UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

VANET-Based optimization of infotainment and traffic efficiency vehicular services

Autor: Marco Gramaglia
Directores: Dr. María Calderón Pastor
Dr. Carlos Jesús Bernardos Cano

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés, Septiembre de 2012

TESIS DOCTORAL

VANET-Based optimization of infotainment and traffic efficiency vehicular services

Autor: Marco Gramaglia
Directora: Dr. María Calderón Pastor
Director: Dr. Carlos Jesús Bernardos Cano

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

"Bueno, es lamentable"
Xavier Hernández Creus
"Habláis de Pedro León como si fuese Zidane o Maradona"
José Mário dos Santos Mourinho Félix
"Nosotros pintamos poco en todo esto"
Josep Guardiola i Sala
"L'Italia é il paese che amo"
Silvio Berlusconi
"El afán de prohibir es algo propio de los estados totalitarios"
Esperanza Aguirre Gil de Biedma
"Why always me"
Mario Balotelli

Acknowledgments

Me gustaría agradecer primeramente a mis directores de tesis, María y Carlos Jesús, por todos los conocimientos que me habéis transmitido durante estos cuatro años. Then, I would like to thank the institution and the people that made possible this thesis by supporting it: the Institute IMDEA Networks and its directors during these years: Arturo Azcorra and Albert Banchs. Quindi, voglio ringraziare la mia famiglia: Lara, Dina e Giovanni che mi hanno aiutato durante tutti questi anni. Adems quiero agradecer, en orden abierto: Andrés, Pablo, Antonio, Manolo, Isaías, Goyo, Ignacio, Isa, José Pablo, Víctor, David, Isaac, Iván, José Alberto, Alberto, Richi, Chema, Elvira, Rubén, Ángel, Paul, Andra, Fabio, Mika, Alex y otras personas de las que seguramente me olvido y que me apoyaron de manera directa o menos (aunque sólo sea por aguantarme) durante todo este tiempo. *Dulcis in fundo*, tutta la gente di Macellai e dintorni, che non mi ha fatto mai mancare la sua amicizia: Teo, Mona, Davide Inge, Gian Marco, Massimo, Dario, Nicola e Davide Barb.

Resumen

El diseño, normalización y futuro despliegue de los sistemas de comunicación vehiculares han sido principalmente impulsados hasta el momento por las aplicaciones de seguridad vial. Hay dos aspectos adicionales de las redes vehiculares que han visto crecer su relevancia en los últimos años: los servicios de Infotainment y los de eficiencia del tráfico. Estos servicios pueden mejorar la experiencia de los conductores y hacer que los sistemas de comunicación vehiculares resulten más atractivos para los usuarios finales. En esta tesis, se proponen mecanismos de optimización para ambos tipos de servicios vehiculares. Los servicios de Infotainment están relacionados con la provisión de las clásicas aplicaciones IP tales como, navegar, acceder al correo electrónico, o a las redes sociales. Los servicios de eficiencia de tráfico permiten añadir nuevas funcionalidades a los sistemas de navegación con los objetivos de: optimizar el uso de las infraestructuras viarias, reducir los tiempos de viaje y consecuentemente, minimizar el impacto ambiental. Acceder a los servicios de Infotainment desde redes vehiculares conlleva cumplir con los protocolos y mecanismos estandarizados que permiten la interconexión de redes heterogéneas a Internet. Hay tres funcionalidades principales que tienen que ser proporcionadas: configuración automática de direcciones, encaminamiento eficaz y gestión de la movilidad. Esta tesis propone mecanismos para hacer frente a los tres aspectos mencionados: una técnica basada en overhearing que mejora un protocolo de configuración automática de direcciones ya estandarizado, un algoritmo de encaminamiento basado en árboles especialmente diseñado para las comunicaciones desde el vehículo a Internet y, un algoritmo de gestión de la movilidad optimizado para entornos vehiculares. En cuanto a los servicios de eficiencia de tráfico, esta tesis propone dos algoritmos que utilizando las técnicas de comunicación vehículo a vehículo permiten monitorizar y pronosticar a corto plazo las condiciones en el tráfico, como es el caso de posibles atascos.

Todos los algoritmos y modelos analíticos descritos en esta tesis han sido validados a través de simulaciones y/o implementaciones usando hardware estándar.

Palabras clave: redes vehiculares, comunicaciones inalámbricas, configuración automática de direcciones, encaminamiento, eficiencia del tráfico.

Abstract

The design, standardization and future deployment of vehicular communications systems have been driven so far by safety applications. There are two more aspects of the vehicular networking that have increased their importance in the last years: *infotainment* and traffic efficiency, as they can improve drivers' experience, making vehicular communications systems more attractive to end-users. In this thesis we propose optimization mechanisms for both types of vehicular services.

Infotainment services are related to the provision of classic IP applications, like browsing, reading e-mail or using social networks. Traffic efficiency services are those accessing new capabilities to the car-navigation systems, aiming at optimizing the usage of road infrastructures, reducing travel times and therefore minimizing the ecological footprint.

Bringing infotainment services to the vehicular environment requires to comply with standard protocols and mechanisms that allow heterogeneous networks to be interconnected in the Internet. There are three main functionalities that have to be provided: *i)* address autoconfiguration, *ii)* efficient routing and *iii)* mobility management.

Regarding infotainment services, this thesis proposes mechanisms tackling the abovenamed aspects: an overhearing technique to improve an already standardized address autoconfiguration protocol; a tree-based routing algorithm especially tailored for vehicle-to-Internet communications and an optimized mobility management approach for vehicular environments.

Regarding traffic efficiency, this thesis proposes two algorithms that make use of vehicular communication techniques to monitor and forecast short-term traffic conditions. We first improved our knowledge on drivers' behavior by analyzing real vehicular data traces, and proposes a mixture model for the vehicles interarrival time. This outcome was used for validating the proposed infotainment optimization as well.

All the algorithms and analytical models described in this thesis have been validated by simulations and/or implementations using standard hardware.

Keywords: Vehicular networks, wireless communications address autoconfiguration, routing, traffic efficiency.

Contents

Chapter 1

Introduction

In the last few years, vehicular ad-hoc networks (VANETs) have been put into the spotlight. Car manufacturers and Public Transport Authorities have been investing on these emerging technologies, mainly driven by the goal of increasing traffic safety. New standards are being or have been proposed in the recent years, like the ones developed by institutions such as the Institute of Electrical and Electronic Engineers (IEEE) or the European Telecommunications Standards Institute Technical Committee for Intelligent Transportation Systems (ETSI TC ITS), or by consortia like the Car-to-Car Communication Consortium (C2C-CC).

Besides safety-related applications, which have unique requirements of delay and priority, and are generally broadcast-based, two families of services can be identified: infotainment and traffic efficiency.

The Infotainment services are those concerning the provision of classic IP services available in the Internet: browsing, reading e-mails, using social networks. Nowadays, users want to enjoy Internet applications continuously, not only when at home or at work. With the increasing success of the ubiquitous computing paradigm and the embedding of smart devices into vehicles, the research in the field of infotainment services has become more

important in the recent past. Enabling infotainment services in VANETs, not only implies providing efficient communications within the vehicular environment, dealing with the unique mobility characteristics of vehicles; but also complying with standard mechanisms and techniques currently used in the Internet.
The second type of services, traffic efficiency, aims at exploiting the ability of forming short-range wireless multi-hop networks to achieve a better usage of road infrastructure hence decreasing the time spent traveling by car and reducing the pollution. By smartly routing vehicles accordingly to the traffic conditions, traffic jams can be avoided, reducing travel times. Vehicles can be used to cooperatively exchange information in real time and warn other vehicles about current or near future congestion.
Among all the open issues that are present in the vehicular networks environment, this thesis focuses on both the aforementioned problems: providing Internet access from vehicles (i.e., infotainment service) and using the VANET paradigm to monitor and forecast the vehicular traffic conditions.
If we look at infotainment services, bringing IP connectivity to VANETs is the main challenge. In order to do so, the following three main aspects have to be addressed:

- *a)* vehicles have to be able to autoconfigure an IP address,
- *b)* an efficient routing of IP datagrams from/to the vehicular network is required,
- *c)* IP mobility mechanisms have to be tailored for vehicular scenarios.

In this thesis we propose a solution to efficiently solve each one of the previously introduced functionalities: *a)* an analytical model and an enhancement of an existing efficient IPv6 address autoconfiguration mechanism (GeoSAC), *b)* a hybrid position based routing for vehicle-to-Internet communications (TREBOL); and *c)* a smart IP mobility algorithm for an efficient use of the available access technologies (SILVIO).
Multi-hop communications in a vehicular environment are heavily influenced by vehicles' mobility. The presence of connectivity gaps between consecutive vehicles can lead to the unfeasibility of communication through a multi-hop path.The work developed in this thesis could count on data coming from vehicular traffic measurement points in the city of Madrid. To analyze these data traces, especially in terms of interarrival times between consecutive vehicles, it not only useful to evaluate the performance of the analytical models proposed throughout the thesis, but also to provide insights on drivers' behavior. This thesis proposes a novel approach to model the distribution of interarrival times that takes into account the interactions between close vehicles.
Studying the interactions between vehicles is not only useful to validate network algorithms, but also to infer the current and short-term future states, in the framework of a traffic efficiency service. However, observing vehicular traffic is an expensive task to

perform. The installation and the possible relocation of the monitoring hardware is expensive and gives a reduced flexibility to the systems. Applying VANET-based techniques to gather the current traffic conditions will reduce the cost and increase the flexibility of the service, contributing to a more capillary dataset of real-time measurements. As VANETs can form a self configured network, vehicles can exchange information among them and gather metrics that cannot be captured using the currently used solutions. In this thesis we propose two solutions for traffic monitoring and short term forecast that exploit vehicular communication.
In order to improve the readability of the thesis, we divided it into two parts. The first one is about the optimization of infotainment services, and includes the proposals that directly deal with the problem of providing Internet connectivity from vehicular environments. The second one tackles the problem of traffic efficiency, and comprises our findings in the field of the vehicular flow analysis and their use in traffic monitoring and forecasting systems.

1.1 Background

There is already a quite extensive state of the art in the field of non-safety based services for vehicular networks, both for infotainment and traffic efficiency. Providing support for IP based services in VANETs is a complex problem that raises a large number of heterogeneous issues. As already stated previously, there are three main functionalities that have to be carried out: address autoconfiguration, routing and mobility management. Despite the common goal of these three research fields, they can be tackled somehow independently, as each one is considering a different piece of the problem. Many of the already proposed solutions are inspired by mobile ad-hoc network (MANET) solutions, with modifications in order to deal with VANETs characteristics (i.e., nodes high mobility).
The multi-hop nature of VANETs and the absence of a single multicast-capable link for signaling, typically makes the use of standard IPv6 autoconfiguration protocols, both stateless (Stateless Address Autoconfiguration, SLAAC) and stateful (Dynamic Host Configuration Protocol, DHCPv6), impossible to be used without any modifications. Several proposals in the literature [1,2] introduce modified versions of DHCP, adapted to work in VANETs environments. Using a stateless mechanism like IPv6 SLAAC offers advantages in terms of scalability as it does not rely on a central entity (or a federation of delegates). This architecture is studied in [3,4]. The solution proposed in [3] is the basis of the work further described in Chapter 2.
In general, using MANET routing protocols in vehicular scenarios is not suitable, due to VANETs' high dynamics and the possibly very short link lifetimes. The common assumption that vehicles are equipped with a positioning device like GPS has led to the

adoption of geographic based solution as a common choice for routing in vehicular networks. Numerous solutions like [5], [6] base their operation on additional VANET-related information such as position of gateways, traffic statistics or trajectory estimation. This extra information is however difficult to gather and expensive (in terms of overhead) to keep up-to-date. The work described in Chapter 3 consists in a routing algorithm for vehicle-to-Internet communications that exploits the geographic location knowledge of the nodes.

Deploying a wireless-based access network for vehicular environments is a difficult task. The deployment of access points to the backhaul network cannot be as dense as needed for a complete direct coverage, due to the high installation costs. Giannoulis *et al.* [7] already proposed mechanisms to efficiently handoff between access points, but without a full wireless coverage seamless connectivity is impossible to achieve. The adoption of a multi-hop communication protocol allows a larger number of nodes to enjoy Internet connectivity via the wireless-based access network but, on the other hand, poses problems of path-stability.

Using the cellular network, vehicles can achieve an "always on" connectivity, but it contributes to add more traffic to an already congested network[1]. In [8] the authors show that, by conveniently switching between access technologies, vehicles can offload a not negligible part of their traffic to the wireless-based access networks. The work outlined in Chapter 4 designs a mechanism for efficiently exploiting both the access technologies, using the cellular network as the default access network, but offloading some non-critical flows when a wireless-based network becomes available.

The second part of this thesis deals with traffic efficiency services. A first useful step is to better understand drivers' behavior as the performance of multi-hop wireless network protocols heavily depends on the availability of the used paths, that is directly influenced by the vehicles' mobility patterns and interactions. The Spanish General Directorate of Traffic and the Madrid City Council, kindly provided us with real vehicular traffic measurements, coming from fixed observation points in the city of Madrid. We used them to develop a model of the interarrival times between vehicles, and have an useful overview about how they can successfully communicate among them using wireless devices (either radio based, infrared [9], or using visible light communications [10, 11]). While previous studies showed that interarrival times between consecutive vehicles can be modeled using an exponential [12] or a lognormal distribution [13], we found a relation between the used probability distribution and the drivers behavior. Chapter 5 details the procedure we used to analyze the vehicular trace dataset and the steps taken to build an exponential-gaussian based mixture model.

The analysis of the interactions among vehicles is not just useful for studying the effectiveness of networking algorithms. Public transportation authorities are nowadays constantly

[1] `http://bit.ly/3GCongestion`

monitoring the traffic state in their principal arteries, and there are commercial solutions that add real time traffic information to the navigation system capabilities. Currently, the devices used for monitoring purposes are expensive and difficult to deploy. Results very similar to the ones obtained by a legacy solution can be achieved by applying VANET-based monitoring solutions, without incurring into high installation costs and providing more flexibility to the monitoring system. Moreover, as vehicles are cooperating and acting as distributed sensors, more than one traffic variable (*flow*, *density* and *speed*) can be gathered at the same time. A previous work [14] shows that the vehicular traffic demand can be short-term forecast using the value of these traffic variables from the recent past. The work described in Chapter 6 exploits Inter-vehicular communication (both among vehicles and between vehicles and the fixed infrastructure) to provide a solution for cheaper and efficient vehicular traffic monitoring and forecasting.

1.2 Overview and Summary

The contributions of this thesis are summarized as follows. This thesis first proposes solutions to optimize the provision of infotainment services in VANETs: a model and an enhancement of the IP address autoconfiguration mechanism standardized by ETSI; a tree based routing protocol for VANETs and a mobility solution for heterogeneous internet access from vehicular networks. Furthermore, an interarrival time between vehicles analysis based on real traces is used in order to validate the effectiveness of the proposed infotainment solutions, but also as basis for the design of traffic efficiency solutions, namely two proposals for the efficient vehicular traffic monitoring and forecasting of its future values.

The first contribution of this thesis is to model and enhance GeoSAC (Geographically Scoped stateless Address Configuration) [3] which is an adaptation to VANETs of the standard IPv6 SLAAC (Stateless Address Autoconfiguration) mechanism, built on top of the ETSI TC ITS system architecture, that was subsequently standardized as ETSI SLAAC. ETSI SLAAC supports geographic addressing and networking, by extending the concept of an IPv6 link to a specific geographic area associated with a point-of-attachment (called Geographical Virtual Link, GVL). The network is intended as composed by vehicles (that carry On Board Units - OBUs) and Road Side Units (RSU) that form a self-organizing wireless network. RSUs periodically send out Router Advertisement (RA) messages which reach the nodes currently located within a well-defined area, and the nodes can then generate IPv6 addresses appending their network identifier derived from the MAC address to the received IPv6 prefix.
The goal of the developed work was to analyze and to design an improvement to ETSI SLAAC based on RA caching and an analytical model to evaluate the address

configuration time. When using ETSI SLAAC, the vehicular network is logically partitioned in non-overlapping areas by selectively filtering received RAs. However, a vehicle within radio range of others in a neighboring area also receives the messages they sent. Therefore, the performance of ETSI SLAAC (in terms of reduction of its configuration time and/or signaling overhead) can be improved by vehicles caching the Router Advertisements sent by other vehicles in adjacent areas and reusing them when crossing the area border, without waiting for the reception of the next Router Advertisement (vehicles are considered to be equipped with positioning devices). Simulations and implementations using Commercially available Off-The-Shelf (COTS) devices show that using the proposed optimization the configuration time can be pulled down to zero under most traffic conditions and different RA configurations, obtaining a potentially seamless IP address configuration. The whole study is detailed in Chapter 2 and was published in [15,16].

The second contribution of this thesis is TREBOL [17] (Tree-Based Routing for Vehicle-to-Internet Communications), a tree-based and configurable routing protocol which benefits from the inherent tree-shaped nature of vehicle-to-Internet traffic (V2I) to reduce the signaling overhead while dealing efficiently with the vehicular dynamics.
TREBOL builds the upstream routing tree by slightly modifying the Router Advertisement messages sent by the Road Side Units with some additional information required to construct the tree piggybacked into them. Once the tree has been built, every node has a parent node that is used as next hop for upstream (vehicle-to-Internet, V2I) traffic. This upstream tree is updated periodically by means of some lightweight signaling sent by the Road Side Unit. For downstream traffic, nodes keep soft-state about routes for currently used destinations. Furthermore, the protocol could also be used to allow nodes to autoconfigure IPv6 addresses when used in conjunction with our GeoSAC overhearing optimization, reducing even more the overall control overhead required by routing and address autoconfiguration functions. Another remarkable feature of the proposed protocol is the wide range of deployment scenarios, mostly defined by the size of the TREBOL area where it may operate, making it suitable for both urban and highways scenarios.
The protocol is validated using a real trace driven simulator and the results show that TREBOL outperforms a well known geographic routing protocol, providing better traffic delivery ratio and allowing at the same time a significant saving of control overhead, aspect considered as critical in wireless VANETs networks, as it does not introduce any beaconing mechanism. Moreover, in [18] TREBOL has been experimentally evaluated using a Linux implementation under lab-controlled realistic scenarios. The proposal is detailed in Chapter 3.

The third contribution of the thesis is the design of an IP mobility solution tai-

lored for VANETs, where each well defined area is represented by an IPv6 prefix. Traveling vehicles traverse many areas during a trip, thus changing IPv6 prefix and point of attachment to the infrastructure network. SILVIO [19] (Seamless Internet 3G and Opportunistic WLAN Vehicular Internet Connectivity) is a mobility mechanism that allows to seamlessly hand off between different RSUs via a multi-hop wireless access through the VANET. However, the knowledge coming from previous works showed us that this could be sometimes unfeasible, due to a lack of multi-hop connectivity. Although using a WLAN interface is desirable because it can provide higher bandwidth, the possibly lack of multi-hop connectivity suggests its opportunistic use. SILVIO uses standardized techniques (e.g., flow mobility) and vehicular modifications of them (e.g., 802.21 Media Independent Handover), allowing to efficiently use the connectivity through the VANET when available. Hence, our IP mobility solution for VANETs considers the possibility of using a cellular network interface as default option, conveniently offloading some selected IP flows as soon as a multi-hop link to a RSU becomes available. We describe this proposal in Chapter 4.

Another contribution arose by the need for a procedure to validate the proposed analytical models. Initially, the distribution used for modeling the interarrival time between two consecutive vehicles was an exponential distribution, as it was already used in many papers in the literature. This model was firstly proposed in [20] and more recently extended by [21]. Later, we used real vehicular traffic measurements collected at different locations in several highways of the city of Madrid to feed the simulators used in [15] and [17]. The final step was to perform an analysis of the traces, checking if they matched the exponential based model.
While in some traces the model fits adequately the traces, in others the experimental data largely deviates from the negative exponential random variable. We wanted to go beyond this first approximation, aiming at a more refined distribution for the interarrival time between vehicles. As the set of data comprised speeds as well, we found a noticeable correlation between them (especially at highways with dense traffic) that can be in some cases particularly high for consecutive cars. With this hint we divided vehicles into two groups, the "burst" vehicles (the ones traveling at similar speeds, close to each other) and the "isolated" vehicles (with no relation between the speed of two consecutive ones). We modeled the first group as a Gaussian distribution and the latter as an exponential distribution. Using statistical tools like the expectation-maximization (EM) algorithm, we could build a weighted mixture model of the two components. After comparing the maximum likelihood of our model with other distributions proposed in the literature (lognormal and exponential) we found that it is the best suited to mimic the behavior observed. The full procedure is detailed and described in Chapter 5.

The last contributions of this thesis relate to traffic efficiency services which have gained strength in the last few years. A study from the University of Texas[2] claims that the total cost of traffic congestion in the U.S. in that year was 100 billion USD.
Current technologies used for traffic monitoring purposes suffer of high installation costs both for the work needed to embed the hardware into the road infrastructure (that can reduce the road capacity and create troubles to the traffic) and for the hardware cost itself. Moreover, current technologies are extremely difficult to relocate in case of a change of requirements.
The use of a vehicular communication based solution can solve these two particular issues. Having vehicles that cooperate by exchanging data among them (and with installed Road Side Units) to make available real time traffic information about a given stretch of road will enable a cheaper and more flexible monitoring paradigm. This outcome can be achieved either using a completely distributed vehicle-to-vehicle (V2V) communication based solution or by taking advantage of already deployed RSUs.
The theory of vehicular traffic is a well known topic that has been extensively researched in the past century. Among all the theories that came up, the Kerner's *Three-phase traffic theory* [22] is one of the most considered. It introduces three possible states of the vehicular traffic: *Free flow*, *Synchronized* and *Traffic jam*. The transitions between the different states happen according to the variation of three fundamental variables the vehicular *flow*, *speed* and *density*. We designed a cooperative distributed algorithm which is able to monitor the three fundamental traffic variables, something that cannot be easily achieved using legacy standard monitoring techniques. This exercise benefited from the knowledge gained while performing the analysis of the real vehicular traces as described before.
Monitoring the traffic variables is also useful for the prediction of short-term future conditions. Our proposal keeps track of the historical series of the observed variables and makes a short term prediction of their future values. This outcome is finally used to detect a possible breakdown and, if true, warn approaching vehicles so that they can change their routes accordingly. Both the monitoring and fully distributed forecasting proposals are detailed in Chapter 6.

Finally, in Chapter 7 we discuss about the contributions of this thesis, identifying some future work.

[2] `http://tti.tamu.edu/documents/mobility-report-2011.pdf`

Part I

Infotainment optimizations

Chapter 2

Efficient IPv6 address autoconfiguration

In this chapter we focus on the auto-configuration of IP addresses by nodes of a Vehicular ad-hoc network (VANET). The internet was built over the robust Internet Protocol (IP) version 4, but emerging technologies like VANETs can benet from functionalities introduced by the new version (IPv6) and are therefore expected to generate a flywheel effect for its deployment. For this reason, all of the aforementioned international committees dening architectures for vehicular communication have included a native IPv6 stack in their protocol stacks. IPv6 provides some standardized mechanisms of IP address auto-configuration, both stateless [23], [24] and stateful [25] that cannot – or at the very least are hard to – be applied without any modification in vehicular environments. The main causes of this fact are the multi-hop nature of VANETs and their lack of a single multicast-capable link for signaling, that prevent current IP address auto-configuration related protocol specifications from being used as-is in VANETs. Therefore, a key research issue is how to auto-configure IPv6 addresses in a VANET. The same problem occurs in general in any unmanaged multi-hop network. Among these, Mobile Ad-hoc Networks

(MANETs) have received a remarkable attention in the research area for years, and there even existed a working group in the IETF[1], called AUTOCONF, that was chartered to work on the standardization of an address auto-configuration solution for MANETs [26].

2.1 State of the art

There are two main approaches that can be followed to integrate IP in a multi-hop vehicular network:

1. Making the IP layer fully aware of the multi-hop nature of VANETs. In this case, the VANET can be defined as a set of IP routers that are interconnected by a multitude of IP links. The high dynamics of each individual link strongly contributes to the overall addressing and routing management overhead. In particular, in order to understand this complexity, we recall the assumption underpinning IP routing, which requires IP addresses assigned to nodes terminating different links to belong to non-overlapping prefixes. Two IP prefixes `p::/l_p` and `q::/l_q` are non-overlapping if and only if there is no IP address `p::a/l_p` configured from `p::/l_p` that also belongs to `q::/l_q`, and vice versa[2]. In order to enable IP routing, an overwhelming amount of short-lived routes is required, posing extremely challenging management issues.

   An example of solution that falls in this category and is particularly designed for VANET environments is the Vehicular Address Autoconfiguration (VAC) solution, proposed by Fazio *et al.* in [1]. This solution exploits the VANETs topology and an enhanced DHCP service with dynamically elected leaders to provide a fast and reliable IP address configuration. VAC organizes leaders in a connected chain such that every node (vehicle) lies in the communication range of at least one leader. This hierarchical organization allows limiting the signaling overhead for the address management tasks. Only leaders communicate with each other to maintain updated information on configured addresses in the network. Leaders act as servers of a distributed DHCP protocol and normal nodes ask leaders for a valid IP address whenever they need to be configured. The main drawbacks of this solution are the assumption of linear topology and group movement which limits the applicability scope, the overhead due to the explicit management signaling (e.g., between leaders) and the possible security threat due to the critical tasks carried out by the leaders. Some of the solutions proposed for Mobile Ad Hoc Networks (MANETs) [26] may also be used for vehicular networks. Most of these solutions and VAC share the

[1] Internet Engineering Task Force: `http://www.ietf.org/`

[2] For example, `2001:DB8:1:1::/64` and `2001:DB8:1:2::/64` are non-overlapping prefixes, while `2001:DB8:1::/48` and `2001:DB8:1:2::/64` are overlapping.

problem that they require modifications to the IP stack of the nodes, as they do not rely on existing standardized IPv6 address auto-configuration solutions.

2. Hiding the multi-hop nature of VANETs from the IP layer. In this approach, the concept of IPv6 link is extended to a set of nodes which might not be directly reachable within one physical hop. A protocol located below IP presents a flat network topology, ensuring that the link seen by the IP layer includes all the nodes of the extended set, even those that are not directly reachable. In this case, existing IP address auto-configuration mechanisms could be used with minor modifications.

   This last approach was followed by the European GeoNet project[3], which contributed to the solution finally standardized by the European Telecommunications Standards Institute (ETSI). Two similar solutions have been proposed: Geographically Scoped stateless Address Configuration (GeoSAC) [3], initially proposed before GeoNet started, and further developed during the lifetime of the project; and [4], that adopts this same concept but has many and essential differences in the realization. The latter solution does not assure compatibility with legacy IPv6 protocol implementations and requires the IPv6 protocol to be geo-aware.

   In this chapter we focus on the solution adopted by the ETSI Technical Committee for Intelligent Transportation Systems (TC ITS), which follows the second approach, hiding the multi-hop nature of the VANET from the IP layer. We next present this system architecture and define the terms used in the rest of the chapter.

2.1.1 ETSI TC ITS IPv6 Integration System Architecture

ETSI TC ITS is developing a set of protocols and algorithms that define an harmonized communication system for European ITS applications taking into account industry requirements like in particular those coming from the Car-to-Car Communications Consortium.

In the ETSI TC ITS network architecture [27], vehicles are equipped with devices called Communication and Control Units (CCUs), which implement the ETSI protocol stack (see Figure 2.1, in which only the part of the stack involved in IPv6 communications is shown). Vehicles can communicate with each other or with fixed roadside ITS stations (also called Roadside Units, RSUs) installed along roads. CCUs and RSUs implement the same network layer functionalities and form a self-organizing network. RSUs can be connected to a network infrastructure, most likely an IP-based network. On-board application hosts including passenger devices attached to the vehicle on-board system are called Application Units (AUs). Passenger devices are assumed to have a standard IPv6 protocol stack, whereas CCUs act as

[3] http://www.geonet-project.eu/

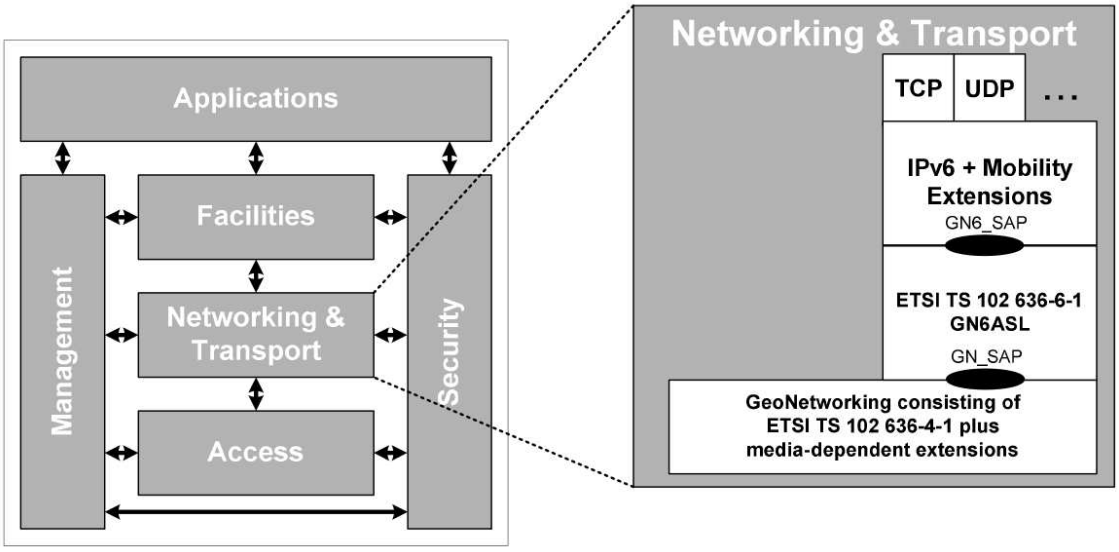


Figure 2.1: GN6ASL in the ITS station architecture.

gateways for the in-vehicle network optionally enhanced with the Network Mobility Basic Support protocol [28].

The ETSI GeoNetworking (GN) protocol [29], published in 2011, plays the role of a sub-IP layer, offering a flat network view to the IPv6 layer and dealing with the multi-hop routing within the VANET (nodes within the same area – i.e., attached to the same IP link – might not be directly reachable, but are portrayed as such by the sub-IP layer). The ETSI has standardized a protocol adaptation sub-layer referred to as the GN6ASL (GeoNetworking to IPv6 Adaptation Sub-Layer) [30] which allows for the transport of IPv6 packets by ETSI GeoNetworking protocol, enabling sub-IP multi-hop delivery of IPv6 packets. The ETSI GN geo-broadcasting capability is used by the GN6ASL in order to shape link-local multicast messages to geographical areas.

Figure 2.2 shows the subset of the ETSI TC ITS system which is relevant to understand how IPv6 is integrated in the ETSI geonetworking architecture and the way the ETSI GN layer is used to logically create links – called Geographical Virtual Links (GVLs) – mapped to areas – called GVL Areas. We will explain this in more detail in Section 2.2. We want to highlight here how IP packets are sent in the system, using the scenario depicted in Figure 2.2. Let us suppose a device within Vehicle C wants to communicate with a node in the Internet. For that communication to happen, the Vehicle C has to send packets to the RSU of its area – that is the next hop at the IP layer – and this requires at the ETSI GN layer Vehicle C to send packets to Vehicle B, which forwards them to Vehicle A, that finally delivers them to the RSU. Note that this multi-hop forwarding is required because

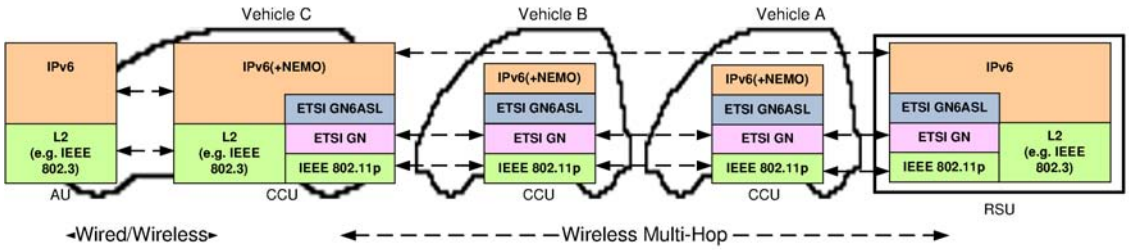


Figure 2.2: IPv6 packet forwarding within an area, and affected protocol layers.

Vehicle C is not within the radio coverage of the RSU. This example shows that in a system architecture based on short range communication devices, the effective provisioning of Internet-based applications over multi-hop communication strongly depends on mobility. Single-hop vehicular Internet access based on WLAN has already been investigated in highway scenarios [31], concluding that the link between CCU and RSU is stable enough to allow for several types of applications. When considering multi-hop communication, the applicability scope of Internet-based applications might need to be reduced to lower speed scenarios (e.g., urban or semi-urban), to a proper ratio of CCUs per installed RSU and to a realistic maximum number of hops (to be determined). Section 2.3 addresses these particular issues, assessing under which conditions it is realistically feasible to support IP unicast multi-hop communications.

2.2 ETSI Stateless address autoconfiguration

The ETSI specification devoted to the integration of IPv6 and the geonetworking architecture not only describes how IPv6 packets are exchanged between ITS stations and how the GN6ASL is presented to the IPv6 layer as a link-layer protocol, but also explains how IPv6 addresses can be automatically configured by ITS stations, namely CCUs. The specification [30] only considers the use of stateless address autoconfiguration schemes, as stateful ones present higher latencies (due to the several round-trip time signaling messages) and require greater management effort. Manual configuration is also not recommended.

The ETSI solution is based on the Geographically Scoped stateless Address Configuration (GeoSAC) solution [3], which can be considered as one particular realization of the ETSI standardized mechanism. In the rest of this thesis we refer to the ETSI IPv6 address stateless autoconfiguration solution as ETSI SLAAC.

ETSI SLAAC adapts the standard IPv6 SLAAC (Stateless Address Auto-Configuration) mechanism so it can be used in multi-hop vehicular ad-hoc networks, by taking advantage of the geographical location awareness capabilities of the vehicles.

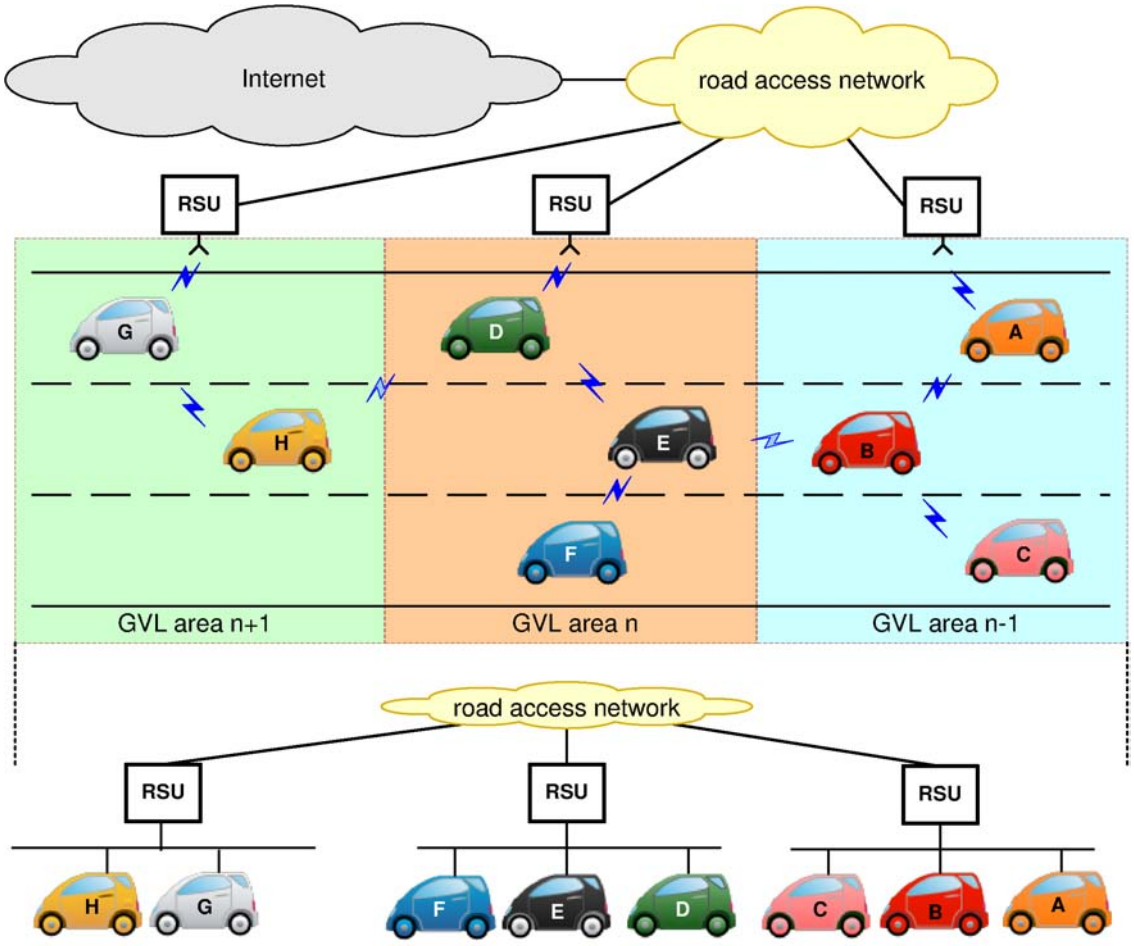


Figure 2.3: Geographical area partitioning and IPv6 virtual link abstraction

In ETSI SLAAC, the concept of IPv6 link is extended to a well defined geographical area (i.e., GVL area) associated with a point of attachment to an infrastructure-based network that plays the role of the IPv6 Access Router (AR).

The GeoNetworking-IPv6 Adaptation Sub-Layer (GN6ASL) (see Figure 2.1) is a sub-IP layer sitting on top of the ETSI GN layer. The ETSI GN layer deals with ad-hoc routing by using geographic location information, while the GN6ASL presents to the IPv6 layer a flat network topology. Consequently, the link seen by the IPv6 layer includes nodes that are not directly reachable but are portrayed as such by the sub-IP layer (see Figure 2.1). This layer provides IPv6 with a link-local multicast capable link, the Geographical Virtual Link (GVL), which includes a non-overlapping partition of the VANET formed by all nodes within a certain geographical area (the GVL area).

Each GVL area is managed by at least one RSU, that acts as an IPv6 Access Router and sends standard IPv6 Router Advertisements (RA), carrying the IPv6 prefix(es) inside the Prefix Information Option (PIO). Nodes receiving the Router Advertisements can then

build a valid IPv6 address out of the included IPv6 prefix, following the standard SLAAC mechanism, i.e., the host generates an address by joining the prefix received from the Router Advertisement and the network identifier derived by its MAC address.

The link-local multicast capability emulation is achieved by relying on the geo-multicast/geo-broadcast capabilities provided by the ETSI GN layer. In particular, in order to be link-local multicast capable, an IP link must provide symmetric reachability [23], which is normally not accomplished by virtual links spanning multiple physical links due to the lack of reference boundaries. Link-local multicast packets are forwarded with geographical knowledge, so that a node processes a packet only if it was addressed to the area where the node is located. The geographic scoping provides non-variable virtual link boundaries which enable symmetric reachability. For Router Advertisements, this means that Router Advertisements must be delivered to – and only to – the nodes that are part of the same IPv6 link, nodes that are actually connected via multiple wireless hops. If a multi-hop path exists, all the nodes within the area will receive a copy of the Router Advertisement, and the IPv6 instance running above the geonetworking will process the message as if the node was directly connected to the access router that issued the message.

It is assumed that MAC addresses (or a different identifier that can be used for IPv6 address generation purposes) of vehicles are unique, at least within macro-regions where vehicles are sold and can potentially communicate with each other (e.g., a continent). This property in fact is highly desirable for security and liability reasons, as it would allow *(i)* forensic teams to rely on vehicular communications to reconstruct accident scenes or other critical situations and, *(ii)* to detect malicious nodes and reduce considerably the effects of network attacks. Despite uniqueness of identifiers, privacy of users can be protected by equipping vehicles with sets of unique identifiers to be used for limited intervals as *pseudonyms* [32]. These identifiers could be assigned by authorities and, when coupled with the usage of digital certificates and cryptographic protection [33], this mechanism can accomplish support for liability as well as privacy protection from malicious users (commonly referred to as *revocable privacy*). Assuming that the IPv6 prefix announced by the RSU is exclusively assigned to this area, the address uniqueness is verified, and therefore no Duplicate Address Detection (DAD) mechanism is required. Note that the proposed solution could be applied to multiple RSUs acting as bridges connected to one single Access Router. This might be a good deployment choice in scenarios where single-hop connectivity to the infrastructure is preferred while it is also required to reduce the number of IPv6 address changes (e.g., city environment).

A technique that maximizes the benefits of ETSI SLAAC consists in shaping the GVL areas assigned to the RSU in a adjacent and logically non-overlapping fashion, as depicted in Figure 2.3. By doing so, the following key advantages are obtained: *(i)* unequivocal

gateway selection is achieved with the infrastructure having full control on it[4], as only one RSU is assigned per geographical area; *(ii)* a network partitioning is obtained that supports movement detection procedures of IPv6 mobility and also allows for location-based services. In particular, a vehicle moving across regions served by different Access Routers experiences a sharp sub-net change, without traversing *gray areas* where Router Advertisements are received from multiple access points (potentially leading to *ping-pong effects*).

Before characterizing and analyzing the performance of the ETSI SLAAC solution, we next analyze under which conditions it is realistically feasible to support IP unicast multi-hop communications in a vehicular environment.

2.3 Effectiveness of Vehicular Multi-hop Communications

Vehicular networks using short-range wireless technologies, such as IEEE 802.11-based ones, rely on multi-hop communications to extend the effective coverage of the Road Side Units deployed on the roadside. One of the main challenges that VANETs pose is the minimum degree of technology penetration that is needed in order to ensure that there is enough density of communication-enabled vehicles to support multi-hop connectivity between the intended peers (e.g., for the case of Internet communications, between the vehicle and the RSU). This problem becomes even more problematic during the time of the day when roads are less busy. In these environments communications can become difficult because radio devices often operate at their design limits (large distances, multi-path signal propagation, critical packet length vs. channel coherence time ratio, etc.), which amplifies the effect of layer-2 inefficiencies due to hidden node scenarios. Furthermore, the probability of having a multi-hop path between two nodes is lower in sparse scenarios. On the other hand, when roads become more crowded, speeds are lower, links are more reliable and the chances for two arbitrary nodes to be connected by at least one stable multi-hop path are higher.

Deploying vehicular networks without dead zones (i.e., areas not served by any RSU) is economically inefficient in non-urban locations. As we have mentioned above, in the ETSI TC ITS architecture vehicles form a self-organized multi-hop network. This multi-hop network is used to forward packets between the RSU and the CCUs within the RSU's area of influence (i.e., associated GVL area), and therefore extends the effective coverage area of the RSU. In order to assess the feasibility of vehicular communications in practical scenarios, it is necessary to evaluate whether wireless multi-hop communications are possible in different vehicular situations. Connectivity in ad-hoc networks has been

[4]More precisely, in this solution gateway selection is performed by the infrastructure itself and not by the nodes as in many MANET approaches.

