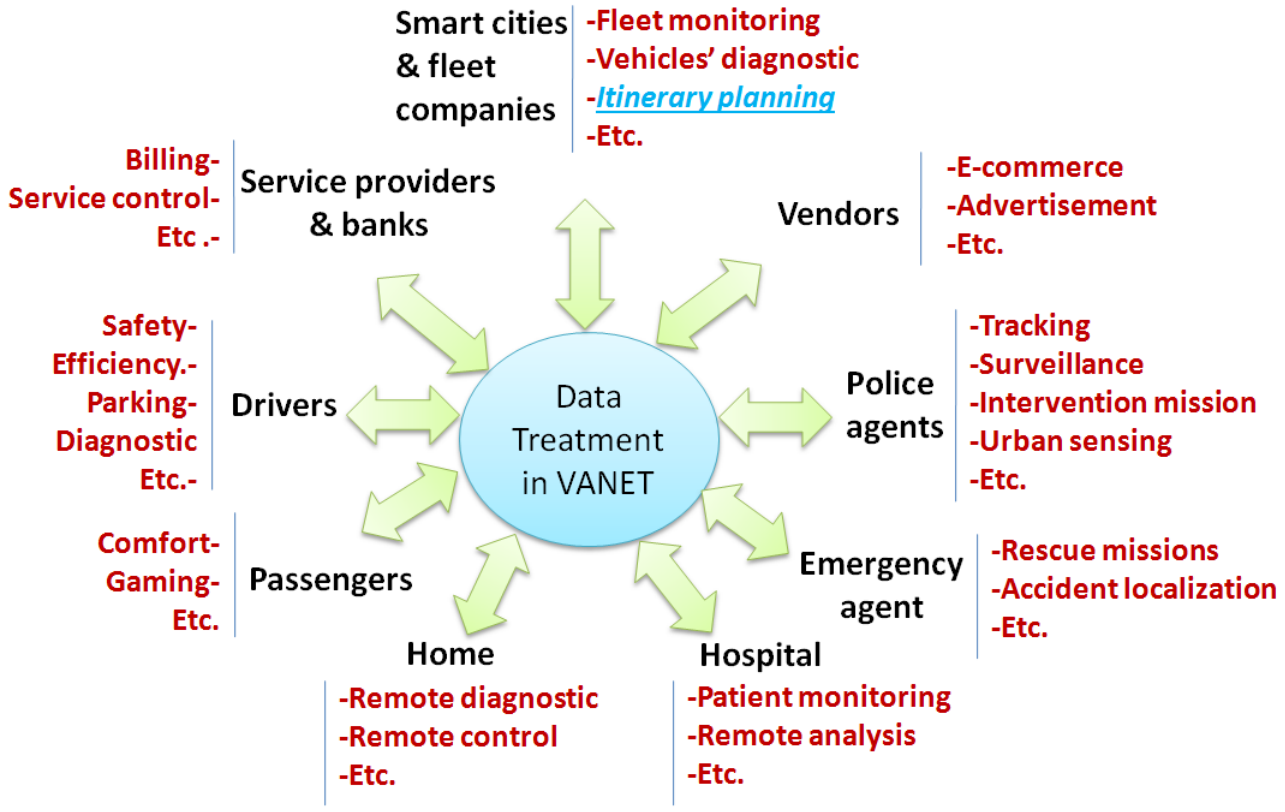



Figure 5.1 – Vehicular data treatment application fields

develop reliable solutions for vehicular network users. Indeed, as a first contribution in this chapter, we design a new application service that is based on a centralized architecture for data collection and continuous analysis of data incoming from various sources namely real-time vehicles data and Internet information to provide the most economic itineraries for traveling from a source to a destination. The proposed application, which we name Itinerary Planning, allows a potentially huge number of users to plan their routes and control their travel time and fuel consumption while having a global overview about traffic and weather conditions in roads based on statistical techniques to fuse information. It is able to process in an off line mode because all the treatment complexity of data is managed by a central server in the infrastructure with Internet connectivity.

Applications developed over vehicular networks require also variable quality of service, a parameter that is highly impacted by the network mobility and random vehicles speed. Therefore, it should be continuously controlled to preserve accurate services. Nowadays vehicles are being enhanced with more than one communication capability to overcome this quality of service degradation and guarantee continuous connections based on the new CALM standard described in section 2.3.2. Offered communication technologies have different characteristics: 4G for example guarantees a high coverage area, whereas IEEE802.11p offers a faster exchange of data without additional costs. For this reason, a handover from one technology to another should be made depending on application requirements while favoring always the best suitable communication medium for every service. This could be only possible by having real-time information about the network, applications requirements, vehicles motion and users preferences. These information are available based on the efficient data collection techniques proposed in the previous chapters. However, they should be accurately exploited and analyzed to have reliable decisions. In a second contribution, we propose a new fuzzy logic-based technique to fuse data for selecting the best network medium for running

applications to preserve quality of service. It is based on the collected data exploitation to analyze the necessity of handover and make switch between the available communication technologies in a smooth and seamless manner. This network selection mechanism highly enhances the capability of the Itinerary Planning application service by offering the possibility to use V2V and V2I communications in a dense networks and 4G for example in a sparse network, thus guaranteeing the application continuity. It guarantees, also, a high quality of service for the other applications offered by vehicular networks based on its fuzzy-based data treatment schema and seamless handover processing.

This chapter is composed by two sections as follows: the first one is devoted to detail the Itinerary Planning application service, where we highlight its components, data analysis process and a proof of concept that we developed using real equipments, and in the second section we introduce the network selection mechanism, explain its architecture and discuss its performances analysis.

## 5.2/ ITINERARY PLANNING SERVICE FOR SMART CITIES

Itinerary planning is an application service developed to allow a potentially huge number of users planning their routes. It is very beneficial for individuals and companies owning fleets of vehicles, to decrease their expenses and preserve the environment by providing the most economic itinerary for a trip. Fire-fighters and police agents also benefit greatly from this service when planning for tracking and rescue missions (European CarCoDe project use-case). The proposed service is based on data collection from different sources and their analysis to provide the best itinerary to a destination. It assesses a refined and precise calculation of the fuel consumption per a trip. Therefore, information about road topologies, weather conditions, traffic and vehicle's internal diagnostics are gathered and analyzed to calculate the fuel consumption of a vehicle from one point to another. For fire-fighters and police agents, the data are also analyzed to offer the fastest route to an incident scene. The application service is offered by a central server in a fixed infrastructure, which has access to other servers in the Internet and responsible of the storage and management of data collected from moving vehicles. It is accessible via mobile equipments (e.g. smart phone, tablet, etc.) using an application interface where information about trip and passengers are introduced to be sent in a request to the central server. The client does not require Internet connection to benefit from the service because all computing complexities are handled by the server which has more privileges and a more general view about the network. In the following, we first, describe the different actors composing the communication architecture of our application, highlight the exchange engendered between clients and the server and analysis process of the different data. Finally, we present a proof of concept with some snapshots highlighting the client side application results.

### 5.2.1/ ITINERARY PLANNING ARCHITECTURE

In this section we detail the proposed architecture for the itinerary planning service processing and depict the role of each entity as well as the used communication technologies. Figure 5.2 highlights a simplified architecture for the itinerary planning architecture with the different components. It is mainly composed as follows:

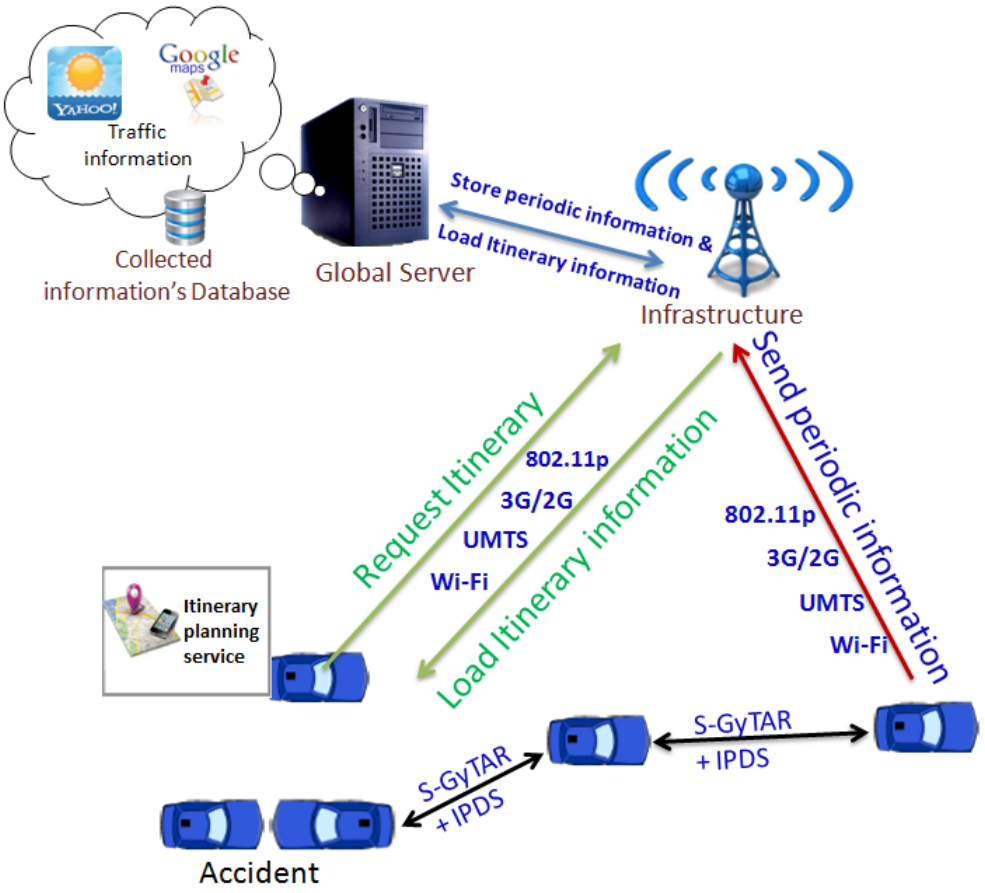



Figure 5.2 – Simplified architecture for Itinerary Planning service

**- The global server,** which is the management center and service provider connected to Internet and has access to all kinds of information collected from vehicles and stored in a database center. This center is composed of four tables; the first one contains vehicle's characteristics (name, weight, aerodynamic coefficient, fuel type, etc.), the second one is dedicated to store fuel's characteristics (fuel identity, fuel name, fuel density ), the third one contains matching information between the specific equivalent fuel consumption (SEFC), mean effective pressure (PME) and engine's RPM while periodically collected data from vehicles are stored in the fourth table. The global server is responsible of the data collection about available service stations and roads' topology from other information centers accessible via Internet. It has the ability to exploit and analyze data to offer different services for every client.

**- The vehicle,** should be enabled with an on-board unit (OBU) connected to embedded sensors and integrated cameras to collect real time information related to the vehicle's internal diagnostics and its environment. Various kinds of information are periodically gathered by the on-board unit and sent to the global server. Among them, we cite some diagnostics which are useful for the itinerary planning service: vehicle's identifier, round per minute (RPM), fuel level, speed, coolant temperature, location, etc. To benefit from offered services such as itinerary planning, the driver should be equipped with an IHM (Interface Human Machine) such as smart phone or tablet to request the needed service and view information provided by the global server.

To enable communication between the different architecture components, three types of connections are recommended. The first is the connection of the global server to Internet which could

be either wired or wireless. The second is the communication between the client's IHM and vehicle's on-board unit which is possible using Bluetooth, IR or WiFi. The last connection is needed between vehicles and the vehicle and infrastructure to collect data and is available via V2V and V2I communication capabilities using the secure intersection-based protocol enhanced with the IPDS mechanism proposed in the previous chapter. However, the vehicle may be enhanced with various access mediums namely 2G/3G, UMTS, WiFi and IEEE802.11p which allow to recover from disconnection in case of sparse networks by switching to a new access medium.

Different actors represent the building blocks of the itinerary planning architecture where each one fulfill a specific role in the process of data collection and exploitation, and service providing. Thus, the global architecture is composed of two main processes: (i) data exploitation and analysis which are delegated to the global server and are based on statistical fusion techniques to combine all kinds of information, and (ii) data exchange to access the proposed service.

#### 5.2.1.1/ ITINERARY PLANNING DATA EXCHANGE

Figure 5.3 illustrates the possible exchange generated by the itinerary planning service each time a client requests it. Potential actors involved in the exchange generated by this service are the driver's IHM (Smart phone, tablet, etc.), the vehicle's on-board unit and the global server benefitting from its access to Internet and data center to provide responses for every request from the client. So, a client books its trip information (source and destination) via an application interface to view possible routes information.

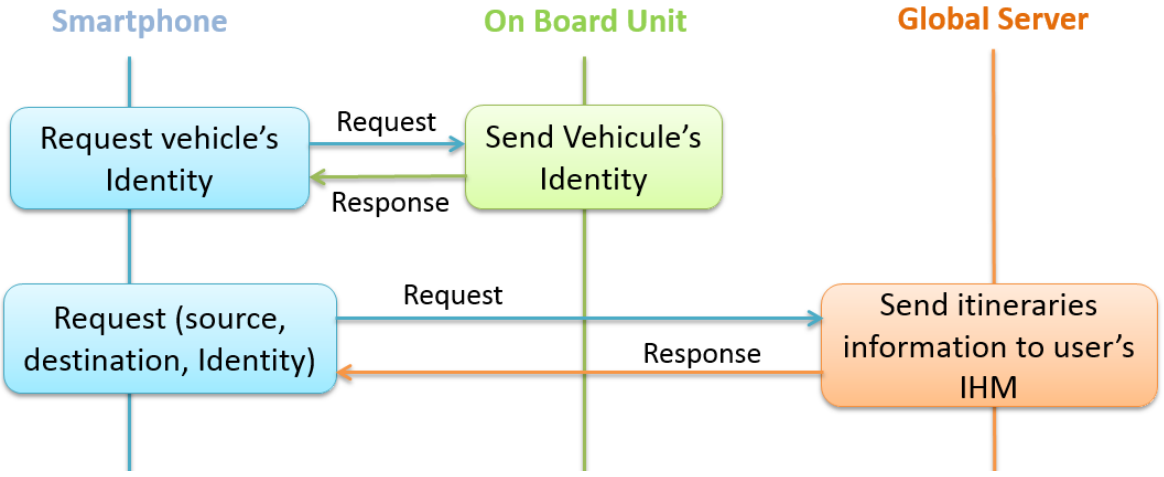



Figure 5.3 – Itinerary planning data exchange

First, the application requests the vehicle's identity from the on-board unit, then sends it with the source and destination information to the global server requesting a possible economic itinerary. After receiving the request, the server searches and evaluates possible routes from source to destination and replies to the client indicating the most economic route and various kinds of useful information such as the travel duration, cost and fuel consumption. However, to get such kind of information the server runs a big number of actions and algorithms in background to analyze data and assess calculation. The most challenging activity processed by the server is how to calculate the fuel consumption over the proposed routes. To do so, the global server uses several APIs through the Internet together with the real-time collected information from vehicles. First, it requests Google API to get all available routes from the given source to the destination, retrieves elevations of multiple points in every route and identifies speed limits per segment. Second, it questions the

Yahoo! Weather API to fetch weather conditions per route. Then, it extracts information about traffic in each route by requesting the Traffic API. These two processes are equivalent to levels 0 and 1 of the data fusion previously reviewed. Finally, and based on all these information, the server calculates the fuel consumption along each route based on the process detailed in subsection 5.2.1.2 (levels 2 and 3 of data fusion). After this, two cases appear: (i) No fuel exhaustion: the fuel level in the vehicle is sufficient to complete the whole trip without need to supply again and (ii) Fuel exhaustion: the fuel level is not sufficient to get the client to destination (the level should not be less than 15%) where the need to provide a path that passes by a gas station. In this case, the server requests Google place API to find the closest gas station to the drop point with its different proposed services and prices and computes the fuel consumption to go through it. Details to calculate the fuel consumption are given by Fig.5.4 and Table 5.1. After computing the total fuel consumption along each route, the total cost is also calculated including the cost of supply within a gas station in case of fuel exhaustion. Finally, upon receiving a response about a planned trip from the server, the IHM displays to the client every route with its associated information and proposes the most economic one and for him to choose the suitable itinerary according to the following characteristics: fuel consumption, traveling time or trip cost. Once the travel starts, the vehicle periodically sends information to the global server as previously described in the data collection process.

#### 5.2.1.2/ ITINERARY PLANNING DATA ANALYSIS

Data collected periodically from vehicles and retrieved from Internet APIs are fused and analyzed by the global server to result the most economic itinerary for a trip after each request from a client. The itinerary planning service is based on the calculation of fuel consumption along each proposed route to a destination. The calculation process must be based on vehicles' and road's characteristics and periodically gathered data for better efficiency and accuracy. This subsection describes the way the global server exploits and analyzes data to calculate the fuel consumption of a vehicle. The whole process of data analysis and fuel consumption is described in the following (see figure 5.4 and Table 5.1):

First of all, the force $F_{wheel}$ needed by the vehicle to move at a given speed is calculated using equation 5.1, where $F_{aero}$, $F_{roll}$ and $F_{incl}$ are respectively the aerodynamic, rolling and gravity forces, $m$ represents the vehicle's mass, $g$ is the gravity value, $i$ is the road inclination given by the call of Google Maps API and $V_{relative}$ is the relative velocity (represents either the difference between vehicle and wind speed if they are in the same direction or the sum if they have opposite directions). The vehicle's mass does not only designate its own weight but also includes weights of passengers and trained trailer if it exists. The wind speed and direction are obtained by requesting Yahoo! Weather API. The vehicle speed $V$ could be deduced from the driving cycle and road's speed limit. $\rho_{air}$ is the air density, $S_{C_x}$ is the aerodynamic penetration coefficient of the frontal surface $S$ and $C_r$ represents the rolling coefficient (depends on $V$, wheal pressure and vehicles characteristics). The air density, normally depends on its temperature but for simplicity it is approximated to a constant value.

$$F_{wheel} = F_{aero} + F_{incl} + F_{roll} \tag{5.1}$$

$$F_{aero} = \frac{1}{2}.\rho_{air}.S_{C_x}.V_{relative}^2 \tag{5.2}$$

$$F_{incl} = m.g.\sin(i) \tag{5.3}$$

$$F_{roll} = C_r.m.g.\cos(i) \tag{5.4}$$

Second, the power $P$ to overcome the force resistance is analyzed based on equation 5.5 and the effective power needed for vehicle's motion (useful energy) is determined as described by equation 5.6 where $\eta$ is the ratio of the overall engine which depends on vehicle's brand and characteristics.

$$P = F_{wheel}.V \tag{5.5}$$

$$P_e = \frac{P}{\eta} \tag{5.6}$$

Third, the auxiliary power $P_{clim}$ provided by the air conditioner is computed and added to the effective power to get the total needed power $P_{tot}$ (see equation 5.7) by the vehicle. To get $P_{clim}$, we based our calculation on some experiments we have made to see the regime variation of the air conditioner because it influences consumption. In fact, we consider the external temperature and fix the setpoint one into the car, thus the air conditioner passes by two regimes: (i) transitional regime which will last until reaching the requested temperature where it is working in a full regime engendering the highest consumption and (ii) permanent regime where it is partially functioning with a low regime depending on the temperature variation in the car (On Off regime). In our experiments we consider the extreme case where the setpoint temperature is the lowest one ($15Celsus$) and the external temperature was about $30Celsus$. These parameters lead to a transitional regime that lasts 4 minutes before to enter the permanent one. $P_{clim}$ could be then determined in function of the engine speed (RPM) knowing the regime.

$$P_{tot} = P_e + P_{clim} \tag{5.7}$$

Fourth, the Specific Equivalent Fuel Consumption (SEFC) should be analyzed. It is given by the vehicle's manufacturer in a form of 3-dimensional curves in function of engine speed (RPM) and brake mean effective pressure in the engine (BMEP) without any specification of how to get it. Every engine has its specific variation of SEFC. For this reason, we propose a program to analyze and extract the SEFC value using provided curves. After that, we multiply the previously calculated SEFC and $P_{tot}$ to get the fuel consumption of the engine $G_f(i;j;k)$ (in gram per hour) for the given road sub-segment, (see equation 5.8).

$$G_f(i,j,k) = P_{tot}.SEFC \tag{5.8}$$

| Notation | Meaning |
|---|---|
| S | Source |
| D | Final Destination |
| V | Vehicle's characteristics |
| R | Routes retrieved with Google Maps API |
| W | Weather retrieved by Yahoo! API |
| E | Elevations values retrieved with Google Maps API |
| Cur_Pos | Current position |
| Ex | Exhausion |

Table 5.1 – Fuel consumption algorithm notations

Finally, to get the total fuel consumption $G_f$ in a route, a fusion of all consumptions along its segments is executed. Furthermore, to consider the traffic state at each route, the server requests

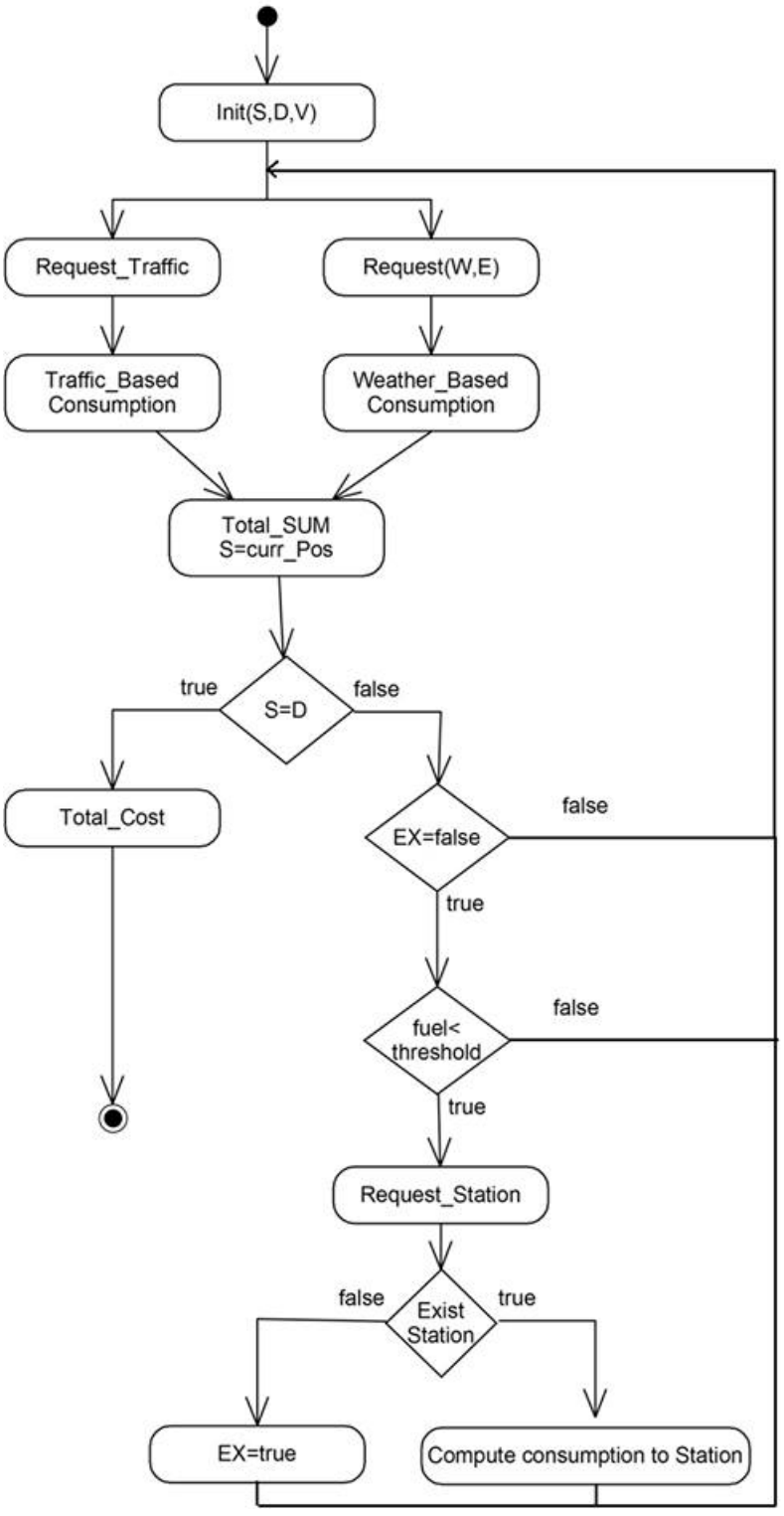



Figure 5.4 – Fuel consumption algorithm

the Traffic API to get information about traffic conditions in roads (e.g. traffic type, location, traffic duration, etc.). The fuel consumption under these conditions is then calculated and introduced to the global consumption.

In case of fuel exhaustion in a route, the main server requests from Google place API the nearest gas stations (radius proportional to the fixed threshold) and computes the quantity of fuel consumed to reach that station from the drop point in the same way as described above. Then, the total consumption of the route in question is updated. The total fuel consumption per route is detailed by algorithm 5:

**Algorithm 5** Pseudo-code of the calculation of fuel consumption per route

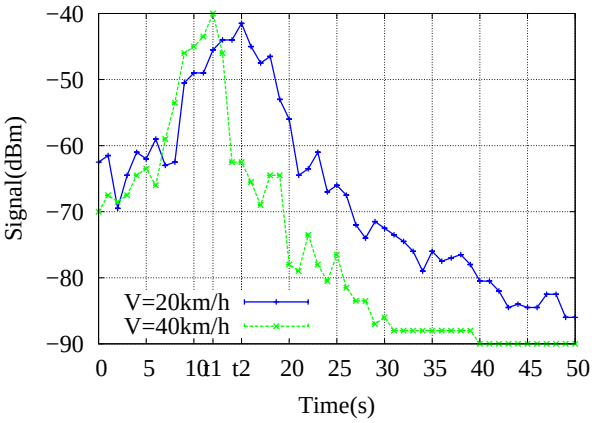
**Input:** Route $i$, State traffic, Service Station location.
**Output:** Total fuel consumption ${G_f}^{route\ i}$.
**for** $j = 1..nb_{segments}(i)$ **do**
  **for** $k = 1..nb_{sub-segments}(i,j)$ **do**
    **if** $Traffic == TRUE$ **then**
      Calculate $T_c(i,j,k)$ fuel traffic consumption with $V = V_{traffic}$;
      Calculate $E_c(i,j,k)$ estimated fuel consumption with V=$V_{limit}$;
      $G_f(i,j) = G_f(i,j) + G_f(i,j,k) - E_c(i,j,k) + T_c(i,j,k)$;
    **end if**
    **if** $Fuel < Threshold$ && $Station == TRUE$ **then**
      Calculate $S_c(i,j,k)$ fuel consumption to service station;
      $G_f(i,j) = G_f(i,j) + G_f(i,j,k) + S_c(i,j,k)$;
    **end if**
  **end for**
  ${G_f}^{route\ i} = {G_f}^{route\ i} + G_f(i,j)$;
**end for**

### 5.2.2/ PROOF OF CONCEPT

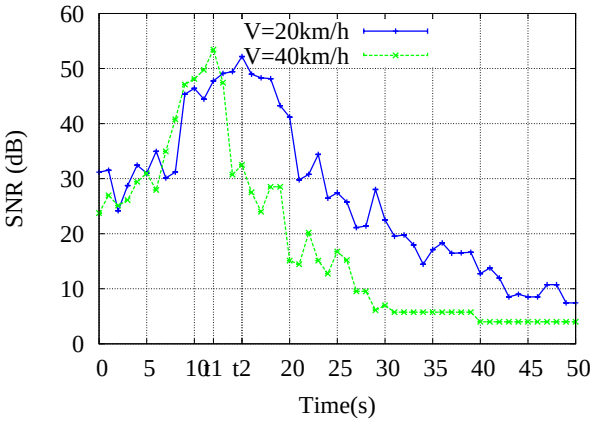
In this section, we describe how we implement a part of the previously presented architecture for the itinerary planning service in order to prove its feasibility and efficiency. The developed platform is mainly composed of two communicating parts: a vehicle and the global server. The vehicle contains an IHM to interact with the driver. Two specific on-board units are connected to each part of the architecture (mobile vehicle and fixed infrastructure where the server is located) to enable them communicate together. For this reason, we perform, first, some tests to be aware about the real performances of V2V/V2I capabilities (IEEE802.11p standard) offered by the specific OBUs before their deployment in our architecture. So, we set a scenario where we place one OBU in a fixed position and integrate another one into a vehicle and test the variations of signal strength, throughput and signal to noise (SNR) in different moving speeds ($20Km/h$ and $40Km/h$). In fact, the static OBU is fixed in the middle of a linear road where the vehicle is going in its direction at the first stage and goes away after that. To retrieve results, we developed a data analysis toolkit based on some shell scripts and a traffic generator called Iperf[1] which is available among existing packages in the OBU operating system. Iperf basically generates traffic between a source and destination and computes exchanged packets with their sizes to get throughput. Results shown by figure 5.5 demonstrate that the throughput keeps a constant variation when the speed increases. However, the distance from the infrastructure highly impacts the signal quality and SNR where they keep increasing when the vehicle approaches the static OBU and decreasing whenever it goes far. The vehicle passes near the infrastructure at time $t1$ (at $t2$ respectively) when the speed is about $20Km/h$ ($40Km/h$ respectively) where the signal and SNR register their highest values. Therefore, we conclude from our experiments that the standard IEEE802.11p is very useful for

[1] https://iperf.fr.

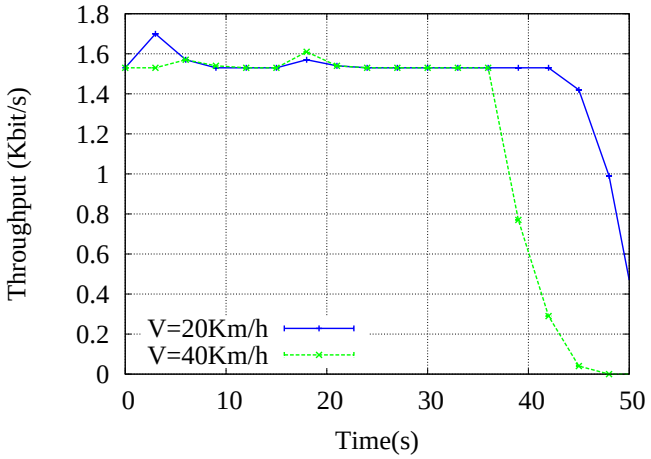
mobile environments because it keeps good throughput despite the degradation of signal quality whenever communicating nodes are in the same radio range. It also represents a good technique for opportunistic communications.




(a) Signal strength




(b) Signal to Noise Ratio




(c) Throughput

Figure 5.5 – IEEE802.11p tested performances

In the global server, we develop the data center which contains vehicle's and fuel's characteristics (vehicle brand, weight, fuel type, fuel density, etc.) together with data collected from vehicles' embedded sensors. The process to calculate fuel consumption was fully developed in the server side based on RESTfull java web services which call several APIs from Internet (Google, Yahoo) together with fresh information from the local data center. However, the process to gather real-time data information about the vehicle is implemented in the vehicle's on-board unit using also a java web service which is scheduled periodically.

To display the route computation results, we implement an Android application which calls web services from the global server. Google Maps APIv2 is used to display possible routes with their related information to the client (elevation, trip cost, time, traffic, etc.) as illustrated by figure 5.7. In case of critical fuel level (less than a threshold), the drop position is located and a fetching for a

closer gas station is initiated. When the gas station is located, a route to it is displayed as shown by figure 5.8. If no closer station is found, a marker is viewed indicating the drop position (Figure 5.6) and an alert is displayed to the driver.

The major components of the proposed architecture together with the itinerary planning service are implemented. Clients do not require to have Internet access to benefit from the itinerary planning application because all processing complexity is managed by the server. Other alternatives to access the server are also envisaged based on a direct connection via 3G/4G and Wi-Fi capabilities.

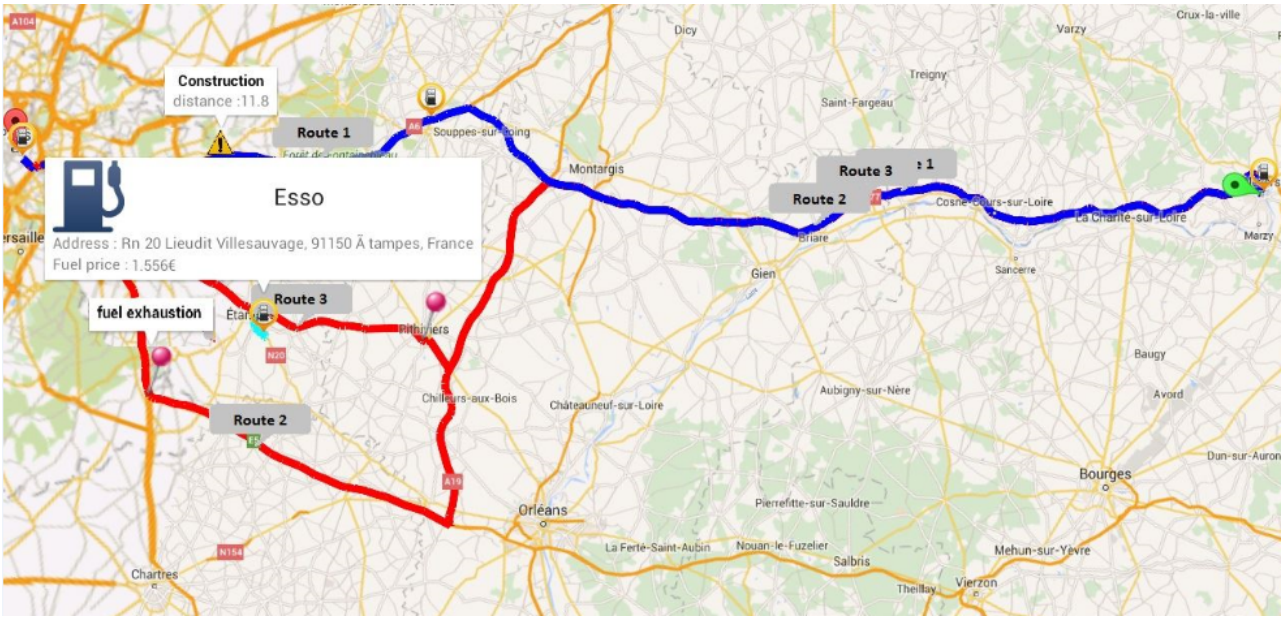



Figure 5.6 – Information about possible routes

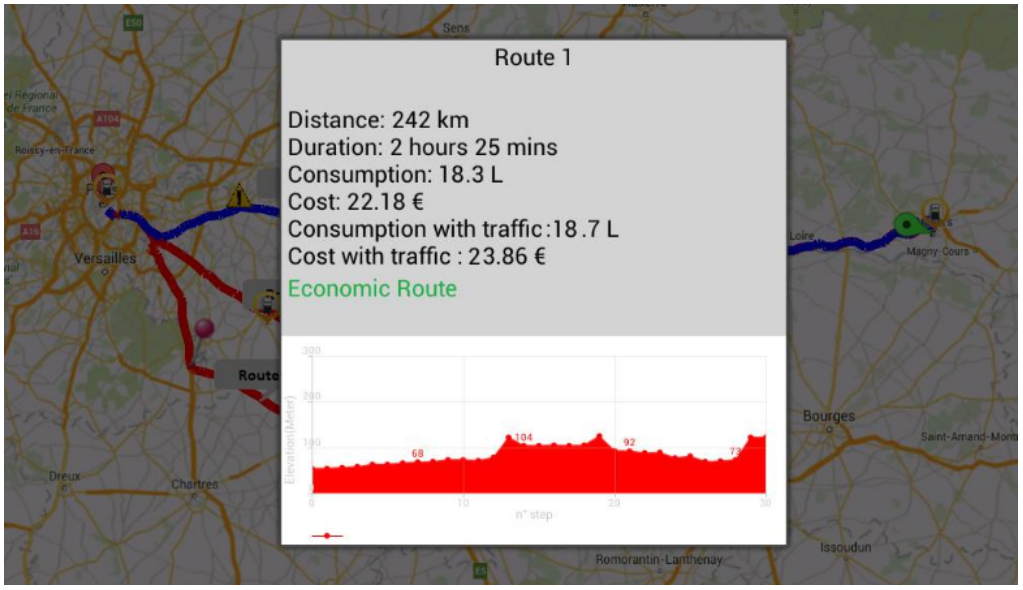



Figure 5.7 – Available information about the most economic itinerary

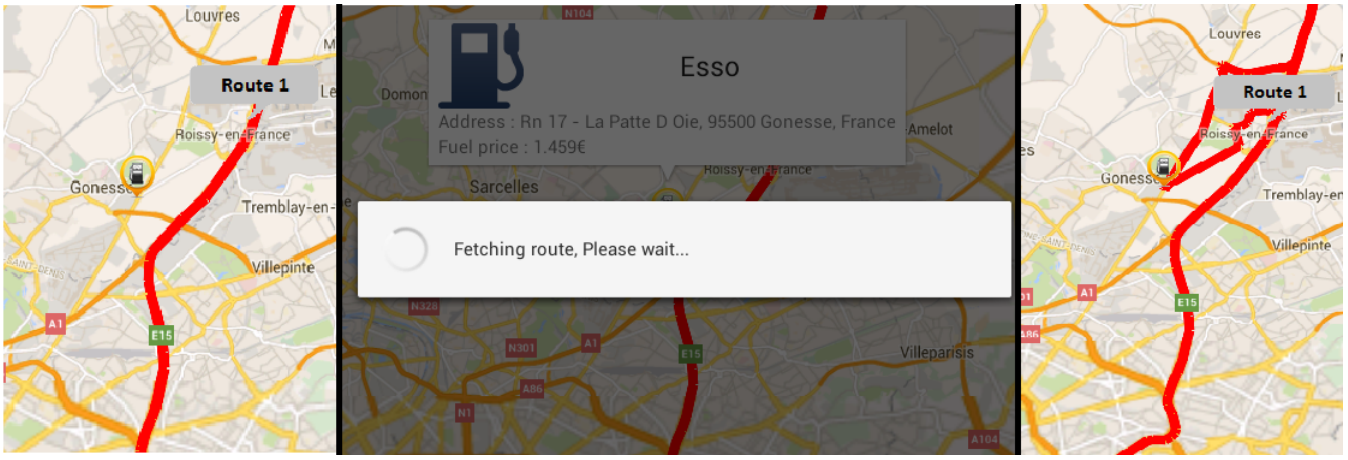



Figure 5.8 – Gas station localization

To conclude, itinerary planning service offers to its clients the possibility to preserve time and money by having a global overview about the network at the time of travel. It takes into account a huge amount of information coming from various sources to enhance the accuracy of analysis. The service is enabled to clients in the global server via a service enabler and is accessible using various kinds of communication technologies (3G/4G, Wi-Fi, 802.11p, etc.). However, the selection of the best suitable communication medium for use in itinerary planning or any other application service should be executed in a smooth and seamless manner where various parameters about network, applications and users should be taken into account to preserve good quality of service. Therefore, we present in the next section our second contribution in this area, which consist in a new fuzzy-based technique to select the best communication medium in vehicles with multiple technologies.