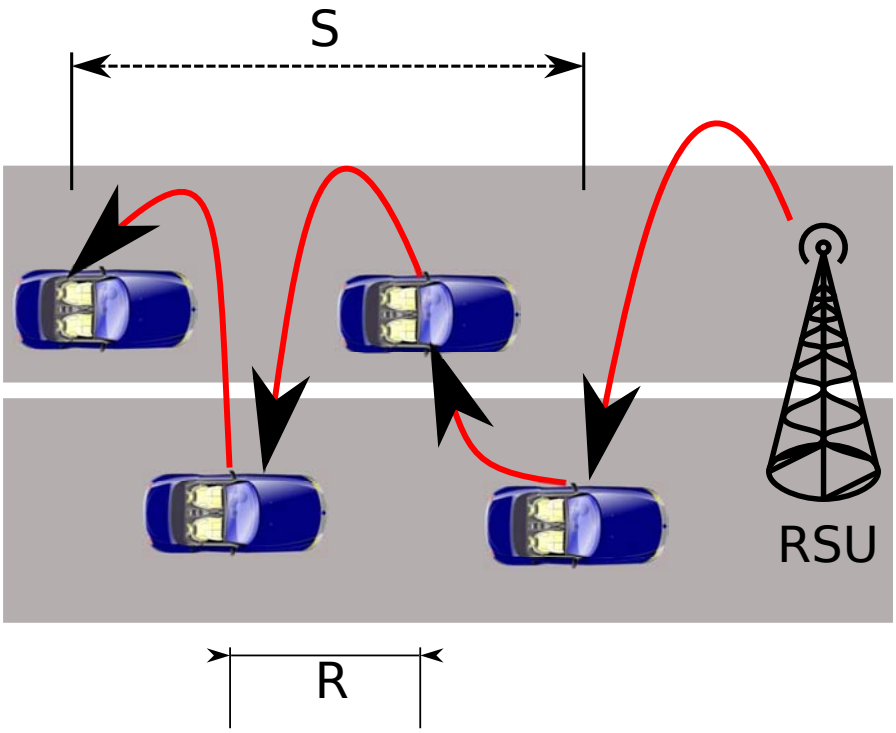


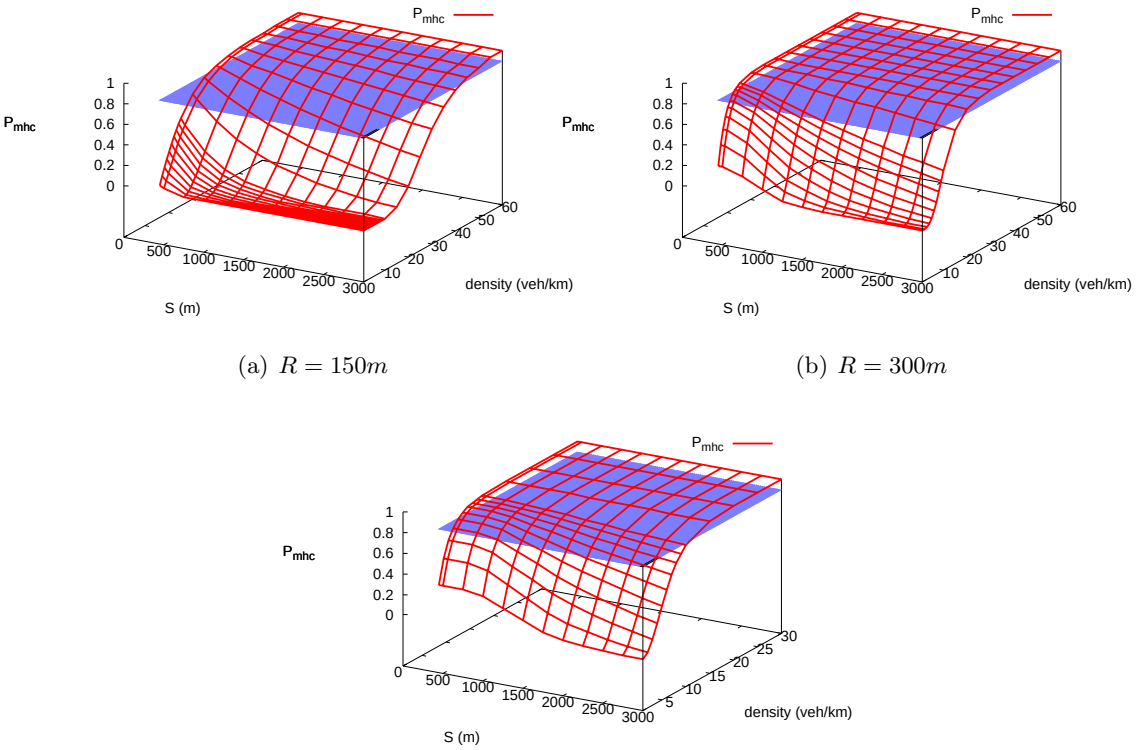
Figure 2.4: Multi-hop connectivity between a car and an RSU.

thoroughly studied, but vehicular networks have special characteristics (in particular, the mobility patterns and range of speeds) that require specific analysis. Some studies have contributed to the characterization of connectivity in vehicular networks (for example, [21, 34], which focus on inter-vehicle connectivity), but in this chapter, we model and analyze the probability of having a multi-hop path between a sender and a receiver, studying the impact of different parameters, such as vehicular speed and density, wireless radio coverage, etc. We present our mathematical model first and then validate it via simulations.

Given two nodes separated by a distance S, $P_{mhc}(S)$ is the probability of having multi-hop connectivity (mhc) or, in other words, the probability that one chain of interconnected vehicles between the two nodes exists. This probability depends – as we show below – on the distance between the two nodes, the radio coverage and the vehicular density. Figure 2.4 shows an example of a chain of interconnected vehicles between a car and an RSU. We model the distance D between consecutive vehicles (inter-vehicle spacing) as exponentially distributed [12, 21], with parameter β, with its Probability Density Function (PDF) given by:

$$f_D(d) = \beta e^{-\beta d}, \quad d \geq 0, \tag{2.1}$$

where β is the vehicular density.



(a) $R = 150m$

(b) $R = 300m$

(c) $R = 450m$

Figure 2.5: P_{mhc}: simulation results (cut in 90% probability).

Let R be the wireless coverage radius. The distance between two consecutive vehicles that are part of a connected multi-hop chain of vehicles (the inter-vehicle gap is smaller than R) follows a truncated exponential distribution [35]:

$$f_{te}(d) = \begin{cases} \frac{\beta e^{-\beta d}}{1-e^{-\beta R}}, & 0 < d < R, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

The length of a multi-hop connected chain of $n+1$ vehicles (Y) can be represented as the sum of n independent exponential truncated variables. The PDF of Y can be obtained by the method of characteristic functions [35]:

$$g_Y(y;n) = \frac{(\beta b)^n}{(n-1)!} e^{-\beta y} \sum_{k=0}^{k_0} (-1)^k \binom{n}{k} (y - kR)^{n-1}; \qquad k_0 R < y < (k_0+1)R \tag{2.3}$$

where $k_0 = 0, 1, \cdots, n-1$, and $b = (1 - e^{-\beta R})$.

Let $a = (k_0{}' + c)R$, where $k_0{}'$ is an integer, and $0 \leq c < 1$. The Cumulative Distribution

Function (CDF) of Y evaluated at a is $G_Y(a;n) = \int_0^a g_Y(y;n)dy$:

$$G_Y(a;n) = \frac{1}{(1-e^{-\beta R})^{-n}} \sum_{k=0}^{k_0} (-1)^k \binom{n}{k} e^{-\beta k R} \qquad Q[2({k_0}' - k + c)R\beta, 2n]. \tag{2.4}$$

where $Q[u,w] = P\left(\chi^2(w) < u\right)$ and $\chi^2(w)$ is a chi-square variable with w degrees of freedom. Since the probability $P(i)$ of having a connected chain of i hops is given by $(1-e^{-\beta R})^i\ e^{-\beta R}$, the PDF and CDF of the length (L) of a connected multi-hop chain of vehicles can be derived using the law of total probability:

$$f_L(l) = \sum_{i=0}^{\infty} P(i \text{ hops}) g_Y(l;i) = \sum_{i=0}^{\infty} (1-e^{-\beta R})^i e^{-\beta R} g_Y(l;i), \tag{2.5}$$

$$F_L(l) = P_L(L \leq l) = \int_0^l f_L(u)du = \sum_{i=0}^{\infty} (1-e^{-\beta R})^i e^{-\beta R} G_Y(l;i). \tag{2.6}$$

Based on this, $P_{mhc}(S)$ is given by:

$$P_{mhc}(S) = 1 - F_L(S). \tag{2.7}$$

Another factor that should be considered to assess the feasibility of vehicular multi-hop communications is the number of available lanes in a road. Our previous analysis is valid regardless of the number of lanes, thanks to the properties of the exponential distribution. If we consider several lanes, and in each one we model the spacing between cars by an exponential distribution, not necessarily with the same mean (the different lanes can have different car densities), the resulting space between cars in the road (independently from the number of lanes) is exponentially distributed with mean the average of the means in each lane. Therefore, we do not assume any particular number of lanes throughout the rest of the chapter, unless indicated explicitly. Note that we are approximating the car distribution assuming that there is no correlation between the lane geometry and the car distribution. This means that we disregard the spatial correlation introduced by traffic regulation and congestion. The consequences of this assumption are evaluated in the next section.

In order to validate our analysis of P_{mhc}, we performed a large amount of experiments via simulation under different traffic conditions. The simulator[5] was developed using MATLAB and it implements the scenario described in this section, namely vehicles

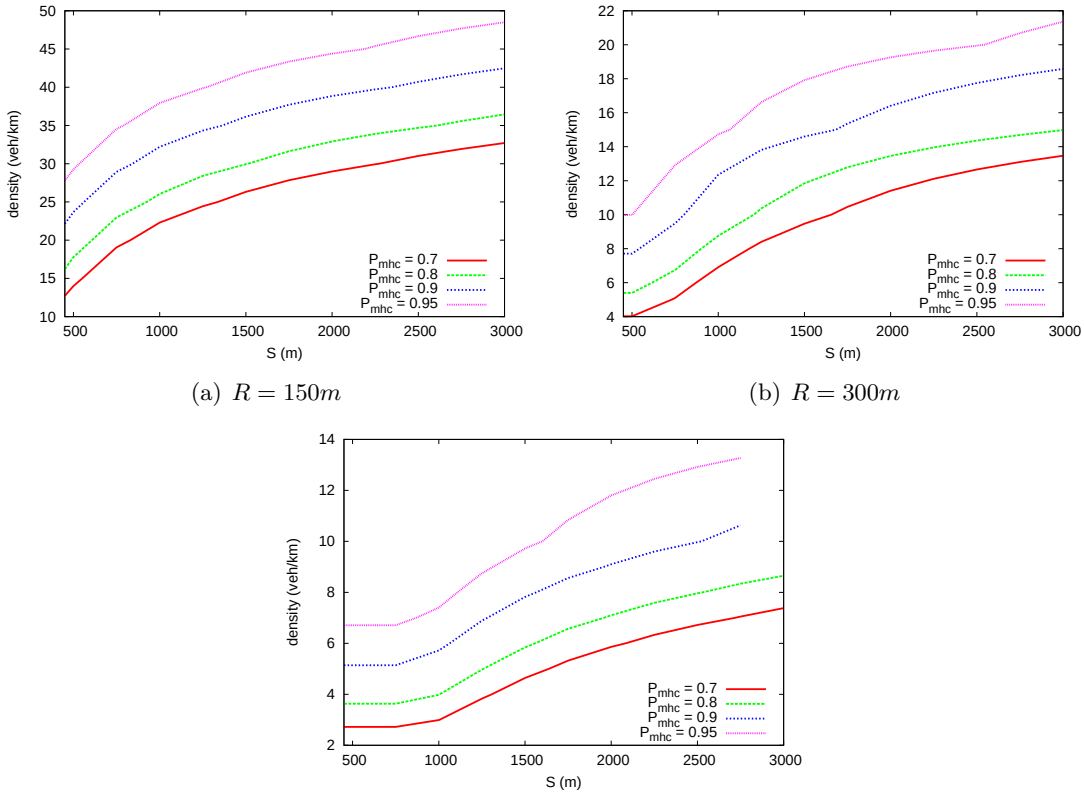
[5]The code of this simulator is available at
`http://fourier.networks.imdea.org/people/~marco_gramaglia/sims/ETSI-SLAAC/`.

distributed in a one-dimensional road, travelling at a predefined and constant speed, with an exponential inter-vehicular distance and a maximum wireless radio coverage, assuming an ideal wireless technology (no packet losses nor collisions and infinite bandwidth). Although the simulator does not consider a real wireless model, we argue that it is enough to show the correctness of our mathematical model, as it fully implements the behavior we are modeling. Obtained results show that our mathematical analysis perfectly models the probability of having multi-hop connectivity (assuming the aforementioned simplifications). Simulation and experimental results are shown in Section 2.4, where we use a more advanced simulator (OMNeT++) that does include a complete wireless model to validate our formulation of the configuration time of the ETSI SLAAC solution.

In the following, we focus on analyzing the scenarios in which unicast communications using a multi-hop vehicular network are feasible. There are three parameters that have an impact on the probability of having multi-hop connectivity between two nodes:

- The distance S between the nodes. The larger this distance is, the lower is the probability of having connectivity. If we focus on the vehicle-to-Internet scenario, this value would be related to the distance between a moving vehicle and the fixed RSU, and therefore it depends on how RSUs are deployed.
- The vehicular density β. The probability of having connectivity increases with the vehicular density. The density depends on the traffic conditions (i.e., the time of the day and road) and the type of road (i.e., there are roads more congested than others). Vehicles density and speed are usually correlated as well, since the minimum safety distance between vehicles depends on the speed [36].
- The wireless coverage radius R. The effective radius depends on the specific wireless access technology, the transmission power at the antenna, the antenna radiation pattern and the instantaneous channel response. The probability of having multi-hop connectivity is obviously very much affected by R, shorter values leading to lower probabilities.

If we fix the value of R, which is equivalent to assuming a reference system, it is interesting to study which is the minimum vehicular density required to ensure a certain probability of multi-hop connectivity between two nodes, depending on their distance. Figure 2.5 depicts the simulation results obtained for three different values of R (150, 300 and 450m), which represent a realistic range of wireless coverage radius for wireless access technologies expected to be used in vehicular communications [37]. The results are plotted in three dimensions, so it can be observed how the vehicular density β and the distance S between the two nodes affect the multi-hop connectivity probability. An horizontal plane for $P_{mhc} = 0.9$ is also depicted in the figures, so we can observe which



(a) $R = 150m$

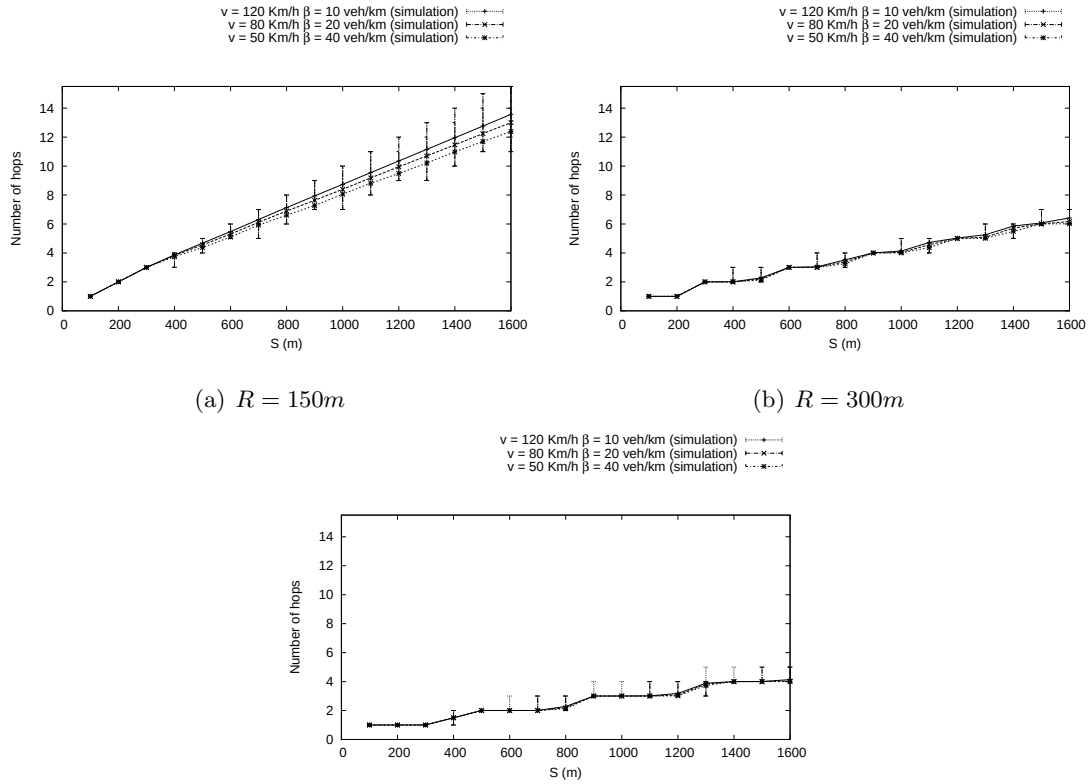
(b) $R = 300m$

(c) $R = 450m$

Figure 2.6: Contours for different values of P_{mhc}.

are the combinations of β and S that result in values of P_{mhc} higher than 90%. The cut (intersection) of horizontal planes corresponding to probabilities of 0.7, 0.8, 0.9 and 0.95 and the 3D curve are shown in Figure 2.6. Using this figure we can find out which is the minimum vehicular density required to achieve a minimum multi-hop connectivity probability between two nodes separated by a given distance. Let's take for example the reference value of $S = 1000m$. From the results in Figure 2.6, we can conclude that if the coverage radius R is 150m, a vehicular density of approximately 35 veh/km or higher ensures that there is multi-hop connectivity in the 90% of the cases. Similarly, 15 veh/km are enough if R is 300m, and 8 veh/km for $R = 450m$. It is important to highlight that these densities are quite low, and that therefore are likely to be found in realistic scenarios with typical traffic conditions. We selected the following three scenarios which mostly cover a wide spectrum of potential traffic scenarios:

- *Urban road:* high vehicular density ($\beta = 80veh/km$) and low speed ($v = 50km/h$).
- *City highway:* moderate vehicular density ($\beta = 50veh/km$) and moderate speed ($v = 80km/h$).
- *Motorway:* low vehicular density ($\beta = 35veh/km$) and high speed ($v = 120km/h$).



(a) $R = 150m$

(b) $R = 300m$

(c) $R = 450m$

Figure 2.7: Number of hops: simulation results.

As it can be observed from Figure 2.6, it is perfectly feasible to have multi-hop connectivity in these three scenarios for most of the potential deployments (i.e., inter-RSU distances). The probability of multi-hop connectivity is not the only parameter that should be considered when assessing the feasibility of vehicular communications, as the number of hops also plays an important role (i.e., the larger this number is, the lower are throughput and reliability). Figure 2.7 shows the average, minimum and maximum values of the number of traversed hops (only for those communications that can take place, i.e., where a multi-hop chain of vehicles exists) for the same scenarios. From these results we can also conclude that it is not efficient from a performance viewpoint to deploy RSUs which are separated by large distances, as the number of hops would get too high, impacting the performance of the communications. It should be noted that vehicles are expected to be equipped with one single wireless radio interface for multi-hop communications using a self configured VANET[6] and therefore the effective throughput decreases with the number of traversed wireless hops in the VANET.

[6]It is also very likely that in the future, cars are equipped with a dedicated interface for safety-related communications and one (or several, but of different technologies) for IP non-safety related communications.

2.4 Analytical Characterization of the ETSI SLAAC's Performance

The main purpose of an IP address auto-configuration protocol is to provide each node with a valid IP address as soon as possible. In the followings we derive an analytical expression of the time required by the ETSI SLAAC solution to configure an address. The address configuration time (T_{conf}) is the time elapsed since a vehicle enters a new geographical area (therefore loosing the connectivity to the old RSU) till the moment in which it can start using the newly configured global IPv6 address. This time depends on several factors, such as the shape and size of the areas, the configuration of the RSUs and ARs, etc. We consider that the time between two consecutive RAs sent by an RSU (or an AR in case the RSU is working in bridge mode) follows a uniform distribution between a minimum value (*MinRtrAdvInterval*) and a maximum value (*MaxRtrAdvInterval*), which we refer to as R_m and R_M respectively [38].

We use the following additional terminology. Let D_{RSU} be the distance between two adjacent RSUs, R the wireless communication range, β the vehicular density, and v the speed of the vehicles[7].

In order to obtain a mathematical expression for T_{conf}, we first calculate the mean length of a chain of vehicles. Based on that, we are able to find out which is – on average – the length of the gap between the multi-hop chain of vehicles from the unconfigured vehicle and the RSU wireless coverage area $\bar{D}_{gap}$ (see Figure 2.8).

The average distance between two consecutive vehicles can be calculated using Eq. (2.2), and it is given by:

$$\bar{D} = \int_{-\infty}^{\infty} d\ f_{te}(d)\ dd = \frac{1 + \beta R - e^{\beta R}}{\beta(1 - e^{\beta R})}. \tag{2.8}$$

The average length of a chain composed by $i+1$ vehicles is:

$$\bar{L}_{chain}(i+1) = i\bar{D}. \tag{2.9}$$

As already seen in Section 2.3, the probability of having a connected chain composed exactly by $i+1$ vehicles is given by: $(1 - e^{-\beta R})^i\ e^{-\beta R}$. From this, we can calculate the average length of a multi-hop connected chain of vehicles:

$$\bar{L}_{chain} = \int_{0}^{\infty} l\ f_L(l) dl = \sum_{i=0}^{\infty} (1 - e^{-\beta R})^i e^{-\beta R} \bar{L}_{chain}(i+1). \tag{2.10}$$

[7] We consider the speed of all vehicles fixed and constant for simplicity of the model. Simulation results that we will present later show that this simplification does not affect the validity of the conclusion of our analysis.

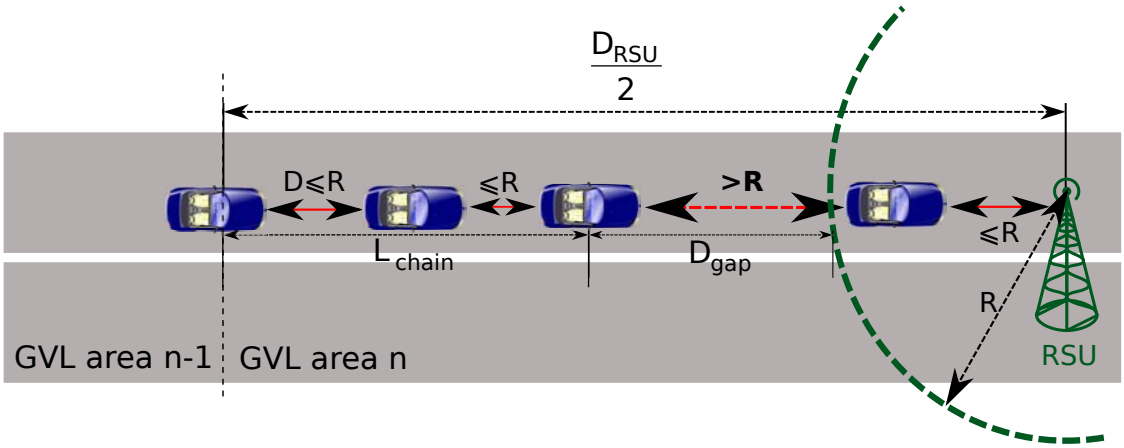


Figure 2.8: ETSI TC ITS IPv6 address configuration

Therefore, the average gap length is given by:

$$\bar{D}_{gap} = \frac{D_{RSU}}{2} - R - \bar{L}_{chain}. \tag{2.11}$$

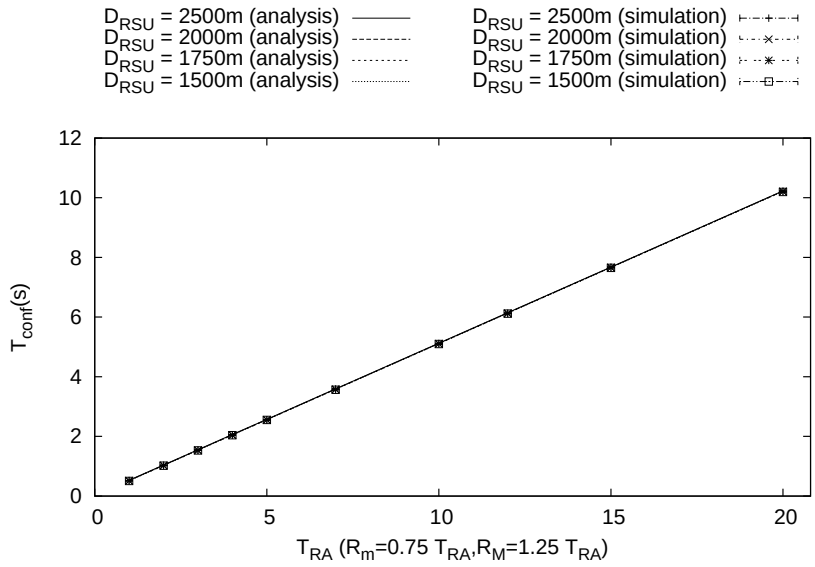
Let T_{RA}^{unsol} be the time elapsed since a vehicle changes area until the RSU sends the next unsolicited RA. The average value of this time is given by [39]:

$$\bar{T}_{RA}^{unsol} = \frac{R_M^2 + R_M R_m + R_m^2}{3(R_M + R_m)}. \tag{2.12}$$

Now we can calculate the average ETSI SLAAC address configuration time ($\bar{T}_{conf}$), by simply considering the two possible configuration situations: *i)* there is on average multi-hop connectivity between the unconfigured vehicle and the RSU (i.e., $\bar{D}_{gap} \leq 0$), and therefore vehicles only need to wait $\bar{T}_{RA}^{unsol}$ for the next unsolicited RA sent by the RSU; *ii)* there is on average no chain of connected vehicles between the unconfigured node and the RSU:

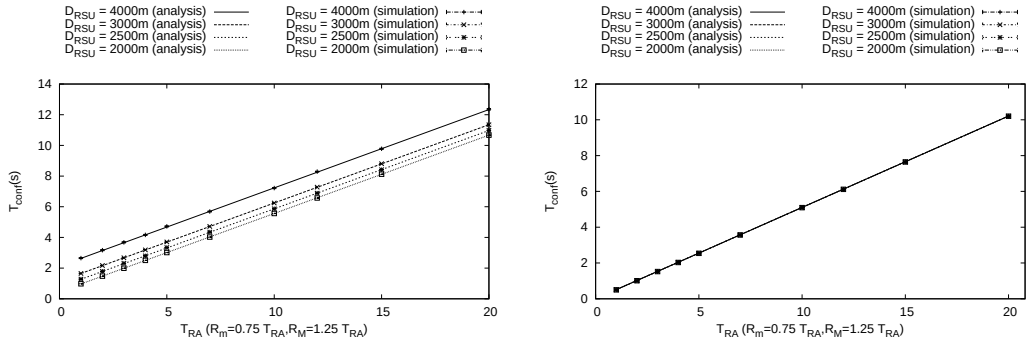
$$\bar{T}_{conf} = \begin{cases} \bar{T}_{RA}^{unsol}, & \bar{D}_{gap} \leq 0, \\ \frac{\bar{D}_{gap}}{v} + \bar{T}_{RA}^{unsol}, & \bar{D}_{gap} > 0. \end{cases} \tag{2.13}$$

In order to validate the accuracy of our model and assess the performance of our solution, we performed the following experiments. We evaluated the configuration time $\bar{T}_{conf}$ under different traffic conditions and for different deployment configuration parameters. The traffic conditions are defined by the vehicular density (β) and speed (v), while the considered deployment configuration parameters are the distance between RSUs (D_{RSU}),



(a) $R = 150m$

Figure 2.9: ETSI SLAAC configuration time (analysis and simulation) for the Urban scenario.

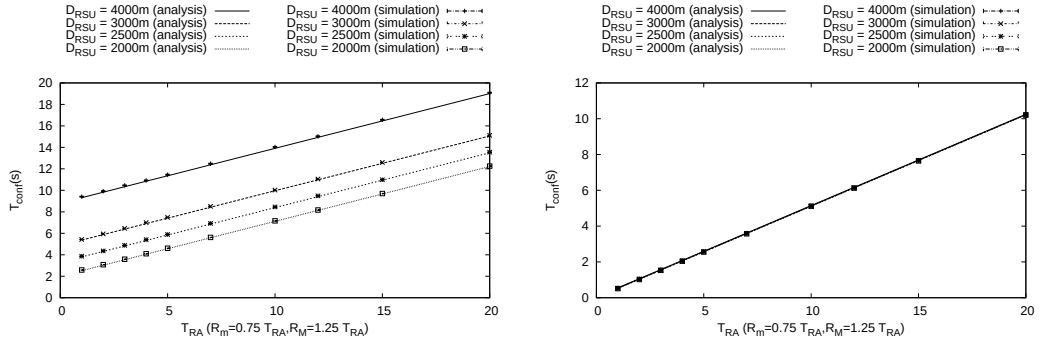


(a) $R = 150m$ (b) $R = 300m$

Figure 2.10: ETSI SLAAC configuration time (analysis and simulation) for the City highway scenario.

the radio wireless coverage of each node (R) and the average time between unsolicited RAs (T_{RA}). The same MATLAB-based simulator that was used in Section 2.3 to assess the effectiveness of multi-hop unicast communications in a vehicular scenario is used in these experiments. Therefore, an ideal wireless technology is assumed. In the next section we also perform an experimental evaluation based on a more complex model, and using the OMNeT++ simulator, that allows us to assess the correctness of the simplifications assumed in our mathematical model of the ETSI SLAAC configuration time, and also to derive some configuration guidelines.

The results obtained from our simulations (with a confidence interval of 95%) are



(a) $R = 150m$ (b) $R = 300m$

Figure 2.11: ETSI SLAAC configuration time (analysis and simulation) for the Motorway scenario.

shown in Figures 2.9–2.11, in which the values calculated from our analysis are also depicted. We make use of the same three scenarios that we used in Section 2.3: urban, city highway and motorway, and we also represent the results for different deployment scenarios (characterized by D_{RSU} and T_{RA}[8]). Note that the range of the average time between Router Advertisements sent by the RSU (T_{RA}) depends on the traffic conditions scenario. This is so because the maximum value that could be configured in a real scenario should allow for vehicles to always have at least one configuration opportunity before changing area, and that means that T_{RA} has to be low enough to allow that a vehicle would be configured – in the worst possible case – when it is within one single hop of the RSU (i.e., the minimum time required by a vehicle to cross the whole coverage area of the RSU should not be higher than R_M). Note that in Figure 2.9 (urban scenario) we only show the case $R = 150m$, as the results for $R = 300m$ and $R = 450m$ are almost exactly the same (the actual difference is negligible). Similarly, for the city highway and motorway scenarios (Figures 2.10 and 2.11) we also skip depicting the results for $R = 450m$, as they are equivalent to those for $R = 300m$. This is due to the fact that in the studied scenarios, the vehicular density proves to be enough to ensure multi-hop connectivity in most of the situations, and therefore $\bar{T}_{conf} \simeq \bar{T}_{RA}^{unsol}$. These are reasonable scenarios in terms of vehicular density, and they are actually the ones in which it makes sense to enable Internet multi-hop communications, as the probability of having multi-hop connectivity to the RSU is high enough, and the configuration time is short enough to support classical IP communications (e.g., infotainment, non-safety). We also analyze later in this chapter sparser scenarios, in which the vehicular density is much lower.

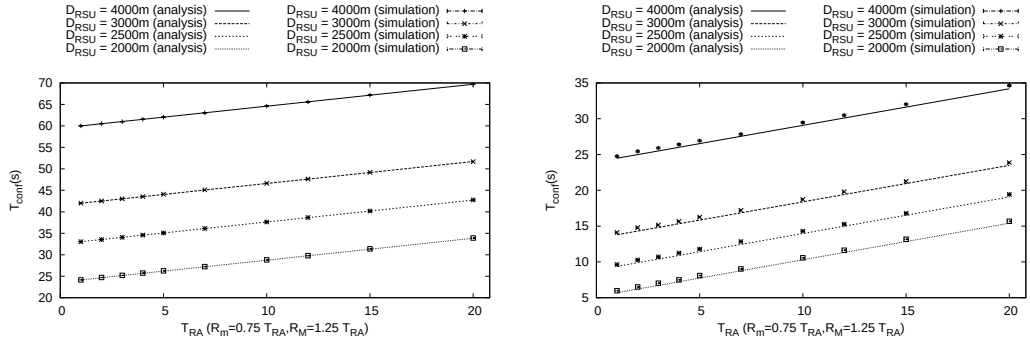
From Figures 2.9–2.11 we can derive different conclusions. First of all, results show that our mathematical analysis perfectly matches our model of the ETSI SLAAC solution configuration time, assuming the simplifications that we have described in this

[8] $R_M = 1.25T_{RA}$ and $R_m = 0.75T_{RA}$.

section, namely: constant and homogeneous speed, perfect collision-free wireless medium, exponential inter-vehicle spacing. Another important conclusion is that in most of the scenarios, the IP address auto-configuration time can be kept very low by properly configuring the time interval between Router Advertisements, without using too aggressive values. Besides, in these scenarios the value of the wireless coverage technology (R) – which depends on the particular access technology, transceiver performance, antenna, and channel conditions – and the distance D_{RSU} between deployed RSUs, do not seem to have a noticeable impact on the resulting configuration time. Nevertheless, we should not forget that shorter values of R combined with larger values of D_{RSU} would lead to longer multi-hop paths, in terms of number of traversed nodes, and therefore lower effective bandwidths. Only for $R = 150m$ and in the motorway scenario (this behavior starts to be noticeable in the city highway scenario), the distance between RSUs has an impact on the configuration time, as the chances to have multi-hop connectivity between an unconfigured node that just enters into an area and the RSU decrease with the distance between them (D_{RSU}). In the motorway scenario, as we could anticipate from the results shown in Figure 2.4, configuration times are higher – though still reasonable – as the probability of having a multi-hop chain between the unconfigured node and the RSU is lower.

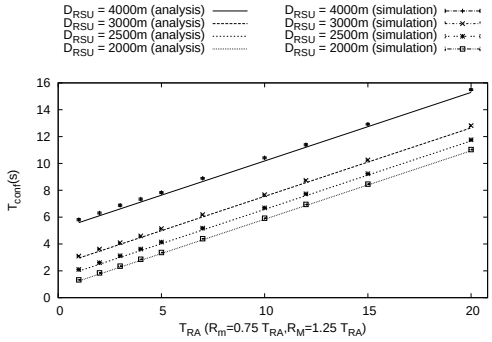
A simple qualitative evaluation of the ETSI SLAAC performance can be done by comparing T_{conf} with the estimated permanency time of a vehicle within a geographical area. For the sake of the example, let's consider a non-extreme case, as the one of an area with a length (D_{RSU}) of 2000m and a wireless radio technology with $R = 300m$, in the city highway scenario (average speed of 80 km/h). In this scenario, a vehicle spends about 90 seconds in the area. By choosing values of T_{RA} smaller than 10 seconds, the ETSI SLAAC solution guarantees that vehicles can have Internet connectivity for more than 80 seconds, as $\bar{T}_{conf}$ is approximately 5 seconds. However, it is important to highlight that the configuration parameters, such as the size and shape of the geographical areas, should be chosen also taking into account the expected traffic conditions. For example, in sparse scenarios, areas should be longer than the physical radio coverage R, while in dense scenarios the opposite case is more appropriate. We derive some simple configuration guidelines in Section 2.5.

In addition to these experiments, we performed some simulations to validate the correctness of our mathematical analysis also in scenarios in which the vehicular density is not high enough to have multi-hop connectivity during most of the time ($\beta = 10$ veh/km, $v = 100$ km/h). Examples of this scenario are city highways and motorways at night, or secondary roads. Results (see Figure 2.12) show that our mathematical analysis also matches the simulation results in sparse scenarios (i.e., with low values of P_{mhc}). Regarding the time required to configure a new address, obtained results confirm what we could anticipate from Figure 2.6, that is, in these scenarios it takes a long time to get an



(a) $R = 150m$

(b) $R = 300m$



(c) $R = 450m$

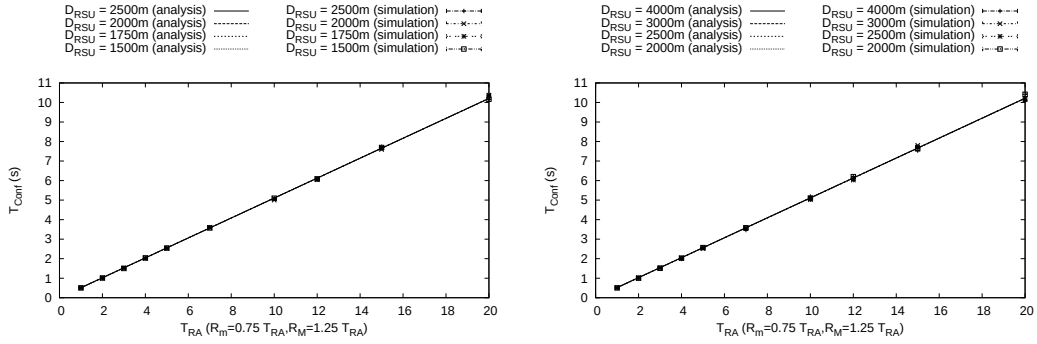
Figure 2.12: ETSI SLAAC configuration time (analysis and simulation) for the Sparse scenario.

address, because in many cases the vehicle is not able to receive an RA until it is within the 1-hop coverage of the RSU. This also means that during a considerable amount of the time a vehicle is visiting a geographical area, it does not benefit from having connectivity with the RSU. It can also be observed that in this kind of scenario R has a bigger impact on the performance, as higher values of R lead to higher multi-hop connectivity probabilities.

2.5 Experimental Evaluation and Configuration guidelines

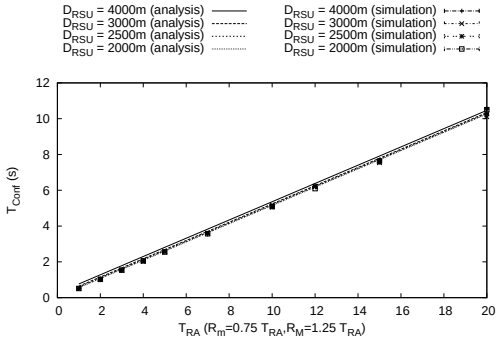
In this section we take a step further and experimentally evaluate the performance of the ETSI SLAAC solution – in terms of address configuration time – eliminating some of the simplifications assumed in the previous section. The goal is to assess if our mathematical model is still good enough when we use a simulation model that better replicates a real environment.

The first step is to use a realistic wireless environment, which is not collision-free and that models the different aspects of the physical and medium access control layers of



(a) Urban road scenario

(b) City highway scenario



(c) Motorway scenario

Figure 2.13: ETSI SLAAC configuration time (analysis and simulation with OMNeT++).

IEEE 802.11. In order to do so, we implemented the ETSI SLAAC solution [9] using Mixim. Mixim[10] is a framework for wireless ad hoc network for the OMNeT++ simulator[11]. It provides the 802.11 MAC layer and many physical layer models (including the widely accepted path-loss, shadowing and large and small-scale fading models [40] [41] [42]). The simulation scenario consists of a road segment where vehicles travel within a homogeneous flow. Vehicles' starting position is generated following an exponential distribution (speed and density are defined by the type of scenario: urban, city highway or motorway, so the number of nodes involved in the simulation changes depending on the vehicular density). At the end of road segment, nodes enter a GVL area (D_{RSU} long) where a RSU is placed half the way ($D_{RSU}/2$ far from the area border). Vehicles are equipped with a standard 802.11g MAC layer, with a bitrate of 6Mb/s. When the simulation runs, the first vehicles are excluded from the results' recollection as they were already located inside the GVL area, but they are needed to build the multi hop chain and let the subsequent vehicles be configured. When a node receives the first Router Advertisement after crossing the area border, its configuration time is recorded. Each simulation is run 20 times using the

[9] The code of this simulator is available at `http://fourier.networks.imdea.org/people/~marco_gramaglia/sims/ETSI-SLAAC/`.

[10] `http://mixim.sourceforge.net/`

[11] `http://www.omnetpp.org/`

same topology with a different seed, and for each parameter set 50 different topologies are generated. Then, the results are averaged on a population of at least $1000 \times nCars$ values, where $nCars$ depends on the chosen vehicular density but, being the road segment length 15km, this value is not smaller than 150. The parameters used in the simulations are summarized in Table 2.1.

The obtained simulation results are shown in Figure 2.13 for the three scenarios we used and for different inter-RSU distances. The results obtained from our mathematical model are also depicted in the figure. Note that since we are using a realistic wireless model, the value of R is no longer a constant value. In order to select an appropriate value to be used in our mathematical model, we performed simulations with OMNeT++ aiming at finding the average wireless coverage radius resulting from the wireless layer settings used in the simulations. The resulting value is $R = 225m$. From the results shown in Figure 2.13, we can observe that our mathematical model approximates quite well the experimental results obtained with OMNeT++.

The last simulation experiment we performed to validate our mathematical analysis is the following. Using the OMNeT++ simulator, we take the position and speed of vehicles on a real road from real traffic traces, and measure the ETSI SLAAC configuration time. The traces were taken at the 4-lane highway A6 in Spain, in the outbound direction from Madrid, and account for the traffic from 8:30 to 9:00 a.m. (which can be considered as near to rush hour). The total amount of samples is 2985. For each sample, we have a time-stamp and the speed of the vehicle. We consider that the measurement point is the border between two geographical areas and that each vehicle keeps the same speed while traversing the area. In our simulation environment, we fix the distance between two RSUs to 2000m. We plot in Figure 2.14 the results obtained from the simulation and our mathematical analysis. In Eq. (2.13) we use the vehicular density calculated from the traces ($\beta = 38.2$ veh/km) and the average speed ($v = 95.11$ km/h). As it can be observed from Figure 2.14, there is a small gap between the simulation results and our mathematical model, although the model still approximates the real performance. This gap is due to the fact that our model is a simplification of the real scenario, and as a result, our analysis is optimistic. Our model does not consider the non-idealities of the wireless media, assumes a perfect exponential inter-vehicle spacing and considers that all vehicles travel at the same speed. We learned from the previous experiments with OMNeT++ (Figure 2.13) that the deviation from considering an ideal wireless media seems to be pretty small. Therefore, we conjecture that the deviation found in Figure 2.14 are due to the fact that inter-vehicle spacing does not exactly follow an exponential distribution[12], and also the fact that vehicles do not all travel at the same speed.

[12] We analyzed the inter-vehicle spacing from the used traces and it is close to an exponential distribution, as suggested in [12], but with some minor deviation. A more detailed analysis of vehicular traces is proposed in chapter 5.

Scenario	Speed [Km/h]	Density [veh/km]
Urban	50	80
City Highway	80	50
Highway	120	35
MAC Layer	802.11g	
Bitrate	6 Mb/s	

Table 2.1: Simulation parameters.

We have shown via simulation experiments that the ETSI SLAAC performance in terms of IP address configuration time is good enough for IP Internet-alike applications. We have also validated our mathematical model for $\bar{T}_{conf}$ and analyzed how the different deployment and configuration parameters, namely D_{RSU}, R and T_{RA}, affect the obtained performance. Based on that, we provide configuration guidelines that enable the ETSI SLAAC solution to guarantee a certain performance (i.e., a target configuration time), depending on the addressed scenario. We do not include R in this analysis, as this is a value that depends on the wireless technology in use, and we consider this as an external parameter that is known beforehand (all cars use the same radio technology).

- If RSUs can obtain information about the vehicular density and speed of the road segment within its assigned geographical area (this information can be obtained in real time from sensors deployed in the road, or via statistical prediction based on historical records), RSUs can dynamically calculate the minimum required T_{RA} to achieve the target configuration time, based on the mathematical analysis described in Section 2.4, namely in Eq. (2.13). Note that we assume that RSUs can be provisioned with the location of neighboring RSUs, and therefore that they know D_{RSU}. The value of T_{RA} can then change dynamically to react and adapt to traffic conditions. In the case that RSUs do not have enough computation resources, they can be provided with a precomputed set of configuration parameters.

- If RSUs do not have any information about current vehicular density, two different configuration strategies could be followed. An optimistic approach would be based on assuming that the vehicular density is enough to safely assume that vehicles have multi-hop connectivity with the RSUs with high probability. In this case, the RSU can make use of Eq. (2.12). On the other hand, a more conservative approach in which no assumption can be made about the vehicular density would be based on selecting a P_{mhc} value that still supports reasonable levels of connectivity and makes use of the mathematical analysis described in Section 2.4. Note that we also assume here that RSUs are configured with D_{RSU}. In this case, dynamic reconfiguration of the RSU is not possible or just very limited (e.g., configuration based on the time of the day).