## 5.3/ Fuzzy logic-based communication medium selection for QoS preservation

As described in section 2.3.2, several techniques were developed in the literature to handle the handover between available communication technologies. However, little works consider this problem in vehicular networks and suffer from the problem of not considering a necessary number of parameters that influence the decision process or not stick to an open standard such as CALM. For this reason, the main concern of this work is to make the management and switch between multiple communication interfaces in an ITS equipment soft and smooth to preserve QoS for running applications. Therefore, a new fuzzy logic-based mechanism is proposed. It is compliant with the CALM architecture and considers applications' requirements and access networks specificities in real-time. It collects information about the network and applications which are presented in linguistic format, fuses and analyzes them based on a set of fuzzy rules to give an accurate decision about the most suitable communication medium to use by every application. In fact, Fuzzy logic was introduced by Lotfi A. Zadeh in 1965 based on the theory of fuzzy sets [2]. This concept aims to provide approximate models for uncertain, ambiguous and vague systems whose modeling with precise mathematical approaches is very difficult or sometimes irrelevant. Fuzzy inference systems based on fuzzy inference rules may be used to overcome the use of extensive mathematical details or complex algorithmic structures in system modeling while good results can still be obtained. Inputs and outputs of a fuzzy inference system (FIS) are provided in a linguistic presentation and

have linguistic values such as low, medium and high. Fuzzy inference rules are a set of if-then rules which are able to decide based on provided formal input values and give output values. In the following, we firstly detail the main building blocks and inference systems of the proposed mechanism, then we highlight the different results we got and discuss them.

### 5.3.1/ FUZZY LOGIC-BASED NETWORK SELECTION MECHANISM

In this section, we first highlight the global architecture of our fuzzy-based network selection mechanism which was developed based on the architecture proposed by the CALM standard to be easy to integrate in real equipments; then, we detail each block that participates in the data exploitation, analysis and decision process about the need of handover and its execution.

#### 5.3.1.1/ GLOBAL ARCHITECTURE

As said above, selecting the best communication medium is a very challenging issue that requires to know various kinds of information about the network, applications and user preferences. The main difficulties arise from the frequently changing parameters of available networks due to vehicle's speed. For this reason, the fuzzy logic-based network selection mechanism architecture includes six main modules presented by figure 5.9: (i) application characteristics module, (ii) network and vehicles parameters set, (iii) network & vehicle parameters collection module, (iv) network inference system (NIS) , (v) network weighting inference system (NWIS) and (vi) network selection inference system (NSIS).

**1- Application characteristics module:** it is responsible of the identification of running applications and determines their requirements in terms of QoS (throughput, delay, etc.), security and performance.

**2- Network & vehicle parameters collection module:** this module gathers information about available networks such as received signal strength intensity (RSSI) and density and retrieves the vehicle speed. These information are very important to get the performances related to each network candidate.

**3- Vehicle & network parameters module:** this module is used to store information collected by the previous module.

**4- Network inference system (NIS) module:** it is a fuzzy logic module used to determine the throughput, latency and packet loss from the fuzzy input values about RSSI, vehicle speed and network density.

**5- Network weighting inference system (NWIS) module:** this module takes outputs given by the previous module together with information coming from the running application, analyzes and treats them based on a set of internal rules to result a weight for each network candidate.

**6- Network selection inference system (NSIS) module:** it is responsible for the final decision about the network to attribute to a running application. It is also based on a set of fuzzy rules which fuse incoming information from the previous module with the cost offered by each network to decide about the best medium to be used.

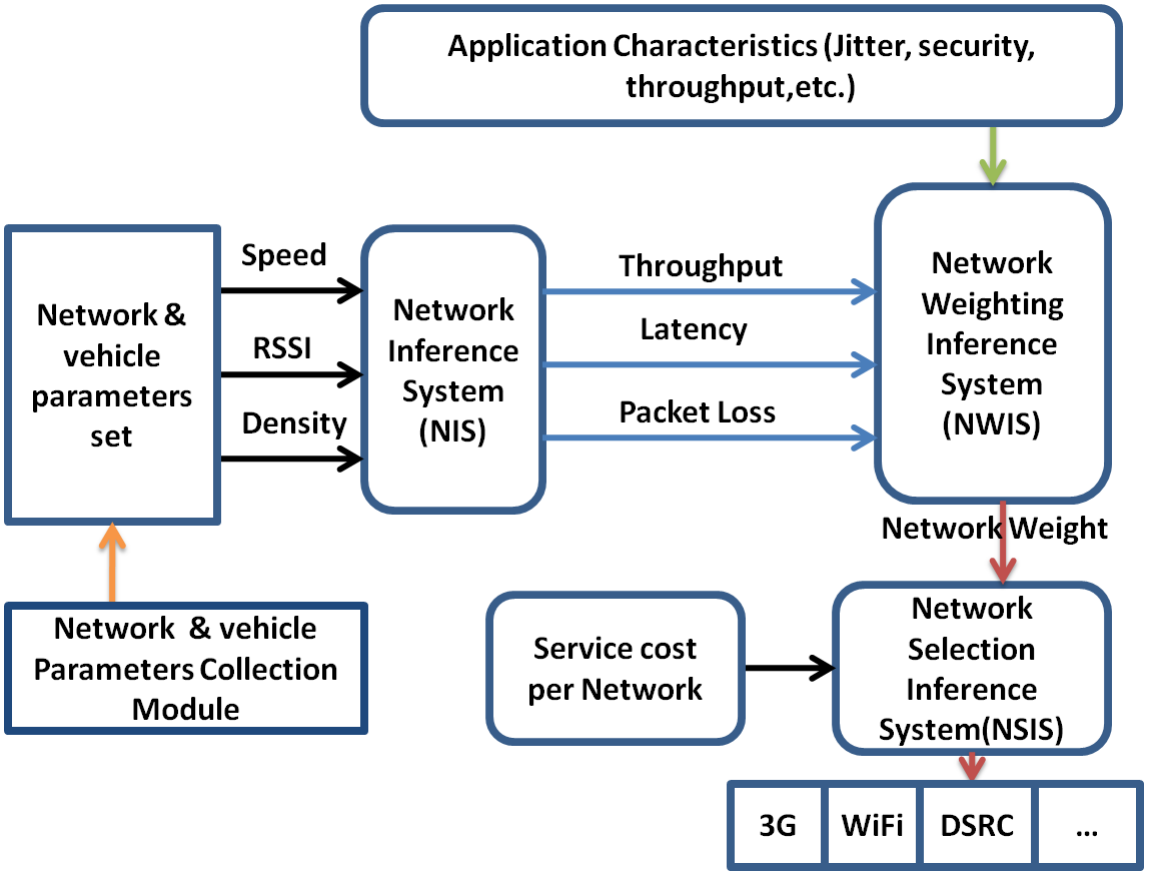



Figure 5.9 – The fuzzy-based network selection architecture

#### 5.3.1.2/ NETWORK INFERENCE SYSTEM (NIS)

This module is responsible for the exploitation and analysis of available data stored in the network & vehicle parameters set. It is a fuzzy logic inference system based on if-then fuzzy rules. Its objective is to extract information characterizing each communication over a candidate access network based on given indications about the vehicle speed, network density and RSSI. This module is composed of 13 fuzzy rules which we give in Table 5.2. Its inputs are presented in a linguistic format as defined in the following:

**- Vehicle speed:** in a vehicular network, roads topology are variable and encompass highways and urban zones. So, a vehicle speed should vary from road to road depending on the type and imposed speed limit. It has the following linguistic values: low, average and high speed. When a vehicle is traveling in a residential area with a speed limit of $50Kmph$, its velocity is designated by the low speed value. The speed is average when it is between $50Kmph$ and $100Kmph$ which is the case of national/regional roads. When a vehicle is traveling with a speed higher than $100Kmph$ it will be considered high.

**- Received signal strength intensity (RSSI):** characterizes the signal quality of a network. In general, its value varies from $-100dBm$ (weak) to $-15dBm$ (strong). Its linguistic values are classified as follows: (i) if the RSSI is less then $-70dBm$, the signal is considered to be weak, (ii) if it is between $-70dBm$ and $-50dBm$, the signal is termed as average and (iii) if it is greater then $-50dBm$, the signal is identified to be strong.

**- Network density:** represents the ratio between the number of neighbors present in the radio range of a vehicle and the maximum allowed number which may cause congestion due to competitive access between nodes. It is calculated based on the number of neighbors given by periodic exchanged beacon frames and the maximum number of vehicles that could be present in a vehicle's radio range ((number of neighbors / maximum number in radio range) * 100). To get the maximum number of vehicles in the radio range we assume that vehicles have the same length and respect the legal fixed distance between each others. The density is considered low when its value is less than $30\%$, medium from $30\%$ to $70\%$ and high if it is greater than $70\%$.

| RSSI | Density | Speed | Throughput | Latency | Packet loss |
|---|---|---|---|---|---|
| weak | low | low | medium | medium | medium |
| average | low | low | high | low | low |
| strong | low | low | high | low | low |
| weak | medium | low | medium | high | medium |
| average | medium | low | medium | medium | medium |
| strong | medium | low | high | medium | low |
| weak | high | low | low | high | high |
| average | high | low | low | high | high |
| strong | high | low | medium | medium | medium |
| strong | low | average | medium | medium | medium |
| average | medium | average | low | high | high |
| weak | medium | average | low | high | high |
| - | - | high | low | high | high |

Table 5.2 – Network inference system (NIS) fuzzy rules

#### 5.3.1.3/ NETWORK WEIGHTING INFERENCE SYSTEM (NWIS)

This module, firstly, reads from the application characteristics module and identifies registered QoS requirements by a running application regarding latency, throughput and packet loss. Then, it maps these QoS parameters to the ones offered by each available access network to check which one best suits the application and assigns to each network a weight. This fuzzy inference system is composed of a set of fuzzy rules to assign a weight for each network ranging from 0 to 1. Designed rules consider to treat each application requirements apart. The VoIP, for example, needs very low packet loss and latency; so only networks that offer this possibility are highly weighted (high weight). For a streaming application, networks offering low and medium metrics will have good weights (high or medium). However, for a downloading application, networks with lower characteristics will have medium weights. We summarize the set of fuzzy rules for the NWIS in Table 5.3 while considering the three types of application we described; streaming, downloading and VoIP. Inputs provided by the NIS module to NWIS are given in a linguistic format as follows:

**- Throughput:** in this work, we consider the throughput to be low when its value is less than $0.5Mb/s$, medium from $0.5Mb/s$ to $1.5Mb/s$ and high if it is greater than $1.5Mb/s$.

**- Latency:** Applications tolerate variable latencies. A VoIP application has very hard requirements regarding latency. The ITU-T G.114 recommends a maximum of $150ms$ one-way latency and most network Service Level Agreements (SLA) specify between $45ms$ and $65ms$ as the maximum latency [170]. For daily human use streaming applications (e.g. playing game, live video conference) $100ms$ is considered to be a low latency. Some other applications may operate even with a high

latency such as file downloading. For this reason, we consider in this work latency to be low when blow than $100ms$, medium between $100ms$ and $150ms$ and high when it is above $150ms$. So, a VoIP application will run under a low latency, streaming under low and medium latencies and downloading runs even under high latencies.

**- Packet loss:** Some applications like VoIP and real time video streaming are very rigid regarding packet loss and do not tolerate more than $1\%$, however others like non real time streaming may tolerate until $4\%$ of packet losses and above this value the QoS of running applications considerably degrades [172]. For this reason the packet loss is considered to be low until $1\%$, medium from $1\%$ to $4\%$ and high above this value.

| Throughput | Latency | Packet loss | Application type | Weight |
|---|---|---|---|---|
| low | low | low / medium | streaming | medium |
| medium | low | low / medium | streaming | high |
| high | low | low / medium | streaming | high |
| low | medium | low / medium | streaming | medium |
| medium | medium | low / medium | streaming | medium |
| high | medium | low / medium | streaming | high |
| low | high | low / medium | streaming | low |
| medium | high | low / medium | streaming | low |
| high | high | low / medium | streaming | medium |
| - | - | high | streaming | low |
| low | - | low / medium | downloading | medium |
| low | - | high | downloading | low |
| medium | - | low | downloading | high |
| medium | - | medium | downloading | medium |
| medium | - | high | downloading | low |
| high | - | low / medium | downloading | high |
| high | - | high | downloading | medium |
| low | low | low / medium | VoIP | medium |
| low | low | high | VoIP | low |
| medium | low | low | VoIP | high |
| medium | low | medium | VoIP | medium |
| medium | low | high | VoIP | low |
| high | low | low/medium | VoIP | high |
| high | low | high | VoIP | low |
| - | medium | - | VoIP | low |
| - | high | - | VoIP | low |

Table 5.3 – Network weighting inference system (NWIS) fuzzy rules

#### 5.3.1.4/ NETWORK SELECTION INFERENCE SYSTEM (NSIS)

At this stage, weights given by the NWIS are combined with the cost offered by each network to satisfy the given service group in order to decide what will be the selected medium. Normally, costs imposed by every network are known and given by the service providers. So, the NSIS considers these costs and chooses the network that offers the best tradeoff between cost and QoS. This inference system is based on a set of rules to make a choice between two mediums and could be

extended to support more rules for more mediums. We summarize the set of the NSIS internal fuzzy inference rules in Table 5.4 considering two mediums (M1 and M2). We consider in these rules that the quality of service of an application is more important than the service cost when the choice is made.

| Weight M1 | Cost M1 | Weight M2 | Cost M2 | Decision |
|---|---|---|---|---|
| low | low | low | low | Keep actual medium |
| high | high | high | high | Keep actual medium |
| medium | medium | medium | medium | Keep actual medium |
| high | - | low / medium | - | M1 |
| medium | - | low | - | M1 |
| low | - | high / medium | - | M2 |
| medium | - | high | - | M2 |
| low | low | low | medium / high | M1 |
| low | medium / high | low | low | M2 |
| low | medium | low | high | M1 |
| low | medium | high | low | M2 |
| medium | low | medium | medium / high | M1 |
| medium | medium / high | medium | low | M2 |
| medium | medium | medium | high | M1 |
| medium | high | medium | medium | M2 |
| high | low | high | medium / high | M1 |
| high | medium | high | low | M2 |
| high | medium | high | high | M1 |
| high | high | high | low / medium | M2 |

Table 5.4 – Network selection inference system (NWIS) fuzzy rules

### 5.3.2/ PERFORMANCE EVALUATION

During this section, we present at first the test environment we use to evaluate the performances of our designed fuzzy-based network selection mechanism together with the evaluation criteria we have chosen. Then, we detail the given results and discuss them.

#### 5.3.2.1/ ENVIRONMENT DESCRIPTION

We implement our network selection mechanism using MATLAB and Simulink [2], then we experiment various kinds of services to see the best communication interface selected by our proposed fuzzy inference system. We consider in our experiments two communicating nodes where each one is equipped with two access mediums: IEEE 802.11p and 3G interfaces. To test switching between the two communication mediums, three types of applications are used: VoIP conversation, streaming and downloading applications where each one belongs to a service group and has its specific QoS requirements. The main used performances criteria we aim to evaluate are defined below:

**- Packet delivery ratio (PDR):** it is the number of received packets in a destination divided by the number of sent packets by a source. It mainly reflects the quality of communication between two

[2] http://fr.mathworks.com/

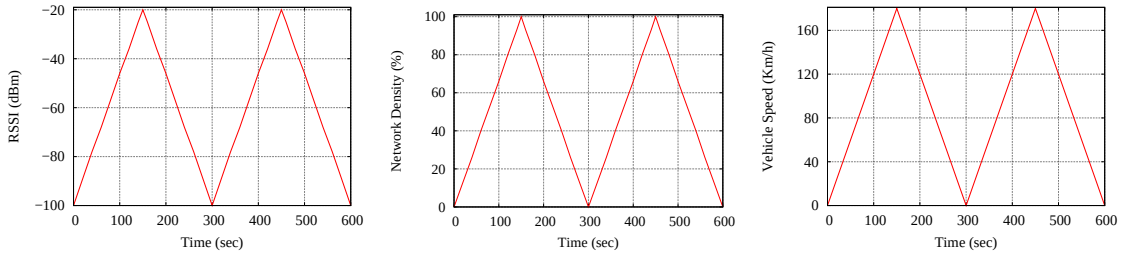
nodes,

**- Packet loss ratio (PLR):** defines the ratio of packets sent by a source but never reach the destination due to the communication quality or link breakage,

**- End to end communication delay:** defines the time that a packet consumes to reach a destination. It is calculated by the difference between the arrival time of a packet to the destination and its departure from the source node.