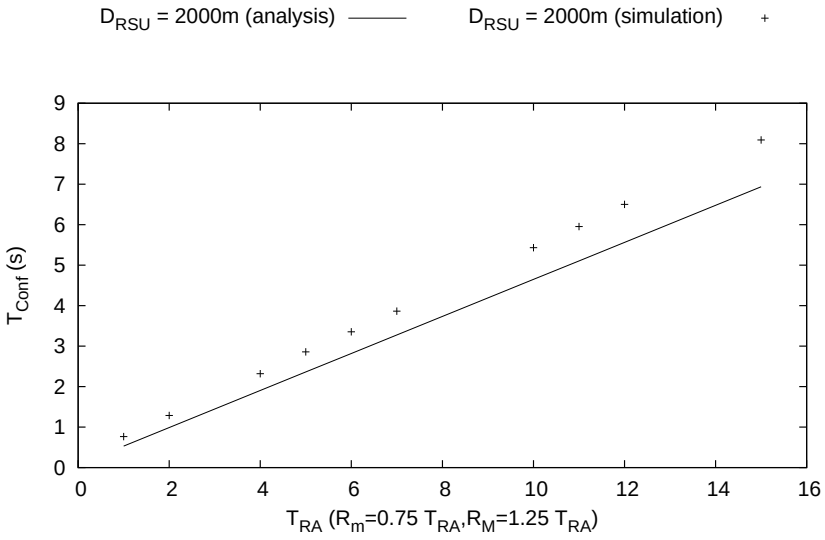


Figure 2.14: ETSI SLAAC configuration time (analysis and simulation with OMNeT++ and real traffic traces).

2.6 Reducing the IP address auto-configuration time and signaling overhead

This section summarizes the main reasons why optimizing the mechanism used to provide nodes with an IPv6 address is important in the context of vehicular communications. The benefits from this optimization are two-fold: reducing the IP address configuration time and minimizing the signaling overhead caused by the solution.

Because the concept of an IPv6 link is associated with a specific GVL area, in ETSI SLAAC, each vehicle must stop using its old IP address and configure a new one every time it changes areas. This reconfiguration involves a time during which the vehicle cannot communicate, lasting until a valid IP address is configured and becomes usable. Obviously, the shorter the time required to configure a new address, the better because the interruption time will be shorter. Unless additional mechanisms are in place, each time a vehicle changes its IP address, it must restart all existing communications. IP mobility solutions (e.g., Mobile IPv6 [38]) have been designed to enable IP address changing without breaking ongoing communications. In order to provide mobility support to the network formed by the CCU and the AUs, Network Mobility (NEMO) solutions have been defined [28] and further refined for the vehicular scenario [43]. The use of IP mobility protocols solves the problem of communication disruptions due to the change of IP address, but it does not avoid the interruption time caused each time the vehicle changes its IP address. Actually, this time is typically increased when IP mobility solutions are used because of the additional time required to complete the signaling with the mobility

anchor point (e.g., the Home Agent in Mobile IPv6/NEMO). It should also be noted that due to the high mobility of nodes, it is even more important to reduce the overall interruption time (called handover latency) because handovers are more frequent than in other scenarios [44].

There is another metric worth improving: the signaling overhead. To configure a new address in ETSI SLAAC, a vehicle waits to receive an unsolicited RA from the RSU of the new GVL area; therefore, the configuration time is bound to the frequency with which the RSU of the area is sending RAs. In addition, due to the way the ETSI GN and ETSI SLAAC abstract an IPv6 link (mapping it to a GVL area), each RA multicast by an RSU is actually flooded within the area managed by the RSU, which therefore occupies the wireless media for more time than does one single-frame transmission. Therefore, reducing the frequency of unsolicited RAs is even more important than in a non vehicular, single-hop wireless scenario.

Driven by these two goals, reducing the address configuration time and keeping the signaling overhead to a minimum (note that there is a trade-off here), we next propose an overhearing-assisted optimization mechanism for ETSI SLAAC.

2.7 Overhearing-Assisted Optimization for ETSI SLAAC

In this section, we present an extension to the original ETSI SLAAC that provides an important performance improvement and a signaling overhead reduction. An overview of the mechanism is first provided, and then the probability of achieving seamless IP address reconfiguration is analytically modeled.

2.7.1 Solution overview

Our approach aims at reducing the IP address configuration time due to physical node movements that lead to a change of GVL area of the IPv6 link and ultimately of the IPv6 address. While the node is configuring a new IPv6 address (configuration time), the vehicle cannot communicate and has to defer its ongoing communications until a new and valid IP address is configured and becomes usable. Note that even if perfect juxtaposition of GVL areas can be logically obtained in ETSI SLAAC, in a practical scenario this separation does not really exist at the physical layer. Nodes within radio range of the forwarder of a Router Advertisement (RA) located in an adjacent GVL area also receive the Router Advertisement. The original ETSI SLAAC mechanism (explained in Section 2.2) mandates that this Router Advertisement should be filtered out at the ETSI GN layer, to achieve perfect logical GVL area division. However, nodes can benefit from *overhearing* Router Advertisements generated at areas other than the one where the

receiving node is located. In this way, vehicles would be able to learn the IPv6 prefix used in a neighboring GVL area before actually entering it, and they would be able to precompute the IPv6 address and default router configuration that should be used when located in that GVL area (i.e., just after crossing the area border).

When a vehicle overhears Router Advertisements from multiple neighboring areas (e.g., cities, road intersections, etc.), it stores the overheard Router Advertisements for some time. The vehicle learns the GVL area from which the Router Advertisement was sent by using the destination information at the ETSI GN layer, which is set to the GVL area by the RSU when multicasting Router Advertisements. By storing these Router Advertisement-area pairs, the vehicle will be able to configure an IP address without waiting for an unsolicited Router Advertisement if it later enters one of the areas about which it has knowledge.

This overhearing-assisted optimization allows shortening of the average IP address configuration time, in addition to a potential reduction of the required overhead. By enabling this optimization, ETSI SLAAC improves its performance because nodes that successfully overhear a Router Advertisement from a GVL area where they later enter can start using their new addresses without waiting for a new Router Advertisement, and with no extra overhead. The improvement obtained depends on the mobility conditions inside the VANET and does not limit the benefits achieved from using non overlapping areas that are described in [3].

This optimization has a cost in terms of additional complexity at the vehicle level because it must store overhead Router Advertisements and perform the required operations to be able to use them when visiting a GVL area for which the vehicle has a matching overhead Router Advertisement stored. However, it is important to emphasize that the solution does not require any kind of support from the network or other vehicles, and therefore it is fully compatible with legacy ETSI SLAAC systems. Because this optimization is local to a vehicle, vehicles implementing this overhearing-assisted optimization will benefit from reduced IP address configuration time, without impacting in any way the performance or operation of other vehicles.

2.7.2 Overhearing probability

We first focus on analytically assessing how probable it is for a node to overhear an RA generated in a neighboring area. Let *overhearing probability* (P_{oh}) be this probability. In this section, we derive an analytical expression for P_{oh}.

We assume that deploying vehicular networks without dead zones uncovered by an RSU is economically inefficient (at least for non-urban and not densely populated scenarios). As we have discussed in Section 2.1, vehicles form a self-organized multi-hop

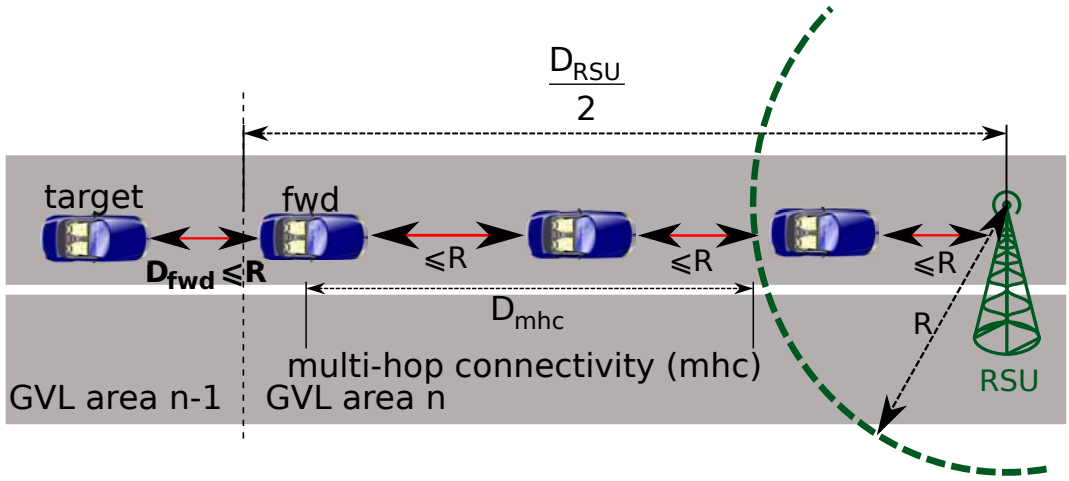


Figure 2.15: Overhearing assisted optimization overview and terminology.

network in the ETSI TC ITS architecture. This multi-hop network is used to forward (at the ETSI GN layer) the RAs sent by an RSU, which flood its associated GVL area and thereby extend the effective coverage area of the RSU.

We use the following terminology throughout the chapter (see Fig. 2.15). A vehicle that is located in the GVL area $n-1$ and must be configured for the area n (i.e., an adjacent area) is called a *target*. A *forwarder* (*fwd*) is a vehicle placed inside the area n that is also within the radio coverage of the *target*. Let D_{RSU} be the distance between two adjacent RSUs, R the wireless communication range, β the vehicular density, and v the speed of the vehicles[13]. A *target* node successfully overhears an RA when the *forwarder* receives an RA and forwards it to the *target* before it enters the GVL area n. In order to be able to receive and forward an RA, the forwarder node should have connectivity with the RSU. As already stated in Section 2.3 we introduce the concept of multi-hop connectivity (*mhc*), a chain of inter-connected vehicles between two nodes.

Given a forwarder node and the RSU, P_{mhc}^{fwd} is the probability of having multi-hop connectivity (that is, of having a chain of inter-connected vehicles between *fwd* and the RSU so that messages can be exchanged between them).

We next model the probability (P_{oh}) of the *target* overhearing an RA originating at the GVL area n. P_{oh}, which can be modeled by splitting the original problem into two complementary sub-problems. Having *mhc* between the RSU and the *forwarder* node is a necessary condition for successfully overhearing an RA. Without *mhc*, no RA *overhearing* (*oh*) is possible; therefore, $oh \subset mhc$. From this condition, it is straightforward to derive that $OH \equiv OH \cap MHC$:

[13]As already done in Section 2.2 We consider the speed of all vehicles to be fixed and constant for the sake of model simplicity. The simulation results we will present later prove that this simplification does not affect the validity of the conclusion of our analysis.

$$P_{oh} = P(OH) = P(OH \cap MHC). \tag{2.14}$$

By applying the conditional probability theorem in Eq. (2.14), we have:

$$P_{oh} = P(OH \cap MHC) = P(OH|MHC)P(MHC) = P_{oh|mhc}\ P_{mhc}^{fwd}, \tag{2.15}$$

where P_{mhc}^{fwd} represents the probability of *mhc* for the *forwarder*. We first focus on $P_{oh|mhc}$.

Let T_{nRA}^{recv} denote the time elapsed from the *target* vehicle being at distance R from an adjacent GVL area border to the time it receives an RA sent by an RSU of that adjacent area. This time can be split in two parts. The first part (T_{fwd}) represents the time elapsed until the *forwarder* vehicle leaves the GVL area $n-1$, enters the next one and becomes ready to forward RAs to the *target*. The second part (T_{RA}^{unsol}) is the time elapsed until an RA from the RSU is received under this *mhc* assumption (see Eq. 2.12). By joining T_{fwd} and T_{RA}^{unsol}, we can express T_{nRA}^{recv} as:

$$T_{nRA}^{recv} = T_{fwd} + T_{RA}^{unsol}. \tag{2.16}$$

Assuming exponentially distributed distances between cars [12], T_{fwd} follows an exponential distribution with parameter β; its probability density function (PDF) is given by:

$$f_{T_{fwd}}(t) = \beta v e^{-\beta v t}, \quad t \geq 0. \tag{2.17}$$

Given that T_{fwd} and T_{RA}^{unsol} are independent, the PDF of T_{nRA}^{recv} is given by:

$$f_{T_{nRA}^{recv}}(t) = \left(f_{T^{fwd}} * f_{T_{RA}^{unsol}}\right)(t) = \begin{cases} \frac{2\left(1-\beta v e^{(-\beta v t)}\right)}{R_M+R_m}, & 0 \leq t \leq R_m, \\ \frac{2\left(\beta v R_M - \beta v t + 1 - e^{-\beta v (t-R_m)} - \beta v (R_M - R_m) e^{-\beta v t}\right)}{\beta v (R_M^2 - R_m^2)}, & R_m < t \leq R_M, \\ \frac{2e^{-\beta v (t-R_M)} - e^{\beta v R_m} - \beta v (R_M - R_m)}{\beta v (R_M^2 - R_m^2)}, & t > R_M. \end{cases} \tag{2.18}$$

Because an RA is overheard only if received by the vehicle before crossing the area border, and it takes R/v seconds for the vehicle to reach the border, the probability $P_{oh|mhc}$ of overhearing an RA is given by:

$$P_{oh|mhc} = \int_0^{\frac{R}{v}} f_{T_{nRA}^{recv}}(t)dt. \tag{2.19}$$

$P_{oh|mhc}$ represents the probability of overhearing an RA, given that there exists multi-hop connectivity between the RSU and the *forwarder* vehicle.

We next model P_{mhc}^{fwd}, which depends on the distance between the RSU and the forwarder node, the radio coverage of the wireless communication technology and the vehicular density.

Given two nodes separated by a distance, D, $P_{mhc}(D)$ is the probability of having multi-hop connectivity between the two nodes. This probability depends on the distance between the two nodes (D), the radio coverage of the wireless communication technology used (R) and the vehicular density (β). In order to have multi-hop connectivity between a forwarder node and the RSU, there should be a chain of connected vehicles (i.e., the distance between two consecutive vehicles must be less than or equal to R) between *fwd* and a vehicle within the direct (single hop) radio coverage of the RSU. If D_{mhc} is the distance between these two nodes (see Fig. 2.15), then the probability of having multi-hop connectivity between the forwarder node and the RSU is given by $P_{mhc}(D_{mhc})$.

We calculate this probability following the same methodology used in Section 2.3, so if we consider the maximum possible value of D_{mhc}, which is given by $\frac{D_{RSU}}{2} - R$, a pessimistic approximation of P_{mhc}^{fwd} (obtained using Eq 2.7) is given by:

$$P_{mhc}^{fwd} = P_{mhc}(D_{mhc}) \geq 1 - F_L\left(\frac{D_{RSU}}{2} - R\right). \tag{2.20}$$

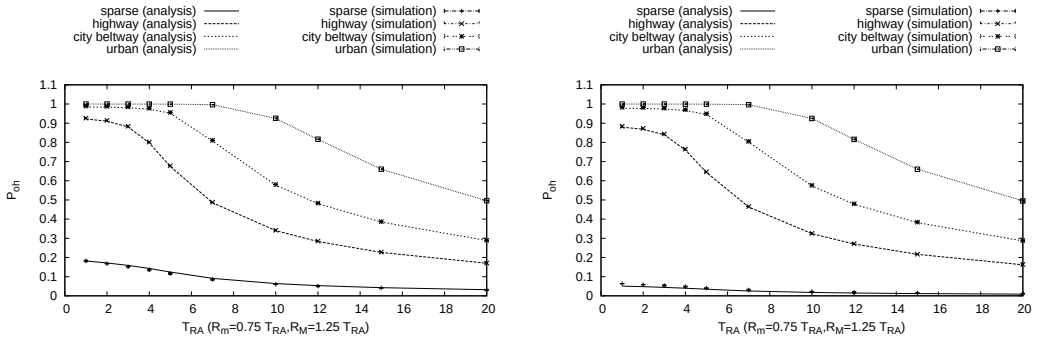
The overhearing probability P_{oh} can then be derived[14] from Eq. (2.15), using Eqs. (2.19) and (2.20):

$$P_{oh} \;=\; P_{oh|mhc}\, P_{mhc}^{fwd} \geq \int_0^{\frac{R}{v}} f_{T_{nRA}^{recv}}(t)dt \;\left[1 - F_L\left(\frac{D_{RSU}}{2} - R\right)\right].$$

We next describe the experiments that we performed to validate our mathematical model. Following the same procedure as in Section 2.2, using a MATLAB-based simulator[15], we conducted a large amount of experiments under different traffic conditions. In Section 2.9, when we validate our analysis for the configuration time, we use a more advanced simulator (OMNeT++), that does include a complete wireless model.

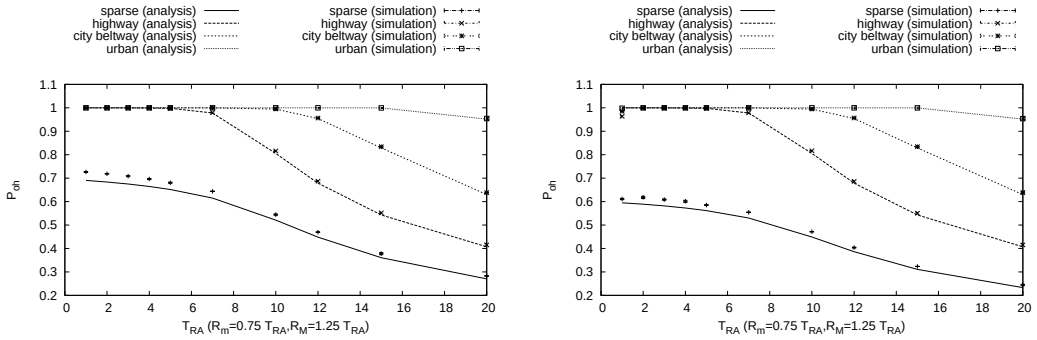
[14]This approximation is also pessimistic because we only consider the first opportunity to receive an unsolicited RA from the RSU, even though there may be more than one before crossing the border.

[15]The code of the simulator is available at
`http://fourier.networks.imdea.org/people/~marco_gramaglia/sims/GeoSAC-sim/`.



(a) $D_{RSU} = 1000m$ (b) $D_{RSU} = 1500m$

Figure 2.16: Overhearing probability, $R = 150m$.



(a) $D_{RSU} = 1500m$ (b) $D_{RSU} = 2000m$

Figure 2.17: Overhearing probability, $R = 300m$.

Because traffic conditions play a critical role in the efficacy of wireless multi-hop communications in a vehicular environment, we studied several configurations in order to validate our model under different conditions. We have selected the following four scenarios, which mostly cover a wide spectrum of the potential traffic scenarios:

- *Urban road:* high vehicular density ($\beta = 80veh/km$) and low speed ($v = 50km/h$).
- *City beltway:* moderate vehicular density ($\beta = 50veh/km$) and moderate speed ($v = 80km/h$).
- *Highway:* low vehicular density ($\beta = 35veh/km$) and high speed ($v = 120km/h$).
- *Sparse:* very low vehicular density ($\beta = 10veh/km$) and moderate speed ($v = 100km/h$). Examples of this scenario are city beltways and highways at night, or secondary roads.

For each of these scenarios, we conducted experiments using two different values of the wireless coverage radio R (150 and 300m) that cover possible IEEE 802.11-based tech-

nologies. Figs. 2.16–2.17 show the analysis and simulation results for P_{oh} versus different average intervals between Router Advertisements for different deployment scenarios (defined by the distance between RSUs, D_{RSU}). Note that different values of D_{RSU} are used depending on the value of R because the coverage radius of the wireless technology has an impact on the deployment. For small values of R, it does not make any sense to deploy RSUs at long distances because the probability of having connectivity is low, and the number of hops is high (which has a negative impact on performance).

It can be noted from these results that our analytical model perfectly matches the results obtained via simulation[16]. From these results, we can also observe that if a short range wireless technology is used ($R = 150m$), high overhearing success probabilities can be achieved only if the RSUs are configured with short inter-RA values (T_{RA}) and only for moderate density scenarios (urban, city beltway and highway). For the case of $R = 300m$ (i.e., for longer range wireless technologies), the probability of overhearing an RA increases (being very close to 100%) without requiring too much resource-consuming T_{RA} configuration settings. These results show that our overhearing-assisted optimization is feasible and can effectively reduce the IP address configuration time in most of the practical scenarios. The results also show that for $R = 300m$ there is a high probability of overhearing an RA even in the sparse scenario. From this result, we can also observe the expected impact that D_{RSU} has on P_{oh}, especially with low vehicular densities.

2.8 Evaluation

In this section, we analyze the performance of our overhearing-assisted optimization, by extending the analysis we have developed in Section 2.7 to obtain an expression for the IP address configuration time when our optimization is enabled. This extension allows us to characterize the gains of our mechanism and compare them to performance when overhearing is not enabled. Finally, because it is likely that a vehicular communications system will make use of an IP mobility solution to transparently keep ongoing IP sessions alive, regardless of the movement of the vehicle (and the subsequent change of IP point of attachment and IP address), we also analyze the impact of the proposed optimization on overall performance when IP mobility is enabled.

2.8.1 IP address configuration time

The most obvious advantage of our overhearing optimization is the reduction of the average IP address configuration time because nodes that successfully overhear a RA

[16]Note that we have also performed simulations under several other traffic conditions, and they supported the accuracy of our analytical model.

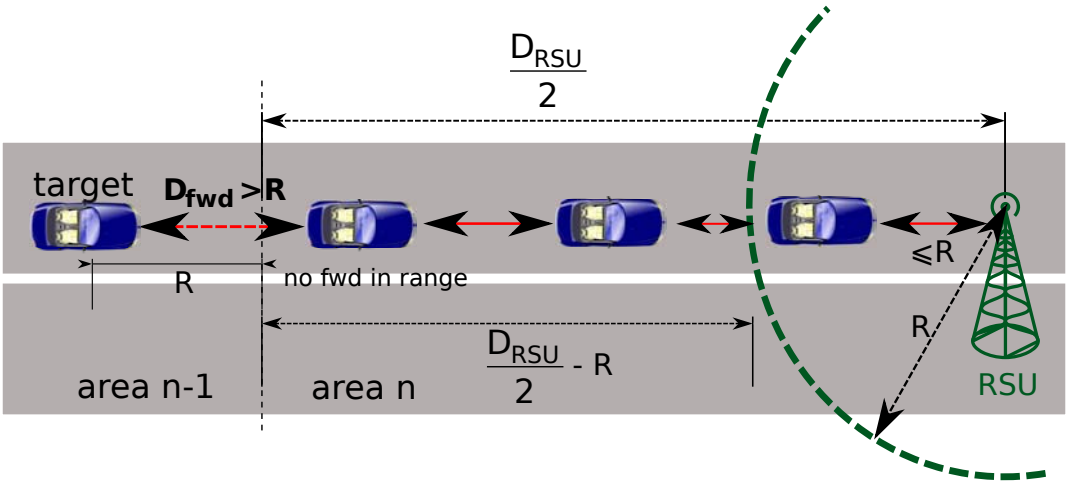


Figure 2.18: No forwarder node within range of the *target* vehicle.

from a neighboring area are able to immediately configure an address if entering into that area afterwards. As already stated in Section 2.2, we define the ETSI SLAAC IP address configuration time (T_{conf}) as the time elapsed from when a vehicle crosses an area border till when it gets a valid IP address that can be used to send and receive packets while located in the new GVL area. If our overhearing optimization is enabled, we call the configuration time T_{conf}^{oh}. A node that overhears an RA from GVL area n while being at area GVL $n-1$ and then enters into GVL area n does not need to wait for any signaling before configuring and starting to use an IP address; therefore, $T_{conf}^{oh} = 0s$ in this case. Depending on the deployment scenario and traffic conditions, it is not always possible for a node to successfully overhear an RA from an area that the node is about to enter. We performed a mathematical analysis that models T_{conf}^{oh}. This model allowed us to evaluate the gains obtained by using our optimization, focusing first on the IP address configuration time reduction compared to the case where plain (i.e., no optimization enabled) ETSI SLAAC is used.

In order to make the analysis easier to follow, we have divided it into different parts, each of them corresponding to a different configuration scenario in which a node might be involved. This approach allows us to derive the average configuration time of ETSI SLAAC when our overhearing optimization is enabled. There are basically four possible situations that have to be considered:

1. There exists a forwarder node (located in the GVL area n) in the wireless coverage of the target node (which is located in the GVL area $n-1$), there is multi-hop connectivity between the forwarder node and the RSU of area n, and the RSU sends an unsolicited RA while the target node has not yet crossed the area border (see Fig. 2.15). In this case (corresponding to overhearing success), the configuration time is 0s.

2. This case is identical to *(a)*, but an RA from GVL area n is not received by the target node while still in GVL area $n-1$. In this case, the configuration time is equal to the time elapsed from the target node crossing the area border until it gets an RA from the RSU. The average of this time ($\bar{T}_{wait\ RA}$) is given by:

$$\bar{T}_{wait\ RA} = \frac{\int_{R/v}^{\infty} t \left(f_{T^{fwd}}^{FWD} * f_{T_{RA}^{unsol}}\right)(t)dt}{\int_{R/v}^{\infty} \left(f_{T^{fwd}}^{FWD} * f_{T_{RA}^{unsol}}\right)(t)dt} - \frac{R}{v}, \tag{2.21}$$

where $f_{T^{fwd}}^{FWD}$ is given by:

$$f_{T^{fwd}}^{FWD}(t) = \begin{cases} f_{T^{fwd}}(t) = \beta v e^{-\beta v t}, & 0 \leq t \leq R/v, \\ 0, & \text{otherwise.} \end{cases} \tag{2.22}$$

3. In this case, there is no forwarder node within radio range of the target node (see Fig. 2.18). The target node has to get direct (i.e., one hop) connectivity from the RSU first and then wait for the next unsolicited RA. The average configuration time ($\bar{T}_{no\ FWD}$) for this case is therefore given by:

$$\bar{T}_{no\ FWD} = \frac{D_{RSU}/2 - R}{v} + \bar{T}_{RA}^{unsol}. \tag{2.23}$$

4. In this case, there exists a forwarder node in wireless coverage of the target node at GVL area $n-1$, but there is no multi-hop connectivity between the forwarder and the RSU of GVL area n (see Fig. 2.19).

   The configuration time is the time required for the forwarder node to get connectivity to the RSU (as the forwarder node moves towards the RSU, the probability of having connectivity with the RSU increases) plus the time until an RA is sent.

   Here, we know that the length of the chain is shorter than $D_{RSU}/2 - R$. By finding the average length of a multi-hop chain, we can obtain the average size of the gap between the last vehicle of the chain and the RSU coverage area border ($\bar{D}_{gap}$ in Fig. 2.19). The final $\bar{T}_{conf}$ for these cases is computed by adding the average delay for getting an unsolicited RA ($\bar{T}_{RA}^{unsol}$).

   We must first calculate the average distance between the *target* and *forwarder*. By definition, the *forwarder* is the farthest vehicle that can relay an RA to the *target* because it is placed at most at R meters away from it. Thus, for a given density and coverage radius, the forwarder is placed at $\bar{D}_{fwd}$. In order to calculate this value, we introduce $\bar{L}_{g_Y(y;k)}$, the average length of a chain composed of a generic set of

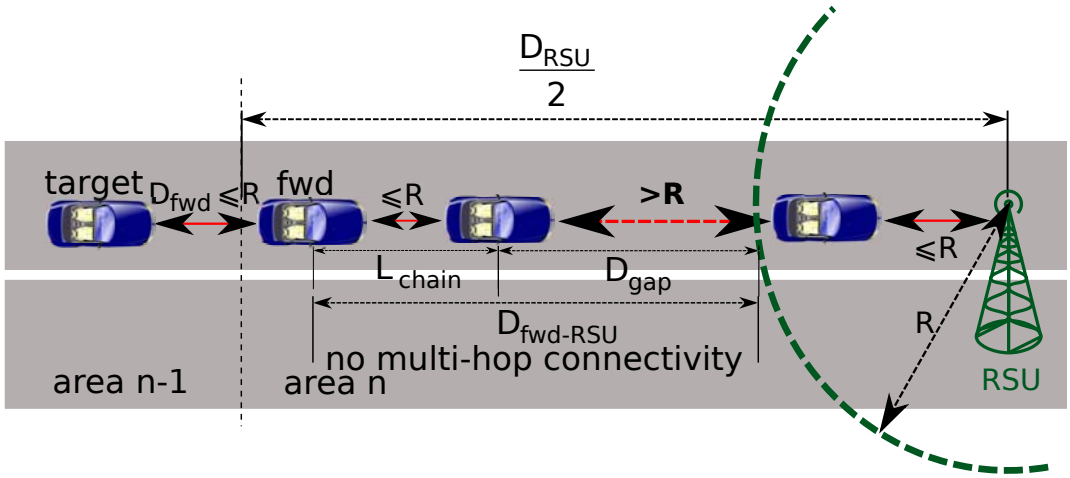


Figure 2.19: No multi-hop connectivity available between the *target* vehicle and the RSU.

vehicles that are exponentially distributed with parameter β and no longer than a maximum value R:

$$\bar{L}_{g_Y(y;k)} = \int_0^R y \; g_Y(y;k)dy. \tag{2.24}$$

The probability of a chain composed by $k+1$ vehicles being shorter than R is $(1-e^{-\beta R})^k \; e^{-\beta R} \; G_Y(R,k)$. From this probability, we can calculate the average distance of the farthest vehicle within R meters from the *target* (i.e., the *forwarder*):

$$\bar{D}_{fwd} = \frac{\sum_0^\infty (1-e^{-\beta R})^k \; e^{-\beta R} \; G_Y(R,k)\bar{L}_{g_Y(y;k)}}{\sum_0^\infty (1-e^{-\beta R})^k \; e^{-\beta R} \; G_Y(R,k)}. \tag{2.25}$$

The average distance between the *forwarder* and the coverage area of the RSU is given by:

$$\bar{D}_{fwd-RSU} = \frac{D_{RSU}}{2} - R - \bar{D}_{fwd}. \tag{2.26}$$

The average length of a chain of vehicles that is shorter than $\bar{D}_{fwd-RSU}$ is:

$$\bar{L}_{chain} = \frac{\sum_0^\infty (1-e^{-\beta R})^k \; e^{-\beta R} \; G_Y(\bar{D}_{fwd-RSU},k) \; \bar{L}_{g_Y(y;k)}}{\sum_0^\infty (1-e^{-\beta R})^k \; e^{-\beta R} \; G_Y(\bar{D}_{fwd-RSU},k)}. \tag{2.27}$$

Therefore, the average gap length is given by:

$$\bar{D}_{gap} = \bar{D}_{fwd-RSU} - \bar{L}_{chain}, \tag{2.28}$$

and from this result, we can calculate the time required to configure an IP address. In this case:

$$\bar{T}_{no\ MHC} = \frac{\bar{D}_{gap}}{v} + \bar{T}_{RA}^{unsol} \tag{2.29}$$

In order to calculate the probability of each of the four identified situations occurring, we performed some probability calculations:

$$\begin{aligned} &P(FWD)\left[P((OH|MHC)|FWD) + P((\overline{OH}|MHC)|FWD)\right] + \\ &P(\overline{FWD})\left[P((OH|MHC)|\overline{FWD}) + P((\overline{OH}|MHC)|\overline{FWD})\right] = 1, \end{aligned} \tag{2.30}$$

where $P(FWD)$ is the probability of a forwarder node existing (i.e., being within R meters from the target node). Because $P((OH|MHC)|\overline{FWD}) = 0$ (it is not possible to have overhearing success if there is no forwarder node) and $P((\overline{OH}|MHC)|\overline{FWD}) = 1$ (if there is no forwarder node, it is impossible to overhear an RA), we can further expand Eq. (2.30) as follows:

$$\begin{aligned} P((OH|MHC) \cap FWD) + P((\overline{OH}|MHC) \cap FWD) + P(\overline{FWD}) = \\ P(OH|MHC) + P((\overline{OH}|MHC) \cap FWD) + P(\overline{FWD}) = \\ P(OH|MHC)P(MHC) + P((\overline{OH}|MHC) \cap FWD)P(MHC) + \\ P(\overline{FWD})P(MHC) + \left[1 - P(\overline{FWD})\right] P(\overline{MHC}) + P(\overline{FWD})P(\overline{MHC}) = \\ P(OH) + P((\overline{OH}|MHC) \cap FWD)P(MHC) + P(\overline{FWD})\left[P(MHC) + P(\overline{MHC})\right] + P(FWD)P(\overline{MHC}) = \\ P(OH) + P((\overline{OH}|MHC) \cap FWD)P(MHC) + P(\overline{FWD}) + P(FWD)P(\overline{MHC}) = 1, \end{aligned} \tag{2.31}$$

in which we have also applied some properties of conditional probabilities. The goal of this analysis was to determine the probabilities of each of the four different situations that we identified previously. Using Eqs. (2.21)–(2.29) and Eq. (2.31), we can obtain an expression for the average IP address configuration time when the overhearing optimization is enabled:

$$\begin{aligned} \bar{T}_{conf}^{oh} = P(OH)\ 0 + P((\overline{OH}|MHC) \cap FWD)P(MHC)\ \bar{T}_{wait\ RA} \\ + P(\overline{FWD})\ \bar{T}_{no\ FWD} + P(FWD)P(\overline{MHC})\ \bar{T}_{no\ MHC}, \end{aligned} \tag{2.32}$$

where:

$$P((\overline{OH}|MHC) \cap FWD) = \int_{R/v}^{\infty} \left(f_{T^{fwd}}^{FWD} * f_{T_{RA}^{unsol}}\right)(t)dt, \tag{2.33}$$

$$P(\overline{FWD}) = 1 - P(FWD) = 1 - e^{-\beta R}, \tag{2.34}$$

and

$$P(MHC) = P_{mhc}^{fwd} = P_{mhc}(D_{mhc}) \geq 1 - F_L\left(\frac{D_{RSU}}{2} - R\right). \tag{2.35}$$

2.9 Performance Analysis and Evaluation

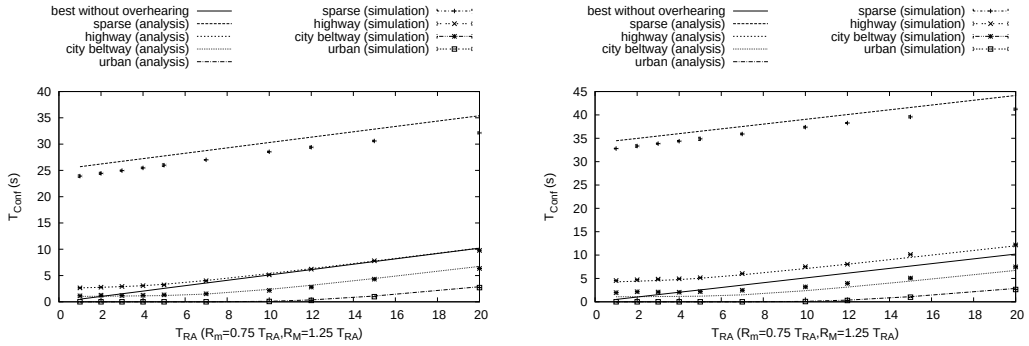
We next validate our mathematical analysis by means of simulation. In order to consider more realistic wireless conditions, we implemented our overhearing optimization for ETSI SLAAC[17] using OMNeT++ and the Mixim framework. Mixim[18] is a framework for a wireless ad hoc network for the OMNeT++ simulator[19]. It provides the 802.11 MAC layer and many physical layer models (including the widely accepted path-loss, shadowing and large and small-scale fading models [40] [41] [42]). The simulation scenario consists of a road segment where vehicles travel within a homogeneous flow. The vehicles' starting positions are generated using an exponential distribution. The speed and density are defined by the type of scenario: urban, city beltway, highway and sparse; so the number of nodes involved in the simulation changes depending on the vehicular density (see Table 2.1). At the end of the road segment, nodes enter a GVL area (D_{RSU} meters long) where an RSU is placed half-way ($D_{RSU}/2$ from the area border). The vehicles are equipped with a standard 802.11g MAC layer, with a bitrate of 6Mb/s. When the simulation starts, vehicles are first excluded from the results' recollection because they were already located inside the GVL area, but they are needed to build the multi-hop chain and to allow the subsequent vehicles to be configured. When a node receives the first Router Advertisement after crossing the area border, its configuration time is recorded. Each simulation is run 20 times using the same topology with a different seed, and for each parameter set, 50 different topologies are generated. The results are averaged on a population of at least $1000 \times nCars$ values, where $nCars$ depends on the chosen vehicular density. Because the road segment length is 15km, in the worst case, this value is approximately 150. The parameters used in the simulations are summarized in Table **??**.

Figures. 2.20–2.21 show the obtained results using the OMNeT++-based simulation. In this case, we used two different values of R: *i)* $R = 225m$, the average coverage value

[17] The code of this simulator is available at
`http://fourier.networks.imdea.org/people/~marco_gramaglia/sims/GeoSAC-sim/`.

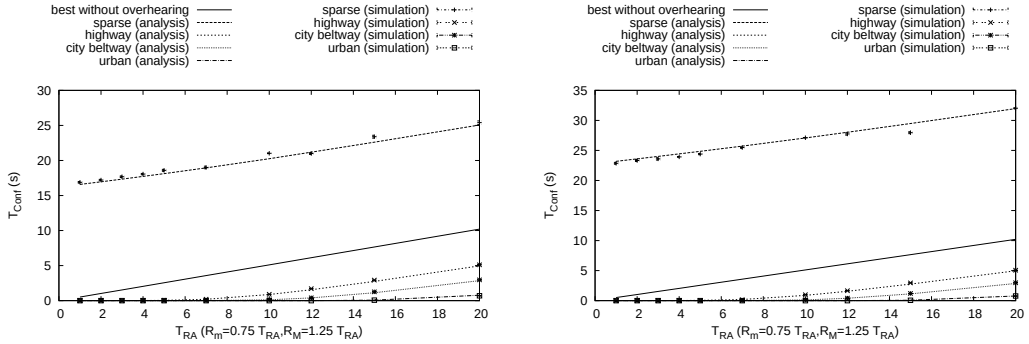
[18] `http://mixim.sourceforge.net/`

[19] `http://www.omnetpp.org/`



(a) $D_{RSU} = 2000m$ (b) $D_{RSU} = 2500m$

Figure 2.20: IP address configuration time (analysis and OMNeT++ simulation results), $R = 150m$.



(a) $D_{RSU} = 2500m$ (b) $D_{RSU} = 3000m$

Figure 2.21: IP address configuration time (analysis and OMNeT++ simulation results), $R = 225m$.

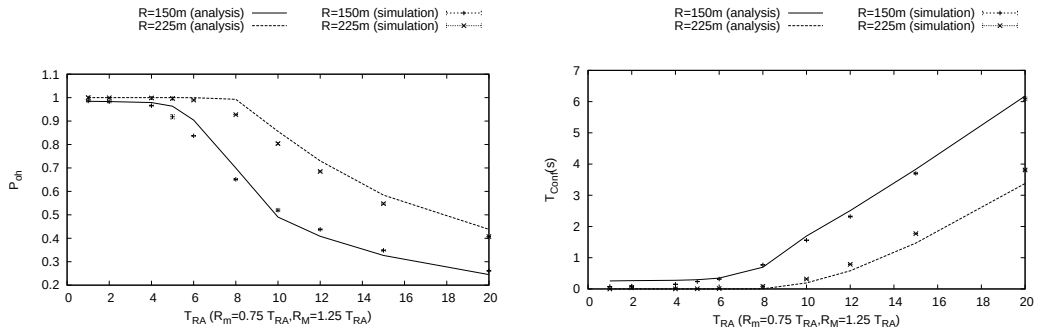
between two wireless nodes in the OMNeT++ simulation, when configured as in our experiments (Figure 2.21), and *b)* $R = 150m$, which is one of the values we used in the previous simulations that helped to better understand the performance of our optimization when the probability of having multi-hop connectivity is lower (Figure 2.20). In addition to the simulation and analytical results, we also depict the best possible value for the IP address configuration time that plain ETSI SLAAC could achieve [3]. This value, which corresponds to the optimistic, non-ideal assumption that there is always multi-hop connectivity between an unconfigured node and the RSU, is equal to $\bar{T}_{RA}^{unsol}$. The simulation results validate our mathematical analysis[20]. They show that for the non-sparse scenarios and values of T_{RA} between 1 and 20 seconds, the average IP address configuration time is always shorter than the best possible value that could be obtained with plain ETSI SLAAC (i.e., without our optimization enabled). In addition, the improvement provided by overhearing optimization is quite large (average configuration time is close to zero for

[20]We also performed a large number of experiments using our MATLAB-based simulator, that also validated our analysis.

several values of T_{RA}). With $R = 150m$, the optimization does not provide any improvement in the sparse scenario. Note that the IP address configuration time displayed in Figures 2.20–2.21 is sometimes larger for the sparse scenario than for plain ETSI SLAAC under the best possible conditions. This is because the value for ETSI SLAAC without overhearing is the one that would be achieved if there is always multi-hop connectivity between the vehicle and the RSU, which is far from true in sparse scenarios. In those scenarios, the IP address configuration time with and without our optimization enabled would be very similar. We also want to mention that the use case scenarios in which it makes sense to deploy an IP multi-hop network to connect vehicles to the Internet ranges from moderate to high vehicular density networks (with proper RSU placement). While providing a more effective support for sparse networks could be possible, that would introduce a lot of complexity and computational costs, while bringing limited benefits, given the low connectivity level that vehicles experience in those scenarios.

The last experiment to validate our analysis and the effectiveness of our overhearing optimization consists in evaluating the configuration time using vehicular traces from a real road in Madrid, Spain. Using the OMNeT++ simulator, we assume the position and speed of vehicles in a real road from traffic traces, evaluate the overhearing probability (Figure 2.22(a)) and measured the ETSI SLAAC configuration time. The traces were taken at the three-lane city beltway M40 in Madrid and accounted for the traffic from 8:30 to 9:00 a.m. (which can be considered as near to rush hour). The total number of samples was 2560. For each sample, we have a time-stamp and vehicle speed. We considered the measurement point to be the border between two GVL areas and assumed that each vehicle maintains the same speed while traversing the area. For our simulation environment, we fixed the distance between two RSUs at 2000m. Figure 2.22(b) shows the results obtained from the simulation and our mathematical analysis. In our mathematical model, we used the vehicular density calculated from the traces ($\beta = 54$ veh/km) and the average speed ($v = 95$km/h). As can be observed from Figure 2.22(b), the results using real vehicular traces confirm our previous findings, showing how overhearing optimization is able to significantly reduce the IP address configuration time.

It is worth highlighting the influence of T_{RA} on the performance gain provided by overhearing optimization. First, with relatively low values of T_{RA} (up to 8 seconds), the average configuration time is generally quite low (zero or close to zero) for all the scenarios. Second, increasing T_{RA} gradually impacts performance by increasing the average configuration time, without abrupt changes. This result basically means that the system can be configured to meet certain performance and signaling overhead requirements by setting up the correct T_{RA} values on the RSUs. Although shorter T_{RA} values provide better performance, there is an additional overhead cost that must be considered. The next section explores in further detail the trade-off between signaling overhead and configuration time.



(a) Overhearing probability

(b) Overhearing-enabled ETSI SLAAC configuration time

Figure 2.22: Analysis and simulation with OMNeT++ and real vehicular traces.

2.9.1 Signaling savings

We have demonstrated in the previous subsection that enabling our overhearing-assisted optimization greatly improves the ETSI SLAAC performance in terms of IP address configuration time. However, our optimization can also, by sending unsolicited RAs less frequently, be used to reduce the signaling required to achieve a certain target configuration time.

From the simulations we have performed, we can obtain what the minimum Router Advertisements frequency required to achieve this target configuration time is, with and without our overhearing optimization enabled. In Table 2.2, we provide some results (for $R = 225m$) to help evaluate the signaling savings that can be obtained. In the urban scenario, for example, a value of $T_{RA} = 15s$ is sufficient to obtain a shorter configuration time than the one obtained by plain ETSI SLAAC with $T_{RA} = 2s$. For the city beltway and highway scenarios, the results are similar. However, the difference in performance decreases as the vehicular density decreases, and the probability of overhearing success decreases. In the highway scenario, for example, T_{RA} can be increased by up to 10 seconds if the goal is to achieve a better performance than with plain ETSI SLAAC and $T_{RA} = 2s$.

2.9.2 Impact on the handover

ETSI SLAAC was designed as a mechanism to enable vehicles in a self-configured VANET to obtain a valid IP address. Getting an IP address is just one of the functionalities needed to connect vehicles to the Internet. As we discussed in Section 1, routing and mobility support are also important components. Routing within a VANET is independent of the IP addressing in the ETSI TC ITS system architecture; therefore, the performance of the IP address auto-configuration protocol does not have an impact on the routing functionality. However, that lack of impact is not the case for IP mobility

Table 2.2: Address configuration time ($R = 225m$).

D_{RSU} (m)	v (km/h)	β (veh/km)	$\bar{T}_{RA}$	$\bar{T}_{conf}$	$\bar{T}^{oh}_{conf}$	Saving
2000	50	80	1 s	0.51 s	0 s	100 %
			2 s	1.02 s	0 s	100 %
			3 s	1.53 s	0 s	100 %
			4 s	2.04 s	0 s	100 %
			5 s	2.55 s	0 s	100 %
			7 s	3.57 s	0 s	100 %
			10 s	5.10 s	0.06 s	98.7 %
			12 s	6.13 s	0.27 s	95.5 %
			15 s	7.65 s	0.97 s	87.26 %
			20 s	10.20 s	2.7 s	73.5 %
2000	80	40	1 s	0.51 s	0 s	100 %
			2 s	1.02 s	0 s	100 %
			3 s	1.53 s	0 s	100 %
			4 s	2.04 s	0 s	100 %
			5 s	2.55 s	0 s	100 %
			7 s	3.57 s	0 s	100 %
			10 s	5.10 s	0.13 s	97.44 %
			12 s	6.13 s	0.39 s	93.53 %
			15 s	7.65 s	1.17 s	84.56 %
			20 s	10.20 s	3.04 s	70.18 %
2000	120	35	1 s	0.51 s	0 s	100 %
			2 s	1.02 s	0 s	100 %
			3 s	1.53 s	0 s	100 %
			4 s	2.04 s	0.02 s	98.76 %
			5 s	2.55 s	0.04 s	98.43 %
			7 s	3.57 s	0.16 s	95.28 %
			10 s	5.10 s	0.93 s	81.60 %
			12 s	6.13 s	1.62 s	73.41 %
			15 s	7.65 s	2.92 s	61.83 %
			20 s	10.20 s	4.87 s	52.24 %

management protocols, in which the speed of IP address acquisition has a direct impact on the overall performance of the mobility solution, and thus on the IP connectivity.

Mobile IPv6 [38] is the standardized solution for providing IP mobility support. The Network Mobility (NEMO) Basic Support protocol [28] is an extension to Mobile IPv6 for enabling the movement of complete networks, instead of just single hosts. Note that it is likely that vehicles will need a network mobility solution because cars are expected to be equipped with many devices with connectivity requirements.

Every time a vehicle changes its IP point of attachment, the mobility protocol signals the movement to a central entity, called the Home Agent (HA), which keeps track of the current location of the mobile node. Every time the vehicle moves, there is an interruption

Table 2.3: Handover delay comparison. City beltway scenario ($R = 225m$).

$RTT(veh, HA)$	T_{RA}	$\bar{T}_{conf}$	$\bar{T}_{conf}^{OH}$	$\bar{T}_{ho}$	$\bar{T}_{ho}^{OH}$	Saving
5.37 ms	1 s	0.51 s	0.00 s	0.52 s	0.02 s	98.96 %
	4 s	2.04 s	0.00 s	2.05 s	0.05 s	99.73 %
	10 s	5.10 s	0.06 s	5.11 s	0.07 s	98.69 %
	20 s	10.20 s	2.70 s	10.21 s	2.71 s	73.48 %
18.32 ms	1 s	0.51 s	0.00 s	0.53 s	0.02 s	96.53 %
	4 s	2.04 s	0.00 s	2.06 s	0.02 s	99.11 %
	10 s	5.10 s	0.06 s	5.12 s	0.08 s	98.44 %
	20 s	10.20 s	2.70 s	10.22 s	2.72 s	73.39 %
138.79 ms	1 s	0.51 s	0.00 s	0.65 s	0.14 s	78.62 %
	4 s	2.04 s	0.00 s	2.18 s	0.14 s	93.63 %
	10 s	5.10 s	0.06 s	5.24 s	0.20 s	96.18 %
	20 s	10.20 s	2.70 s	10.34 s	2.84 s	72.53 %

time (called handover latency) during which the vehicle cannot send or receive packets until all the mobility operations are completed. This handover time can be expressed as:

$$T_{ho} = T_{MD} + T_{conf} + RTT(veh, HA), \tag{2.36}$$

where T_{MD} represents the time required by the vehicle to detect that it has changed its point of attachment, T_{conf} represents the time required to configure a valid IP address at its new location, and $RTT(veh, HA)$ represents the round trip time between the vehicle and the corresponding HA. $T_{MD} = 0s$ because, as with ETSI SLAAC, a change in its point of attachment corresponds to a change of GVL area, which the vehicle can be easily detect by monitoring its GPS coordinates. $RTT(veh, HA)$ depends on the distance between the vehicle and its HA, and it is typically on the order of milliseconds. T_{conf} is the main component in Eq. (2.36). Therefore, reducing the address configuration time has a clear impact on the overall performance.

Table 2.3 shows the average ETSI SLAAC handover delay ($\bar{T}_{ho}$), with and without the overhearing optimization enabled for the city beltway scenario (with $R = 225m$) and the different components used in the calculation of this delay. Because the mobility signaling delay depends on the RTT between the vehicle and its mobility anchoring point (its HA), and because this delay depends on the location of these two entities, three different values of $RTT(veh, HA)$ were used in this analysis, representing "local," "regional" and "continental" delays (we used measurements taken from the PingER – Ping end-to-end reporting – project[21]). For each of these delay values, different Router Advertisement intervals were used. The savings in the overall handover delay achieved by the use of the

[21] `http://www-iepm.slac.stanford.edu/pinger/`

overhearing optimizations are more than 70% in the scenarios analyzed. In addition, the absolute handover latency values that are obtained when the overhearing optimization is enabled are small enough to even enable applications with certain latency constraints to be deployed in a vehicular network.

2.10 Summary

In this chapter, we proposed a model and an enhancement for GeoSAC an address autoconfiguration protocol that extends the standard IPv6 SLAAC to the vehicular environment that has been lately standardized by ETSI as ETSI SLAAC. We first analyzed the effectiveness of multi-hop wireless communications in vehicular environments. Then we focused on characterizing the address autoconfiguration timings achieved using ETSI SLAAC. In order to improve the efficiency of the autoconfiguration algorithm and reduce the configuration times we proposed an optimization based on the overhearing of router advertisements. The results coming from the simulation and the implementation of the algorithm show the correctness of the proposed analytical models and the improvement achieved using our optimization.

Chapter 3

Hybrid position based routing

In the previous chapter we proposed an efficient mechanism to provide addressing autoconfiguration capabilities in a vehicular environment. In this chapter we address the second functionality needed within a VANET: efficient routing. Due to the high degree of mobility of the nodes in a vehicular network, routing protocols proposed for classical multi-hop ad-hoc networks do not provide a performance good enough. In this chapter, we propose a tree-based routing protocol called Tree-based routing for Vehicle-to-Internet Communications (TREBOL). The proposed solution, by exploiting several distinctive characteristics of an Internet vehicular scenario, significantly improves the performance in terms of packet delivery ratio and signalling overhead. Moreover, this approach can be used in conjunction with the addressing scheme proposed in Chapter 2, to furthermore reduce the control overhead introduced and provide the basic functionalities (as seen in Chapter 1) needed to connect a VANET to the Internet.

3.1 State of the art

Vehicular networks exhibit unique properties such as the high dynamics of the nodes (e.g., the link lifetime is subject to vehicles' movements). These particularities make the use of standard MANET routing protocols (either proactive or reactive) not suitable for this kind of environment. The knowledge of participant nodes' position information, typically supplied by a GPS receiver, can be exploited by VANET-specific routing solutions to increase their performance [45]. In particular, the need for tailored routing protocols to VANETs has led to two main families of position-based algorithms [46] according to the type of information they handle: Basic geographic protocols and Information-enriched geographic protocols.

In basic geographic protocols [47, 48], an intermediate node forwards a packet to the direct neighbor which is the *closest* to the geographic position of the destination, operation known as greedy forwarding. So, each node has to be aware of *i)* the position of its direct neighbors, and *ii)* the position of the final destination. To this end, nodes send periodic beacon messages informing neighboring nodes about their identifier, position and other relevant information. However, the selection of a proper beaconing interval becomes really important to find a good trade-off between control overhead and up-to-date neighborhood information. The lower interval, the more up-to-date information is acquired, but at the cost of extra control overhead, interferences and more frequent wireless collisions. As for the position of the final destination, this information is provided by a *location service*. This functionality may be centralized (i.e., nodes update their new locations on a location server) or distributed (e.g., the source node floods a message asking for the position of the destination node), but in any case, the location service is another source of control overhead.

Information-enriched geographic protocols [5], [6], [49] base their operation on the existence of additional (i.e., besides position) information specific to VANET scenarios such as maps, statistics about traffic density on different roads, number of lanes per road, speed limits or information about trajectory estimations. These protocols, instead of following a greedy approach (e.g., choosing as next hop the *closest* neighbor to the destination), can take wiser forwarding decisions (e.g., choosing as next hop the *best* neighbor). At first glance one would expect that the more information is available the better the routing protocol performance is. However, the performance of these protocols depends very much on how accurate this additional information is, since the forwarding decisions that are taken might be erroneous or really far from optimal. Taking into account that the information they are dealing with is, in many cases highly dynamic (e.g., speed or density of cars), there is a non negligible probability that this information is stale or outdated when it is considered for forwarding. On the other hand, keeping this information updated may be costly in terms of control overhead.

Position-based protocols can support information exchange with the Internet, considering that roadside units are just other nodes that participate in the routing protocol, but at the cost of a significant control overhead. However, we argue that vehicle-to-Internet unicast communications exhibit a common set of characteristics that may be exploited by the VANET routing protocol. In particular, not all network nodes behave in the same way: roadside units (RSU) play a critical role, since they operate as relays to the Internet. The required network connectivity graph is anchored at the RSU (i.e., all data traffic traverses the RSU), as opposed to other vehicle scenarios, in which a mesh graph is desired. As a contribution of this thesis, we propose TREBOL, a tree-based routing protocol flexible enough to quickly react to topology changes, which aims at enabling unicast vehicle-to-Internet communications. Besides, forwarding in TREBOL is not based on positions, so neither beacon messages nor location service information is needed, allowing great savings in terms of control overhead.

3.2 TREBOL

In TREBOL, data forwarding decisions are based on IPv6 addresses (i.e., it is a topological routing protocol). Data paths follow a tree built by the TREBOL protocol, which is formed using position information (e.g., vehicles are assumed to have a GPS receiver) to minimize the control overhead load. We describe next how this is achieved. We assume for the time being that nodes are already provided with IPv6 addresses that can be used by the routing protocol (we describe how TREBOL can function also as address autoconfiguration protocol in Subsection 3.2.1).

The main issue is how to build and update the tree in order to tackle the frequent topology changes in VANETs. The upstream tree (i.e., the tree used in the forwarding of data packets from the vehicle to the Internet) is built and updated when each node learns about its parent upon receiving periodical configuration messages (CM) sent by the roadside unit (RSU). It is assumed that each Road Side Unit plays the role of relay (i.e., forwarding traffic from/to the Internet) for the vehicles within a limited geographical area, known as TREBOL area (see Figure 3.1). Thus, configuration messages sent by a Road Side Unit are spread within its TREBOL area. On the other hand, the creation of the downstream tree (i.e., the tree used in the forwarding of data packets from the Internet to the vehicle) follows a reactive approach: each node learns who are its children on a per data packet basis, as part of the forwarding of data packets.

As already mentioned, TREBOL builds and refreshes the upstream tree by using periodical configuration messages (identified by a unique and incremental *sequence number*) which are initially sent by the Road Side Unit and then regenerated and sent by a subset of the VANET nodes. Once a node receives a CM with a newer sequence number, the

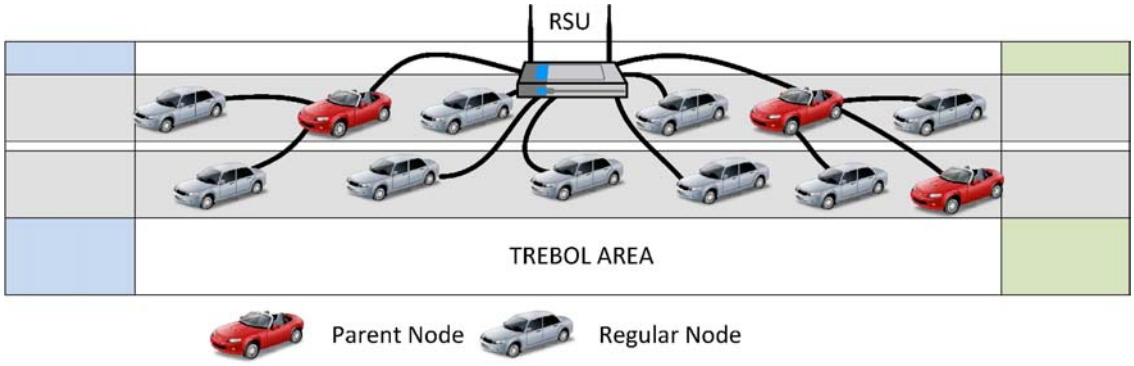


Figure 3.1: TREBOL area.

sender of that CM becomes the parent of the receiving node, and the forwarding state is updated accordingly (i.e., the parent is used as next hop for upstream data traffic towards the Internet). Then, the node regenerates the CM (i.e., updating some fields but keeping the original sequence number) and sets a backoff timer. Only if this backoff timer expires, the node broadcasts this regenerated CM to its neighbors. In the meantime, if the node receives another CM with this same sequence number (i.e., sent by another node with a shorter backoff time), it cancels the sending of the regenerated CM. It is worth mentioning that only if a node sends a regenerated CM, it has the chance to become a *parent* node. *Parent* nodes take the responsibility of forwarding data traffic from/to the Internet from/to its descendants, so a critical issue in TREBOL is to select as parents those nodes that according to their characteristics (e.g., speed, position, etc.) lead to more stable trees. The CMs sent by the Road Side Unit include the following information:

- *areaBoundary*: geographic information describing the TREBOL area. Nodes outside this area receiving a CM discard the message.
- *sendPos*: geographic position of the sender of the CM. It is set initially to the location of the RSU and then overwritten with the position of the last node that regenerated and sent the CM.
- $prefR$: value that represents the preferred distance between consecutive *parents* (i.e., nodes with children). Lower values imply more dense, populated trees, while higher ones imply sparse trees.
- R: value fixing the maximum allowed distance between the receiver and the sender (i.e. the RSU or a potential parent node) of the CM. If the sender is farther away from the receiver node than R (i.e., *sendPos* field), then the CM is discarded. In this way, R serves as a virtual wireless coverage radius.
- $prefS$: value that represents the preferred speed of nodes sending regenerated CMs (i.e., potential *parent* nodes). It is set by the RSU. This value is used to preserve

the stability of the tree selecting as *parent* nodes those that travel at similar speeds (closer to $prefS$).

- $maxSpeedDiff$: nodes whose speed differs more than this value from $prefS$ will be prevented from sending regenerated CMs (i.e., becoming *parent* nodes).
- D_{pos} and D_{speed}: these two values set the maximum value for the backoff timer. The higher these values are, the more time is required to build the tree. On the other hand, too short values might cause many wireless collisions.

Selecting the potential *parent* nodes is a completely distributed process based on a backoff timer:

$$T_{backoff} = \frac{|(|(pos - sendPos)| - prefR)|}{R} \times D_{pos} + \frac{|speed - prefS|}{maxSpeedDiff} \times D_{speed} \quad (3.1)$$

where *pos* is the node position and *speed* is the node speed. A node that is located at a distance $prefR$ from the sender of the CM, and that travels at a speed of $PrefSpeed$ would immediately send the regenerated CM ($T_{backoff} = 0\ s$). After waiting $T_{backoff}$ seconds, the node sends the regenerated CM (updates the *sendPos* field) only if it has not received another CM with the same sequence number from one of its neighbors before. In this way, the shorter the $T_{backoff}$ of a node is, the more likely the node sends a regenerated CM becoming a potential *parent* (i.e., assuming the responsibility of having children and forwarding their data traffic).

On the other hand, the TREBOL downstream tree (i.e., the tree followed to deliver data traffic from the Internet to the vehicle) is built and refreshed on a per data packet basis as part of the data packets forwarding process. A node will be aware of the identity (i.e., the IPv6 address) of its descendants (i.e., downstream nodes in the tree) when it receives data traffic addressed to the Internet from one of its children (i.e., the child has selected the node as next hop for traffic towards the Internet). Thus, upon receiving a data packet to the Internet, the node learns the identity of the descendant (i.e., the source address of the data packet) and updates the corresponding forwarding state information (i.e., the child which forwarded this data packet becomes the next hop for downstream data traffic towards the descendant).

3.2.1 Address Autoconfiguration Support

CM messages are received by all the nodes within a TREBOL area. So far we have assumed that VANET nodes are already provided with an IP address that can then be

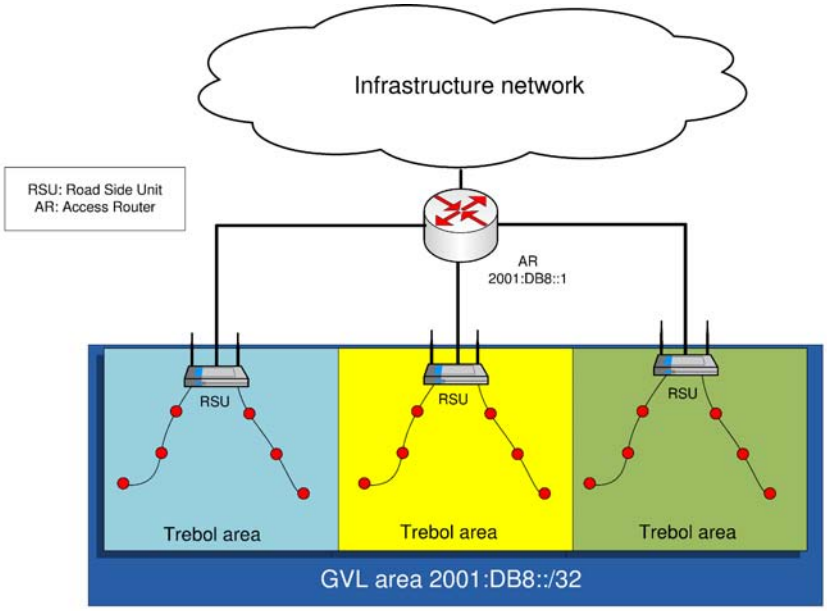


Figure 3.2: Example of TREBOL deployment hierarchy.

used by the TREBOL routing mechanism as identifier in the forwarding process. The same CM messages could also be used to convey prefix information, allowing nodes to autoconfigure IP addresses in a way similar to the standard IPv6 SLAAC [24]. In fact, the most straightforward approach is to slightly modify the IPv6 SLAAC in a way similar to the one proposed in Chapter 2 to reach an integration with TREBOL.

The RSU sends standard Router Advertisements (RAs) messages, containing the prefix(es) allocated to the TREBOL area, and have the on-link flag (L) unset [23]. The two minor modifications that TREBOL introduces consist of: *i)* RAs are regenerated by each *parent* node, keeping the same prefix, and *ii)* RAs are used by all VANET nodes (including *parent* nodes, which are also routers) to autoconfigure an address from the prefix. These RAs are extended with additional options to carry the fields defined in the CMs (needed by TREBOL routing). In order to avoid unnecessary control overhead, Duplicate Address Detection (DAD) is disabled, since we can safely assume that in a vehicular environment there exist unique identifiers that can be used to generate IPv6 addresses.

The main advantage of using this autoconfiguration mechanisms is that it reduces the overall control overhead required by combining routing and address autoconfiguration functions using a single set of signaling messages. Note that this address autoconfiguration mechanism feature could be disabled if required, since TREBOL can also work with different IP address autoconfiguration solutions.

From a deployment perspective, a goal is to configure TREBOL so it provides routing

trees as much stable as possible, without imposing too high performance penalties. Out of the parameters that can be configured, R is determined by the chosen wireless technology, and $prefR$ can be expressed as a fraction of it. When selecting $prefR$, there is a tradeoff that needs to be considered: higher $prefR$ values lead to shorter, but less reliable/stable routes, as more nodes located at the border of the coverage would be selected as *parent* nodes. On the other hand, shorter $prefR$ values lead to more stable, but longer routes. $prefS$ can be fixed by taking the speed limit in the zone or by the RSU taking the average speed (by sampling the vehicles' speed in real time). $maxSpeedDiff$ can be expressed as a fraction of $prefS$.

There is a wide range of deployment scenarios where TREBOL might operate. These scenarios are mostly defined by the size of the TREBOL area, which is conditioned by different aspects, such as performance, vehicular density, cost considerations, etc. In large TREBOL areas (i.e. one single RSU provides service to a geographical area reasonably large), associating an IPv6 prefix to a TREBOL area does not introduce any issues. However, in small TREBOL areas, it is more convenient to associate the same IPv6 prefix to several adjacent TREBOL areas, avoiding the cost imposed by frequent IP address changes.

TREBOL easily supports a flexible association of IP prefixes to multiple TREBOL areas by introducing a simple hierarchy, with the possibility of having several RSU connected to a single Access Router (AR) on the infrastructure (see Figure 3.2.1). The AR plays the role of the *parent* of the RSUs (this is statically configured, without making use of the backoff timer).

3.3 Simulation results

In order to evaluate the performance of the TREBOL routing protocol, we conduct simulations based on real vehicular traces. We compare the performance obtained with TREBOL with a pure geographic based routing protocol: the Greedy Perimeter Stateless Routing for Wireless Networks (GPSR) [47]. Additionally, we also compare some merit figures with the ones that would be obtained with an "optimal" (ideal) routing protocol, in which each node knows the best route to/from the Internet at any time. In this experimental evaluation we focus on the following performance metrics: the packet delivery ratio, the number of hops to reach the RSU and the control overhead.

3.3.1 Simulation environment

We run a set of trace-driven simulations with input data coming from real traffic measurements taken from the orbital highway M-40, as already seen in Chapter 2. In this

Simulation framework	OMNeT++ and MIXIM
Wireless Device	802.11g @ 6Mb/s
Channel Model	Pathloss with channel fading
Coverage radius	225m
Distance between RSUs [m]	1000, 1500, 2000, 3000
Data traffic	ICMP Echo Request / Reply (packet size: 1KB)

Table 3.1: Simulation settings.

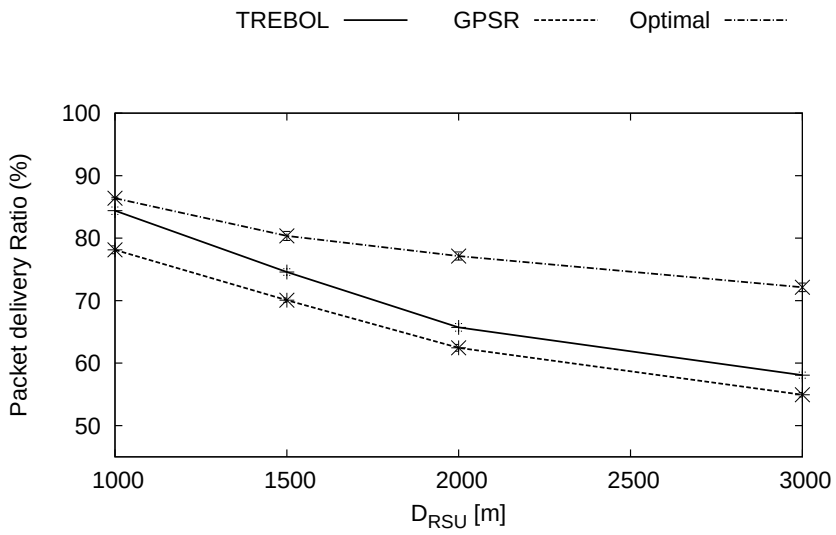


Figure 3.3: The packet delivery ratio with 95% confidence intervals

road, vehicles can span over three lanes (with an average speed of 90 km/h and a density of around 50 veh/km). Simulation settings are summarized in Table 3.1.

TREBOL and GPSR are configured to allow a fair comparison between them. For GPSR, the average time between beacons messages is set to 1 sec (uniformly distributed): $f_{GPSR} = 1msg/sec$. For TREBOL, the time between CMs is set to 1 sec as well (uniformly distributed): $f_{TREBOL} = 1msg/sec$. $prefR$ parameter is set to 180m and the $prefS$ is configured to be equal to the scenario's average speed.

3.3.2 Results analysis

Our goal is to compare TREBOL and GPSR in relation to the following three parameters: packet delivery ratio, average number of hops and signaling load.

Figure 3.3 shows the packet delivery ratio obtained in the simulations for TREBOL, GPSR and also the "optimal" value that could be achieved by an *ideal* routing protocol, under different deployment scenarios – characterized by the distance between RSUs

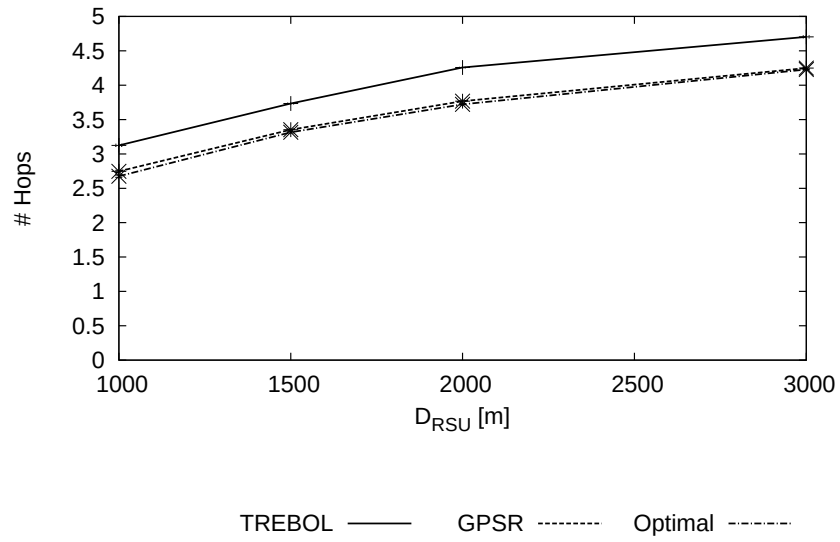


Figure 3.4: The average number of hops with 95% confidence intervals

(D_{RSU}). The "optimal" protocol always finds the best path if it exists, so when D_{RSU} increases the probability of having a gap in the multihop path is higher. As it can be observed, TREBOL provides a higher delivery ratio than GPSR.

The second metric we are interested in analyzing is the average length (i.e. number of hops) of the routes computed by TREBOL. Figure 3.4 shows the obtained results, including also GPSR and the "optimal" routing protocol. As expected, TREBOL uses slightly longer routes, as it tries to come up with routes composed of *parent* nodes that are separated by $prefR$ meters. The average route length achieved by GPSR is very close to the ideal one as GPSR tries to use the shorter possible route, by making use of the farthest forwarding node in the direction towards the destination. This however has an impact on the resulting packet delivery ratio, as the next-hop selected as best by GPSR might become unreachable due to nodes' movement, and this is not detected until the next beaconing period (i.e., until GPSR finds another best next-hop, data packets are lost).

The last important metric we analyze is the signaling load. We define the Relative Routing Load (RRL) as the ratio of the total number of control messages generated by routing protocol X within the routing domain in a given amount of time, compared with the number of messages generated by routing protocol Y:

$$RRL_Y^X = \frac{\text{\# signaling messages sent by X}}{\text{\# signaling messages sent by Y}}.$$

Since the signaling overhead is constant and independent of the data traffic generation

D_{RSU} (m)	RRL
1000	6.68
1500	6.88
2000	7.03
3000	7.16

Table 3.2: Relative routing load of GPSR compared to TREBOL

rate for both GPSR and TREBOL, we can evaluate the advantage provided by TREBOL in terms of control messages savings by looking at RRL^{GPSR}_{TREBOL}:

$$RRL^{GPSR}_{TREBOL} =\leq \frac{\beta D_{RSU} f_{GPSR}}{(D_{RSU}/prefR + 2) f_{TREBOL}} \leq \beta prefR = \frac{prefR}{\gamma}, \qquad (3.2)$$

where β is the vehicular density and γ is the average inter-vehicular distance in the area (i.e. the distance between two consecutive vehicles). With TREBOL on average only one node every $prefR$ meters has to regenerate and send a CM, while with GPSR every node has to perform beaconing.

Using the average inter vehicular distance obtained from the traces ($\gamma = 18.55m$) the calculated RRL^{GPSR}_{TREBOL} is 9.70. We have also performed simulations, measuring RRL^{GPSR}_{TREBOL} (see Table 3.2). The results are coherent with the analytical formulation in Eq. (3.2) as the calculated value is a limit superior. As observed, TREBOL provides an important signaling overhead saving due to the fact that in GPSR every node has to periodically send beacons – in order to keep its position updated into the other nodes' neighbor tables – while in TREBOL only *parent* nodes send signaling messages (and on average there is only one *parent* node every $prefR$ meters).

To sum up, the performance evaluation results show that TREBOL provides a better performance than GPSR – being this performance similar to the one achieved by the "optimal" one in terms of packet delivery ratio and average route length – while outperforming GPSR in terms of control overhead.

3.4 Implementation

The number and high mobility of nodes (i.e., vehicles) and lack of connectivity due to sparse traffic (i.e. low vehicle density) are some of the challenges that VANETs have to face to, in comparison to more traditional mobile ad hoc networks (MANETs). These problems, together with other low-layer related issues (e.g., severe wireless conditions at high speed and obstacles), make it very difficult to conduct realistic experiments. Some testbeds for VANET applications have been deployed so far, among them we highlight the following: Cartel and Cabernet projects at MIT [50, 51], Dome and DieselNet at Amherst [52], VanLan by Microsoft Research [53] and C-VeT at UCLA [54].

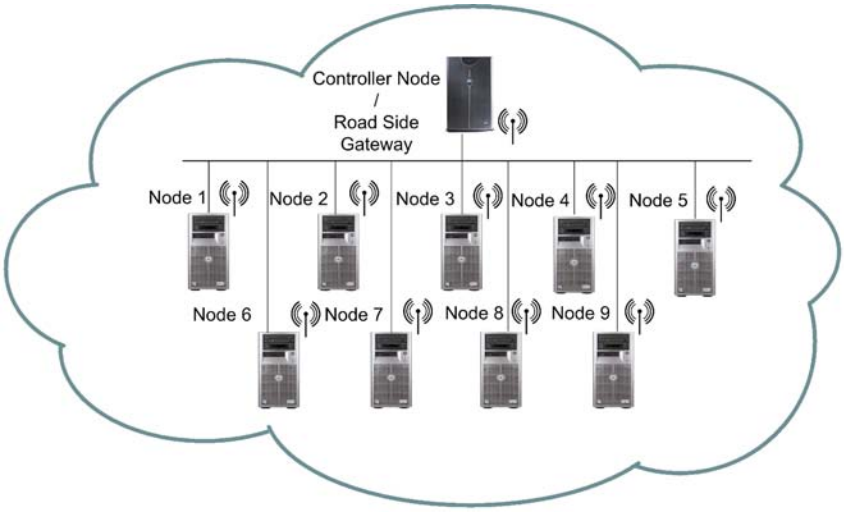


Figure 3.5: Deployed testbed.

Our long-term goal is to develop a low-cost, flexible VANET emulation platform using COTS wireless devices to get more insight of some aspects of VANET protocols that cannot be investigated using a simulator (tool already to evaluate TREBOL in Section 3.3), as for example implementation complexity or behaviour of the protocol with real data traffic. This kind of testbed does not aim at looking for a realistic emulation of the wireless medium, but we rather focus on how the protocol reacts using a multi-hop connectivity map based on nodes position and movement. Studying the behaviour of a VANET protocol using an emulated environment fed by real traffic traces and realistic mobility patterns can help to analyze how the protocol operates under conditions that cannot be easily reproduced in a simulator. We already used COTS devices, the well-known wireless SOHO router Linksys WRT54GL [1], during the evaluation of the protocols detailed in Chapter 2, but in this section we report on first validation experiences using a smaller PC-based testbed consisting in 10 PCs running a Linux system and TREBOL as VANET protocol to be analyzed. The installed OS distribution is Ubuntu 10.10 running a 2.6.35 kernel. Each node is equipped with an Atheros AR5001X wireless card managed by the `ath5k` driver. We are currently working on the migration of this PC-based testbed to the Linksys-based one.

Figure 3.5 depicts the architecture of the testbed deployed in a laboratory of the Department of Telematics Engineering of the Universidad Carlos III de Madrid. All nodes in the testbed are configured in `ad-hoc` mode, belong to the same IBSS, and therefore are configured on the same IEEE 802.11a channel (i.e., 140). Nodes are placed in a reduced space, and consequently can all directly communicate with each other (as they are all within 1-hop radio coverage), thus creating a full-mesh topology. Note that an Ethernet

[1] `http://www.linksysbycisco.com/EU/en/products/WRT54GL`

network is used for controlling and result-gathering purposes.

In order to emulate a dynamic multi-hop connectivity map of the nodes traveling in a road, it is necessary to know the complete route of each vehicle in the considered stretch, a feature provided by the SUMO [2] microscopic traffic simulator. SUMO supports many mechanisms for providing the input traffic rate of the system, but our choice was to use real vehicular traces kindly supplied by the Madrid city council. The measurements, collected at a fixed observation point along the M-30[3] orbital motorway, provide a time mark and the sensed speed for each vehicle that goes through the checkpoint. Feeding the simulator with real traces which have a resolution of 0.1 seconds gives a good estimation of the nodes position at any time. The final output provided by SUMO is a trace file for each vehicle providing the vehicle's position and speed at every time step. Using these traces we can calculate the connectivity map for each node at any time. For this evaluation we used the unit disk coverage rule (with the parameter coverage radius, R) but more complex rules can be used, taking into account the vehicles relative speed or the presence of obstacles. The outcome of this procedure is to obtain, for each node, the connectivity map at any moment.

In our testbed all nodes are within 1-hop direct radio coverage. To emulate a multi-hop connectivity environment we artificially inhibit the wireless connections using a software module. We implemented a library that, using the `ip6tables` Kernel API [4], can emulate a dynamically changing multi-hop wireless connectivity graph. As each vehicle is bound to a single machine the connectivity mapping can be represented using the wireless card MAC addresses. Hence, at each time step, the firewall rules are updated allowing traffic coming only from the *neighbor* wireless cards and, thus, creating an emulated virtual topology over the full-mesh real one.

This approach requires time synchronization among nodes, a task that we accomplished running a pacemaker module in the controller node. The synchronization is kept by the reception of broadcast time-step messages on the Ethernet control network. By merging the time-step information with the generated vehicle trace, each node can create a snapshot of its current connectivity map. The TREBOL software client is in charge of processing the time-step messages and updating position, speed and neighbors set (using the aforementioned library) for each node. The positioning information is also used by TREBOL in order to calculate the backoff timer while processing a new Configuration Message (CM), embedded into a Router Advertisement (RA) as explained in Chapter 2. TREBOL has to change the forwarding table in two situations: during the tree-refreshing phases and when the node is forwarding data on behalf of a child node. In the first case, the RA source address (i.e., the parent node address) is taken as the default gateway to the

[2] `http://sumo.sourceforge.net/`
[3] `http://en.wikipedia.org/wiki/M30_motorway`
[4] `http://www.netfilter.org/projects/iptables/`

Internet. In the latter case TREBOL, using the `libpcap`[5] API, gets the source address of the data packets it is relaying and updates the routing table accordingly. `Netlink` sockets [55] are used for both of the tasks. Finally, the software keeps track of all the routes added for the downstream tree and periodically cleans the forwarding table, removing all the unused entries.

3.5 Implementation results

In Section 3.3, we focused on comparing TREBOL with a geographic based routing protocol using a simulator. In this one, we aim at evaluating the behavior of TREBOL in a more realistic environment, looking at how the algorithm reacts to possible wireless malfunctions and how real data traffic requirements are met by the emulated moving network. We have selected three metrics: *i)* the parent nodes placement, *ii)* the total tree construction time, and, *iii)* the TCP throughput.

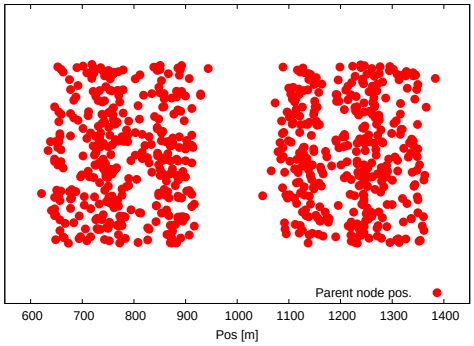
$prefR$	$\bar{T}_{build}$ [s]	Min [s]	Max [s]
180	0.087	0.022	0.214
150	0.088	0.025	0.21
120	0.09	0.03	0.215

Table 3.3: Tree construction times

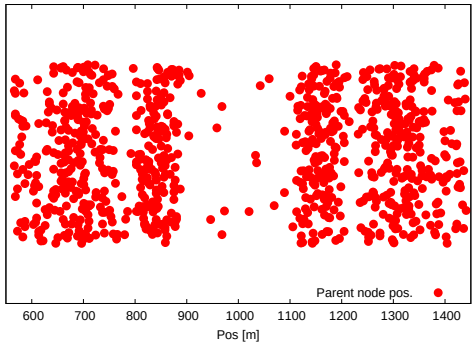
Regarding the first of these metrics, analysis of the parent locations selected by TREBOL, this metric can provide insights on how the protocol reacts to configuration parameters changes. Moreover, it can also prove the resilience of the algorithm to losses of Router Advertisements. We set up a emulation scenario representing a stretch of road 2Km long, with an RSU placed half way, and a total of 9 cars entering the road (at position 0) with times and speeds reflecting the actual ones in a real scenario as explained in Section 3.4.

The average distance between the first and the last vehicle of the queue is around 500m. The test is run 40 times leading to about 240 tree-refreshing phase (the time between consecutive RAs, T_{RA}, is uniformly distributed between 1.5s and 2.5s). Results are shown in Figure 3.6. In the sub-figures we can see the effect of varying one of the TREBOL configuration parameters, $prefR$ (which influences on the desired distance between consecutive parents): 120m for Figure 3.6(a), 150m for Figure 3.6(b) and 180m for Figure 3.6(c) keeping the coverage radius R at 200m. We can see that the protocol behaves as expected: using a $prefR$ value close to the maximum coverage radius forces parent nodes to be more separated between them, while shorter values make the parents topology much denser.

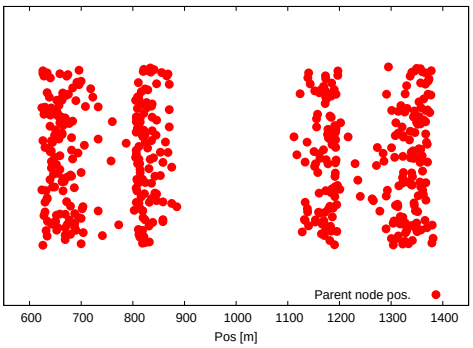
[5] `http://www.tcpdump.org/`



(a) $prefR = 120m$



(b) $prefR = 150m$

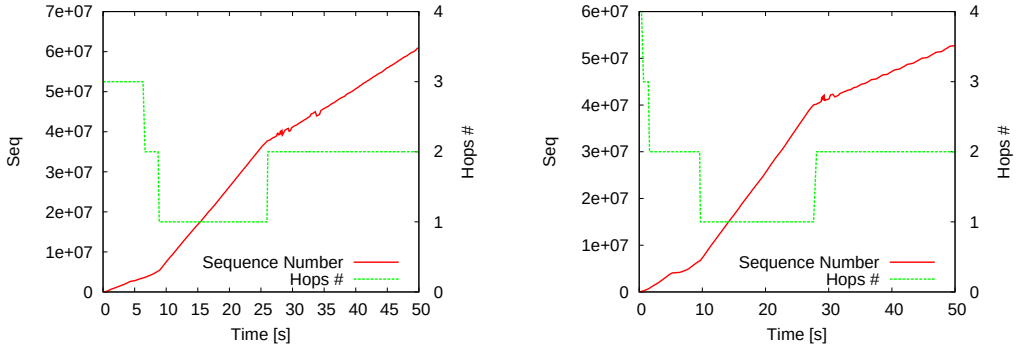


(c) $prefR = 180m$

Figure 3.6: Node Placement.

Another important aspect is the time required to build the tree. This time depends on the real topology (well placed nodes have shorter backoff times) and on the configuration parameters. The number of chosen parents (value influenced by $prefR$) might have an impact on this time: the higher the number of the hops, the higher the tree construction time. Choosing too short values for the parameters that have an impact on the backoff timer might cause problems for parent selection, as retransmission attempts would be scheduled too close. On the other hand higher values will increase the tree construction time, worsening the effect of asymmetrical paths. Notice that during tree-refreshing phase the downstream and upstream tree may not match exactly for short periods of time, which may cause asymmetrical paths.

Results (with the same setup as the previous test) are shown in Table 3.5. With this setup, we fixed $D_{pos} = 0.04s$ and $D_{speed} = 0.01s$. The tree construction time (T_{build}), covering 2 or 3 hops (according to Figure 3.6), is on average around $0.9s$. This value includes the backoff procedure, the wireless medium access time and the packet processing time for each hop. Also the minimum and the maximum $\bar{T}_{build}$ does not seem to be significantly affected by $prefR$, but only by D_{pos} and D_{speed}.
We also wanted to evaluate how a network running TREBOL performs under real data



(a) TCP Throughput run #1 (b) TCP Throughput run #2

Figure 3.7: TCP Throughput and the hop distance from the RSU

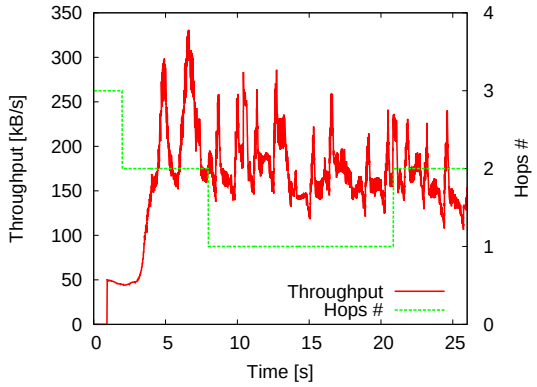


Figure 3.8: UDP Throughput and the hop distance from the RSU

traffic conditions . The first test we run consists in streaming a video [6] from a server at the infrastructure to a vehicle. For this purpose we used RTP/RTCP [56] for delivering the multimedia flow. The request for the video flow is done using the RTSP [57] protocol.

Figure 3.8 shows that the throughput consumed by the video is not influenced by the mechanism used by TREBOL to update the routes. Although the bitrate value varies over the time (the video is encoded using Ogg/Vorbis [58, 59], a VBR codec), the average value is kept stable with the number of hops.

Finally, we evaluated the performance of a TCP connection (namely a HTTP session) to check the effect that asymmetrical paths may have on the offered throughput. As already mentioned, the TREBOL tree refresh process can lead to asymmetric paths, which are known to affect TCP performance [60]. Although available rate and RTT do not change dramatically in our testbed, we wanted to make sure that the TREBOL tree refresh process would not affect the TCP performance heavily. Our test was done using a node initially placed 400m far from the RSU. Once it is configured, the node starts

[6] `http://www.sintel.org/`

downloading a huge text file stored in a Web Server running in the RSU, while the node keeps traveling along the road, first getting closer to the RSU (in terms of distance and hence of hop numbers) and then moving away.

Results in Figure 3.7 show that the tree refresh process only affects the TCP connection while moving away from the RSU (i.e., when increasing the number of hops). There, due to the increasing RTT value, the throughput oscillates for short periods. However, as shown in Figure 3.7, TCP Cubic [61] (the used TCP flavor, it comes by default in Linux kernels since the version 2.6.19) can easily manage this situation.