#### 5.3.2.2/ RESULTS AND ANALYSIS

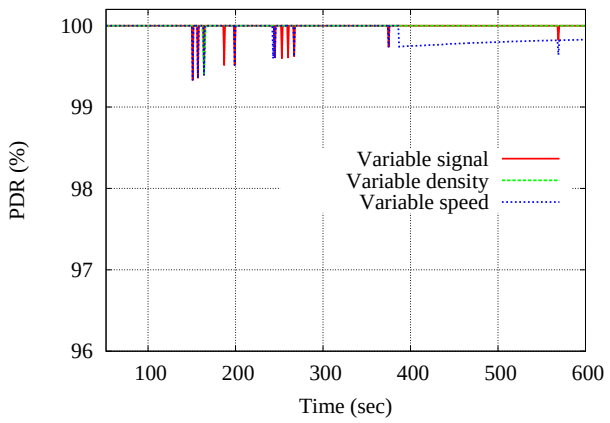
We firstly analyze the impact of each parameter collected from the network (speed, RSSI and density) on the communication performances of each application type. Parameters vary as shown by figure 5.10. Results presented by figure 5.11 demonstrate that the fuzzy network selection mechanism is able to preserve a high QoS (high PDR, low PLR and low end to end delay) and supports applications with rigid requirements when the network performances change. The delay doesn't exceed $30ms$ and remains in the low interval which is supported by all kinds of applications (signal and density curves are overlapping in the figure). The PLR slightly increases at the phase of network changing and regains its permanent value without exceeding $1\%$ (maximum allowed loss in a VoIP application). The results shown in figure 5.11 are given the same when we proceed to the change of application (VoIP, Streaming, Download). These results are achieved thanks to the smooth switching between available access mediums and the consideration of various kinds of parameters related to networks, applications and user preferences.



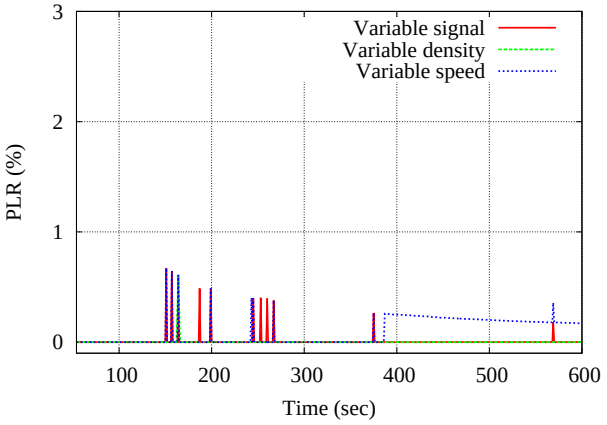

(a) RSSI (b) Network density (c) Speed

Figure 5.10 – Network parameters variation

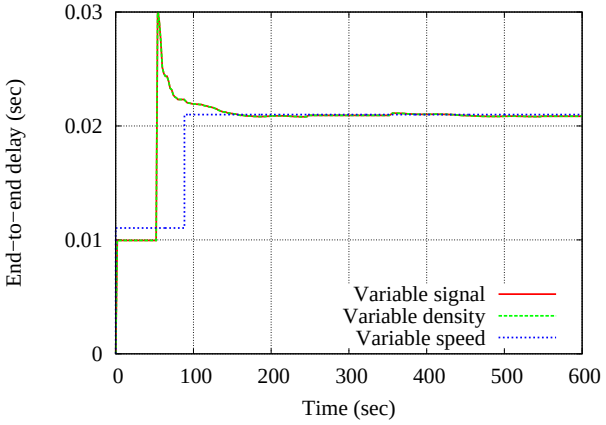
Secondly, we compare our designed fuzzy logic-based network selection mechanism to a RSSI-based algorithm and to a communication with only one medium (IEEE 802.11p). As the received signal strength of a 3G medium always keeps a relatively constant intensity, we consider that the RSSI of the 802.11p medium is changing and we measure the offered QoS regarding packet loss, delivery ratio and communication delay. As shown by figure 5.12, the fuzzy based network selection technique outperforms the RSSI-based algorithm and offers a high delivery ratio (more than $99\%$), low packet loss (less than $1\%$) and low end-to-end delay (less than $30ms$). These results are due to the consideration of various types of information related to the network, application requirements and user preferences by the designed fuzzy system compared to the RSSI-based algorithm, which only takes into account the RSSI variation of a medium. We also notice in our simulations that the frequency of changing between the two communicating mediums is less in the fuzzy system than RSSI-based technique (3 changes vs. 6 over 600s of simulation time) which limits the effects of ping-pong problem. This can be explained by the fact that for the RSSI-based algorithm switching to a new medium is made whenever the received signal intensity is lower than a fixed threshold,




(a) PDR



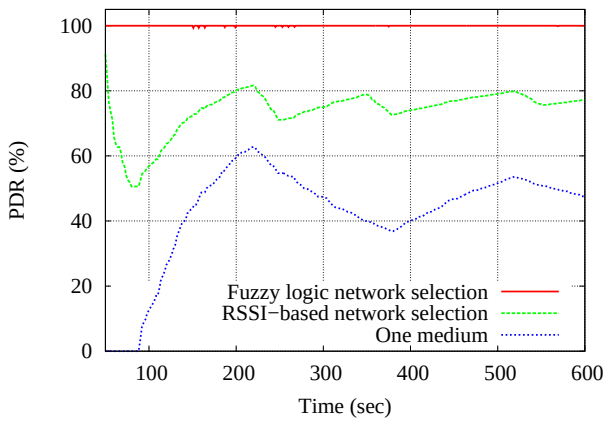

(b) PLR



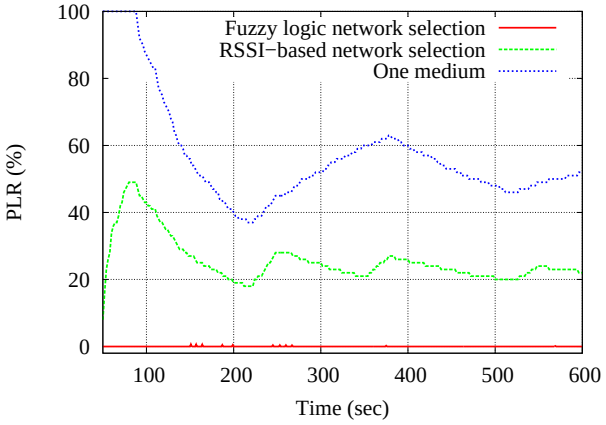

(c) End-to-end delay

Figure 5.11 – Fuzzy logic network selection under variable parameters

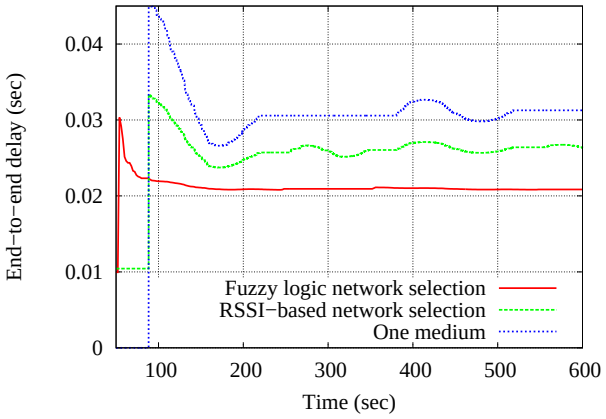
however for the fuzzy logic-based system the medium change is executed only when all metrics degrade and a new tradeoff is found among candidate access networks. Therefore, we conclude that our fuzzy logic-based network selection mechanism is able to preserve running applications from potential disconnections because it offers a seamless handover between available communication mediums and is responsible of packets handling from medium to medium without losses in real-time. The proposed fuzzy system shows similar performances with the three kinds of applications we have tested (VoIP, streaming, downloading). It is also easy to introduce in real ITS equipments because it is compliant with the CALM standard and does not require high computing capacities.




(a) PDR




(b) PLR




(c) End-to-end delay

Figure 5.12 – Fuzzy logic vs RSSI-based network selection

## 5.4/ CONCLUSION

Exploiting and analyzing data flowing in the network in an efficient and accurate manner to enhance the driving conditions in a vehicular network and protect ITS-application users, especially, drivers from potential disconnections and low quality of service, was the main concern of our contributions during this chapter. For this aim, we considered the data collected from the network and developed a technique for their fusion to design first an itinerary planning application service able to protect the environment and preserve time and money (fuel consumption) for its clients. We also, used the collected data to control and enhance the quality of service offered by services and applications developed over vehicular networks using a new fuzzy logic-based network selection mechanism able to select the best suitable communication medium for a running application. Itinerary planning service, based on a centralized architecture, exploits and fuses various kinds of data from several sources namely real-time collected data from vehicles and information retrieved from Internet APIs to calculate finely and accurately the most economic itinerary for a trip requested by the client. Unlike other applications, itinerary planning gives a global overview about road traffic and the real consumption of a vehicle during its trip because it considers roads elevation, weather conditions

and temperature together with vehicles engine characteristics. The new fuzzy logic-based network selection mechanism is able to manage available communication mediums in a vehicle while preserving higher performances of these applications. It considers all parameters that may impact the decision accuracy about the best network namely networks' characteristics, applications' requirements and user preferences. It is able to fuse and analyze gathered data from the network, to take a decision in real-time and switch to a new medium if needed in a seamless and smooth manner. To conclude, this chapter presents the way we analyze and exploit data collected from a vehicular network to ease traveling and offer better quality of service to deploy ITS-applications and benefit from their use in an efficient and accurate manner.

In the next chapter, we conclude this thesis and we present our future work.

# 6

# CONCLUSION AND PERSPECTIVES

## 6.1/ CONCLUSION

In the few past years, vehicular networks have gained the interest of research communities and car manufacturers where a vehicle is being instrumented with new communication technologies and a huge number of sensors to increase its awareness about the environment and its interactivity with its neighbors. This has led to the increase of the size of data flowing in such kind of networks which are characterized by their heterogeneity, incompleteness and redundancy. These data are of high importance to build ITS-applications and should be collected and analyzed to be accurately exploited. However, ensuring a secure and efficient data collection, exploitation and analysis still present a very challenging field since vehicular networks are characterized by their frequent changing topology, frequent fragmentation and random & high vehicles speed. Furthermore, vehicles have always been connected which make them vulnerable and usually exposed to various attacks. In addition, to offer reliable and accurate services for vehicular network users, data should be efficiently treated and analyzed depending on the application requirements where quality of service should be preserved to satisfy clients. For this reason, we focused our attention during this thesis on how to exploit vehicles capabilities to overcome the encountered challenges and highly benefit from the services and applications that may be developed over a vehicular network to increase drivers and passengers safety, offer a good management for time and fuel and preserve client access to services. Hence, in this thesis we are dealing with two main aspects: data collection and data exploitation. In the first aspect, we proposed a new optimized data collection architecture together with a routing protocol and security mechanisms with the aim of increasing the security and efficiency of the exchanged data. Whereas in the second aspect, we explored new ways to efficiently and accurately exploit and analyze data for offering new services that are able to preserve users' time and money while planning for their trips, and ensure a better exploitation of available access networks in a vehicle to preserve a high quality of service for running applications and prevent from uncomfortable disconnections. Below, we conclude our contributions for both data collection and exploitation.

We firstly, devote our interest to study the different available ways to collect data from a vehicular network and assess its exchange between nodes. So, we reviewed the major routing contributions and we analyzed them to highlight their weaknesses. Afterward, we proposed a new technique to optimally deploy a reduced number of data harvesters in an urban area for collecting data. This technique, based on a meta-heuristic called Variable Neighborhood Search, is able to give in a reduced time the optimal number of vehicles needed to harvest a specific urban region and assign to each one its travel area, a result that was confirmed by its evaluation on the Manhattan roads topology extracted from OpenStreatMap (OSM). The data collected by harvesters should be delivered to a fixed management center, which is the entity responsible for their deployment, to be

exploited and analyzed. So we proposed two new contributions to assess a secure data routing. In the first one, we provided a new secure intersection-based routing protocol (S-GyTAR), which only considers non malicious vehicles in the routing process. This protocol relies on a clustered architecture to monitor vehicles and detect malicious ones to be evicted from the network. The routing process is based on a decision making in each intersection based on a weighting technique that takes into account traffic information and distance from destination. During a road segment data is being relayed between vehicles based on a position prediction technique to choose forwarders and envisage a store and forward technique to recover from possible disconnections. In the second contribution, we proposed a new intrusion detection and prevention system (IPDS) to decrease the vulnerability window of attacks by monitoring network members and detecting attackers before they persist in their attacks. The results given by the two last contributions demonstrated their efficiency and reliability to deliver and protect data in a vehicular network by reducing delays and increasing delivery ratio in a malicious environment.

In a second step during this thesis, we dedicated our effort to investigate the different techniques to exploit collected data from the network to propose various services to multiple users. Hereafter, we proposed new solutions to efficiently exploit data with the aim of enhancing travel conditions in one hand and protecting applications' quality of service in the other hand. Therefore, two contributions have seen the light during this study under the umbrella of data exploitation and analysis. The first one consists in an itinerary planning application service that offers to its users the most economic way to travel based on the data exploitation and analysis from the network and other Internet sources with a consideration of the real characteristics of vehicles and necessary factors that may impact the travel duration and cost namely weather conditions, roads topology and traffic data. The second contribution, mainly aims to preserve running applications continuity and quality of service based on an efficient exploitation of available communication mediums. It consists on a network selection technique that is able to evaluate available access networks and make handover in a smooth and seamless manner based on a fuzzy logic-based data analysis of information from the network, applications and user preferences. The results given by our fuzzy-based network selection prove its ability to protect users and applications from losses.

Finally, during this thesis we have made an extensive study of vehicles communication and sensing capabilities to be able to exploit them to assess a secure exchange and offer reliable services to a huge number of clients based on the efficient exploitation and analysis of the sensed data. We believe that we achieved our target by providing a secure data routing while considering vehicular network specificities and the proposition of a new reliable application with robust connection. Our work is open to possible extensions to enlarge the treated fields and face new emerging challenges. Hence, possible enhancements will be the aim of our future work described in the next section.

## 6.2/ PERSPECTIVES

This section is devoted to study new research perspectives and possible direct applications of the contributions provided during this thesis. In fact, this thesis proposes several concrete works that are able to be applied in the court or long term. Yet, these works can be further extended in some manners.

Regarding our contributions proposed for data collection, we aim to enhance the security by considering additional kinds of attacks such as the monitoring and timing attacks to increase the integrity of the flowing data and secure its holder by further developing new techniques based on data integrity verification. For this aim, we envisage the exploitation of web semantic ontologie to characterize data and attackers. We also aim to assess an analytical study of the impact of real traffic lights

and stop signs in a vehicular network on the proposed secure routing protocol. This impact will be analyzed based on its effects on the communication delay and delivery ratio. Further enhancements are also planned to enable the proposed data collection techniques supporting non-delay tolerant applications by decreasing the storing delays of packets in case of sparse networks. This can be done by the exploitation of different communication capabilities and prioritization technique to organize the data depending on application requirements. The bootstrapping period used by our IPDS will be further investigated to get its real value in function of authentication and connection establishment durations. It will be analytically defined in function of the time consumed by each process. Further possibilities are also envisaged to combine our Kalman filtering technique used in IPDS with game theory techniques presented in appendix A to increase prediction capabilities.

In the data exploitation and analysis part, we aim to develop new services based on the real-time collected data from vehicles to help for engine status diagnostics & anomaly detection, empty parking places detection to decrease the passive traffic, etc. These services will be developed based on the same architecture proposed in our itinerary planning application to alleviate data management into vehicles and enlarge their view. The itinerary planning service will be further enhanced to consider more information that may affect a trip such as drivers' behavior and include a new alerting system to alert drivers about abnormal events detected in its surrounding. It will be also augmented with other properties to be considered and exploited in other fields (smart-cities for example) with new parameters and requirements such as the assistance of citizens that are sensitive to pollution or noise by offering less polluted routes to follow and identifying quiet areas in a city to be visited. Finally, our fuzzy logic-based network selection mechanism will be augmented with new rules to manage more available networks before being implemented in real ITS equipments as it is compliant with the CALM standard.

# 7

# PUBLICATIONS

- **Journals (1)**
  - T. Bouali, S-M. Senouci, and H Sedjelmaci. ”A distributed detection and prevention scheme from malicious nodes in vehicular networks”, Accepted in International Journal of Communication Systems (Wiley), to appear in 2016.
- **International conferences (6)**
  - T. Bouali, E.-H. Aglzim, S.-M. Senouci ”Optimization of data harvesters deployment in an urban areas for an emergency scenario”, IEEE Global Information Infrastructure Symposium - GIIS’2013 , pp.1,6, 28-31 Oct. 2013.
  - T. Bouali, E.-H. Aglzim, S-M. Senouci ”A secure intersection-based routing protocol for data collection in urban vehicular networks”, IEEE GLOBECOM 2014, Austin, USA, 8-12 December, 2014.
  - H. Sedjelmaci, T. Bouali, S-M. Senouci ”Detection and prevention from misbehaving intruders in vehicular networks”, IEEE GLOBECOM 2014, Austin, USA, 8-12 December, 2014.
  - H. Sedjelmaci, T. Bouali, M.A. Messous S-M. Senouci ”How to prevent cyber-attacks in inter-vehicle communication network?”, IEEE CFIP-NOTERE, Paris, France, July 22-24, 2015.
  - T. Bouali, M. Elhami, A. Ben Sassi, S-M. Senouci, T. Sophy and A. Kribeche ”A Vehicular Network Architecture for Data Collection: Application to an Itinerary Planning Service in Smart Cities”, In Global Information Infrastructure Symposium (GIIS2015), October 2015.
  - T. Bouali, S-M. Senouci and H. Sedjelmaci ”A distributed prevention scheme from malicious nodes in VANETs’ routing protocols”, Accepted in IEEE WCNC 2016, Doha, Qatar, 3-6 April 2016.
  - T. Bouali and S-M. Senouci ”A fuzzy Logic-Based communication medium selection for QoS preservation in vehicular networks”, Accepted in IEEE ICC 2016, Kuala Lumpur, Malaysia, 23-27 May 2016.
- **Articles for the Layman (1)**
  - T. Bouali, S-M. Senouci, M. Albano, A. Lallet, F. Ben Badis, ”Platform for Smart Car-to-Car Communication”, IEEE AHSN News Letter, to appear in December 2015.
- **Talks ( 1)**

  - T. Bouali, S-M. Senouci ”Green Itinerary Planning Application for Smart Cities”, In workshop on vehicular networks in today’s society, Castelo Branco, Portugal, 2-3 June 2015.

- **Journals and conferences under submission (3)**

  - H. Sedjelmaci, S.-M. Senouci, T. Bouali, ”Using Bayesian Game to Predict and Prevent Misbehaving Intruders in Vehicular Ad hoc Networks”, Submission to IEEE Transaction on Vehicular technology Journal.
  - T. Bouali, S.-M. Senouci, M. Albano and A. Lallet ”Data Collection and Analysis in a Vehicular Network”, Submission to S.I on Vehicular Communications Journal (Elsevier).

# BIBLIOGRAPHY


**[1]** KALMAN, R. E. **A New Approach to Linear Filtering and Prediction Problems 1**. 35–45.

**[2]** ZADEH, L. **Fuzzy sets**. *Information Control 8* (1965), 338–353.

**[3]** MERKLE, R. C. **Secure communications over insecure channels**. *Commun. ACM 21*, 4 (Apr. 1978), 294–299.

**[4]** HORNIK, K., STINCHCOMBE, M., AND WHITE, H. **Multilayer feedforward networks are universal approximators**. *Neural Netw. 2*, 5 (July 1989), 359–366.