3.6 Summary

In this chapter, we proposed TREBOL, a tree based routing protocol for vehicle-to-Internet (V2I) connectivity in VANETs. TREBOL benefits from the inherent tree-shaped nature of V2I traffic to reduce the signalling overhead while dealing efficiently with the vehicular dynamics. Results from simulations show that our proposal outperforms a greedy-geographic routing algorithm in terms of packet delivery ratio and introduces less overhead. Finally we showed the feasibility of the implementation of TREBOL using standard Linux PCs, proving how it can efficiently deal with real data streams.

Chapter 4

Seamless Internet Cellular and Opportunistic WLAN Vehicular Connectivity

Providing Internet connectivity anywhere, anytime and in the best possible conditions (both from the user and provider point of view) is no longer an academic exercise without no strong real demand to answer to, but a critical research and engineering problem. It involves not only academia, but also companies and governments, and with quite a big economical footprint.

Nowadays users may enjoy Internet connectivity from a quite broad number of scenarios thanks to the popularity of WLAN based access, e.g., at work, home, school, airports, coffee shops, etc. Moreover, in the recent years, cellular based connectivity (e.g., 3G) has also become quite popular and affordable, being available almost anywhere within populated areas. Within this new picture of different and cheap connectivity options available, the list of devices from our daily life that are connected to the Internet is no longer

restricted to our PC and laptop, being now very common to find that it also includes phones, TVs, gaming consoles, video players, hard drives, or even our home appliances.

This chapter focuses on one particular scenario where Internet connectivity has not yet been fully integrated and exploited: vehicular environments. So far most efforts have been oriented to safety services and traffic efficiency, while Internet communications have been considered to be of a much lower priority. This situation is reflected on the work performed by the main standardization bodies working on vehicular networks. Motivated by the goal of reducing the number of accidents and their consequences these bodies have specified communication architectures that will allow cars to cooperatively exchange safety critical information among them. The ETSI has just recently finalized the standardization of the mechanisms [30] required to integrate IPv6 in the harmonized communication system for European ITS [27]. Further details about some of these mechanisms can be found in Chapter 2.

4.1 State of the art

Vehicular communication scenarios can be divided in two main categories: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). V2V communications are mainly used for safety applications, while V2I ones are mostly used for traffic efficiency and Internet connectivity. We identify two main approaches that can be followed to enable V2I communications:

- *Use of 3G.* Cellular technologies such as 3G or the forthcoming enhanced LTE technology allow users to obtain relatively high bandwidth from mobile platforms. Using 3G to connect vehicles is the simplest approach from a deployment and solution complexity viewpoint, as it does not require any additional protocol development and reuses existing operators' communications infrastructure.

- *Use of VANETs.* The use of WLAN-based VANETs is a widely considered approach to provide connectivity in vehicular environments in a cheap and scalable way.

Existing 3G infrastructures are already suffering from the high demand of subscribers [62, 63]. A wide adoption of 3G as default access technology for the vehicular scenario would increase even more the problems faced by mobile operators. New communication paradigms like Internet of things or cloud computing will lead to a growth of low-consuming and network-capable devices, so it is expected that such devices will become available also in the automotive world. Moreover, national projects like CoCar[1] [64]

[1] `http://www.aktiv-online.org/english/aktiv-cocar.html`

or research studies like [65] envision the use of the cellular network also for V2V communications. Using the 3G network to provide connectivity for those purposes would worsen the problem that operators are trying to solve. Operators are looking for offloading techniques that would allow their 3G networks to relieve by selectively moving some traffic to WLAN access networks when available, like in this case [66].

On the other hand, the use of multi-hop networks based on short range wireless technologies (such as WLAN) in an environment as mobile and dynamic as the vehicular one, also poses significant challenges, which have been extensively analyzed by the research community in the last years [67, 68]. Besides, connectivity in VANETs depends on the traffic density (as already seen in Chapter 2 or in [69]) and, therefore, it is not granted in all cases (e.g., in sparse scenarios). The use of single-hop WLAN access poses less challenges, although it presents the obstacle of the deployment effort required to equip road segments with access points connected to a network infrastructure. The use of multi-hop networks helps considerably in reducing this difficulty, as the density of access points needed is reduced, and its feasibility has been analyzed and in several recent works [70,71].

In this chapter we present SILVIO, our solution to efficiently provide vehicles with Internet connectivity. An overview of the addressed scenario and the basics of the protocol operation are first presented, before describing the details of the main components of the solution.

4.2 Background

SILVIO aims at providing Internet connectivity in a vehicular scenario like the one shown in Fig. 4.1, in which most vehicles contain several devices demanding connectivity. We argue that vehicles need to be equipped with two different wireless communications technologies: *i)* 3G, to provide close-to always available connectivity, and *ii)* WLAN, to provide opportunistic access through a multi-hop VANET whenever possible. So far, most of the proposed solutions use a single technology, or at most, suggest to use one for data forwarding and the other for complementary signaling purposes [65]. The use of one single technology is not suitable in real scenarios, as no one can independently meet the actual connectivity requirements posed by vehicular users (i.e., always-available connectivity, mobility support, moderate to high bandwidth) and, operators (i.e., requisites related to scalability and cost). The use of 3G allow to meet the availability and mobility requirements and partially the bandwidth one, but aggravates the scalability concern of mobile operators: 3G infrastructures are currently quite overloaded and there is no room for significant improvement using existing technologies (as the radio spectrum is limited). The use of WLAN access technologies potentially enables the provision of higher bandwidth to vehicular users, but poses important challenges in mobility manage-

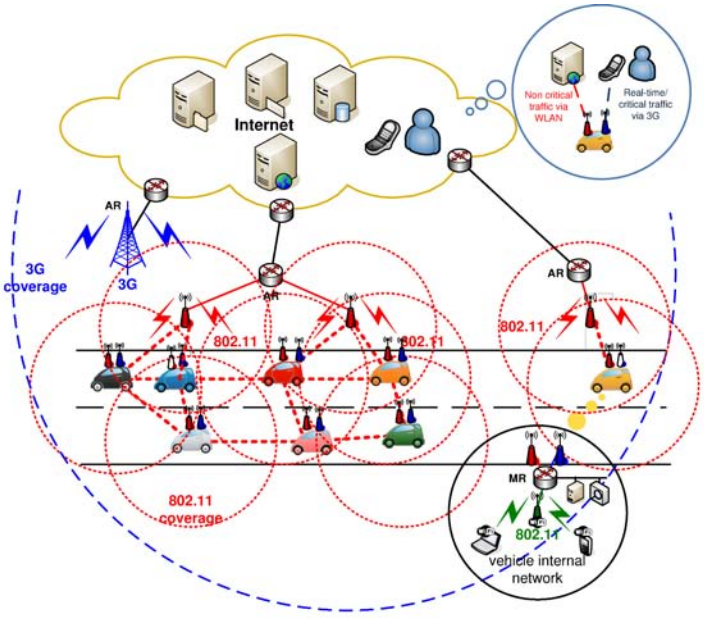


Figure 4.1: Vehicular Internet connectivity: scenario and solution overview

ment and cannot achieve the same level of connection availability. Even if a multi-hop VANET-alike scheme is used, full coverage cannot be ensured and additionally, routing in multi-hop networks poses important difficulties. Last but not least, mobility management becomes harder in this scenario, due to the frequent changes of layer-2 point of attachment.

SILVIO benefits from the simultaneous use of the two different access networks: 3G and WLAN. Since a permanent WLAN good connectivity cannot be ensured (even if a multi-hop scheme is adopted) SILVIO uses the 3G access network for critical or real-time constrained traffic (e.g., voice calls), which cannot afford any delays or interruptions. The WLAN connectivity is sensed, detected and used opportunistically for data packets of less critical applications, which can cope with some minimal delays and can also benefit from higher instantaneous available bandwidths (e.g., web browsing, file sharing, or video streaming, among others). Note that a key characteristic of our proposal is that this second category of data traffic is transparently moved from the 3G to the WLAN access – depending on the availability of WLAN connectivity – without any application support needed, thanks to the use of an IP flow mobility approach. Also, when WLAN is not available, all traffic is sent and received via the 3G interface, which generally implies that non-critical applications enjoy less bandwidth than when WLAN is also used.

To achieve this seamless and transparent traffic management with mobility support, SILVIO makes use of the following key technologies:

- *Network Mobility support.* Since a vehicle is equipped with more than one device

demanding IP connectivity, it is desirable to adopt an approach that provides connectivity to a set of devices, without any special support required on those. Note that some of these devices could be sensors with limited capabilities and processing power. This feature is achieved by using network mobility mechanisms.

- *3G offloading support.* The 3G interface is responsible for providing close-to always available connectivity, while the WLAN one is opportunistically used to offload non-critical traffic from the 3G network. This is achieved by using IP flow mobility mechanisms.

- *VANET connectivity support.* The most widely adopted approach to provide connectivity via WLAN in a vehicular scenario is the use of a multi-hop VANET. This requires specific IP address auto-configuration and routing protocols designed to operate in the vehicular environment, and enhanced for the V2I communications scenario. This is achieved by using TREBOL (as seen in Chapter 3), a vehicular addressing and routing protocol specially designed to support the V2I scenario.

- *Interface management support.* Due to the high mobility nature of the tackled scenario, handover management is of critical importance, as the impact caused by moving traffic between the 3G and WLAN interface has to be minimal from both the application and the user perspectives. This is achieved by using optimized interfaces and layer-2 handover mechanisms based on the IEEE 802.21 standard.

- *Smart handover procedures.* A critical part of the solution is the intelligence responsible for deciding when a flow has to be handed off to a different access network and when it has to be brought back. There are different handover-decision approaches that can be followed to maximize the overall performance (i.e., average bandwidth, end-to-end delay, etc), while minimizing the potential side effects that handover procedures (e.g., signaling) may cause. The trade-offs of the different approaches need to be assessed in order to take the best decision. The design and analysis of the handover procedures represents one of the main novelties of this thesis, and it is presented in this chapter.

We next cover each of the previous aspects in more detail, explaining the role played in SILVIO by each of the technologies, and highlighting the novel aspects of our proposal.

4.2.1 Basic Internet connectivity provision: Network Mobility

Vehicles contain several devices demanding to have Internet connectivity: internal sensors, on-board computers, infotainment back-seat boards, etc.; but also external devices, such as laptops or PDAs, carried by passengers. This basically means that a vehicle

can be seen as a network (or even as a set of networks) that is moving while connected to the Internet. In SILVIO these in-vehicular networks are provided with connectivity by means of using the NEMO Basic Support protocol.

The Network Mobility (NEMO) Basic Support protocol [28], proposed by the IETF[2], extends the basic end-host mobility solution, Mobile IPv6 [38], to provide network mobility support. In this solution, a mobile network (known also as *Network that Moves* – NEMO[3]) is defined as a network whose attachment point to the Internet varies with time. The router within the NEMO that connects to the Internet is called the Mobile Router (MR). It is assumed that the NEMO has a Home Network, connected to the Internet, where it resides when it is not moving. Since the NEMO is part of the Home Network, the Mobile Network Nodes (MNNs) have configured addresses belonging to one or more address blocks assigned to the Home Network: the Mobile Network Prefixes (MNPs). These addresses remain assigned to the NEMO even when it is away from home. Of course, these addresses only have topological meaning when the NEMO is at home. Thus, when the NEMO is away from home, packets addressed to the Mobile Network Nodes will still be routed to the Home Network, and redirected by the Home Agent (HA) to the current location of the MR. When the NEMO is connected to a visited network, the MR acquires an address from the visited network, called the Care-of Address (CoA), where the routing architecture can deliver packets without any additional mechanism (see Fig. 4.2).

4.2.2 Offloading the 3G network: flow mobility

In SILVIO, mobile routers are not only equipped with a 3G interface, but also with a WLAN one, that is opportunistically used to offload part of the traffic from the 3G one. Let's consider that the WLAN interface provides intermittent connectivity to the Internet (how this connectivity is achieved is the subject of Section 4.2.3), these connectivity opportunities can be used to dynamically move (i.e., offload) part of the total traffic sent/received by/at the vehicle to the WLAN access, reducing in this way the load of the 3G network. We describe in Section 4.3 the design of the automatic mechanisms (the *intelligence*) that decide which packets are moved, when they are moved, and from which interface to which interface. Next we explain the mechanism used by SILVIO to provide the capability of seamlessly moving traffic between interfaces, which is based on IP flow mobility extensions.

IP flow mobility refers to the movement of selected IP flows from one interface to another, of course minimizing the impact on the user experience. In order to do so, an IP mobility protocol should allow for the simultaneous use of different care-of addresses associated to the same home address. This feature must be able to selectively send packets

[2] `http://www.ietf.org/`

[3] NEMO can mean NEtwork MObility or NEtwork that MOves according to the context.

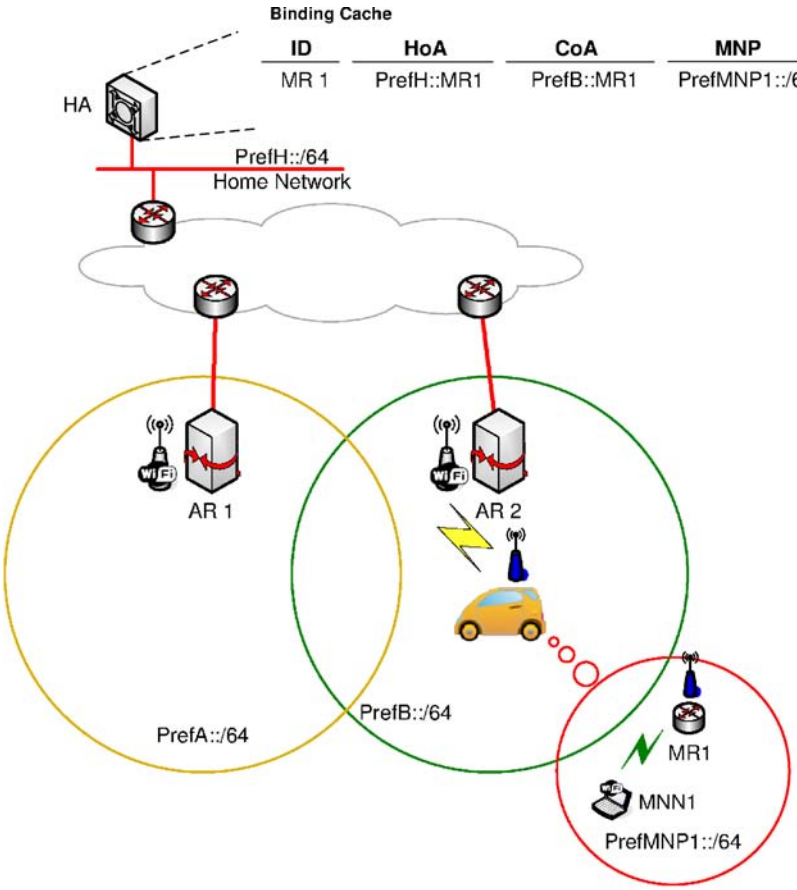


Figure 4.2: NEMO Basic Support protocol operation overview.

addressed to the same node via different access networks (identified by its CoAs). Regular Mobile IPv6 and NEMO B.S. do not provide flow mobility, so to enable it, the IETF has standardized the basic components that are required. These components are: *i)* multiple care-of address registration support (standardized in the RFC 5648 [72]), *ii)* flow bindings support (standardized in RFC 6088 [73] to allow mobile routers to bind one or more IP flows to a specific care-of address), and *iii)* traffic selectors definition (standardized in RFC 6089 [74]).

So far we have described the tools required to provide in-vehicle devices with Internet connectivity using one primary access technology (e.g., 3G), and to selective and opportunistically offload part of the traffic to another secondary access technology. In SILVIO, this secondary technology is WLAN, with connectivity being provided via a multi-hop VANET. We next describe how this is done in SILVIO.

4.2.3 Multi-hop WLAN vehicular Internet connectivity: addressing and routing

Providing wireless connectivity in VANETs in a cheap and scalable way has been a widely researched topic in the past decade. Finally, the need to achieve a cooperative and low latency network, motivated the use of dedicated short range communications devices for these purposes (like the IEEE 802.11p-capable ones). Vehicles traveling together form an ad hoc network, where the actual wireless coverage is extended using multiple hops. Vehicles take advantage of fixed nodes placed along the roads, called Road Side Units (RSUs), especially for safety-related purposes. However, as RSUs are connected to the infrastructure network, they can also be used to provide vehicles with Internet connectivity using a multi-hop wireless path.

As already stated throughout this thesis, there are at least two major functionalities that have to be enabled to efficiently connect vehicles to the Internet: *Address configuration* and *Routing*. Mobility management is also an important feature, which we have intentionally not included in the previous list, because SILVIO handles mobility in an integrated way, not limited to the access technology used (VANET in this case). Therefore, in this subsection we focus on the IP address configuration of the WLAN interface, and how packets are routed within the VANET. This is a problem that has been quite heavily investigated in the past, but that still poses may interesting research challenges.

SILVIO assumes the use of TREBOL as defined in chapter 3 to enable routing and addressing within the VANET, although it is integrated with the flow handover intelligence of SILVIO to decide when to opportunistically perform traffic offloading to the WLAN-based VANET (as explained in Section 4.3).

4.2.4 Seamless Interface Management: vehicular-aware IEEE 802.21

Since WLAN connectivity is intermittent, the mechanisms must efficiently detect when it is available, and also predict when it is expected to disappear (e.g., because the terminal is leaving the coverage area of its current point of attachment). This prediction should be done with enough anticipation, so the handover can be prepared and performed in such a way that packet losses are minimized.

SILVIO integrates and extends the IEEE 802.21 standard to enable an enhanced interface management, capable of providing seamless intra and inter-technology handovers. We next briefly present the IEEE 802.21 standard, before identifying the possible handover situations that may occur in our vehicular scenario, and explaining how SILVIO supports them all.

The IEEE 802.21 [75, 76] (or Media Independent Handover, MIH) technology is an enabler for the optimization of handovers between heterogeneous IEEE 802 systems as

well as between 802 and cellular systems. The goal is to provide the means to facilitate and improve the intelligence behind handover procedures, allowing vendors and operators to develop their own strategy and handover policies. For this purpose, the IEEE 802.21 aims at optimizing the handover procedure between heterogeneous networks by adding a technology independent function (Media Independent Handover Function, MIHF) which improves the communication between different entities, either locally (mobile node) or remotely (network functions).

Sharing information allows handover algorithms to guarantee seamlessness while moving across different points of attachment in the network and the use of common commands greatly simplifies the design of the algorithms.

MIH defines three main mobility services:

- The Media Independent Event Service (MIES) provides event classification, event filtering and event reporting, corresponding to dynamic changes in link characteristics, link status and link quality. This service is particularly important for SILVIO, since WLAN availability needs to be known as precisely as possible in order to benefit from it without introducing interruptions in the on-going communications.
- The Media Independent Command Service (MICS) enables MIH clients to manage and control the link behavior related to handovers and mobility. It also provides the means to mandate actions to lower layers, in a local or in a remote protocol stack.
- The Media Independent Information Service (MIIS) provides details on the characteristics and services provided by the serving and surrounding networks. The information enables effective system access and effective handover decisions.

The information exchange occurs between lower layers and higher layers, taking always the MIH Function as reference. Furthermore, information can be shared locally, within the same protocol stack, or remotely, between different network entities.

The IEEE 802.21 defines different roles according to the relationship between the network based MIHFs and the mobile node/router. In this way, it defines the concept of Point of Service (PoS) and Point of Attachment (PoA). The former identifies a network based MIHF that talks directly to a mobile node, while the latter corresponds to the network side end-point of an L2 link that includes the mobile node as the other endpoint.

In order to understand the role played by the IEEE 802.21 in SILVIO as well as the value added by our proposed extensions, we describe next the three potential flow handover scenarios that may arise:

- *3G-to-WLAN.* In this scenario, a vehicle can only send and receive traffic via a 3G network. At a certain point, WLAN connectivity to the RSU – either single or

multi-hop – becomes available. The mobile router deployed in the vehicle needs to be notified of such a WLAN availability event in order to be able to perform a flow handover, offloading part of the in-vehicle traffic from the 3G network to the VANET. IEEE 802.21 Link Up events can be used in this scenario to notify the MR about WLAN availability.

Regular IEEE 802.21 Link Up events are used to inform of 1-hop radio availability with a given point of attachment. However, in order to gain Internet connectivity, a mobile router in the VANET does not only need a WLAN point of attachment within its radio coverage, but also that this PoA is either the RSU or is connected (via a multi-hop path) to the RSU. Therefore, 1-hop radio connectivity between a vehicle and another PoA is not enough to trigger a Link Up event, unless the PoA is a RSU. SILVIO benefits from the TREBOL configuration messages used to regenerate the routing tree as enablers of Link Up events. If a vehicle receives a CM message, this means that the message has been successfully forwarded from the RSU to the receiver, and that can be used as a hint that the PoA is indeed able to provide Internet connectivity.

Upon detection and flow handover decision, the mobile router configures an IPv6 address valid on the VANET and sends the required IP mobility messages (i.e., flow binding updates) to its home agent, so it can proceed to forward packets belonging to the offloaded flow(s) to the IPv6 address configured in the VANET. Note that the time required to carry out this IP mobility signaling does not cause any traffic interruption, as packets continue being forwarded via the 3G (which is always on and available) until the home agent is updated and moves the flow(s) to the new WLAN access.

- *WLAN-to-3G.* In this scenario, a vehicle is simultaneously connected to the Internet via 3G and WLAN, so certain flows are being sent/received via the WLAN interface. If the vehicle cannot keep connectivity via its WLAN interface, the IP flows that were using the WLAN access have to be moved back to the 3G (waiting for a future offloading opportunity). Therefore, the mobile router needs to be notified of such an event, so it can send the mobility signalling required to move the flows back to 3G. This can be done by using IEEE 802.21 Link Down and Link Going Down events.

  Note that in a multi-hop VANET scenario, a vehicle can lose connectivity because of two different reasons: *i)* its parent (i.e., the node that is currently using to connect to the Internet) is no longer reachable; or *ii)* the chain of connected vehicles between the vehicle and the RSU becomes unconnected (i.e., one parent node becomes unreachable somewhere in the path). Note that *i)* is a particular case of *ii)*. In this case, it is very important to predict with enough anticipation if the WLAN access is about to become unusable, so the IP mobility signalling and associated

state updates are all done before the WLAN access becomes unusable (with the obvious packet losses).

Link Down and Link Going Down messages are triggered in SILVIO as follows: when a node detects that its parent node is going to become unreachable imminently, it advertises that event to all nodes (if any) in the VANET that are sending traffic which is traversing that parent node, so they can all move the traffic to their 3G interface. Note that a vehicle obtains information about all the nodes that are sending traffic in the VANET as part of regular TREBOL operation.

SILVIO uses layer-2 measurements to predict future wireless connectivity failures, by setting a threshold on the minimum RSSI that can be accepted as "good radio conditions". If the RSSI sensed from its parent node falls below this threshold, then the vehicle assumes that this radio link is about to fail shortly (e.g., because the nodes are moving away) and generates the signalling required to trigger a Link Going Down event. In [77] Meireles *et al.* give a comprehensive evaluation of the signal strength and the packet delivery ratio between two vehicles equipped with 802.11p devices[4]. In SILVIO, the threshold value used to trigger a Link Going Down event is configured according to the results reported in this evaluation report.

- *WLAN-to-WLAN.* Due to the logical division in areas adopted by TREBOL, not only inter-technology handovers may happen, but also intra-technology ones between different areas, causing the IP address configured on the WLAN interface to change (note that IPv6 addresses are anchored at the RSUs, and changing area implies using a different RSU). Therefore, this type of handover also needs to be predicted, to avoid packet losses.

  Existing IEEE 802.21 procedures and messages cannot be used to inform about a future potential change of point of attachment and IP address. SILVIO defines a new *Link Going Up* message that is sent when this event is predicted, allowing the mobile router to perform the required mobility signalling of a subsequent change of area in advance. SILVIO generates Link Going Up messages by overhearing CM messages from neighboring areas (in a way similar to what is reported in Chapter 2). When a node receives a CM that has been generated in a different area, it has to filter it to preserve the logical division in areas. However, the information contained in the message can be used to learn the IPv6 address of RSU on the neighboring area, as well of the IPv6 prefixes that should be used for address configuration purposes there.

[4]These results have been confirmed by us performing similar experiments using equivalent wireless hardware and drivers.

Algorithm 1 S-direct pseudo code.

```
1: if vehicle attached only to the 3G network then
2:     if (WLAN available) & (# hops = 1) then
3:         vehicle attaches also to WLAN
4:         vehicle offloads best-effort traffic to WLAN
5:     end if
6: else
7:     if vehicle attached to 3G and WLAN then
8:         if (vehicle loses Internet connectivity via WLAN) then
9:             vehicle moves back offloaded traffic to 3G
10:        end if
11:    end if
12: end if
```

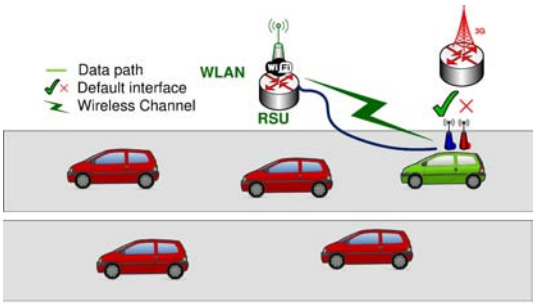


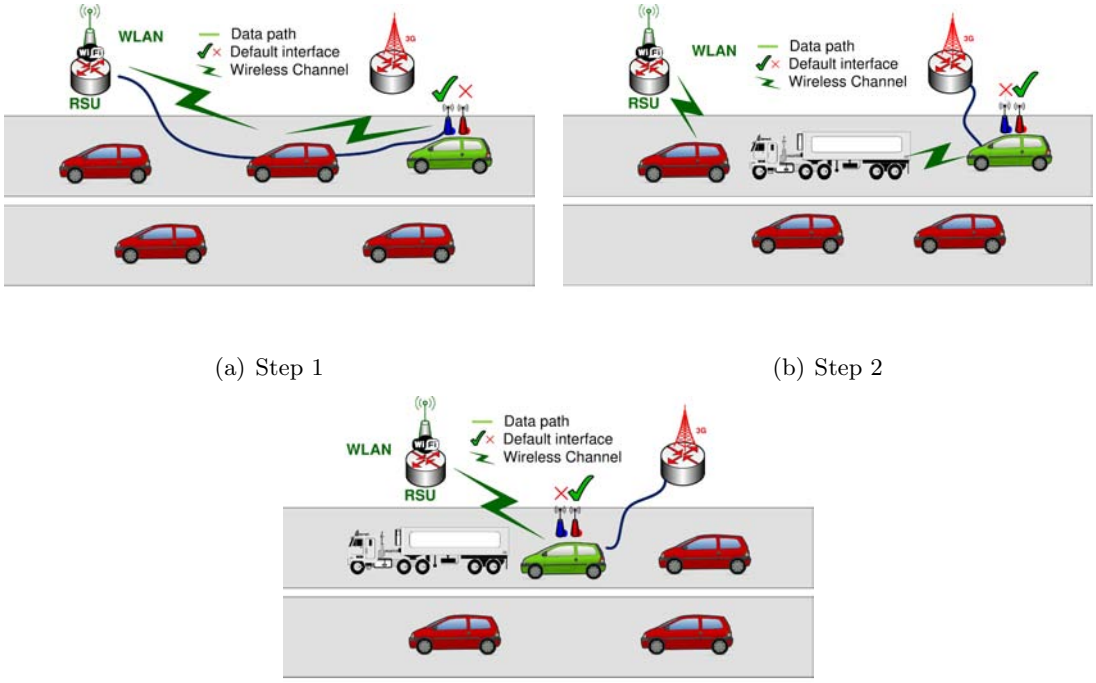
Figure 4.3: S-Direct mode of operation.

4.3 Protocol operation

In order to design a flow handover mechanism suitable for multi-hop wireless VANETs, a careful evaluation of many aspects – such as the vehicle mobility and the presence of two-way vehicular traffic – needs to be conducted. From the point of view of complexity, the simplest technique consists in not fully exploiting the multi-hop nature of VANETs and only use WLAN connectivity when the vehicle is directly connected to the RSU (i.e., 1-hop reachability).

Taking this as starting point, more complex and performance appealing approaches can be devised. Another aspect that needs to be taken into account is the cost associated to a flow handover. Since WLAN and 3G connectivity might exhibit disparate round-trip times, TCP connections can be affected as a result of a flow handover. Additionally, too aggressive handover approaches, may lead to handovers to WLAN when the quality and stability of the connectivity are not yet good enough, causing the so-called ping-pong effects (i.e., sequence of handovers back and forth between WLAN and 3G).

Another issue to be considered is related to the actual deployment of RSUs, which are usually placed alongside the roads. Given a fixed installation point, vehicles can get



(a) Step 1

(b) Step 2

(c) Step 3

Figure 4.4: S-Conservative

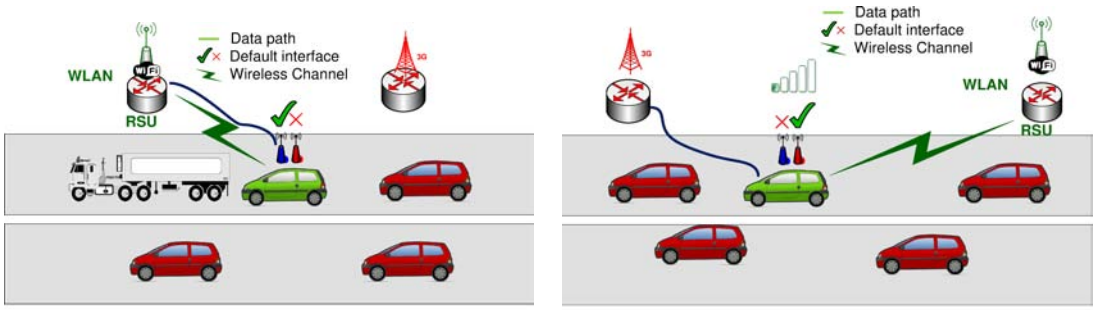
closer or go away to/from the RSU. However, while vehicles approaching to the RSU would take advantage of better wireless condition and even use a higher data rate [78], vehicles moving away would face the opposite.
Based on the previous analysis, we can highlight that there are many different handover approaches that can be followed, from those that are very conservative and aims at limiting the number of handovers, to those that are quite greedy and target using the VANET as much as possible. In order to illustrate and analyze the impact of this range of behaviors, we consider the following four different SILVIO modes of operation:

- SILVIO direct (*S-direct*). This mode, detailed in Algorithm 1 and Fig. 4.3, is the simplest solution we propose. The mobile router chooses to use the VANET just when it is under direct wireless connectivity with the RSU. The high reliability of the wireless channel makes this solution the safest in terms of stability of the attempted handovers, but, on the other hand, it does not exploit the potential benefits of the multi-hop WLAN connectivity. This algorithm provides the least number of handovers (i.e. just one) and the shortest time using the VANET so it is used in comparison to other strategies to evaluate their effectiveness.

- SILVIO conservative (*S-conservative*). This mode, detailed in Algorithm 2, tries to use a possible multi-hop WLAN connectivity through the VANET (i.e., not only

Algorithm 2 S-conservative pseudo code

```
1: if vehicle attached only to the 3G network then
2:     if (WLAN available) & (# hops ≤ t-hops-wlan) & (# attempts < 1) &
   (vehicle getting closer to the RSU) then
3:         vehicle attaches also to WLAN
4:         vehicle offloads best-effort traffic to WLAN
5:     end if
6: else
7:     if vehicle attached to 3G and WLAN then
8:         if (vehicle loses Internet connectivity via WLAN) then
9:             vehicle moves back offloaded traffic to 3G
10:        end if
11:    end if
12: end if
```



(a) Step 3 (b) Step 4

Figure 4.5: S-Persistent.

when directly connected to the RSU). In order to mitigate the possible ping-pong effect the mobile router is allowed to perform a handover to the multi-hop WLAN just once per TREBOL area.

That is, once the mobile router backs off to the 3G network, it will not try again to obtain connectivity via the WLAN interface until it changes area. Moreover, the VANET WLAN is only used if the number of hops between the mobile router and the RSU is smaller than a configurable threshold (`t-hops-wlan`). In Fig. 4.4 the method of operation is detailed. When a vehicle finds a new available path through the multi-hop VANET, it performs a flow handover to WLAN (Fig. 4.4(a)).

However, if the WLAN connectivity cannot be maintained (e.g., due to an obstacle between two hops, as shown in Fig. 4.4(b)), the mobile router switches back to the 3G interface, and keeps using it even if a new valid path is discovered (Fig. 4.4(c)).

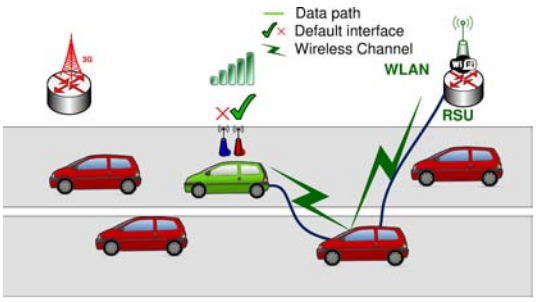
- SILVIO persistent (*S-persistent*). This mode of operation is slightly more aggressive than the ones already presented before. It tries to make use of the VANET longer

Algorithm 3 S-persistent and S-sticky pseudo code.

```
1: if vehicle attached only to the 3G network then
2:     if (WLAN available) & (vehicle getting closer to the RSU) then
3:         if (# hops ≤ t1-hops-wlan) & (# attempts < t-ho-attempts) then
4:             vehicle attaches also to WLAN
5:             vehicle offloads best-effort traffic to WLAN
6:         else
7:             if (# hops ≤ t2-hops-wlan) then
8:                 vehicle attaches also to WLAN
9:                 vehicle offloads best-effort traffic to WLAN
10:            end if
11:        end if
12:    end if
13: else
14:    if vehicle attached to 3G and WLAN then
15:        if (vehicle loses WLAN connectivity) then
16:            vehicle moves back offloaded traffic to 3G
17:        end if
18:    end if
19: end if
```

by allowing the mobile router to hand off between 3G and WLAN more than once per area, but only while the vehicle is getting closer to the RSU, and if the number of hops between the router and the RSU is smaller than `t1-hops-wlan`.
The maximum number of handover attempts from 3G to WLAN is limited by a configurable threshold (`t-ho-attempts`). However there is a exception: if the VANET route is shorter than `t2-hops-wlan` hops, then the number of 3G to WLAN handover attempts is not limited.
Algorithm 3 describes the handover behaviour of this mode. As shown in Fig. 4.5 when the mobile router finds a new path to the RSU, it hands off again the selected flow to the WLAN. When moving away, if the mobile router decides that the WLAN link does not satisfy its quality constraints, it moves the flows back to the 3G network, without looking to other possible paths in the VANET.

- SILVIO sticky (*S-sticky*). This is the greediest operation mode, in which the mobile router tries to keep using the VANET even when moving away from the RSU. From a pure handover management point of view it is exactly the same as the S-persistent mode (therefore Algorithm 3 is also valid to describe the operation of this mode). The only difference (depicted in Fig. 4.6) between S-persistent and S-sticky modes is that in the latter case the mobile router is allowed to find an alternative multi-hop path using the WLAN even if the node is moving away. This variation is only possible while the mobile router is connecting to the RSU via a multi-hop path shorter than a certain threshold (called `t-multihop`).



(a) Step 4

Figure 4.6: S-Sticky.

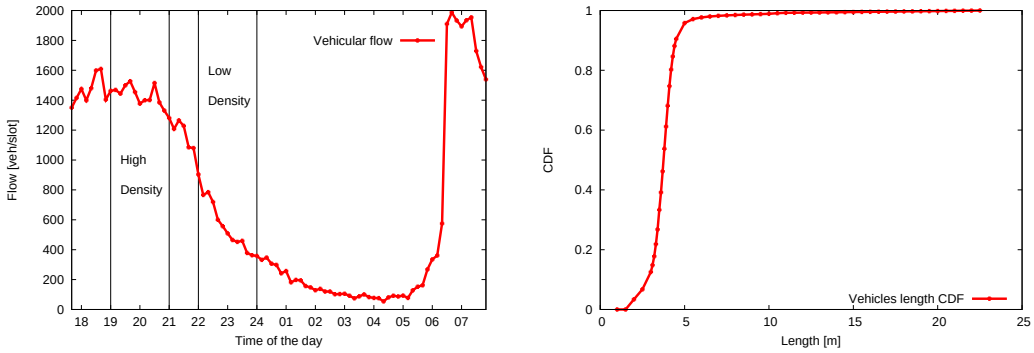
Note that in the rest of SILVIO operation modes that is not allowed, and therefore if a mobile router required to change TREBOL parent when moving away from the RSU to keep its connectivity, that would trigger a handoff to 3G.

4.4 Evaluation

This section provides an experimental evaluation of SILVIO. The main aim of this evaluation is two-fold: *i)* to study the performance of the different modes of SILVIO (i.e., different algorithms for smart vehicular handover management) and compare them with a solution based the sole use of 3G, and *ii)* to understand the trade-offs of each of the SILVIO operating modes. We start by introducing the evaluation framework, since it is a very important piece of our work, and then we present and analyze the obtained results.

4.4.1 Evaluation framework

As already pointed out in Chapter 3, extensively evaluating network protocols for VANETs is a task difficult to accomplish. This reasoning leads us to using simulation as the only feasible tool to conduct our study on the performance of SILVIO. However, modeling the wireless environment precisely in this scenario is quite hard, and virtually impossible in practical terms. Therefore, some simplifications and assumptions need to be made, which more often than not, tend to invalidate the obtained results, as the simulation engine does not take into account aspects as important as the vehicular mobility patterns or the obstruction that vehicles themselves may represent to the wireless signal. Boban *et al.* [79] introduced for the first time the problem of considering other vehicles as obstruction for wireless communications. Our simulation framework takes into account the obstacles represented by large vehicles (e.g., trucks, vans) as we explain later in this section.



(a) The vehicular flow of the used traffic traces (b) The CDF of the vehicles length

Figure 4.7: Vehicular traffic characteristics.

In order to overcome the problem of oversimplifying the position and speed of vehicles during each simulation run, we build our simulated scenario upon realistic vehicular traces. In particular, we use SUMO fed with real 12-hour vehicular traces (see Fig. 4.7(a)). These traffic traces, kindly provided by the Madrid city council, were collected at a fixed measurement point placed along the M-30 orbital motorway in Madrid.

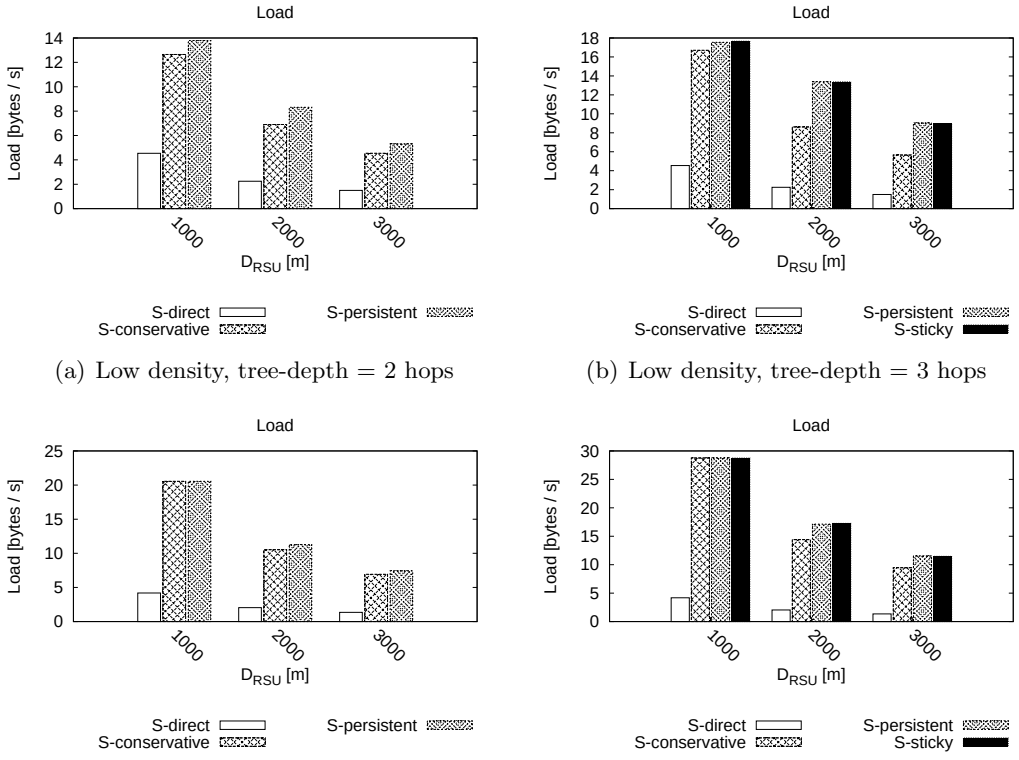
Our data set includes also the length of the vehicles (see Fig. 4.7(b)). This information is used to infer the type of the vehicle: short (e.g., less than 6 m) and large in other case (accounting for less than the 3% of the total). Our simulations take into account the type of vehicle, with large vehicles representing obstacles that block the wireless signal.

Our evaluation framework is based on Java simulator that merges the output coming from SUMO with the calculated RSSI values. RSSI values are calculated from [77] depending on the distance between vehicles[5]. The value of the inter-vehicle distance fixes the average and the standard deviation of a Gaussian distribution from which the actual value of the RSSI is calculated. The same work [77] shows that the minimum RSSI value that provides a packet delivery ratio greater than 90% is 12, for a wireless card with an Atheros-based chipset.

The wireless channel quality between two nodes is considered to be bad if the RSSI remains for more than 0.3 s below the reference value of 12. Additionally, if a large vehicle (i.e., longer than 6 m) appears between a vehicle and its next-hop vehicle, the channel quality is considered to be under the threshold as well. Note that this last rule is not applied for the direct connectivity between a vehicle and the RSU, as we claim that RSUs will be installed in elevated positions where the influence of large vehicles would be negligible.

For the experiments, a simulated city highway scenario is used. Road side units are deployed at different distances (denoted by D_{RSU}, which is the distance between two

[5]The authors of [77] kindly provided us the full experiment traces, that we used to estimate the RSSI values for a given inter-vehicle distance in both Line-of-Sight (LOS) and non LOS conditions.



(a) Low density, tree-depth = 2 hops

(b) Low density, tree-depth = 3 hops

(c) High density, tree-depth = 2 hops

(d) High density, tree-depth = 3 hops

Figure 4.8: Control Overhead

consecutive RSUs): every 1000, 2000 and 3000 meters. Each trace comprises 500 vehicles, while the total running time depends on the actual vehicular flow. We divided the traces into two categories (see Fig. 4.7(a)): high density (i.e., 500-vehicle data sets between 19 and 21h) and low density (i.e., 500-vehicle data sets between 22 and 24h).

4.4.2 Results and performance analysis

In this section we present a performance analysis based on the results obtained with the simulator described above. Using the simulator on the previously described scenario, the following statistics can be obtained: time spent by each vehicle in 3G and WLAN, number of hops to the RSU when connected via WLAN, number of handovers per TREBOL area and signalling overhead. Note that, while the developed simulation based framework does not completely model a vehicular network, it does cover the critical components required to evaluate the previous metrics. The goal of this analysis is not only to characterize the performance of the solution, but also to study how each of the operation modes of SILVIO behaves, as well as the impact of the different configuration parameters (i.e., the thresholds). This allows us to gain understanding on how to deploy SILVIO, depending on which is the target scenario.

D_{RSU} [m]	Density	Time 3G [s]	Time WLAN [s]	Offload [%]
1000	LO	17.94 ± 0.09	22.57 ± 0.09	55.72
1000	HI	19.50 ± 0.10	24.48 ± 0.10	55.66
2000	LO	59.08 ± 0.11	22.78 ± 0.11	27.83
2000	HI	64.98 ± 0.11	25.11 ± 0.11	27.87
3000	LO	100.42 ± 0.09	22.79 ± 0.09	18.50
3000	HI	110.75 ± 0.10	25.27 ± 0.10	18.58

Table 4.1: Handover strategy: S-direct

While presenting the performance results, we analyze them from two different points of view: the mobile operator's and the user's. SILVIO has been designed with these two players in mind, as it is a solution that aims at providing the best possible connectivity experience to vehicular users, while being feasible and not aggravating the congestion problems that operators are already facing in their 3G networks.

We first start from the operator's view point. A critical performance figure of any system is the control overhead. SILVIO makes use of signalling messages to dynamically move flows between interfaces. Depending on the type of the handover, the control overhead is different:

- *3G-to-WLAN.* IP mobility messages (BU/BA) are sent via the WLAN interface to inform the home agent about the flow(s) handoff(s).

- *WLAN-to-3G.* In this case, in addition of the IP mobility messages (sent via the 3G interface), the TREBOL parent node predicting an imminent loss of connectivity informs those nodes down in the tree that are sending traffic through it, so they can switch to 3G before the WLAN connectivity is lost.

- *WLAN-to-WLAN.* In this case only IP mobility signaling is required.

We evaluate the average load (in bytes/sec) generated by each vehicle while moving and crossing different TREBOL areas. Note that this load depends on the number of flow handovers performed. According to [74], the signalling used by the mobile router to inform the home agent about changes in the flow handling consists of two additional headers to the standard Binding Update message (BU_H), namely Binding Identifier Mobility Option (BIM_H) and Flow Identification Mobility Option (FIM_H). Then a traffic selector (TS_p, defined in [73]) has to be specified for each updated flow. We consider, for this evaluation, that each node performs the update for a single flow, so one IPv6 traffic selector is included in the message. The overhead injected by an update is therefore given by:

Table 4.2: Handover strategy: S-conservative

tree-depth	D_{RSU} [m]	Density	Time 3G [s]	Time WLAN [s]	Offload [%]	# Handoffs	Improvement [%]
2	1000	LO	13.99 ± 0.10	26.52 ± 0.10	65.47	1.14	20.60
	1000	HI	12.63 ± 0.08	31.35 ± 0.08	71.28	1	28.70
	2000	LO	57.33 ± 0.10	24.53 ± 0.10	29.97	1.29	5.66
	2000	HI	63.58 ± 0.08	26.51 ± 0.08	29.43	1.19	5.54
	3000	LO	98.54 ± 0.11	24.68 ± 0.11	20.03	1.24	4.54
	3000	HI	109.54 ± 0.08	26.48 ± 0.08	19.47	1.13	1.80
3	1000	LO	13.29 ± 0.10	27.22 ± 0.10	67.19	1.09	17.51
	1000	HI	12.47 ± 0.08	31.50 ± 0.08	71.64	1	28.07
	2000	LO	57.79 ± 0.08	24.07 ± 0.08	29.41	1.38	7.67
	2000	HI	63.59 ± 0.11	26.50 ± 0.11	29.41	1.24	5.60
	3000	LO	99.38 ± 0.08	23.83 ±0.08	19.34	1.35	8.26
	3000	HI	110.30 ± 0.10	25.73 ±0.10	18.91	1.21	4.78

$$\begin{aligned} O_{NEMO} &= IPv6_H + BU_H + BIM_H + FIM_H + TS_P = \\ &= 40 + 8 + 24 + 8 + 104 = 184\text{bytes}, \end{aligned} \tag{4.1}$$

where $IPv6_H$ represents the length of the IPv6 header. The TREBOL related signalling required to notify about a predicted loss of connectivity to the RSU depends on the number of affected nodes that need to be notified (i.e., the nodes that are sending traffic via a multi-hop path that is about to get disconnected). In order to illustrate this control overhead, we assume that on average half of the vehicles are affected when a WLAN-to-3G handover is triggered. This basically means that about 50 and 16 vehicles are affected for the case of high and low density cases respectively (i.e., for a radio coverage of 300 m). Hence the overhead introduced by TREBOL is given by:

$$\begin{aligned} O_{TREBOL} &= IPv6_H + RA_H + TREBOL_H + \\ &+ (n_{hosts} \times host_{EUI48}) = \\ &= 40 + 16 + 4 + (n_{hosts} \times 6), \end{aligned} \tag{4.2}$$

where RA_H and $TREBOL_H$ are the Router Advertisement and TREBOL headers, respectively. The hosts EUI48 addresses $host_{EUI48}$ are used as their unique identifiers. Using these values the total overhead is 156 bytes for the low density case and 360 bytes for the high density one. The load caused by this message is for a single hop, so it has to be multiplied for the maximum depth of the tree generated by TREBOL.

Fig. 4.8 shows the per-node control overhead of the four different SILVIO operating modes considered (S-direct, S-conservative, S-persistent and S-sticky), for different vehicular densities and deployment scenarios. As expected, the signalling overhead of S-direct

tree-depth	D_{RSU} [m]	Density	Time 3G [s]	Time WLAN [s]	Offload [%]	# Handoffs	Improvement [%]
2	1000	LO	11.26 ± 0.08	29.25 ± 0.08	72.20	1.34	29.59
	1000	HI	11.10 ± 0.09	32.87 ± 0.09	74.75	1	34.31
	2000	LO	51.21 ± 0.09	30.65 ± 0.09	37.44	2.00	34.52
	2000	HI	54.48 ± 0.09	35.61 ± 0.09	39.53	1.60	41.83
	3000	LO	92.71 ±0.10	30.51 ± 0.10	24.76	1.86	33.84
	3000	HI	100.03 ± 0.11	36.00 ± 0.11	26.46	1.58	42.43
3	1000	LO	10.86 ± 0.08	29.65 ± 0.08	73.20	1.32	31.38
	1000	HI	11.02 ± 0.11	32.96 ± 0.11	74.95	1	34.65
	2000	LO	44.78 ± 0.09	37.08 ± 0.09	45.29	3.41	62.74
	2000	HI	45.01 ± 0.09	45.08 ± 0.09	50.04	2.52	79.55
	3000	LO	86.47 ± 0.10	36.75 ± 0.10	29.82	3.51	61.21
	3000	HI	90.32 ± 0.09	45.70 ± 0.09	33.60	2.65	80.83

Table 4.3: Handover strategy: S-persistent.

is the lowest one, as it can only offload traffic when the vehicle is directly connected to the RSU. As expected, when SILVIO operates more aggressively in terms of WLAN stickiness, the generated overhead increases with the number of handovers. S-conservative is the next one after S-direct in terms of overhead, due to its limit on the number of handover attempts. Note that the average control load decreases with the size of the TREBOL area, as the handover frequency decreases too. In all cases, the overall control load is very low, and therefore it does not represent a burden on the feasibility of the solution deployment.

A performance metric that is relevant to both users and operators is the amount of time that the solution allows to effectively offload traffic from 3G to WLAN.

For each of the three considered inter-RSU distances and each of the two vehicular densities, we calculate the time spent by a non-critical (i.e., one that is dynamically moved depending on the availability of WLAN connectivity) flow through the 3G and WLAN interfaces. Table 4.2 shows the obtained results while traversing one area for the S-direct approach, namely the time spent in 3G, the time spent in WLAN and the offloading percentage, meaning the portion of the total where offloading from 3G to WLAN is possible. Average values and 95% confidence intervals are provided. For $D_{RSU} = 1000m$, S-direct is able to offload traffic about half of the time a vehicle spends in the area. That means that during this time the user obtains on average a higher throughput, since WLAN connectivity via one single hop to the RSU is usually faster than a 3G connection. In addition to this advantage from the user viewpoint, S-direct also benefits the mobile operator, as it helps reducing the load of its 3G network. Obtained results show that as the size of the TREBOL area increases, the benefits of offloading decreases, since nodes are only allowed to connect to WLAN when directly connected to the RSU. From this result, we can conclude that it is worth exploring how the other SILVIO operation modes perform.

Table 4.2 shows the results for S-conservative, for two different values of `tree-depth`.

tree-depth	D_{RSU} [m]	Density	Time 3G [s]	Time WLAN [s]	Offload [%]	# Handoffs	Improvement [%]
2	1000	LO	10.51 ± 0.11	29.99 ± 0.11	74.04	1.30	32.89
	1000	HI	10.15 ± 0.10	33.81 ± 0.10	76.90	1	38.15
	2000	LO	49.53 ± 0.09	32.32 ± 0.09	39.49	2.13	41.90
	2000	HI	52.40 ± 0.09	37.68 ± 0.09	41.82	1.63	50.09
	3000	LO	91.35 ± 0.09	31.85 ± 0.09	25.85	2.11	39.74
	3000	HI	98.33 ± 0.09	37.68 ± 0.09	27.70	1.49	49.12
3	1000	LO	10.46 ± 0.09	30.04 ± 0.09	74.17	1.34	33.13
	1000	HI	10.12 ± 0.10	33.84 ± 0.10	76.97	1	38.30
	2000	LO	44.16 ± 0.10	37.69 ± 0.10	46.04	3.40	65.45
	2000	HI	43.69 ± 0.08	46.39 ± 0.08	51.49	2.58	84.78
	3000	LO	85.89 ± 0.08	37.32 ± 0.08	30.29	3.46	63.74
	3000	HI	89.05 ± 0.11	46.96 ± 0.11	34.52	2.61	85.83

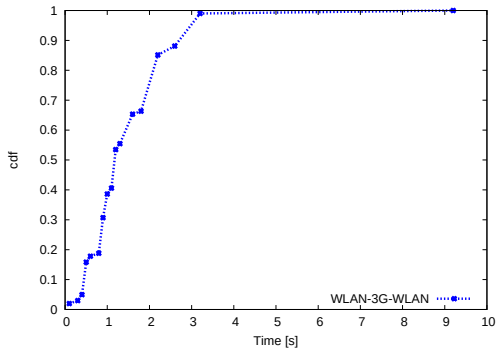
Table 4.4: Handover strategy: S-sticky, t-multihop=2 moving away.

This table, in addition to the results shown for S-direct, also includes the average number of handoffs from WLAN to 3G, as well as the performance improvement – in terms of WLAN usage – over S-direct. In the simulations, the value chosen for `t-hops-wlan` is always equal to the value used for `tree-depth`.

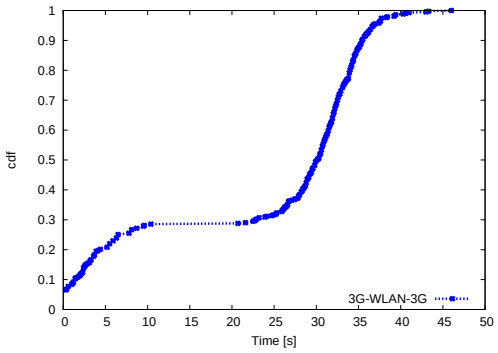
From the obtained results, it is worth highlighting the improvement achieved for the deployment scenario in which RSUs are installed every 1000m. This can be explained by the fact that S-conservative allows vehicles to try getting multi-hop connectivity through the VANET, though only once per area. This is actually the reason of this improvement not being bigger for longer D_{RSU} values (2000 and 3000 m).

From the obtained results, we can conclude that S-direct or S-conservative modes do not allow for fully benefiting from multi-hop WLAN connectivity through the VANET, as the time spent on WLAN per area is practically the same, regardless of the value of D_{RSU}. This is a expected result, since S-direct can only use the WLAN access when directly connected to the RSU, and S-conservative does not allow for more than one multi-hop WLAN connectivity attempt. Table 4.4.2 shows the results obtained with S-persistent, which is the first mode that allows for several multi-hop connectivity attempts. Note that `t1-hops-wlan` and `t2-hops-wlan` are equal to the value used for `tree-depth` in the simulations, and that `t-ho-attempts`=∞. Results show that S-persistent is able to significantly increase the performance also when the distance between RSUs increases, at the cost of a higher number of handovers.

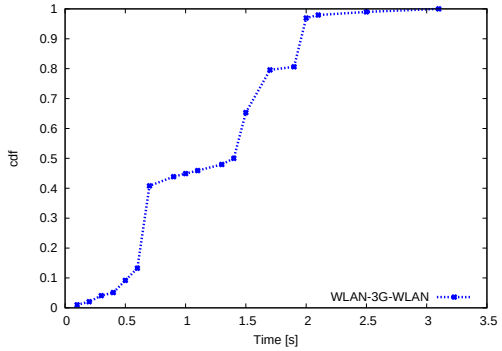
Finally, results for S-sticky are presented in Table 4.4.2, showing similar results to what is obtained with S-persistent. If we look at all the results, it can be observed that increasing `t-hops-WLAN` improves the time exploiting WLAN connectivity. However, this improvement does not come without costs, as we explain next. Fig. 4.9 shows the downside of increasing `t-hops-WLAN`. On one hand the periods of WLAN connectivity are larger, as there are more opportunities to connect to the VANET, but on the other hand, this also causes a more intermittent connectivity, with "holes" in the WLAN multi-hop path that



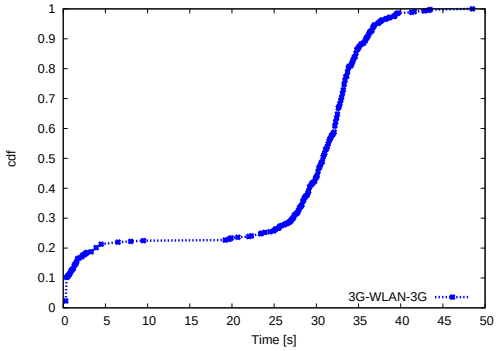
(a) Time spent by a vehicle using only the 3G network in a WLAN-to-3G-to-WLAN scenario, $t - multihop = 2$



(b) Time spent by a vehicle using only the 3G network in a WLAN-to-3G-to-WLAN scenario, $t - multihop = 3$



(c) Time spent by a vehicle using also the WLAN network in a 3G-to-WLAN-to-3G scenario, $t - multihop = 2$



(d) Time spent by a vehicle using also the WLAN network in a 3G-to-WLAN-to-3G scenario, $t - multihop = 3$

Figure 4.9: Cumulative distribution function of the time spent connected to 3G and WLAN.

have to be fixed by moving to the 3G network the affected flows. Besides, these periods of time spent on the 3G while finding another offloading opportunity are on average shorter. This consecutive and frequent handoffs may have a negative impact on the user's experience. As an example, TCP RTT estimation may get affected by flow handoffs, due to the likely disparate access delays of WLAN and 3G networks.

S-sticky gets on average better results than S-persistent due to the fact that, using this strategy, vehicles try to keep the WLAN connectivity using a multi-hop path even when they are traveling away from the RSU. We should note that staying connected using the WLAN when moving away from the RSU can, however, be a double-edged sword, as sharing the same RSU for vehicles moving in opposite direction results in an unfair usage of the WLAN resources. Since installing separate RSUs for each direction is not economically feasible, it is better, from a system point of view to deploy the S-persistent solution, which offers a very similar performance, while not suffering from the unfair WLAN sharing issue. Based on the obtained results, it seems that the best operation

mode is S-sticky using a value of `t-hops-WLAN` not very high (i.e., equal or less than 2) to avoid an excessive number of handovers. However, if the deployment mode does not allow to set up different RSUs for each driving direction, then S-persistent mode proves to be more appropriate. In both cases, the improvement gains in terms of amount time connecting to the VANET are around the 80%.

4.5 Comparison with previous work

Maximizing the time vehicles are connected to Internet through an 802.11 based network is a research topic that has been explored in the last few years. In [7], Giannoulis et al. propose an enhanced handoff mechanism for 802.11 WLAN devices. Using a smarter handoff strategy oriented to frequent access point changes they get an average throughput 2.5 times greater than the default handover policy implemented in the standard driver. However their work is focused just on improving the 1-hop connectivity to the APs and, during their test drives, they experienced throughput outages depending on the physical wireless coverage. Moreover they do not evaluate the coexistence of their proposal with standard IP mobility solutions and obviate the IP address configuration process that is, especially in vehicular environments, a crucial task.

Annese *et al.* propose in [80] to use a modified version of the BATMAN [81] layer-2 routing protocol for providing wireless connectivity to the infrastructure network. They were able to handoff between several APs without loss of connectivity, achieving a quite constant throughput. Despite providing layer-2 routing makes easier to solve mobility problems (i.e., just one IP address configuration process, use of the same address through different APs, etc.), on long term the flat view of the network from the layer-3 perspective can cause scalability issues.

A position based handoff strategy was proposed by Deshpande *et al.* in [82]. By exploiting the drivers' habit to daily drive similar routes (e.g., home to work, home to school) the authors define a handover protocol that connects to the best available AP depending on the vehicle position. Their mechanism is composed by a learning phase, where vehicles keep track of the APs signal strength for each position in their route and a handover process that selects the AP to connect to, accordingly with the measurements previously performed. In the same work the authors propose a pre-fetch mechanism that spans the download of a user requested file over all the APs that are encountered on the route. Although the authors do not propose any IP mobility solution to be coupled with the mechanism and completely disregards the address auto-configuration tasks, this work provides a good approximation of the availability of WLAN coverage in a real scenario.

In a more recent work, Deshpande *et al.* [83] extended their testbed to a metro-scale WLAN network, performing long drives that lasted more than 5 hours. Their throughput

measurements, although do not consider at all IP mobility or addressing configuration issues, show that with a feasible APs deployment they could get good throughput values during all the drive and for a fair amount of time (the time fraction with throughput equal to zero was around 30%). The authors show also a comparison of the throughput obtained with the WLAN device to the one obtained with the 3G network. The results they show state that when both technologies are available, the throughput achieved by WLAN overcome in the 90% of the cases the one obtained with the 3G.

A similar work was developed by Balasubramanian et al. in [8], where the authors tested the WLAN and 3G access in three different cities. Despite they results show that the bandwidth achievable using a WLAN device is not always higher than the one achievable with the 3G network, the need of offloading the 3G network still justified a mechanism for switching between the two technologies. Hence they propose *Wiffler* a sub-IP layer that, based on the traffic characteristics (e.g., needed bandwidth or delay) and on the prediction of the WLAN connectivity in the near future, switches the user flows to the best interface. Results show that a considerable part of the transferred data (up to 60%, depending on the allowed delay) can be offloaded from the 3G network to the WLAN.

Different issues posed by the integration of 3G and WLAN networks have been extensively studied, such as interworking architecture, mobility management and QoS aspects [84, 85]. However, previous works assume terminals connect directly to the WLAN access points giving them access the Internet, while our work analyze the case of having WLAN multi-hop paths to the access points.

The cited works show promising results achievable from the use of WLAN access in vehicular environments. However these works completely ignore IP mobility-related issues such as address auto-configuration or seamless handoff between APs or wireless technology. Moreover they just consider 1-hop WLAN connectivity while most of the already standardized solution for vehicular networks envision the use of multi-hop wireless communications. These two factors led us to propose a complete framework for providing seamless mobility in vehicular environments between heterogeneous technologies, namely multi-hop WLAN and 3G.

On the other hand, there also exists a considerable amount of previous work in the area of IP mobility in heterogeneous environments, even considering flow mobility [86] and offloading techniques [87]. However, most of the IP mobility works proposed so far [88] do not deal with the specifics of the vehicular scenario, and those that tackle it do not attempt to opportunistically benefit from multi-hop WLAN connectivity, enhancing in this way both users' and operators' interests, as SILVIO does. Last but not least, the level of integration of the IP mobility support (at flow level) with multi-hop addressing and routing mechanisms adopted by SILVIO, and the detailed analysis of different handover

strategies, are also novel contributions of this thesis.

4.6 Summary

In this chapter we presented SILVIO, a compound mobility management solution for vehicular environments. The cellular network is used to obtain "always on" connectivity, while some selected flow are opportunistically offloaded to the wireless multi-hop VANETs when a path to the Road Side Unit becomes available. Intelligent handoff procedures are obtained using already standardized techniques (i.e. flow mobility) or vehicular modification of them (802.11 Media Indipendent Handovers). Results obtained by simulations show that non-critical flows can be efficiently offloaded from the cellular network, using the wireless multihop VANETs up to the 80% of the time.

Part II

Traffic efficiency optimizations

Chapter 5

Vehicular trace analysis

An interesting intermediate phase that we had to go through, before designing traffic efficiency algorithms, is to correctly understand the interaction among vehicles traveling in the same road. This analysis is also useful to check the correctness of the assumptions made while validating the *Infotainment* optimizations previously described in this thesis. The analysis of the distance between vehicles in roads is a key factor in, e.g., designing vehicular networks protocols or planning a supporting infrastructure to improve vehicular connectivity. The work detailed in this chapter proposes a Gaussian-exponential mixture model to characterize the time distance between vehicles in a highway lane, based on measurements collected at different locations in several highways of the city of Madrid. The model arises from the observed behavior that some vehicles travel very close together, like in a burst mode, showing Gaussian inter-arrival times, while other vehicles are somehow isolated, showing exponentially distributed inter-arrival times. The experiments show that such a Gaussian-exponential mixture model accurately characterizes inter-vehicle times observed from real traces. This model for the inter-vehicle times can be used for improving the VANET-based traffic efficiency algorithms.

5.1 Goals

Before vehicular services can be successfully deployed, it is necessary to identify the requirements to build a vehicular network efficiently and effectively. Thus, characterizing the distribution of vehicles and their speed in highways is a required step in order to be able to design mechanisms that can operate efficiently in the vehicle environment and that can cope with its particular nature. Analyzing the connectivity level of a vehicular network requires building a model of vehicular traffic from real measurements. Depending on which technology the vehicles use to communicate, it is necessary one model or another. When vehicles communicate using wireless short-range technologies (i.e., IEEE 802.11p [89]), the traffic model must consider the aggregation of vehicles from different lanes in a highway. However, for the case of infrared communications [9], or the recent cases of visible light communications [10, 11], a model for single-lane traffic is necessary.

A number of previous modeling studies have attempted to characterize traffic inter-arrival times in the past [13, 20, 90]. For instance, the authors in [20] claim that vehicle inter-arrival times follow an exponential distribution, while the authors in [13] consider the log-normal distribution as the most suitable one to characterize inter-vehicular times. Furthermore, these two studies also assume that inter-arrival times are independent and identically distributed random variables. However, as we will show in the next sections, this is not the case, as there actually are some dependences between consecutive vehicles, especially at highways with dense traffic.

Indeed, we show from our traces that many vehicles travel close together like in a burst fashion, while others are isolated. Such bursty vehicles exhibit Gaussian inter-arrival times, while the others show exponentially distributed times. Based on these observations, we propose an exponential-Gaussian mixture model to characterize the inter-arrival times of vehicles, as observed from real traffic measurements collected from several highways in the surroundings of the city of Madrid, Spain.

5.2 Data set description

Table 5.1 summarizes the data set used in this work. This comprises $M = 15$ traces collected from different measuring points on four important highways in the surroundings of the city of Madrid, in Spain. Note that the traces are presented in decreasing order of traffic density, in terms of vehicles per hour. The considered highways are:

- M-40 and M-50 which are two orbital highways that surround the City of Madrid. Their speed limits are 100 and 120 km/h respectively.

Figure 5.1: Location of the highways and the measurement points.

- A-6 is a highway linking the north-west of Spain and Madrid. Its speed limit is 120 km/h.

- A-1 is a highway linking the north-east of Spain and Madrid. Its speed limit is also 120 km/h.

Fig. 5.1 shows the four highways along with the location of the measuring points (A to E in the figure).

The traces were collected in the morning of four consecutive working days, at two different time intervals: from 8.30 to 9.00 AM and from 11.30 to 12.00 AM. For trace j ($j = 1, \ldots, 15$), the i-th vehicle measured ($i = 1, \ldots, N_j$, where N_j refers to the number of samples in the j-th trace) generates the following values: arrival time T_i, speed v_i and lane in the highway. In the following, t_i shall denote the inter-arrival time between the i-th and the $i-1$-th vehicles (i.e., $t_i = T_i - T_{i-1}$). Hence, the i-th measurement of the j-th trace comprises the tuple $\{v_i, t_i\}_j$ (we discard the sample corresponding to the first arriving vehicle of the trace as its inter-arrival time cannot be defined). Note that, according to the classification in [13], these traffic traces can be considered as free-flow non-rush hour traffic with moderate traffic volume and high speed.

Table 5.1: Data set collection.

Trace #	Highway [location in Fig. 5.1]	Veh/h	Avg. Speed [Km/h]	Lane
1	M-40 [D]	2038	102.64	L
2	A-6 [B]	1800	65.62	R
3	A-6 [B]	1744	68.20	R
4	A-6 [B]	1688	68.33	R
5	A-6 [B]	1522	85.88	L
6	A-6 [B]	1506	85.82	L
7	A-6 [A]	1466	89.13	C
8	A-6 [B]	1090	77.38	C
9	M-50 [E]	914	81.47	C
10	M-50 [E]	888	108.97	L
11	M-50 [E]	684	97.33	R
12	A-6 [B]	593	87.61	R
13	A-1 [C]	482	97.71	R
14	A-1 [C]	462	96.08	R
15	A-1 [C]	462	95.42	R

5.3 Analysis of the arrival process

In this section we start by analyzing the vehicle arrival process at a given lane in a highway for some representative cases of Table 5.1. We first analyze the vehicle inter-arrival times t_i, and then we use the travel speeds v_i to build a model that is able to both mimic the figures resulting from the traces and provide some insight on drivers' behavior.

5.3.1 Is the arrival process a Poisson process?

We first focus on the inter-arrival times between vehicles in a highway lane. This is, we are interested in analyzing whether the vehicle arrival process can be modeled by a Poisson process or not. Our motivation is that the classical "Poissonian assumption" is very common in the literature, partly because of its inherent analytical tractability (see, e.g., [21,91]). For the arrival process to be Poissonian, the inter-arrival times should follow an exponential random variable [92]. In order to characterize the time between vehicles t_i's, then, we analyze if their empirical distribution function matches the one from an exponential.

The cumulative distribution function (CDF) F of an exponential random variable of mean λ^{-1} is given by

$$F(t) = 1 - e^{-\lambda t}.$$

Therefore, for the sake of ease, it is often better to use the Complementary CDF (CCDF),

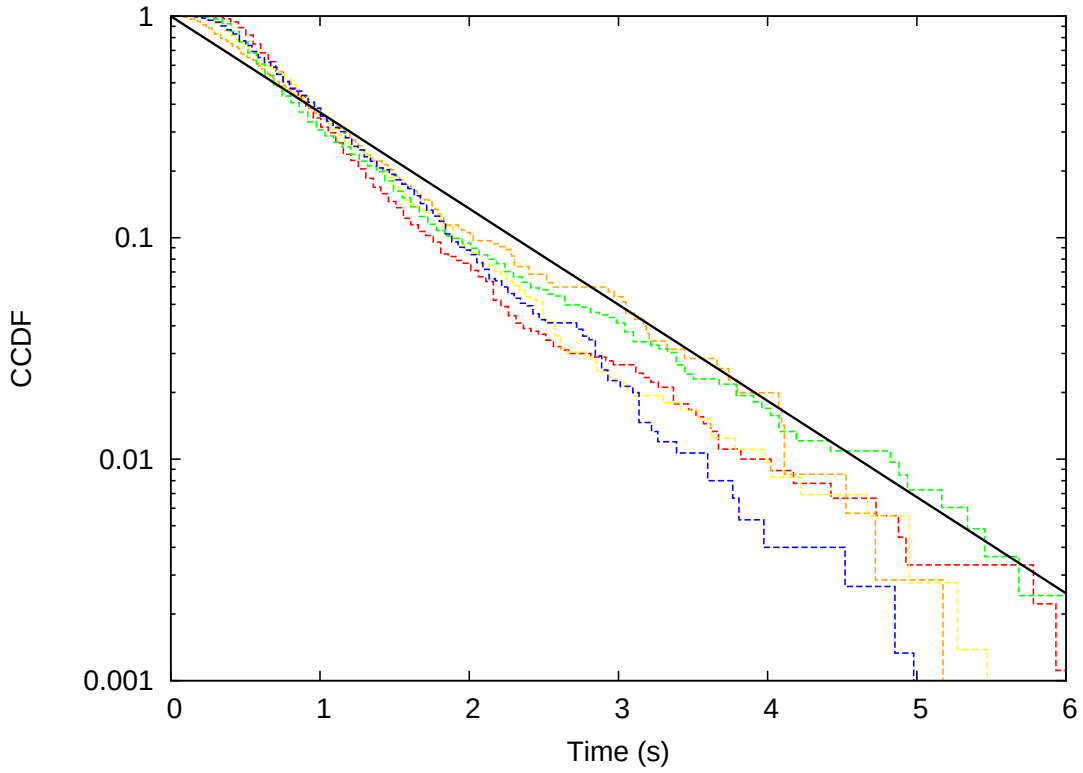


Figure 5.2: Empirical CCDFs of the normalized time between arrivals (dashed lines) and theoretical CCDF of an exponential distribution (solid line).

as its logarithm follows a straight line with slope $-\lambda$, i.e.,

$$\log(1 - F(t)) = -\lambda t.$$

In order to analyze the time between cars, we first normalize each of the traces from Table 5.1 by their respective means λ_i^{-1}, and then compute the empirical CCDF. We plot the resulting figures in logarithmic scale in Fig. 5.2 using dashed lines for Traces 1–5, along with the theoretical CCDF of an exponential random variable with $\lambda = 1$ using a solid line. It can be seen that the experimental data largely deviates from the exponential random variable, and therefore *we must reject the hypothesis that vehicle arrival times follow a Poisson distribution*. In order to have more statistically-based confidence on this result, we performed a Kolmogorov-Smirnoff (K-S) goodness of fit test [93] on the complete set of data, which rejects the hypothesis of exponential inter-arrival times with 95% confidence *in all the traces* of Table 5.1. These results are not shown for the sake of brevity.

In addition to the above finding, there are two other observations that can be derived from Fig. 5.2. First, inter-arrival times seem to be *shifted*, i.e., for each trace there seems to

be a minimum value of t before the CCDF starts decreasing. Given that we are analyzing the arrival process in a highway lane, this constitutes a quite expected result, as vehicles do not overlap. Second, the experiments also show that, after a time threshold around 2 and 3 s, the experimental CCDFs show an exponential decay (i.e., a straight line in the log CCDF plot), which suggests that the exponential behavior is still somehow present in the traces.

5.3.2 On the dependence between consecutive vehicles

Note that, had the CDF of the inter-arrival times resembled that of an exponential random variable, this would *not* have implied that the times between vehicles t_i are independent and identically distributed (i.i.d.) random variables. Indeed, it is often observed that cars travel together following some sort of *bursts*, even in non-congested highways. Actually, it is quite common to observe this phenomenon when a number of cars are following a slow moving vehicle (e.g., a truck) that cannot be overtaken easily. In these situations, the distances between vehicles are typically short, and *their traveling speeds are similar*, a behavior that introduces some inter-vehicle dependences in the arrival process.

In order to analyze if this behavior can be observed in our traces, we compute the correlation plot of the speed sequences v_i, resulting in Fig. 5.3. From the figure, we can see that (1) for some traces, there exists a noticeable correlation between speeds, and (2) such correlation in some cases is particularly high for consecutive cars (Lag=1).

Based on these results, we conjecture that, for some traces, some vehicles travel like bursts (i.e., relatively close together with similar speeds), while others are somehow isolated from the rest (i.e., with large distances between them and no dependences with the previous or next car). We will analyze with more detail this behavior in the next section.

5.3.3 Identifying bursts of vehicles

We claim that vehicles belonging to the same burst can be identified because they travel at a similar speed to that of the head of the burst, and the distances between them are typically short. To prove this assumption, we analyze the relationship between consecutive speeds, i.e., v_i and v_{i+1}, depending on the time distance t_{i+1} between them. We proceed as follows. Let δ denote a time threshold. Based on this δ, for each trace we construct two sets of points: in one set (denoted as $S_{\leq\delta}$) we put those speed pairs from cars that were relatively close, while in the other set (denoted as $S_{>\delta}$) we put the other pairs, i.e.,

$$(v_i, v_{i+1}) \in S_{\leq\delta} \text{ if } t_{i+1} \leq \delta,$$

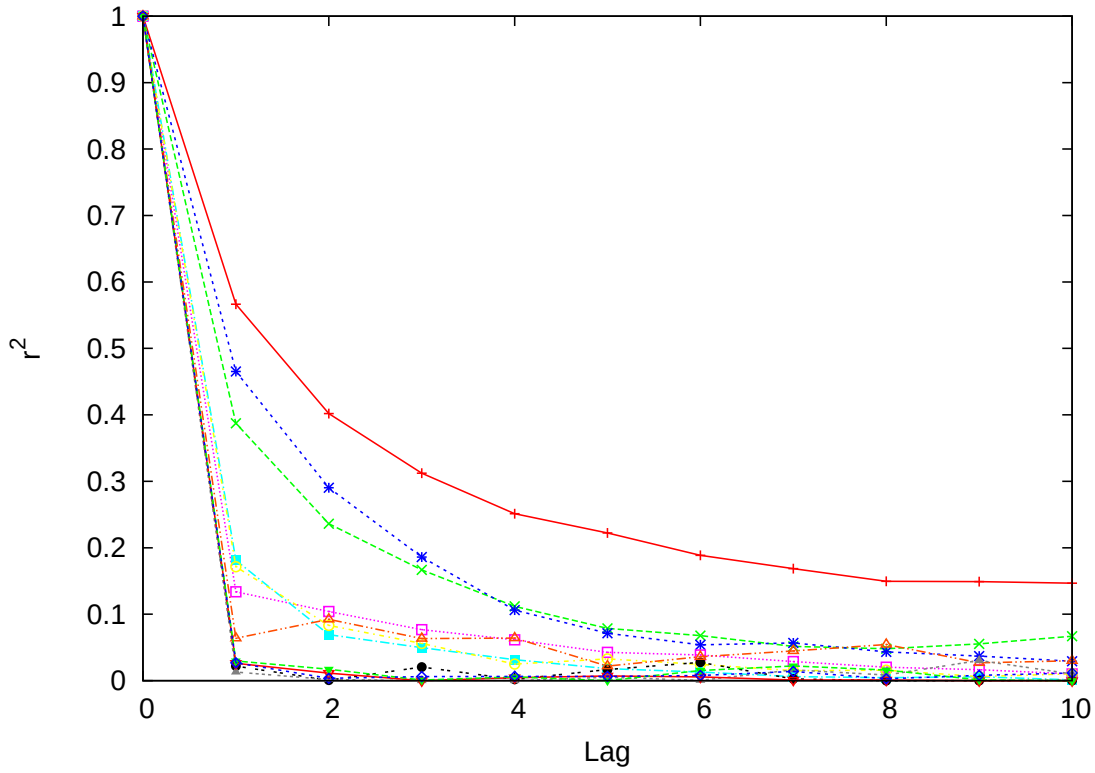


Figure 5.3: Correlation plot of the sequences of speeds v_i.

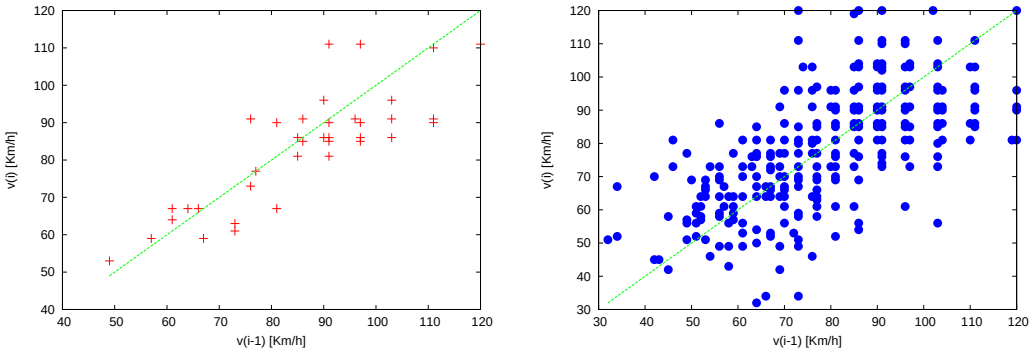
$$(v_i, v_{i+1}) \in S_{>\delta} \text{ if } t_{i+1} > \delta.$$

Fig. 5.4 shows the corresponding scatter plots of speeds for the case of Trace 9 with $\delta = 1\ s$. That is, for all pairs of vehicles that are separated by less than one second we plot in Fig. 5.4(a) their speeds vs. the speed of its predecessor, while we do the same for vehicles traveling more than one second apart in Fig. 5.4(b). It can be seen that, for the case of $S_{\leq\delta}$, consecutive vehicles have similar speeds (the scatter plot is clearly placed around the $y = x$ line), while for the case of $S_{>\delta}$ this behavior cannot be observed.

In order to quantify the speed similarity, we compute the Pearson product-moment correlation coefficient r for each data set. For the case of $S_{\leq\delta}$ the resulting value is $r = 0.84$, which can be considered as a clear indicator of a "strong" correlation[1], while for the case of $S_{>\delta}$ the resulting value is noticeable smaller, $r = 0.6$, although still non negligible[2]. Hence, *the time between vehicles and their speeds are not independent random variables* since, as shown, vehicles that are close together (at least, for the case of $t_i \leq 1\ s$) present high speed correlation.

[1]At least, according to the somehow arbitrary thresholds proposed in Cohen, J. (1988), Statistical power analysis for the behavioral sciences (2nd ed.) Hillsdale, NJ. : Lawrence Erlbaum Associates.

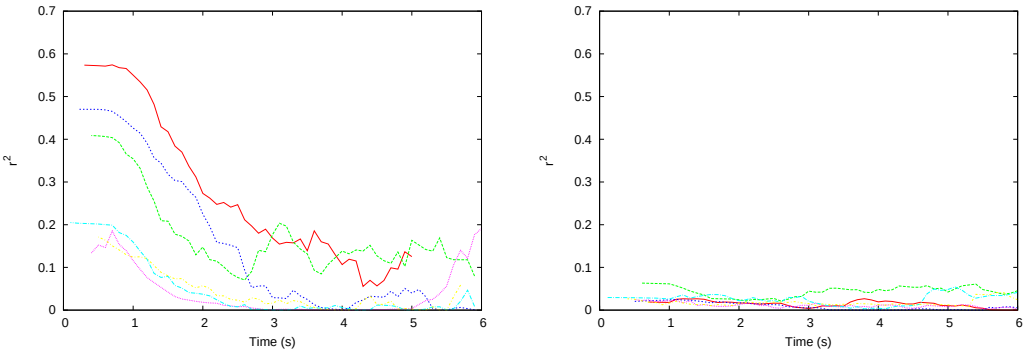
[2]Note that the traffic regulations necessarily introduce some correlation, as drivers tend to stay nearer to the speed limits. As we will see next, only in very sparse roads the correlation between speeds vanishes.



(a) Scatter plot of $S_{\leq\delta}$ (b) Scatter plot of $S_{>\delta}$

Figure 5.4: Scatter plot of the speed of consecutive vehicles (v_{i-1}, v_i) depending on the time distance t_i.

So far, we have only studied the case of $\delta = 1$ s for a given trace. In order to analyze the impact of this parameter, and whether we can observe a similar behavior in other traces or not, we perform a sweep on δ between 0.1 s and 6 s, and for each delta value we compute the correlation parameter r^2 for the data set $S_{\leq\delta}$. We plot the resulting values in Fig. 5.5, where the six traces with the highest density (in terms of vehicles per distance) are plot in the left figure, and the other traces are plot in the right figure. The results can be summarized as follows:



(a) Dense traces (b) Sparse traces

Figure 5.5: Correlation of the speed between consecutive vehicles when their relative distances is $t_i \leq \delta$.

- For the case of Fig. 5.5(a) (dense traces), when distances between vehicles are small, i.e., $\delta < 1.5$ s, there is a relatively large correlation between speeds.
- In the same figure, when $\delta \in (1.5, 2.5)$ there is a steep descent in r^2.
- Finally, for the case of dense traces, when $\delta > 2.5$ s the behavior of r is relatively flat and independent to changes on δ.

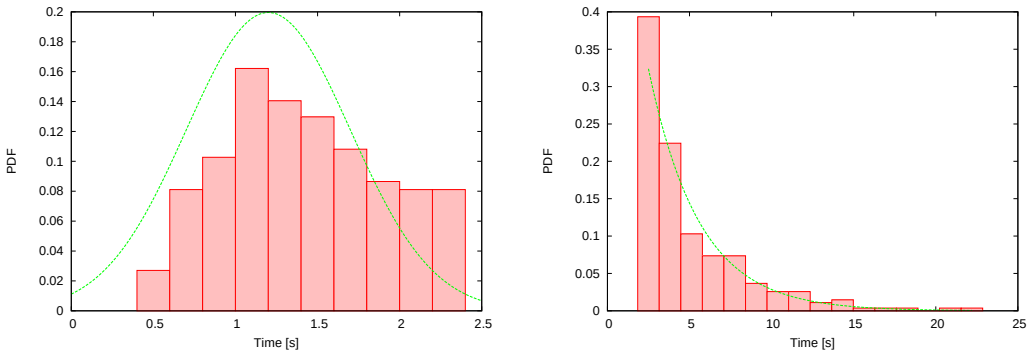
- On the other hand, for the case of Fig. 5.5(b) (small densities), there is no apparent speed correlation between consecutive vehicles regardless of the value of the parameter δ, as the behavior of r^2 is practically flat.

These results, derived from the use of formal statistical tools, have actually a very intuitive explanation: for the case of sparse lanes, drivers are quite unlikely to travel in bursts, and therefore there is no apparent correlation between the speeds of consecutive vehicles regardless of the time distances between them. On the other hand, for the case of dense lanes, vehicles are more likely to travel in bursts, and therefore the correlation values are large if the relative distances are small. However, still in this case there is a threshold value for this distance ($\delta \approx 2.5\ s$), that once crossed the relation between one vehicle and its predecessor vanishes.

Based on the above, we could label vehicles as *bursty* or *isolated*, depending on whether they are traveling together (with similar speeds) or not. In order to look inside the underlying distribution of the arrival process, we analyze the distribution of the inter-arrival time of Fig. 5.4, but dividing the t_i's into two sets according to the identified threshold $\delta_t = 2.5\ s$: in set τ_B we put those values below this δ_t threshold (i.e., the ones that correspond to bursty arrivals), while in set τ_I we put the values above the threshold (the ones corresponding to isolated arrivals),

$$t_i \in \tau_B \text{ if } t_i \leq 2.5\ s,$$

$$t_i \in \tau_I \text{ if } t_i > 2.5\ s.$$



(a) Set τ_B ($t_i \leq 2.5\ s$) (b) Set τ_I ($t_i > 2.5\ s$)

Figure 5.6: Histogram of the time between arrivals.

The resulting histograms for each set of data are plot in Fig. 5.6. According to these histograms, we further confirm the behavior that we have identified, i.e., there is a δ threshold that separates two very different types of arrivals. On one hand, we see when vehicles are relatively separated (τ_I) the empirical distribution seems to match the

probability distribution function (PDF) of an exponential random variable, which we plot in a continuous line in Fig. 5.6(b). This way, when vehicles are not traveling in bursts the arrival process could match a Poisson process. On the other side, however, we observe that when the time distances are relatively short (τ_B) the empirical distribution does not resemble that of an exponential random variable, but instead it is similar to the PDF of a Gaussian distributed random variable, which we plot in a continuous line in Fig. 5.6(a). Note that, because of the way we have performed the division into τ_I and τ_B, both random variables are truncated (more specifically, the normal variable is truncated between 0 and 2.5 s, while the exponential variable is shifted 2.5 s).

We try to provide an explanation for the observed results in the following. When vehicles are in a burst, i.e., the case of τ_B, drivers are "traveling together", and therefore the time between vehicles is very tied to the drivers' *reaction time*. We argue that this reaction time is around a few seconds, and it is reasonable to assume that it follows a Gaussian distribution (some human-related random variables that follow a normal distribution are, e.g., the distribution of height or the I.Q.). When vehicles are not bursty but isolated, the case of τ_I, the arrival process does indeed match a Poisson process, like other arrival process quite common in the literature (e.g., customers arrival in a queue, number of phone calls at a call center).

We claim, based on the above, that *the arrival process in a highway can be modeled using two different random variables: one exponential and one Gaussian.* However, we note that throughout the above analysis we have not derived a rigorous methodology to estimate the parameters of these variables, nor have assessed whether the resulting function will be able to mimic the observed experimental values or not. Furthermore, the value $\delta_t = 2.5\ s$ used to make the *hard* decision between exponential and normal can be seen as "upper bound" for all traces (as seen in Fig. 5.5), but not necessarily the value that fits best. In addition, the hard decision itself of classifying samples into τ_B and τ_I results in the truncation of the used variables, which could prevent a proper fitting of the model.