**[5]** PIYUSHIMITA, T., AND ASHISH, K. S. **Data fusion for travel time prediction: a statement of requirements**, vol. vol 22. 1993.

**[6]** TARKO, A., AND ROUPHAIL, N. **Travel time fusion in advance**. *Pacific Rim TransTech Conference: A Ride into the Future* (July 1993).

**[7]** PERKINS, C. E., AND BHAGWAT, P. **Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers**. *SIGCOMM Comput. Commun. Rev. 24*, 4 (Oct. 1994), 234–244.

**[8]** BERKA, S., TIAN, X., AND TARKO, A. **Data fusion algorithm for ADVANCE**. 1995.

**[9]** HASSOUN, M. H. **Fundamentals of Artificial Neural Networks**, 1st ed. MIT Press, Cambridge, MA, USA, 1995.

**[10]** IVAN, J. N., SCHOFER, J. L., KOPPELMAN, F. S., AND MASSONE, L. L. E. **Real-time data fusion for arterial street incident detection using neural networks**. *Transportation Research Record*, 1497 (1995).

**[11]** THOMAS, N. E. **Multi-sensor, multivariate, and multi-class incident detection system for arterial streets**. *Proceedings of the 16th International Symposium on Transportation and Traffic theory (ISTTT), Pergamon, Elsevier* (March 1996).

**[12]** DASARATHY, B. V., AND MEMBER, S. **Sensor Fusion Potential Exploitation - Innovative Architectures and Illustrative Applications**.

**[13]** IVAN, J. N. **Neural network representations for arterial street incident detection data fusion1**. *Transportation Research Part C: Emerging Technologies 5*, 3–4 (1997), 245 – 254.

**[14]** JOHNSON, D., AND MCGEOCH, L. **The Traveling Salesman Problem: A Case Study in Local Optimization**. 1997, pp. 1–103.

**[15]** PARK, V., AND CORSON, M. **A highly adaptive distributed routing algorithm for mobile wireless networks**. In *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE* (Apr 1997), vol. 3, pp. 1405–1413 vol.3.

**[16]** O'MAHONY, D. **Umts: the fusion of fixed and mobile networking**. *Internet Computing, IEEE 2*, 1 (Jan 1998), 49–56.

**[17]** PERKINS, C. E., AND ROYER, E. M. **Ad-hoc On-Demand Distance Vector Routing**. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications* (Washington, DC, USA, 1999), WMCSA '99, IEEE Computer Society, pp. 90–.

**[18]** STEINBERG, A. N., BOWMAN, C. L., AND WHITE, F. E. **Revisions to the JDL data fusion model**. B. V. Dasarathy, Ed., vol. 3719, SPIE, pp. 430–441.

**[19]** CORBERAN, A., MARTI, R., AND ROMERO, A. **Heuristics for the mixed rural postman problem**. *Computers & Operations Research 27*, 2 (2000), 183 – 203.

**[20]** KARP, B., AND KUNG, H. T. **Gpsr: Greedy perimeter stateless routing for wireless networks**. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2000), MobiCom '00, ACM, pp. 243–254.

**[21]** KLEIN, L. A. **Dempster-shafer data fusion at the traffic management center**. *Transportation Research Board 79th Annual Meeting* (March 2000).

**[22]** NOUR-EDDIN, E. F. **Travel time estimation via evidential data fusion**. *Recherche Transport Securite*, 68 (2000), 15 – 30.

**[23]** PEI, G., GERLA, M., AND CHEN, T.-W. **Fisheye state routing: a routing scheme for ad hoc wireless networks**. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on* (2000), vol. 1, pp. 70–74 vol.1.

**[24]** HANSEN, P., AND MLADENOVIĆ, N. **Variable neighborhood search: Principles and applications**. *European Journal of Operational Research 130*, 3 (2001), 449 – 467.

**[25]** PARIKH, C., PONT, M. J., AND JONES, N. B. **Application of dempster-shafer theory in condition monitoring applications: A case study**. *PATTERN RECOGNITION LETTERS 22* (2001), 22–6.

**[26]** SCHOLKOPF, B., AND SMOLA, A. J. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. MIT Press, Cambridge, MA, USA, 2001.

**[27]** DOUCEUR, J. R. **The sybil attack**. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems* (London, UK, 2002), IPTPS '01, Springer-Verlag, pp. 251–260.

**[28]** BACHIR, A., AND BENSLIMANE, A. **A multicast protocol in ad hoc networks inter-vehicle geocast**. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual* (April 2003), vol. 4, pp. 2456–2460 vol.4.

**[29]** BLUM, J., ESKANDARIAN, A., AND HOFFMAN, L. **Mobility management in ivc networks**. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* (June 2003), pp. 150–155.

**[30]** LOCHERT, C., HARTENSTEIN, H., TIAN, J., FUSSLER, H., HERMANN, D., AND MAUVE, M. **A routing strategy for vehicular ad hoc networks in city environments**. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* (June 2003), pp. 156–161.

**[31]** TIAN, J., HAN, L., AND ROTHERMEL, K. **Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks**. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE* (Oct 2003), vol. 2, pp. 1546–1551 vol.2.

**[32]** EL-BAZ, M.-A. **A genetic algorithm for facility layout problems of different manufacturing environments**. *Comput. Ind. Eng. 47*, 2-3 (Nov. 2004), 233–246.

**[33]** FIGUEIREDO, T. H. D. P., LOUREIRO, ANTONIO A F, B., AND WONG, H. C. **Malicious node detection in wireless sensor networks**. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International* (April 2004), pp. 24–.

**[34]** KORKMAZ, G., EKICI, E., ÖZGÜNER, F., AND ÖZGÜNER, U. **Urban multi-hop broadcast protocol for inter-vehicle communication systems**. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks* (New York, NY, USA, 2004), VANET '04, ACM, pp. 76–85.

**[35]** LIU, G., LEE, B.-S., SEET, B.-C., FOH, C. H., WONG, K. J., AND LEE, K.-K. **A routing strategy for metropolis vehicular communications.** In *ICOIN* (2004), H.-K. Kahng, Ed., vol. 3090 of *Lecture Notes in Computer Science*, Springer, pp. 134–143.

**[36]** NADEEM, T., DASHTINEZHAD, S., LIAO, C., AND IFTODE, L. **Trafficview: Traffic data dissemination using car-to-car communication**. *SIGMOBILE Mob. Comput. Commun. Rev. 8*, 3 (July 2004), 6–19.

**[37]** NAMBOODIRI, V., AGARWAL, M., AND GAO, L. **A study on the feasibility of mobile gateways for vehicular ad-hoc networks**. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks* (New York, NY, USA, 2004), VANET '04, ACM, pp. 66–75.

**[38]** PAILA, T., LUBY, M., LEHTONEN, R., ROCA, V., AND WALSH, R. **Flute — file delivery over unidirectional transport**. Internet RFC 3926, October 2004.

**[39]** WU, H., FUJIMOTO, R., GUENSLER, R., AND HUNTER, M. **Mddv: A mobility-centric data dissemination algorithm for vehicular networks**. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks* (New York, NY, USA, 2004), VANET '04, ACM, pp. 47–56.

**[40]** XU, B., OUKSEL, A. M., AND WOLFSON, O. **Opportunistic Resource Exchange in Inter-Vehicle Ad-Hoc Networks**. In *Mobile Data Management* (2004), pp. 4–12.

**[41]** ALSHAER, H., AND HORLAIT, E. **An optimized adaptive broadcast scheme for inter-vehicle communication**. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st* (May 2005), vol. 5, pp. 2840–2844 Vol. 5.

**[42]** DOTZER, F., FISCHER, L., AND MAGIERA, P. **Vars: a vehicle ad-hoc network reputation system**. In *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a* (June 2005), pp. 454–456.

**[43]** GIUDICI, F., AND PAGANI, E. **Spatial and traffic-aware routing (star) for vehicular systems.** In *HPCC* (2005), L. T. Yang, O. F. Rana, B. D. Martino, and J. Dongarra, Eds., vol. 3726 of *Lecture Notes in Computer Science*, Springer, pp. 77–86.

**[44]** LITTLE, T., AND AGARWAL, A. **An information propagation scheme for vanets**. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE* (Sept 2005), pp. 155–160.

**[45]** LOCHERT, C., MAUVE, M., FUSSLER, H., AND HARTENSTEIN, H. **Geographic routing in city scenarios**. *SIGMOBILE Mob. Comput. Commun. Rev. 9*, 1 (Jan. 2005), 69–72.

**[46]** SANTOS, R. A., EDWARDS, A., EDWARDS, R. M., AND SEED, N. L. **Performance evaluation of routing protocols in vehicular ad-hoc networks**. *Int. J. Ad Hoc Ubiquitous Comput. 1*, 1/2 (Nov. 2005), 80–91.

**[47]** CAPRA, L., AND MUSOLESI, M. **Autonomic trust prediction for pervasive systems.** In *AINA (2)* (2006), IEEE Computer Society, pp. 481–488.

**[48]** GOVINDAN, Y.-J. K. R., KARP, B., AND SHENKER, S. **Lazy cross-link removal for geographic routing**. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2006), SenSys '06, ACM, pp. 112–124.

**[49]** KORKMAZ, G., EKICI, E., AND ÖZGÜNER, F. **An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems**. In *IEEE International Conference on Communications (ICC)* (2006), vol. 1, IEEE, pp. 423–428.

**[50]** LEINM, T., MAIH, C., BOX, P. O., SCHOCH, E., KARGL, F., LEINMUELLER, T., AND MAIHOEFER, C. **Improved Security in Geographic Ad hoc Routing through**. *Proceedings of the 3rd international workshop on Vehicular ad hoc networks* (2006), 57–66.

**[51]** PATWARDHAN, A., JOSHI, A., FININ, T., AND YESHA, Y. **A data intensive reputation management scheme for vehicular ad hoc networks**. In *Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on* (July 2006), pp. 1–8.

**[52]** SUN, W., YAMAGUCHI, H., YUKIMASA, K., AND KUSUMOTO, S. **Gvgrid: A qos routing protocol for vehicular ad hoc networks**. In *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on* (June 2006), pp. 130–139.

**[53]** SZARVAS, M., SAKAI, U., AND OGATA, J. **Real-time pedestrian detection using lidar and convolutional neural networks**. In *Intelligent Vehicles Symposium, 2006 IEEE* (2006), pp. 213–218.

**[54]** XIAO, B., YU, B., AND GAO, C. **Detection and localization of sybil nodes in VANETs**. *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks - DIWANS '06* (2006), 1.

**[55]** YIN, X., LIU, W., AND GUAN, L. **Research on automatic incident detection algorithm based on fusion of freeway mainline information and toll collection information**. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on* (Oct 2006), vol. 4, pp. 2915–2919.

**[56]** BASIR, O., AND YUAN, X. **Engine fault diagnosis based on multi-sensor information fusion using dempster-shafer evidence theory**. *Inf. Fusion 8*, 4 (Oct. 2007), 379–386.

**[57]** DING, Y., WANG, C., AND XIAO, L. **A static-node assisted adaptive routing protocol in vehicular networks**. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks* (New York, NY, USA, 2007), VANET '07, ACM, pp. 59–68.

**[58]** GERLACH, M. **Trust for vehicular applications**. In *Autonomous Decentralized Systems, 2007. ISADS '07. Eighth International Symposium on* (March 2007), pp. 295–304.

**[59]** HWANG, J. P., CHO, S. E., RYU, K. J., PARK, S., AND KIM, E. **Multi-classifier based lidar and camera fusion**. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE* (Sept 2007), pp. 467–472.

**[60]** JERBI, M., SENOUCI, S.-M., MERAIHI, R., AND GHAMRI-DOUDANE, Y. **An improved vehicular ad hoc routing protocol for city environments**. In *Communications, 2007. ICC '07. IEEE International Conference on* (June 2007), pp. 3972–3979.

**[61]** LEE, K., HAERRI, J., LEE, U., AND GERLA, M. **Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios**. In *Globecom Workshops, 2007 IEEE* (Nov 2007), pp. 1–10.

**[62]** MERWE, J. V. D., DAWOUD, D., AND MCDONALD, S. **A survey on peer-to-peer key management for mobile ad hoc networks**. *ACM Computing Surveys 39*, 1 (Apr. 2007), 1–es.

**[63]** NAMBOODIRI, V., AND GAO, L. **Prediction-based routing for vehicular ad hoc networks**. *Vehicular Technology, IEEE Transactions on 56*, 4 (July 2007), 2332–2345.

**[64]** NAUMOV, V., AND GROSS, T. **Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks**. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (May 2007), pp. 1919–1927.

**[65]** RAYA, M., AND HUBAUX, J.-P. **Securing vehicular ad hoc networks**. *J. Comput. Secur. 15*, 1 (Jan. 2007), 39–68.

**[66]** WISITPONGPHAN, N., TONGUZ, O., PARIKH, J., MUDALIGE, P., BAI, F., AND SADEKAR, V. **Broadcast storm mitigation techniques in vehicular ad hoc networks**. *Wireless Communications, IEEE 14*, 6 (December 2007), 84–94.

**[67]** ATALAH, K., MACIAS, E., AND SUAREZ, A. **A Proactive Horizontal Handover Algorithm Based on RSSI Supported by a New Gradient Predictor**. *Ubiquitous Computing and Communication Journal* (2008).

**[68]** CHANG, W.-R., LIN, H.-T., AND CHEN, B.-X. **TrafficGather: An Efficient and Scalable Data Collection Protocol for Vehicular Ad Hoc Networks**. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE* (2008), pp. 365–369.

**[69]** COMESAFETY. **D31 European ITS Communication Architecture**. *Technical report, Information Society Technologies* (2008).

**[70]** DE LA OLIVA, A., BANCHS, A., SOTO, I., MELIA, T., AND VIDAL, A. **An overview of ieee 802.21: media-independent handover services**. *Wireless Communications, IEEE* (Aug 2008).

**[71]** GRELLIER, É. **Optimization of vehicle routing in the context of reverse logistics: modeling and resolution using hybrid methods**. 2008.

**[72]** KASSAR, M., KERVELLA, B., AND PUJOLLE, G. **An overview of vertical handover decision strategies in heterogeneous wireless networks**. *Computer Communications* (June 2008).

**[73]** KUMAR, R., AND RAO, S. **Directional Greedy Routing Protocol (DGRP) in Mobile Ad-Hoc Networks**. In *Information Technology, 2008. ICIT '08. International Conference on* (Dec 2008), pp. 183–188.

**[74]** LABERTEAUX, K. P., HAAS, J. J., AND HU, Y.-C. **Security certificate revocation list distribution for vanet**. In *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking* (New York, NY, USA, 2008), VANET '08, ACM, pp. 88–89.

**[75]** LEE, K., LE, M., HARRI, J., AND GERLA, M. **LOUVRE: Landmark Overlays for Urban Vehicular Routing Environments**. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th* (Sept 2008), pp. 1–5.

**[76]** PAPADIMITRATOS, P., BUTTYAN, L., HOLCZER, T., SCHOCH, E., FREUDIGER, J., RAYA, M., MA, Z., KARGL, F., KUNG, A., AND HUBAUX, J. P. **Secure vehicular communication systems: Design and architecture**. *Comm. Mag. 46*, 11 (Nov. 2008), 100–109.

**[77]** PAPADIMITRATOS, P. P., MEZZOUR, G., AND HUBAUX, J.-P. **Certificate revocation list distribution in vehicular communication systems**. In *Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking* (New York, NY, USA, 2008), VANET '08, ACM, pp. 86–87.

**[78]** RAYA, M., PAPADIMITRATOS, P., GLIGOR, V., AND HUBAUX, J.-P. **On data-centric trust establishment in ephemeral ad hoc networks**. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (April 2008).

**[79]** SCHNAUFER, S., AND EFFELSBERG, W. **Position-based unicast routing for city scenarios**. In *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a* (June 2008), pp. 1–8.

**[80]** ZHANG, C., LU, R., LIN, X., HO, P.-H., AND SHEN, X. **An efficient identity-based batch verification scheme for vehicular sensor networks**. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (April 2008).

**[81]** ZHAO, J., AND CAO, G. **VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks**. *Vehicular Technology, IEEE Transactions on 57*, 3 (May 2008), 1910–1922.

**[82]** ABUELELA, M., AND OLARIU, S. **Automatic incident detection in vanets: A bayesian approach**. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th* (April 2009), pp. 1–5.

**[83]** BOUASSIDA, M. S., GUETTE, G., SHAWKY, M., AND DUCOURTHIAL, B. **Sybil nodes detection based on received signal strength variations within vanet.** *I. J. Network Security 9*, 1 (2009), 22–33.

**[84]** ETSI TR. **Vehicular Communications** . 1–81.

**[85]** HAAS, J. J., HU, Y.-C., AND LABERTEAUX, K. P. **Design and analysis of a lightweight certificate revocation mechanism for vanet**. In *Proceedings of the Sixth ACM International Workshop on VehiculAr InterNETworking* (New York, NY, USA, 2009), VANET '09, ACM, pp. 89–98.

**[86]** HUANG, X., AND FANG, Y. **Performance study of node-disjoint multipath routing in vehicular ad hoc networks**. *Vehicular Technology, IEEE Transactions on 58*, 4 (May 2009), 1942–1950.

**[87]** JERBI, M., SENOUCI, S.-M., RASHEED, T., AND GHAMRI-DOUDANE, Y. **Towards efficient geographic routing in urban vehicular networks**. *Vehicular Technology, IEEE Transactions on 58*, 9 (2009), 5048–5059.

**[88]** LI, T., LI, Y., AND LIAO, J. **A contention-based routing protocol for vehicular ad hoc networks in city environments**. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on* (June 2009), pp. 482–487.

**[89]** NOUR-EDDIN, E. F., LAWRANCE, A., AND OLIVIER, D. M. **Improving travel time estimates from inductive loop and toll collection data with dempster-shafer data fusion**. *Transportation Research Record: Journal of the Transportation Research Board*, 2129 (2009), 73 – 80.

**[90]** NZOUONTA, J., RAJGURE, N., WANG, G., AND BORCEA, C. **VANET Routing on City Roads Using Real-Time Vehicular Traffic Information**. *IEEE Transactions on Vehicular Technology 58*, 7 (Sept. 2009), 3609–3626.

**[91]** OLIVERA, J. A., CORTAZAR, I., PINART, C., SANTOS, A. L., AND LEQUERICA, I. **Vanba: A simple handover mechanism for transparent, always-on v2v communications.** In *VTC Spring* (2009), IEEE.

**[92]** SALHI, I., CHERIF, M. O., AND SENOUCI, S. M. **A new architecture for data collection in vehicular networks**. In *Proceedings of the 2009 IEEE international conference on Communications* (Piscataway, NJ, USA, 2009), ICC'09, IEEE Press, pp. 2705–2710.