In the next section we first formally define the random variable that we claim can model the observed behavior; then, we will describe a methodology to obtain the parameters for this model; finally, we will assess the relative ability of the proposed variable to model the behavior observed in the real traces, as compared to previous proposals.

5.4 An exponential-gaussian mixture model of inter-arrival times

Based on the results from the previous section, we propose a Gaussian-exponential mixture model to characterize the vehicle inter-arrival times in a highway lane. More

specifically, we propose a weighted mixture of:

- A Gaussian random variable, to mimic the behavior caused by vehicles traveling together in bursts (as seen in τ_B).
- A shifted exponential variable, to model the Poisson arrival process of isolated cars (the one observed in τ_I).

This way, our proposed random variable $f_A(t)$ to model the time between vehicles arriving at a given point in a highway lane is formally defined as

$$f_A(t) = w_G \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} + w_E \lambda e^{-\lambda(t-m)},$$

with the following set Θ of parameters:

- μ and σ, which are the mean and standard deviation of the Gaussian random variable.
- λ and m, which are the rate and the shift of the exponential random variable.
- w_G and w_E, which are the the weights of each distribution (and therefore $w_N + w_E = 1$).

Our model is, then, an *unknown* mixture of a normal and an exponential random variables, whose parameters are also *unknown*. This is a well-known problem in statistics, namely computing the maximum likelihood (ML) of a set of parameters (the unobserved latent variables specified above) using a set of observed data (the inter-arrival times), than can be solved using an expectation-maximization (EM) algorithm [94, 95]. To estimate the required set of parameters, we implement the following EM-based algorithm using Matlab:

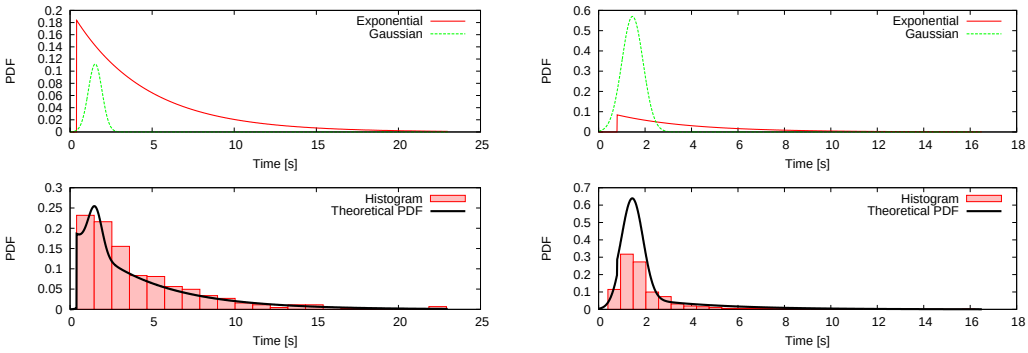
1. We initially set $m = 0$, i.e., we consider an unshifted exponential random variable.
2. We run the EM algorithm for either 200 iterations or until the algorithm converged to a set of $\{\mu, \sigma, \lambda, w_G\}$ parameters.
3. We compute the log-likelihood (denoted as LL) of the sequence of time between arrivals t_i with the obtained Θ, i.e.,

$$LL = \log(\mathcal{L}(t_i|\Theta)).$$

4. The above LL can be seen, after a run of the EM algorithm, as a function f_{EM} of the set of observations t_i and the shift parameter of the exponential m, i.e., $LL = f_{\text{EM}}(t_i, m)$ that, given m, assesses the likelihood of the data according to the obtained model. Therefore, f_{EM} quantifies the goodness of a given value of m.

5. We finally perform a sweep on the m parameter from 0 to 3 s in steps of 0.05 s, to obtain the value that maximizes the LL (i.e., the function f_{EM} defined above).

In order to illustrate the operation of the EM algorithm, we plot in Fig. 5.7 (above) the weighted components of the exponential and the normal random variables for the case of trace 10 (a sparse trace) and trace 3 (a dense trace). We also plot in Fig. 5.7 (below) the histograms of the empirical traces, along with the resulting theoretical PDFs from the operation of our algorithm. It can be seen in Fig. 5.7(a) that, for the case of a sparse trace, the normal component is relatively small and most of the observed behavior can be captured using a shifted exponential variable. On the other hand, for the case of a dense trace, Fig. 5.7(b) shows that the observed values are most likely to be modeled with a Gaussian variable, while the relative component of the exponential variable (in this case shifted) is quite small.



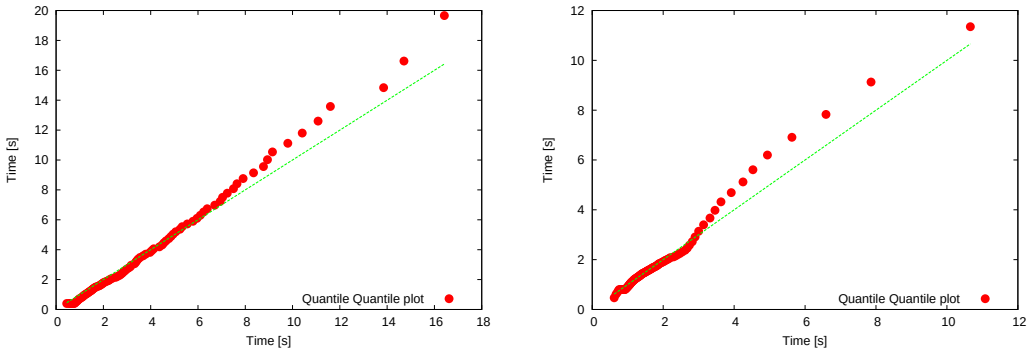
(a) Sparse trace, # 10

(b) Dense trace, # 3

Figure 5.7: Components of the Gaussian-exponential mixture and resulting PDF for two traces.

In order to obtain a better representation of the matching between the empirical and the obtained distributions, we plot in Fig. 5.8 the resulting Q-Q plot for the same two traces used in the previous figure. The results visually confirm that for both for the case of dense traffic and the case of sparse traffic, our proposed model as described by $f_A(t)$ is able to mimic the observed values of the distribution of inter-arrival times.

We run our EM algorithm on the 15 traces considered in this study, with the resulting values shown in Table 5.2. It is interesting to observe that these figures, derived from a numerical search as defined by the algorithm, are indeed quite intuitive and further confirm the results from the analysis of Section 5.3 and the proposed methodology. Indeed, we can



(a) Sparse trace (b) Dense trace

Figure 5.8: Q-Q plot for the resulting theoretical and empirical distributions.

Table 5.2: Resulting values of Θ for the traces of Table 5.1.

Trace #	w_N	w_E	μ	σ	λ	m
1	0.58	0.42	1.44	0.46	0.36	0.50
2	0.65	0.45	1.45	0.35	0.31	0.80
3	0.63	0.37	1.50	0.50	0.31	0.80
4	0.50	0.50	1.06	0.34	0.49	0.65
5	0.37	0.63	1.77	0.66	0.37	0.50
6	0.30	0.70	1.87	0.64	0.37	0.40
7	0.02	0.98	0.85	0.09	0.12	0.50
8	0.01	0.99	1.11	0.48	0.13	0.55
9	0.05	0.95	2.44	0.87	0.13	1.00
10	0.04	0.96	0.73	0.10	0.16	0.40
11	0.11	0.89	1.52	0.41	0.11	0.65
12	0.09	0.91	1.08	0.41	0.17	0.90
13	0.14	1.04	0.30	0.86	0.22	1.00
14	0.21	1.57	0.58	0.79	0.26	1.10
15	0.30	1.84	0.69	0.70	0.38	1.00

emphasize the following three results: *i*) the relative weight of the exponential variable w_E increases as the density of vehicles decrease (note that the results are presented in decreasing order of densities, see Table 5.1); *ii*) by observing the values of $\mu + \sigma$, it is clear that distances between vehicles when they travel in bursts are typically well below the identified threshold of 2.5 s; *iii*) in all cases, the shift parameter m of the exponential random variable is placed between the two extreme values used in our numerical search.

Finally, in order to assess the ability of the proposed model to capture the observed behavior, comparing it against previous work, we proceed as follows[3]. For each of the traces in Table 5.1 we compute the parameters for three different models:

[3]Note that in Section 5.3 we performed a K-S test to reject the hypothesis that the inter-arrival times followed an exponential random variable with mean one. However, for the case of distributions with estimated parameters, the values used for the standard K-S test are invalid [96].

Table 5.3: Resulting log-likelihood of the three models considered for the inter-arrival time between vehicles.

Trace #	LL_{EM}	LL_E	λ^{-1}	LL_L	μ	σ
1	**-1.29**	-1.59	1.76	-1.32	0.32	0.64
2	**-1.18**	-1.51	1.99	**-1.18**	0.52	0.53
3	**-1.15**	-1.50	2.06	-1.19	1.05	0.85
4	**-1.18**	-1.47	2.12	**-1.18**	0.55	0.57
5	-1.24	-1.41	2.36	**-1.23**	0.66	0.62
6	-1.22	-1.39	2.45	**-1.20**	0.71	0.61
7	**-1.08**	-1.15	3.49	**-1.08**	0.96	0.75
8	**-1.12**	-1.19	3.29	**-1.12**	0.90	0.76
9	**-1.02**	-1.08	3.92	**-1.02**	1.07	0.76
10	**-1.01**	-1.06	4.05	-1.02	1.05	0.85
11	**-0.87**	-0.90	5.24	-0.89	1.28	0.89
12	**-0.80**	-0.84	5.85	-0.81	1.49	0.78
13	**-0.66**	-0.69	7.31	**-0.66**	1.71	0.76
14	**-0.69**	-0.72	7.37	-0.70	1.73	0.77
15	**-0.68**	-0.70	7.78	-0.69	1.72	0.84

- The proposed mixture, with the set of parameters Θ as given by our EM algorithm.
- An exponential random variable (as proposed in [20]), computing its ML rate λ as given by Matlab.
- A log-normal distribution (as proposed in [21]), with its ML parameters μ and σ as computed by Matlab for a given set t_i.

For each of the traces, we compute the log-likelihood of the observed values for these three model, denoted as LL_{EM}, LL_E and LL_L, respectively, with the obtained results presented in Table 5.3. We also show in the table the estimated parameters (λ^{-1}, μ and σ) for the other exponential and log-normal models. For each trace j we mark with a bold font the largest value of LL out of the three obtained likelihoods (in case of a numerical tie, we mark both numbers). The results can be summarized as follows:

- Our model achieves the highest performance, as it provides the largest likelihood in 86.6 % of the cases (13 out of the 15 traces).
- The log-normal distribution is the second best model, being able to outperform our model in 2 out of the 15 traces.
- Finally, the exponential random variable is never able to match the likelihood of the other two alternatives.

Based on these results, we conclude that our model is the best suited to mimic the behavior observed in a highway lane. In addition, note that the model not only provides a

better "numerical" performance than those of [21,20], but it also provides valuable insight on drivers' behavior, some of which have been neglected so far, e.g., the dependence between consecutive arrivals. As we describe in the next section, part of our future work consists on revisiting this assumption on the i.i.d. of the random variable modeling the vehicles' arrivals.

5.5 Summary

In this chapter, we analyzed the inter-arrival times of vehicles from a set of traces collected at different locations in the city of Madrid. Two types of cars can be identified: those traveling together at short distances with similar speeds (bursty behavior), and those traveling at large distances with uncorrelated speeds (isolated ones). While the first category can be modeled with a Gaussian distribution, the second one exhibit an exponential behavior. Based on this findings, we proposed a Gaussian-exponential mixture model that characterizes the two groups. Finally, we compared the accuracy of our model with previously proposed ones, showing its validity.

Chapter 6

Traffic monitoring and short-term prediction

Traffic congestion is a major economic and social problem of the modern society. While social aspects are difficult to quantify, the economical impact is easier to estimate. The 2011 Urban Mobility Report published by Texas University [1] claims that the total cost for traffic jams in the U.S. in that year was $100 billion. Traffic congestion does not only impact on factors like fuel consumption or increased pollution, but also on loss of working hours. Critical roads are constantly monitored to provide a quick picture of the traffic congestion level, and warn drivers in case of congestion, e.g., using broadcast techniques such as the FM radio (via the Traffic Message Channel TMC or even by the radio-speakers) or traffic panels. There are three critical aspects in traffic management systems: i) traffic monitoring, ii) congestion detection/prediction, and iii) efficient information spreading. Existing traffic monitoring techniques present several drawbacks, as they are not flexible (measurement points cannot be easily moved) and are very expensive to deploy and maintain. Only congestion detection can be performed, as prediction tech-

[1] http://tti.tamu.edu/documents/mobility-report-2011.pdf

niques require more information than current deployed monitoring systems can provide. Finally, there is no information spreading mechanism that can efficiently warn all drivers. During the last few years, vehicular communications have been extensively researched with the aim of enabling vehicles to exchange information among them and also with the infrastructure (e.g., the Internet). While the main goal has been traffic safety, traffic efficiency and information/entertainment services have also been addressed. There are two main paradigms considered to enable vehicular communications: the use of traditional cellular networks, and the use of vehicular ad-hoc networks (VANETs). While the former is a mature technology that is widely available, its performance in vehicular scenarios and its associated costs are the main deployment barriers. On the other hand, VANETs are expected to be a complementary technology, enabling vehicles to share information in real time, especially within a limited geographical region. Throughout this chapter we will describe two proposals for the efficient traffic monitoring and prediction based on vehicular communications. The first proposal is VIL (Virtual Induction Loop), a mechanism that uses vehicular communications to gather information about traffic variables (*Flow* and *Speed*) in a cheaper way than currently adopted techniques. The second proposal is ABEONA, an algorithm based on vehicle-to-vehicle communications for the monitoring and the short term prediction of vehicular traffic congestion.

6.1 Current monitoring techniques

The current state of the art in the field of traffic monitoring is well divided into three main categories of solutions: floating cars, cameras and induction loops. The floating cars techniques, firstly proposed by Kerner *et al.* in [97], consists in anonymously tracking cellular devices that passengers take with them and hence obtain their speed profiles during the time. This technology is nowadays used by commercial products such as TomTom's HQ Routes [98].

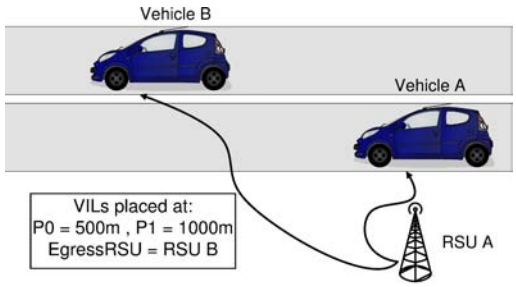
Video Cameras have been used for remote surveillance for many years and, with the improvements in the data elaboration capabilities, they are currently used to monitor the road traffic states. Each vehicle is uniquely identified by its license plate and then tracked over a defined stretch of road.

The most common and reliable technology used to collect traffic data is induction loops [99]. These loops are embedded in roadways in a square formation that generates a magnetic field. Magnetic loops count the number of vehicles, and collect some information for each vehicle traversing the loop such as the instant of time, speed, lane and type of vehicle. This technology has been widely deployed all over the world in the last decades. However, the implementation and maintenance costs are expensive [100].

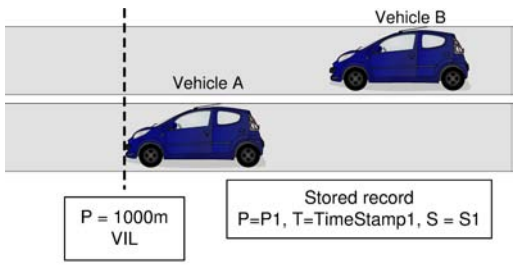
The three mentioned techniques have some flaws. Cameras and induction loops are expensive and difficult to install, as they require works to settle the fixed infrastructure.

Moreover they do not allow flexible configurations, as they can just monitor a well defined zone.
Floating cars are a scalable solution but the information they provide might be coarse. Beside this, they just provide information about speed or travel times, while more complex data can be desirable (i.e. vehicular flow or vehicle types).
The recent advances in vehicular communications open the way for new solutions based on this paradigm. By effectively combining infrastructure to Vehicle and Vehicle to infrastructure communications (I2V and V2I) with common automotive devices like GPS, transit data can be collected with sufficient precision, at least the same as the previous approaches.
For example, Garelli *et. al* [101] proposed a distributed sampling system to estimate the vehicular density within a certain zone. In [102] Skordylis et.al. introduce a more complex framework, used to probe the vehicular speed into a well defined area. The retrieved information is then spread throughout the VANET employing different V2V based data dissemination techniques.
The ETSI Technical Committee for Intelligent Transport Systems (ETSI TC ITS[2]) is currently developing a set of protocols and algorithms that define a harmonized communication system for European ITS applications. In the standardized Communication Architecture [103] are defined different types of ITS stations with the capability of communicating between them using different access technologies. In particular, the IEEE 802.11p [104] at the 5.9GHz band, an amendment to the 802.11 protocol especially tailored for vehicular networking, is one of these access technologies.
Through the continuous exchange of messages between vehicles (Vehicle-to-Vehicle or V2V communications), and between vehicles and infrastructure nodes (Vehicle-to-Infrastructure or V2I communications), real-time information about current road traffic conditions are cooperatively collected and shared.
ITS stations, vehicles and RSUs, periodically broadcast secure Cooperative Awareness Messages (CAM) [105] to neighboring ITS stations that are located within a single hop distance. CAMs are distributed using 802.11p and provide information of presence, positions as well as basic status of communicating ITS stations (e.g., current acceleration, occupancy of the vehicle, current heading of the vehicle,...). The periodic exchange of CAMs helps ITS stations to support higher layer protocols, and cooperative applications, including road safety and traffic efficiency applications. In the next section we propose VIL, a flexible and simple solution that incurs in low communication overhead. VIL service could be easily deployed using Context-Aware Messages (CAM) [105], standardized by ETSI to improve safety and traffic efficiency in roads. The basic idea behind VIL is to define *virtual loops* whose position is advertised by roadside units (RSUs) along the road. Vehicles, which are supposed to be equipped with GPS, detect when they are traversing

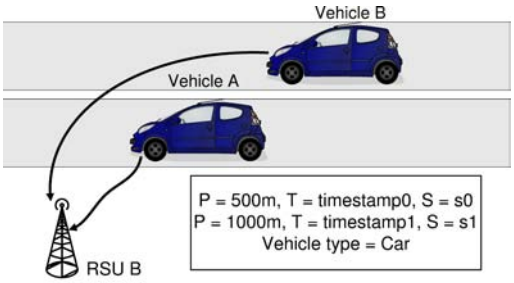
[2] `http://portal.etsi.org/its/its_tor.asp`



(a) The *ingress RSU* announce the virtual loop



(b) Vehicles record their sample



(c) Vehicles send gathered data to an *egress RSU*

Figure 6.1: The measurement steps

one of these virtual loops and store its state at that moment (e.g., time, speed and, lane). As soon as the vehicle gets close to a RSU, it delivers this recorded information to the RSU, which in turn collects information related to several vehicles and VILs, and sends it to the central ITS station. To summarize, with the deployment of VIL, more roads than those currently equipped with a monitoring infrastructure, mainly induction loops, could be easily observed without requiring additional costs (i.e., nowadays only major urban cities can afford monitoring infrastructure).

6.2 Monitoring traffic using Virtual Induction Loops

A virtual induction loop (VIL) is a virtual line playing the same role as a legacy magnetic induction loop. This way, VIL service gathers real time information of the vehicles traversing this virtual line.

VIL is a traffic efficiency service that makes use of existing secure CAM messages sent by RSUs and vehicles [105]. The full operation is shown in Figure 5.1. First, the *ingress RSU* announces in its CAM messages the positions of the virtual induction loops present in the stretch of road under its influence. These CAM messages also include information on the identities of the *egress RSUs*, that is, the nearest RSUs the vehicle may find in its way, after traversing the virtual loops in this stretch, depending on the vehicle's trajectory (see Figure 6.1(a)). It is important to note two details: first, for a particular stretch

Simulation framework	OMNeT++, Veins and SUMO
Wireless Device	802.11g @ 6Mb/s
Channel Model	Pathloss with channel fading
Monitored zone length	1500m
VIL position [m]	500, 1000
RSU broadcasting interval [s]	uniform(0.75,1.25)

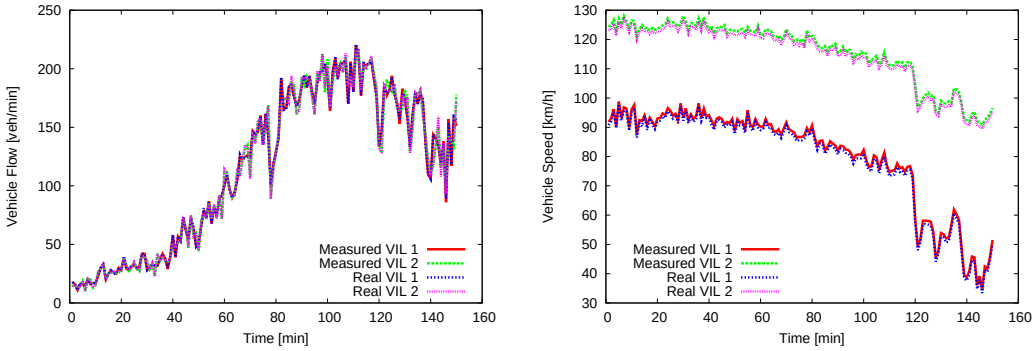
Table 6.1: Simulation settings

apart from having several *egress RSUs*, it is also possible to have several *ingress RSUs* (i.e., announcing the virtual loops) and second, a RSU may play the role of *egress RSUs* for a stretch and the role of *ingress RSUs* for the next stretch.
On the other hand, GPS devices are now cheap enough to be included in vehicles configurations, as part of the on-board computer system. They supply two valuable pieces of information: the current position and the current time in UTC form. From these two variables it is straightforward to calculate the current speed as well. Thus, for each announced virtual loop a vehicle encounters during its transit within the monitored stretch (see Figure 6.1(b)), it records its state when traversing the virtual loop (e.g., timestamp, speed, lane, etc.).
Then, the vehicle is aware of being in the radio coverage of an *egress RSU* once it receives a CAM message broadcast by an *egress RSU* (i.e., the CAM message includes the identity of this *egress RSU*). From that moment and while being in the RSU's coverage area, CAM messages sent by the vehicle also include the information gathered when it traversed each of the virtual loops in the last stretch (see Figure 6.1(c)). In addition to the basic information (e.g., timestamp and speed), the vehicles can also upload another useful data such as the lane or their category (e.g., traffic control centers often want to know the percentage of heavy lorries). Finally, *egress RSUs* send the gathered data to the traffic control center. Once there, the information on traffic conditions is elaborated and eventually redistributed to drivers.

6.2.1 Evaluation

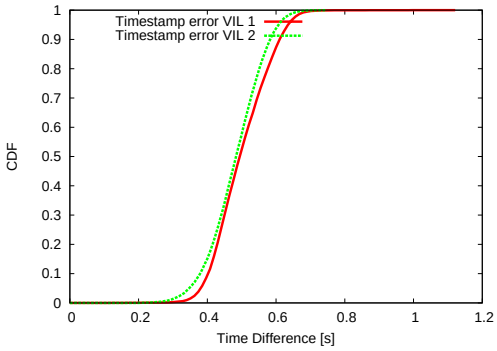
We chose to evaluate our proposal by simulation. In order to achieve realistic results we built our software using the Veins framework[3] for OMNeT++ [4] [106], using a methodology similar to the one employed for the solution described in the previous chapters. The simulation parameters are summarized in Table 6.1. The goal of our simulator is to realistically investigate the effectiveness of VIL. To this aim, we consider both realistic wireless conditions and realistic vehicular patterns. To achieve the second goal we fed our simulator using real vehicular traces captured by magnetic induction loops. The data

[3] `veins.car2x.org`
[4] `www.omnetpp.org`



(a) Vehicular flow

(b) Average speed



(c) Error CDF

Figure 6.2: Simulation results

was kindly provided to us by the Madrid City Council and was collected along the M-30 orbital motorway. The trace is 2.5-hour long (from 8:00 am to 10:30 am) and it contains data of about 17.000 vehicles.

The monitored stretch is 1500m long and it is a 6-lane single carriageway road. There are has one *ingress RSUs*, one *egress RSUs* and two virtual loops in the stretch.

Aiming at introducing realistic noise to the experiment, we introduce a Gaussian white noise for the two position components (both latitude and longitude) with a parameter $\sigma = 4m$ what is a fair error assumption for highways under clear conditions.

In our experiments the refresh frequency of the GPS is $f_r = 1Hz$ (update frequency of its position and current time) which is an additional source of error in the experiments. However, this refresh frequency is considered realistic taking into account the features of the current commercial GPS devices.

For the first 500m, where the first VIL is placed, vehicles' speed is forced to be close (+/- 2.5 m/s) to the departure one, then vehicles are allowed to increase their speed, up to the speed limit fixed at 130 km/h.

Figure 5.2 depicts the obtained simulation results. As it is topically done, the vehicular metrics: flow and mean speed are summarized into coarser bins, in our case one-minute sized. In Figures 6.2(a) and 6.2(b), we compare the flow and the mean speed obtained

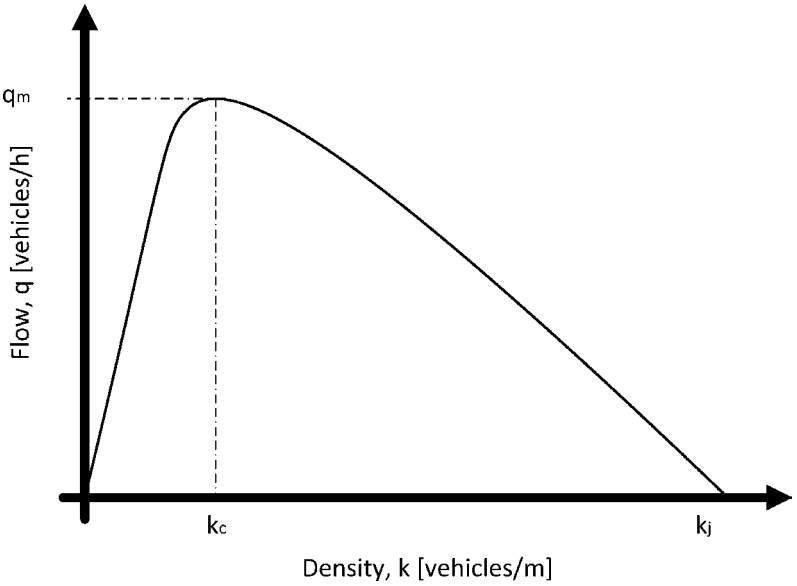


Figure 6.3: A flow-density curve

by VIL at the *egress RSU* with the real ones from the vehicular traces. It can be noticed that despite the inaccuracy in latitude and longitude, and the errors introduced by the refresh frequency, the gap between the two curves is almost negligible.
Figure 6.2(c) shows in a CDF, the difference (in absolute value) between the real crossing timestamps and the ones measured by the VIL system. As it can be observed, almost all the crossing timestamps obtained by VIL fall into ± 0.8 s of difference with the real ones.
A relevant parameter to be taken into account is the control overhead that the system introduces over the air. Due to the use of existing CAMs, no new messages are introduced by VIL system. However the size of CAM messages is increased with the VIL information. In particular, CAMs broadcast by *ingress RSUs* include the location of the virtual loops in the area under their influence (around 8 bytes per virtual loop) and the identity of the nearest *egress RSUs* (8 bytes per RSU). Considering the CAM announce 3 virtual loops and 5 *egress RSUs* the increase in the CAM size is 64 bytes. As for, the CAM messages sent by the vehicles when they are in the coverage area of an *egress RSU*, the size of CAMs is increased by 6 bytes (4 bytes the timestamp and 2 bytes the speed) for each virtual loop the vehicle has crossed. So, the increase in the CAM size can be considered of negligible impact.

6.3 Distributed traffic congestion prediction

A Traffic Management System should be able to detect or predict traffic congestion situation [107, 108]. While the detection of traffic jams is effectively carried out by monitoring the vehicles' average speed, the prediction needs a more complex analysis.

We next present a solution called ABEONA which shows the feasibility of a VANET-based traffic monitoring and prediction service. The key selling points of the proposed solution are the following:

- ABEONA is a flexible and cost-feasible solution that allows monitoring traffic conditions in real-time.
- ABEONA is based on a cooperative and distributed knowledge of traffic conditions (average vehicular speed, flow and density).
- using this knowledge, ABEONA is able to forecast short-term traffic conditions (with a window of 15 to 30 minutes).

The aforementioned points make possible to warn other drivers of a future forecasted traffic congestion event, so an alternative route can be planned, reducing the overall travel time.
Along this section, we first overview the classical theory used to model vehicular traffic including a validation analysis using real traces from the city of Madrid. Then, we introduce the theoretical basis for traffic density and flow estimation and prediction.

6.3.1 Modeling vehicular traffic

Vehicular traffic has been investigated since the 50s [109, 90], and we can now find micro- , meso- and macroscopic models available, many of them reviewed in [110]. In this article we focus on the macroscopic ones, which treat each vehicle as a particle that moves along a one-dimensional trajectory, taking their basis from fluid dynamic and hydrodynamic models. Basically, each vehicle is a particle that has to go from an origin to a destination, coordinated with other particles (vehicles) to avoid colliding with them. These interactions can eventually lead to the creation of a traffic jam [111]. All of these models are based on drivers' behavior and analyze three fundamental factors: speed v, flow q and density k. The vehicular flow [veh/h] represents the number of vehicles entering (and leaving) a stretch of road in a time window. The vehicular density [veh/m] is the number of vehicles travelling in that stretch of road at a given time; while by speed we mean the average speed [km/h]. Figure 6.3 shows the fundamental relation between flow and density. Parameters q and k are directly proportional until reaching a critical density k_c that corresponds to a maximum flow q_m. For lower densities, vehicles are travelling under free-flow conditions, while for higher densities, vehicles behavior depends on their neighbors. For values of the density k higher than k_c, q and k are inversely proportional. When k is equal to the jam density k_j, q is equal to 0 (i.e., vehicles are not moving). The vehicular speed v fixes the shape of the relation function. This relation comes from the

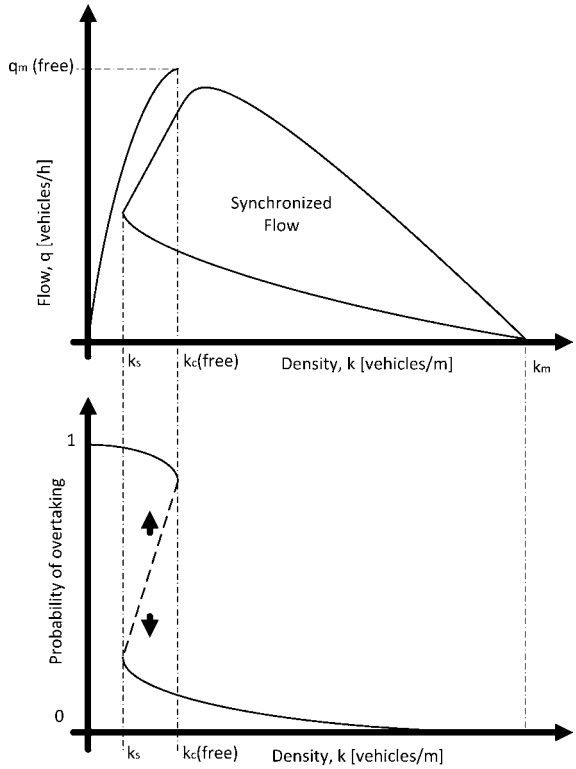
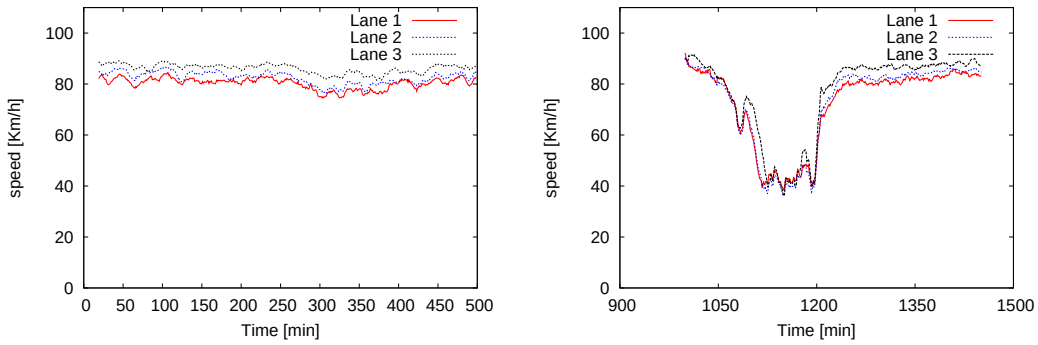


Figure 6.4: Phase transitions diagram

law of conservation of vehicles, firstly stated by Greenshields in [112].
Among the many theories proposed since the 70s analyzing the interaction of vehicles, one of the most accepted studies is the "Three-phase traffic theory" firstly proposed by Kerner in [22] and then extended by the same author in [113] that classifies traffic state into three categories:

- *Free flow*: when the vehicular density is reasonably low and drivers can easily keep their desired speed and overtake slower vehicles. The average speed of each lane is independent.
- *Synchronized traffic*: when the density is such that overtaking is not easy, so that the average speed of each lane sharply decreases and synchronizes. Drivers behavior is influenced by other drivers.
- *Traffic jam*: when vehicles are stopped or follow a stop and go pattern.

Kerner also studied the transitions between the different phases, concluding that they can be modeled as first-order phase transitions (like the gas-liquid-solid transitions of a thermodynamic system, according to the pressure and temperature conditions). This basically means that a change (even small) in one of the three variables (density k, flow q or

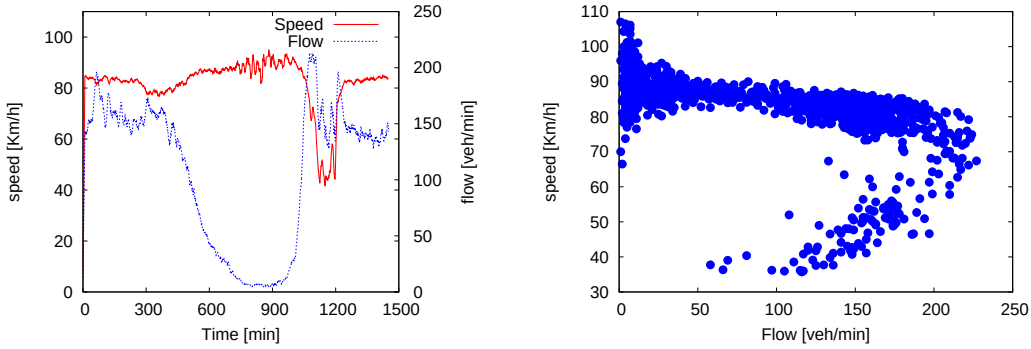


(a) Free flow traffic (b) Traffic jam

Figure 6.5: Average speed of different lanes

speed v) can lead to a state change, with consequently possible (even high) oscillations of the other variables. Figure 6.4 shows an example of a "Free flow $\rightarrow$ Synchronized traffic" transition. The phase transition is (mostly) related to the easiness of overtaking that the drivers experience. When the road is in "Free flow" state, the probability of overtaking is quite high. When the road is congested this probability is close to zero, as vehicles are travelling at similar speed (if not stopped) and it is difficult to find room for a possible overtake. The same overtaking probability can appear for different values of density within the window k_s and $k_c(free)$. While in this "undefined area", a perturbation of one of the three variables can result in a different outcome: for example, a wrong maneuvering can lead to the other drivers to temporarily slow down and then come back to "Free flow" state, while a road suffering a continued increase in the traffic demand will end up in the transition from the "Free flow" to the "Synchronized traffic'" state. From a user-centric viewpoint, the most interesting variable is the speed, as the lower the speed is, the higher the travel time is. In "Free flow" conditions, vehicles travel at a speed that depends on the road state, the vehicle type and the weather conditions, and it is ultimately limited by the roads speed limit. Interestingly, the speed is the variable that has the least linear behavior, being an abrupt decrease possible when traffic experiences a change from "Free flow" to "Synchronized traffic" state.

In order to assess the correctness of Kerner's model we have used real traffic measurements from the city of Madrid, obtained from its own monitoring infrastructure (mainly composed by induction loops and cameras). We have used a 24-hour trace collected by an induction loop placed at the km 4.4 of the M30 orbital motorway, northbound, which at this point has six lanes available and a speed limit of 90km/h. Figure 6.5(a) shows the relation between the average speed of different lanes in "Free flow" state, while Figure 6.5(b) shows the same for "Synchronized traffic" state. The measurements confirm Kerner's theory, as in "Free flow" state, average speeds are independent, while in "Synchronized traffic" state they tend to converge to a unique value (this indicating a



(a) Flow and Speed trends

(b) Speed depending on flow level

Figure 6.6: Traffic variables from real measurements

decreased capacity of overtaking). Similarly, Figures 6.6(a) and 6.6(b) show the complete relations of the two variables that can be directly captured by an induction loop: flow and speed. Figure 6.6(a) depicts the complete speed and flow time series and it can be observed that the transition between Free flow and Synchronized traffic states is a first order transition that involves more than one variable. As already shown in Figure 6.4 the transition is characterized by an hysteresis that can also be noticed in Figure 6.6(b): for the same flow rate q=150 veh/min, values around 40-50 km/h and around 90 km/h are possible.

We can conclude from this section that the analysis of flow, density and speed variables allows to characterize the traffic state. Therefore, the next step is to come up with a mechanism that enables its efficient monitoring in order to be able to predict short-term traffic conditions.

6.4 ABEONA: VANET-assisted traffic forecasting

As highlighted previously, in order to correctly understand the traffic state, all the three variables (speed, density and flow) have to be monitored, as a perturbation in one of them can trigger a transition. However, it is difficult to measure or estimate density using the three classic monitoring techniques revised above. While the floating cars technique is a cheap mechanism, data from every vehicle cannot be easily obtained and it is not straightforward to extend it for monitoring flow and density. Alternatively, cameras can measure all the three variables, but they have a high deployment and maintenance costs, and introduce privacy issues. Last but not least, induction loops, although cheaper than cameras, can only measure two of the three variables: speed and flow.

The forecast of the vehicular traffic demand has been researched by physicist and civil engineers, with two main areas of interest: long-term and short-term prediction.

Long-term forecasting exploits the so-called seasonality property of vehicular traffic: people tend to move following regular patterns [82]. The threshold between long term and short-term prediction is usually set around 15-20 minutes. Predictions that aim at longer times are usually considered as long-term.
In [114], Kanoh *et al.* propose and evaluate a neural-network based solution. A different approach, using the analysis of the flow time series, was presented by Thomas *et al.* [115]. Both of them are validated using real measurements and claim that, if used in conjunction with intelligent transportation systems, prediction techniques can help optimizing flows and other aspects of road conditions (e.g., waiting times at traffic lights). These two proposals use a composite algorithm, comprising a section for long term forecasting and a section for short-term. However, while the analysis for long term forecasting is more complex, the short-term part is usually easier, depending on the conditions from the past 15-20 minutes.
Chrobok *et al.* propose in [14] different methods for short-term traffic forecasting, and they find out that even a very simple forecasting method like linear prediction can achieve good results in estimating the short-term traffic demand, allowing to infer the future traffic.
We next describe our traffic congestion prediction framework, called ABEONA, which is composed of two main aspects: i) collaborative flow and density estimation, and ii) short-term prediction.

6.4.1 Algorithm overview

ABEONA makes use of V2V communications to exchange data among vehicles which are used as distributed "sensors" and also of V2I communications, as certain events (such as the prediction of traffic congestion) might need to be made known to the infrastructure, for example to a traffic control center that can then warn other drivers. The work presented in this chapter does not specifically address how predicted events are disseminated, but focuses on how the prediction is actually performed.
ABEONA predicts future traffic state based on current and past monitored conditions. In order to do so, ABEONA needs a spatial-temporal reference. Assuming that vehicles are equipped with a GPS device, the time reference can be easily retrieved from it with an accuracy of 1s. Each vehicle groups the monitored data in 1-minute sets, identified by an epoch value (i.e. the current minute) called *EpochId*. The spatial reference is provided by dividing the road into different regions, each of them identified by a unique *RoadId* identifier, and including this information into an enriched version of the digital map used by the in-vehicle navigation system. The knowledge of the *RoadId* and *EpochId* values allows the use of time persistent content distribution techniques, firstly proposed

Figure 6.7: An example of road region division

by Leontiadis *et al.* in [116]. ABEONAs basic idea consists in the following: each vehicle periodically broadcasts the sensed *Flow* and *Density* within its region (identified by the *RoadId*), not only about the current epoch, but also about the most recent past ones. This ensures that the information required at each road region to forecast future traffic conditions remains available at the area. Vehicles traveling within the region cooperative store and share this information. An example is shown in Figure 6.7, where vehicles are only broadcasting data regarding *RoadId A* while within this road region. Once a vehicle leaves *RoadId A* and enters *RoadId B*, it stops broadcasting *RoadId A* related information and waits to receive information about *RoadId B* sent by other vehicles that are currently in that region. As mentioned before, the messages exchanged by vehicles belonging to the same road region include both data related to the current epoch and a historical record. We next describe this in more detail.

6.4.2 Current epoch data management

A first goal of ABEONA is to correctly evaluate the flow and the density for the current *EpochId*. In order to do so, there are three key parameters that need to be exchanged by cars: a *VehicleId* that uniquely identifies each vehicle, and the *Latitude,*

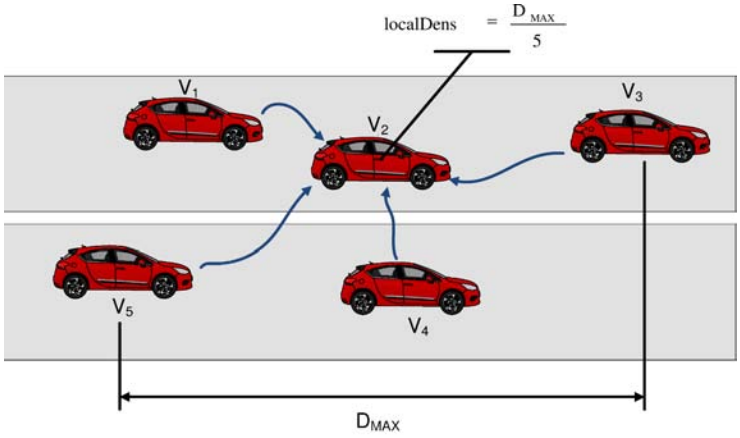


Figure 6.8: A density estimation example

Longitude pair, which indicates the vehicle current position. Vehicles include this information in ABEONA beacons that are broadcast periodically.

6.4.2.1 Density Estimation

The density of a given road region can be independently estimated by each vehicle using information included in the beacons received from other vehicles. Each vehicle, by listening to these beacons, can build an updated map of the surrounding vehicles. A local estimation of the current density can be inferred by taking the distance between the pair of neighbors that are the furthest away from each other and dividing it by the current number of neighbors (see Figure 6.8. This value, which is calculated independently by each vehicle, is included in the ABEONA beacons broadcast to all the neighbors. With the information collected from the neighbors, each vehicle calculates a weighted average of the density estimated locally and the one estimated by its neighbors. This estimation can be even furthermore improved by time-averaging it. A schematic representation of this part of algorithm is shown in Algorithm 4.

6.4.2.2 Flow estimation

Estimating the current vehicular flow in a distributed way is a more complex task, as vehicles need to agree on a common observation reference point. For this process we reuse the concept of *Virtual Induction Loop* (VIL) as already proposed in Section 6.2, a reference line placed across the road that emulates the behavior of a standard magnetic induction loop. The information about the position of the VIL of each road segment

Algorithm 4 Density estimation algorithm

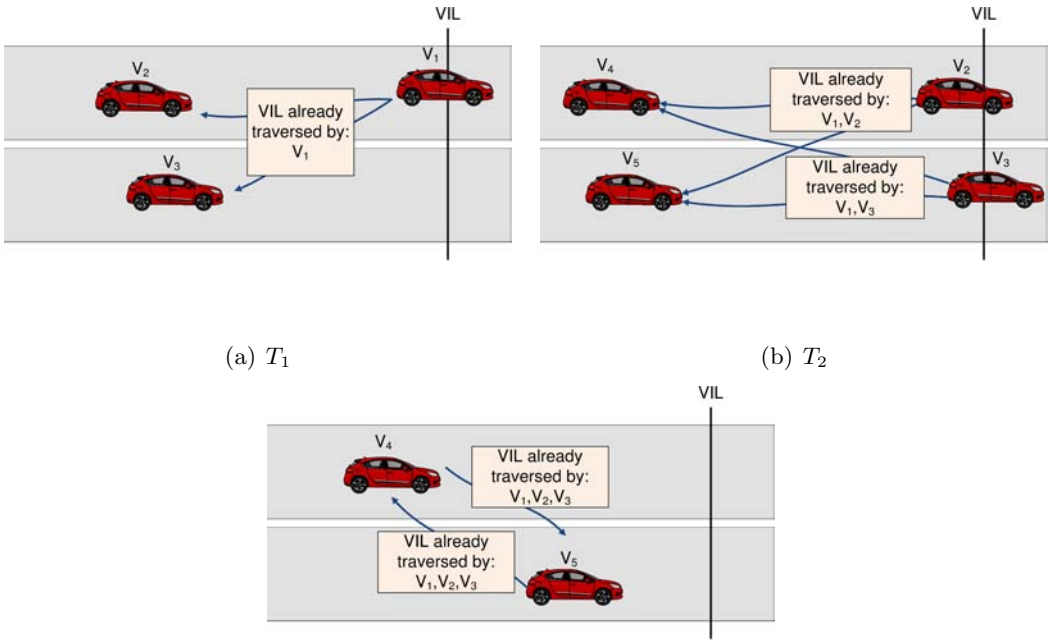
```
void evaluateDensityUsingBeacon(Beacon receivedBeacon)
{
  insertIfDistinct(receivedBeacon.getVehicleId(),
                   receivedBeacon.getPosition());
  distance = getFurthestAwayDistance();
  localDens = (localDens + (lastLocalDens *
            (now()-lastUpdated)/elapsedTimeInRoad);
  localDens = (localDens + beacon.getDens())/2

  elapsedTimeInRoad+=now()-lastUpdated();
  lastUpdated = now();
  lastLocalDens = distance/neighborTable.size();
}
```

(there is only one per *RoadId*) is included in the enriched digital map used by the in-vehicle navigation system.

ABEONAs flow estimation algorithm works as follows (see Figure 5.9 and Algorithm 5). Each vehicle maintains a VIL table, containing the list of the vehicles that have crossed the VIL reference line during the current epoch. When a vehicle traverses the VIL reference line, it adds its own *VehicleId* in its VIL table. Information from the VIL table is included in the ABEONA beacons broadcast by each vehicle, so neighbors can merge this information with the one contained in their local VIL table. By doing this, information is kept consistent among all the vehicles from the same road region. However, in order to limit the beacon size and avoid sending too much redundant information (beacons are usually sent every 1 or 2 seconds), only a small set of random entries from the VIL table is included in each beacon. As beacons are sent very often and each vehicle is receiving information from multiple sources, vehicles end up building a consistent VIL table or just with negligible deviations.

6.4.3 Historical data record management

Congestion prediction is based on the analysis of current and recent past traffic conditions. ABEONA tracks the density and flow values estimated per epoch, storing them in a historic epoch local table. This table contains values regarding the past `observationWindow` values. When entering a new epoch, vehicles insert their estimated flow and density values into the epoch table. If the total number of records in the epoch table is greater than `observationWindow`, the oldest one is discarded. This information is included in the beacons, and updated with the knowledge obtained from other beacons, in order to keep the information "stuck" to the corresponding `RoadId`. Since the local tables



(a) T_1 (b) T_2

(c) T_3

Figure 6.9: Cooperative estimation of the traffic flow

are shared among the different vehicles from the same road region, the information used by each participant tends to be consistent (even for a vehicle that has just entered into the region, as it learns the information kept by its neighbors from the received beacons), though minor deviations may appear. Vehicles merge the information locally maintained for each EpochId with the data received from beacons as follows.
If a vehicle does not have information about a particular epoch, then the data from the beacon is used and inserted in the local table. If the vehicle already has information for this epoch, a new density value is calculated as the average of the densities contained in the beacon and in the local table. Regarding the flow, the highest value from the ones found in the beacon and local table is used as estimated value. More details about this procedure can be found in Algorithm 6.

6.4.4 Beacon size considerations

ABEONA makes use of periodic beacons to share the information needed to forecast short-term traffic conditions. In order to minimize wireless channel congestion, the size of the beacons should be kept small. Each beacon basically contains the following information: VehicleId, RoadId, geographical position, EpochId, VIL table (containing the VehicleId of a random set of every car that has traversed the VIL reference line during current EpochId) and historic epoch table (containing the latest estimated *flow, densitity*

Algorithm 5 Flow estimation algorithm within current epoch

```
void processGPSOutput(GPSSentence sentence)
{
  vilRef = this.getCurrentRoad().getVilReference();
  this.currentPosition = sentence.getCurrentPosition()
  if (this.hasTraversed(vilRef))
    {
      alreadyTraversed = true;
    }
}

void evaluateFlowUsingBeacon(Beacon receivedBeacon)
{
  insertIntoVILTableIfDistinct(receivedBeacon.getVilInfo());
  includedInBeaconSet = pickRandomValues();
  if (this.alreadyTraversed == true )
  {
    includedInBeaconSet.add(this.getVehicleId());
  }
}
```

pairs). Considering the use of 64-bit fields for each of the identifiers and the geographical positions, a 10-entry VIL table size and a 20-minute epoch table size, the total ABEONA beacon size is 255 bytes. Using a 802.11g network card, with a transmission rate of 6 Mb/s, such a beacon (including MAC and IPv6 header) can be sent in 584 μs.

6.4.5 Lack of connectivity issues

One of the main barriers of VANET-based approaches is the intermittent connectivity problem, which may appear due to a lack of connectivity between two vehicles. This problem, already studied in Chapter 2, does not impact ABEONA as we can argue that in every situation in which traffic monitoring is worth being performed (i.e., it aids in predicting a traffic congestion event), the traffic density is such that guarantees that there is connectivity between the vehicles. This observation is supported by the empirical results shown in Figure 6.6(a): in the moments preceding an abrupt speed drop (i.e., when the traffic monitoring system should be operating), the vehicular flow is enough to maintain a continuous wireless coverage. On the other hand, in those situations where the vehicular density is not enough to maintain continuous connectivity it is very unlikely that a traffic jam is about to happen (unless the traffic jam is originated by non-predictable causes, such as accidents).

6.4.6 Forecasting future traffic conditions

It is known that certain values of flow and density necessarily lead to a sharp drop in the average speed. Each vehicle can autonomously estimate the future values of density

Algorithm 6 Historical data record management algorithm

```
void onEpochChange()
{
  insertIntoEpochTable(currentEpochId,localDens,localFlow);
  currentEpochId++;
  localDens=0;
  localFlow=0;
}


void processHistoricalData(Beacon receivedBeacon)
{
  for (EpochInfoEntry entry:receivedBeacon.getEpochInfoTable())
  {
    if (alreadyExist(entry))
    {
      newDens = (getEpochInfo(entry.getEpochId()).getDensity()
                + entry.getDensity())/2;
      newFlow = max(getEpochInfo(entry.getEpochId()).getDensity().
                    entry.getFlow());
      updateEpochInfo(entry.getEpochId(),newDens,newFlow);
    }
    else
    {
      insertEpochInfo(entry.getEpochId(),entry.getDensity(),
                      entry.getFlow());
    }
  }
}
```

and flow (q_e and k_e), using the current and historic epoch information, and compare them with a set of reference values (q_r and k_r). These reference values represent the threshold leading to a state transition between "Free flow" and "Synchronized traffic" states, and are characteristic for each road region, so they can be included in the enriched digital map. Note that these values are quite stable for a given road, so the typical update pattern of navigation software is sufficient to keep these parameters up-to-date. In order to forecast future density and flow values, ABEONA uses a linear prediction algorithm [14], namely the linear least squares.

Although more complex estimators could be used to furthermore refine the prediction outcome, this simple algorithm has proven to be more than sufficient (as shown in the next section). When a vehicle detects that $q_e > q_r$ and $k_e > k_r$ this means that a traffic state change is forecasted, triggering a warning message to be sent to the traffic control center. This message (which just contains the RoadId and VehicleId) can be sent using a cellular network or, if available, using the VANET through an RSU. The traffic control center keeps track of the warning messages received, marking a given road region to be "likely to become congested" if more than a preconfigured number of warning messages is received from different vehicles over a preconfigured window time. When a road region is marked as "likely to become congested", the traffic control center shares this information

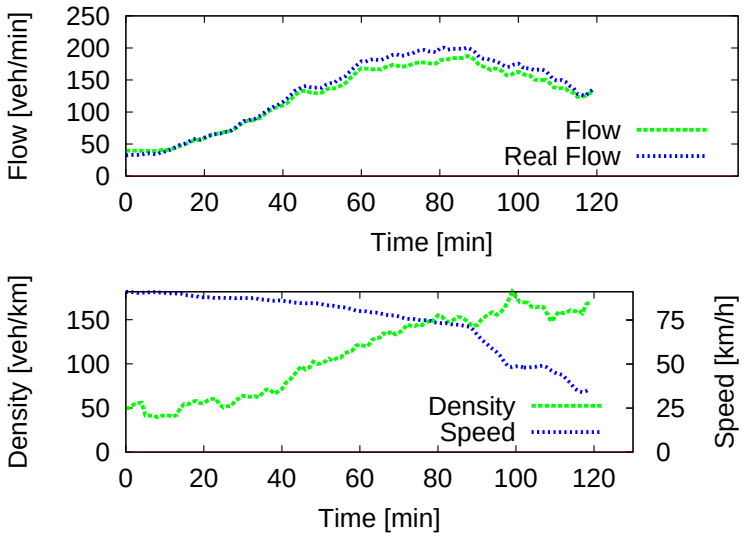


Figure 6.10: ABEONA flow estimation

with all the vehicles that can benefit from knowing it, by using the cellular network, the VANET or both.

6.5 Experimental evaluation

In order to perform a validation and evaluation of the performance of ABEONA, and given the difficulties associated to conducting real experiments of a vehicular mechanism that involves real-life traffic conditions, we have decided to use trace-driven simulations. The simulation environment is the following: ABEONA has been implemented in OMNeT++ and SUMO has been used to emulate vehicles behavior. A three-lane wide, 1 km-length road region has been used, placing the VIL reference line at a fixed point on each simulation run. Vehicles are equipped with a standard IEEE 802.11g card, being their initial positions and speeds taken from real traces from M30 road, provided by the Madrid City Hall.

We first assess ABEONAs capability of estimating traffic flow. The upper part of Figure 6.10 shows the real flow and the one estimated by ABEONA before and during a traffic congestion situation. The lower part of the figure depicts the vehicular density. Speed is also depicted, in order to better identifying when a traffic congestion event happens.

The key ABEONA performance metric is its capacity to forecast future traffic state transitions based on the estimated flow and density parameters. We next evaluate this performance using 20 minutes of previous monitored data and a prediction window of 10 minutes, this means that ABEONA needs a history of 20 minutes of flow and density estimated values to be able to predict their future values with 10 minutes in advance.

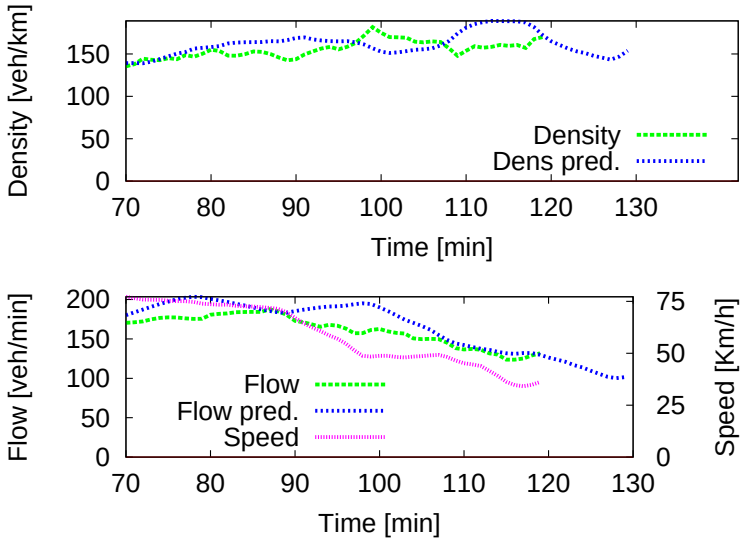


Figure 6.11: ABEONA flow and density prediction

Figure 6.11 shows the estimated and predicted density (top) and flow (bottom), as well as the speed (to help identifying the transition to Synchronized traffic traffic state). It can be observed that ABEONA predicted density and flow closely follow the estimated values, therefore enabling the forecast of future traffic conditions.
Last, but not least, and in order to help understanding the benefits that ABEONA could bring in, we have studied the following simple use case scenario. We have simulated a typical day-life commute route between the city of Aranjuez and Getafe in Spain, using two different roads: through A4 and R4. These two roads are parallel, R4 is a toll road longer than A4, so under low traffic conditions, the fastest itinerary is the one via A4 (total time: 27 minutes), but as soon as the traffic increases, the total time might get to 46 minutes. If ABEONA were in use, vehicles could have predicted the traffic congestion and react deciding to use R4 instead, leading to a total commuting time of 29 minutes (36% saving of time).

There are two parameters that affect ABEONAs performance: the size of the past data used to make the prediction, and the time in advance that this prediction is made (prediction window). In the results shown in Figure 6.10, we set up these values to be 20 and 10 minutes respectively, which are the ones providing best results for this road. We next evaluate the impact of using different values. In Figure 6.12 we fix the size of the historical data used to 30 minutes and modify the prediction window, while in Figure 6.13 we do the same for the case of 15 minutes. The figures also show the reference values q_r and d_r used by ABEONA to trigger a congestion event prediction. Results show that using longer past data sets makes the prediction algorithm to react more slowly, which may cause that the prediction of traffic congestion is performed late. On the other hand, using shorter data sets leads to the opposite effect, as short temporal conditions heavily

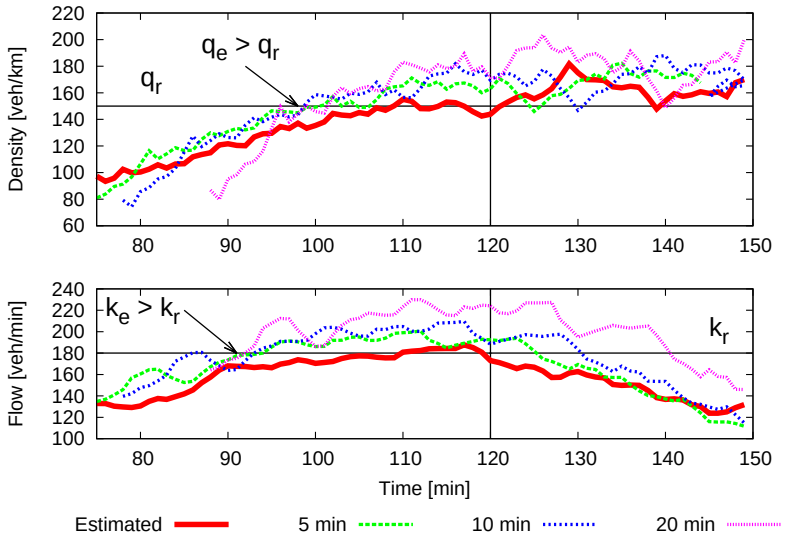


Figure 6.12: Impact of past data size (30 minutes) and prediction window

affect in the prediction outcome, thus resulting in potential false alarms. Finally, as expected, shorter prediction windows lead to more accurate predictions, but at the cost of leaving less time to react and re-plan the route.

6.6 Summary

In this chapter we detailed the operation of two services for traffic efficieny (one for traffic monitoring and one for traffic prediction) using vehicular communications. We propose the use of virtual induction loops (VIL) to replace the legacy induction loops currently used by Public Transport Authorities. The resulting advantages from the deployment of VIL consists in an easier observation of more roads than those currently equipped with a monitoring infrastructure, without requiring additional costs (i.e., nowadays only major urban cities can afford monitoring infrastructure). However, one of the goal of Intelligent Transportation Systems is to forecast future traffic congestions, hence, in this Chapter we also proposed ABEONA, a traffic congestion prediction framework. By observing two fundamental traffic variables (*Flow* and *Density*) the future values of the vehicular traffic state can be predicted. We validated our traffic efficiency algorithms by simulation. In both of the cases, the introduced error compared to the legacy methodologies and the wireless overhead are negligible. Moreover, ABEONA can successfully forecast future (20 min.) traffic conditions and warn drivers against a forthcoming traffic jam.

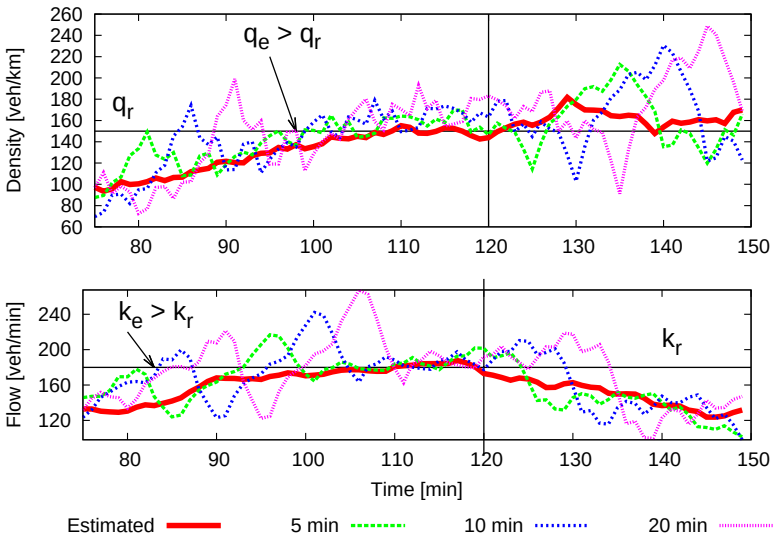


Figure 6.13: Impact of past data size (15 minutes) and prediction window

Chapter 7

Conclusion and future work

The effort spent on vehicular networking by the research community in the past few years demonstrates how this technology is envisioned to play a key role in the future of Intelligent Transportation Systems (ITS). The introduction of the vehicular ad-hoc network (VANET) paradigm enables new services and applications that will make trips safer, more comfortable and with a reduced ecological footprint.
However, despite the increasing interest on VANETs, many important aspects are still yet to be defined. Inter-vehicular communications were firstly proposed for safety-related purposes, so this family of solutions is already in an advanced stage of development. On the other hand, both infotainment (i.e., providing classic IP connectivity to the vehicular environment) and traffic efficiency services have increased their relevance recently. In this thesis we have proposed optimizations for infotainment and traffic efficiency.
Regarding infotainment, this thesis proposes a fully-integrated solution for granting Internet access from vehicular networks. More in detail, we pointed out three fundamental aspects that a comprehensive solution for VANET-based internetworking must address:

- **Address autoconfiguration:** Vehicular networks will be composed by a large number of nodes with high mobility characteristics. Therefore, assigning a valid

IPv6 address to each node without incurring in high routing inefficiencies is a task difficult to accomplish. GeoSAC [3] is an address autoconfiguration protocol that extends the standard IPv6 SLAAC to the vehicular environment that has been lately standardized by ETSI as ETSI SLAAC. In this thesis we first analytically evaluated this solution in terms of address configuration time, and then improved it using an overhearing technique. The results coming from extensive simulations and implementation in off-the-shelf devices show how our analytical model matches realistic scenarios and how our overhearing mechanism can tear down the address configuration times down to zero seconds in many cases. The solution is detailed in Chapter 2 and was published in [15, 16].

- **Efficient routing:** Tackling the routing problem in vehicular environments is a task that has to deal with many issues as high node mobility or short-lived links that make standard MANET-based solutions difficult to adapt. Road side units are central entities when providing internet access from vehicles, as all the exchanged data traffic has to pass through them. We exploited this singularity and proposed TREBOL, a tree based routing protocol for VANETs. Simulation results show that our proposal outperforms a greedy-geographic routing algorithm in terms of packet delivery ratio and introduces less overhead. TREBOL control overhead can be furthermore reduced using TREBOL in conjunction with the ETSI SLAAC, as the same set of control messages can be used also to provide address autoconfiguration capabilities within the VANET. Moreover, we implemented TREBOL using standard Linux PCs, showing that our proposed algorithm can deal with multiple TCP and UDP flows. TREBOL is explained in detail in Chapter 3 and was published in [17, 18].

- **Mobility management:** People need connectivity everywhere at every time with the best possible conditions according to the *always best connected* paradigm. Currently, there are two available technologies that can be used to achieve internet connectivity: wireless multi-hop VANETs and cellular networks (i.e., 3G or the forthcoming LTE). While cellular networks can achieve always-on connectivity, their current congestion state suggests their offloading to guarantee a better utilization. Moreover, user devices placed inside vehicles should enjoy seamless connectivity, even when switching from an access technology to another. Our proposal, SILVIO, is based on the key idea that vehicles use cellular connectivity to guarantee always on connectivity, but opportunistically offload some non-critical flows if an Internet connection through a multi-hop wireless path becomes available. SILVIO employs standardized technologies such as flow mobility, and a vehicular-aware version of the 802.21 MIH service to achieve smart handoff procedures. Results obtained using a trace-driven simulator show that users can increase the time spent using the

multi-hop wireless connectivity up to 80%, with a negligible control overhead. We described the full solution in Chapter 4 (published in [19]).

All the simulators developed for the evaluation of the proposed protocols are trace-driven and are fed with real vehicular traces coming from fixed observation points in the municipality of Madrid. Modeling the interarrival times between consecutive vehicles has been a key point in our research, especially in the work developed in Chapter 2. Therefore, a second research line arose from here, with the goal of understanding the complex interactions among vehicles traveling in the same stretch of road, and their possible applications in the field of inter-vehicle communications and traffic efficiency:

- **Vehicular traces characterization:** Characterizing the interarrival times between vehicles is a key factor for designing networking protocols for vehicular environments or implementing an auxiliary infrastructure to increase vehicular connectivity. Previous work that already aimed at this goal provided different solutions proposing, for example, the exponential or the lognormal distribution to model the time distance between two vehicles. However, we analyzed the vehicular traces we used to feed our simulators to give a more comprehensive explanation to these phenomena. We found that, in general, the vehicle arrivals do not follow a Poisson process, and their distribution is not i.i.d. The interarrival times, hence, cannot be considered as memoryless sequence as there is a speed-correlation between vehicles. According to this, we identified two classes of vehicles: the "bursty ones" (traveling together at short distance, with similar speeds) and the "isolated ones" (traveling at larger distance, with uncorrelated speeds). After this analysis, we found that the first class can be modeled using a Gaussian distribution and the exponential distribution can be used to fit the second one. Finally, we proposed a mixture model that can better fit the empirical data compared to the previously proposed ones. Chapter 5 details the developed analysis, including the used EM algorithm used to fine-tune the mixture model parameters. This work was published in [117].
  The interactions among vehicles are useful also from a traffic efficiency perspective. There exists a broad literature about the physics of vehicular traffic, either for detection of traffic jams or short-term forecasting. With the introduction of the VANET paradigm, new tools for data gathering and analysis are available. In this thesis we detailed two proposals that employs vehicular communications to improve traffic efficiency.

- **Traffic monitoring and short-term prediction:** Keeping the traffic conditions monitored is a crucial task to perform in order to optimize the roads utilization. Although traditional solutions are already running with profit, in this thesis we proposed a VANET-based solution that allows scalable monitoring without incurring into high installation costs. Simulation results show that both the introduced

error compared to the legacy methodologies and the wireless overhead are negligible. Moreover, this solution provides a configurable level of granularity, providing a higher data resolution just when and where it is needed. However, a traffic management system should be able to predict traffic congestion situations. Therefore, in this thesis we also proposed ABEONA, a traffic congestion prediction framework, based on the cooperative monitoring and estimation of two of the three principal variables that define macroscopic traffic behavior: density and flow. We have provided a comprehensive overview of the pillars of the mathematical models that define the relationships between these three variables and the possible traffic states: "Free flow", "Synchronized traffic" and "Traffic jam", which furthermore allows performing short-term traffic state prediction. Based on this theoretical basis we designed a VANET-assisted mechanism that enables vehicles equipped with a GPS navigation system to estimate current flow and density, by cooperating and sharing locally monitored information. The availability of these accurate estimated parameters, allows vehicles to perform a short-term forecast of a potential congestion event. The simulation-based validation analysis, using real traffic traces, supports the feasibility and correctness of our proposal, which can help drivers to properly re-plan their routes, leading to important savings in commuting time. We described in detail these proposals in Chapter 6.

There are many research directions arising from this thesis. From the infotainment point of view, we plan to extend our routing protocol (TREBOL) to efficiently support multicast traffic as it is the default paradigm for Internet-to-vehicle communications, and therefore complement it to fully support infotainment traffic in vehicular networks.
We also intend to analyze more vehicular data traces to furthermore refine our interarrival times model and find other interesting insights such as how to detect and predict the moment of the decongestion of a road. This would provide additional valuable information to the one already provided by ABEONA, and improve the car navigation systems route planning algorithms.
Finally, we foresee to apply the vehicular traffic state knowledge to the proposed infotainment protocols, in order to choose the best configuration parameters according to the current and short term traffic conditions. For example, a "network controlled" version of SILVIO can be implemented, using the vehicular traffic state knowledge to smartly switch between different handover policies.

References

[1] M. Fazio, C. Palazzi, S. Das, and M. Gerla, "Vehicular Address Configuration," in *Proc. of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet).* GLOBECOM 2006 San Francisco, CA, USA, 2006.

[2] B. K. Mohandas and R. Liscano, "IP Address Configuration in VANET using Centralized DHCP," in *Proc. of the 33rd IEEE Local Computer Networks Conference.* IEEE, 2008.

[3] R. Baldessari, C. J. Bernardos, and M. Calderon, "GeoSAC - Scalable Address Autoconfiguration for VANET Using Geographic Networking Concepts," in *PIMRC 2008*, Cannes (France), September 2008.

[4] J. Choi, Y. Khaled, M. Tsukada, and T. Ernst, "IPv6 support for VANET with geographical routing," in *The 8th International Conference on Intelligent Transport System Telecomunications (ITST 2008)*, October 2008.

[5] I. Leontiadis and C. Mascolo, "GeOpps: Opportunistic geographical routing for vehicular networks," in *In Proceedings of the IEEE Workshop on Autonomic and Opportunistic Communications*, 2007.

[6] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in *IEEE INFOCOM*, vol. 3, 1997, pp. 1405–1413.

[7] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting vehicular mobility in urban multi-hop wireless networks," in *Proceeding of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 54–66.

[8] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services.* New York, NY, USA: ACM, 2010, pp. 209–222.

[9] S. Yamada, "The strategy and deployment plan for vics," *IEEE Communications Magazine*, vol. 34, no. 10, pp. 94–97, October 1996.

[10] C. Liu, "The Case for Vehicular Visible Light Communication (V2LC): Architecture, Services and Experiments," M.S. Thesis, Rice University, Houston, TX,, June 2010.

[11] S. Okada, T. Yendo, T. Yamazato, T. Fujii, M. Tanimoto, and Y. Kimura, "On-vehicle Receiver for Distant Visible Light Road-to-Vehicle Communication," in *IEEE Intelligent Vehicles Symposium*, Shaanxi (China), July 2009.

[12] H. Reijmers and R. Prasad, "The influence of vehicle distribution models on packet success probability on a three lane motorway," in *48th IEEE Vehicular Technology Conference, 1998. VTC 98*, vol. 3, 1998.

[13] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1555, October 2007.

[14] R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg, "Different methods of traffic forecast based on real data," *European Journal of Operational Research*, vol. 155, no. 3, pp. 558 – 568, 2004, ¡ce:title¿Traffic and Transportation Systems Analysis¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221703004788

[15] M. Gramaglia, I. Soto, C. Bernardos, and M. Calderon, "Overhearing-assisted optimization of address autoconfiguration in position-aware vanets," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 7, pp. 3332 –3349, sept. 2011.

[16] M. Gramaglia, C. Bernardos, I. Soto, M. Calderon, and R. Baldessari, "Ipv6 address autoconfiguration in geonetworking-enabled vanets: characterization and evaluation of the etsi solution," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 19, 2012.

[17] M. Gramaglia, M. Calderon, and C. Bernardos, "Trebol: Tree-based routing and address autoconfiguration for vehicle-to-internet communications," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd.* IEEE, 2011, pp. 1–5.

[18] M. Gramaglia, C. Bernardos, M. Calderon, and A. de la Oliva, "Performance evaluation of a tree-based routing and address autoconfiguration for vehicle-to-internet communications," in *ITS Telecommunications (ITST), 2011 11th International Conference on.* IEEE, 2011, pp. 45–50.

[19] M. Gramaglia, C. Bernardos, and M. Calderon, "Seamless internet 3g and opportunistic wlan vehicular connectivity," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, p. 183, 2011.

[20] H. Reijmers and R. Prasad, "A Model Based Connectivity Improvement Strategy for Vehicular Ad hoc Networks," in *IEEE Vehicular Technology Conference - VTC 1996)*, Ottawa (Canada), September 1996, pp. 1785–1789.

[21] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1555, October 2007.

[22] B. Kerner, "The physics of traffic," *Physics world*, pp. 25–30, 1999.

[23] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, September 2007.

[24] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Standard), September 2007.

[25] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[26] C. J. Bernardos, M. Calderon, and H. Moustafa, "Survey of IP address auto-configuration mechanisms for MANETs," IETF, draft-bernardos-manetautoconf-survey-04.txt (work-in-progress), November 2008.

[27] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network architecture," ETSI TS 102 636-3 V1.1.1, March 2010.

[28] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, January 2005.

[29] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-part 1: Media-Independent Functionality," ETSI TS 102 636-4-1 (work in progress), May 2011.

[30] ——, "Intelligent Transport Systems (ITS); Vehicular Communications; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols," ETSI TS 102 636-6-1 V1.1.1, March 2011.

[31] J. Ott and F. Kutscher, "The Drive-Thru Architecture: WLAN-based Internet Access on the Road," in *Proceedings VTC Fall*, May 2004.

[32] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar, "Support of Anonymity in VANETs – Putting Pseudonymity into Practice," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Hong Kong, March 2007.

[33] C. Harsch, A. Festag, and P. Papadimitratos, "Secure Position-Based Routing for VANETs," in *VTC Fall*, Baltimore, MD, USA, October 2007.

[34] S. Yousefi, E. Altaian, and R. El-Azouzi, "Study of connectivity in vehicular ad hoc networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, Apr. 2007, pp. 1 –6.

[35] L. Bain and D. Weeks, "A note on the truncated exponential distribution," *The Annals of Mathematical Statistics*, pp. 1366–1367, 1964.

[36] R. Haberman, *Mathematical models: mechanical vibrations, populations dinamics and traffic flows : an introduction to applied mathematics*. SIAM, Society for Industrial and Applied Mathematics, 1998.

[37] A. Festag, R. Baldessari, and H. Wang, "On Power-Aware Greedy Forwarding in Highway Scenarios," in *Proceedings of 4th International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, March 2007, pp. 31–36.

[38] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[39] Y.-H. Han, J. Choi, and S.-H. Hwang, "Reactive Handover Optimization in IPv6-Based Mobile Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 9, pp. 1758–1772, September 2006.

[40] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision," in *Simutools '08: Proceedings of the 1st international conference on*

Simulation tools and techniques for communications, networks and systems & workshops, 2008, pp. 1–8.

[41] J. K. Cavers, *Mobile Channel Characteristics*. Kluwer Academic Publishers, 2000.

[42] M. K. Simon and M.-S. Alouini, "Digital communications over fading channels (m.k. simon and m.s. alouini; 2005) [book review]," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 3369–3370, 2008.

[43] C. J. Bernardos, I. Soto, M. Calderon, F. Boavida, and A. Azcorra, "VARON: Vehicular Ad-hoc Route Optimisation for NEMO," *Computer Communications*, vol. 30, no. 8, pp. 1765 – 1784, June 2007.

[44] Q. Mussabbir, W. Yao, Z. Niu, and X. Fu, "Optimized FMIPv6 using IEEE 802.21 MIH services in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3397–3407, 2007.

[45] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance comparison of two location based routing protocols for ad hoc networks," 2002, pp. 1678–1687.

[46] F. J. Ros, V. Cabrera, J. A. Sanchez, J. A. Martinez, and P. M. Ruiz, *Vehicular Networks: Techniques, Standards and Applications Book*. CRC Press, 2009, ch. Routing in Vehicular Networks.

[47] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243–254.

[48] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Lazy cross-link removal for geographic routing," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2006, pp. 112–124.

[49] C. Lochert, H. Hartenstein, J. Tian, H. Fuessler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *In Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 156–161.

[50] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," in *4th ACM SenSys*, Boulder, CO, November 2006.

[51] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular content delivery using wifi," in *14th ACM MOBICOM*, San Francisco, CA, September 2008.

[52] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive WiFi Connectivity for Moving Vehicles," in *Proc. ACM SIGCOMM*, August 2008.

[53] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding wifi-based connectivity from moving vehicles," in *In IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[54] P. Lutterotti, G. Pau, D. Jiang, M. Gerla, and L. Delgrossi, "C-vet, the ucla vehicular testbed: An open platform for vehicular networking and urban sensing," in *International Conference on Wireless Access for Vehicular Environments (WAVE 2008)*, 2008.

[55] J. Salim, H. Khosravi, A. Kleen, and A. Kuznetsov, "Linux Netlink as an IP Services Protocol," RFC 3549, July 2003.

[56] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.

[57] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, April 1998.

[58] I. Goncalves, S. Pfeiffer, and C. Montgomery, "Ogg Media Types," RFC 3534, September 2008.

[59] L. Barbato, "RTP Payload Format for Vorbis Encoded Audio," RFC 5215, August 2008.

[60] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on tcp performance," in *Mobile Computing and Networking*, pp. 77–89.

[61] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, pp. 64–74, July 2008.

[62] J. Wortham, "Customers Angered as iPhones Overload AT&T," http://www.nytimes.com/2009/09/03/technology/companies/03att.html.

[63] D. Kellogg, "Average U.S. Smartphone Data Usage Up 89% as Cost per MB Goes Down 46%," http://bit.ly/3GCongestion.

[64] Y. Zang, S. Sories, G. Gehlen, and B. Walke, "Towards a European Solution for Networked Cars - Integration of Car-to-Car technology into cellular systems for vehicular communication in Europe," Speech, ITU, Geneva, Switzerland, p. 14, Mar 2009. [Online]. Available: http://www.comnets.rwth-aachen.de

[65] I. Lequerica, P. M. Ruiz, and V. Cabrera, "Improvement of vehicular communications by using 3G capabilities to disseminate control information," *IEEE Network*, vol. 24, pp. 32–38, January 2010.

[66] D. Telekom, "Deutsche Telekom and iPass are partnering to launch a global WiFi solution meeting the needs of smartphone users from carriers worldwide," http://www.telekom.com/dtag/cms/content/dt/en/1025630.

[67] W. Enkelmann, "FleetNet - applications for inter-vehicle communication," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003, pp. 162–167.

[68] W. J. Maen M. Artimy, William Robertson, "Vehicular Ad Hoc Networks: An Emerging Technology Toward Safe and Efficient Transportation," in *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks*, A. Boukerche, Ed. Wiley, 2008, p. Ch. 14.

[69] J. Marquez-Barja, C. Calafate, J.-C. Cano, and P. Manzoni, "Multi-Layer Performance Evaluation of a Content Delivery Framewirk for Urban Vehicular Networks," in *Proc. of IEEE International Conference on Communications Workshops (ICC 2010).* IEEE, 2010.

[70] A. B. Reis, S. Sargento, and O. K. Tonguz, "On the Performance of Sparse Vehicular Networks with Road Side Units," in 2011 IEEE 73rd Vehicular Technology Conference," in *IEEE Vehicular Technology Conference (VTC)*, May 2011.

[71] W. Viriyasitavat, O. Tonguz, and F. Bai, "Dynamics of Network Connectivity in Urban Vehicular Networks," *IEEE Journal on Selected Areas of Communications, Special Issue on Vehicular Communications and Networkings*, vol. 29, no. 3, pp. 515 – 533, Mar 2010.

[72] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. N. ami, "Multiple Care-of Addresses Registration," RFC 5648 (Proposed Standard), Internet Engineering Task Force, Oct. 2009.

[73] G. Tsirtsis, G. Giarreta, H. Soliman, and N. Montavont, "Traffic Selectors for Flow Bindings," RFC 6088 (Proposed Standard), Internet Engineering Task Force, Jan. 2011.

[74] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, "Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support," RFC 6089 (Proposed Standard), Internet Engineering Task Force, Jan. 2011.

[75] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, "An overview of ieee 802.21: media-independent handover services," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 96–103, 2008.

[76] K. Taniuchi, Y. Ohba, V. Fajardo, S. Das, M. Tauil, Y. Cheng, A. Dutta, D. Baker, M. Yajnik, and D. Famolari, "IEEE 802.21: media independent handover: features, applicability, and realization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 112–120, 2009.

[77] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros, "Experimental Study on the Impact of Vehicular Obstructions in VANETs," in *2nd IEEE Vehicular Networking Conference (VNC 2010)*. Jersey City, NJ: IEEE, December 2010, pp. 338 – 345.

[78] A. U. Joshi and P. Kulkarni, "Vehicular wifi access and rate adaptation," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 423–424. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851243

[79] M. Boban, T. T. V. Vinhoza, J. Barros, M. Ferreira, and O. K. Tonguz, "Impact of Vehicles as Obstacles in Vehicular Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 15–28, January 2011.

[80] S. Annese, C. Casetti, C.-F. Chiasserini, N. D. Maio, A. Ghittino, and M. Reineri, "Seamless Connectivity and Routing in Vehicular Networks with Infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 501–514, 2011.

[81] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)," Internet Engineering Task Force, apr 2008. [Online]. Available: http://tools.ietf.org/html/draft-openmesh-b-a-t-m-a-n-00

[82] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular WiFi access," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09. New York, NY, USA: ACM, 2009, pp. 263–276.

[83] P. Deshpande, X. Hou, and S. R. Das, "Performance comparison of 3G and metro-scale WiFi for vehicular network access," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 301–307.

[84] Y. Xiao, K. K. Leung, Y. Pan, and X. Du, "Architecture, mobility management, and quality of service for integrated 3G and WLAN networks," *Journal Wireless Communications & Mobile Computing*, vol. 5, no. 7, pp. 805–823, Nov 2005.

[85] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient mobility management for vertical handoff between WWAN and WLAN," *Communications Magazine*, vol. 41, no. 11, pp. 102–108, Nov 2003.

[86] T. Melia, C. J. Bernardos, A. de la Oliva, F. Giust, and M. Calderon, "IP Flow Mobility in PMIPv6 Based Networks: Solution Design and Experimental Evaluation," *Wireless Personal Communications*.

[87] A. de la Oliva, C. J. Bernardos, M. Calderon, T. Melia, and J. C. Zuniga, "IP flow mobility: smart traffic offload for future wireless networks," *Communications Magazine*, vol. 49, no. 10, pp. 124–132, Oct 2011.

[88] S. Cespedes, X. Shen, and C. Lazo, "IP mobility management for vehicular communication networks: challenges and solutions," *Communications Magazine*, vol. 49, no. 5, pp. 187–194, May 2011.

[89] "IEEE 802.11p/D2.01, Draft Amendement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Wireless Access in Vehicular Environments," March 1997.

[90] R. Haberman, *Mathematical models : mechanical vibrations, population dynamics, and traffic flow : an introduction to applied mathematics / Richard Haberman.* Englewood Cliffs, N.J. : Prentice-Hall, 1977.

[91] Y. Yang, Z. Mi, J. Yang, and G. Liu, "A model based connectivity improvement strategy for vehicular ad hoc networks," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, 2010, pp. 1 –5.

[92] S. M. Ross, *Introduction to Probability Models, Ninth Edition.* Orlando, FL, USA: Academic Press, Inc., 2006.

[93] J. H. Pollard, *A Handbook of Numerical and Statistical Techniques.* CUP Archive, 1979.

[94] J. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," ICSI TR-97-021, Tech. Rep., 1998.

[95] J. A. Hernández and I. W. Phillips, "Weibull mixture model to characterise end-to-end internet delay at coarse time-scales," *IEE Proc. Communications*, vol. 153, no. 2, pp. 295–304, April 2006.

[96] J. Durbin, "Kolmogorov-smirnov tests when parameters are estimated," in *Empirical Distributions and Processes*, ser. Lecture Notes in Mathematics, P. Gaenssler and P. Rvsz, Eds. Springer Berlin / Heidelberg, 1976, vol. 566, pp. 33–44, 10.1007/BFb0096877.

[97] B. Kerner, C. Demir, R. G. Herrtwich, S. L. Klenov, H. Rehborn, M. Aleksi, and A. Haug, "Traffic state detection with floating car data in road networks," in *Proceedings of IEEE Intelligent Transportation Systems, 2005.* IEEE, 2005, pp. 44–49. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2005.1520133

[98] TomTom, "How tomtom's hd traffic™and iq routes™data provides the very best routing," White paper, 1991, available online (16 pages). [Online]. Available: www.tomtom.com/lib/doc/download/HDT_White_Paper.pdf

[99] R. Anderson, "Electromagnetic loop vehicle detectors," *Vehicular Technology, IEEE Transactions on*, vol. 19, no. 1, pp. 23–30, 1970.

[100] G. Leduc, "Road traffic data: Collection methods and applications," EC JRC IPTS: Working Papers on Energy, Transport and Climate Change. JRC 47967, 2008, available online (55 pages). [Online]. Available: http://ftp.jrc.es/EURdoc/JRC47967.TN.pdf

[101] L. Garelli, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Mobsampling: V2v communications for traffic density estimation," in *VTC Spring*. IEEE, 2011, pp. 1–5.

[102] A. Skordylis and N. Trigoni, "Efficient data propagation in traffic-monitoring vehicular networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 680 –694, sept. 2011.

[103] ETSI, "Intelligent Transport Systems (ITS); Communications Architecture," ETSI EN 302 665: V1.1.1), September 2009.

[104] D. Jiang and L. Delgrossi, "Ieee 802.11p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, may 2008, pp. 2036 –2040.

[105] ETSI, "Intelligent Transport Systems (ITS); Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service," ETSI TS 102 637-2: V1.2.1, March 2011.

[106] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[107] R. Bauza, Bauza, J. Gozalvez, Gozalvez, and J. Sanchez-Soriano, Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications," in *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks*, ser. LCN '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 606–612. [Online]. Available: http://dx.doi.org/10.1109/LCN.2010.5735780

[108] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1537–1548, 2011.

[109] M. J. Lighthill and G. B. Whitham, "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads," *Royal Society of London Proceedings Series A*, vol. 229, pp. 317–345, May 1955.

[110] S. P. Hoogendoorn and P. H. L. Bovy, "State-of-the-art of vehicular traffic flow modelling," in *Delft University of Technology, Delft, The*, 2001, pp. 283–303.

[111] P. Benedetto and T. Andrea, "Vehicular traffic: A review of continuum mathematical models," *Encyclopedia of Complexity and Systems Science*, vol. 22, pp. 9727–9749, 2009. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-30440-3_576

[112] B. D. Greenshields, "A study of traffic capacity," vol. 14, 1934, pp. 448–477.

[113] B. Kerner, *The physics of traffic*. Springer Verlag, 2004.

[114] H. Kanoh, T. Furukawa, S. Tsukahara, K. Hara, H. Nishi, and H. Kurokawa, "Short-term traffic prediction using fuzzy c-means and cellular automata in a wide-area road network," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, sept. 2005, pp. 381 – 385.

[115] T. Thomas, W. Weijermars, and E. van Berkum, "Predictions of urban volumes in single time series," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 71 –80, march 2010.

[116] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent content-based information dissemination in hybrid vehicular networks," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, march 2009, pp. 1 –10.

[117] M. Gramaglia, P. Serrano, J. Hernández, M. Calderon, and C. Bernardos, "New insights from the analysis of free flow vehicular traffic in highways," *IEEE WoWMoM*, 2011.