**[93]** TOROYAN, T. **Global status report on road safety.** *Injury prevention : journal of the International Society for Child and Adolescent Injury Prevention 15*, 4 (2009), 286.

**[94]** WANG, L., AND BINET, D. **MADM-based network selection in heterogeneous wireless networks: A simulation study**. *International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology* (May 2009).

**[95]** ZHU, H., LU, R., SHEN, X., AND LIN, X. **Security in service-oriented vehicular networks**. *Wireless Communications, IEEE 16*, 4 (Aug 2009), 16–22.

**[96]** BRICKLEY, O., KOUBEK, M., REA, S., AND PESCH, D. **A network centric simulation environment for calm-based cooperative vehicular systems**. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques* (ICST, Brussels, Belgium, Belgium, 2010), SIMUTools '10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 75:1–75:10.

**[97]** CHEN, C., ZHANG, J., COHEN, R., AND HO, P.-H. **A trust modeling framework for message propagation and evaluation in vanets**. In *Information Technology Convergence and Services (ITCS), 2010 2nd International Conference on* (2010), pp. 1–8.

**[98]** ESPOSITO, F., VEGNI, A., MATTA, I., AND NERI, A. **On modeling speed-based vertical handovers in vehicular networks: "dad, slow down, i am watching the movie"**. In *GLOBECOM Workshops (GC Wkshps), IEEE* (Dec 2010).

**[99]** ETSI EN 302 665. **Communications Architecture**. 1–44.

**[100]** ETSI TS. **Vehicular Communications ; Basic Set of Applications ; Part 1 : Functional Requirements**. 1–60.

**[101]** FADLULLAH, Z., TALEB, T., VASILAKOS, A., GUIZANI, M., AND KATO, N. **Dtrab: Combating against attacks on encrypted protocols through traffic-feature analysis**. *Networking, IEEE/ACM Transactions on 18*, 4 (Aug 2010), 1234–1247.

**[102]** IBRI, S., DRIAS, H., AND NOURELFATH, M. **A parallel hybrid ant-tabu algorithm for integrated emergency vehicle dispatching and covering problem**. *Int. J. Innov. Comput. Appl. 2*, 4 (Nov. 2010), 226–236.

**[103]** KAMINI, K., AND KOUMAR, R. **VANET Parameters and Applications : A Review**. *Global Journal of Computer Science and Technology 10*, 7 (2010), 72–77.

**[104]** LEE, K., UICHIN, L., AND GERLA, M. **Survey of Routing Protocols in Vehicular Ad Hoc Networks**. *Information Science Reference* (2010), 149–151.

**[105]** LEE, K. C., CHENG, P.-C., AND GERLA, M. **GeoCross: A geographic routing protocol in the presence of loops in urban scenarios**. *Ad Hoc Networks 8*, 5 (2010), 474 – 488. Vehicular Networks.

**[106]** LIN, Y., CHEN, Y., AND LEE, S. **Routing protocols in vehicular ad hoc networks: A survey and future perspectives**. *J. Inf. Sci. Eng. 26*, 3 (2010), 913–932.

**[107]** MINHAS, U. F., ZHANG, J., TRAN, T., AND COHEN, R. **Towards expanded trust management for agents in vehicular ad-hoc networks**. *International Journal of Computational Intelligence Theory and Practice (IJCITP) 5*, 1 (2010).

**[108]** NOWATKOWSKI, M., AND OWEN, H. **Scalable certificate revocation list distribution in vehicular ad hoc networks**. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* (Dec 2010), pp. 54–58.

**[109]** NOWATKOWSKI, M., AND OWEN, H. L. **Certificate revocation list distribution in vanets using most pieces broadcast**. In *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the* (March 2010), pp. 238–241.

**[110]** PRASANTH, K., DURAISWAMY, K., JAYASUDHA, K., AND CHANDRASEKAR, C. **Improved packet forwarding approach in vehicular ad hoc networks using rdgr algorithm**. *CoRR abs/1003.5437* (2010).

**[111]** BAKHOUYA, M., GABER, J., AND LORENZ, P. **An adaptive approach for information dissemination in vehicular ad hoc networks**. *Journal of Network and Computer Applications 34*, 6 (2011), 1971 – 1978. Control and Optimization over Wireless Networks.

**[112]** DAEINABI, A., RAHBAR, A. G. P., AND KHADEMZADEH, A. **VWCA: An efficient clustering algorithm in vehicular ad hoc networks**. *Journal of Network and Computer Applications 34*, 1 (2011), 207 – 222.

**[113]** GROVER, J., GAUR, M. S., LAXMI, V., AND PRAJAPATI, N. K. **A Sybil Attack Detection Approach using Neighboring Vehicles in VANET**. 151–158.

**[114]** HAAS, J., HU, Y.-C., AND LABERTEAUX, K. **Efficient certificate revocation list organization and distribution**. *Selected Areas in Communications, IEEE Journal on 29*, 3 (March 2011), 595–604.

**[115]** NOUR-EDDIN, E. F., HENRY, L., AND AJEESH, K. **Data fusion in intelligent transportation systems: Progress and challenges - a survey**. *Information Fusion 12*, 1 (2011), 4 – 10. Special Issue on Intelligent Transportation Systems.

**[116]** POP, P., SITAR, C., ZELINA, I., LUPSE, V., AND CHIRA, C. **Heuristic algorithms for solving the generalized vehicle routing problem**. *International Journal of Computers Communications & Control 6*, 1 (2011), 158–165.

**[117]** REMY, G., SENOUCI, S.-M., JAN, F., AND GOURHANT, Y. **LTE4V2X: LTE for a Centralized VANET Organization**. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE* (2011), pp. 1–6.

**[118]** RUJ, S., CAVENAGHI, M., HUANG, Z., NAYAK, A., AND STOJMENOVIC, I. **On data-centric misbehavior detection in vanets**. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE* (Sept 2011), pp. 1–5.

**[119]** RUJ, S., CAVENAGHI, M. A., HUANG, Z., NAYAK, A., AND STOJMENOVIC, I. **On Data-Centric Misbehavior Detection in VANETs**. *2011 IEEE Vehicular Technology Conference (VTC Fall)* (Sept. 2011), 1–5.

**[120]** RYU, M.-W., CHA, S.-H., AND CHO, K.-H. **A vehicle communication routing algorithm considering road characteristics and 2-hop neighbors in urban areas**. *The Journal of Korean Institute of Communications and Information Sciences 36*, 5B (2011), 464–470.

**[121]** RYU, M.-W., CHA, S.-H., KOH, J.-G., KANG, S., AND CHO, K.-H. **Position-based routing algorithm for improving reliability of inter-vehicle communication.** *TIIS 5*, 8 (2011), 1388–1403.

**[122]** SUMRA, I. A., AHMAD, I., HASBULLAH, H., AND ISKANDAR, B. S. **Classes of attacks in vanet**. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International* (April 2011), pp. 1–5.

**[123]** WAGH, A., LI, X., AND QIAO, C. **Human centric data fusion in Vehicular Cyber-Physical Systems**. *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (Apr. 2011), 684–689.

**[124]** ZHANG, J. **A survey on trust management for vanets**. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on* (March 2011), pp. 105–112.

**[125]** ABROUGUI, K., BOUKERCHE, A., AND RAMADAN, H. **Performance evaluation of an efficient fault tolerant service discovery protocol for vehicular networks**. *Journal of Network and Computer Applications 35*, 5 (2012), 1424 – 1435. Service Delivery Management in Broadband Networks.

**[126]** BABER, A., AMJAD, F., AND ZOU, C. **Optimal roadside units placement in urban areas for vehicular networks**. *2012 IEEE Symposium on Computers and Communications (ISCC) 0* (2012), 000423–000429.

**[127]** BERNSEN, J., AND MANIVANNAN, D. **RIVER: A reliable inter-vehicular routing protocol for vehicular ad hoc networks** . *Computer Networks 56*, 17 (2012), 3795 – 3807.

**[128]** BHATTASALI, T., AND CHAKI, R. **A survey of recent intrusion detection systems for wireless sensor network**. *CoRR abs/1203.0240* (2012).

**[129]** CALAFATE, C. T., FORTINO, G., FRITSCH, S., MONTEIRO, J., CANO, J.-C., AND MANZONI, P. **An efficient and robust content delivery solution for ieee 802.11p vehicular environments**. *Journal of Network and Computer Applications 35*, 2 (2012), 753 – 762.

**[130]** CHA, S.-H., LEE, K.-W., AND CHO, H.-S. **Grid-based predictive geographical routing for inter-vehicle communication in urban areas.** *IJDSN 2012* (2012).

**[131]** CHANG, S., QI, Y., ZHU, H., ZHAO, J., AND SHEN, X. **Footprint: Detecting Sybil attacks in urban vehicular networks**. *IEEE Transactions on Parallel and Distributed Systems 23*, 6 (2012), 1103–1114.

**[132]** CHERIF, M. O., AND SENOUCI, S.-M. **A geographical self-organizing approach for vehicular networks.** *JCM 7*, 12 (2012), 885–898.

**[133]** DATTA, S., DHAR, S., BERA, R. N., AND RAY, A. **ANP based vertical handover algorithm for vehicular communication**. *International Conference on Recent Advances in Information Technology(RAIT)* (2012).

**[134]** FATHY, M., GHOLAMALITABARFIROUZJAEE, S., AND RAAHEMIFAR, K. **Improving QoS in VANET Using MPLS**. *Procedia Computer Science 10*, 0 (2012), 1018 – 1025. ANT 2012 and MobiWIS 2012 .

**[135]** FOGUE, M., GARRIDO, P., MARTINEZ, F. J., CANO, J.-C., CALAFATE, C. T., AND MANZONI, P. **Evaluating the impact of a novel message dissemination scheme for vehicular networks using real maps**. *Transportation Research Part C: Emerging Technologies 25*, 0 (2012), 61 – 80.

**[136]** GAZDAR, T., RACHEDI, A., BENSLIMANE, A., AND BELGHITH, A. **A distributed advanced analytical trust model for vanets**. In *Global Communications Conference (GLOBECOM), 2012 IEEE* (2012), pp. 201–206.

**[137]** GOLESTAN, K., SEIFZADEH, S., KAMEL, M., KARRAY, F., AND SATTAR, F. **Vehicle localization in vanets using data fusion and v2v communication**. In *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications* (New York, NY, USA, 2012), DIVANet '12, ACM, pp. 123–130.

**[138]** KAKKASAGERI, M., AND MANVI, S. **Multiagent driven dynamic clustering of vehicles in VANETs**. *Journal of Network and Computer Applications 35*, 6 (2012), 1771 – 1780.

**[139]** LIU, X., AND DATTA, A. **Modeling context aware dynamic trust using hidden markov model**. In *AAAI* (2012).

**[140]** MARQUEZ-BARJA, J., CALAFATE, C. T., CANO, J.-C., AND MANZONI, P. **MACHU: A novel vertical handover algorithm for vehicular environments**. *Wireless Telecommunications Symposium* (Apr. 2012).

**[141]** MEHAR, S., SENOUCI, S.-M., AND REMY, G. **Dissemination protocol for heterogeneous cooperative vehicular networks.** In *Wireless Days* (2012), IEEE, pp. 1–6.

**[142]** MERSHAD, K., ARTAIL, H., AND GERLA, M. **ROAMER: Roadside Units as message routers in VANETs**. *Ad Hoc Networks 10*, 3 (2012), 479 – 496.

**[143]** PALOMAR, E., DE FUENTES, J. M., GONZALEZ-TABLAS, A. I., AND ALCAIDE, A. **Hindering false event dissemination in VANETs with proof-of-work mechanisms**. *Transportation Research Part C: Emerging Technologies 23*, 0 (2012), 85 – 97. Data Management in Vehicular Networks.

**[144]** REMY, G., SENOUCI, S.-M., JAN, F., AND GOURHANT, Y. **LTE4V2X - Collection, dissemination and multi-hop forwarding.** In *ICC* (2012), IEEE, pp. 120–125.

**[145]** SABAHI, F. **Impact of Threats on Vehicular Adhoc Network Security**. *International Journal of Computer Theory and Engineering Vol 4*, 5 (2012), 840–842.

**[146]** TOUTOUH, J., GARCÍA-NIETO, J., AND ALBA, E. **Intelligent OLSR routing protocol optimization for vanets**. *IEEE T. Vehicular Technology 61*, 4 (2012), 1884–1894.

**[147]** ALTAYEB, M., AND MAHGOUB, I. **A Survey of Vehicular Ad hoc Networks Routing Protocols**. *International Journal of Innovation and Applied Studies 3*, 3 (2013), 829–846.

**[148]** ANWAR, F., MASUD, M. H., AND LATIF, S. **Fuzzy Logic based Handoff Latency Reduction Mechanism in Layer 2 of Heterogeneous Mobile IPv6 Networks**. *IOP Conference Series: Materials Science and Engineering* (2013).

**[149]** CHERIF, M. O., SENOUCI, S.-M., AND DUCOURTHIAL, B. **Efficient data dissemination in cooperative vehicular networks**. *Wireless Communications and Mobile Computing 13*, 12 (2013), 1150–1160.

**[150]** DESHPANDE, S. G. **Classification of Security attack in Vehicular Adhoc network: A survey**. *IJETTCS Vol 2*, 2 (2013), 371–377.

**[151]** FERRUS, R., SALLENT, O., BALDINI, G., AND GORATTI, L. **LTE: the technology driver for future public safety communications**. *Communications Magazine, IEEE 51*, 10 (October 2013), 154–161.

**[152]** GOLESTAN, K., KARRAY, F., AND KAMEL, M. **High level information fusion through a fuzzy extension to multi-entity bayesian networks in vehicular ad-hoc networks**. In *Information Fusion (FUSION), 2013 16th International Conference on* (July 2013), pp. 1180–1187.

**[153]** INSTITUTE OF ELECTICAL AND ELECTRONICS ENGINEERS. **IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages (IEEE Std 1609.2-2013)**, vol. 2013. 2013.

**[154]** KHALEGHI, B., KHAMIS, A., AND KARRAY, F. O. **Multisensor data fusion: A review of the state-of-the-art**. *Information Fusion 14*, 1 (Aug. 2013), 28–44.

**[155]** MEHAR, S., AND SENOUCI, S.-M. **An optimization location scheme for electric charging stations**. *IEEE SaCoNet* (2013), 1–5.

**[156]** WAGH, A., LI, X., SUDHAAKAR, R., ADDEPALLI, S., AND QIAO, C. **Data fusion with flexible message composition in Driver-in-the-Loop vehicular CPS**. *Ad Hoc Networks 11*, 7 (Sept. 2013), 2083–2095.

**[157]** WANG, S.-S., AND LIN, Y.-S. **PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks**. *Computer Communications 36*, 2 (2013), 170 – 179.

**[158]** XIANG, Y., LIU, Z., LIU, R., SUN, W., AND WANG, W. **GeoSVR: A map-based stateless VANET routing**. *Ad Hoc Networks 11*, 7 (2013), 2125 – 2135. Theory, Algorithms and Applications of Wireless Networked Robotics Recent Advances in Vehicular Communications and Networking.

**[159]** ZHANG, J., CHEN, C., AND COHEN, R. **Trust modeling for message relay control and local action decision making in vanets**. *Sec. and Commun. Netw. 6*, 1 (Jan. 2013), 1–14.

**[160]** ZHAO, W., ZHANG, L., ZHU, W., AND ZHAO, Y. **Multilevel cluster-based information fusion in vehicle ad hoc networks**. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing* (Aug 2013), pp. 1357–1362.

**[161]** ABDELGADER M S, A., AND LENAN, W. **The Physical Layer of the IEEE 802.11p WAVE Communication Standard : The Specifications and Challenges**.

**[162]** DUA, A., KUMAR, N., AND BAWA, S. **A systematic review on routing protocols for Vehicular Ad Hoc Networks**. *Vehicular Communications 1*, 1 (2014), 33–52.

**[163]** MALANDRINO, F., BORGIATTINO, C., CASETTI, C., CHIASSERINI, C.-F., FIORE, M., AND SADAO, R. **Verification and inference of positions in vehicular networks through anonymous beaconing**. *IEEE Transactions on Mobile Computing 13*, 10 (Oct 2014), 2415–2428.

**[164]** RANA, S., RANA, S., AND PUROHIT, K. C. **A review of various routing protocols in vanet**. *International Journal of Computer Applications 96*, 18 (June 2014), 28–35. Full text available.

**[165]** SEDJELMACI, H., AND SENOUCI, S. **A new intrusion detection framework for vehicular networks**. In *Communications (ICC), 2014 IEEE International Conference on* (June 2014), pp. 538–543.

**[166]** SEDJELMACI, H., SENOUCI, S.-M., AND ALI, A.-R. M. **An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks**. *Internet of Things Journal, IEEE 1*, 6 (Dec 2014), 570–577.

**[167]** SOARES, V. N., RODRIGUES, J. J., AND FARAHMAND, F. **GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks**. *Information Fusion 15*, 0 (2014), 102 – 113. Special Issue: Resource Constrained Networks.

**[168]** ABBAS, N., AND SAADE, J. J. **A Fuzzy Logic Based Approach for Network Selection in WLAN / 3G Heterogeneous Network**. *CCNC, IEEE* (2015).

**[169]** BEHERA, P. K. **Optimization of Vertical Handoff Performance Parameters in Heterogeneous Wireless Networks**. *International Journal of Modern Engineering Research (IJMER)*.

**[170]** QOS, V. http://www.voip-info.org/wiki/view/QoS.

**[171]** WARDSAUTO. **World vehicle population tops 1 billion units**. http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/nationaltrans-portationstatistics/html/table0409.html.

**[172]** WILLIAMS, B. J. **Troubleshooting IP Video Quality of Service**.

# List of Figures

# List of Tables

# GLOSSARY

- **2-Opt:** is an optimization algorithm proposed by Croes in 1958 for local search to solve the traveling salesman problem. It is based on the exchange between possible vertices in the circuit to ameliorate the cost. 42, 44–46, 50
- **2G:** is the second generation of mobile telecommunications technology. 5, 18, 84
- **3G:** is the third generation of mobile telecommunications technology. 5, 18, 21, 84, 91, 92, 98, 101
- **4G:** is the fourth generation of mobile telecommunications technolog, also called LTE. 5, 18, 82, 91, 92
- **BS:** Base Station. 7, 10
- **C-ITS:** Cooperative Intelligent Transport Systems and Services. 5, 13
- **CALM:** Communication Access for Land Mobiles. 18, 19, 22, 82, 93, 102, 107
- **CAM:** Cooperative Awareness Message. 9
- **CRL:** Certificate Revocation List. 26, 28–30, 32–35
- **CSMA:** the acronym of Carrier Sence Multiple Access, it consists on a set of protocols for access to the communication medium and uses a Request To Send (RTS) to solicit the access and a Clear To Send (CTS) to notify that the medium is clear. 9
- **DoS:** Denial of Service. 6, 23, 38, 51–53, 59, 61, 65, 72, 74, 76, 136
- **DSRC:** Dedicated Short Range Communication. 5, 18, 35, 36
- **ECC:** Elliptic Curve Cryptography. 26–28
- **eNodeB:** evolved Node B is a base station of 3G network. It provides a cellular network with a radius up to several kilometers in interurban areas. 7, 10
- **ETSI:** European Telecommunications Standard Institute. 18, 102
- **G5:** a spectrum allocated in Europe for road safety and traffic efficiency applications.. 1
- **GPS:** Global Positioning System. 5, 12, 14, 17, 21
- **I2V:** Infrastructure-to-Vehicle. 7
- **IDS:** Intrusion Detection System. 22, 36, 38, 58, 76, 77

- **IEEE:** Electrical and Electronics Engineers. 1, 7, 21, 22, 25–30, 35, 63, 82, 84, 89, 90, 98, 101
- **IPDS:** Intrusion Detection and Prevention System. 2, 3, 51, 52, 61, 63–65, 74–78, 84, 106, 121, 122
- **ITS:** Intelligent Transportation System. 1, 9, 14–19, 40, 78, 82, 93, 102, 103, 105, 107
- **LTE:** Long Term Evolution. 1, 7
- **MANET:** Mobile Ad-hoc Network. 7
- **OBU:** On-Board Unit. 1, 5, 7, 83, 89, 90
- **PKI:** Public Key Infrastructure. 26
- **QoS:** Quality of Service. 3, 4, 8, 13, 18, 21, 32, 93, 96–99, 101
- **RSU:** Road Side Unit. 7, 10, 16, 32–35, 38, 39, 42, 52, 129–134, 137
- **S-GyTAR:** Secure Greedy Traffic Aware Routing Protocol. 2, 3, 51, 52, 60, 78, 106
- **SDS:** Security Data Store. 25, 26, 30
- **SNF:** store and forward, which is an alternative used in gready routing protocols when no neighbor is available to forward a packet and is based on the technique of carrying the packet by a node until finding a good forwarder. 11, 56
- **String Exchange:** is an optimization algorithm for global search which aims to exchange arcs between two or more circuits if a cost is ameliorated. 42, 45, 50
- **UMTS:** Universal Mobile Telecommunications System. 7, 18, 84
- **V2I:** Vehicle-to-Infrastructure. 1, 2, 5–7, 23, 24, 32–34, 36, 40, 82, 84, 89, 130
- **V2V:** Vehicle-to-Vehicle. 1, 2, 5–7, 21, 23, 24, 32–36, 40, 82, 84, 89, 130
- **VANET:** Vehicular Ad-hoc Network. 6, 7, 9, 11, 12, 14, 16, 20–22, 29, 37, 72, 73
- **VNS:** Variable Neighborhood Search. 42, 44, 45, 50
- **WAVE:** Wireless Access in Vehicular Environment, which is the amercan normalization of the European G5. 22, 25, 26, 35, 36

# I

# APPENDIX

# A

# GAME THEORY DEPLOYMENT FOR ATTACK PREDICTION IN A VEHICULAR NETWORK

This work is the result of a collaboration with another member of our team work in which we studied the applicability of game theory on the vehicular network attacks prediction. Unlike the intrusion detection and prevention system proposed in this thesis which relies on Kalman filter, in this work, we considered the use of game theory to model interactions between attackers and monitor agents to predict the future behavior of vehicles. In the rest of this chapter, we detail the type of attack to be faced, the considered monitoring architecture and different strategies deployed by each player in the game theory.

## A.1/ INTRODUCTION

In this chapter, we target to present a new way we have developed based on game theory to protect data exchanged between vehicles and predict attackers in an hybrid network that contains a set of deployed infrastructure (RSU). However, this study mainly concerns safety application security and more precisely diffused alerts in case an abnormal event is detected. In fact and as stated in the state of the art of this thesis, protecting safety applications in a vehicular network is very critical since they are attractive targets due to the relevant information that vehicles can manage. So, a false alert sent to neighbors by a malicious node may cause several damages such as accidents or even time waste. The proposed intrusion detection and prevention schema relies on game theory to predict the malicious behavior of an attacker and categorizes the monitored vehicles into an appropriate list according to their future attack severity. The attack-defense problem is formulated as a Bayesian game between the attacker and an RSU, and the future states' prediction of a suspected behavior is determined thanks to Bayesian Nash Equilibrium (BNE). Furthermore, in this research, we aim to detect one of the most dangerous attacks that could occur in vehicular network safety applications, which is a False alert's generation attack. Therefore, we propose a set of detection rules to identify such malicious behavior when it occurs. Simulation results conducted on NS3.17 simulator prove that the proposed malicious behavior's prediction technique has the ability to predict with high accuracy a future malicious behavior of a monitored vehicle even when the number of vehicles is high (i.e. scalable). In addition, our detection and prevention schema requires a low overhead to detect and prevent the attackers' occurrence. The rest of this chapter is organized as follows: in section A.2, we present the network architecture we aim to secure together with a description of some relevant types of false alerts and the policy used to detect them. In section A.3, we detail our malicious behavior prediction mechanism. Section A.4 provide a performance analysis of the proposed mechanism and results discussion. Finally, a conclusion is

provided in section A.5.

## A.2/ BACKGROUND & DETECTION POLICY

In this section, we describe, first, the network architecture that we attempt to secure. Second, we present the attack that we aim to detect and the proposed corresponding detection rules.

### A.2.1/ NETWORK ARCHITECTURE

The network architecture we attempt to secure is illustrated in figure A.1. This one is similar to the one used in the literature [80] [95] where they assume vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications. We use a dissemination protocol to disseminate an alert message to alert other vehicles when an event is detected. We note that, forwarders are selected according to their capability to provide a higher progress towards the destination (i.e. geographic area or RSU) [95]. In this research, we focus on three alerts' categories: Post Crash Notification (PCN), Road Hazard Condition Notification (RHCN) and Stopped/Slow Vehicle Advisor (SVA).

To identify a malicious node, each vehicle has the ability to monitor its neighbors and is called in the following *monitoring vehicle*. However, when a malicious behavior is detected, the monitoring vehicle sends its feedback (i.e. monitoring list) to the RSU as explained bellow and illustrated in figure A.1. We assume that RSUs are connected to the application server through wired communication using a Transport Layer Security (TLS) protocol. This latter aims to provide authentication and encryption in order to transmit sensitive data [101].

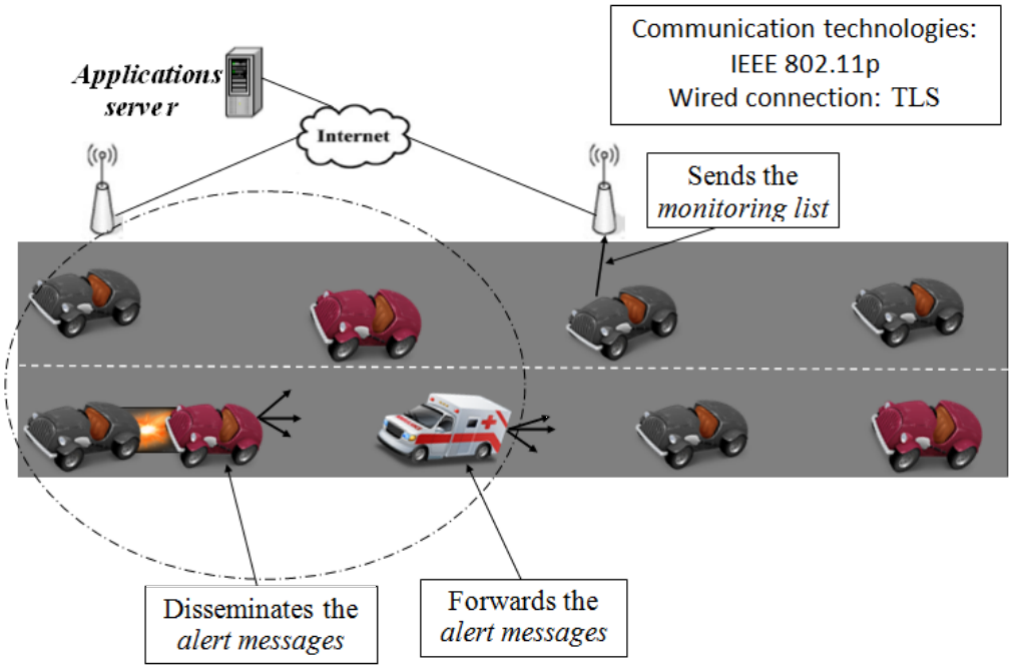



Figure A.1 – Considered vehicular network architecture

### A.2.2/ ATTACK MODEL & DETECTION POLICY

To detect the malicious vehicle with a high accuracy, we apply a mutual monitoring concept, i.e. each vehicle monitors its neighbors in a promiscuous mode. This means that, the monitoring vehicle can hear all the packets that circulate within its radio range. The monitoring process aims to apply a set of rules to model a normal behavior of a vehicle after the message alert is raised, which are explained as follows:

When a vehicle disseminates an alert message about an event, the vehicles that are located within its radio range monitor its behavior to detect if this alert is correct. According to the authors in [118], when a node generates one of these alerts cited above, its speed should decrease and it should change lane. Hence, the monitoring vehicle analyzes the speed/lane of a monitored vehicle at both the moment and after the alert generation. As illustrated in figure A.2 (a), the vehicle $vm$ (attacker) sends periodically to its neighbor $v1$ Beacon messages that include its position coordinates $(x^{vm}_{beacon}, y^{vm}_{beacon})$, time when the message was generated $(t_{beacon})$ and its current speed $(vm_{speed-beacon})$. Furthermore, when the alert message is sent $vm$, it includes information about the alert type (PCN, RHCN or SVA), its new position coordinates $(x^{vm}_{alert}, y^{vm}_{alert})$, time when the alert was generated $(t_{alert})$ and its speed $(vm_{speed-alert})$. In this case, $v1$ considered in this example as the monitoring node, computes the speed traveled between the alert generation and a subsequent beacon message as shown in equation A.1 where $d = \sqrt{(x^{vm}_{beacon} - x^{vm}_{alert})^2 + (y^{vm}_{beacon} - y^{vm}_{alert})^2}$.

$$Speed_{vm} = \frac{d}{t_{beacon} - t_{alert}} \tag{A.1}$$

When the formula A.2 does not hold and the monitored vehicle does not change lane, $vm$ will be detected as a node that generates a false alert.

$$\left\{ \begin{array}{c} Speed_{vm} \leq vm_{speed-alert} \\ vm_{speed-beacon} < vm_{speed-alert} \end{array} \right\} \tag{A.2}$$

As illustrated in figure A.2 (b), when a malicious behavior is detected, the monitoring vehicle $(v1)$ disseminates a $monitoring\_message$ to inform its legitimate neighbors. This message includes: the source identity $(v1)$, the malicious vehicle's identity $(vm)$ and detection time. Furthermore, as shown in figure A.2 (b), $v1$ neighbors monitor its behavior to verify if its speed has decreased and lane has changed. In case, a false alert claimed by $v1$ is correct, neighbors store in their monitoring list the identity of the malicious vehicle and detection time. Otherwise, $v1$ is designated as a node that disseminates false information and it will be stored in a monitoring list. At the end, this list will be forwarded to the nearest RSU as illustrated in figure A.1. However, when the RSU is out of range, a store and forward mechanism is launched.

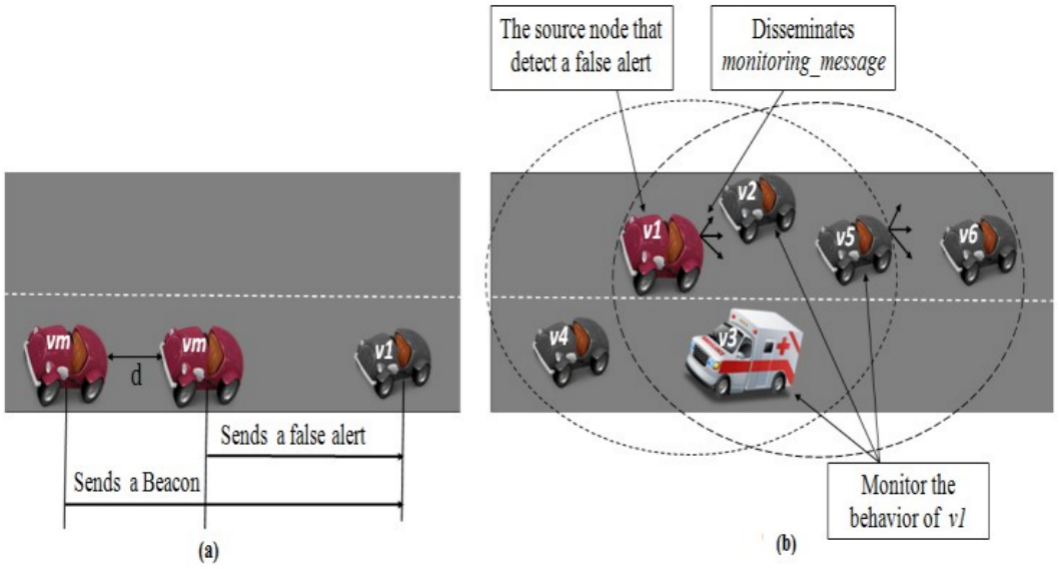



Figure A.2 – Attack detection scenario:(a)monitoring the target node behavior,(b)dissemination of malicious vehicle identity and surveillance of the monitoring vehicle behavior

The RSU computes the malicious probability ($MP \in [0,1]$) related to a vehicle $v_j$ during a detection period $\delta t$ as shown in equation A.3. Afterward, each RSU exchanges with others the monitoring list. Each list includes the malicious vehicles' identities, time they were detected, RSU identity where they were attached and their $MP$ value. We note that $\delta t$ will be determined in our simulation experiments. Later on, with the help of game theory, the RSU predicts the future malicious behavior of a monitored vehicle and takes the final decision about its categorization.

$$MP(\delta t) = \frac{\sum_{i=1}^{n} R_i * nb^{j}_{detection}}{m} \tag{A.3}$$

$nb^{j}_{detection}$ (equal to $n$) presents the rate of monitoring vehicles that detect a vehicle $v_j$ as an attacker. $R$ is the reliability detection of the monitoring vehicles, which depends on their detection history. In case, monitoring vehicles provide correct detections during their passage through RSU range, $\sum_{i=1}^{n} R_i$ is equal to 1, otherwise $\sum_{i=1}^{n} R_i$ tends to 0 when vehicles persist to send a false detection. The calculation of R is calculated using a voting mechanism as presented in [165]. m is the number of $v_j$'s neighbors during a detection period $\delta t$.

## A.3/ MALICIOUS BEHAVIOR PREDICTION BASED ON GAME THEORY

In this section, we describe our intrusion detection and decision performed between vehicles and RSU. First of all, we provide the payoff matrix of the game related to the monitored vehicle and RSU players and define a set of strategies that could occur between players. Afterwards, based on these strategies and with the help of Nash equilibrium concept, we predict the malicious behavior of an attacker in future states. Finally, according to this malicious behavior's prediction approach, we categorize the malicious vehicle according to its future attack severity into an appropriate list.

### A.3.1/ GAME FORMULATION

In our approach, we consider two players: RSU and the monitored vehicle, that is suspected by its neighbors as malicious. Each player chooses to perform a specific action in order to maximize its payoff. $J_v$ and $J_{RSU}$ denote respectively the RSU and the monitored vehicle players in the following. The RSU can perform one of these two pure strategies: detect, and categorize the monitored vehicle into a selected list (see subsection A.3.3) or wait. The vehicle can attack or wait. At each state, our game represents an interaction between the RSU and vehicles located within its radio range. Here, we divide time into regular intervals called time-slots; each one of them represents a State. Therefore, with the help of the Bayesian game concept, we model the different strategies that could occur between an attacker and the RSU, and the future states' prediction of a suspected behavior is determined thanks to Bayesian Nash Equilibrium (BNE).

In the following, we present the payoff matrix of the game between the monitored vehicle and RSU (see Table A.1), where some notations are defined as follows:

- $nb-false-detection_{i,j}$: the rate of attackers $v_j$ that $RSU_i$ does not detect, i.e false negative rate,
- $nb-false-positive_{i,j}$: the rate of normal vehicles $v_j$ that are considered by $RSU_i$ as malicious, i.e false positive rate,
- $nb-attacks-detection_{i,j}$: the rate of attackers $v_j$ that $RSU_i$ detects,
- $nb-forwarder_j$: the rate of legitimate vehicles that accept and forward the false alert messages from a malicious vehicle,
- Cost: the percentage of overhead that RSU needs to detect and categorize the monitored vehicle into the appropriate list,
- $P=(p,1-p)$, where $p$ and $1-p$ are the probabilities with which the player performs a detect & categorize or wait actions, respectively,
- $Q=(q,1-q)$, where $q$ and $1-q$ are the probabilities with which the attacker player performs an attack or wait (i.e. do not attack) actions, respectively.

| RSU \ Vehicle | Attack | Wait |
|---|---|---|
| Detect & Categorize | $(X_{11},Y_{11})$ | $(X_{12},Y_{12})$ |
| Wait | $(X_{21},Y_{21})$ | $(X_{22},Y_{22})$ |

Table A.1 – Payoff matrix of intrusion detection game

$$X_{11}=nb-attacks-detection_{i,j}-(Cost+nb-false-detection_{i,j}) \tag{A.4}$$

$$X_{12}=-(nb-false-positive_{i,j}+Cost) \tag{A.5}$$

$$X_{21}=-nb-false-detection_{i,j} \tag{A.6}$$

$$X_{22}=0 \tag{A.7}$$

$$Y_{11}=(nb-false-positive_{i,j}+nb-forwarder_j)-nb-attacks-detection_{i,j} \tag{A.8}$$

$$Y_{12} = nb - false - positive_{i,j} \tag{A.9}$$

$$Y_{21} = nb - false - detection_{i,j} + nb - forwarder_j \tag{A.10}$$

$$Y_{22} = 0 \tag{A.11}$$

### A.3.2/ MALICIOUS BEHAVIOR PREDICTION

The utility of Bayesian Nash Equilibrium (BNE) is to predict the future behavior of a target vehicle since it aims to analyze the interaction between the attacker and RSU, and determine the permanent state, i.e. each player has an interest in performing the same action. According to Nash, there is mixed strategy BNE {player1(action1,$p^*$), player2(action2,$q^*$)} in which both players do not change their actions. As a result, we use the BNE concept to predict the sate when the malicious vehicle persists to attack (i.e. does not switch to a normal behavior).

**Theorem [1].** *There is a mixed strategy BNE {$J_{RSU}$(detect and categorize, $p^*$), $J_v$(attack, $q^*$)} in which the malicious vehicle attacks when the probability $q > q^*$ and the RSU triggers its detection (categorization) action when $p < p^*$.*

*Proof.* The mixed strategy of the RSU is defined as follows: $P = (p, 1-p)$, and the expected payoff of the malicious vehicle for playing attack and wait actions are:

$U_{Jv}(attack) = Y_{11} * p + Y21 * (1-p) = (nb - false - detection_{i,j} + nb - forwarder_j) - nb - attacks - detection_{i,j} * p$

$U_{Jv}(wait) = Y_{12} * p + Y_{22} * (1-p) = nb - false - positive_{i,j} * p$

The malicious vehicle plays an attack action when $U_{Jv}(attack) > U_{Jv}(wait)$. Therefore, we get:

$p < p^*$, with $p^* = \frac{nb-false-detection_{i,j}+nb-forwarder_j}{nb-false-positive_{i,j}+nb-attacks-detection_{i,j}}$ and $p^* \in [0,1]$

The mixed strategy of a malicious vehicle is defined as follows: $Q = (q, 1-q)$ and the expected payoff of the RSU for playing detect (categorize) and wait actions are:

$U_{Jrsu}$(detect and categorize)=$X_{11} * q + X_{12} * (1-q) = (nb - attacks - detection_{i,j} - nb - false - detection_{i,j} + nb - false - positive_{i,j}) * q - nb - false - positive_{i,j} - Cost$

$U_{Jrsu}(wait) = X_{21} * q + X_{22} * (1-q) = -nb - false - detection_{i,j} * q$

The RSU plays the detection and categorize when $U_{Jrsu}$(detect and categorize)$> U_{Jrsu}(wait)$. Therefore, we get:

$q^* > q$, where $q^* = \frac{nb-false-positive_{i,j}+Cost}{nb-attacks-detection_{i,j}+nb-false-positive_{i,j}}$ and $q^* \in [0,1]$ □

As a result, we conclude that when the attack probability of a malicious vehicle is above $q^*$ and the detection probability of the RSU is lower than $p^*$, both players do not change their actions. Therefore, when this equilibrium is reached, such malicious vehicle is detected as an attacker that persists to attack on the future states (which represent a Permanent state) and hence stored in the Revocation_Black list (as explained in subsection A.3.3).

### A.3.3/ VEHICLES CATEGORIZATION

It is not interesting to remove the vehicle directly when it exhibits a malicious behavior since it could switch to a normal behavior and keep this pattern during its passage through a network [118]. Thereby, the RSU categorizes the monitored vehicles into four lists:

**- White list:** The vehicle that acts as normal node during its passage through RSU's range, its $MP$ is equal to 0.

**- White & Gray list:** The vehicle oscillates between a normal and malicious behavior. However, in the future states, the rate of switching to malicious behavior is less than the tendency to switch to a normal behavior, $q1^* = MP < q2^*$, such vehicle is defined as Abnormal node.

**- Gray list:** The vehicle oscillates between a normal and malicious behavior. However, in the future states the rate of switching to malicious behavior is more probable than to switching to a normal behavior, $q2^* = MP$, such vehicle is defined as suspected node.

**- Revocation_Black list:** The vehicle oscillates between a normal and malicious behavior. However, in the future states it persists to act maliciously, i.e. does not switch to a normal behavior. In other words, its $MP$ converges to 1. In this case, $MP > q^*$ (Nash equilibrium).

Here, $q^* > q2^* > q1^*$ where the probability $q^*$ is determined according to the Nash equilibrium concept described in the previous subsection. However, the optimal values of the probabilities $q1^*$ and $q2^*$ are determined in our experiments which should satisfy our application requirement, i.e. high accuracy prediction. We note that, the vehicles that are stored in a Revocation_Black list will be totally prevented from communicating with legitimate vehicles. As cited above, RSUs exchange this list with others and each one informs the vehicles located within its radio range about the identities of these malicious nodes.

## A.4/ PERFORMANCE EVALUATION

In this section, we evaluate the performances of the proposed intrusion prevention mechanism. Therefore, we begin by a description to the simulation environment and used parameters, then we proceed for a determination of the optimal thresholds related to the Bayesian Nash Equilibrium previously described, and finally we discuss given results.

### A.4.1/ ENVIRONMENT DESCRIPTION

The proposed Intrusion prevention technique was implemented using NS3.17 simulator. In our simulation, we used two topologies: two parallel highways (2x3 lanes) and an urban scenario generated by the Simulation of Urban Mobility (SUMO) simulator. Furthermore, we simulated a square area of $3000x3000m^2$. Our measurements are based on averaging the results obtained from 15 simulation runs. Moreover, in our simulation, we vary the number of malicious vehicles from 10 to 40% of overall nodes. We insert, three categories of attackers: abnormal node, suspect node and malicious node that persist to attack, which are stored in a White & Gray, Gray and Revocation_Black lists, respectively. The main simulation parameters are summarized in Table A.2.

| Parameter | Value |
|---|---|
| Simulation area | $3000x3000m^2$ |
| Simulation time | 180sec |
| Radio range | 300m |
| Number of vehiles | 50 - 250 |
| Vehicle speed | 90 to 160 Km/h |
| Detection period ($\delta t$) | 7 sec |

Table A.2 – Simulation parameters for the game theory based IPDS

We compare our intrusion detection and prevention schema with two detection frameworks: Intrusion Detection Framework for Vehicular Network (IDFV) [165] and Data-Centric Misbehavior Detection (DCMD)[118]. Here, we evaluate the prediction performance; in other words the accuracy prediction (i.e. detection and false positive rates) of the future behavior of a monitored vehicle and its categorization in an appropriate list. Finally, we evaluate the overhead generated by the proposed schema. These metrics are defined hereafter.

**- Detection Rate (DR):** ratio of correctly identified malicious vehicles and their categorization in appropriate lists,

**- False Positive Rate (FPR):** ratio between the number of normal vehicles that are incorrectly classified as malicious,

**- Communication Overhead (COV):** the amount of information generated by the vehicle.

### A.4.2/ Optimal probabilities thresholds

The identification of the optimal probabilities $q1^*$ (that categorizes the monitored vehicles into White & Gray list) and $q2^*$ (related to the monitored vehicles that are stored in a Gray list) is crucial. In one hand, when these values are low, the number of false positives is important. In the other hand, when these thresholds are high, a great number of malicious vehicles are not detected. Therefore, a tradeoff between detection and false positive rates should be considered. The probability thresholds are computed as described in equations A.12 and A.13.

$$q1^* = q^* - \delta \tag{A.12}$$

$$q2^* = q^* - \gamma \tag{A.13}$$

We varied the values $\delta$ and $\gamma$, afterward we compute the detection and false positive rates for low and high number of malicious vehicles (10%, 20%, 30% and 40% of overall nodes). Therefore, the optimal probability thresholds that make a balance between detection and false positive rates are selected. We note that $q^*$ is determined according to the BNE concept described in the previous section.

| Number of attackers (%) | $\delta(q1^*)$ | $\gamma(q2^*)$ | DR(%) | FPR(%) |
|---|---|---|---|---|
| 10 | 0.42 | 0.28 | 100 | 0 |
| 20 | 045 | 0.29 | 98 | 0.5 |
| 30 | 0.49 | 0.34 | 97.33 | 1.15 |
| 40 | 0.56 | 0.37 | 97 | 1.35 |

Table A.3 – Optimal thresholds

According to our simulation results, we found that there are optimal probabilities thresholds ($q^*, q1^*$ and $q2^*$) that allow us to detect and predict with a high accuracy the future malicious behavior of a monitored vehicle and hence categorize it in an appropriate list. Furthermore, the optimal values of $\delta$ and $\gamma$ that make a good tradeoff between the detection and false positive rates are illustrated in Table A.3.

### A.4.3/ RESULTS

As previously stated, we first compare the performance of the intrusion prevention mechanism to the IDFV and DCMD in terms of detection and false positive rates. Both frameworks attempt to detect the current attacks that occur in the network, where the first one targets to detect DoS, whereas the second aims false alert attack. Afterward, we analyze the overhead generated by the prevention mechanism. The number of vehicles varies from 50 to 250. We consider to treat the worst case where the number of malicious nodes in the network is equal to 40%, thus we deploy thresholds given in A.3 relative to this number.

#### A.4.3.1/ ACCURACY PREDICTION

According to figure A.3(a) and (b), we show that the game theory-based intrusion mechanism exhibits a high accuracy prediction compared to IDFV and DCMD since the detection rate is over 98% and the false positive rate is lower than 2% when the number of vehicles increases, which demonstrate that it is scalable. These results are achieved thanks to the malicious behavior prediction concept that has the ability to predict with a high accuracy future misbehaves and categorize responsible nodes in the appropriate list. This is unlike IDFV and DCMD frameworks, where there is no prevention technique since they aim to detect only current attackers that occur in the network.

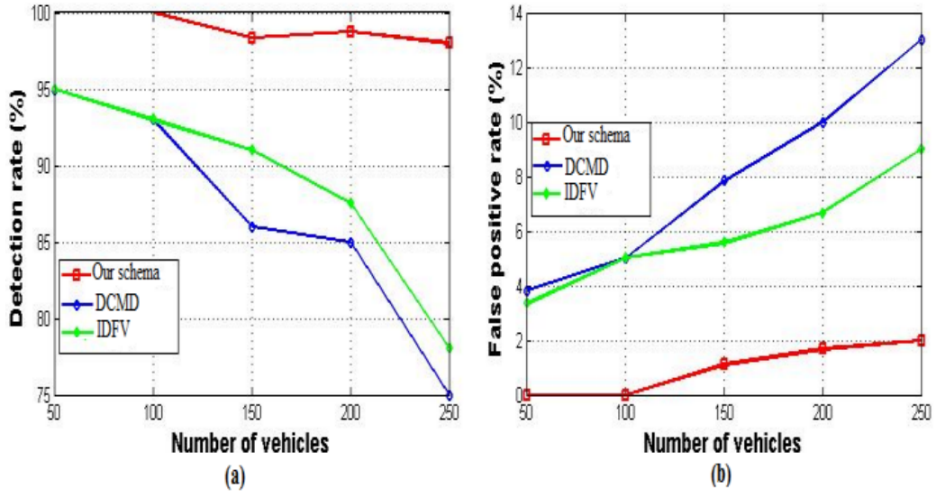



Figure A.3 – Accuracy prediction:(a)Detection rate and (b)False positive rate

#### A.4.3.2/ OVERHEAD

In this research, we propose a hybrid intrusion detection and decision schema, where the detection is performed in a distributed manner between vehicles and the decision is performed by a centralized RSU. We compare our hybrid schema with a distributed one in terms of overhead. As

illustrated in figure A.4 , both schemas generate the same amount of overhead when the number of vehicles is equal to 50. However, when this number increases, the distributed schema generates a high overhead compared to the hybrid one. This result is achieved since in a distributed schema, each vehicle exchanges with its neighbors the malicious lists in order to propagate them to all nodes within a network. Therefore, such schema leads to an increase on overhead, specifically when the number of malicious vehicles is important. Furthermore, in a hybrid schema only the RSU disseminates the complete lists to the vehicles (located within it radio range) since it has the lists of all malicious vehicles that are detected within a network.

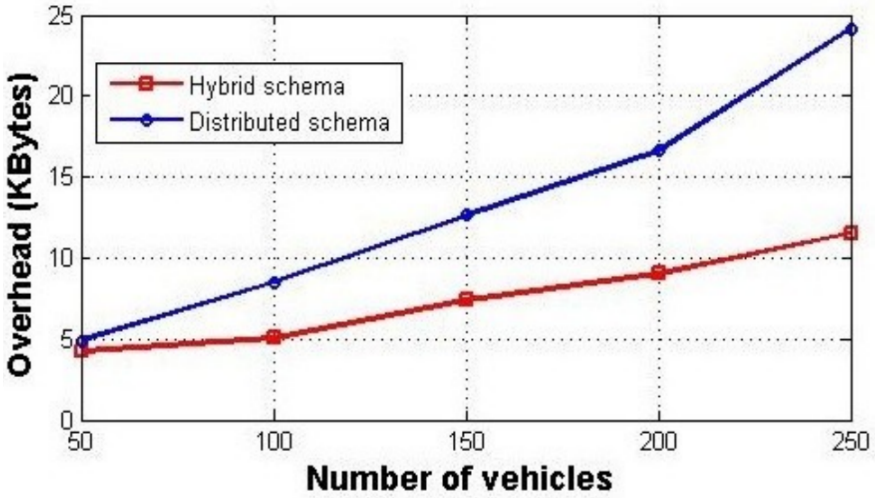



Figure A.4 – Communication overhead

## A.5/ CONCLUSION

In this appendix, we proposed and designed a lightweight and efficient intrusion detection and prevention mechanism for vehicular networks. To point out, with the help of game theory, our technique has the ability to predict with a high accuracy a future malicious behavior of an attacker and categorize it into an appropriate list according to the future attack severity. According to our simulation results, we demonstrated that our intrusion detection and prevention schema exhibits a high detection rate with more than 98%, low false positive rate (close to 2%) and generates a low overhead when the number of malicious vehicles is very high. Furthermore, the proposed prediction technique is flexible and could be embedded either on a flat or a hierarchical topology. Therefore, in a near future we aim to embed the schema in a cluster-based topology and evaluate these both topologies in term of generated overhead.

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

## Abstract:

Nowadays, automotive area is witnessing a tremendous evolution due to the increasing growth in communication technologies, environmental sensing & perception aptitudes, and storage & processing capacities that we can find in recent vehicles. Indeed, a car is being a kind of intelligent mobile agent able to perceive its environment, sense and process data using on-board systems and interact with other vehicles or existing infrastructure. These advancements stimulate the development of several kinds of applications to enhance driving safety and efficiency and make traveling more comfortable. However, developing such advanced applications relies heavily on the quality of the data and therefore can be realized only with the help of a secure data collection and efficient data treatment and analysis. Data collection in a vehicular network has been always a real challenge due to the specific characteristics of these highly dynamic networks (frequent changing topology, vehicles speed and frequent fragmentation), which lead to opportunistic and non long lasting communications. Security, remains another weak aspect in these wireless networks since they are by nature vulnerable to various kinds of attacks aiming to falsify collected data and affect their integrity. Furthermore, collected data are not understandable by themselves and could not be interpreted and understood if directly shown to a driver or sent to other nodes in the network. They should be treated and analyzed to extract meaningful features and information to develop reliable applications. In addition, developed applications always have different requirements regarding quality of service (QoS). Several research investigations and projects have been conducted to overcome the aforementioned challenges. However, they still did not meet perfection and suffer from some weaknesses. For this reason, we focus our efforts during this thesis to develop a platform for a secure and efficient data collection and exploitation to provide vehicular network users with efficient applications to ease their travel with protected and available connectivity. Therefore, we first propose a solution to deploy an optimized number of data harvesters to collect data from an urban area. Then, we propose a new secure intersection based routing protocol to relay data to a destination in a secure manner based on a monitoring architecture able to detect and evict malicious vehicles. This protocol is after that enhanced with a new intrusion detection and prevention mechanism to decrease the vulnerability window and detect attackers before they persist their attacks using Kalman filter. In a second part of this thesis, we concentrate on the exploitation of collected data by developing an application able to calculate the most economic itinerary in a refined manner for drivers and fleet management companies. This solution is based on several information that may affect fuel consumption, which are provided by vehicles and other sources in Internet accessible via specific APIs, and targets to economize money and time. Finally, a spatio-temporal mechanism allowing to choose the best available communication medium is developed. This latter is based on fuzzy logic to assess a smooth and seamless handover, and considers collected information from the network, users and applications to preserve high quality of service.

**Keywords:** VANET, Data collection, Security, Network Selection, Data analysis, Routing, Quality of service

## Résumé :

De nos jours, la filiale automobile connait une évolution énorme en raison de la croissance évolutive des technologies de communication, des aptitudes de détection et de perception de l'environnement, et des capacités de stockage et de traitement présentes dans les véhicules. En effet, une voiture est devenue une sorte d'agent mobile capable de percevoir son environnement et d'en collecter des informations, de communiquer avec les autres véhicules ou infrastructures présentes sur la route, et de traiter les données collectées. Ces progrès stimulent le développement de plusieurs types d'applications qui vont permettre d'améliorer la sécurité et l'efficacité de conduite et de rendre le voyage des automobilistes plus confortable. Cependant, ce développement repose beaucoup sur les données collectées et donc ne pourra se faire que via une collecte sécurisée et un traitement efficace de ces données détectées. La collecte de données dans un réseau véhiculaire a toujours été un véritable défi en raison des caractéristiques spécifiques de ces réseaux fortement dynamiques (changement fréquent de topologie, vitesse élevée des véhicules et fragmentation fréquente du réseau), qui conduisent à des communications opportunistes et non durables. L'aspect sécurité, reste un autre maillon faible de ces réseaux sans fils vu qu'ils sont par nature vulnérables à diverses types d'attaques visant à falsifier les données recueillies et affecter leur intégrité. En outre, les données recueillies ne sont pas compréhensibles par eux-mêmes et ne peuvent pas être interprétées et comprises si montrées directement à un conducteur ou envoyées à d'autres nœuds dans le réseau. Elles doivent être traitées et analysées pour extraire les caractéristiques significatives et informations pour développer des applications utiles et fiables. En plus, les applications développées ont toujours des exigences différentes en matière de qualité de service (QdS). Plusieurs travaux de recherche et projets ont été menées pour surmonter les défis susmentionnés. Néanmoins, ils n'ont pas abouti à la perfection et souffrent encore de certaines faiblesses. Pour cette raison, nous focalisons nos efforts durant cette thèse au développement d'une plateforme de collecte efficace et sécurisée de données dans un réseau de véhicules ainsi que l'exploitation de ces données par des applications améliorant le voyage des automobilistes et la connectivité des véhicules. Pour ce faire, nous proposons une première solution visant à déployer de manière optimale des véhicules, qui auront la tâche de recueillir des données, dans une zone urbaine. Ensuite, nous proposons un nouveau protocole de routage sécurisé permettant de relayer les données collectées vers une destination en se basant sur un système de détection et d'expulsion des véhicules malveillants. Ce protocole est par la suite amélioré avec un nouveau mécanisme de prévention d'intrusion permettant de détecter des attaquants au préalable en utilisant les filtres de Kalman. En deuxième partie de thèse, nous nous sommes concentré sur l'exploitation de ces données en développant une première application capable de calculer de manière fine l'itinéraire le plus économique pour les automobilistes ou tout gestionnaire de flottes de véhicules. Cette solution est basée sur les données influents sur la consommation de carburant et collectées à partir des véhicules eux mêmes et aussi d'autres sources d'informations dans l'Internet et accessibles via des API spécifiques. Enfin, un mécanisme spatio-temporel permettant de choisir le meilleur médium de communication disponible a été développé. Ce dernier est basé sur la logique floue et considère les informations recueillies sur les réseaux, les utilisateurs et les applications pour préserver de meilleure qualité de service.

**Mots-clés :** VANET, Collecte de données, Sécurité, Selection du réseau, Analyze de données, Routage, Qualité de Service

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr